

**KOMPRESI CITRA MEDIS SINAR-X DADA  
MENGGUNAKAN ALGORITMA *FRAKTAL*  
DAN *ARITHMETIC CODING***

**SKRIPSI**

**WINDY ADINDA  
191401028**



**PROGRAM STUDI S-1 ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2023**

**KOMPRESI CITRA MEDIS SINAR-X DADA  
MENGGUNAKAN ALGORITMA *FRAKTAL*  
DAN *ARITHMETIC CODING***

**SKRIPSI**

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah  
Sarjana Komputer

**WINDY ADINDA**  
**191401028**



**PROGRAM STUDI S-1 ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2023**

## PERSETUJUAN

Judul : KOMPRESI CITRA MEDIS SINAR-X DADA  
 MENGGUNAKAN ALGORITMA *FRAKTAL*  
 DAN *ARITHMETIC CODING*

Kategori : SKRIPSI

Program Studi : SARJANA (S-1) ILMU KOMPUTER

Nama : WINDY ADINDA

Nomor Induk Mahasiswa : 191401028

Program Studi : SARJANA (S-1) ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA

Medan, 06 Desember 2023

Disetujui oleh:  
Pembimbing II

Hayatunnufus S.Kom M.Cs  
NIP. 199207192020012001

Disetujui oleh:  
Pembimbing I

Dr. Maya Silvi Lydia B.Sc., M.Sc.  
NIP. 197401272002122001

Diketahui/disetujui oleh  
Program Studi S-1 Ilmu Komputer



KOMPRESI CITRA MEDIS SINAR-X DADA  
MENGGUNAKAN ALGORITMA *FRAKTAL*  
DAN *ARITHMETIC CODING*

SKRIPSI

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan masing-masing yang telah disebutkan sumbernya.

Medan, 06 Desember 2023

WINDY ADINDA  
191401028

## PENGHARGAAN

Penulis ingin menyampaikan ungkapan terima kasih dan rasa syukur kepada Allah Subhanahu Wa Ta'ala yang telah memberikan rahmat dan kekuatan-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul Kompresi Citra Medis Sinar-x Dada Menggunakan Algoritma Fraktal dan Arihtmetic Coding.

Dengan selesainya skripsi ini, penulis ingin mengungkapkan rasa terima kasih atas dukungan, saran, bantuan, dan motivasi yang diberikan oleh pihak-pihak yang terlibat, yakni:

1. Allah Subhanahu Wa Ta'ala dan Nabi Muhammad Sallalahu 'Alaihi Wasallam.
2. Bapak Dr. Muryanto Amin, S.Sos., M.Si sebagai Rektor Universitas Sumatera Utara.
3. Ibu Maya Silvi Lydia B.Sc., M.Sc sebagai Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara dan sebagai Dosen Pembimbing I yang dengan penuh kesabaran dan tulus memberikan bimbingan, ilmu, kritik, saran dan dorongan motivasi selama proses penyelesaian skripsi ini.
4. Ibu Dr. Amalia, S.T., M.T sebagai Ketua Program Studi S1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
5. Ibu Hayatunnufus S.Kom M.Cs sebagai Dosen Pembimbing II dengan penuh kesabaran dan tulus memberikan bimbingan, ilmu, kritik, saran dan dorongan motivasi selama proses penyelesaian skripsi ini.
6. Keluarga tercinta, Ibu, Ayah, Kakak dan Adik-adik yang dengan penuh sabar dan kasih sayang memberikan motivasi, semangat, dukungan moral dan material serta mendoakan kesuksesan penulis.
7. Seluruh angkatan 2019 S-1 Ilmu Komputer yang telah berjuang bersama di bangku perkuliahan dengan penulis.

8. Serta kepada semua pihak yang sudah membantu dalam memberikan dorongan motivasi kepada penulis meskipun semua tidak bisa disebutkan secara individual.

Semoga skripsi ini bisa memberikan manfaat bagi penulis dan semua pembacanya. Kiranya Allah Subhanahu Wa Ta'ala memberikan keberkahan kepada semua yang terlibat.

Medan, 06 Desember 2023

Penulis

Windy Adinda

## ABSTRAK

Diagnostik citra medis modern melibatkan sekumpulan besar citra medis yang sering kali didasarkan pada sinar-x yang dilakukan untuk melihat kondisi bagian dalam tubuh. Di era kesehatan digital, karena sebagian dari citra medis ini berukuran sangat besar, menyimpan dan mentransfernya selalu menjadi masalah penting bagi sistem informasi layanan kesehatan untuk menangani ukuran file pemeriksaan digital yang semakin meningkat. Sehingga, memerlukan penyimpanan data yang besar dan waktu pengiriman yang lama. Maka, perlu untuk mengimplementasikan kompresi citra yang efektif dan efisien menggunakan algoritma kompresi citra. Pada Penelitian ini menggunakan algoritma Fraktal dan Arithmetic Coding untuk melakukan kompresi pada citra medis sinar-x dada. Hasil dari penelitian ini menunjukkan bahwa dengan melakukan kompresi citra menggunakan algoritma Fraktal dan Arithmetic Coding kurang cocok jika diterapkan pada citra yang berukuran besar, karena dari segi waktu akan menjadi tidak efisien. Untuk mengukur hasil kualitas citra menggunakan PSNR (Peak Signal Noise to Ratio) pada kedua format citra JPG dan PNG tidak ada perbedaan yang mencolok, nilai PSNR tertinggi terdapat pada citra berformat JPG (33,78 dB). Dari segi rasio kompresi, citra hasil format PNG lebih menghemat ruang penyimpanan daripada citra format JPG.

**Kata Kunci :** *Fraktal, Arithmetic Coding, Kompresi, Citra Digital, Lossy, Lossless, PSNR*

## ABSTRACT

Modern medical image diagnostics involves a large set of medical images that are often based on x-rays taken to view the internal condition of the body. In the era of digital health, as some of these medical images are very large in size, storing and transferring them has always been an important issue for healthcare information systems to handle the increasing size of digital examination files. Thus, requiring large data storage and long transmission time. So, it is necessary to implement effective and efficient image compression using image compression algorithms. This research uses Fractal and Arithmetic Coding algorithms to compress chest x-ray medical images. The results of this research show that image compression using Fractal and Arithmetic Coding is not suitable if applied to large images, because in time it will be inefficient. To measure the results of image quality using PSNR (Peak Signal Noise to Ratio) in both JPG and PNG there is no significant difference, the highest PSNR value is found in JPG format images (33.78 dB). In terms of compression ratio, the PNG saves more storage space than the JPG format image.

**Keywords:** *Fractal, Arithmetic Coding, Compression, Digital Image, Lossy, Lossless, PSNR*

## DAFTAR ISI

	Halaman
<b>PERSETUJUAN.....</b>	<b>iii</b>
<b>PERNYATAAN.....</b>	<b>iii</b>
<b>PENGHARGAAN.....</b>	<b>v</b>
<b>ABSTRAK .....</b>	<b>vii</b>
<b>ABSTRACT .....</b>	<b>viii</b>
<b>DAFTAR ISI.....</b>	<b>ix</b>
<b>DAFTAR GAMBAR.....</b>	<b>xi</b>
<b>DAFTAR TABEL .....</b>	<b>xii</b>
<b>BAB 1 PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Metodologi Penelitian .....	4
1.7 Penelitian Relevan.....	6
1.8 Sistematika Penulisan.....	9
<b>BAB 2 LANDASAN TEORI .....</b>	<b>10</b>
2.1 Kompresi .....	10
2.2 Citra Digital.....	12
2.2.1 Jenis-jenis Citra.....	12
2.2.2 Format <i>File</i> Citra .....	15
2.3 Algoritma.....	16
2.3.1 Algoritma Fraktal .....	16
2.3.1.1 Transformasi Affine .....	19
2.3.1.2 Partition Iterated Function System (PIFS) .....	20
2.3.2 Algoritma <i>Arithmetic Coding</i> .....	21
2.4 <i>Compression Ratio</i> (Rasio Kompresi).....	25

2.5	<i>Running Time</i> .....	25
2.6	<i>Peak Signal to Noise Ratio (PSNR)</i> .....	25
<b>BAB 3 ANALISIS DAN PERANCANGAN.....</b>		<b>27</b>
3.1	Analisis Sistem .....	27
3.1.1	Analisis Masalah .....	27
3.1.2	Analisis Kebutuhan .....	27
3.1.2.1	Kebutuhan Fungsional .....	27
3.1.2.2	Kebutuhan Non-Fungsional .....	28
3.1.3	Arsitektur Umum .....	28
3.1.4	Analisis Proses .....	29
3.2	Pemodelan Sistem .....	30
3.2.1	<i>Use Case Diagram</i> .....	30
3.2.2	<i>Activity Diagram</i> .....	31
3.2.3	<i>Sequence Diagram</i> .....	32
3.3.	Diagram Alir ( <i>flowchart</i> ).....	33
3.3.1	<i>Flowchart</i> Kompresi dan Dekompresi Algoritma <i>Fraktal</i> .....	34
3.3.2	<i>Flowchart</i> Kompresi dan Dekompresi Algoritma <i>Arithmetic Coding</i> ..	36
<b>BAB 4 IMPLEMENTASI DAN PENGUJIAN.....</b>		<b>38</b>
4.1.	Implementasi Sistem .....	38
4.1.1	Kompresi <i>Fraktal</i> pada Citra Medis Sinar-x Dada .....	38
4.1.2	Kompresi <i>Arithmetic Coding</i> pada Citra Medis Sinar-x Dada.....	47
4.2	Pengujian Sistem .....	61
4.2.1	Data Pengujian .....	61
4.2.2	Pengujian Kompresi Citra Medis Sinar-x Dada dengan Format PNG..	62
4.2.2	Pengujian Kompresi Citra Medis Sinar-x Dada dengan Format JPG..	70
<b>BAB 5 KESIMPULAN DAN SARAN.....</b>		<b>82</b>
5.1	Kesimpulan.....	82
5.2	Saran .....	82
<b>DAFTAR PUSTAKA .....</b>		<b>83</b>

## DAFTAR GAMBAR

<b>Gambar 2. 1</b> (a) Proses Kompresi (b) Proses Dekompresi .....	10
<b>Gambar 2. 2</b> Contoh Kompresi Lossy .....	11
<b>Gambar 2. 3</b> Contoh Kompresi Lossless .....	12
<b>Gambar 2. 4</b> Koordinat Citra Digital.....	12
<b>Gambar 2. 5</b> Citra RGB .....	13
<b>Gambar 2. 6</b> Citra <i>Grayscale</i> .....	13
<b>Gambar 2. 7</b> Citra Biner .....	14
<b>Gambar 2. 8</b> Citra Rontgen Dada .....	15
<b>Gambar 2. 9</b> Model Sistem Kompresi Citra Dengan Fraktal .....	17
<b>Gambar 2. 10</b> Model Sistem Dekompresi Citra Dengan Fraktal.....	19
<b>Gambar 2. 11</b> Potongan Piksel Citra Digital .....	22
<b>Gambar 2. 12</b> Potongan Nilai <i>Gray Level</i> Citra Digital .....	22
<b>Gambar 3. 1</b> Arsitektur Umum Sistem .....	29
<b>Gambar 3. 2</b> Use Case Diagram Sistem .....	31
<b>Gambar 3. 3</b> Activity Diagram Sistem .....	32
<b>Gambar 3. 4</b> Sequence Diagram Sistem .....	33
<b>Gambar 3. 5</b> (a) <i>Flowchart</i> Kompresi <i>Fraktal</i> (b) <i>Flowchart</i> Dekompresi <i>Fraktal</i> ..	35
<b>Gambar 3. 6</b> (a) Flowchart Kompresi Arithmetic Coding (b) Flowchart Dekompresi Arithmetic Coding.....	37
<b>Gambar 4. 1</b> Contoh Potongan Piksel pada Citra .....	38
<b>Gambar 4. 2</b> Blok Domain.....	39
<b>Gambar 4. 3</b> Blok Range .....	39
<b>Gambar 4. 4</b> Blok Domain yang diskalakan.....	41
<b>Gambar 4. 5</b> Blok Range <sub>1</sub> .....	42
<b>Gambar 4. 6</b> Blok Range <sub>2</sub> .....	42
<b>Gambar 4. 7</b> Blok Range <sub>3</sub> .....	43
<b>Gambar 4. 8</b> Blok Range <sub>4</sub> .....	44
<b>Gambar 4. 9</b> Blok Range Terpilih .....	45
<b>Gambar 4. 10</b> Matriks Citra Grayscale 8x8 Piksel .....	47
<b>Gambar 4. 11</b> Grafik Rasio Kompresi .....	79
<b>Gambar 4. 12</b> Grafik Nilai PSNR .....	79

**DAFTAR TABEL**

<b>Tabel 1. 1</b> Penelitian Relevan .....	6
<b>Tabel 2. 1</b> Probabilitas Frekuensi .....	22
<b>Tabel 2. 2</b> Proses <i>Encoding</i> .....	23
<b>Tabel 2. 3</b> Proses <i>Decoding</i> .....	25
<b>Tabel 4. 1</b> Spesifikasi Jenis Perangkat Implementasi.....	38
<b>Tabel 4. 2</b> Probabilitas dan Range untuk Gambar 4.10.....	47
<b>Tabel 4. 3</b> Proses Encoding untuk Gambar 4.10 .....	54
<b>Tabel 4. 4</b> Proses Decoding untuk Gambar 4.10 .....	60
<b>Tabel 4. 5</b> Hasil Pengujian Citra dengan Ekstensi PNG .....	62
<b>Tabel 4. 6</b> Hasil Pengujian Citra dengan Ekstensi JPG.....	70

## BAB 1

### PENDAHULUAN

#### 1.1 Latar Belakang

Pencitraan medis adalah salah satu bidang yang paling aktif dan berkembang pesat dalam penelitian medis dan diagnosis klinis. Diagnostik citra medis modern melibatkan sekumpulan besar citra medis yang sering kali didasarkan pada sinar-x. Hal ini dilakukan untuk melihat keadaan bagian dalam tubuh, mulai dari tulang, persendian, hingga organ dalam. Sinar-x dapat mengidentifikasi berbagai penyakit, termasuk patah tulang, osteoporosis, infeksi, masalah pencernaan, radang jantung, dan kanker payudara (Tamin, 2021).

Di era kesehatan digital, karena sebagian dari citra medis ini berukuran sangat besar, menyimpan dan mentransfernya masih menjadi masalah penting bagi sistem informasi layanan kesehatan untuk menangani ukuran file pemeriksaan digital yang semakin meningkat. Sehingga, memerlukan bandwith dan penyimpanan data dalam jumlah besar dan waktu pertukaran data yang lama. Oleh karena itu, perlu untuk mengimplementasikan kompresi citra agar dapat meningkatkan efisiensi dan aksesibilitas dalam penelitian kesehatan digital serta mendukung pertukaran data yang aman dan efisien.

Metode kompresi citra ada dua jenis yaitu kompresi *lossless* dan kompresi *lossy*. Kompresi *lossless* merupakan jenis kompresi dimana data yang dikompresi bisa dikembalikan ke bentuk aslinya tanpa kehilangan informasi. Namun, rasio kompresi dari kompresi *lossless* sangat rendah, sehingga sulit untuk memenuhi persyaratan transmisi gambar medis yang sebenarnya. Sedangkan kompresi *lossy* merupakan jenis kompresi citra yang hasil dekompresi citra berbeda dengan citra aslinya karena hilangnya informasi, namun masih bisa diterima oleh persepsi mata. Metode ini mengungguli kompresi *lossless* dalam hal rasio kompresi. (Sutoyo *et.al* 2009).

Kompresi *Fraktal* adalah jenis kompresi *lossy* yang menggunakan sifat *self-similarity* dari objek *Fraktal*. Kesamaan diri yang tepat menunjukkan bahwa objek

*Fraktal* terdiri dari salinan yang diperkecil dari dirinya sendiri kemudian diturunkan transformasinya yang disebut transformasi *affine* (Khailili *et al*, 2013).

Berdasarkan penelitian sebelumnya yang dilakukan oleh (Shuai Liu *et al*, 2019) meneliti tentang kompresi citra medis MRI menggunakan metode *Fraktal*. Penelitian tersebut menyimpulkan bahwa kecepatan kompresi meningkat sekitar 2 hingga 3 kali lipat dan *PSNR* (*Peak Signal to Noise Rasio*) meningkat hampir 10 kali lipat. Namun, waktu kompresinya lebih lama.

Penelitian terdahulu juga dilakukan oleh (Hasanujjaman *et al*, 2019) yang berjudul *Fractal Image Compression of an Atomic Image using Quadtree Decomposition*. Sebagai hasil dari penerapan algoritma yang diusulkan, diperoleh PSNR maksimum 30,12 dB untuk gambar Lena dan rasio kompresi maksimum 25,96 untuk gambar MRI.

Sebelum Penelitian diatas, (Panjavarnam.B dan P.T.V. Bhuvaneswari, 2017) juga telah melakukan penelitian yang berjudul *Suitability Analysis of Fractal Compression Technique for Medical Image*. Penelitian tersebut menghasilkan bahwa teknik Fractal image compression (FIC) menghasilkan rasio kompresi yang lebih tinggi sekitar 8 kali lebih besar dari Set Partitioning in Hierarchical Trees (SPIHT) dengan tingkat kualitas citra hasil rekonstruksi yang dapat diterima. Oleh karena itu disimpulkan bahwa, untuk transmisi gambar medis, FIC cocok karena menawarkan rasio kompresi yang lebih tinggi yang dapat menghemat bandwidth transmisi secara efisien.

Algoritma *Arithmetic Coding* metode kompresi lossless yang merupakan bagian dari pengkodean entropi yang mengubah data menjadi bentuk data lain dengan menggunakan lebih sedikit bit karakter lebih sering dan lebih sedikit bit karakter. Dalam melakukan kompresi foto \*.bmp, algoritma *Arithmetic Coding* melakukan kompresi lebih cepat dibandingkan dengan metode Shannon-Fano, serta hasil kompresi dan rasio kompresi yang dicapai oleh algoritma *Arithmetic Coding* lebih unggul dibandingkan algoritma Shannon-Fano. *Arithmetic Coding* memiliki rata-rata rasio 62,88% sedangkan Shannon-Fano memiliki rata-rata rasio 61,73%. (Syahfitri, 2008).

Berdasarkan penelitian sebelumnya yang dilakukan oleh (Thafseela dan Saravanan, 2018) yaitu *Analysis of Stereoscopic Image Compression using Arithmetic Coding and Huffman Coding*. Dari hasil analisis terlihat bahwa meskipun kinerja kedua algoritma mirip, waktu eksekusi yang dilakukan oleh *Huffman coding* jauh lebih lama dibanding *Arithmetic coding*. Oleh karena itu, *Arithmetic coding* lebih baik daripada *Huffman coding*. *Arithmetic coding* menghasilkan CR (15.4929) dan PSNR yang lebih tinggi.

Penelitian terdahulu juga dilakukan oleh (Md. Ahsan dan Rubaiyat, 2017) yang berjudul *Edge-based Transformation and Entropy Coding for Lossless Image Compression*. ETEC menawarkan kompresi yang lebih baik daripada *Joint Photographic Experts Group Lossless* (JPEG-LS) dan *Set Partitioning in Hierarchical Trees* (SPIHT). ETEC menggunakan *Arithmetic Coding* mendapatkan hasil rata- rata rasio kompresi 31.46 dan rata- rata waktu kompresi 99,02 detik.

Berdasarkan referensi diatas, maka untuk penelitian ini penulis akan menggunakan Algoritma *Fraktal* dan *Arithmetic Coding* untuk mengompresi citra medis sinar-x dada. Dengan menggunakan kedua algoritma ini diharapkan dapat menghemat ruang penyimpanan dengan mempertahankan tingkat kualitas yang baik pada citra medis dan mempercepat pengiriman citra ditinjau dari karnrasio kompresi, waktu kompresi yang lebih singkat dan PSNR yang tinggi.

## 1.2 Rumusan Masalah

Bersumber dari kondisi diatas, karena sebagian dari citra medis berukuran sangat besar, maka menyimpan dan mentransfernya masih menjadi kendala dalam hal penyimpanan dan pengiriman citra medis. Oleh karena itu, untuk menangani ukuran file pemeriksaan digital yang semakin meningkat, maka penulis akan melakukan kompresi citra medis menggunakan algoritma *Fraktal* dan *Arithmetic Coding* untuk meminimalkan penggunaan memori tanpa mengurangi informasi atau kualitas gambar dan mempersingkat pengiriman citra ditinjau dari nilai waktu kompresi yang lebih cepat, nilai PSNR dan rasio kompresi yang lebih tinggi.

### **1.3 Batasan Masalah**

Penulis membatasi ruang masalah dalam melakukan penelitian ini. Berikut batasan-batasan masalah pada penelitian ini:

1. Metode kompresi citra yang dipakai adalah *Fraktal* dan *Arithmetic Coding*.
2. Citra yang digunakan berekstensi jpg (.jpg) dan png (,png)
3. Citra yang digunakan adalah citra medis sinar-x dada dari dataset yang diambil dari *Kaggle*.
4. Ukuran maksimal citra yang digunakan adalah 2048 x 2048.
5. Menggunakan bahasa pemrograman Python.
6. Parameter yang digunakan antara lain PSNR, rasio kompresi dan waktu kompresi.

### **1.4 Tujuan Penelitian**

Tujuan dari penelitian ini adalah membangun sistem kompresi citra medis sinar-x dada dengan menggunakan algoritma *Fraktal* dan *Arithmetic Coding* sehingga dapat menghemat ruang penyimpanan dan mempercepat waktu pengiriman citra.

### **1.5 Manfaat Penelitian**

Diharapkan penelitian ini akan menghasilkan suatu sistem yang dapat memperkecil ukuran citra dengan mempertahankan kualitas gambar sehingga, dapat menghemat ruang penyimpanan dan mempercepat waktu pengiriman citra. Selain itu, sebagai bahan referensi bagi peneliti lain yang ingin membahas topik yang berkaitan dengan penelitian ini.

### **1.6 Metodologi Penelitian**

Dalam penelitian ini, digunakan metodologi sebagai berikut:

1. Studi Literatur

Pada bagian ini penelitian diawali dengan mengumpulkan referensi dari banyak sumber dan studi literatur melalui jurnal, buku, e-book, karya ilmiah, skripsi, atau makalah yang terkait dengan algoritma *Fraktal* dan *Arithmetic Coding*.

## 2. Analisis dan Perancangan

Pada bagian ini, penulis melakukan analisis dan melakukan perancangan yang direpresentasikan dalam sebuah *flowchart* (diagram alir), *use-case diagram*, *sequence diagram*.

## 3. Implementasi Sistem

Dalam tahap ini, sistem dirancang memakai bahasa pemrograman Python. Sehingga sistem dapat melakukan kompresi citra menggunakan algoritma *Fraktal* dan *Arithmetic Coding*.

## 4. Pengujian Sistem

Pada bagian ini, akan dilakukan pengujian terhadap sistem yang dirancang untuk menjamin bahwa sistem berjalan sesuai dengan rancangan.

## 5. Dokumentasi

Dalam bagian ini akan dilakukan pendokumentasian mulai dari tahap analisis hingga tahap pengujian yang akan menjadi bentuk skripsi.

## 1.7 Penelitian Relevan

Beberapa penelitian yang relevan dengan penelitian ini antara lain:

**Tabel 1. 1** Penelitian Relevan

No	Judul Penelitian	Penulis	Tahun	Masalah	Hasil
1.	A Fast Fractal Based Compression for MRI Images	Shuai Liu, Weiling Bai, Nianyin Zeng, and Shuihua Wang	2019	Gambar MRI dalam jumlah besar memerlukan ruang penyimpanan yang besar dan bandwidth transmisi sistem PACS dalam penyimpanan internal dan diagnosis jarak jauh	Kecepatan kompresi meningkat sekitar 2 hingga 3 kali lipat dari algoritma tradisional dan PSNR juga meningkat lebih signifikan, hanya saja membutuhkan waktu kompresi yang lebih besar.
2.	Fractal Image Compression of an Atomic Image using Quadtree Decomposition	Hasanujjaman <i>et al</i>	2019	Pengurangan kompleksitas waktu algoritma kompresi gambar dan pengembangan metode untuk mencapai rasio kompresi yang lebih baik dan PSNR yang lebih tinggi dalam konteks pembuatan atomic image	Diperoleh PSNR maksimum 30,12 dB untuk Gambar Lena dan rasio kompresi maksimum 25,96 untuk gambar MRI.
3.	Analysis of Stereoscopic Image Compression	Thafseela dan Saravanan	2018	Membandingkan hasil dua metode yaitu menggunakan kompresi stereoskopis Pengkodean aritmatika (SCAC)	Dari hasil analisis terlihat bahwa meskipun kinerja kedua algoritma mirip, waktu eksekusi yang

	using Arithmetic Coding and Huffman Coding			dan kompresi stereoskopis menggunakan Pengkodean Huffman (SCHC) untuk melihat kinerja yang lebih baik.	dilakukan oleh stereoscopic compression using Huffman coding (SCHC) jauh lebih lama dibanding stereoscopic compression using Arithmetic coding (SCAC).
4.	Codec Citra Berbasis Fractal dan Entropy Coding	I Made Aditya Virgiawan1, I Made Oka Widyatara dan Rukmi Sari Hartati	2019	Banyaknya informasi penting yang terdapat pada citra medis menyebabkan ukuran file citra medis tergolong besar. Kebutuhan media penyimpanan data yang besar dan waktu pengiriman yang lama disebabkan oleh data citra medis yang berukuran besar.	Metode Arithmetic lebih memiliki keunggulan dalam bidang PSNR, Rasio Kompresi, Size Image yang dihasilkan beserta Time Taken di dalam proses kompresi, dimana dari 15 gambar citra yang diuji 100% gambar lebih baik hasil dengan menggunakan metode Arithmetic dibandingkan dengan menggunakan metode Adaptive Huffman.
5	Suitability Analysis of Fractal Compression Technique for Medical Image	Panjavarnam.B dan P.T.V. Bhuvaneswari	2017	Volume gambar dalam sistem informasi medis semakin meningkat. Hal ini menyebabkan perlunya pemampatan gambar untuk penyimpanan dan transmisi	Rasio kompresi yang lebih tinggi sekitar 8 kali lebih besar dari Set Partitioning in Hierarchical Trees (SPIHT) dengan tingkat kualitas citra hasil rekonstruksi yang dapat

				yang efektif.	diterima.
6	Edge-based Transformation and Entropy Coding for Lossless Image Compression	Md. Ahsan dan Rubaiyat	2017	Mengatasi tantangan ukuran gambar dalam dunia digital dengan mengusulkan metode transformasi berbasis tepi dan pengodean entropi (ETEC) untuk mengurangi ukuran gambar tanpa kehilangan konten, serta membandingkannya dengan teknik pemampatan yang sudah ada	Edge-based transformation and entropy coding (ETEC) menawarkan kompresi yang lebih baik daripada Joint Photographic Experts Group Lossless (JPEG-LS) dan Set Partitioning in Hierarchical Trees (SPIHT) . ETEC menggunakan Arithmetic Coding menghasilkan rata-rata rasio kompresi 31.46 dan rata-rata waktu kompresi 99,02 detik.

## 1.8 Sistematika Penulisan

Berikut sistematika penulisan dari penelitian ini:

### **BAB 1 PENDAHULUAN**

Bagian ini mencakup latar belakang, rumusan masalah, batasan masalah, manfaat penelitian, tujuan penelitian dan sistematika penulisan.

### **BAB 2 LANDASAN TEORI**

Bagian ini berisi tinjauan teoritis yang berkaitan dengan Kompresi, algoritma yang digunakan yaitu *Fraktal* dan *Arithmetic Coding*.

### **BAB 3 ANALISIS DAN PERANCANGAN**

Bagian ini mencakup analisis dan perancangan sistem dengan *use case* diagram, *activity* diagram, *sequence* diagram dan *flowchart* (diagram alir).

### **BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM**

Bagian ini mencakup pengimplementasian dan pengujian terhadap sistem yang dirancang.

### **BAB 5 KESIMPULAN DAN SARAN**

Bagian ini mencakup kesimpulan penulis dari hasil uji sistem yang telah dilakukan dan saran penulis untuk pengembangan penelitian kedepannya.

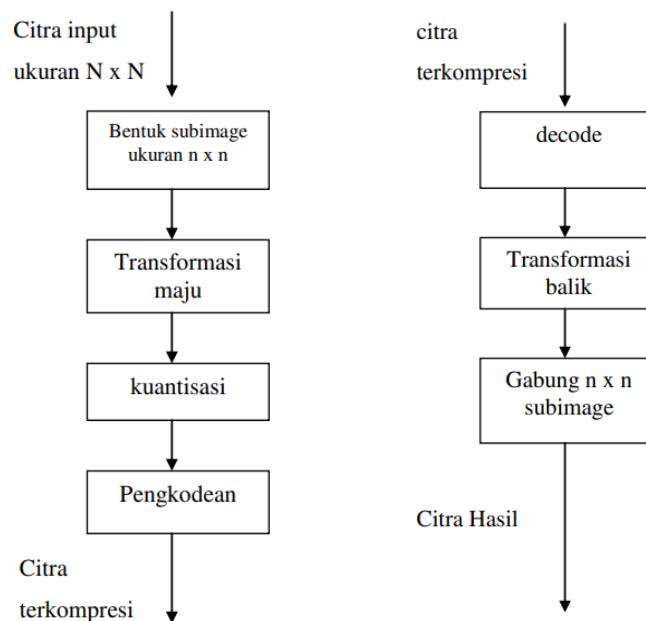
## BAB 2

### LANDASAN TEORI

#### 2.1 Kompresi

Teknik yang mengubah aliran data masukan (sumber atau data mentah asli) menjadi aliran data yang lebih kecil (*bitstream* hasil, atau aliran terkompresi) disebut kompresi data. Data adalah sekelompok fakta. Bentuk jamak dari “datum” adalah data, yang berarti “informasi yang diberikan” dalam bahasa Latin. Data dapat berupa teks, gambar, audio, atau simbol lain yang dapat digunakan manusia untuk melihat dunia, benda, peristiwa, atau konsep (David, 2007).

Tujuan kompresi gambar adalah meminimalkan jumlah memori yang diperlukan untuk menyimpan citra digital. Strategi utama yang digunakan dalam kompresi gambar digital adalah menghilangkan data berlebihan dalam gambar sehingga memori yang diperlukan untuk merepresentasikan gambar tersebut lebih kecil dari citra asli (Madenda, 2015) Dalam kompresi gambar digital, terdapat dua proses kompresi yang ditunjukkan seperti pada gambar 1.1.



**Gambar 2. 1 (a) Proses Kompresi (b) Proses Dekompresi**

(Sumber : Putra, D. 2010)

## 1. Kompresi Citra (*images compression*)

Kompresi citra adalah proses mengurangi ukuran file atau data yang merepresentasikan citra digital, tanpa mengorbankan secara signifikan kualitas visual citra tersebut. Tujuan kompresi citra adalah menghemat ruang penyimpanan dan/atau mempercepat transmisi citra melalui jaringan.

## 2. Dekompresi Citra (*images decompression*)

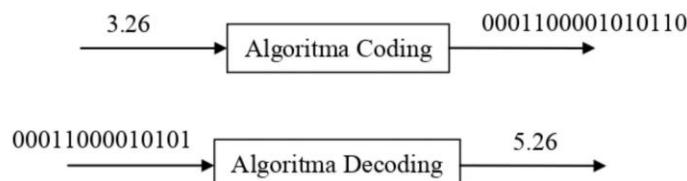
Dekompresi citra adalah proses mengembalikan citra yang telah dikompresi ke bentuk aslinya atau mendekati bentuk aslinya. Selama proses dekompresi, data yang telah dikompresi dikembalikan ke bentuknya yang semula sehingga dapat dilihat atau digunakan lagi. Proses dekompresi biasanya berjalan sebaliknya dari proses kompresi, dengan mengembalikan informasi yang telah dihapus atau dikodekan selama kompresi.

### 2.1.1 Jenis-jenis Kompresi

Teknik kompresi terdiri dari dua jenis yakni:

#### a. Kompresi *Lossy*

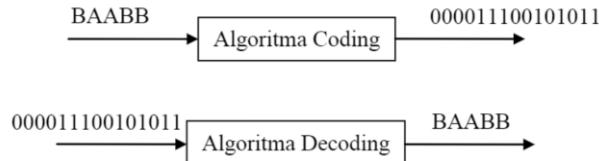
Kompresi *lossy* merupakan jenis kompresi dimana hasil citra yang dikompresi berbeda dengan citra aslinya karena kehilangan informasi, namun tetap dapat diterima oleh persepsi mata (Sutoyo *et.al*, 2009).



**Gambar 2. 2** Contoh Kompresi *Lossy*  
(*Sumber : Putra, D. 2010*)

#### b. Kompresi *Lossless*

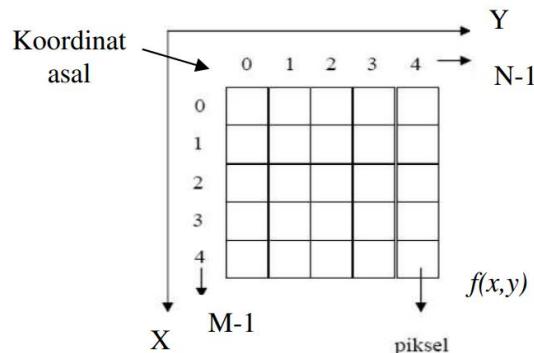
Kompresi *lossless* adalah jenis kompresi citra dimana citra yang dikompresi sama dengan citra aslinya. Namun, rasio kompresi dari kompresi *lossless* sangat rendah, sehingga sulit untuk memenuhi persyaratan transmisi gambar medis yang sebenarnya (Sutoyo *et.al*, 2009).



**Gambar 2. 3** Contoh Kompresi *Lossless*  
(Sumber : Putra, D. 2010)

## 2.2 Citra Digital

Suatu bayangan dapat dijelaskan sebagai suatu representasi dalam dua dimensi dari intensitas cahaya di setiap titik  $(x, y)$  dalam ruang. Dalam konteks ini,  $(x, y)$  merepresentasikan koordinat, sementara  $f$  menggambarkan tingkat kecerahan atau intensitas cahaya di titik tersebut. Ketika citra kontinu ini didiskritisasi menjadi koordinat dan tingkat kecerahan, maka kita dapat menyebutnya sebagai citra digital.



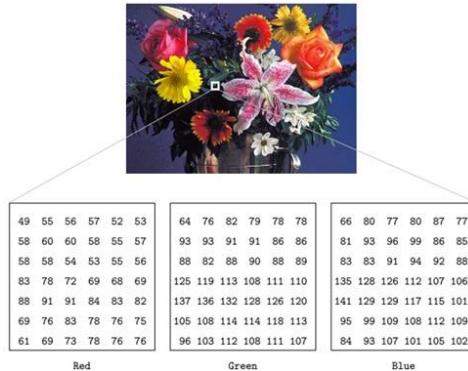
**Gambar 2. 4** Koordinat Citra Digital

### 2.2.1 Jenis-jenis Citra

#### 1. Citra Warna (RGB)

Dalam citra berwarna (RGB), terdapat tiga elemen warna utama: merah (R), hijau (G), dan biru (B). Setiap elemen warna memiliki nilai yang berkisar antara 0 hingga 255, dan masing-masing piksel menggunakan total 24 bit (8 bit untuk R, 8 bit untuk G, dan 8 bit untuk B). Ini menghasilkan

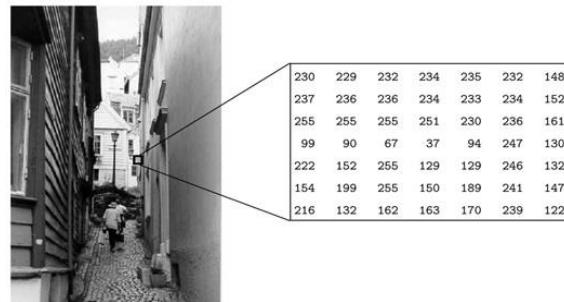
sebanyak  $255^3$  atau 16.777.216 kombinasi warna yang mungkin. Jenis gambar ini sering dikenal sebagai gambar berwarna 24-bit.



**Gambar 2. 5 Citra RGB**

## 2. Citra *Grayscale*

Setiap piksel dalam gambar *grayscale* menunjukkan derajat keabuan yang berkisar antara 0 (hitam) hingga 255 (putih). Artinya setiap piksel berukuran 8 bit atau 1 byte pada rentang nilai 0 hingga 255. Gambar di bawah adalah contoh gambar *grayscale*.

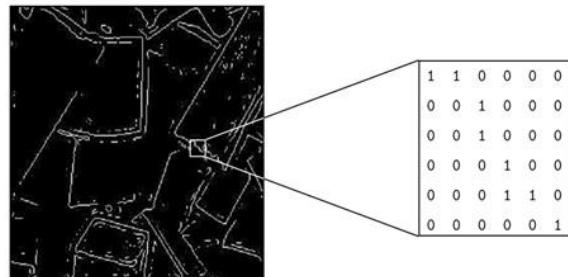


**Gambar 2. 6 Citra *Grayscale***

## 3. Citra Biner

Citra biner adalah jenis citra digital yang hanya menggunakan dua warna, yaitu hitam dan putih. Dalam citra biner, setiap piksel hanya memiliki dua nilai, biasanya 0 untuk warna hitam dan 1 untuk warna putih. Ini digunakan untuk menggambarkan objek atau area dengan kontras tinggi terhadap latar belakangnya. Pada gambar 2.7 merupakan contoh gambar biner,

di mana putih merepresentasikan piksel tepi dan hitam merepresentasikan latar belakang.



**Gambar 2.7** Citra Biner

#### 4. Citra Medis

Citra medis biasanya memuat banyak informasi, sehingga ukurannya cukup besar (Khairil, 2008). Untuk meminimalkan kerusakan, diperlukan ruang penyimpanan khusus selain foto medis yang dipakai sebagai referensi pasien di kemudian hari untuk mempersingkat waktu selama proses penyimpanan dan pemindahan (Soesanti, 2008).

Sinar-x, sering juga disebut gambar rontgen, adalah teknik diagnostik yang menerapkan sinar-x. Dimana sinar-x akan masuk ke bagian tubuh dan ditangkap pada film sehingga menghasilkan gambaran komponen tubuh yang diterangi. Mesin x-ray berisi sinar-x yang dihasilkan oleh gambar. Gambar medis dibuat menggunakan berbagai teknologi, termasuk x-ray, *magnetic resonance imaging* (MRI), *computer tomography* (CT), ultrasound, dan lain-lain. Sampel citra medis ditunjukkan oleh gambar 2.8 dibawah.



**Gambar 2. 8** Citra Rontgen Dada

### 2.2.2 Format File Citra

Format file adalah struktur dari sebuah file untuk merepresentasikan file digital. Setiap file memiliki kelebihan dan kekurangan dari setiap formatnya, berikut format file citra yang umum digunakan yaitu :

1. *Joint Photographic Experts Group* (JPEG)

Format JPEG merupakan format file citra yang umum digunakan dan bersifat *lossy*. Citra grayscale 8 bit serta citra berwarna 24 bit juga didukung oleh format ini. karena keuniversalannya maka jenis format JPEG ini dapat dikirim dan dilihat secara bebas namun ukuran file JPEG lebih kecil dibandingkan format file lainnya.

2. *Portable Network Graphics* (PNG)

Format PNG adalah format gambar raster *lossless*. Citra yang diindeks dalam palet warna asli 8 bit, 24 bit serta 48 bit didukung oleh format file PNG.

3. *Bitmap* (BMP)

Format BMP adalah format gambar yang menampilkan gambar visual sebagai kumpulan titik-titik (pixsel) yang tersimpan di memori komputer tetapi tidak dikompresi (*uncompressed image*). Citra berwarna, citra biner, dan citra hitam-putih (*grayscale*) semuanya merupakan bagian dari format BMP.

4. *Graphics Interchange Format* (GIF)

Format GIF adalah jenis format file citra yang menggunakan format gambar bitmap dan mendukung 8 bit per piksel di setiap gambarnya. Format GIF memiliki palet warna terbatas dengan maksimal 256 warna RGB dan tidak mendukung CMYK (*Cyan Magenta Yellow Black*).

#### 5. *Tagged Image File Format* (TIFF)

Format TIFF merupakan format gambar jenis raster yang bersifat lossy. Format file TIFF ini biasanya digunakan untuk printing. File TIFF tidak didukung secara otomatis oleh browser sehingga diharuskan untuk menginstal add-on atau ekstensi untuk merendernya di browser.

### 2.3 Algoritma

Algoritma adalah serangkaian instruksi atau prosedur sistematis yang digunakan untuk memecahkan kesulitan atau masalah logis dan matematis dengan bantuan komputer (Sismoro, 2005).

Algoritma berasal dari ungkapan “algoris” dan “ritmis”, dan istilah ini pertama kali digunakan pada tahun 825 M oleh Abu Ja'far Muhammad Ibnu Musa Al Khawarizmi dalam karyanya Al-Jabr Wa-al Muqabla. Algoritma didefinisikan dalam dunia pemrograman sebagai suatu teknik yang terdiri atas rangkaian langkah-langkah yang terorganisir dan sistematis untuk mengatasi suatu masalah dengan menggunakan komputer (Jando & Nani, 2018).

#### 2.3.1 Algoritma Fraktal

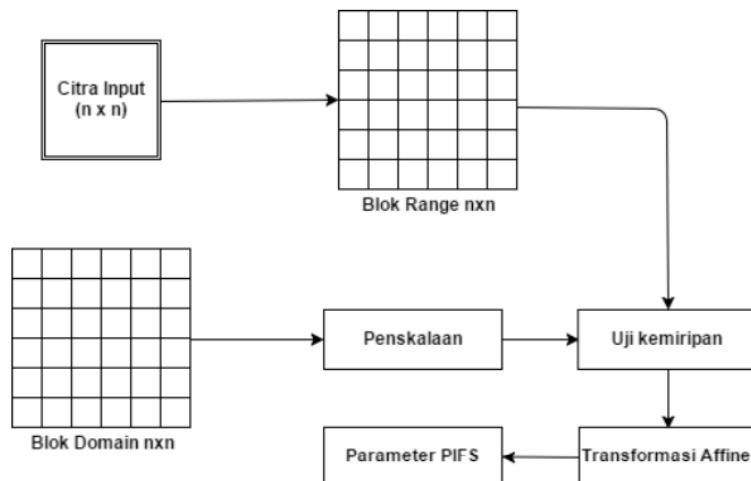
*Frangere* merupakan bahasa latin dari Fraktal yang berarti memecah sesuatu ke dalam potongan-potongan yang tidak beraturan sehingga bentuknya dapat disatukan dalam satu buah nama tunggal. *Fraktal* juga bisa berarti fractus (Mandelbrot) yang berarti memecahkan, menciptakan bentuk-bentuk yang tidak umum. Bentuk-bentuk fraktal mulai ditemukan pada sekitar abad ke-19 sebagai kumpulan struktur yang tidak cocok untuk bentuk-bentuk Euclid dan Newton.

Seorang spesialis *Fraktal*, Barnsley, mengatakan bahwa *Fraktal* adalah sebuah subset. Secara umum digambarkan sebagai suatu bentuk geometri yang dibangun dari potongan-potongan yang mempunyai bentuk yang sama jika bentuk geometri tersebut diperbesar.

*Fraktal* adalah bentuk geometris kasar yang dapat diamati "dibagi-bagi". Beberapa *Fraktal* dapat dibagi menjadi beberapa bagian yang seluruhnya berhubungan dengan *Fraktal* asli. *Fraktal* diklaim mengandung perincian tak terhingga dan strukturnya mirip pada tingkat perbesaran yang berbeda. *Fraktal* dapat dibentuk dalam banyak keadaan dengan mengulangi suatu pola, umumnya secara rekursif atau berulang.

Algoritma *Fraktal* adalah jenis kompresi lossy yang menggunakan sifat self-similarity yang berarti bahwa dalam suatu citra ada bagian yang mirip dengan bagian lainnya pada citra tersebut.

Kompresi dengan *Fraktal* melalui beberapa tahapan atau proses, berikut ini diberikan model sistem secara garis besar :



**Gambar 2. 9** Model Sistem Kompresi Citra Dengan *Fraktal*

Algoritma *Fraktal* untuk mengompresi citra digital adalah sebagai berikut:

- Membaca citra asli
- Tentukan ukuran matriks citra awal
- Tentukan ukuran blok domain

- d. Tentukan ukuran blok *range*
- e. Ubah ukuran blok domain menjadi  $\frac{1}{2}$  ukuran aslinya.
- f. Pada setiap blok *range* :
  - 1. Hitung rms (*root mean square*) untuk mencari persamaan antara blok *range* ke-*i* dan seluruh blok domain, seperti ditunjukkan dalam persamaan (2.1). Dalam persamaan tersebut  $n$  mewakili jumlah piksel dalam citra,  $z$  mewakili blok domain dan  $z'$  mewakili blok *range*.

$$d_{\text{rms}} = \frac{1}{n} \sqrt{\sum_{i=1}^n \sum_{j=1}^n (z'_{ij} - z_{ij})^2} \dots \dots \dots (2.1)$$

- 2. Dengan menggunakan persamaan (2.2), dilakukan proses perhitungan transformasi *affine* dijalankan antara blok *range* ke-*i* dengan blok domain yang dipilih. Dimana  $w_i$  memetakan titik (x,y) dengan intensitas  $z$  yang dalam blok domain,  $o_i$  mewakili offset kecerahan,  $s_i$  mewakili kontras,  $e$  dan  $f$  merupakan perpindahan sudut kiri blok domain ke sudut kiri blok *range*.

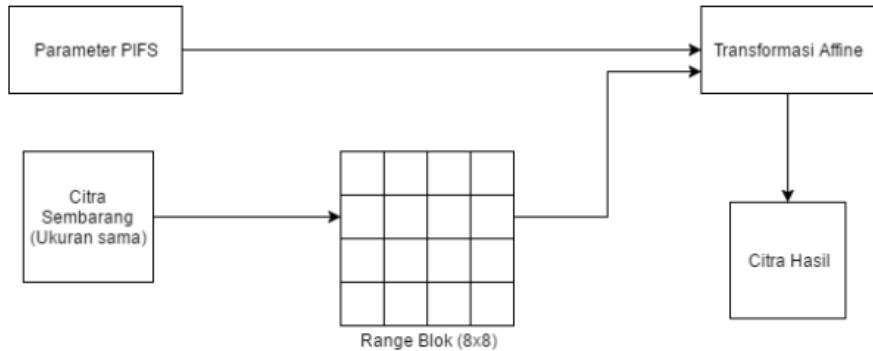
$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & s_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix} \dots \dots \dots (2.2)$$

- 3. Simpan koefisien transformasi *affine* ke-*i* dalam bentuk  $si$  yang merupakan faktor kontras piksel yang diperoleh dari persamaan (2.3),  $o_i$  merupakan kecerahan piksel yang terdapat pada persamaan (2.4), dan koefisien  $e_i$  dan  $f_i$  yang merupakan perpindahan sudut kiri blok domain ke sudut kiri blok *range* yang dipilih.  $d$  adalah piksel blok domain dan  $r$  adalah piksel blok *range*.

$$s = \frac{[n \sum_{i=1}^n d_i r_i - \sum_{i=1}^n d_i \sum_{i=1}^n r_i]}{[n \sum_{i=1}^n d_i^2 - (\sum_{i=1}^n d_i)^2]} \dots \dots \dots (2.3)$$

$$o = \frac{1}{n} [\sum_{i=1}^n r_i - s \sum_{i=1}^n d_i] \dots \dots \dots (2.4)$$

g. Semua parameter akan disimpan ke dalam PIFS.



**Gambar 2. 10** Model Sistem Dekompresi Citra Dengan *Fraktal*

Gambar 2.10 menjelaskan metode dekompresi *Fraktal*. Algoritma *Fraktal* untuk mendekompreksi citra digital adalah sebagai berikut:

- Membaca parameter PIFS
- Membaca citra random dengan ukuran yang identik dengan citra masukan.
- Membuat blok *range* berukuran 8 x 8 dari citra random tersebut.
- Lakukan transformasi *affine* dengan paramater PIFS untuk setiap blok *range*.
- Citra hasil terbentuk.

#### 2.3.1.1 Transformasi *Affine*

Salah satu transformasi objek yang sering dipakai pada grafik komputer adalah transformasi *affine*. Secara umum transformasi adalah metode yang mengubah bentuk suatu benda tanpa menyebabkannya kehilangan bentuk dasarnya. Transformasi *Affine* adalah transformasi yang menjaga garis sejajar tetap utuh. Untuk menghasilkan gambaran keluaran yang diperlukan, pendekatan transformasi *affine* ini mengkombinasikan prosedur mendasar seperti rotasi, translasi, dan penskalaan ke dalam sebuah proses. Pendekatan *Fraktal* sering kali menggunakan transformasi *affine* untuk menggambarkan transformasi kontraktif. Keterkaitan antara ketiga proses ini dapat dijelaskan dengan menggunakan rumus perhitungan transformasi *affine*. Berikut rumus transformasi *affine*.

$$\begin{vmatrix} x^2 \\ y^2 \end{vmatrix} = A \times \begin{vmatrix} x_2 \\ y_2 \end{vmatrix} + B \quad \dots \dots \dots \quad (2.5)$$

Rumus berikut menggabungkan proses pencukuran (*shearing*) ( $\rho_i$ ,  $\rho_j$ ), translasi ( $t_i$ ,  $t_j$ ), rotasi sebesar  $\theta$  derajat, dan penskalaan ( $\lambda_i$ ,  $\lambda_j$ ).

$$\begin{bmatrix} i' \\ j' \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \lambda_i & \rho_j & 0 \\ \rho_i & \lambda_j & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Matriks Scale dan Shear}} \underbrace{\begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Matriks rotasi}} \underbrace{\begin{bmatrix} 1 & 0 & t_i \\ 0 & 1 & t_j \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Matriks translasi}} \begin{bmatrix} i \\ j \\ 1 \end{bmatrix} \quad \dots \dots \dots \quad (2.6)$$

$$= \begin{bmatrix} \lambda_i \cos \theta + \rho_j \sin \theta & \rho_j \cos \theta - \lambda_i \sin \theta & (\lambda_i t_i + \rho_j t_j) \cos \theta + (\rho_j t_i - \lambda_i t_j) \sin \theta \\ \rho_i \cos \theta + \lambda_j \sin \theta & \lambda_j \cos \theta - \rho_i \sin \theta & (\rho_i t_i + \lambda_j t_j) \cos \theta + (\lambda_j t_i - \rho_i t_j) \sin \theta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ 1 \end{bmatrix}$$

#### 2.3.1.2 Partition Iterated Function System (PIFS)

Faktanya, mendapatkan satu gambar yang mirip dengan keseluruhan gambar cukup sulit. IFS tidak dapat digunakan dengan gambar apa pun. Dengan begitu IFS hanya bisa digunakan untuk objek *Fractal* (benda yang identik satu sama lain). Mencari kesamaan lokal, atau bagian gambar yang serupa dengan bagian lain dari gambar, mungkin memberikan kesamaan diri pada gambar mana pun. Pendekatan PIFS merupakan salah satu metode untuk menentukan kesamaan diri lokal. Pendekatan ini digunakan untuk mengompresi gambar dan sejak itu menjadi prosedur umum dalam pengkodean *Fractal*.

Untuk menemukan kesamaan lokal pada suatu gambar, PIFS membaginya menjadi blok *range* dan blok domain. Blok domain bisa tumpang tindih, namun blok *range* tidak bisa. Transformasi  $w_i$  terdapat pada persamaan (2.2) melalui ekspansi.

Hasil titik (x,y) mewakili domain piksel dengan nilai z, dan (x',y') mewakili rentang piksel dengan nilai baru z'. Ketika ukuran blok domain

turun dua kali lipat, faktor  $e_i$  dan  $f_i$  mentranslasikan menjadi blok *range*. Nilai  $e_i$  dan  $f_i$  mewakili perpindahan dari koordinat kiri atas blok domain ke koordinat kiri atas blok *range*. Koefisien  $s$  mewakili nilai kontras dan koefisien  $o$  adalah nilai kecerahan.

### **2.3.2 Algoritma Arithmetic Coding**

Abramson dan Peter Elias menemukan teknik *Arithmetic Coding* pada awal tahun 1960an. Memberikan *range* atau interval  $[0 ; 1]$  untuk setiap simbol, dengan setiap *range* dipecah menjadi beberapa  *subrange* yang ukurannya sebanding dengan probabilitasnya. Ini adalah konsep inti dari *Arithmetic Coding*. Semakin besar probabilitas suatu simbol, semakin besar range yang ditetapkan untuk simbol tersebut.

Algoritma *Arithmetic Coding* adalah algoritma kompresi *lossless* yang merupakan suatu bagian dari entropy coding yang mengkonversi suatu data ke dalam bentuk data yang lain dengan lebih sering menggunakan sedikit bit dan jarang menggunakan lebih banyak bit karakter.

Pengkodean aritmatika menggantikan bilangan bulat *floating point* untuk salah satu simbol urutan masukan. Semakin kompleks dan panjang pesan yang dikodekan, maka semakin banyak elemen yang dibutuhkan. Pengodean Aritmatika menghasilkan bilangan yang kurang dari satu dan lebih besar atau sama dengan nol. Setiap simbol yang akan dikodekan diberi serangkaian probabilitas yang akan menghasilkan nomor keluaran.

Gambar 2.11 menunjukkan contoh citra digital dalam bentuk matriks. Persamaan (2.7) selanjutnya dapat digunakan untuk menghitung nilai probabilitas setiap nilai *gray level*.

$$\text{probabilitas} = \frac{\text{frekuensi}}{\text{jumlah data}} \dots \dots \dots \dots \quad (2.7)$$

255	50	50	255	25	25
25	50	25	120	180	50
25	75	25	50	75	120
120	180	180	255	255	25
255	255	120	25	255	255
255	255	50	25	255	255

**Gambar 2. 11** Potongan Piksel Citra Digital**Tabel 2. 1** Probabilitas Frekuensi

<b>Gray Level</b>	<b>Frekuensi</b>	<b>Probabilitas</b>	<b>Range</b>
255	12	$12/36 = 1/3$	0 – 0,33
50	6	$6/36 = 1/6$	0,33 – 0,5
25	9	$9/36 = 1/4$	0,5 – 0,75
120	4	$4/36 = 1/9$	0,75 – 0,86
180	3	$3/36 = 1/12$	0,86 – 0,94
75	2	$2/36 = 1/18$	0,94 – 1
Total	36	1	0 – 1

Gambar 2.11 memiliki 36 nilai *gray level*, sehingga dipakai 4 nilai piksel seperti terlihat pada gambar 2.12.

255	50	50	255	25	25
25	50	25	120	180	50
25	75	25	50	75	120
120	180	180	255	255	25
255	255	120	25	255	255
255	255	50	25	255	255

**Gambar 2. 12** Potongan Nilai *Gray Level*

Setelah probabilitas setiap nilai *gray level* ditentukan, setiap simbol akan diberi *range* dengan nilai antara 0 dan 1, berdasarkan probabilitas yang ada. Selanjutnya dilakukan proses *encoding*.

Proses *encoding*, algoritma *Arithmetic Coding*:

```

Langkah 1: Set low = 0.0 (kondisi awal)
Langkah 2: Set high = 1.0 (kondisi awal)
Langkah 3: While (simbol input masih ada) do
Langkah 4: Ambil simbol input
Langkah 5: Code-Range = high - low
Langkah 6: High = low + Code Range*high range
           (simbol)
Langkah 7: Low = low + Code Range*low range (simbol)
Langkah 8: End While
Langkah 9: Output Low

```

Pertama untuk simbol 255. Nilai *code range* (kondisi awal)  $1,0 - 0,0 = 1$

```

high_range (S) = 0,33
low range (S) = 0,00

```

Kemudian diperoleh nilai :

$$\begin{aligned}
 \text{High} &= \text{low} + \text{Code Range} * \text{high range} \\
 &= 0,00 + 1,0 * 0,33 = 0,33 \\
 \text{Low} &= \text{low} + \text{Code Range} * \text{low range} \\
 &= 0,00 + 1,0 * 0,00 = 0,00
 \end{aligned}$$

Kemudian untuk simbol 50. Nilai *code range*  $0,33 - 0,00 = 0,33$

```

high_range (A) = 0,5
low_range (A) = 0,33

```

Kemudian diperoleh nilai :

$$\begin{aligned}
 \text{High} &= \text{low} + \text{Code Range} * \text{high range} \\
 &= 0,00 + 0,33 * 0,5 = 0,165 \\
 \text{Low} &= \text{low} + \text{Code Range} * \text{low range} \\
 &= 0,00 + 0,33 * 0,33 = 0,33 = 0,0189
 \end{aligned}$$

Proses *encoding* secara keseluruhan dapat dilihat pada tabel 2.2

**Tabel 2. 2** Proses *Encoding*

<b>Simbol</b>	<b>Low</b>	<b>High</b>	<b>CodeRange</b>
	0	1	
255	0	0,33	1
50	0,1089	0,165	0,33
25	0,13695	0,150975	0,0561
50	0,14157825	0,1439625	0,014025

Dari proses diatas didapat nilai  $low = 0,14157825$ . Kemudian nilai ini yang akan dipakai untuk proses *decode*.

Selanjutnya akan dilakukan proses *decode* menggunakan algoritma *Arithmetic decoding*. Misalkan untuk gambar 2.10 akan dilakukan *decoding*.

Proses *decoding*, algoritma *Arithmetic Coding*:

```

Langkah 1: Ambil Encoded Symbol (ES)
Langkah 2: Do
Langkah 3: Cari range dari simbol yang melingkupi
           Encoded Symbol (ES)
Langkah 4: Cetak simbol
Langkah 5: Code Range = high_range - low_range
Langkah 6: Encoded Symbol = Encoded Symbol -
           low_range
Langkah 7: Encoded Symbol = Encoded Symbol / Code
           Range
Langkah 8: Until simbol habis

```

Mula-mula, didapat *encoded number* 0,14157825 dengan simbol 255. Proses *decoding* untuk simbol 255 yaitu :

```

Low = 0,0
High = 0,33
Code Range = 0,33 - 0,00 = 0,33
Encoded number = 0,14157825 - 0,00 = 0,14157825
Encoded number = 0,14157825 / 0,33 = 0,429025

```

Proses *decoding* dilakukan hingga simbol habis. Proses *decoding* secara keseluruhan tercantum pada tabel 2.3.

**Tabel 2. 3 Proses Decoding**

<i>Encoded Symbol</i>	<i>Simbol</i>	<i>Low</i>	<i>High</i>	<i>Code Range</i>
0,429025	255	0	0,33	0,33
0,5825	50	0,33	0,5	0,17
0,33	25	0,5	0,75	0,25
0	50	0,33	0,5	0,27

#### **2.4      *Compression Ratio* (Rasio Kompresi)**

*Compression Ratio* adalah proses untuk mengukur sejauh mana citra telah dikompresi. Semakin tinggi rasio kompresi, semakin besar penghematan ruang penyimpanan atau bandwidth yang dicapai. Rumus rasio kompresi citra dinyatakan secara sistematis sebagai berikut (Pu, 2006).

$$CR = \left( \frac{\text{Ukuran file asli}}{\text{Ukuran file terkompresi}} \right)$$

#### **2.5      *Running Time***

*Running time* adalah periode waktu yang diperlukan oleh suatu algoritma untuk mengatasi suatu masalah pada komputer, yang dihitung mulai dari awal eksekusi program hingga akhirnya. Ketika prosesor tidak memulai dan berhenti secara bersamaan, waktu dihitung dari permulaan prosesor pertama hingga penyelesaian prosesor terakhir (Siahaan, 2017).

#### **2.6      *Peak Signal to Noise Ratio* (PSNR)**

*Peak Signal to Noise Ratio* adalah ukuran kualitas citra yang telah dikompresi atau restorasi. Semakin besar angka PSNR maka semakin tinggi kualitas gambar atau sinyal yang diproses, karena menunjukkan bahwa gangguan atau distorsi pada proses tersebut relatif kecil. Berikut ini adalah rumus untuk menghitung nilai PSNR.

$$\left\{ \begin{array}{l} PSNR = 10 \times \log_{10} \left[ \frac{(2^8 - 1)^2}{MSE} \right] \\ MSE = \frac{1}{n} \sum_{i=1}^n (X_i - Y_i)^2 \end{array} \right.$$

MSE (*mean square error*) adalah metode evaluasi untuk mengukur tingkat error atau kesalahan .

Keterangan :

$n$  = jumlah piksel untuk gambar

$X_i$  dan  $Y_i$  = nilai gradasi piksel ke- $i$  untuk masing-masing citra X dan Y.

## **BAB 3**

### **ANALISIS DAN PERANCANGAN**

#### **3.1 Analisis Sistem**

Dalam membangun sistem yang sistematis, diperlukan desain yang baik. Tahap awal dalam pengembangan sistem yang akan menggunakan teknik identifikasi elemen-elemen yang dibutuhkan adalah analisis sistem. Hal ini dilakukan untuk menjamin sistem berfungsi dengan baik dan komponen-komponen sistem beroperasi dan berinteraksi sesuai dengan tujuan dan sasaran sistem. Sistem ini memiliki tiga tahap analisis yakni sebagai berikut.

##### **3.1.1 Analisis Masalah**

Proses menemukan sumber-sumber kesulitan yang ada pada sistem yang ingin dirancang disebut sebagai analisis masalah. Hal ini dilakukan untuk menjamin bahwa sistem berfungsi sesuai dengan tujuan. Adapun pokok bahasan dalam penelitian ini yakni kompresi pada citra medis menggunakan algoritma *Fraktal* dan *Arithmetic Coding* untuk meminimalkan penggunaan memori agar lebih efisien dan mempersingkat pengiriman citra.

##### **3.1.2 Analisis Kebutuhan**

Dalam merancang sistem, analisis kebutuhan menjadi landasan penting untuk mengidentifikasi, mendefinisikan dan memahami kebutuhan yang diperlukan oleh sistem. Ada dua jenis analisis kebutuhan, diantaranya sebagai berikut:

###### **3.1.2.1 Kebutuhan Fungsional**

Jenis kebutuhan yang menjelaskan apa yang sistem atau aplikasi harus lakukan disebut kebutuhan fungsional. Kebutuhan fungsional menjelaskan secara rinci tentang fungsi, fitur, dan proses yang harus dijalankan pada sistem atau aplikasi. Berikut kebutuhan fungsional dalam sistem ini:

1. Pada proses ini dapat meng-inputkan citra.

2. Sistem mampu mengambil nilai piksel pada citra berekstensi \*.jpg dan \*.png
3. Sistem mampu melakukan pengkompresian citra menggunakan algoritma *Fraktal* dan *Arithmetic Coding*.
4. Menghitung rasio kompresi, nilai PSNR, dan waktu kompresi.
5. Citra hasil kompresi ditampilkan

### 3.1.2.2 Kebutuhan Non-Fungsional

Jenis kebutuhan yang menjelaskan bagaimana sistem atau aplikasi harus melakukannya disebut sebagai kebutuhan non-fungsional. Dibawah ini merupakan kebutuhan non-fungsional dalam sistem ini:

#### 1. Penampilan (Performance)

Sistem yang dirancang mampu memberikan output berdasarkan kompresi dan dekompresi data file gambar.

#### 2. Keekonomis

Ketika digunakan, sistem harus berfungsi dengan baik dan tidak menghasilkan catatan eror.

#### 3. Penggunaan (Efesiensi)

Sistem yang dirancang harus mudah digunakan dan dijalankan oleh pengguna.

#### 4. Pemantauan (Pengendalian)

Sistem yang dirancang harus senantiasa diperiksa untuk memastikan kinerja dan kegunaan sistem selalu terjaga dan keluarannya sesuai dengan tujuan.

#### 5. Pengarsipan (Dokumentasi)

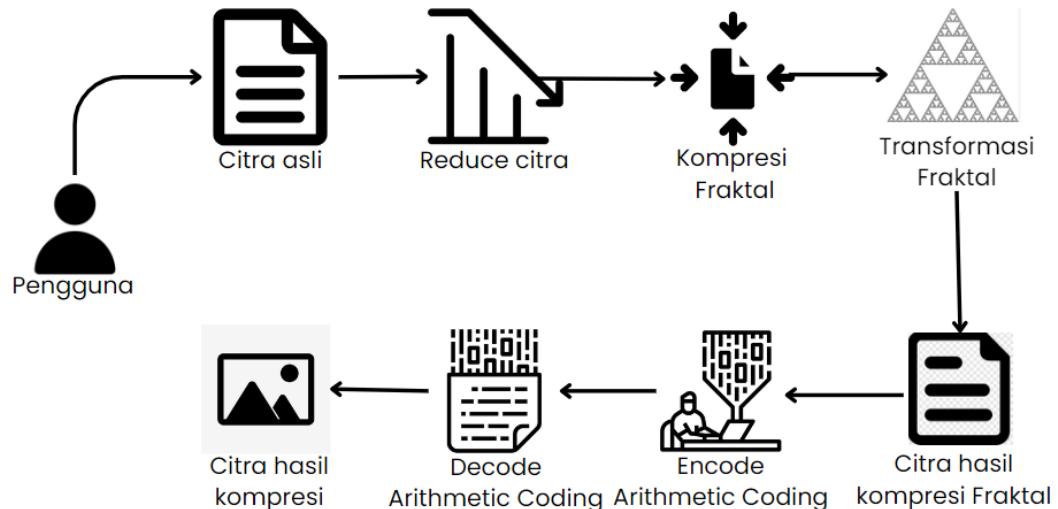
Sistem yang dirancang mampu menyimpan baik data yang akan disampaikan maupun data citra hasil.

#### 6. Manajemen Kualitas

Sistem yang dirancang harus berkualitas tinggi dan akurat dalam prosedur kompresi dan dekompresi.

### 3.1.3 Arsitektur Umum

Arsitektur umum adalah rancangan yang dapat digunakan untuk merepresentasikan prosedur, jalur dan interaksi antar komponen sistem.



**Gambar 3. 1** Arsitektur Umum Sistem

Berikut merupakan penjelasan alur proses arsitektur umum sistem pada gambar 3.2:

Proses kompresi dimulai dengan pengguna menginputkan citra asli, kemudian citra di *reduce* dengan nilai (2, 4, 8 dan 16). Citra ditampilkan dalam bentuk matriks yang mewakili nilai *grayscale* tiap piksel. Lalu akan dilakukan transformasi *Fraktal*. Dari proses tersebut, dihasilkan sebuah citra hasil kompresi *Fraktal*. Citra hasil kompresi *Fraktal* akan di-*encode* dan di-*decode* menggunakan algoritma *Arithmetic Coding*, lalu akan didapat citra hasil kompresi yang bersifat *lossless*.

### 3.1.4 Analisis Proses

Sistem dibuat menggunakan algoritma *Fraktal* dan algoritma *Arithmetic Coding*. Algoritma *Fraktal* menggunakan teknik pemetaan citra ke dalam dirinya sendiri dengan menggunakan transformasi matematika yang disebut transformasi *affine*. Algoritma ini bekerja dengan mencari kesamaan pola atau *Fraktal* yang terdapat pada citra dan kemudian menyimpan pola tersebut

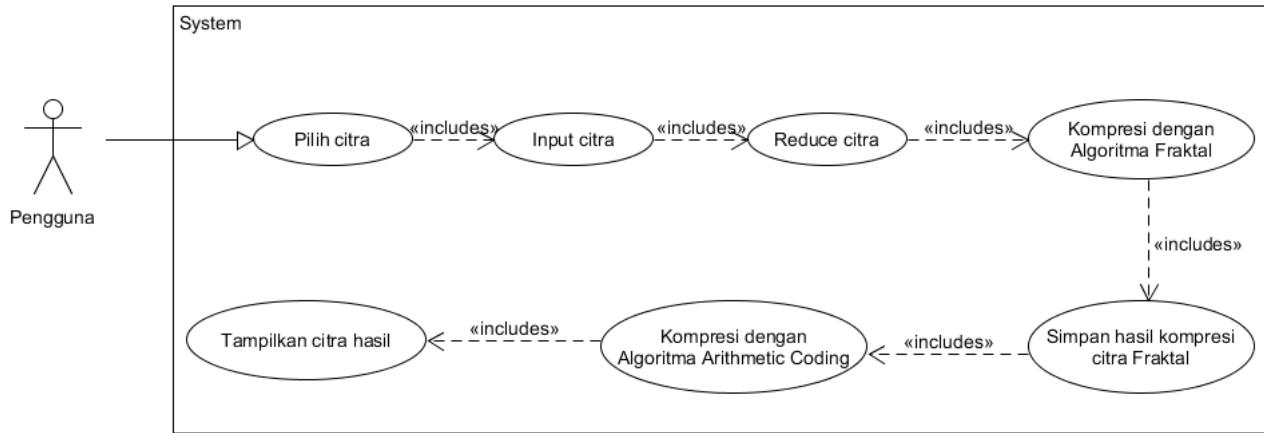
sebagai kode *Fraktal*. Kode *Fraktal* tersebut kemudian digunakan untuk merekonstruksi citra pada saat dekompresi. *Arithmetic coding* adalah metode kompresi citra yang menggunakan probabilitas kemunculan setiap simbol dalam citra untuk membentuk kode biner yang lebih pendek. Dalam kombinasi ini, algoritma *Fraktal* digunakan untuk mengidentifikasi dan merepresentasikan pola *Fraktal* pada citra, sedangkan *Arithmetic Coding* digunakan untuk mengompresi data yang dihasilkan oleh algoritma *Fraktal* menjadi kode biner yang lebih pendek.

### 3.2 Pemodelan Sistem

Pemodelan sistem digunakan untuk merepresentasikan sistem yang ada secara visual, sehingga dapat membantu dalam pemahaman sistem tersebut. Tujuan utama dari pemodelan sistem adalah untuk mengembangkan gambaran yang lebih baik tentang sistem dan memudahkan analisis, perencanaan, pengembangan, pengujian, dan perbaikan sistem. UML (*Unified Modeling Language*) merupakan salah satu bahasa pemodelan yang paling umum dipakai untuk memodelkan sistem. UML adalah standar industri yang ditetapkan secara luas yang memungkinkan pengguna untuk merepresentasikan struktur, perilaku, dan interaksi antara komponen dalam sebuah sistem.

#### 3.2.1 *Use Case Diagram*

*Use case diagram* adalah komponen rekayasa perangkat lunak yang merepresentasikan hubungan antara pengguna (aktor) dengan sistem. *Use case* diagram membantu dalam mengidentifikasi kinerja utama yang harus diimplementasikan dalam suatu sistem, serta bagaimana pengguna berinteraksi dengan fitur-fitur tersebut.

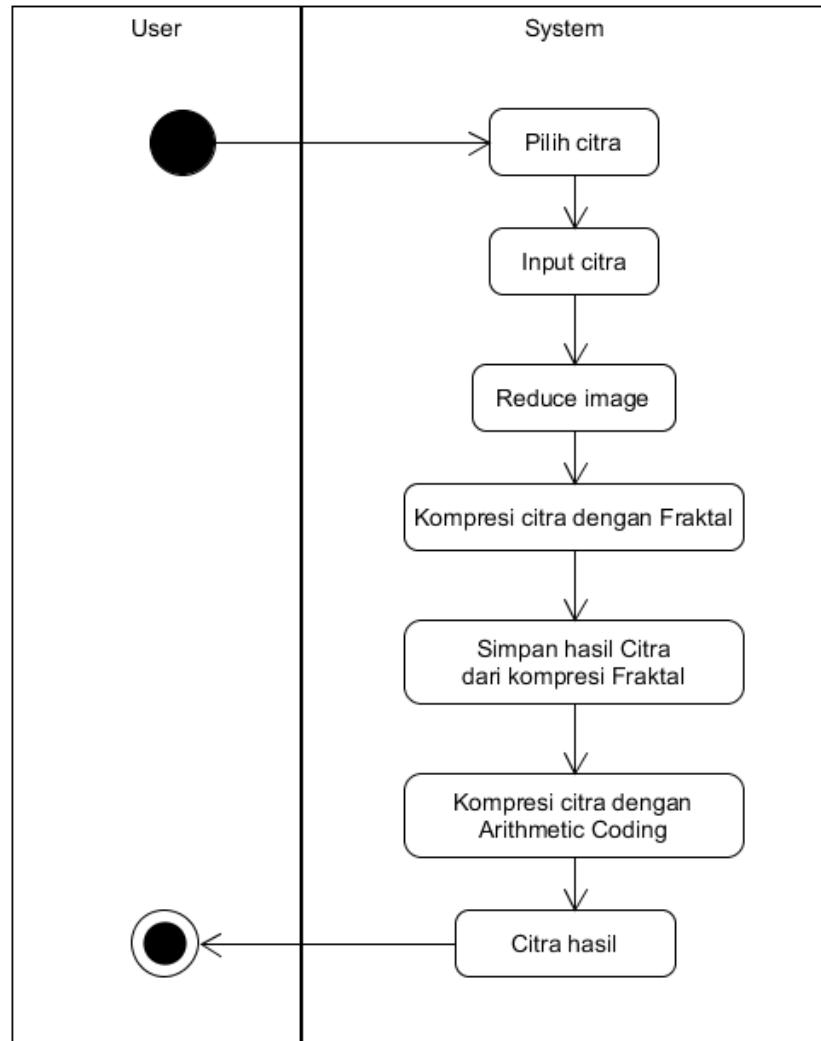


**Gambar 3. 2 Use Case Diagram Sistem**

Pada gambar 3.2 menampilkan satu aktor yaitu pengguna. Pengguna melakukan kompresi diawali dengan memilih citra, setelah itu citra akan diinput dan di-reduce. Selanjutnya akan dilakukan kompresi citra menggunakan algoritma *Fraktal* dan *Arithmetic Coding* lalu hasil kompresi akan ditampilkan.

### 3.2.2 Activity Diagram

Diagram aktivitas adalah alat visual yang dipakai dalam pemodelan sistem untuk merepresentasikan alur kerja atau proses sistem. Diagram ini termasuk dalam notasi *Unified Modeling Language* (UML) dan sering dipakai dalam pengembangan perangkat lunak untuk menggambarkan tahapan dalam suatu aktivitas atau proses.



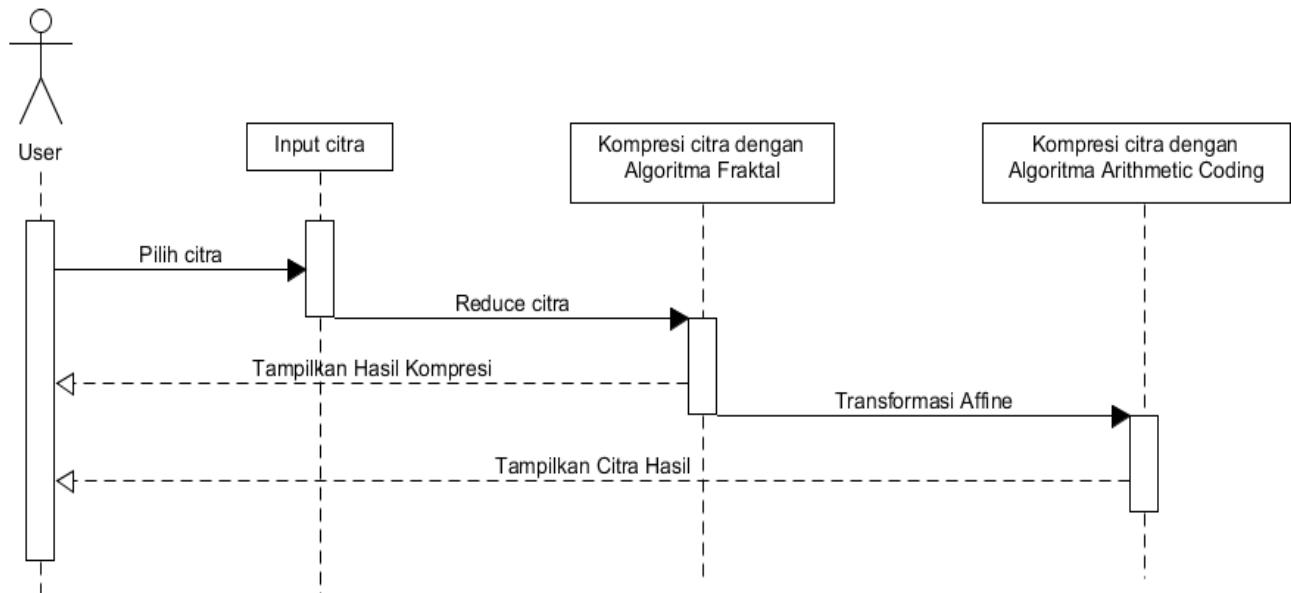
**Gambar 3. 3** Activity Diagram Sistem

Pada gambar 3.3 menampilkan alur aktivitas antara user dengan sistem ketika melakukan kompresi yang diawali dengan memilih citra, setelah itu citra akan diinput dan di-reduce. Selanjutnya akan dilakukan kompresi citra menggunakan algoritma *Fraktal* dan *Arithmetic Coding* lalu akan didapatkan citra hasil.

### 3.2.3 Sequence Diagram

Dalam bahasa pemodelan *Unified Modeling Language* (UML), *sequence* diagram adalah jenis diagram yang dipakai dalam mewakili interaksi elemen

dalam suatu sistem atau proses. Diagram ini menampilkan pemanggilan metode antar objek dalam urutan kronologis, yang merepresentasikan objek berinteraksi satu sama lain dalam satu skenario.



**Gambar 3. 4 Sequence Diagram Sistem**

Pada gambar 3.4 menampilkan urutan waktu antara user dan sistem. Dimana, interaksi antara user ke sistem ditunjukkan dengan tanda panah yang utuh dan tanggapan dari sistem ke user ditandai dengan tanda panah putus-putus.

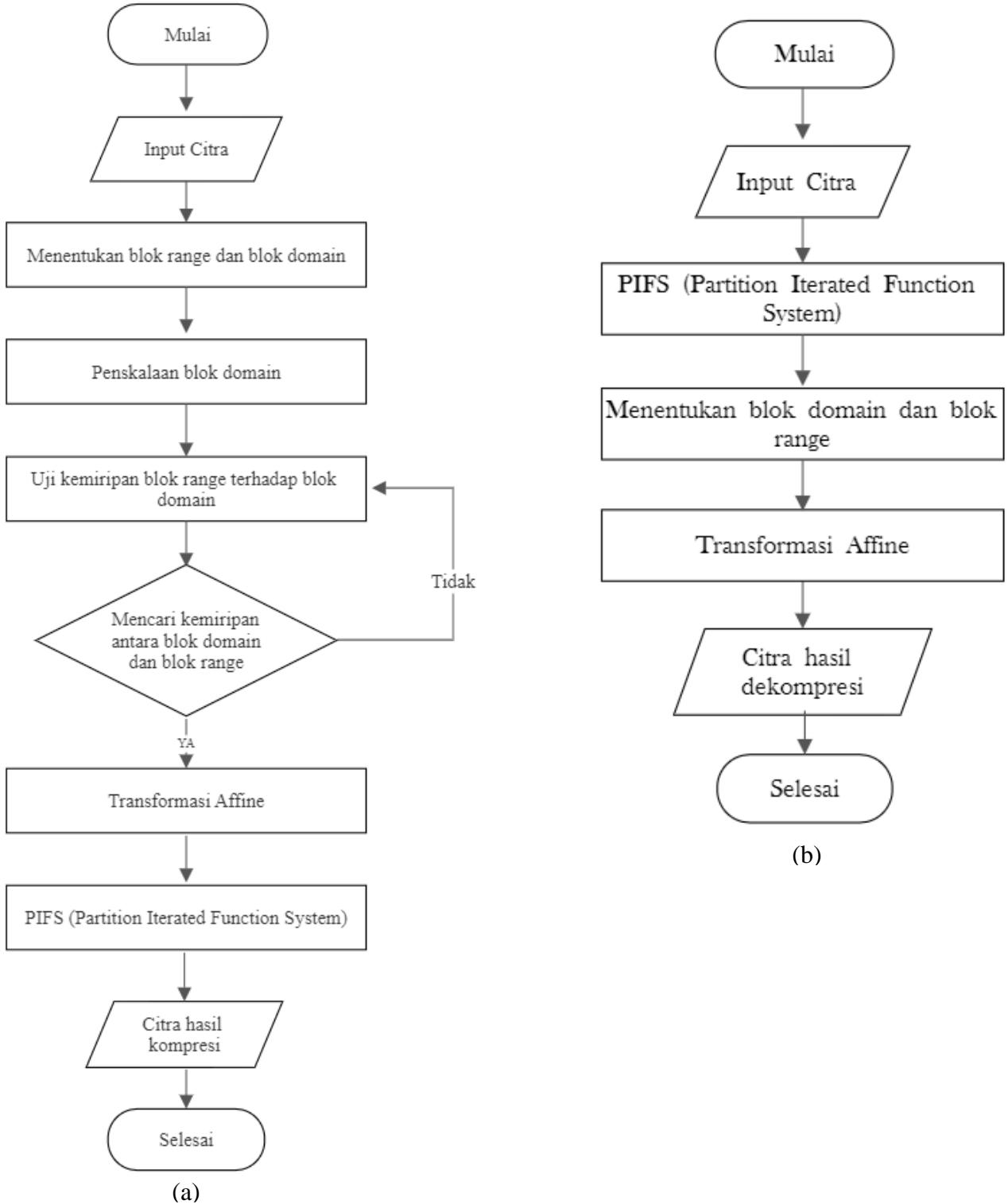
### 3.3. Diagram Alir (*flowchart*)

*Flowchart* (diagram alir) adalah representasi visual dari tahapan program dan proses prosedural. Tujuan mendasar diagram alur (*flowchart*) adalah untuk menggambarkan suatu metode pemecahan masalah dengan menggunakan simbol-simbol baku secara sederhana, rinci, rapi, dan jelas. Diagram alir penelitian ini didasarkan pada diagram alir representasi sistem standar, *flowchart* kompresi *Fraktal*, *flowchart* dekompresi *Fraktal*, *flowchart* kompresi *Arithmetic Coding* dan *flowchart* dekompresi *Arithmetic Coding*.

### 3.3.1 Flowchart Kompresi dan Dekompresi Algoritma *Fraktal*

Proses kompresi pada gambar 3.5 dimulai dengan memasukkan citra, lalu menentukan blok domain dan blok *range*. Selanjutnya dilakukan penskalaan blok domain dengan ukuran  $\frac{1}{2}$  kali ukuran semula. Langkah selanjutnya mencari kesamaan antara blok domain dengan blok *range* yang dihitung menggunakan rumus rms (*root mean square*). Kemudian menghitung transformasi *affine* antara blok domain dengan blok *range* terpilih, parameter transformasi *affine* kemudian disimpan di PIFS.

Pada Proses dekompresi, dilakukan dengan merekonstruksi citra sembarang yang ukurannya yang identik dengan citra aslinya. Kemudian sistem akan membaca parameter PIFS yang disimpan pada proses kompresi. Selanjutnya menentukan blok domain dan blok *range*. Lalu transformasi *affine* digunakan sebagai pembentukan ulang citra. Hasil akhir dari transformasi tersebut adalah citra keluaran yang bersifat *lossy*.

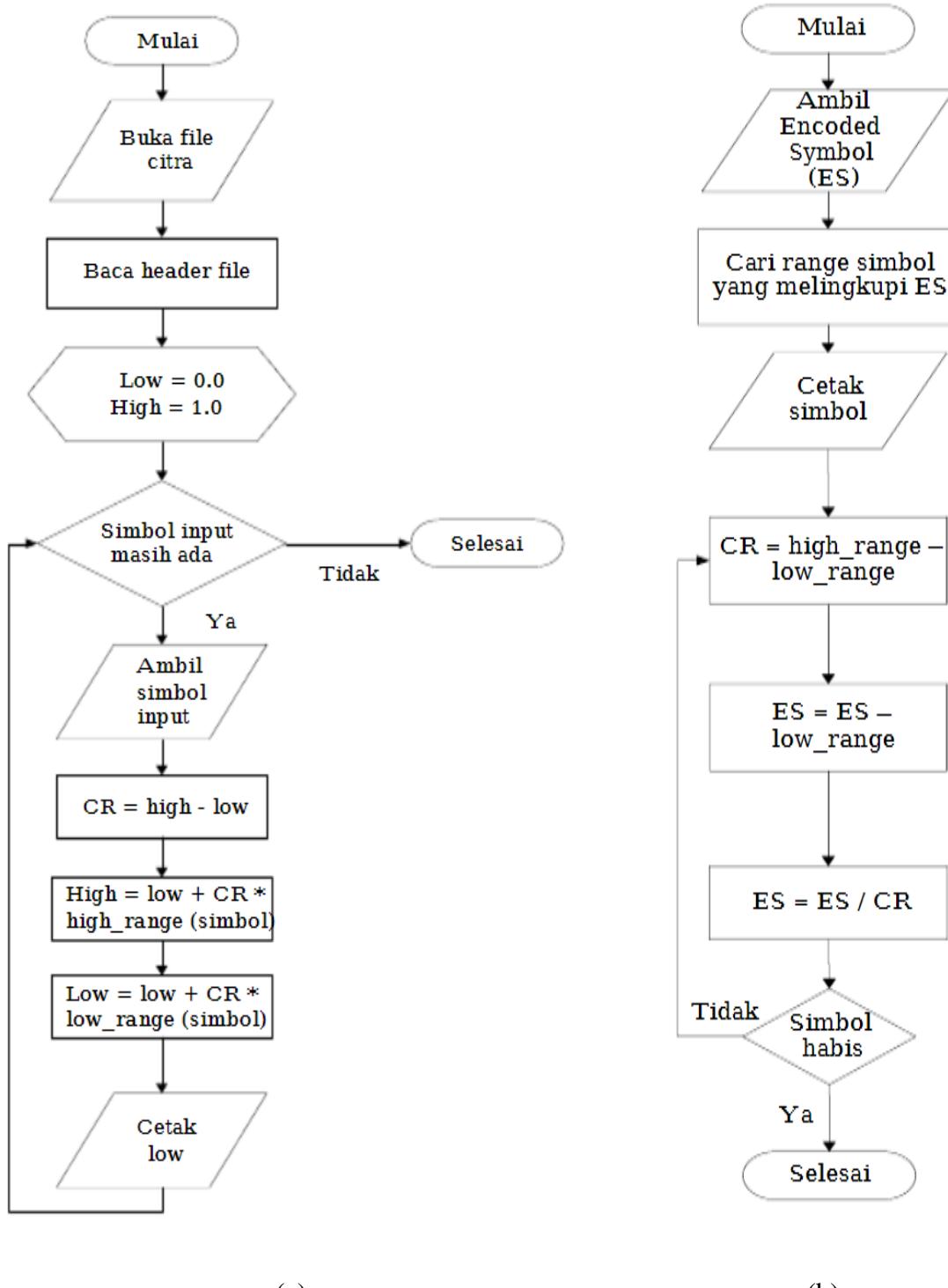


**Gambar 3.5** (a) *Flowchart Kompresi Fraktal* (b) *Flowchart Dekompressi Fraktal*

### 3.3.2 Flowchart Kompresi dan Dekompresi Algoritma *Arithmetic Coding*

Proses kompresi citra dengan algoritma *Arithmetic Coding* dimulai dengan memasukkan citra. Lalu, akan dipakai variabel *low* dan *high* menggambarkan interval (*low*, *high*) dengan kondisi awal nilai *low* = 0 dan *high* = 1. Kemudian akan dilakukan perulangan jika kondisi simbol input belum habis maka ambil simbol dan dilakukan proses perhitungan untuk mencari nilai code *range*, *high*, *low* dan nilai *low* akan dicetak. Perulangan dilakukan sampai simbolnya habis.

Proses dekompresi citra dengan algoritma *Arithmetic Coding* dimulai dengan ambil *encoded symbol* (ES) yang diambil dari data terakhir nilai *low* pada proses kompresi. Lalu akan dilakukan pencarian *range* dari tiap simbol yang mencakup *encoded symbol* (ES) lalu simbol akan dicetak. Kemudian dilakukan proses perhitungan *code range* dan *encoded symbol* sampai simbol habis.



**Gambar 3. 6** (a) *Flowchart Kompresi Arithmetic Coding* (b) *Flowchart Dekompresi Arithmetic Coding*

## BAB 4

### IMPLEMENTASI DAN PENGUJIAN

#### 4.1. Implementasi Sistem

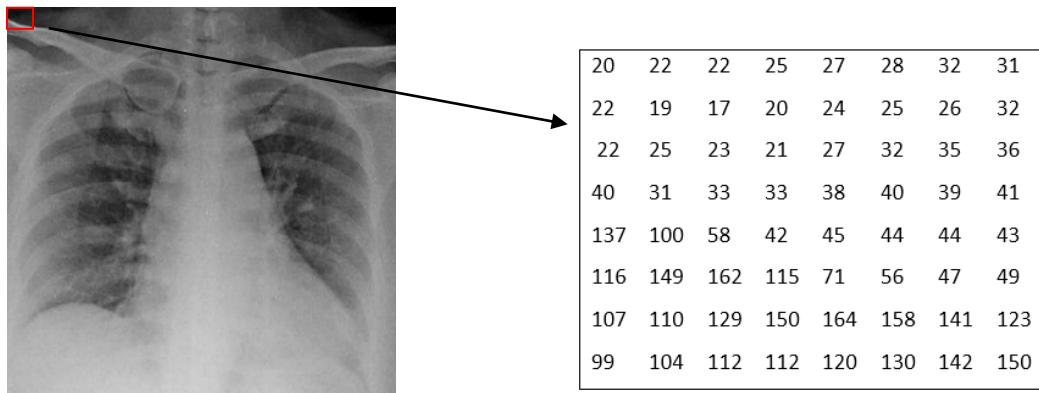
Implementasi sistem mengacu pada proses konversi desain sistem yang dibuat ke dalam bentuk nyata yang berfungsi. Implementasi Kompresi Citra Medis Sinar-x Dada Menggunakan Algoritma *Fraktal* dan *Arithmetic Coding* membutuhkan spesifikasi perangkat yang tercantum pada tabel 4.1 di bawah ini:

**Tabel 4. 1** Spesifikasi Jenis Perangkat Implementasi

Perangkat	Jenis Perangkat	Spesifikasi
Perangkat Keras	Prosesor	Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz
	Memori	8GB DDR4
Perangkat Lunak	Sistem Operasi	Windows 10
	Perangkat Pengembang	Google Colab

##### 4.1.1 Kompresi *Fraktal* pada Citra Medis Sinar-x Dada

Berikut merupakan contoh potongan piksel dari sebuah citra yang akan dilakukan proses perhitungan pada kompresi Fraktal.



**Gambar 4. 1** Contoh Potongan Piksel pada Citra

Berikut adalah proses kompresi citra yang diterapkan pada piksel-piksel di atas dengan menggunakan algoritma *Fraktal* :

- Menentukan ukuran blok domain dan ukuran blok *range* dan membagi citra.

20	22	22	25	27	28	32	31
22	19	17	20	24	25	26	32
22	25	23	21	27	32	35	36
40	31	33	33	38	40	39	41
137	100	58	42	45	44	44	43
116	149	162	115	71	56	47	49
107	110	129	150	164	158	141	123
99	104	112	112	120	130	142	150

**Gambar 4. 2 Blok Domain**

Blok-blok yang berukuran lebih besar ini disebut dengan blok ranah (domain block). Blok-blok ranah adalah blok-blok yang saling tumpang tindih. Gambar diatas merupakan blok domain yang berukuran 8x8.

<b>R<sub>1</sub></b>				<b>R<sub>2</sub></b>			
20	22	22	25	27	28	32	31
22	19	17	20	24	25	26	32
22	25	23	21	27	32	35	36
40	31	33	33	38	40	39	41
137	100	58	42	45	44	44	43
116	149	162	115	71	56	47	49
107	110	129	150	164	158	141	123
99	104	112	112	120	130	142	150

**R<sub>1</sub>**                           **R<sub>2</sub>**

**Gambar 4. 3 Blok Range**

Blok-blok yang berukuran lebih kecil ini disebut dengan blok jelajah (range block). Blok-blok jelajah adalah blok-blok yang tidak tumpang tindih. Gambar diatas merupakan blok range yang tiap rangenya berukuran 4x4.

2. Blok domain diskalakan menjadi 1/2 kali ukuran semula, sebagai berikut :

a. Nilai domain pada *range* pertama :

$$\frac{20 + 22 + 22 + 19}{4} = \frac{83}{4} = 20.75$$

$$\frac{22 + 25 + 17 + 20}{4} = \frac{84}{4} = 21$$

$$\frac{22 + 25 + 40 + 31}{4} = \frac{117}{4} = 29.25$$

$$\frac{23 + 21 + 33 + 33}{4} = \frac{110}{4} = 27.5$$

b. Nilai domain pada *range* kedua :

$$\frac{27 + 28 + 24 + 25}{4} = \frac{104}{4} = 26$$

$$\frac{32 + 31 + 26 + 32}{4} = \frac{121}{4} = 30.25$$

$$\frac{27 + 32 + 38 + 40}{4} = \frac{137}{4} = 34.25$$

$$\frac{35 + 36 + 39 + 41}{4} = \frac{151}{4} = 37.75$$

c. Nilai domain pada *range* ketiga :

$$\frac{137 + 100 + 116 + 149}{4} = \frac{502}{4} = 125.5$$

$$\frac{58 + 42 + 162 + 115}{4} = \frac{377}{4} = 94.25$$

$$\frac{107 + 110 + 99 + 104}{4} = \frac{420}{4} = 105$$

$$\frac{129 + 150 + 112 + 112}{4} = \frac{503}{4} = 125.75$$

d. Nilai domain pada *range* keempat :

$$\frac{45 + 44 + 71 + 56}{4} = \frac{216}{4} = 54$$

$$\frac{44 + 43 + 47 + 49}{4} = \frac{183}{4} = 45.75$$

$$\frac{164 + 158 + 120 + 130}{4} = \frac{572}{4} = 143$$

$$\frac{141 + 123 + 142 + 150}{4} = \frac{556}{4} = 139$$

Dari hasil perhitungan diatas maka diperoleh blok domain yang diskalakan yaitu :

20.75	21	26	30.25
29.5	27.5	34.25	37.75
125.5	94.25	54	45.75
105	125.75	143	139

**Gambar 4.4** Blok Domain yang diskalakan

Selanjutnya, akan dilakukan pencarian kemiripan atau kesamaan antara blok *range* dan blok domain yang telah diperkecil menjadi setengah dari ukuran aslinya dengan menggunakan RMS. Adapun rumus untuk menghitung RMS adalah :

$$d_{rms} = \frac{1}{n} \sqrt{\sum_{i=1}^n \sum_{j=1}^n (z'_{ij} - z_{ij})^2} \dots \dots \dots \dots \dots (4.1)$$

Keterangan :

$z$  = intensitas nilai pada blok domain

$z'$  = intensitas nilai pada blok *range*

$n$  = jumlah piksel citra

1. Menghitung RMS blok *range* pertama

20	22	22	25
22	19	17	20
22	25	23	21
40	31	33	33

**Gambar 4. 5 Blok Range<sub>1</sub>**

$$\begin{aligned}
 d_{rms} &= \frac{1}{n} \sqrt{\sum_{i=1}^n \sum_{j=1}^n (z'_{ij} - z_{ij})^2} \\
 (d_{rms})_1 &= \frac{1}{16} \sqrt{(z'_{1-1}-z_1)^2 + (z'_{2-2}-z_2)^2 + (z'_{3-3}-z_3)^2 + \dots + (z'_{16-16}-z_{16})^2} \\
 &= \frac{1}{16} \sqrt{(20 - 20.75)^2 + (22 - 21)^2 + (22 - 26)^2 + \dots + (33 - 139)^2} \\
 &= \frac{1}{16} \sqrt{(-0.75)^2 + (1)^2 + (-4)^2 + \dots + (-106)^2} \\
 &= \frac{1}{16} \sqrt{53846.685} \\
 &= \frac{232846.68}{16} \\
 (d_{rms})_1 &= 14.5
 \end{aligned}$$

2. Menghitung RMS blok *range* kedua

27	28	32	31
24	25	26	32
27	32	35	36
38	40	39	41

**Gambar 4. 6 Blok Range<sub>2</sub>**

$$\begin{aligned}
 (d_{rms})_2 &= \frac{1}{n} \sqrt{\sum_{i=1}^n \sum_{j=1}^n (z'_{ij} - z_{ij})^2} \\
 (d_{rms})_2 &= \frac{1}{16} \sqrt{(z'_{1-1}-z_1)^2 + (z'_{2-2}-z_2)^2 + (z'_{3-3}-z_3)^2 + \dots + (z'_{16-16}-z_{16})^2}
 \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{16} \sqrt{(27 - 20.75)^2 + (28 - 21)^2 + (32 - 26)^2 + \dots + (41 - 139)^2} \\
&= \frac{1}{16} \sqrt{(6.25)^2 + (7)^2 + (6)^2 + \dots + (-98)^2} \\
&= \frac{1}{16} \sqrt{(46557.6875)} \\
&= \frac{215.772305}{16} \\
(\mathbf{d}_{\text{rms}})_2 &= 13.48
\end{aligned}$$

### 3. Menghitung RMS blok *range* ketiga

137	100	58	42
116	149	162	115
107	110	129	150
99	104	112	112

**Gambar 4.7** Blok Range<sub>3</sub>

$$\begin{aligned}
(\mathbf{d}_{\text{rms}})_3 &= \frac{1}{n} \sqrt{\sum_{i=1}^n \sum_{j=1}^n (z'_{ij} - z_{ij})^2} \\
(\mathbf{d}_{\text{rms}})_3 &= \frac{1}{16} \sqrt{(z'_1 - z_1)^2 + (z'_2 - z_2)^2 + (z'_3 - z_3)^2 + \dots + (z'_{16} - z_{16})^2} \\
&= \frac{1}{16} \sqrt{(137 - 20.75)^2 + (100 - 21)^2 + (58 - 26)^2 + \dots + (112 - 139)^2} \\
&= \frac{1}{16} \sqrt{(116.25)^2 + (79)^2 + (32)^2 + \dots + (-27)^2} \\
&= \frac{1}{16} \sqrt{(84731.6875)} \\
&= \frac{291.087079}{16} \\
(\mathbf{d}_{\text{rms}})_3 &= 18.19
\end{aligned}$$

### 4. Menghitung RMS blok *range* keempat

45	44	44	43
71	56	47	49
164	158	141	123
120	130	142	150

**Gambar 4. 8 Blok Range<sub>4</sub>**

$$\begin{aligned}
 (\mathbf{d}_{\text{rms}})_4 &= \frac{1}{n} \sqrt{\sum_{i=1}^n \sum_{j=1}^n (z'_{ij} - z_{ij})^2} \\
 (\mathbf{d}_{\text{rms}})_4 &= \frac{1}{16} \sqrt{(z'_1 - z_1)^2 + (z'_2 - z_2)^2 + (z'_3 - z_3)^2 + \dots + (z'_{16} - z_{16})^2} \\
 &= \frac{1}{16} \sqrt{(24.25)^2 + (23)^2 + (22)^2 + \dots + (11)^2} \\
 &= \frac{1}{16} \sqrt{(24035.1875)} \\
 &= \frac{155.032859}{16} \\
 (\mathbf{d}_{\text{rms}})_4 &= 9.68
 \end{aligned}$$

Setelah didapat nilai  $d_{\text{rms}}$  dari tiap blok domain terhadap blok *range*, kemudian menentukan  $d_{\text{rms}}$  yang mempunyai nilai terkecil. Nilai  $d_{\text{rms}}$  yang terkecil ini menunjukkan bahwa persentase kemiripan yang paling tinggi. Dari perhitungan diatas, yang mempunyai nilai  $d_{\text{rms}}$  terkecil adalah  $(d_{\text{rms}})_2$  dan  $(d_{\text{rms}})_4$ .

Selanjutnya dengan menggunakan persamaan (2.2) dilakukan proses perhitungan transformasi *affine* antara blok *range* ke *i* dengan blok domain yang dipilih. Dimana  $w_i$  memetakan titik  $(x,y)$  dengan intensitas  $z$  yang terdapat dalam blok domain,  $o_i$  mewakili offset kecerahan (nilai  $o_i$  negatif akan membuat citra menjadi gelap dan nilai  $o_i$  positif meningkatkan kecerahan gambar), dan parameter  $s_i$  adalah kontras ( $s_i=0$  maka piksel menjadi gelap,  $s_i=1$  maka kontrasnya tidak berubah,  $s_i$  diantara 0 dan 1 akan mengurangi kontras,  $s_i$  bernilai di atas 1 maka kontras akan meningkat),  $e$  dan  $f$

merupakan perpindahan sudut kiri atas blok domain ke sudut kiri atas blok *range* yang bersesuaian.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & S_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix}$$

Menghitung nilai s :

20	22	22	25
22	19	17	20
22	25	23	21
40	31	33	33

**Gambar 4.9** Blok *Range* Terpilih

$$s = \frac{[n \sum_{i=1}^n d_i r_i - \sum_{i=1}^n d_i \sum_{i=1}^n r_i]}{[n \sum_{i=1}^n d_i^2 - (\sum_{i=0}^n d_i)^2]}$$

$$s = \frac{[16 \sum_{i=1}^{16} d_i r_i - \sum_{i=1}^{16} d_i \sum_{i=1}^{16} r_i]}{[16 \sum_{i=1}^{16} d_i^2 - (\sum_{i=0}^{16} d_i)^2]}$$

$$s = \frac{[16 ((d_1 r_1) + (d_2 r_2) + (d_3 r_3) + \dots + (d_{16} r_{16})) - (d_1 + d_2 + d_3 + \dots + d_{16}) - (r_1 + r_2 + r_3 + \dots + r_{16})]}{16 (d_1^2 + d_2^2 + d_3^2 + \dots + d_{16}^2) - (d_1 + d_2 + d_3 + \dots + d_{16})^2}$$

$$s = \frac{[16 ((20.75 \times 20) + (21 \times 22) + \dots + (139 \times 33)) - (20.75 + 21 + 26 + \dots + 139) - (20 + 22 + 22 + \dots + 33)]}{16 (20.75^2 + 21^2 + 26^2 + \dots + 139^2) - (20.75 + 21 + 26 + \dots + 139)^2}$$

$$= \frac{16(29479.25) - 664.25}{16(102937.688) - 1122010.56}$$

$$= \frac{471668 - 664.25}{1647003.01 - 1122010.56}$$

$$= \frac{471003.75}{524992.45}$$

$$s = 0.89 \text{ (Kontras citra berkurang)}$$

Menghitung nilai o :

$$o = \frac{1}{n} \left[ \sum_{i=1}^n r_i - s \sum_{i=1}^n d_i \right]$$

$$o = \frac{1}{16} \left[ \sum_{i=1}^{16} r_i - s \sum_{i=1}^{16} d_i \right]$$

$$= \frac{[(r_1 + r_2 + r_3 + \dots + r_{16}) - 3.92(d_1 + d_2 + d_3 + \dots + d_{16})]}{16}$$

$$= \frac{[(20+22+22+\dots+33)-0.89(20.75+21+26+\dots+139)]}{16}$$

$$= \frac{395 - 0.89 (1059.25)}{16}$$

$$= \frac{395 - 942.7325}{16}$$

$$= \frac{-547.7325}{16}$$

$$= -34.23 \text{ (kecerahan citra akan menjadi gelap)}$$

Perhitungan parameter  $ei$  dan  $fi$  menjadi sederhana karena keduanya hanya merupakan perpindahan sudut kiri atas blok domain ke sudut kiri atas blok range yang sesuai. Oleh karena itu, nilai dari  $ei$  dan  $fi$  adalah sebagai berikut:

$$ei = 6$$

$$fi = 1$$

Setelah memperoleh semua parameter transformasi *affine*, maka kondisi berikut terpenuhi.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 0.89 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 6 \\ 1 \\ -34.23 \end{bmatrix}$$

#### 4.1.2 Kompresi *Arithmetic Coding* pada Citra Medis Sinar-x Dada

Contoh penerapan algoritma *Arithmetic Coding* pada citra adalah misalkan pada citra hasil kompresi *Fractal* yang terdapat pada gambar 3.3 akan dilakukan proses *encoding*. Pada tahap ini, akan ditentukan probabilitas untuk setiap piksel. Setelah probabilitas setiap piksel telah ditentukan, setiap piksel akan diberi nilai antara 0 dan 1, berdasarkan probabilitas yang sudah ditentukan.

<b>22</b>	<b>22</b>	<b>22</b>	<b>25</b>	30	30	29	30
<b>20</b>	<b>25</b>	<b>22</b>	<b>22</b>	25	30	30	30
<b>20</b>	<b>25</b>	<b>22</b>	<b>20</b>	30	35	35	40
<b>35</b>	<b>30</b>	<b>35</b>	<b>35</b>	35	38	40	40
107	100	68	50	35	20	50	60
145	133	123	118	100	60	60	78
130	118	130	118	140	140	120	118
118	110	130	118	130	140	140	149

**Gambar 4. 10** Matriks Citra *Grayscale* 8x8 Piksel

Matriks di atas menunjukkan citra *grayscale* yang berukuran 8x8 piksel. Kemudian, akan dihitung jumlah munculnya frekuensi setiap simbol pada citra. Setelah itu, probabilitas setiap simbol akan dihitung, dan setiap simbol akan diberi nilai antara 0 dan 1, berdasarkan probabilitas yang ada. Adapun nilai probabilitas tiap simbol tercantum pada tabel 4.2.

**Tabel 4. 2** Probabilitas dan *Range* untuk Gambar 4.10

No	Simbol	Frekuensi	Probabilitas	<i>Range</i>
1	22	6	$6/64=0,09375$	0 - 0,09375
2	25	4	$4/64=0,0625$	0,09375 - 0,15625
3	30	8	$8/64=0,125$	0,15625 - 0,28125
4	29	1	$1/64=0,015625$	0,28125 - 0,296875
5	20	4	$4/64=0,0625$	0,296875 - 0,359375

6	35	7	$7/64=0,109375$	0,359375 - 0,46875
7	40	3	$3/64=0,046875$	0,46875 - 0,515625
8	38	1	$1/64=0,015625$	0,515625 - 0,53125
9	107	1	$1/64=0,015625$	0,53125 - 0,546875
10	100	2	$2/64=0,03125$	0,546875 - 0,578125
11	68	1	$1/64=0,015625$	0,578125 - 0,59375
12	50	2	$2/64=0,03125$	0,59375 - 0,625
13	60	3	$3/64=0,046875$	0,625 - 0,671875
14	145	1	$1/64=0,015625$	0,671875 - 0,6875
15	133	1	$1/64=0,015625$	0,6875 - 0,703125
16	123	1	$1/64=0,015625$	0,703125 - 0,71875
17	118	6	$6/64=0,09375$	0,71875 - 0,8125
18	78	1	$1/64=0,015625$	0,8125 - 0,828125
19	130	4	$4/64=0,0625$	0,828125 - 0,890625
20	140	4	$4/64=0,0625$	0,890625 - 0,953125
21	120	1	$1/64=0,015625$	0,953125 - 0,96875
22	110	1	$1/64=0,015625$	0,96875 - 0,984375
23	149	1	$1/64=0,015625$	0,984375 - 1
Total		64		

Selanjutnya akan dilakukan proses *Arithmetic encoding* pada gambar 4.10 yaitu sebagai berikut:

### 1. Perhitungan Simbol 22

$$\text{Low\_range (22)} = 0,0$$

$$\text{High} = 1,0$$

$$\text{Low} = 0,0$$

$$\text{CR} = \text{High} - \text{Low}$$

$$= 1,0 - 0,0$$

$$= 1$$

$$\text{High\_range (22)} = 0,09375$$

Kemudian diperoleh nilai-nilai berikut:

$$\text{High} = \text{low} + \text{CR} * \text{high\_range (22)}$$

$$= 0 + 1 * 0,09375 = 0,09375$$

$$\text{Low} = \text{low} + \text{CR} * \text{low\_range (22)}$$

$$= 0 + 1 * 0,0 = 0,0$$

## 2. Perhitungan Simbol 22

$$High = 0,09375$$

$$Low = 0,0$$

$$CR = High - Low$$

$$= 0,09375 - 0,0$$

$$= 0,09375$$

$$High\_range(22) = 0,09375$$

$$Low\_range(22) = 0$$

Kemudian diperoleh nilai-nilai berikut:

$$High = low + CR * high\_range(22)$$

$$= 0 + 0,09375 * 0,09375$$

$$= 0,0087890625$$

$$Low = low + CR * low\_range(22)$$

$$= 0 + 0,09375 * 0$$

$$= 0$$

## 3. Perhitungan Simbol 22

$$High = 0,0087890625$$

$$Low = 0$$

$$CR = High - Low$$

$$= 0,0087890625 - 0$$

$$= 0,0087890625$$

$$High\_range(22) = 0,09375$$

$$Low\_range(22) = 0$$

Kemudian diperoleh nilai-nilai berikut:

$$High = low + CR * high\_range(22)$$

$$= 0 + 0,0087890625 * 0,09375$$

$$= 0,000823974609375$$

$$Low = low + CR * low\_range(22)$$

$$= 0 + 0,0087890625 * 0$$

$$= 0$$

## 4. Perhitungan Simbol 25

$$High = 0,000823974609375$$

$$Low = 0$$

$$CR = High - Low$$

$$= 0,000823974609375 - 0$$

$$= 0,000823974609375$$

$$High\_range(25) = 0,15625$$

$$Low\_range(25) = 0,09375$$

Kemudian diperoleh nilai-nilai berikut:

$$High = low + CR * high\_range(25)$$

$$= 0 + 0,000823974609375$$

$$* 0,15625$$

$$= 0,00012874603271484$$

$$Low = low + CR * low\_range(25)$$

$$= 0 + 0,000823974609375$$

$$* 0,9375$$

$$= 0,0000772476196289062$$

## 5. Perhitungan Simbol 20

$$High = 0,00012874603271484$$

$$Low = 0,0000772476196289062$$

$$CR = High - Low$$

$$= 0,00012874603271484 -$$

$$\begin{aligned} & 0,0000772476196289062 \\ & = 0,000051498413085938 \end{aligned}$$

$$High\_range (20) = 0,359375$$

$$Low\_range (20) = 0,29687$$

Kemudian diperoleh nilai-nilai berikut:

$$\begin{aligned} High &= low + CR * high\_range (20) \\ &= 0,0000772476196289062 + \\ &\quad 0,00005149841308593 * 0,2593 \end{aligned}$$

75

$$= 0,00009575486183167$$

$$\begin{aligned} Low &= low + CR * low\_range (20) \\ &= 0,000077247619628 + \\ &\quad 0,000051498413086844 * 0,296 \end{aligned}$$

87

$$= 0,0000925362110137939$$

## 6. Perhitungan Simbol 25

$$High = 0,00009575486183167$$

$$Low = 0,0000925362110137939$$

$$\begin{aligned} CR &= High - Low \\ &= 0,00009575486183167 - \\ &\quad 0,0000925362110137939 \\ &= 0,000003218650817871 \end{aligned}$$

$$High\_range (25) = 0,15625$$

$$Low\_range (25) = 0,09375$$

Kemudian diperoleh nilai-nilai berikut:

$$\begin{aligned} High &= low + CR * high\_range (25) \\ &= 0,0000925362110137939 + \end{aligned}$$

$$\begin{aligned} & 0,000003218650817871 \\ & * 0,15625 \\ & = 0,00009303912520409 \\ Low &= low + CR * low\_range (25) \\ &= 0,0000925362110137939 + \\ &\quad 0,000003218650817871 \\ & * 0,09375 \\ & = 0,0000928379595279694 \end{aligned}$$

## 7. Perhitungan Simbol 22

$$\begin{aligned} High &= 0,00009303912520409 \\ Low &= 0,0000928379595279694 \\ CR &= High - Low \\ &= 0,00009303912520409 - \\ &\quad 0,0000928379595279694 \\ &= 0,000000201165676117 \end{aligned}$$

$$High\_range (22) = 0,09375$$

$$Low\_range (22) = 0,0$$

Kemudian diperoleh nilai-nilai berikut:

$$\begin{aligned} High &= low + CR * high\_range (22) \\ &= 0,0000928379595279694 + \\ &\quad 0,000000201165676117 * 0,093 \\ 75 &= 0,00009285681881011 \\ Low &= low + CR * low\_range (22) \\ &= 0,0000928379595279694 + \\ &\quad 0,000000201165676117 * 0,0 \\ &= 0,0000928379595279694 \end{aligned}$$

### 8. Perhitungan Simbol 22

$$\begin{aligned}
 High &= 0,00009285681881011 \\
 Low &= 0,0000928379595279694 \\
 CR &= High - Low \\
 &= 0,00009285681881011 - \\
 &\quad 0,0000928379595279694 \\
 &= 0,000000018859282136
 \end{aligned}$$

$$High\_range (22) = 0,09375$$

$$Low\_range (22) = 0,0$$

Kemudian diperoleh nilai-nilai berikut:

$$\begin{aligned}
 High &= low + CR * high\_range (22) \\
 &= 0,0000928379595279694 + \\
 &\quad 0,000000018859282136 \\
 &\quad *0,09375 \\
 &= 0,00009283972758567
 \end{aligned}$$

$$\begin{aligned}
 Low &= low + CR * low\_range (22) \\
 &= 0,0000928379595279694 + \\
 &\quad 0,000000018859282136 * 0,0 \\
 &= 0,0000928379595279694
 \end{aligned}$$

### 9. Perhitungan Simbol 20

$$\begin{aligned}
 High &= 0,00009283972758567 \\
 Low &= 0,0000928379595279694 \\
 CR &= High - Low \\
 &= 0,00009283972758567 - \\
 &\quad 0,0000928379595279694 \\
 &= 0,0000000017680577
 \end{aligned}$$

$$High\_range (20) = 0,359375$$

$$Low\_range (20) = 0,296875$$

Kemudian diperoleh nilai-nilai berikut:

$$\begin{aligned}
 High &= low + CR * high\_range (20) \\
 &= 0,0000928379595279694 + \\
 &\quad 0,0000000017680577 \\
 &\quad *0,359375 \\
 &= 0,00009283859492371 \\
 Low &= low + CR * low\_range (20) \\
 &= 0,0000928379595279694 + \\
 &\quad 0,0000000017680577 \\
 &\quad *0,296875 \\
 &= 0,0000928384844200991
 \end{aligned}$$

### 10. Perhitungan Simbol 25

$$\begin{aligned}
 High &= 0,00009283859492371 \\
 Low &= 0,0000928384844200991 \\
 CR &= High - Low \\
 &= 0,00009283859492371 - \\
 &\quad 0,0000928384844200991 \\
 &= 0,000000000110503606
 \end{aligned}$$

$$High\_range (25) = 0,15625$$

$$Low\_range (25) = 0,09375$$

Kemudian diperoleh nilai-nilai berikut:

$$\begin{aligned}
 High &= low + CR * high\_range (25) \\
 &= 0,0000928384844200991 + \\
 &\quad 0,000000000110503606
 \end{aligned}$$

$$\begin{aligned}
 & *0,15625 & = 0,0000928384947798122 \\
 & = 0,00009283850168629 \\
 Low & = low + CR * low\_range (25) \\
 & = 0,0000928384844200991+ \\
 & \quad 0,000000000110503606 \\
 & *0,09375 \\
 & = 0,0000928384947798122
 \end{aligned}$$

### 11. Perhitungan Simbol 22

$$\begin{aligned}
 High & = 0,00009283850168629 \\
 Low & = 0,0000928384947798122 \\
 CR & = High-Low \\
 & = 0,00009283850168629- \\
 & \quad 0,0000928384947798122 \\
 & = 0,000000000006906475
 \end{aligned}$$

$$\begin{aligned}
 High\_range (22) & = 0,09375 \\
 Low\_range (22) & = 0,0
 \end{aligned}$$

Kemudian diperoleh nilai-nilai berikut:

$$\begin{aligned}
 High & = low + CR * high\_range (22) \\
 & = 0,0000928384947798122+ \\
 & \quad 0,000000000006906475 \\
 & *0,09375 \\
 & = 0,00009283849542729
 \end{aligned}$$

$$\begin{aligned}
 Low & = low + CR * low\_range (22) \\
 & = 0,0000928384947798122+ \\
 & \quad 0,000000000006906475*0,0
 \end{aligned}$$

### 12. Perhitungan Simbol 20

$$\begin{aligned}
 High & = 0,00009283849542729 \\
 Low & = 0,0000928384947798122 \\
 CR & = High-Low \\
 & = 0,00009283849542729- \\
 & \quad 0,0000928384947798122 \\
 & = 0,00000000000647482
 \end{aligned}$$

$$\begin{aligned}
 High\_range (20) & = 0,359375 \\
 Low\_range (20) & = 0,296875
 \end{aligned}$$

Kemudian diperoleh nilai-nilai berikut:

$$\begin{aligned}
 High & = low + CR * high\_range (20) \\
 & = 0,0000928384947798122+ \\
 & \quad 0,00000000000647482 \\
 & *0,359375 \\
 & = 0,0000928384950125
 \end{aligned}$$

$$\begin{aligned}
 Low & = low + CR * low\_range (20) \\
 & = 0,0000928384947798122+ \\
 & \quad 0,00000000000647482 \\
 & *0,296875 \\
 & = 0,0000928384949720334
 \end{aligned}$$

### 13. Perhitungan Simbol 35

$$\begin{aligned}
 High & = 0,0000928384950125 \\
 Low & = 0,0000928384949720334 \\
 CR & = High-Low \\
 & = 0,0000928384950125 -
 \end{aligned}$$

$\begin{aligned} & 0,0000928384949720334 \\ & = 0,00000000000000040468 \end{aligned}$ <p><math>High\_range (35) = 0,46875</math></p> <p><math>Low\_range (35) = 0,359375</math></p> <p>Kemudian diperoleh nilai-nilai berikut:</p> $\begin{aligned} High &= low + CR * high\_range (30) \\ &= 0,0000928384949865765+ \\ &\quad 0,0000000000000004426 \\ &\quad *0,28125 \\ &= 0,00009283849498782 \end{aligned}$ $\begin{aligned} Low &= low + CR * low\_range (30) \\ &= 0,0000928384949865765+ \\ &\quad 0,0000000000000004426 \\ &\quad *0,15625 \\ &= 0,0000928384949872681 \end{aligned}$ $\begin{aligned} Low &= low + CR * low\_range (35) \\ &= 0,0000928384949720334+ \\ &\quad 0,00000000000000040468 \\ &\quad *0,296875 \\ &= 0,0000928384949865765 \end{aligned}$	<p>Kemudian diperoleh nilai-nilai berikut:</p> $\begin{aligned} High &= low + CR * high\_range (35) \\ &= 0,0000928384949865765+ \\ &\quad 0,0000000000000004426 \\ &\quad *0,28125 \\ &= 0,00009283849498782 \end{aligned}$ $\begin{aligned} Low &= low + CR * low\_range (30) \\ &= 0,0000928384949865765+ \\ &\quad 0,0000000000000004426 \\ &\quad *0,15625 \\ &= 0,0000928384949872681 \end{aligned}$ <p><b>15. Perhitungan Simbol 35</b></p> $\begin{aligned} High &= 0,00009283849498782 \\ Low &= 0,0000928384949872681 \\ CR &= High - Low \\ &= 0,00009283849498782 - \\ &\quad 0,0000928384949872681 \\ &= 0,000000000000000553 \end{aligned}$ $\begin{aligned} High\_range (35) &= 0,46875 \\ Low\_range (35) &= 0,359375 \end{aligned}$ <p>Kemudian diperoleh nilai-nilai berikut:</p> $\begin{aligned} High &= low + CR * high\_range (35) \\ &= 0,0000928384949872681+ \\ &\quad 0,000000000000000553 \\ &\quad *0,359375 \\ &= 0,00009283849498753 \end{aligned}$
<p><b>14. Perhitungan Simbol 30</b></p> $\begin{aligned} High &= 0,000092838494991 \\ Low &= 0,0000928384949865765 \\ CR &= High - Low \\ &= 0,000092838494991 - \\ &\quad 0,0000928384949865765 \\ &= 0,000000000000000426 \end{aligned}$ $\begin{aligned} High\_range (30) &= 0,28125 \\ Low\_range (30) &= 0,15625 \end{aligned}$	<p><b>14. Perhitungan Simbol 30</b></p> $\begin{aligned} High &= 0,000092838494991 \\ Low &= 0,0000928384949865765 \\ CR &= High - Low \\ &= 0,000092838494991 - \\ &\quad 0,0000928384949865765 \\ &= 0,000000000000000426 \end{aligned}$ $\begin{aligned} High\_range (30) &= 0,28125 \\ Low\_range (30) &= 0,15625 \end{aligned}$

$$\begin{aligned}
 Low &= low + CR * low\_range (35) \\
 &= 0,0000928384949872681+ \\
 &\quad 0,0000000000000000553 \\
 &\quad *0,296875 \\
 &= 0,0000928384949874669
 \end{aligned}$$

$$\begin{aligned}
 High\_range (35) &= 0,46875 \\
 Low\_range (35) &= 0,359375
 \end{aligned}$$

Kemudian diperoleh nilai-nilai berikut:

$$\begin{aligned}
 High &= low + CR * high\_range (35) \\
 &= 0,0000928384949874669+ \\
 &\quad 0,000000000000000061 \\
 &\quad *0,359375 \\
 &= 0,00009283849498750 \\
 Low &= low + CR * low\_range (35) \\
 &= 0,0000928384949874669+ \\
 &\quad 0,000000000000000061 \\
 &\quad *0,296875 \\
 &= 0,0000928384949874887
 \end{aligned}$$

## 16. Perhitungan Simbol 35

$$\begin{aligned}
 High &= 0,00009283849498753 \\
 Low &= 0,0000928384949874669 \\
 CR &= High-Low \\
 &= 0,00009283849498753- \\
 &\quad 0,0000928384949874669 \\
 &= 0,000000000000000061
 \end{aligned}$$

Berikut hasil *encoding* yang tercantum pada tabel 4.3:

**Tabel 4. 3** Proses *Encoding* untuk Gambar 4.10

No	Simbol	Low	High	CR
		0,0	1,0	
1	22	0,0	0,09375	1,0
2	22	0,0	0,0087890625	0,09375
3	22	0,0	0,000823974609375	0,0087890625
4	25	0,0000772476196289062	0,00012874603271484	0,000823974609375
5	20	0,0000925362110137939	0,00009575486183167	0,000051498413085938

6	25	0,0000928379595279694	0,00009303912520409	0,000003218650817871
7	22	0,0000928379595279694	0,00009285681881011	0,000000201165676117
8	22	0,0000928379595279694	0,00009283972758567	0,000000018859282136
9	20	0,0000928384844200991	0,00009283859492371	0,0000000017680577
10	25	0,0000928384947798122	0,00009283850168629	0,000000000110503606
11	22	0,0000928384947798122	0,00009283849542729	0,000000000006906475
12	20	0,0000928384949720334	0,0000928384950125	0,000000000000647482
13	35	0,0000928384949865765	0,000092838494991	0,00000000000040468
14	30	0,0000928384949872681	0,00009283849498782	0,0000000000004426
15	35	0,0000928384949874669	0,00009283849498753	0,0000000000000553
16	35	0,0000928384949874887	0,0000928384949875	0,00000000000000061

Dari proses tersebut, data nilai *low* terakhir adalah 0,0000928384949874887 nilai ini kemudian akan dipakai untuk mewakili citra pada gambar 4.10 yang telah di-*encoding*.

Selanjutnya akan dilakukan proses *decoding* dari algoritma *Arithmetic Coding*, sebagai berikut :

$$\text{ES (encoded symbol)} = 0,0000928384949874887$$

$$\begin{aligned}
 & \text{1. Perhitungan Simbol 22 dengan} & \text{CR} &= \text{high\_range}(22) - \text{low\_range}(22) \\
 & \text{nilai ES} = 0,0000928384949874887 & &= 0,09375 - 0,0 \\
 & & &= 0,09375 \\
 & \text{High\_range (22)} & & \text{ES} = \text{ES} - \text{low\_range}(22) \\
 & \text{Low\_range (22)} & & 
 \end{aligned}$$

$$\begin{aligned}
 &= 0,0000928384949874887 - 0,0 \\
 &= 0,0000928384949874887 \\
 \text{ES} &= \text{ES} / \text{CR} \\
 &= 0,0000928384949874887 / \\
 &\quad 0,09375 \\
 &= 0,000990277279866546 \\
 &= 0,09375 \\
 &= 0,0105629576519098 - 0,0 \\
 &= 0,0105629576519098 \\
 \text{ES} &= \text{ES} / \text{CR} \\
 &= 0,0105629576519098 / 0,09375 \\
 &= 0,112671548287038
 \end{aligned}$$

## 2. Perhitungan Simbol 22 dengan nilai ES = 0,000990277279866546

$$\begin{aligned}
 \text{High\_range} (22) &= 0,09375 \\
 \text{Low\_range} (22) &= 0,0 \\
 \text{CR} &= \text{high\_range}(22) - \text{low\_range}(22) \\
 &= 0,09375 - 0,0 \\
 &= 0,09375 \\
 \text{ES} &= \text{ES} - \text{low\_range}(22) \\
 &= 0,000990277279866546 - 0,0 \\
 &= 0,000990277279866546 \\
 \text{ES} &= \text{ES} / \text{CR} \\
 &= 0,000990277279866546 / 0,09375 \\
 &= 0,0105629576519098
 \end{aligned}$$

## 3. Perhitungan Simbol 22 dengan nilai ES = 0,0105629576519098

$$\begin{aligned}
 \text{High\_range} (22) &= 0,09375 \\
 \text{Low\_range} (22) &= 0,0 \\
 \text{CR} &= \text{high\_range}(22) - \text{low\_range}(22) \\
 &= 0,09375 - 0,0
 \end{aligned}$$

## 4. Perhitungan Simbol 25 dengan nilai ES = 0,112671548287038

$$\begin{aligned}
 \text{High\_range} (25) &= 0,15625 \\
 \text{Low\_range} (25) &= 0,09375 \\
 \text{CR} &= \text{high\_range}(25) - \text{low\_range}(25) \\
 &= 0,15625 - 0,09375 \\
 &= 0,0625 \\
 \text{ES} &= \text{ES} - \text{low\_range}(25) \\
 &= 0,112671548287038 - 0,09375 \\
 &= 0,0189215482870381 \\
 \text{ES} &= \text{ES} / \text{CR} \\
 &= 0,0189215482870381 / 0,0625 \\
 &= 0,30274477259261
 \end{aligned}$$

## 5. Perhitungan Simbol 20 dengan nilai ES = 0,30274477259261

$$\begin{aligned}
 \text{High\_range} (20) &= 0,359375 \\
 \text{Low\_range} (20) &= 0,296875 \\
 \text{CR} &= \text{high\_range}(20) - \text{low\_range}(20)
 \end{aligned}$$

$$\begin{aligned}
 &= 0,359375 - 0,296875 \\
 &= 0,0625 \\
 \text{ES} &= \text{ES} - \text{low\_range}(20) \\
 &= 0,30274477259261 - 0,359375 \\
 &= 0,00586977259261001 \\
 \text{ES} &= \text{ES} / \text{CR} \\
 &= 0,00586977259261001 / 0,0625 \\
 &= 0,0939163614817602 \\
 &\quad \text{Low\_range (22)} = 0,0 \\
 &\quad \text{CR} = \text{high\_range}(22) - \text{low\_range}(22) \\
 &\quad = 0,09375 - 0,0 \\
 &\quad = 0,09375 \\
 \text{ES} &= \text{ES} - \text{low\_range}(22) \\
 &= 0,00266178370816306 - 0,0 \\
 &= 0,00266178370816306 \\
 \text{ES} &= \text{ES} / \text{CR} \\
 &= 0,00266178370816306 / 0,09375 \\
 &= 0,0283923595537393
 \end{aligned}$$

## 6. Perhitungan Simbol 25 dengan nilai ES = 0,0939163614817602

$$\begin{aligned}
 \text{High\_range (25)} &= 0,15625 \\
 \text{Low\_range (25)} &= 0,09375 \\
 \text{CR} &= \text{high\_range}(22) - \text{low\_range}(25) \\
 &= 0,15625 - 0,09375 \\
 &= 0,0625 \\
 \text{ES} &= \text{ES} - \text{low\_range}(25) \\
 &= 0,0939163614817602 - 0,09375 \\
 &= 0,000166361481760191
 \end{aligned}$$

$$\begin{aligned}
 \text{ES} &= \text{ES} / \text{CR} \\
 &= 0,000166361481760191 / \\
 &\quad 0,0625 \\
 &= 0,00266178370816306
 \end{aligned}$$

## 7. Perhitungan Simbol 22 dengan nilai ES = 0,00266178370816306

$$\text{High\_range (22)} = 0,09375$$

## 8. Perhitungan Simbol 22 dengan nilai ES = 0,0283923595537393

$$\begin{aligned}
 \text{High\_range (22)} &= 0,09375 \\
 \text{Low\_range (22)} &= 0,0 \\
 \text{CR} &= \text{high\_range}(22) - \text{low\_range}(22) \\
 &= 0,09375 - 0,0 \\
 &= 0,09375 \\
 \text{ES} &= \text{ES} - \text{low\_range}(22) \\
 &= 0,0283923595537393 - 0,0 \\
 &= 0,0283923595537393 \\
 \text{ES} &= \text{ES} / \text{CR} \\
 &= 0,0283923595537393 / 0,09375 \\
 &= 0,302851835239886
 \end{aligned}$$

## 9. Perhitungan Simbol 20 dengan nilai ES = 0,302851835239886

$$\begin{aligned}
 & High\_range(20) = 0,359375 \\
 & Low\_range(20) = 0,296875 \\
 & CR = high\_range(20) - low\_range(20) \\
 & \quad = 0,359375 - 0,296875 \\
 & \quad = 0,0625 \\
 & ES = ES - low\_range(20) \\
 & \quad = 0,302851835239886 - 0,359375 \\
 & \quad = 0,00597683523988557 \\
 & ES = ES / CR \\
 & \quad = 0,00597683523988557 / 0,0625 \\
 & \quad = 0,0956293638381691
 \end{aligned}$$

#### **10. Perhitungan Simbol 25 dengan nilai ES = 0,0956293638381691**

$$\begin{aligned}
 & High\_range(25) = 0,15625 \\
 & Low\_range(25) = 0,09375 \\
 & CR = high\_range(25) - low\_range(25) \\
 & \quad = 0,15625 - 0,09375 \\
 & \quad = 0,0625 \\
 & ES = ES - low\_range(25) \\
 & \quad = 0,0956293638381691 - 0,09375 \\
 & \quad = 0,00187936383816911
 \end{aligned}$$

$$\begin{aligned}
 & ES = ES / CR \\
 & \quad = 0,00187936383816911 / 0,0625 \\
 & \quad = 0,0300698214107058
 \end{aligned}$$

#### **11. Perhitungan Simbol 22 dengan nilai ES = 0,0300698214107058**

$$\begin{aligned}
 & High\_range(22) = 0,09375 \\
 & Low\_range(22) = 0,0 \\
 & CR = high\_range(22) - low\_range(22) \\
 & \quad = 0,09375 - 0,0 \\
 & \quad = 0,09375 \\
 & ES = ES - low\_range(22) \\
 & \quad = 0,0300698214107058 - 0,0 \\
 & \quad = 0,0300698214107058 \\
 & ES = ES / CR \\
 & \quad = 0,0300698214107058 / 0,09375 \\
 & \quad = 0,320744761714195
 \end{aligned}$$

#### **12. Perhitungan Simbol 20 dengan nilai ES = 0,320744761714195**

$$\begin{aligned}
 & High\_range(20) = 0,359375 \\
 & Low\_range(20) = 0,296875 \\
 & CR = high\_range(20) - low\_range(20) \\
 & \quad = 0,359375 - 0,296875 \\
 & \quad = 0,0625 \\
 & ES = ES - low\_range(20) \\
 & \quad = 0,320744761714195 - 0,359375 \\
 & \quad = 0,0238697617141952 \\
 & ES = ES / CR \\
 & \quad = 0,0238697617141952 / 0,0625 \\
 & \quad = 0,381916187427123
 \end{aligned}$$

**13. Perhitungan Simbol 35 dengan nilai ES = 0,381916187427123**

$$\text{High\_range}(35) = 0,46875$$

$$\text{Low\_range}(35) = 0,359375$$

$$\begin{aligned} \text{CR} &= \text{high\_range}(35) - \text{low\_range}(35) \\ &= 0,46875 - 0,39375 \\ &= 0,109377 \end{aligned}$$

$$\begin{aligned} \text{ES} &= \text{ES} - \text{low\_range}(35) \\ &= 0,381916187427123 - 0,359375 \\ &= 0,0225411874271231 \end{aligned}$$

$$\begin{aligned} \text{ES} &= \text{ES} / \text{CR} \\ &= 0,0225411874271231 / 0,109377 \\ &= 0,206087088026944 \end{aligned}$$

**14. Perhitungan Simbol 30 dengan nilai ES = 0,206087088026944**

$$\text{High\_range}(30) = 0,28125$$

$$\text{Low\_range}(30) = 0,15625$$

$$\begin{aligned} \text{CR} &= \text{high\_range}(20) - \text{low\_range}(20) \\ &= 0,28125 - 0,15625 \\ &= 0,125 \end{aligned}$$

$$\begin{aligned} \text{ES} &= \text{ES} - \text{low\_range}(20) \\ &= 0,206087088026944 - 0,15625 \\ &= 0,0498370880269442 \end{aligned}$$

$$\begin{aligned} \text{ES} &= \text{ES} / \text{CR} \\ &= 0,0498370880269442 / 0,125 \\ &= 0,398696704215554 \end{aligned}$$

**15. Perhitungan Simbol 35 dengan nilai ES = 0,398696704215554**

$$\text{Low\_range}(35) = 0,359375$$

$$\text{High\_range}(35) = 0,46875$$

$$\begin{aligned} \text{CR} &= \text{high\_range}(35) - \text{low\_range}(35) \\ &= 0,46875 - 0,39375 \\ &= 0,109377 \end{aligned}$$

$$\begin{aligned} \text{ES} &= \text{ES} - \text{low\_range}(35) \\ &= 0,398696704215554 - 0,359375 \\ &= 0,0393217042155536 \end{aligned}$$

$$\begin{aligned} \text{ES} &= \text{ES} / \text{CR} \\ &= 0,0393217042155536 / 0,109377 \\ &= 0,359375 \end{aligned}$$

**16. Perhitungan Simbol 35 dengan nilai ES = 0,359375**

$$\text{High\_range}(35) = 0,46875$$

$$\text{Low\_range}(35) = 0,359375$$

$$\begin{aligned} \text{CR} &= \text{high\_range}(35) - \text{low\_range}(35) \\ &= 0,46875 - 0,39375 \\ &= 0,109377 \end{aligned}$$

$$\begin{aligned} \text{ES} &= \text{ES} - \text{low\_range}(35) \\ &= 0,359375 - 0,359375 \\ &= 0,0 \end{aligned}$$

$$\begin{aligned} \text{ES} &= \text{ES} / \text{CR} \\ &= 0,0 / 0,109377 \\ &= 0,0 \end{aligned}$$

Karena nilai ES = 0, maka proses perhitungan dihentikan. Adapun data hasil *decoding* tercantum pada tabel 4.4 berikut

**Tabel 4. 4** Proses *Decoding* untuk Gambar 4.10

No	ES	Simbol	Low	High	CR
1	0,0000928384949874887	22	0	0,09375	0,09375
2	0,000990277279866546	22	0	0,09375	0,09375
3	0,0105629576519098	22	0	0,09375	0,09375
4	0,112671548287038	25	0,09375	0,15625	0,0625
5	0,30274477259261	20	0,296875	0,359375	0,0625
6	0,0939163614817602	25	0,09375	0,15625	0,0625
7	0,00266178370816306	22	0	0,09375	0,09375
8	0,0283923595537393	22	0	0,09375	0,09375
9	0,302851835239886	20	0,296875	0,359375	0,0625
10	0,0956293638381691000	25	0,09375	0,15625	0,0625
11	0,0300698214107058000	22	0	0,09375	0,09375
12	0,3207447617141950000	20	0,296875	0,359375	0,0625
13	0,3819161874271230000	35	0,359375	0,46875	0,109377
14	0,2060870880269440000	30	0,15625	0,28125	0,125
15	0,3986967042155540000	35	0,359375	0,46875	0,109377
16	0,359375000000000	35	0,359375	0,46875	0,109377

17	0,0	-	-	-	-
----	-----	---	---	---	---

Maka, diperoleh  $ES=0,0000928384949874887$  bersesuaian dengan citra pada gambar 4.10

## 4.2 Pengujian Sistem

Pada bagian ini, sistem diuji untuk mengevaluasi keberhasilan implementasi metode yang telah dirancang.

### 4.2.1 Data Pengujian

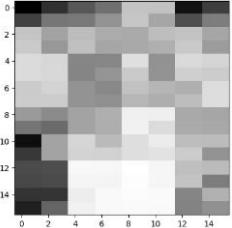
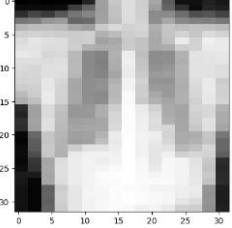
Pada skripsi ini data yang dipakai adalah citra medis sinar-x dada yang diambil dari dataset yang merupakan citra *grayscale* yang terdiri dari 30 gambar. Data citra ini akan diproses dengan algoritma *Fraktal* dan *Arithmetic Coding*. Pada skripsi ini akan digunakan citra dengan ukuran piksel 128x128, 256x256, 512x512, 1024x1024 dan 2048x2048 dengan ekstensi (.jpg) dan (.png), dan *reduce* citra  $2^n$  (2, 4, 8 dan 16).

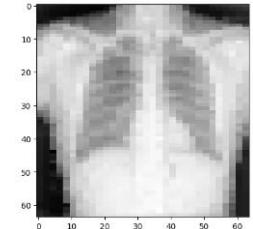
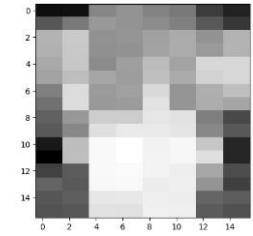
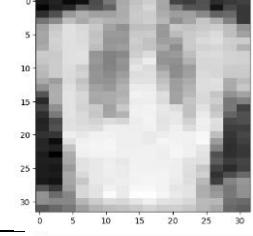
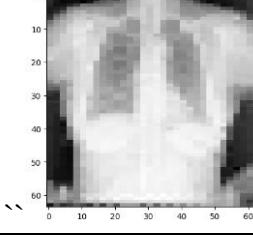
Selanjutnya akan dilihat citra hasil mana yang paling baik diantara ekstensi (.jpg) dan (.png) dengan parameter yang telah ditentukan yaitu waktu kompresi, rasio kompresi dan PSNR.

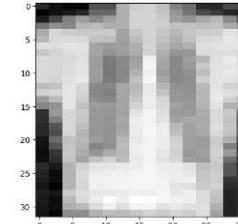
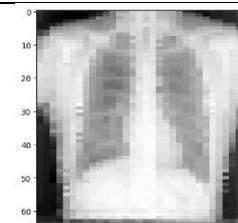
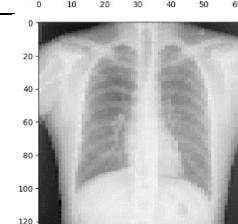
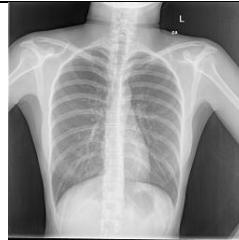
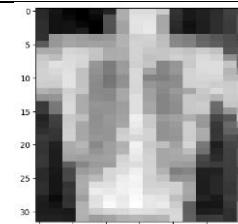
#### 4.2.2 Pengujian Kompresi Citra Medis Sinar-x Dada dengan Format PNG.

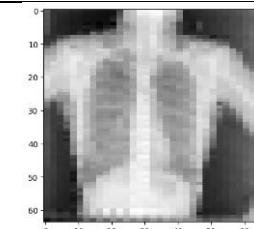
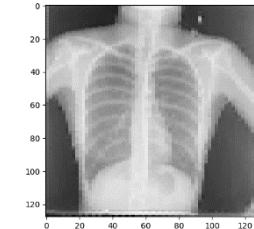
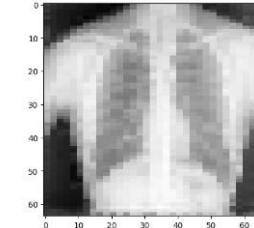
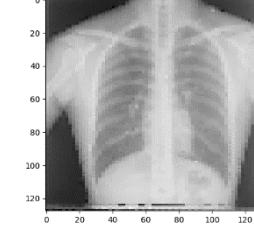
Citra medis sinar-x dada dikompres dengan menggunakan algoritma *Fraktal* dan *Arithmetic Coding*. Hasil uji coba kompresi dengan algoritma *Fraktal* dan *Arithmetic Coding* tercantum pada tabel 4.5.

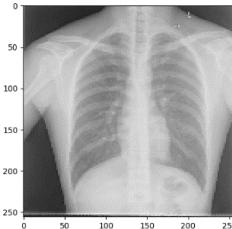
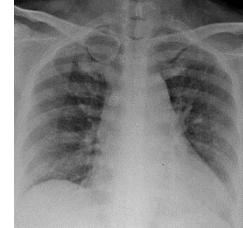
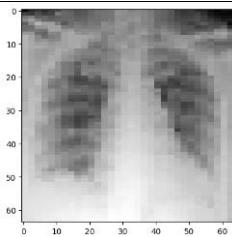
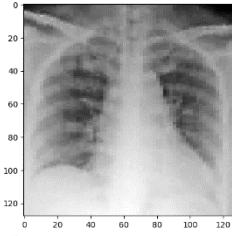
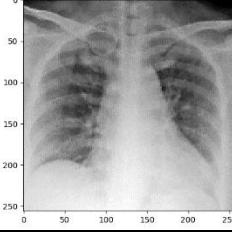
**Tabel 4. 5** Hasil Pengujian Citra dengan Ekstensi PNG

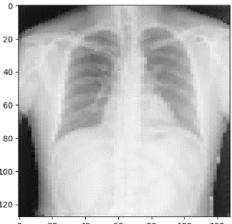
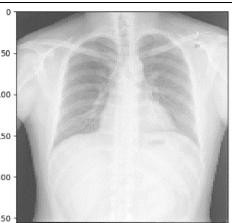
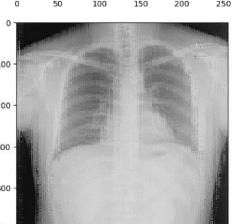
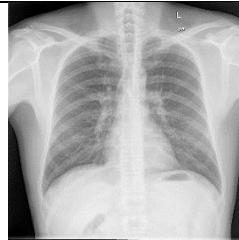
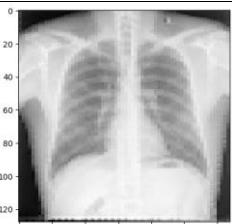
Citra Asli	Reduce Citra	Ukuran Citra (Piksel)		Ukuran Citra (Byte)		Waktu Kompresi (s)	PSNR (dB)	MSE	Rasio Kompresi	Citra Hasil
		Asli	Hasil	Asli	Hasil					
	8	128x128	16x16	17375	343	2.768	15.58	0.15	50.65	
	4		32x32		372					

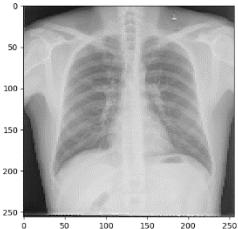
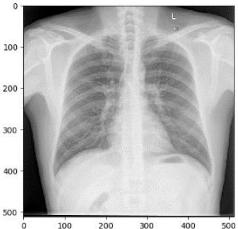
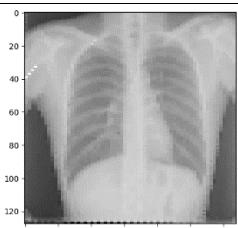
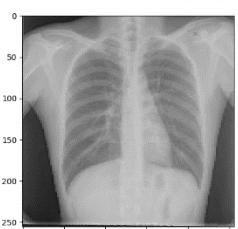
	2		64x64		441	7.327	26.56	0.04	39.39	
	8	128x128	16x16	16750	347	1.668	16.65	0.13	48.27	
	4		32x32		376	3.555	22.27	0.07	44.54	
	2		64x64		447	7.219	25.96	0.04	37.47	

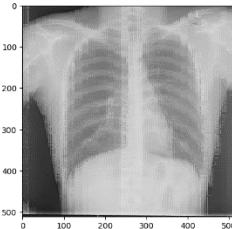
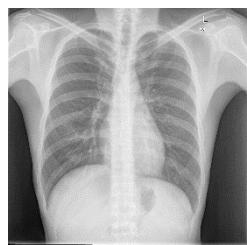
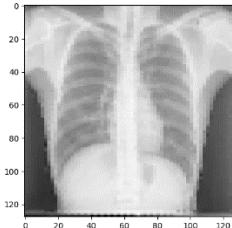
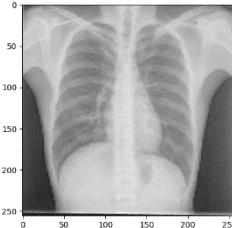
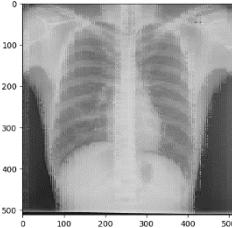
	8	256x256	32x32	367	3.242	22.19	0.07	159.32		
	4		64x64	58472	441	12.94	26.33	0.04		
	2		128x128	672	90.53	29.62	0.03	87.011		
	8	256x256	32x32	60325	374	1.824	21.95	0.07	161.296	

	4		64x64		451	18.107	24.53	0.05	133.75	
	2		128x128		709	77.65	28.65	0.03	85.04	
	8	512x512	64x64	218447	450	15.27	25.27	0.05	485.43	
	4		128x128		711	111.91	28.4	0.04	307.23	

	2		256x256		1576	1379.76	32.80	0.02	138.6	
	8	512x512	64x64	258986	461	7.554	29.48	0.02	561.791	
	4		128x128		735	78.4	33.14	0.01	352.361	
	2		256x256		1738	1249.73	35.53	0.01	149.01	

	8		128x128		661	109.73	28.02	0.03	1545.287	
	4	1024x1024	256x256	1021435	1475	1160.56	31.85	0.02	692.49	
	2		512x512		4467	18442.15	35.10	0.05	228.66	
	8	1024x1024	128x128	804804	715	76.466	30	0.03	1125.6	

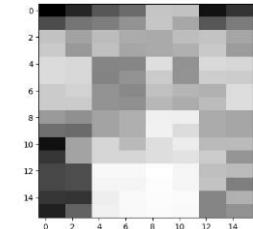
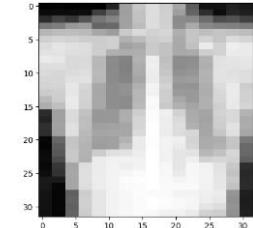
	4		256x256		1609	1309	33.61	0.02	500.18	
	2		512x512		7914	29874.84	36.89	0.01	101.69	
	16		128x128		719	100.19	29.29	0.03	4509.86	
	8	2048x2048	3242595	256x256	1593	849.74	33.62	0.02	2035.52	

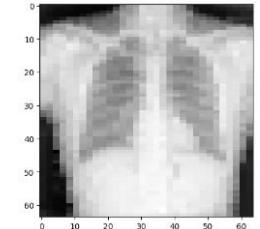
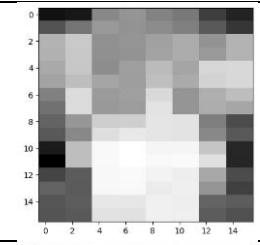
	4		512x512		4739	21705.21	35.32	0.01	684.23	
	16	2048x2048	128x128	3253651	762	80.22	29.01	0.03	4269.88	
	8		256x256		1715	1309.94	33.54	0.02	1897.17	
	4		512x512		5144	17659.36	24.93	0.05	632.51	

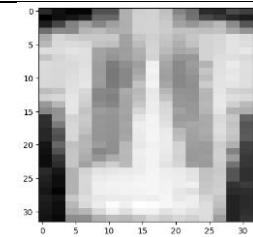
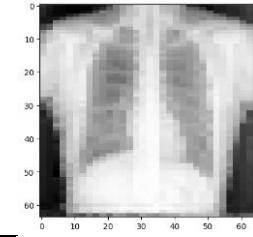
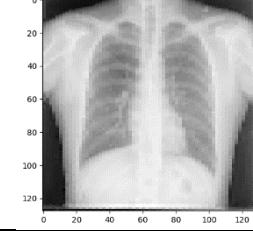
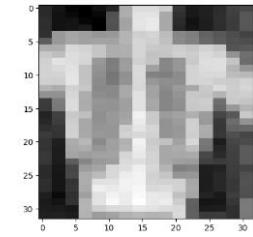
#### 4.2.2 Pengujian Kompresi Citra Medis Sinar-x Dada dengan Format JPG.

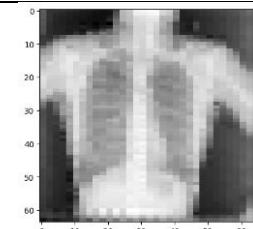
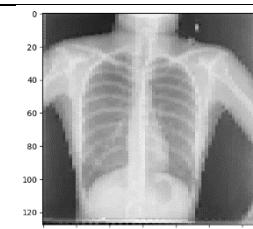
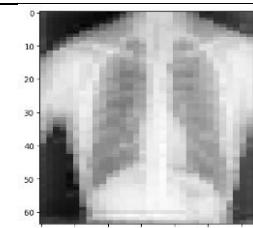
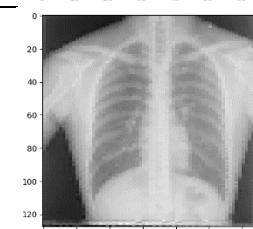
Citra medis sinar-x dada dikompres dengan menggunakan algoritma *Fraktal* dan *Arithmetic Coding*. Hasil uji coba kompresi dengan algoritma *Fraktal* dan *Arithmetic Coding* tercantum pada tabel 4.6.

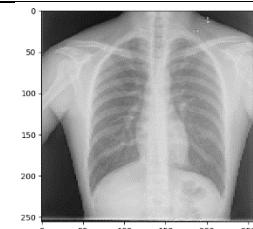
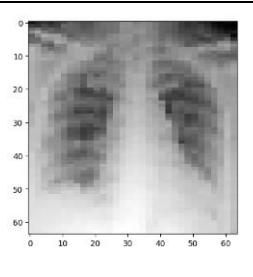
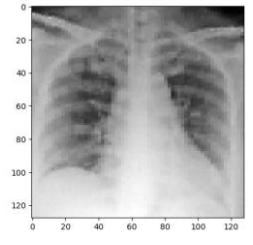
**Tabel 4. 6** Hasil Pengujian Citra dengan Ekstensi JPG

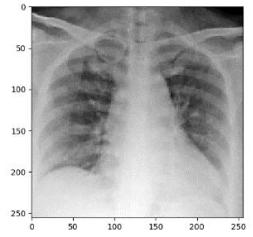
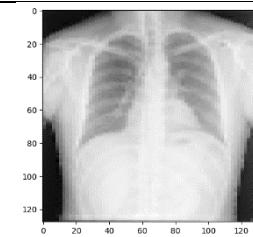
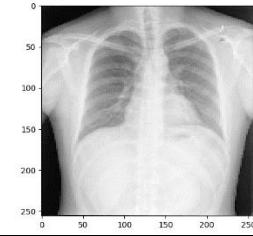
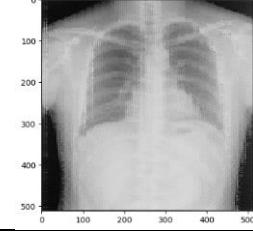
Citra Asli	Reduce Citra	Ukuran Citra (Piksel)		Ukuran Citra (Byte)		Waktu Kompresi (s)	PSNR (dB)	MSE	Ratio Kompresi	Citra Hasil
		Asli	Hasil	Asli	Hasil					
	8	128x128	16x16	6805	502	1.496	15.63	38.91	13.55	
	4		32x32		831	4.296	22.67	17.60	8.18	

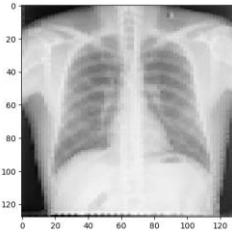
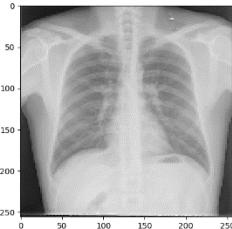
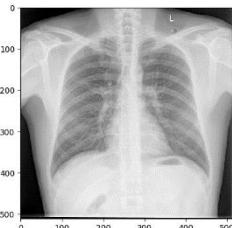
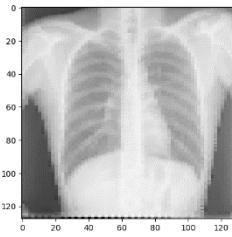
	2		64x64		2132	9.441	26.56	11.58	3.19	
	8	128x128	16x16	493	1.645	16.63	33.96	13.33		

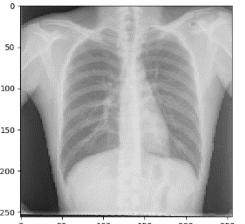
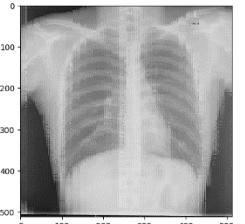
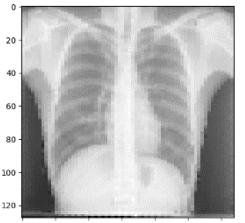
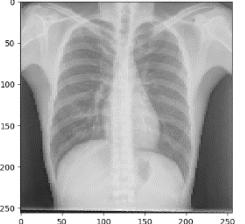
	8		32x32		867	2.115	22.17	18.36	22.14	
	4	256x256	64x64	19198	2162	8.879	26.61	11.12	8.87	
	2		128x128		6008	74.9	29.72	8.13	3.19	
	8	256x256	32x32	20142	925	2.236	21.94	19.01	21.77	

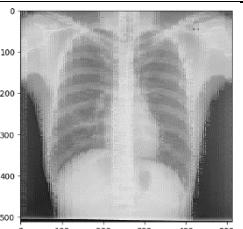
	4		64x64		2210	5.973	24.56	14.25	9.11	
	2		128x128		6420	78.62	28.80	8.91	3.13	
	8	512x512	64x64	60676	2135	7.317	25.96	11.92	28.41	
	4		128x128		6400	80.286	29.60	8.07	9.48	

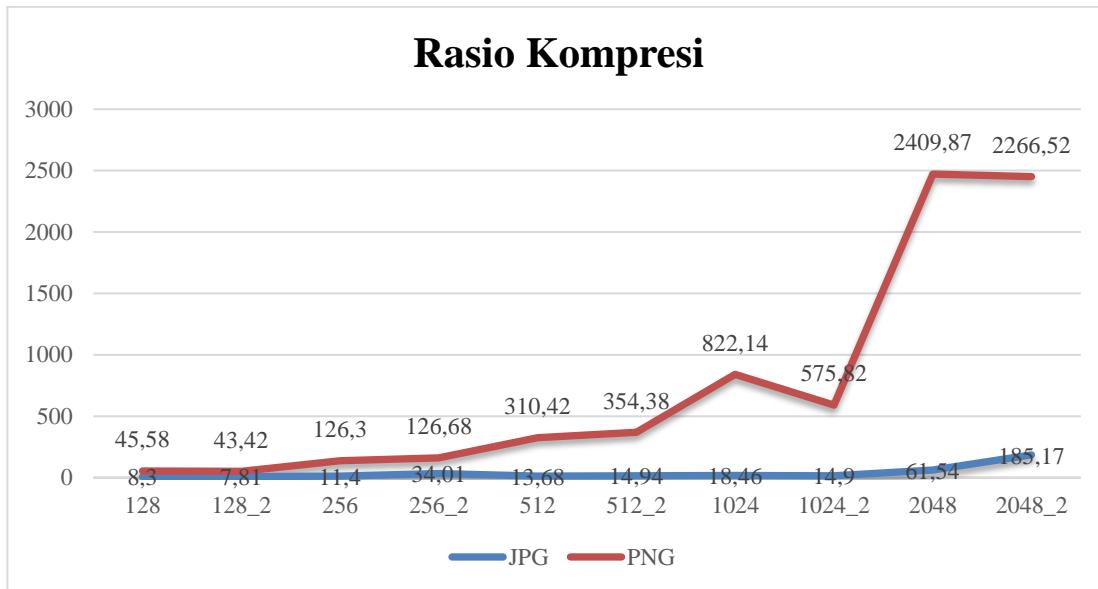
	2		256x256		19097	900.564	32.89	5.65	3.17	
	8	512x512	64x64	55972	1807	3.469	29.65	6.84	30.97	
	4		128x128		5216	79	33.19	4.58	10.73	

	2		256x256		17816	1283.527	35.61	3.49	3.14	
	8	1024x1024	128x128	220444	5524	84.18	30.85	7.08	39.90	
	4		256x256		16965	1664.42	34.95	4.55	12.99	
	2		512x512		88435	19305.98	25.01	2.30	2.49	

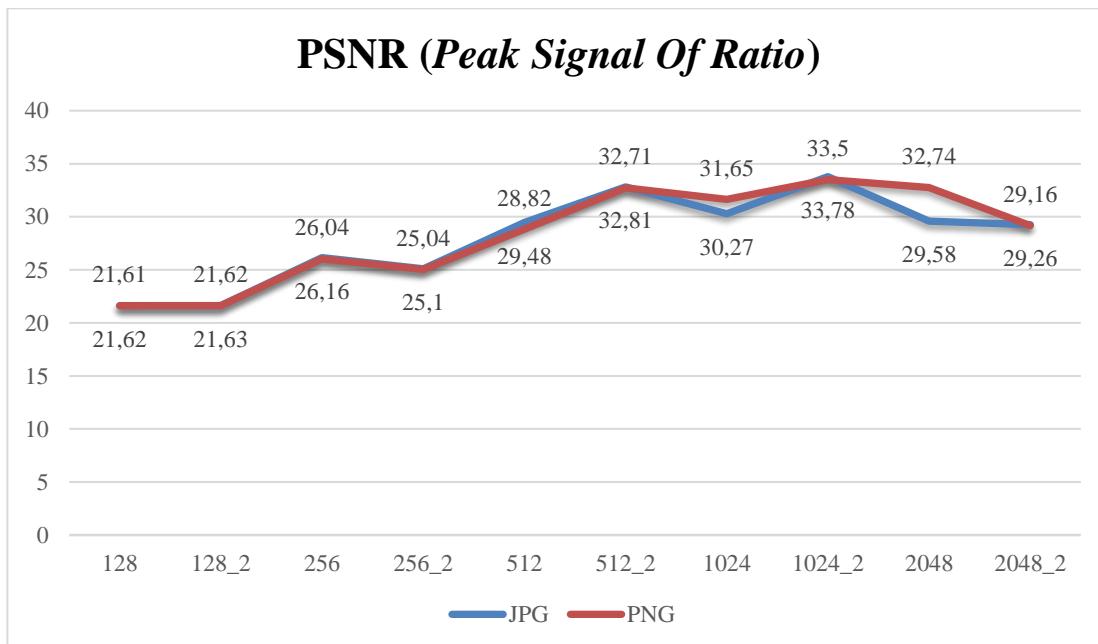
	8		128x128		6099	89.052	29.93	7.8	32.06	
	4	1024x1024	256x256	195547	18381	1143.267	33.93	5.12	10.63	
	2		512x512		97114	22750	37.5	2.08	2.01	
	16	2048x2048	128x128	814344	6218	52.44	29.51	8.26	130.96	

	8		256x256		18588	1499.85	33.88	5.07	43.81	
	4		512x512		82506	17148.39	25.35	13.75	9.87	
	16		128x128		6180	78.432	29.17	8.55	132.65	
	2048x2048	819805	256x256		18950	1357.99	33.69	5.27	43.26	

	4		512x512		88502	23.336,2	24.94	14.43	9.26	
--	---	--	---------	--	-------	----------	-------	-------	------	---



**Gambar 4. 11** Grafik Rasio Kompresi



**Gambar 4. 12** Grafik Nilai PSNR

Pembahasan dari pengujian diatas adalah :

1. Berdasarkan dari hasil pengujian yang dilakukan menggunakan kombinasi dua algoritma kompresi *Fraktal* dan *Arithmetic Coding* menghasilkan ukuran citra yang bergantung pada nilai *reduce* citra, semakin besar nilai *reduce* citra, maka ukuran citra hasil akan semakin kecil, dan semakin kecil nilai *reduce* citra, maka ukuran citra hasil semakin besar.
2. Dari hasil pengujian yang dilakukan ditinjau dari segi waktu kompresi, citra berekstensi JPG dan PNG memiliki prinsip yang sama yaitu semakin besar ukuran citra, maka waktu kompresi semakin lama.
3. Jika dilihat dari nilai *mean square error* (MSE), citra berekstensi PNG cenderung memiliki nilai MSE yang sangat rendah karena citra yang berekstensi PNG jika dikompres akan mempertahankan kualitas aslinya. Semakin rendah nilai MSE, maka hasil kompresi semakin mendekati ke citra asli.
4. Semakin kecil rasio kompresi, maka hasil kompresi citra akan semakin mendekati citra aslinya dan sebaliknya, semakin besar rasio kompresi maka semakin menjauhi citra aslinya. Dalam penelitian ini, citra berekstensi PNG memiliki rasio kompresi lebih besar daripada rasio kompresi citra berekstensi JPG sehingga, citra hasil berekstensi PNG menjauhi citra aslinya namun lebih menghemat ruang penyimpanan.
5. Semakin tinggi nilai PSNR maka semakin mendekati citra aslinya, dan sebaliknya semakin kecil nilai PSNR maka semakin buruk kualitas hasil citranya. Berdasarkan dari hasil penelitian yang dilakukan, kualitas citra hasil kompresi (PSNR) berekstensi JPG dan PNG tidak ada perbedaan yang mencolok. Namun, untuk nilai PSNR tertinggi terdapat pada citra berekstensi JPG yaitu sebesar 33,78 dB pada citra berukuran 1024 x 1024 piksel yang dapat dilihat pada gambar 4.12.
6. Dari penelitian ini, dapat dilihat bahwa untuk kompresi citra dengan menggunakan kombinasi dua algoritma ini yaitu *Fraktal* dan *Arithmetic*

*Coding*, kurang cocok jika diterapkan pada citra yang berukuran besar. Karena dari segi waktu akan menjadi tidak efisien.

## **BAB 5**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan implementasi dan hasil uji kompresi citra medis sinar-x dada dengan menggunakan algoritma *Fraktal* dan *Arithmetic Coding*, dapat disimpulkan bahwa kombinasi algoritma kompresi *lossy* dan *lossless* yaitu *Fraktal* dan *Arithmetic Coding* diharapkan akan menghasilkan citra tanpa kehilangan informasi dan dapat meminimalkan ruang penyimpanan. Namun dari hasil pengujian yang telah dilakukan, kombinasi algoritma ini kurang cocok jika diterapkan pada citra yang berukuran besar, karena dari segi waktu akan menjadi tidak efisien. Untuk mengukur hasil kualitas citra menggunakan PSNR pada kedua ekstensi citra JPG dan PNG tidak ada perbedaan yang mencolok, nilai PSNR tertinggi terdapat pada citra berekstensi JPG yaitu sebesar 33,78 dB. Dari segi rasio kompresi, citra hasil berekstensi PNG lebih menghemat ruang penyimpanan daripada citra berekstensi JPG.

#### **5.2 Saran**

Adapun saran untuk penelitian ini adalah dilakukan penelitian lebih lanjut agar menemukan kombinasi yang cocok untuk masing-masing algoritma ini pada citra yang berukuran besar sehingga didapatkan hasil kompresi yang lebih baik dan agar waktu kompresi citra lebih efisien dapat menggunakan spesifikasi perangkat implementasi yang mumpuni untuk melakukan kompresi citra yang berukuran besar.

## DAFTAR PUSTAKA

- Hasanujjaman, et al. 2019. “Fractal Image Compression of an Atomic Image Using Quadtree Decomposition.” 2019 Devices for Integrated Circuit (DevIC), IEEE, 2019, pp. 501–04.
- Kabir, Md Ahsan., Mondal, M Rubaiyat Hussain. 2017. Edge-based Transformation and Entropy Coding for Lossless Image Compression. International Conference on Electrical, Computer and Communication Engineering (ECCE).
- Khairil Anwar, A. S. (2008, Agustus). Kompresi Citra Medis Menggunakan Discrete Wavelet Transform (Dwt) Dan Embedded Zerotreewavelet (Ezw). Jurnal Matematika, 11, 73-77.
- Lidya, Syahfitri Kartika. 2012. Implementasi Dan Analisis Kinerja Algoritma Arithmetic Coding Dan Shannon-Fano Pada Kompresi Citra Bmp. Skripsi. Fakultas Ilmu Komputer dan Teknologi Informasi USU. Medan.
- Liu, Shuai; Bai, Weiling; Zeng, Nianyin; Wang, Shuihua (2019). *A Fast Fractal Based Compression for MRI Images*. IEEE Access, 7(), 62412– 62420. doi:10.1109/ACCESS.2019.2916934
- Khalili, F., Celenk, K., and Akinlar, M. A. 2013. Medical Image Compression Using Quad-tree Fractals and Segmentation. *IOP Conference Series: Materials Science and Engineering*.
- Panjavamam, B; Bhuvaneswari, P. T. V. (2017). Suitability Analysis of Fractal Compression Technique for Medical Image. 2017 Ninth International Conference on Advanced Computing (ICoAC), (), 127– 133. doi:10.1109/ICoAC.2017.8441424
- Poolakkachalil, Thafseela Koya dan Chandran, Saravanan. 2018. Analysis of Stereoscopic Image Compression using Arithmetic Coding and Huffman Coding. International Conference on Inventive Research in Computing Applications (ICIRCA 2018).
- Pu, Ida Mengyi. 2006. *Fundamental Data Compression*. London: elsevier.

- Putra, D. 2010. *Pengolahan Citra Digital*. Yogyakarta: Penerbit ANDI.
- Salomon, D., & Motta. G. 2010. *Handbook of Data Compression*. Fifth Edition, Computer Science Department. California State University, Northridge, Springer-Verlag, London.
- Saputra, Aan Kurniawan dkk. 2015. Apliksai Kompresi File Citra Menggunakan Algoritma Arithmetic Coding Berbasis Java. SemantIK. Kendari.
- Siahaan, R. (2017). Perbandingan Kompleksitas Waktu Teoretis dan Real Time Algoritma Strand Sort, Sieve Sort, Gnome Sort. Tersedia dari Repository USU Database.
- Sismoro, Heri. 2005. *Logika Informatika, Algoritma dan Pemrograman Komputer*. Jogyakarta: Jogyakarta
- Soesanti, I. (2008). Kompresi Citra Medis Menggunakan Alihragam Cosinus Diskret. Jurnal Ilmiah Semesta Teknika, 11, 1-17.
- S. Madenda, Pengolahan Citra & Video Digital, Jakarta: Erlangga, 2015.
- T. SutoyT.Sutoyo, Edy Mulyanto, Vincent Suhartono, Oky Dwi Nurhayati, W. (2009). Teori Pengolahan Citra Digital. Buku Pengolahan Citra Digital. AND IYogyakarta.
- Tamin, Rizki Pradana., 2021. Foto Rontgen Ini yang Harus Anda Ketahui. Tersedia : <https://www.alodokter.com/foto-rontgen-ini-yang-harus-anda-ketahui>. [5 November 2022].
- Virgiawan, I. M., Widayantara, I. M., & Sari Hartati, R. (2019). Codec Citra Berbasis fractal Dan entropy coding. Majalah Ilmiah Teknologi Elektro, 18(3), 313. <https://doi.org/10.24843/mite.2019.v18i03.p03>
- Yousif, R., & Salman, N. (2021). Image compression based on arithmetic coding algorithm. *Iraqi Journal of Science*, 329–334. <https://doi.org/10.24996/ijss.2021.62.1.31>
- Untuk source code lengkap dapat dilihat pada link berikut  
<https://github.com/windyadinda/Kompresi-Citra-Medis-Menggunakan-Algoritma-Fraktal-dan-Arithmetic-Coding.git>