

APLIKASI DEKOMPOSISI SERTA REKOMBINASI NILAI PIKSEL UNTUK PENGACAKAN CITRA DIGITAL BERDASARKAN *CHAOTIC SEQUENCE*

Jeremia Siregar¹⁾, Dedi Cahyadi²⁾

Teknik Informatika, Institut Sains dan Teknologi TD. Pardede, Medan
Jl. DR. TD.Pardede No. 8, Medan 20153, Sumatera Utara, Indonesia

¹⁾jeremiasiregar@istp.ac.id

²⁾dedipangaribuan12@gmail.com

Abstrak

Citra rahasia banyak dikirimkan melalui *internet* dan jaringan *wireless*, yang menyebabkan adanya kemungkinan citra dapat diakses oleh pihak yang ilegal dan pada akhirnya mengakibatkan kerugian bagi pemilik citra. Untuk mengatasi masalah keamanan ini, maka citra dapat diacak sedemikian rupa, sehingga citra rahasia hanya dapat diakses oleh orang-orang tertentu. Penelitian mengacak citra dengan mengubah nilai piksel dari sistem bilangan desimal menjadi bilangan basis empat (kuartener), kemudian mengurai (dekomposisi) keempat bit kuartener dan melakukan pengacakan terhadap keempat posisi bit berdasarkan pada bilangan acak (*chaotic sequence*) yang dihasilkan oleh algoritma *logistic mapping*, kemudian bit hasil pengacakan digabungkan kembali (rekombinasi) untuk menghasilkan nilai piksel baru. Aplikasi dapat digunakan untuk memproteksi citra yang bersifat rahasia dengan mengacak dan merekonstruksi kembali citra melalui proses dekomposisi dan rekombinasi bit kuartener (bilangan basis 4) dari nilai piksel citra.

Kata kunci: pengacakan, citra, dekomposisi, *rekombinasi*, *chaotic sequence*

Abstract

Many secret images are sent via the internet and wireless networks, which causes the possibility of the image to be accessed by illegal parties and ultimately results in losses for the image owner. To overcome this security problem, the image can be scrambled in such a way that the secret image can only be accessed by certain people. Research randomizes the image by changing the pixel value from the decimal number system to a base four (quaternary) number, then decomposing the four quaternary bits and randomizing the four bit positions based on random numbers (*chaotic sequences*) generated by the *logistic mapping* algorithm, then The randomized bits are recombined (*recombined*) to produce new pixel values. Applications can be used to protect secret images by randomizing and reconstructing the image through the process of decomposition and recombination of quaternary bits (base 4 numbers) of the image pixel value.

Keywords: randomization, image, decomposition, recombination, *chaotic sequence*

1. Pendahuluan

Citra digital memiliki peranan yang penting dalam teknologi multimedia. Seiring dengan implementasi citra yang semakin luas, maka bertambah pula kebutuhan proteksi terhadap citra untuk menjaga kerahasiaan dari citra digital. Enkripsi citra telah memiliki banyak aplikasi dalam berbagai bidang, seperti komunikasi *internet*, sistem multimedia, citra medis, *tele-medicine* (diagnosa medis jarak jauh) dan komunikasi di bidang militer. Citra rahasia banyak dikirimkan melalui *internet* dan jaringan *wireless*, sehingga memperbesar peluang citra dapat diakses oleh pihak yang ilegal dan pada

akhirnya mengakibatkan kerugian bagi pemilik citra. Untuk mengatasi masalah keamanan ini, maka citra harus mendapatkan pengamanan sedemikian rupa, sehingga citra rahasia hanya dapat diakses oleh orang-orang tertentu.

Pengacakan (*scrambling*) adalah salah satu cara untuk mengamankan informasi di dalam citra digital. Terdapat beberapa teknik dalam pengacakan citra digital yang secara umum dibagi menjadi dua kategori. Kategori pertama adalah mengacak posisi piksel (*spatial domain*) yang dapat dilakukan dengan menggunakan *Arnold transformation*, *Baker transformation*, *Magic transformation*, *Hilbert curve*,

dan *Gray code transformation*. Kategori kedua adalah mengacak nilai piksel citra dengan menggunakan *pseudo random number generator*. Teknik ini mengacak nilai piksel dengan menggunakan operasi XOR antara nilai piksel dan hasil bilangan acak yang merupakan *output* dari *pseudo random number generator*.

Penelitian ini menerapkan pengacakan dengan dua teknik sekaligus, yaitu mengacak posisi piksel (*spatial position*) dan nilai piksel sekaligus. Penelitian menerapkan teknik yang baru, yaitu dengan melakukan dekomposisi terhadap posisi dan nilai piksel dari citra awal, dan melakukan rekombinasi kembali untuk mengembalikan citra awal. Proses pengacakan *chaotic sequence* berdasarkan pada *logistic mapping*. *Logistic mapping* merupakan penghasil bilangan acak yang sederhana dan banyak digunakan. *Logistic mapping* mampu menghasilkan deretan bilangan yang acak berdasarkan nilai kunci μ ($3.569945 < \mu < 4$) dan nilai awal x_0 ($0 < x_0 < 1$). Penelitian mengubah nilai piksel dari sistem bilangan desimal menjadi bilangan basis 4, atau disebut juga dengan *quaternary*. Pengacakan kemudian mengubah posisi dari keempat bilangan berdasarkan deretan bilangan acak (*chaos*) dari *logistic mapping*, sehingga pada akhirnya mengubah nilai piksel secara keseluruhan. Oleh karena pengacakan dilakukan pada 4 set bilangan (bilangan basis 4), maka nilai kunci yang dibutuhkan juga berjumlah 4 set, yaitu $\{\mu_1, x^1_0\}$, $\{\mu_2, x^2_0\}$, $\{\mu_3, x^3_0\}$ dan $\{\mu_4, x^4_0\}$. Algoritma dianggap cukup aman karena mampu melakukan pengacakan terhadap posisi spasial dan nilai piksel citra sekaligus (Wang, dkk, 2015).

Berdasarkan uraian di atas, maka dipilih penelitian dengan judul “**Aplikasi Dekomposisi serta Rekombinasi Nilai Piksel dalam Pengacakan Citra Digital Berdasarkan Chaotic Sequence**” sebagai tugas akhir.

2. Landasan Teori

2.1. Citra Digital

Citra adalah suatu gambaran atau kemiripan dari suatu objek. Citra analog tidak dapat direpresentasikan dalam komputer, sehingga tidak dapat diproses oleh komputer secara langsung. Agar dapat diproses oleh komputer, citra analog harus dikonversi menjadi citra digital (Andono, dkk, 2017).

Citra adalah sebuah representasi visual dari suatu objek, seseorang atau suatu lokasi peristiwa yang dihasilkan oleh sebuah peralatan seperti kaca, lensa atau kamera. Representasi visual berdimensi dua, meskipun representasi tersebut mewakili objek atau lokasi peristiwa tiga dimensi (Siahaan dan Sianipar, 2020).

Citra digital adalah citra yang dapat diolah oleh komputer dan merupakan citra yang dihasilkan oleh

peralatan digital. Citra digital dapat diproses oleh komputer karena terdapat sistem sampling dan kuantisasi di dalam peralatan digital (Andono, dkk, 2017).

Sistem *sampling* adalah sistem yang mengubah citra kontinu menjadi citra digital dengan cara membagi citra analog menjadi M baris dan N kolom, sehingga dihasilkan citra digital. Semakin besar nilai M dan N , maka semakin halus citra digital yang dihasilkan. Pertemuan antara baris dan kolom disebut dengan piksel. Sistem kuantisasi adalah sistem yang melakukan perubahan intensitas analog ke intensitas diskrit, sehingga dengan proses ini dimungkinkan untuk membuat gradasi warna sesuai dengan kebutuhan. Kedua sistem ini yang bertugas untuk memotong citra menjadi M baris dan N kolom (proses *sampling*), sekaligus menentukan besar intensitas yang terdapat di titik tersebut (proses kuantisasi), sehingga menghasilkan resolusi citra digital (Andono, dkk, 2017).

Sebuah citra digital dapat didefinisikan sebagai sebuah fungsi dua dimensi, $f(x,y)$ dimana x dan y adalah koordinat bidang (spasial), dan amplitudo dari setiap pasangan koordinat (x,y) dinamakan sebagai intensitas citra pada titik tersebut. Istilah level keabuan mengacu pada intensitas dari citra monokrom, sedangkan citra berwarna menggunakan sistem warna RGB yang memuat tiga citra komponen individual, yaitu *red*, *green* dan *blue*. Hasil dari proses *sampling* dan kuantisasi adalah matriks yang membuat nilai-nilai riil dan dianggap sebagai representasi dari citra digital (Sianipar, 2018).

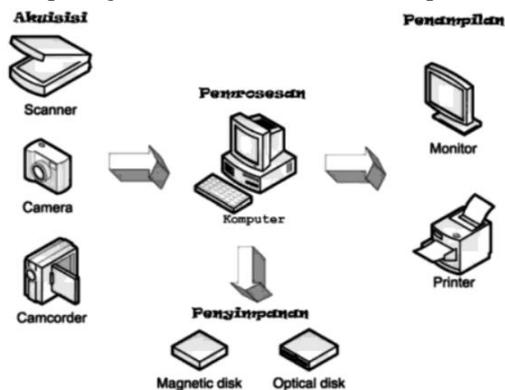
2.2. Pemrosesan Citra Digital

Pemrosesan citra adalah sebuah bidang multi-disipliner, dengan kontribusi dari sejumlah cabang ilmu khususnya matematika, fisika dan ilmu komputer dan teknik elektro. Pemrosesan citra digital juga merupakan bagian dari bidang penelitian pengenalan pola, pembelajaran mesin, kecerdasan buatan dan visi manusia (Siahaan dan Sianipar, 2020).

Pemrosesan citra digital dapat didefinisikan sebagai ilmu untuk memodifikasi citra digital menggunakan komputer digital. Terdapat tiga level dari operasi pemrosesan citra, yaitu (Siahaan dan Sianipar, 2020):

1. Level rendah. Operasi-operasi pada level rendah adalah penekanan derau, peningkatan kontras dan lainnya, dimana masukan dan keluaran adalah citra.
2. Level menengah. Operasi-operasi pada level menengah adalah ekstraksi atribut atau fitur (deteksi tepi, kontur, region) dari citra.
3. Level tinggi. Operasi-operasi pada level tinggi adalah analisis dan interpretasi dari isi citra.

Pemrosesan citra adalah sebuah bidang multi-disipliner, dengan kontribusi dari sejumlah cabang ilmu khususnya matematika, fisika dan ilmu komputer dan teknik elektro. Pemrosesan citra digital juga merupakan bagian dari bidang penelitian pengenalan pola, pembelajaran mesin, kecerdasan buatan dan visi manusia. Komponen-komponen yang terlibat dalam sistem pemrosesan citra digital dapat dilihat pada gambar 2.3 (Siahaan dan Sianipar, 2020).



Sumber: Andono, dkk, 2017

Gambar 1. Komponen-Komponen dalam Pemrosesan Citra Digital

2.3. Pengacakan

Teori *chaos* (pengacakan) sudah menjadi topik penelitian yang atraktif di dalam bidang keamanan informasi. Karakteristik yang menarik dari *chaos* adalah sensitifitasnya terhadap nilai awal (*initial value*). Jika nilai awal sistem *chaos* diubah sedikit saja, misalnya sebesar 10^{-10} , maka bila sistem *chaos* tersebut diiterasikan sejumlah kali, hasil iterasinya berbeda signifikan dengan sebelumnya. Sensitivitas ini sangat berguna di dalam kriptografi karena bersesuaian dengan prinsip *diffusion* dari Shannon dalam merancang sebuah algoritma kriptografi. Dengan prinsip *diffusion* ini, maka perubahan satu bit nilai awal chaos dapat menyebabkan cipherteks tetap tidak berhasil didekripsi (Munir, 2016).

Bilangan acak adalah deretan nilai yang acak dan tidak dapat diprediksi secara keseluruhan. Untuk menghasilkan bilangan acak merupakan hal yang sulit, kebanyakan pembangkit bilangan acak (*random bit / random number generator*) mempunyai beberapa bagian yang dapat diprediksi dan berhubungan. Kebanyakan RNG mengulang string yang sama setelah melakukan n putaran, sedangkan ada beberapa RNG lainnya menghasilkan nilai acak dengan berfokus pada suatu area tertentu dan mendistribusikannya secara seragam (Tjahjono, dkk, 2016).

Random number generator (RNG) adalah suatu peralatan komputasional yang dirancang untuk menghasilkan suatu urutan nilai yang tidak dapat ditebak polanya dengan mudah, sehingga urutan nilai

tersebut dapat dianggap sebagai suatu keadaan acak (random). RNG ini tidak dapat diterapkan dalam prakteknya. Bilangan acak yang dihasilkan oleh komputer sekalipun tidak benar-benar acak dan kebanyakan bilangan acak yang diterapkan dalam kriptografi juga tidak benar-benar acak, tetapi hanya berupa acak semu. Ini berarti bahwa bilangan acak yang dihasilkan itu dapat ditebak susunan atau urutan nilainya. Dalam kriptografi, bilangan acak sering dibangkitkan dengan menggunakan pembangkit bilangan acak semu (*pseudo random number generator*) (Tjahjono, dkk, 2016).

Random number generator merupakan pembangkit bilangan acak secara nulerik/aritmatika menggunakan komputer yang sering digunakan untuk proses perhitungan (Fauziah, dkk, 2016). Suatu *pseudo random number generator* (PRNG) merupakan suatu algoritma yang menghasilkan suatu urutan nilai dimana elemen-elemennya bergantung pada setiap nilai yang dihasilkan. *Output* dari PRNG tidak betul-betul acak, tetapi hanya mirip dengan properti dari nilai acak. Kebanyakan algoritma dari *pseudo random number generator* ditujukan untuk menghasilkan suatu sampel yang secara seragam terdistribusi. PRNG ini sering digunakan dalam kriptografi pada proses pembentukan kunci dari metoda kriptografi. Tingkat kerumitan dari PRNG ini menentukan tingkat keamanan dari metode kriptografi. Semakin rumit (kompleks) PRNG yang digunakan maka semakin tinggi tingkat keamanan dari metoda kriptografi. Pembangkit bilangan acak yang sering diimplementasikan adalah *Linier Congruential Generator* (LCG), *Linear Feedback Shift Register* (LFSR) dan *Logistic Map* (Tjahjono, dkk, 2016).

Beberapa teknik yang digunakan dalam pengacakan citra digital yang secara umum dibagi menjadi dua kategori. Kategori pertama adalah mengacak posisi piksel (*spatial domain*) yang dapat dilakukan dengan menggunakan *Arnold transformation* (Han dan Hu, 2006), *Baker transformation* (Li dan Xu, 2005), *Magic transformation* (Shen dkk, 2005), *Hilbert curve* (Lin dan Cai, 2004), dan *Gray code transformation* (Wang dan Yin, 2005). Pengacakan pada kategori pertama ini melakukan perpindahan posisi koordinat x (kolom) dan y (baris) dari piksel di dalam citra, sehingga citra menjadi teracak. Pengacakan ini mampu menghilangkan informasi dari bentuk citra, namun warna yang dimiliki oleh citra tidak mengalami perubahan karena piksel hanya dipindahkan dari satu posisi ke posisi yang lain di dalam citra.

Kategori kedua adalah mengacak nilai piksel citra dengan menggunakan *pseudo random number generator* (Wang, 2011). Teknik ini mengacak nilai piksel dengan menggunakan operasi XOR antara

nilai piksel dan hasil bilangan acak yang merupakan *output* dari *pseudo random number generator* (Ye, 2010). Pengacakan pada kategori kedua ini tidak melakukan perpindahan posisi piksel, namun melakukan pengacakan terhadap warna piksel, sehingga citra mengalami perubahan warna.

2.4. Pengacakan Citra Digital dengan Logistic Mapping

2.4.1. Logistic Mapping

Logistic mapping adalah sistem pengacakan yang sederhana dan digunakan secara luas. Ekspresi matematika dari logistic mapping dapat dilihat pada persamaan 2 (Wang, dkk, 2015).

$$x_{k+1} = f(x_k) = \mu x_k (1-x_k) \dots \dots \dots (1)$$

Variabel μ pada persamaan (1) merupakan konstanta, $x_k \in (0, 1)$ dan $k \in N$. *Logistic mapping* akan menghasilkan deretan bilangan acak saat μ berada di antara *range* nilai 3.56994 hingga nilai 4. Urutan pengacakan yang dihasilkan oleh pemetaan *Logistic mapping* dipengaruhi oleh nilai awal (Wang, dkk, 2015).

Sebuah citra dapat didefinisikan sebagai fungsi dua dimensi $f(x, y)$, dimana x dan y adalah koordinat spasial. Nilai f adalah intensitas nilai abu-abu piksel dari citra pada posisi spasial (x, y) . Intensitas nilai abu-abu piksel memiliki 256 skala, yaitu dari nilai 0 hingga 255. Pengacakan dapat dilakukan dengan mengurai (dekomposisi) dan menggabungkan kembali (rekombinasi) nilai piksel dari citra digital. Nilai piksel didekomposisi dalam urutan tempat ratusan, puluhan dan satuan. Kemudian, sebuah piksel baru direkombinasi dengan memilih satu piksel sembarang dari kelompok tempat ratusan, puluhan dan satuan digit (Wang, dkk, 2015).

Nilai abu-abu piksel baru tidak hanya berbeda dari nilai piksel asli, namun juga posisi tempat ratusan, puluhan dan satuannya berbeda. Dengan demikian, posisi spasial piksel juga teracak. Oleh karena itu, citra digital yang diacak dengan cara ini menjadi tidak teratur pada posisi piksel dan nilai piksel abu-abu, yang menunjukkan bahwa algoritma ini dapat mengacak posisi piksel dan nilai abu-abu secara bersamaan (Wang, dkk, 2015).

Namun apabila pengacakan ini dilakukan pada bilangan basis sepuluh, misalnya dua piksel bernilai masing-masing 205 dan 189, maka terdapat kemungkinan hasil transformasi menjadi dua piksel yang berbeda yaitu nilai 285 dan 109. Dalam hal ini, piksel dengan nilai 285 melampaui lingkup skala abu-abu citra digital. Untuk mengatasi masalah ini, nilai piksel pertama-tama dikonversi dari bilangan basis sepuluh (desimal) menjadi bilangan basis empat (kuartener). Angka maksimum yang dapat diwakili oleh angka kuarterner empat digit adalah 33334, yang persis sama dengan 255 ketika

dikonversi menjadi bilangan desimal. Oleh karena itu, jika bilangan desimal dikonversi menjadi kuarterner dan piksel tersebut didekomposisi dan direkombinasi kembali, maka nilai piksel akan selalu tetap dalam lingkup skala abu-abu gambar yang diwakili di komputer (Wang, dkk, 2015).

2.4.2. Algoritma Pengacakan dan Rekombinasi

Langkah pertama dari proses pengacakan citra digital adalah membaca nilai citra digital dan mengekstraksinya menjadi matriks abu-abu P , kemudian matriks dua dimensi P diubah menjadi matriks satu dimensi $p(t)$, dimana t merupakan jumlah piksel dari citra. Setiap piksel diubah menjadi bilangan basis empat (kuartener), dan setiap digit dari kiri ke kanan dimasukkan ke masing-masing satu matriks. Dengan demikian, akan diperoleh empat buah matriks, yaitu $r_1(t)$, $r_2(t)$, $r_3(t)$ dan $r_4(t)$ (Wang, dkk, 2015).

Nilai μ_1 dan x^1_0 merupakan kunci rahasia, dan barisan nilai acak dengan presisi ganda (dua bilangan di belakang koma), $\{x_1, x_2, x_3, \dots, x_i\}$ dihasilkan dengan algoritma *Logistic Mapping*. Kemudian bilangan acak disusun kembali dengan urutan menaik (*ascending*) dan sebuah deretan angka baru $\{x'_1, x'_2, x'_3, \dots, x'_i\}$ diperoleh. Posisi lama i yang dipindahkan ke posisi baru j akan membentuk matriks (i, j) dimana $\{(i, j) | 1 \leq i \leq t, 1 \leq j \leq t\}$. Matriks (i, j) diaplikasikan pada $r_1(t)$, atau dengan kata lain elemen ke- i akan ditempatkan pada elemen ke- j untuk menghasilkan matriks baru $r'_1(t)$. Proses ini diulangi untuk nilai kunci $\{\mu_2, x^2_0\}$, $\{\mu_3, x^3_0\}$ dan $\{\mu_4, x^4_0\}$, dan diperoleh matriks baru $r'_2(t)$, $r'_3(t)$ dan $r'_4(t)$. Kemudian, dari kiri ke kanan, elemen $r'_1(t)$ merupakan digit pertama, $r'_2(t)$, $r'_3(t)$ dan $r'_4(t)$ merupakan digit kedua, ketiga dan keempat dalam bilangan kuarterner. Bilangan kuarterner dikonversi kembali menjadi bilangan desimal dan membentuk matriks piksel baru yaitu P' . Dengan demikian, diperoleh sebuah citra digital yang teracak (Wang, dkk, 2015).

Algoritma pengacakan yang dilakukan terhadap citra digital dengan *Logistic Mapping* adalah sebagai berikut: (Wang, dkk, 2015)

1. Baca nilai piksel dari citra asli dan masukkan kunci enkripsi x_0 dan μ , dimana $x^i_0 \in (0, 1)$, dimana $\mu_1 \in (3.569945, 4)$, dan $i = 1, 2, 3, 4$.
2. Konversi nilai piksel dari desimal ke bilangan basis empat (kuartener) dan dekomposisi menjadi empat bagian.
3. Hasilkan bilangan acak dengan menggunakan *logistic mapping* berdasarkan pada nilai x_0 dan μ . Lakukan rekombinasi untuk menghasilkan piksel baru.

4. Konversi nilai piksel baru, dari bentuk kuartener ke bentuk desimal.
5. Proses enkripsi selesai dan tampilkan citra teracak.

Proses rekonstruksi merupakan kebalikan dari proses pengacakan. Citra asli hanya dapat dikembalikan ketika nilai kunci pengacakan dimasukkan dan proses rekonstruksi dilakukan (Wang, dkk, 2015).

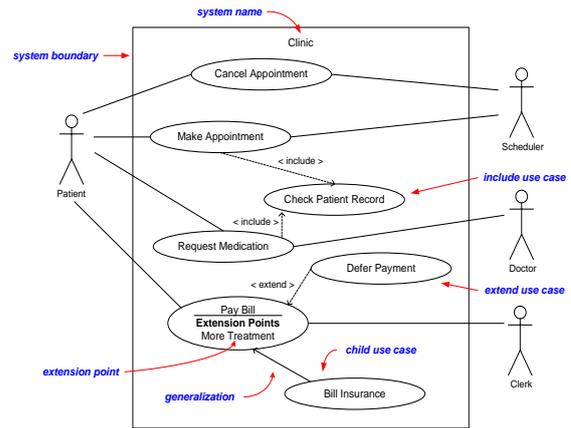
2.5. Unified Modeling Language (UML)

UML merupakan salah satu bentuk *language* atau bahasa yang didefinisikan sebagai bahasa *visual* untuk menjelaskan, memberikan spesifikasi, merancang, membuat model, dan mendokumentasikan aspek aspek dari sebuah sistem (Herpendi, 2016). UML adalah bahasa pemodelan untuk memodelkan sistem atau perangkat lunak yang berparadigma "berorientasi objek". Pemodelan (*modeling*) digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa, sehingga lebih mudah dipelajari dan dipahami. Adapun tujuan pemodelan adalah sebagai sarana analisis, pemahaman, visualisasi, dan komunikasi antar anggota tim pengembang, serta sebagai sarana dokumentasi, yang bermanfaat untuk menelaah perilaku perangkat lunak secara seksama serta bermanfaat untuk melakukan pengujian terhadap perangkat lunak yang telah selesai dikembangkan (Adi, 2015).

UML adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model atau proses dari sebuah sistem. UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka UML lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET, walaupun demikian UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C (Dharwiyanti dan Wahono, 2015).

2.5.1. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Use case adalah interaksi atau dialog antara sistem dan actor, termasuk pertukaran pesan dan tindakan yang dilakukan oleh sistem. Use case diagram dapat dilihat pada gambar 2.4.



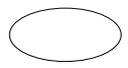
Sumber: Dharwiyanti dan Wahono, 2015

Gambar 2. Contoh Use Case Diagram

Simbol-simbol yang biasanya digunakan dalam suatu use case diagram dapat dilihat pada tabel 1.

Tabel 1. Simbol pada Use Case Diagram

No	Simbol	Nama	Keterangan
1.		Actor	Menspesifikasi kan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2.		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3.		Generalization	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).

4.		<i>Include</i>	Menspesifikasi kan bahwa <i>use case</i> sumber secara eksplisit.
5.		<i>Extend</i>	Menspesifikasi kan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7.		<i>System</i>	Menspesifikasi kan paket yang menampilkan sistem secara terbatas.
8.		<i>Use case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
9.		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10.		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan

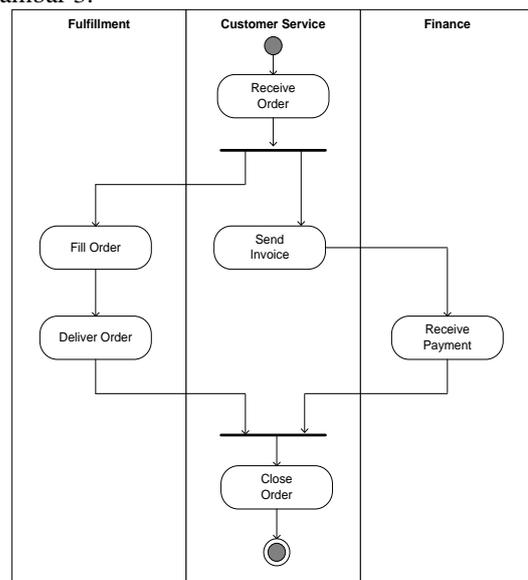
			mencerminkan suatu sumber daya komputasi.
--	--	--	---

Sumber: Herpendi, 2016

2.5.2. Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas (Dharwiyanti dan Wahono, 2015).

UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. *Activity diagram* dapat dilihat pada gambar 3.



Gambar 3. Contoh Activity Diagram

Sumber: Dharwiyanti dan Wahono, 2015

Simbol-simbol yang biasanya digunakan di dalam *activity diagram* dapat dilihat pada tabel 2.

Tabel 2. Simbol pada Activity Diagram

No.	Simbol	Nama	Keterangan
1.		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi

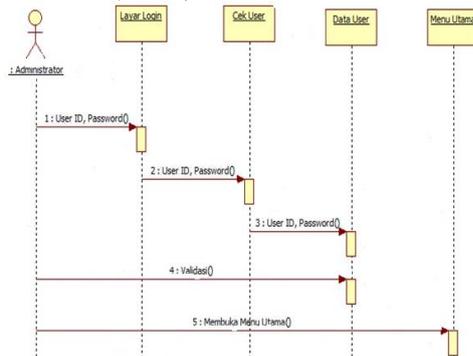
			satu sama lain.
2.	●	Initial Node	Bagaimana objek dibentuk atau diawali.
3.	⦿	Activity Final Node	Bagaimana objek diakhiri.
4.	—	Fork Node	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

Sumber: Herpendi, 2016

2.5.3. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. Sequence diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait) (Dharwiyanti dan Wahono, 2015).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu, diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Contoh sequence diagram dapat dilihat pada gambar 4 (Dharwiyanti dan Wahono, 2015).



Sumber: Dharwiyanti dan Wahono, 2015

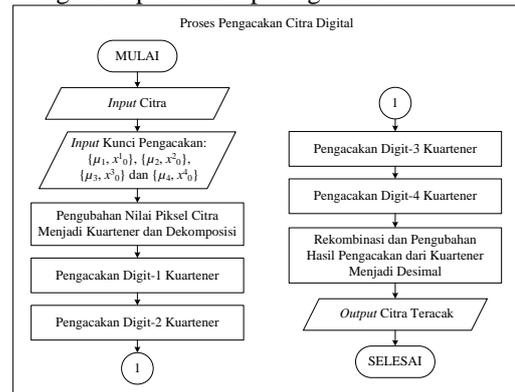
Gambar 4. Contoh Sequence Diagram

3. Hasil dan Analisa

3.1. Analisis Proses

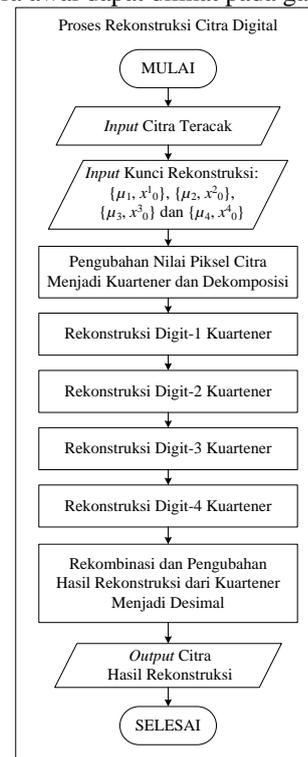
Proses utama di dalam penelitian terdiri dari dua proses, yaitu proses pengacakan citra digital dan proses rekonstruksi citra digital, melalui dekomposisi dan rekombinasi nilai piksel dengan menggunakan

logistic mapping. Flowchart dari proses pengacakan citra digital dapat dilihat pada gambar 5.



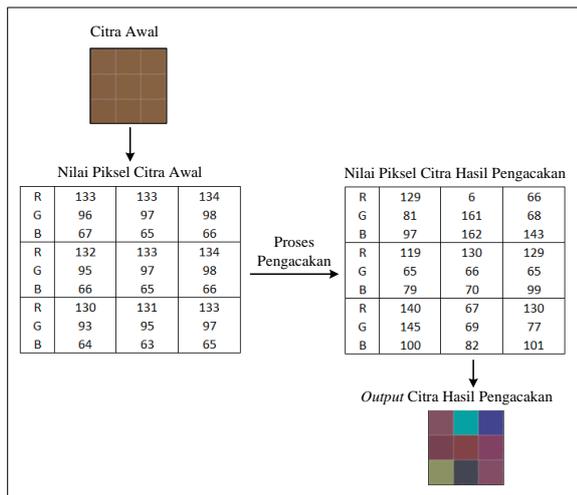
Gambar 5. Flowchart Proses Pengacakan Citra

Proses rekonstruksi citra digital merupakan kebalikan dari proses pengacakan. Proses rekonstruksi mengembalikan posisi dan nilai digit kuartener dari posisi pengacakan ke posisi awal, sehingga diperoleh citra awal, dengan syarat input kunci rekonstruksi harus sama dengan nilai kunci pengacakan. Flowchart dari proses rekonstruksi citra teracak menjadi citra awal dapat dilihat pada gambar 6.



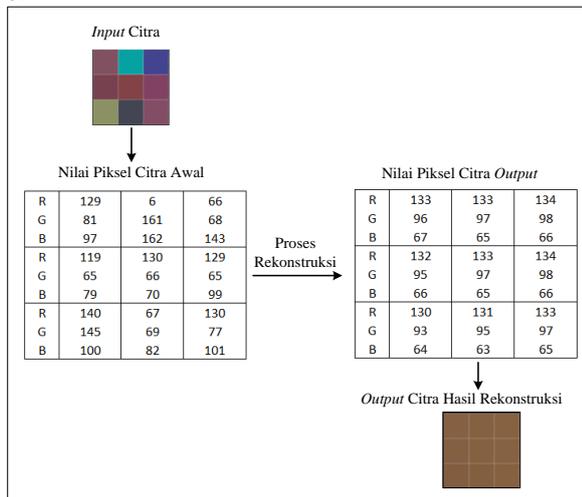
Gambar 6. Flowchart Proses Rekonstruksi Citra

Matriks hasil pengacakan $p'(t)$ disusun kembali menjadi nilai RGB dari piksel untuk menghasilkan output berupa citra teracak, seperti terlihat pada gambar 7.



Gambar 7. Output Citra Teracak

Matriks hasil rekonstruksi $p'(t)$ disusun kembali menjadi nilai RGB dari piksel untuk menghasilkan output berupa citra awal, seperti terlihat pada gambar 8.



Gambar 8. Output Citra Rekonstruksi

3.2. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan dengan memahami kebutuhan dari sistem dan mengembangkan sistem yang dapat memenuhi kebutuhan yang diharapkan sehingga diperoleh hasil akhir sesuai dengan yang dibutuhkan. Aplikasi dekomposisi serta rekombinasi nilai piksel dalam pengacakan citra digital berdasarkan *chaotic sequence* harus memenuhi kebutuhan fungsional sebagai berikut:

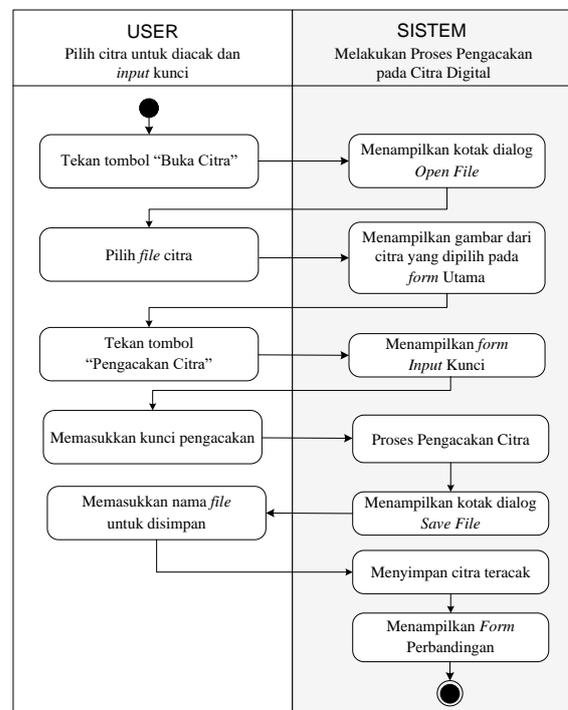
1. Aplikasi harus dapat memproses file citra dalam format bmp, jpg dan png.
2. Aplikasi harus memiliki empat set nilai kunci berupa, $\{\mu_1, x^1_0\}$, $\{\mu_2, x^2_0\}$, $\{\mu_3, x^3_0\}$ dan $\{\mu_4, x^4_0\}$, dengan presisi sebesar 10 digit desimal di belakang koma.

3. Aplikasi harus dapat melakukan pengacakan terhadap citra digital.
4. Aplikasi harus dapat mengembalikan citra teracak kembali ke citra awal melalui proses rekonstruksi bila nilai kunci yang digunakan sama dengan nilai kunci pengacakan.
5. Aplikasi harus dapat menampilkan proses perbandingan MSE antara citra sebelum dan sesudah proses.

3.3. Pemodelan Sistem

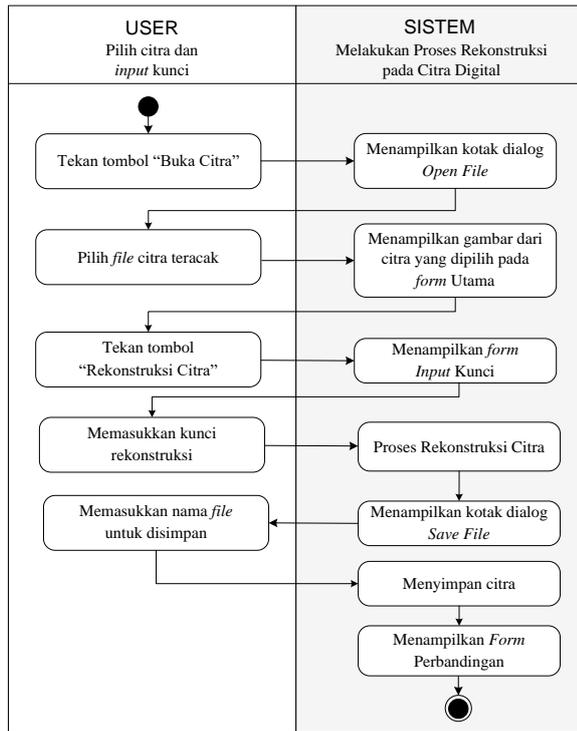
Pemodelan sistem dapat dilakukan dengan menggunakan *Unified Modeling Language (UML)*. Diagram *UML* menjelaskan mengenai tingkah laku sistem dan bukan bagaimana cara kerja algoritma.

Proses pengacakan citra yang terjadi di dalam aplikasi dapat digambarkan dengan *activity diagram* seperti terlihat pada gambar 9.



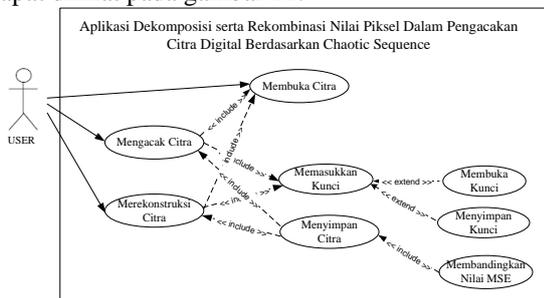
Gambar 9. Activity Diagram dari Proses Pengacakan Citra

Gambar 7 memperlihatkan diagram aktivitas bila user ingin melakukan proses pengacakan terhadap citra. Rekonstruksi merupakan kebalikan dari proses pengacakan. Proses rekonstruksi citra yang terjadi di dalam aplikasi dapat digambarkan dengan *activity diagram* seperti terlihat pada gambar 10.



Gambar 10. Activity Diagram dari Proses Rekonstruksi Citra

Use case merupakan salah satu diagram UML. Pemodelan sistem dengan menggunakan use case dapat dilihat pada gambar 11.



Gambar 11. Diagram Use Case dari Aplikasi

4. Kesimpulan

Setelah menyelesaikan perancangan aplikasi dekomposisi serta rekombinasi nilai piksel dalam pengacakan citra digital berdasarkan chaotic sequence, beberapa kesimpulan yang dapat ditarik adalah sebagai berikut:

1. Aplikasi dapat digunakan untuk memproteksi citra yang bersifat rahasia dengan mengacak dan merekonstruksi kembali citra melalui proses dekomposisi dan rekombinasi bit kuartener (bilangan basis 4) dari nilai piksel citra.
2. Aplikasi akan berhasil mengembalikan citra hasil pengacakan ke citra awal melalui proses rekonstruksi bila nilai kunci yang digunakan sama dengan nilai kunci pada proses

pengacakan. Hal ini dikarenakan logistic mapping yang digunakan sebagai chaotic sequence akan menghasilkan bilangan acak yang sama bila nilai kunci sama, sehingga proses rekonstruksi akan berhasil mengembalikan posisi bit kuartener ke posisi awalnya.

5. Ucapan Terima Kasih

Ucapan terima kasih kepada

1. Bapak Jeremia Siregar, S.Kom, M.Kom selaku dekan fakultas sekaligus dosen pembimbing penulis.
2. Bapak Ir. PHP Sibarani, M.Si selaku rektor
3. Bapak Drs. Swingly Purba, M.Sc selaku ketua jurusan Teknik Informatika
4. Seluruh Dosen dan Staff Fakultas teknologi industri.
5. Yang paling istimewa kepada kedua orang tua dan keluarga yang telah memberikan semangat.
6. semua teman-teman ISTP yang telah memberikan semangat.

Daftar Pustaka

- [1] Adi, N., 2015, **Rekayasa Perangkat Lunak Berorientasi Objek dengan Metode USDP**, Yogyakarta: Andi.
- [2] Andono, dkk, 2017, **Pengolahan Citra Digital**, Yogyakarta: Andi.
- [3] Das, D., dan Lanjewar, U.A., 2015, **Realistic Approach of Strange Number System from Unary to Decimal**, Nagpur, India: JMT Arts & JJP Science College.
- [4] Dharwiyanti dan Wahono, 2015, **Pengantar Unified Modeling Language (UML)**, Kuliah IlmuKomputer.com.
- [5] Fauziah, dkk, 2016, **Analisis Implementasi Random Number Generator (RNG) pada Simulasi Antrian Menggunakan Aplikasi Berbasis Framework**, Jakarta: Universitas Nasional.
- [6] Herpendi, 2016, **Aplikasi Pengelolaan Nilai Akademik Mahasiswa dan Daftar Peserta dan Nilai Akhir (DPNA)**, Pelaihari: Politeknik Negeri Tanah Laut.
- [7] Munir, R., 2016, **Algoritma Enkripsi Citra Berbasis Chaos dengan Penggabungan Teknik Permutasi dan Teknik Substitusi Menggunakan Arnold Cat Map dan Logistic Map**, Bandung: ITB.
- [8] Putra, D., 2015, **Pengolahan Citra Digital**, Yogyakarta: Andi.
- [9] Siahaan, V., dan Sianipar, R.H., 2020, **Pemrosesan Citra Digital dengan Matlab**, Balige: Balige Publishing.
- [10] Sianipar, R.H., 2018, **Dasar Pemrosesan Citra Digital**, Yogyakarta: Andi.

- [11] Sutoyo, dkk, 2015, **Teori Pengolahan Citra Digital**, Yogyakarta: Andi.
- [12] Tjahjono, dkk, 2016, **Implementasi Unique Code Nominal Transfer Menggunakan Metode Linear Congruential Generator untuk Order Deposit**, Pasuruan: Universitas Merdeka.
- [13] Wang, dkk, 2015, **Digital Image Scrambling Algorithm Based on Chaotic Sequence and Decomposition and Recombination of Pixel Values**, Guilin, China: Guilin University of Electronic Technology.