SKRIPSI

IMPLEMENTASI KRIPTOGRAFI DNA PADA PENGAMANAN CITRA CT SCAN PASIEN COVID-19

Disusun dan diajukan oleh

MUTHIA AMANAH ARUM

H071171303

PROGRAM STUDI SISTEM INFORMASI DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

2021

IMPLEMENTASI KRIPTOGRAFI DNA PADA PENGAMANAN CITRA CT SCAN PASIEN COVID-19

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin

MUTHIA AMANAH ARUM

H071171303

PROGRAM STUDI SISTEM INFORMASI DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

2021

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama	: Muthia Amanah Arum
NIM	: H071171303
Program Studi	: Sistem Informasi
Jenjang	: S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

IMPLEMENTASI KRIPTOGRAFI DNA PADA PENGAMANAN CITRA CT SCAN PASIEN COVID-19

adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan belum pernah dipublikasikan dalam bentuk apapun.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 2 Juni 2021

menyatakan,

Muthia Amanah Arum

NIM: H071171303

IMPLEMENTASI KRIPTOGRAFI DNA PADA PENGAMANAN CITRA CT SCAN PASIEN COVID-19

Disusun dan diajukan oleh

MUTHIA AMANAH ARUM

H071171303

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin pada tanggal 2 Juni 2021 dan dinyatakan telah memenuhi syarat kelulusan.

Menyetujui,

Pembimbing Utama	**Pembimbing Pertama**
Dr. Hendra, S.Si., M.Kom	**Andi Muhammad Anwar, S.Si., M.Si**
NIP: 19760102 200212 1 001	NIP: 19901228 201803 1 001

Ketua Program Studi,

Dr. Muhammad Hasbi, M.Sc.

NIP: 19630720 198903 1 003

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama	: Muthia Amanah Arum
NIM	: H071171303
Program Studi	: Sistem Informasi
Judul Skripsi	: Implementasi Kriptografi DNA pada Pengamanan Citra CT Scan Pasien COVID-19

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

DEWAN PENGUJI

		Tanda tangan
Ketua	: Dr. Hendra, S.Si., M.Kom	(.....)
Sekretaris	: Andi Muhammad Anwar, S.Si., M.Si.	(.....)
Anggota	: Dr.Eng. Armin Lawi, S.Si., M.Eng.	(.....)
Anggota	: Dr. Muhammad Hasbi, M.Sc.	(.....)

Ditetapkan di : Makassar

Tanggal : 2 Juni 2021

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN UNIVERSITAS HASANUDDIN DEPARTEMEN MATEMATIKA FAKULTAS MIPA

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT karena atas berkat dan rahmat-Nya penulis dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Komputer. Penulis menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini. Oleh karena itu, pada kesempatan ini, dengan segala kerendahan hati penulis menyampaikan terima kasih yang setulus-tulusnya kepada:

1. Orang tua penulis, ayahanda **Alauddin, S.Pd., M.Si.** dan ibunda **Rumhaedah Abidin, S.Pd.**, adik-adik penulis yaitu **Hidayat Dwi Putra** dan **Hanna Althafunnisa**, serta kucing-kucing penulis yaitu **Coto, Cece, Lili, Jojo, Lolo, Jeje, Mumu, Momo, Riri, Coko, Ciku, Boba, Kuku, dan Nana** yang senantiasa menyemangati penulis selama menyusun skripsi di rumah dan menyelipkan nama penulis dalam setiap doanya.
2. Rektor Universitas Hasanuddin, Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin, dan Ketua Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin beserta seluruh jajarannya.
3. Bapak **Dr. Hendra, S.Si., M.Kom.** dan bapak **Andi Muhammad Anwar, S.Si., M.Si.** atas kesediaannya menjadi pembimbing utama dan pertama penulis, serta membimbing penulis dalam proses penyusunan skripsi ini sampai selesai. Juga kepada bapak **Dr.Eng. Armin Lawi, S.Si., M.Eng.** dan bapak **Dr. Muhammad Hasbi, M.Sc.** atas waktunya dan kesediannya untuk menjadi penguji, serta memberikan kritik, saran, dan masukan kepada penulis dalam proses penyusunan skripsi ini.
4. Dosen Departemen Matematika, dan terkhusus kepada ibu dan bapak dosen Program Studi Sistem Informasi Fakultas MIPA Universitas Hasanuddin untuk semua ilmu yang telah diberikan kepada penulis selama menempuh pendidikan di jenjang strata 1.
5. Teman-teman TIIINNS, **Andi Nurul Azizah A. Jasruddin, S.S.**, **Rusmainnah, S.Si.**, **Sitti Nur Arfaitha Azis, A.Md.Gz.**, **Hastrie Ainun,**

S.H., **Husnul Khatimah Najamuddin, A.Md.MM.**, dan **Nurul Azizah, S.Ked.**, atas loyalitasnya sejak 2012 dan segala kerecehannya, dramanya, kebersamaannya baik dunia maupun akhirat, serta semangat dan doa yang diberikan kepada penulis.

6. Teman-teman YTS, **Andi Nur Wahyuningsih**, **Firda Irianti Arifin**, **Tria Fidyah Mandarmahesti**, dan **Andi Dea Ihdinasari**, atas loyalitasnya sejak 2015, *trigger* yang sangat berpengaruh, serta semangat dan doa yang diberikan kepada penulis.
7. Teman-teman Gepeng, **Nur Khairunisa**, **Eka Kurnia**, **Eka Fitriani**, **Nurfadila Firdani Salam**, **Fadhillah Putri Taha**, **Geby Nionsi**, **Siti Rabiatul Adawiyah**, **Mir Ataini Aprilia**, dan **Azzahra Mubarikah**, atas loyalitasnya sejak menjadi mahasiswa baru di Sistem Informasi, berbagi suka dan duka selama menjadi mahasiswi kuat dan tangguh Sistem Informasi yang telah bertahan hingga saat ini.
8. Teman-teman **Sistem Informasi 2017** atas dukungan dan kerjasamanya selama kurang lebih 4 tahun di Universitas Hasanuddin, terkhusus kepada **Muhammad Arizki** yang selalu ada, membantu, mendoakan, serta mendukung kapanpun dan dimanapun, **Farhan Ramdhani** dan **Fadhil Hidayat Amin** selaku sobat pencari cuan andalan, juga **Muh. Fiqih Hamda**, **Khawaritzmi Abdallah Ahmad**, **Ilmi Kalam**, **Muhammad Nur**, dan **Kennedy** sebagai kawan belajar terambis yang menjadikan dunia perkuliahan Sistem Informasi menjadi lebih mudah.
9. ***Google***, ***StackOverflow***, ***GeeksforGeeks***, dan website-website lainnya yang telah membantu penulis dalam membantu menyelesaikan skripsi ini.

dan kepada pihak-pihak yang tidak disebutkan, terima kasih atas bantuan, dukungan, dan doa yang dipanjatkan kepada penulis baik sebelum, selama, dan setelah proses penyusunan skripsi ini.

Makassar, 2 Juni 2021

Penulis

PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Hasanuddin, saya yang bertanda tangan di bawah ini:

Nama	: Muthia Amanah Arum
NIM	: H071171303
Program Studi	: Sistem Informasi
Departemen	: Matematika
Fakultas	: Matematika dan Ilmu Pengetahuan Alam
Jenis karya	: Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul:

Implementasi Kriptografi DNA pada Pengamanan Citra

CT Scan Pasien COVID-19

beserta perangkat yang ada (jika diperlukan). Terkait dengan hal di atas, maka pihak universitas berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar pada tanggal 2 Juni 2021

Yang menyatakan

(Muthia Amanah Arum)

ABSTRAK

Pandemi COVID-19 yang merebak ke seluruh wilayah di Indonesia mengakibatkan banyak pola hidup manusia menjadi berubah, salah satunya adalah perubahan aktivitas dari luring (luar jaringan) menjadi daring (dalam jaringan). Daring tentunya berkaitan erat dengan proses transaksi data, dimana pada saat daring aktivitas pengiriman data menjadi lebih tinggi yang mengakibatkan tingginya potensi pencurian dan berujung penyalahgunaan data oleh orang-orang yang tidak bertanggungjawab, salah satunya adalah penyalahgunaan data rahasia pasien COVID-19. Oleh sebab itu diperlukan pengamanan pada data tersebut, yang salah satu jenisnya adalah pengamanan pada citra CT scan paru-paru pasien COVID-19 dengan kriptografi DNA. Sebelum dilakukan proses pengamanan citra, terlebih dahulu dilakukan proses pembuatan kunci simetris *K*, kemudian dilanjutkan dengan proses pembuatan rangkaian OTP DNA, lalu dienkripsi sedemikian rupa sehingga membentuk citra baru yang secara visual sangat berbeda dengan citra aslinya. Pada penelitian ini, ditambahkan sebuah metode yang dinamakan modifikasi biner yang tujuannya untuk membentuk citra enkripsi berukuran $n \times n$ piksel. Dengan bahasa pemrograman Python, proses enkripsi dan dekripsi yang dilakukan pada citra menunjukkan bahwa kriptografi DNA bekerja dengan baik dilihat dari waktu komputasi yang relatif singkat, menggunakan memori dan CPU yang cenderung sedikit, serta kerusakan citra yang dihasilkan juga ternilai besarnya dengan menghitung tingkat kerusakan citra menggunakan RMSE (*root mean squared error*) dan SSIM (*structural similarity index measure*). Dari penelitian ini, diperoleh pula bahwa tingkat kerusakan citra dipengaruhi oleh nilai kunci simetris *K* yang dibuat, dimana semakin besar nilai *K* maka semakin kecil tingkat kerusakan citranya.

Kata kunci: *COVID-19, citra digital, enkripsi, kriptografi DNA*.

ABSTRACT

The COVID-19 pandemic that spread throughout Indonesia has caused many people's lifestyle to change, one of which is the change of activity from offline to online. Online condition is certainly closely related to the process of data transaction, where at the time of online data transaction activity becomes higher which results in a high potential for theft and leads to misuse of data by irresponsible people, one of which is the misuse of confidential data of COVID-19 patients. Therefore, security is needed on the data, which one of the types is the security on CT scan image of the lungs of COVID-19 patients with DNA cryptography. Before the process of securing image, first done the process of creating symmetric key *K*, then continued with the process of making OTP DNA sequence, then encrypted in such a way that it forms a new image that is visually very different from the original image. In this study, a method called binary modification was added that was intended to form an $n \times n$ pixel-sized encryption image. With Python programming language, the encryption and decryption processes performed on image show that DNA cryptography works well judging by the relatively short compute time, using less memory and CPU, and the resulting image error is also invaluable by calculating the rate of image error using RMSE (root mean squared error) and SSIM (structural similarity index measure). From this study, it was also obtained that the level of image error is influenced by the symmetrical key value *K* made, where the greater the value of *K*, the smaller the level of image error.

Keywords: *COVID-19, digital image, encryption, DNA cryptography.*

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1. Latar Belakang

Pandemi COVID-19 yang akhir-akhir ini merebak ke seluruh wilayah Indonesia mengakibatkan setiap aktivitas yang dilakukan mau tidak mau dipindahkan ke rumah. Bagaimana tidak, dengan tingginya tingkat penyebaran ditambah dengan tingkat kematian yang tak kunjung landai membuat khawatir setiap orang yang memiliki aktivitas di luar. Pekerjaan yang tadinya bisa dilakukan secara luring, tetapi akibat pandemi sebagian besar dilakukan secara daring.

Berbicara tentang daring, tentunya kita tidak lepas dengan transaksi data. Transaksi data yang dimaksud adalah saling bertukar (mengirim dan menerima) data. Bukan hanya itu, karena aktivitas pengiriman data yang sebagian besar dilakukan secara daring, mengakibatkan tingginya potensi pencurian data oleh orang-orang yang tidak bertanggungjawab. Data yang dikirim juga tentunya bukan sembarang data, apalagi jika menyangkut data rahasia seperti data pasien COVID-19.

Seperti yang kita ketahui, karena pandemi COVID-19 muncul pula stigma sosial terkait COVID-19, sehingga para pasien COVID-19 diberikan label, stereotip, didiskriminasi, diperlakukan berbeda, dan/atau mengalami pelecehan status karena terasosiasi dengan sebuah penyakit, yang dalam hal ini COVID-19 (Tim Administrator Situs Kawal COVID-19, 2020).

Untuk itu, penulis merasa perlunya pengamanan pada data-data pasien COVID-19, yang salah satu jenisnya adalah citra CT *scan* paru-paru. Pengamanan yang akan dilakukan adalah pengamanan citra yang dalam hal ini adalah citra CT *scan* pasien COVID-19 dengan kriptografi DNA. Kriptografi DNA dipilih sebagai algoritma pengamanan pada penelitian ini karena sampai sejauh ini kriptografi DNA dikenal dengan penggunaan memorinya yang kecil dan waktu komputasinya yang relatif singkat. Tujuan pengamanannya tidak lain agar data yang dikirimkan bisa diamankan dari pelaku-pelaku kriminal yang berpotensi menyalahgunakan data tersebut, serta diharapkan dapat menciptakan rasa aman kepada pasien

COVID-19 agar tidak larut dalam stigma sosial. Penelitian yang telah dijelaskan penulis sebelumnya akan ditulis pada skripsi berjudul "Implementasi Kriptografi DNA pada Pengamanan Citra CT *Scan* Pasien COVID-19".

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan beberapa masalah sebagai berikut:

1) Bagaimana mengimplementasikan algoritma kriptografi DNA pada citra CT *scan* pasien COVID-19?
2) Bagaimana analisis performa dari algoritma kriptografi DNA dalam mengamankan citra CT *scan* pasien COVID-19?

1.3. Tujuan Penelitian

Dengan memperhatikan latar belakang dan rumusan masalah di atas, maka tujuan penelitian ini adalah:

1) Mampu mengimplementasikan algoritma kriptografi DNA, baik enkripsi maupun dekripsi pada citra CT *scan* pasien COVID-19.
2) Mampu menganalisis performa algoritma kriptografi DNA dalam mengamankan citra CT *scan* pasien COVID-19.

1.4. Manfaat Penelitian

Penelitian ini diharapkan dapat digunakan dalam proses pengamanan data rekam medik pasien COVID-19 yang dalam hal ini citra CT *scan* paru-paru pasien COVID-19, serta diharapkan ke depannya bisa digunakan untuk pengamanan data rekam medik pasien COVID-19 yang lebih lanjut, baik itu berupa data citra maupun data teks. Juga apabila memungkinkan, diharapkan dapat digunakan untuk mengamankan data-data lain yang tentunya tidak kalah penting, sehingga penelitian ini tidak terbatas terhadap data di bidang kesehatan, juga mencakup aspek kehidupan lainnya seperti di bidang pendidikan, pemerintahan, keuangan, sosial budaya, dan aspek-aspek kehidupan lainnya.

1.5. Batasan Masalah

Adapun batasan masalah yang diteliti untuk mencegah pembahasan yang terlalu luas, yaitu:

1) Citra yang diamankan pada penelitian ini adalah citra CT *scan* paru-paru pasien COVID-19 dengan format citra .tiff.
2) Aturan pasangan basa yang digunakan adalah aturan 1 yang bisa dilihat pada tabel 2.1 di bab 2.

BAB II
TINJAUAN PUSTAKA

2.1. COVID-19

Penyakit epidemi yang disebabkan oleh sebuah *novel coronavirus* bernama *Severe Acute Respiratory Syndrome Coronavirus 2* (SARS-CoV-2) atau (2019-nCoV) disebut penyakit *coronavirus disease-19* (COVID-19). Pada Desember 2019, virus corona dengan asal yang tidak diketahui tersebar di provinsi Hubei, Cina. Kehadiran COVID-19 dimanifestasikan oleh beberapa gejala, mulai dari gejala asimtotatik/ringan hingga penyakit parah dan kematian. Infeksi virus meluas secara internasional dan WHO mengumumkan Darurat Kesehatan Masyarakat yang Menjadi Perhatian Internasional (Esakandari et al., 2020).

Gejala COVID-19 yang paling umum adalah demam, batuk kering, dan kelelahan. Gejala lain yang kurang umum dan dapat mempengaruhi beberapa pasien termasuk kehilangan rasa atau bau, hidung tersumbat, konjungtivitis (juga dikenal sebagai mata merah), sakit tenggorokan, sakit kepala, nyeri otot atau sendi, berbagai jenis ruam kulit, mual atau muntah, diare, serta menggigil atau pusing. Gejala tersebut biasanya ringan, namun beberapa orang menjadi terinfeksi tetapi hanya memiliki gejala yang sangat ringan atau bahkan tidak sama sekali. Gejala penyakit COVID-19 yang parah meliputi sesak napas, kehilangan selera makan, kebingungan, nyeri atau tekanan yang terus menerus di dada, dan temperatur tinggi (di atas 38° C) (World Health Organization, 2020).

Perkembangan terakhir mengenai data sebaran kasus khususnya di Indonesia sendiri, jumlah kasus terkonfirmasi sudah mencapai angka sekitar 836 ribu jiwa, jumlah kasus sembuh sekitar 688 ribu jiwa, serta jumlah kasus meninggal sekitar 24 ribu jiwa. Data ini terhitung sejak awal munculnya COVID-19 di Indonesia hingga tanggal 11 Januari 2021 yang diperoleh dari beberapa sumber yang akurat seperti situs Satuan Tugas Penanganan COVID-19 dan Situs Kawal COVID-19.

2.2. Citra Digital

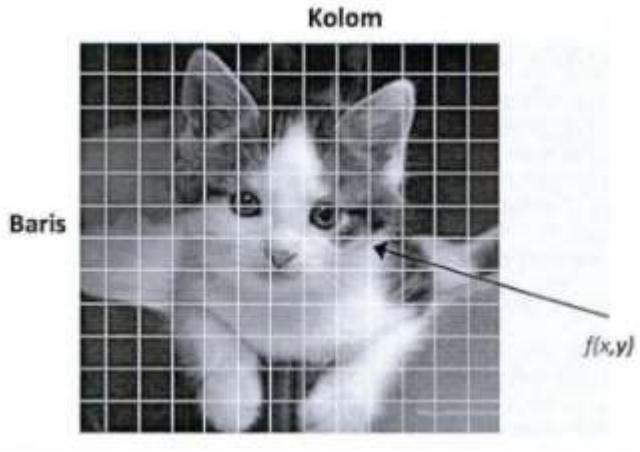
Citra adalah bagian yang penting dan mendasar dari kehidupan sehari-hari. Atas dasar individu, atau orang-ke-orang, gambar digunakan untuk menalar,

menafsirkan, mengilustrasikan, mewakili, menghafal, mendidik, berkomunikasi, mengevaluasi, menavigasi, survei, menghibur, dan lain-lain (Awcock & Thomas, 1995).

Citra digital dapat didefinisikan sebagai fungsi *f*(*x*, *y*) berukuran M baris dan N kolom, dengan *x* dan *y* adalah koordinat spasial, dan amplitudo *f* di titik koordinat (*x*, *y*) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut (Putra, 2010). Citra digital dapat ditulis dalam bentuk matriks sebagai berikut.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}$$

Nilai pada suatu irisan antara baris dan kolom (pada posisi *x*, *y*) disebut dengan *pixel*. Ilustrasi digitalisasi citra dapat dilihat pada gambar 2.1.



Gambar 2.1. Ilustrasi digitalisasi citra

2.3. CT *Scan*

Tomografi adalah perekaman bayangan bagian dalam tubuh pada bidang yang telah ditentukan sebelumnya dengan menggunakan tomograf (Dorland, 2010). Istilah *computer tomography*, atau CT, mengacu pada prosedur pencitraan sinar-X terkomputerisasi di mana berkas sinar-X yang sempit ditujukan ke pasien dan dengan cepat diputar di sekitar tubuh, menghasilkan sinyal yang diproses oleh komputer mesin untuk menghasilkan gambar penampang atau "irisan" dari tubuh.

Irisan ini disebut gambar tomografi dan berisi informasi yang lebih rinci daripada rontgen konvensional.

Tidak seperti sinar-X konvensional yang menggunakan tabung sinar-X tetap, CT *scanner* menggunakan sumber sinar-X bermotor yang berputar di sekitar bukaan melingkar dari sebuah donat struktur yang disebut *gantry*. Selama CT *scan*, pasien berbaring di tempat tidur yang bergerak perlahan melalui *gantry*, sementara tabung sinar-X berputar di sekitar pasien, menembakkan sinar-X ke seluruh tubuh. CT *scanner* menggunakan detektor sinar-X digital khusus, yang terletak tepat di seberang sumber sinar-X. Saat sinar-X meninggalkan pasien, sinar-X diambil oleh detektor dan dikirim ke komputer. Setiap kali sumber sinar-X menyelesaikan satu putaran penuh, komputer CT menggunakan teknik matematika yang canggih untuk membuat potongan gambar 2D pasien. Ketebalan jaringan yang direpresentasikan di setiap potongan gambar bisa berbeda-beda tergantung mesin CT yang digunakan, tapi biasanya berkisar 1-10 milimeter. Saat potongan penuh selesai, gambar disimpan dan tempat tidur bermotor dipindahkan ke depan secara bertahap ke dalam *gantry*. Proses pemindaian sinar-X kemudian diulang untuk menghasilkan potongan gambar lain. Proses ini berlanjut hingga jumlah irisan yang diinginkan terkumpul. Irisan gambar dapat ditampilkan satu per satu atau ditumpuk bersama oleh komputer untuk menghasilkan gambar 3D dari pasien yang menunjukkan kerangka, organ, dan jaringan serta kelainan apa pun yang coba diidentifikasi oleh dokter. Metode ini memiliki banyak keuntungan termasuk kemampuan untuk memutar gambar 3D dalam ruang atau untuk melihat irisan secara berurutan, sehingga lebih mudah untuk menemukan tempat yang tepat untuk menemukan masalah (*National Institute of Biomedical Imaging*, 2019).

2.4. Kriptografi

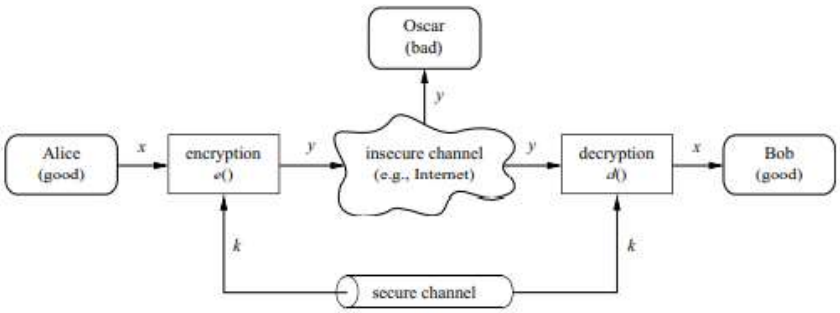
Kriptologi terbagi menjadi dua, yaitu “kriptografi” dan “kriptanalisis”. Kriptografi adalah ilmu menulis rahasia dengan tujuan menyembunyikan makna sebuah pesan. Sedangkan kriptanalisis adalah ilmu (terkadang disebut seni)

"merusak" sistem kripto. Tujuan kriptanalisis adalah untuk menguji keamanan dari metode kripto (Paar & Pelzl, 2009).

Kriptografi adalah ilmu yang mempelajari teknik matematis yang berkaitan dengan aspek keamanan informasi seperti kerahasiaan (*confidentiality*), integritas data (*data integrity*), otentikasi entitas (*entity authentication*), dan otentikasi asal data (*data origin authentication*) (Menezes et al., 1996). Kriptografi sendiri terdiri dari tiga cabang utama, yaitu algoritma kriptografi simetris, asimestris, dan protokol kriptografi.

2.4.1. Algoritma Kriptografi Simetris

Algoritma kriptografi simetris, yang banyak orang mengiranya sebagai konsep kriptografi yang sesungguhnya, dimana terdapat dua pihak memiliki metode enkripsi dan dekripsi yang masing-masing saling berbagi kunci rahasianya. Semua kriptografi dari zaman kuno hingga tahun 1976 secara ekslusif berdasarkan metode kriptografi simetris. *Cipher* simetris masih digunakan secara luas, khususnya untuk enkripsi data dan pemeriksaan integritas pesan.



Gambar 2.2. Sistem kripto kunci simetris

Variabel *x*, *y*, dan *k* pada gambar 2.2 adalah variabel-variabel yang penting dan memiliki nama tersendiri, yaitu:

- *x* dinamakan *plaintext* atau *cleartext*,
- *y* dinamakan *ciphertext*,
- *k* dinamakan *key* (kunci),
- himpunan dari kunci-kunci yang dapat digunakan dinamakan *keyspace*.

Kemudian ada analogi sederhana untuk menggambarkan kunci dari kriptografi simetris. Anggaplah ada sebuah brankas dengan kunci yang kuat. Hanya

Alice dan Bob yang memiliki duplikat dari kunci tersebut. Proses enkripsi pesan dapat dianalogikan sebagai menyimpan pesan di brankas, dan untuk mendekripsi pesan, yang dianalogikan sebagai membaca pesan, Bob menggunakan kuncinya dan membuka brankas, yang untuk lebih jelasnya dapat dilihat pada gambar 2.3.

Gambar 2.3. Analogi dari enkripsi simetris: brankas dengan satu kunci

2.4.2. Algoritma Kriptografi Asimetris

Algoritma kriptografi asimetris, yaitu konsep kriptografi dimana dua pihak masing-masing memiliki kunci rahasia seperti pada kriptografi simetris dan juga kunci publik. Pada tahun 1976 jenis *cipher* yang berbeda diperkenalkan oleh Whitfield Diffie, Martin Hellman, dan Ralph Merkle. Algoritma kriptografi asimetris dapat digunakan untuk aplikasi seperti tanda tangan digital dan pembentukan kunci, dan juga untuk enkripsi data klasik.

Gambar 2.4. Analogi dari enkripsi kunci publik: brankas dengan kunci publik untuk menyimpan pesan dan kunci rahasia untuk mengambil pesan

Pada gambar 2.4 dijelaskan analogi kriptografi asimetris yang memiliki prinsip kerja mirip seperti mengirimkan pesan ke kotak pos, dimana setiap orang dapat mengirim surat, seperti mengenkripsi, namun hanya orang dengan kunci pribadi (rahasia) yang dapat mengambil surat tersebut, seperti mendekripsi. Dari analogi ini juga dapat diketahui bahwa kunci yang dimiliki tiap orang terdiri dari dua bagian, yaitu kunci publik dan kunci rahasia.

2.4.3. Protokol Kriptografi

Protokol kriptografi, yang kurang lebih berhubungan dengan penerapan algoritma kriptografi, dimana algoritma kriptografi simetris dan asimetris dianggap sebagai blok bangunan yang tujuannya untuk mengamankan aplikasi seperti pada pengamanan komunikasi internet. Skema *Transport Layer Security* (TLS), yang digunakan pada setiap *web browser*, adalah contoh dari protokol kriptografi.

2.5. One-Time Pad

Pembuatan kunci acak berdasarkan *One-Time Pad* pertama kali diperkenalkan oleh Vernam dan dikenal dengan nama sandi Vernam. Kemudian diperluas oleh teori Shannon dengan menunjukkan bahwa OTP memiliki kerahasiaan yang sempurna (Kaundal, 2014).

Berdasarkan teori Shannon:

- Ukuran kunci harus setidaknya sama dengan panjang *plaintext*
- Harus benar-benar acak
- Tidak untuk digunakan kembali
- Harus terjaga kerahasiaannya.

One-Time Pad menghasilkan kunci acak yang tidak berhubungan secara statistik dengan *plaintext*. Ukuran dari kunci tersebut harus lebih besar atau sama dengan panjang dari *plaintext*. Keamanan dari *One-Time Pad* sepenuhnya bergantung pada kunci yang acak. Jadi, tidak ada pola atau keteraturan yang dapat digunakan oleh kriptanalisis untuk menyerang *ciphertext*. Teknik OTP dianggap mampu untuk membuat penyusup kesulitan dalam menebak kunci yang tepat (Stallings, 2017).

Pada penelitian ini, pembentukan kunci acak dilakukan dengan menggunakan *pseudo-random generator* yang dibuat dengan menggunakan *library* Python yaitu *random* dan menggunakan fungsi random.choice() untuk mendapatkan rangkaian OTP DNA yang benar-benar acak dan tidak berulang penggunaannya. Pengirim dan penerima, setelah menggunakan suatu rangkaian OTP DNA untuk satu proses (enkripsi dan dekripsi), harus menghancurkan kunci tersebut, sehingga membuat penyusup tidak mengetahui urutan yang benar.

2.6. Kriptografi DNA

Menurut Adleman (1994) pada *Molecular computation of solution to combinatorial problems*, kriptografi DNA adalah bidang baru berdasarkan penelitian pada komputasi DNA dan teknologi baru, seperti: PCR (*Polymerase Chain Reaction*), *Microarray*, dan sebagainya (Terec et al., 2011). Kriptografi DNA melibatkan penyandian teks biasa menggunakan teknik komputasi DNA (Omer & Farooq, 2015).

Manfaat dari kriptografi DNA diperoleh dari riset terhadap komputasi DNA, tetapi komputasi DNA tidak sama dengan kriptografi DNA, serta terdapat perbedaan yang mendasar diantara keduanya. Dalam komputasi DNA, teknologi DNA digunakan untuk memecahkan masalah komputasi yang sulit, sedangkan dalam kriptografi DNA, berbagai masalah biologis yang sulit digunakan sebagai dasar keamanan kriptografi DNA. Proses kriptografi DNA dapat dianggap komputasi DNA, tetapi tidak semua komputasi DNA berhubungan dengan kriptografi DNA (Maniyath & Supriya, 2011).

Kriptografi DNA dipilih sebagai algoritma kriptografi karena seperti yang diketahui bahwa sebagian besar algoritma kriptografi melibatkan memori dan komputasi yang besar, yang seperti kita ketahui sekarang adalah abad dimana terjadi ledakan informasi, dan informasi telah menjadi sumber daya strategis yang sangat penting terutama bagi perusahaan besar. Itulah mengapa keamanan informasi atau data menjadi sangat penting. Dengan kemajuan teknologi informasi dan perkembangan teknik baru, ancaman penyadapan yang dihadapi oleh pengirim dan penerima telah meningkat. Dari beberapa penelitian-penelitian yang telah berkembang, diketahui bahwa satu gram DNA mengandung 10^{21} basis DNA dan dapat menyimpan 10^{8} terabyte memori. Satu triliun bit data biner dapat disimpan dalam satu desimeter kubik larutan DNA. Selain itu, komputasi berbasis DNA membutuhkan waktu yang lebih singkat dibandingkan dengan algoritma lain (Omer & Farooq, 2015).

Rangkaian DNA terdiri dari empat basa asam nukleat (yang selanjutnya disingkat menjadi basa): A (adenin), C (sitosin), G (guanin), dan T (timin), dimana

A dan T saling melengkapi serta G dan C saling melengkapi. Karena 0 dan 1 saling melengkapi dalam biner, maka 00 dan 11 saling melengkapi, 01 dan 10 juga saling melengkapi (X. Y. Wang et al., 2015). Dengan menggunakan empat basis A, C, G, dan T untuk menyandikan 00, 01, 10, dan 11, terdapat 8 jenis aturan pengkodean yang memenuhi aturan pasangan basa Watson-Crick (Paul et al., 2016). Aturan pasangan basa Watson-Crick selengkapnya bisa dilihat pada tabel 2.1.

Tabel 2.1. Aturan Pasangan Basa Watson-Crick

Aturan	1	2	3	4	5	6	7	8
00	A	A	T	T	G	G	C	C
01	G	C	G	C	A	T	A	T
10	C	G	C	G	T	A	T	A
11	T	T	A	A	C	C	G	G

Disini, sistem konversi yang digunakan adalah aturan ke-1 yang mengubah A menjadi 00, G menjadi 01, C menjadi 10, dan T menjadi 11. Juga, digunakan teknik pertukaran kunci simetris untuk menghitung panjang rangkaian OTP DNA. Sebelum masuk ke proses enkripsi dan dekripsinya, terlebih dahulu dibuat kunci simetris K. Pengirim 'A' maupun penerima 'B' menyepakati dua bilangan prima (g dan n). x dan y adalah dua sembarang bilangan yang dipilih oleh pengirim 'A' dan penerima 'B'. Algoritma pertukaran kuncinya adalah sebagai berikut:

1) Hitung $R = (g*x) \bmod n$ dan $S = (g*y) \bmod n$.
   Pengirim mengirim nilai R ke penerima (B) dan penerima mengirim nilai S ke pengirim (A).
2) Pengirim menghitung kunci rahasia $K1 = (S*x) \bmod n$.
3) Penerima menghitung kunci rahasia $K2 = (R*y) \bmod n$.
4) Terakhir, nilai dari K diperoleh, $K = K1 = K2$.

Nilai K pada poin (4) adalah jumlah basa yang ada pada rangkaian nukleotida. Misalnya, jika nilai $K = 6$, maka satu rangkaian nukleotida adalah ACGATA (panjangnya 6) terdiri dari enam basa. Panjang dari kunci akhir atau rangkaian OTP DNA menjadi K*panjang dari *plaintext*.

Selanjutnya langkah-langkah untuk membuat rangkaian OTP DNA adalah sebagai berikut:

1) Balik posisi rangkaian DNA terlebih dahulu (dimana jumlah total dari rangkaian DNA sama dengan panjang dari *plaintext*; misalkan jika ukuran *plaintext* adalah 16 bit, maka terdapat 16 rangkaian nukleotida yang setiap nukleotida terdiri dari enam basa).
2) Cari komplemen dari rangkaian DNA dan anggaplah tiap nukleotida berada pada posisi 0, 1, 2, …, *n* dimana *n* adalah jumlah bit *plaintext* dikurang 1. Kemudian tambahkan 'A' pada akhir nukleotida pertama, 'C' pada akhir nukleotida kedua, 'G' pada akhir nukleotida ketiga, 'T' pada akhir nukleotida keempat, dan seterusnya sampai semua nukleotida telah disisipkan basa tambahan.

Setelah membuat kunci simetris *K* dan rangkaian OTP DNA, selanjutnya melakukan enkripsi dan dekripsi. Langkah-langkah algoritma enkripsinya adalah sebagai berikut:

1) Tentukan *plaintext* yang akan dikirim, ubah ke kode ASCII kemudian ubah ke kode biner.
2) Simpan kode biner (*plaintext*) menurut kolom dari kiri ke kanan dalam sebuah matriks *m*n*, katakan *Z*.
3) Cari cerminan dari matriks diatas (*Z*), katakan *M*.
4) Pilih sembarang rangkaian DNA yang panjangnya sama dengan panjang dari *plaintext* dan susun menurut kolom dalam bentuk matriks *m*n* (dimana nilai *m* dan *n* sama dengan nilai *m* dan *n* sebelumnya), katakan *N*.
5) Gunakan operasi XOR antara matriks *M* dan *N* untuk mendapatkan matriks baru katakan *MN*.
6) Ambil elemen matriks *MN* menurut kolom dan susun menjadi sebuah baris untuk mendapatkan rangkaian biner.
7) Lakukan *scanning* pada rangkaian biner dari kiri ke kanan untuk mencari kemunculan '1'. Setelah '1' ditemukan, cocokkan posisinya di rangkaian OTP DNA yang telah dibuat sehingga diperoleh rangkaian DNA yang sesuai dengan rangkaian OTP DNA. Kemudian hapus basa yang telah disisipkan tadi.

8) Ulangi langkah ke-7 untuk semua kemunculan ‘1’ lalu gabung semuanya sehingga membentuk rangkaian DNA baru. Setelah itu, cari komplemen dari rangkaian DNA tersebut untuk memperoleh *ciphertext*.
9) Pengirim mengirimkan rangkaian DNA dalam bentuk paket ke penerima.

Sedangkan langkah-langkah algoritma dekripsinya adalah sebagai berikut:

1) Setelah *ciphertext* diterima, cari komplemennya. Kemudian cari rangkaian yang sesuai di rangkaian OTP DNA dan sisipkan kembali basa yang sesuai.
2) Ulangi langkah ke-1 sampai selesai.
3) Catat posisi rangkaian DNA dari rangkaian OTP DNA dan bentuk rangkaian binernya dengan menulis ‘1’ untuk semua kemunculannya. Posisi lain dalam rangkaian biner adalah ‘0’.
4) Susun menurut kolom rangkaian biner yang diperoleh dalam matriks *m***n* katakan *MN*.
5) Ambil sembarang rangkaian DNA yang sama dengan yang digunakan pada operasi algoritma enkripsi sebelumnya dan susun menurut kolom ke dalam sebuah matriks *m***n* (dimana *m*, *n* adalah nilai yang sama digunakan sebelumnya), katakan *N*.
6) Lakukan operasi XOR antara *MN* dan *N* untuk mendapatkan matriks *M*.
7) Cari bayangan dari *M*, katakan *Z*.
8) Susun menurut kolom elemen dari matriks *Z* menjadi sebuah baris untuk membentuk rangkaian biner.
9) Ubah rangkaian biner ke kode ASCII yang kemudian diubah menjadi pesan rahasia.

Pengirim harus mengirimkan:

1) Kunci yang dihitung untuk mesin pengirim dan penerima.
2) Sembarang rangkaian DNA yang sama dengan panjang *plaintext* (digunakan untuk operasi XOR) untuk mesin pengirim dan penerima.
3) Rangkaian OTP DNA
4) *Ciphertext*.

Contoh:

Misalkan *plaintext* nya adalah “Hi guys!” untuk enkripsi dan dekripsi. Ubah menjadi kode ASCII lalu ke kode biner, maka diperoleh hasil konversi karakter menjadi biner yang ditunjukkan pada tabel 2.2:

Tabel 2.2. Konversi plaintext menjadi kode ASCII dan biner

Karakter	ASCII	Biner
H	72	01001001
i	105	01101001
spasi	32	00100000
g	103	01100110
u	117	01110101
y	121	01111001
s	115	01110011
!	33	01000001

1. Pembuatan kunci simetris K

Panjang dari *plaintext* adalah 64 bit (mengandung 8 karakter yang setiap karakter terdiri dari 8 bit). Misalkan dipillih bilangan prima $g = 7$, $n = 53$, kemudian pengirim memilih nilai acak misalkan $x = 8$ dan penerima memilih nilai acak misalkan $y = 2$.

1) Hitung $R = (g*x) \bmod n$ dan $S = (g*y) \bmod n$.

   $R = (7*8) \bmod 53 = 3$ dan

   $S = (7*2) \bmod 53 = 14$.

2) Pengirim menghitung kunci rahasia

   $K1 = (S*x) \bmod n = (14*8) \bmod 53 = 6$

3) Penerima menghitung kunci rahasia

   $K2 = (R*y) \bmod n = (3*2) \bmod 53 = 6$

4) Diperoleh nilai K dimana $K = K1 = K2 = 6$.

2. Pembuatan rangkaian OTP DNA

Panjang dari rangkaian OTP DNA adalah *K**panjang dari *plaintext* yaitu (6*64) = 384. Misalkan rangkaian DNA dengan panjang 384 adalah sebagai berikut:

TTAACA GCAGGA GCGCAG TATCTT GGCGTC AGCACA CCTTCA TCAGAT TTGTTA TGCATG GCATGC CAAACG TGCATC GGCCCG TGGTCT CCATAG AGCCAG TGTATA TCGCTT TTCTCT CCCAAA GTGGAA TGGACT ATCTCG TAAAGC CCAACA AGTCCC CGCTAA ATTGTG GAGCGG AATAAT GTCCGA TGCTTG GAGGGT GACAAC GGTGTG ACCGAT ACTACA CCAGGA CCCAAG GGGTAA TCGTGC GAACTG CCGGGG TTTAGC GCTCCC CTAATT CTGTCC TGTTTT GTCGCC CACATT AAGAGC TTTAAA GTCCGC GAACGG AGGGCT GCTATG GAACCA AATTCT TTAGTA ACACAC GTACGC TTGTCA CCCCAT. (terdapat 64 nukleotida dan masing-masing terdiri dari 6 basa).

1) Balik posisi rangkaian DNA, menjadi:

CCCCAT TTGTCA GTACGC ACACAC TTAGTA AATTCT GAACCA GCTATG AGGGCT GAACGG GTCCGC TTTAAA AAGAGC CACATT GTCGCC TGTTTT CTGTCC CTAATT GCTCCC TTTAGC CCGGGG GAACTG TCGTGC GGGTAA CCCAAG CCAGGA ACTACA ACCGAT GGTGTG GACAAC GAGGGT TGCTTG GTCCGA AATAAT GAGCGG ATTGTG CGCTAA AGTCCC CCAACA TAAAGC ATCTCG TGGACT GTGGAA CCCAAA TTCTCT TCGCTT TGTATA AGCCAG CCATAG TGGTCT GGCCCG TGCATC CAAACG GCATGC TGCATG TTGTTA TCAGAT CCTTCA AGCACA GGCGTC TATCTT GCGCAG GCAGGA TTAACA.

Disini yang dibalik adalah posisi nukleotidanya, bukan posisi basanya. Misalkan pada rangkaian awal DNA, nukleotida pertama adalah TTAACA, karena posisinya dibalik maka posisinya berada di paling akhir, kemudian menyusul ke nukleotida kedua yaitu GCAGGA yang kemudian dibalik maka posisinya berada di kedua terakhir, dan seterusnya sampai

pada nukleotida terakhir yang karena dibalik maka posisinya berada di awal rangkaian.

2) Cari komplemen dari rangkaian DNA, menjadi:

GGGGTA AACAGT CATGCG TGTGTG AATCAT TTAAGA CTTGGT CGATAC TCCCGA CTTGCC CAGGCG AAATTT TTCTCG GTGTAA CAGCGG ACAAAA GACAGG GATTAA CGAGGG AAATCG GGCCCC CTTGAC AGCACG CCCATT GGGTTC GGTCCT TGATGT TGGCTA CCACAC CTGTTG CTCCCA ACGAAC CAGGCT TTATTA CTCGCC TAACAC GCGATT TCAGGG GGTTGT ATTTCG TAGAGC ACCTGA GTGGAA CCCAAA TTCTCT TCGCTT TGTATA AGCCAG GGTATC ACCAGA CCGGGC ACGTAG GTTTGC CGTACG ACGTAC AACAAT AGTCTA GGAAGT TCGTGT CCGCAG ATAGAA CGCGTC CGTCCT AATTGT.

Anggaplah tiap nukleotida berada pada posisi 0, 1, 2, …, 63. Kemudian tambahkan ‘A’ pada akhir nukleotida pertama, ‘C’ pada akhir nukleotida kedua, ‘G’ pada akhir nukleotida ketiga, ‘T’ pada akhir nukleotida keempat, dan seterusnya sampai semua nukleotida telah disisipkan basa tambahan, menjadi:

GGGGTAA AACAGTC CATGCGG TGTGTGT AATCATA TTAAGAC CTTGGTG CGATACT TCCCGAA CTTGCCC CAGGCGG AAATTTT TTCTCGA GTGTAAC CAGCGGG ACAAAAT GACAGGA GATTAAC CGAGGGG AAATCGT GGCCCCA CTTGACC AGCACGG CCCATTT GGGTTCA GGTCCTC TGATGTG TGGCTAT CCACACA CTGTTGC CTCCCAG ACGAACT CAGGCTA TTATTAC CTCGCCG TAACACT GCGATTA TCAGGGC GGTTGTG ATTTCGT TAGAGCA ACCTGAC GTGGAAG CCCAAAT TTCTCTA TCGCTTC TGTATAG AGCCAGT GGTATCA ACCAGAC CCGGGCG ACGTAGT GTTTGCA CGTACGC ACGTACG

AACAATT AGTCTAA GGAAGTC TCGTGTG CCGCAGT ATAGAAA CGCGTCC CGTCCTG AATTGTT.

3. Algoritma enkripsi

1) Sesuai dengan tabel 2.2, biner dari *plaintext* adalah 01001001 01101001 00100000 01100110 01110101 01111001 01110011 01000001.
2) Simpan kode biner (*plaintext*) menurut kolom dari kiri ke kanan dalam sebuah matriks *m***n*, katakan *Z*.

$$Z = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Aturan penyusunan binernya disesuaikan dengan ukuran matriks yang dibuat. Karena pada contoh dibuat matriks berukuran 8*8, maka 8 bit pertama disusun di kolom pertama, kemudian 8 bit kedua disusun di kolom kedua, dan seterusnya hingga ke 8 bit terakhir yang disusun di kolom terakhir.

3) Cari cerminan dari matriks diatas (*Z*), katakan *M*.

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

4) Pilih sembarang rangkaian DNA yang panjangnya sama dengan panjang dari *plaintext* dan susun menurut kolom dalam bentuk matriks *m***n* (dimana nilai *m* dan *n* sama dengan nilai *m* dan *n* sebelumnya), katakan *N*. Misalkan dipilih GTTCGTCTCGTATCTAACTACCAGTACGCTCA dan kode binernya adalah 01111110011110111001110011101100001011 00101000011100100110111000. Aturan penyusunan binernya sama seperti penyusunan biner pada matriks *Z*.

$$N = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

5) Gunakan operasi XOR antara matriks *M* dan *N* untuk mendapatkan matriks baru katakan *MN*.

$$MN = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

6) Ambil elemen matriks *MN* menurut kolom dan susun menjadi sebuah baris untuk mendapatkan rangkaian biner, diperoleh 00111111000010001110010110011001010010101000000110100000111100 01.

7) Lakukan *scanning* pada rangkaian biner dari kiri ke kanan untuk mencari kemunculan ‘1’. Setelah ‘1’ ditemukan, cocokkan posisinya di rangkaian OTP DNA yang telah dibuat sehingga diperoleh rangkaian DNA yang sesuai dengan rangkaian OTP DNA. Sesuai dengan rangkaian biner yang diperoleh tadi, menurut langkah ke-7, menjadi:

   CATGCGG TGTGTGT AATCATA TTAAGAC CTTGGTG CGATACT TTCTCGA GACAGGA GATTAAC CGAGGGG CTTGACC CCCATTT GGGTTCA TGGCTAT CCACACA ACGAACT TTATTAC GCGATTA GGTTGTG TAGAGCA AGCCAGT GGTATCA CCGGGCG AGTCTAA GGAAGTC TCGTGTG CCGCAGT AATTGTT.

   Kemudian hapus basa yang telah disisipkan tadi, menjadi:

   CATGCG TGTGTG AATCAT TTAAGA CTTGGT CGATAC TTCTCG GACAGG GATTAA CGAGGG CTTGAC CCCATT GGGTTC TGGCTA CCACAC ACGAAC TTATTA GCGATT

GGTTGT TAGAGC AGCCAG GGTATC CCGGGC AGTCTA GGAAGT TCGTGT CCGCAG AATTGT.

8) Setelah itu, cari komplemen dari rangkaian DNA tersebut untuk memperoleh *ciphertext*. Diperoleh hasil akhir sebagai berikut:

   GTACGC ACACAC TTAGTA AATTCT GAACCA GCTATG AAGAGC CTGTCC CTAATT GCTCCC GAACTG GGGTAA CCCAAG ACCGAT GGTGTG TGCTTG AATAAT CGCTAA CCAACA ATCTCG TCGGTC CCATAG GGCCCG TCAGAT CCTTCA AGCACA GGCGTC TTAACA.

4. Algoritma dekripsi

1) Setelah *ciphertext* diterima, misalkan GTACGC ACACAC TTAGTA AATTCT GAACCA GCTATG AAGAGC CTGTCC CTAATT GCTCCC GAACTG GGGTAA CCCAAG ACCGAT GGTGTG TGCTTG AATAAT CGCTAA CCAACA ATCTCG TCGGTC CCATAG GGCCCG TCAGAT CCTTCA AGCACA GGCGTC TTAACA, cari komplemennya, menjadi CATGCG TGTGTG AATCAT TTAAGA CTTGGT CGATAC TTCTCG GACAGG GATTAA CGAGGG CTTGAC CCCATT GGGTTC TGGCTA CCACAC ACGAAC TTATTA GCGATT GGTTGT TAGAGC AGCCAG GGTATC CCGGGC AGTCTA GGAAGT TCGTGT CCGCAG AATTGT. Kemudian cari rangkaian yang sesuai di rangkaian OTP DNA dan sisipkan kembali basa yang sesuai.
2) Ulangi langkah ke-1 sampai selesai. Diperoleh: CATGCGG TGTGTGT AATCATA TTAAGAC CTTGGTG CGATACT TTCTCGA GACAGGA GATTAAC CGAGGGG CTTGACC CCCATTT GGGTTCA TGGCTAT CCACACA ACGAACT TTATTAC GCGATTA GGTTGTG TAGAGCA AGCCAGT GGTATCA CCGGGCG AGTCTAA GGAAGTC TCGTGTG CCGCAGT AATTGTT.
3) Catat posisi rangkaian DNA dari rangkaian OTP DNA dan bentuk rangkaian binernya dengan menulis ‘1’ untuk semua kemunculannya. Posisi lain dalam rangkaian biner adalah ‘0’. Hasilnya adalah

0011111100001000111001011001100101001010100000011010000011110001.

4) Susun menurut kolom rangkaian biner yang diperoleh dalam matriks *m***n* katakan *MN*. Aturan penyusunan binernya sama seperti penyusunan biner pada proses enkripsi sebelumnya.

$$MN = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

5) Ambil sembarang rangkaian DNA yang sama dengan yang digunakan pada operasi algoritma enkripsi sebelumnya dan susun menurut kolom ke dalam sebuah matriks *m***n* (dimana *m*, *n* adalah nilai yang sama digunakan sebelumnya), katakan *N*.

$$N = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

6) Lakukan operasi XOR antara *MN* dan *N* untuk mendapatkan matriks *M*.

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

7) Cari bayangan dari *M*, katakan *Z*.

$$Z = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

8) Susun menurut kolom elemen dari matriks *Z* menjadi sebuah baris untuk membentuk rangkaian biner, menjadi 0100100101101001001000000110011001110101011110010111001101000001.
9) Ubah rangkaian biner ke kode ASCII yang kemudian diubah menjadi pesan rahasia. Hasilnya diperoleh sesuai dengan pesan yang diinginkan yaitu “Hi guys!”.

2.7. *Root Mean Squared Error*

RMSE (*Root Mean Squared Error*) didefinisikan sebagai akar kuadrat dari kuadrat rata-rata kesalahan atau beda antara citra yang akan dibandingkan dengan citra asli. Sederhananya, RMSE adalah akar kuadrat dari MSE (*Mean Square Error*). RMSE dihitung menggunakan rumus:

$$RMSE = \sqrt{\frac{1}{M * N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I(i,j) - K(i,j)]^2}$$

dimana *M* dan *N* adalah lebar dan tinggi citra, $I(i,j)$ adalah citra yang akan dibandingkan, $K(i,j)$ adalah citra asli, *i* dan *j* adalah baris dan kolom dari piksel dari kedua citra. RMSE dapat digunakan untuk mengukur kemiripan dari sebuah citra, dimana jika nilai RMSE nol maka citra yang dibandingkan identik atau sama persis dengan citra aslinya. RMSE juga dapat digunakan pula untuk mengevaluasi kualitas dari sebuah citra, dimana jika semakin besar nilai RMSE maka semakin kecil distorsi pada citra yang dibandingkan (Asamoah et al., 2018). Namun pada penelitian ini nilai RMSE akan digunakan untuk mengukur tingkat kemiripan dari citra yang telah didekripsi dengan citra asli.

2.8. *Structural Similarity Index Measure*

SSIM (*Structural Similarity Index Measure*) adalah suatu metode yang digunakan untuk mengukur kemiripan diantara kedua gambar. SSIM dihitung menggunakan rumus:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{({\mu_x}^2 + {\mu_y}^2 + C_1)({\sigma_x}^2 + {\sigma_y}^2 + C_2)}$$

yang merupakan pengembangan dari rumus:

$$SSIM(x,y) = [l(x,y)]^{\alpha} \cdot [c(x,y)]^{\beta} \cdot [s(x,y)]^{\gamma}$$

dimana $x = \{x_i | i = 1, 2, \dots, N\}$ dan $y = \{y_i | i = 1, 2, \dots, N\}$ merupakan dua sinyal gambar non negatif yang telah disejajarkan satu sama lain, μ_x, ${\sigma_x}^2$, dan σ_{xy} masing-masing adalah rata-rata dari *x*, varians dari *x*, dan kovarians dari *x* dan *y*. μ_x dan σ_x adalah perkiraan estimasi dari luminansi dan kontras dari *x*, σ_{xy} mengukur kecenderungan dari *x* dan *y* untuk bervariasi bersama-sama, yang hal tersebut merupakan indikasi dari *Structural Similarity*, dan $\alpha, \beta,$ dan γ adalah parameter untuk menentukan hubungan relatif dari ketiga komponen yang pada persamaan diatas nilai ketiganya sama dengan 1. Perbandingan luminansi, kontras, dan struktur dapat diukur menggunakan rumus:

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{{\mu_x}^2 + {\mu_y}^2 + C_1}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{{\sigma_x}^2 + {\sigma_y}^2 + C_2}$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

dimana $C_1, C_2,$ dan C_3 adalah konstan yang masing-masing didefinisikan oleh:

$$C_1 = (K_1L)^2, C_2 = (K_2L)^2, dan\ C_3 = \frac{C_2}{2}$$

dimana *L* adalah rentang dinamis dari nilai piksel (*L* = 255 untuk citra *grayscale* 8 bit/piksel), $K_1 \ll 1$ dan $K_2 \ll 1$ merupakan dua konstan skalar.

SSIM sesuai dengan pengertiannya digunakan untuk mengukur kemiripan diantara kedua gambar, dimana untuk hasil yang efektif disarankan apabila gambar yang akan dibandingkan memiliki ukuran yang sama (Z. Wang et al., 2004).

Karena SSIM merupakan pengembangan dari MSE, maka fungsi dari SSIM sama seperti MSE yang digunakan untuk mengukur tingkat kemiripan dari suatu citra. Tetapi terdapat sedikit perbedaan di hasil akhirnya dimana pada MSE citra identik akan menghasilkan nilai nol sedangkan pada SSIM kemiripan citra direpresentasikan dalam rentang 0 sampai 1, dimana 0 menunjukkan ketidakmiripan citra dan 1 menunjukkan kemiripan citra. Jadi pada SSIM apabila nilainya semakin mendekati 1 maka citra tersebut semakin mirip dengan citra aslinya.

SSIM di dunia pemrosesan citra digital sudah sering digunakan dengan bantuan *library structural_similarity* dengan fungsi *ssim(imageA, imageB)* yang dimiliki oleh *Python*, dimana *imageA* merupakan gambar pembanding dan *imageB* merupakan gambar yang akan dibandingkan.