SKRIPSI

IMPLEMENTASI *HYBRID CRYPTOSYSTEM* MENGGUNAKAN ALGORITMA SPRITZ DAN ALGORITMA RSA DALAM PENGAMANAN CITRA MEDIS

Disusun dan diajukan oleh:

HUSNUL KHATIMAH SYAHRUDDIN
D121201026

PROGRAM STUDI SARJANA TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
GOWA
2024

LEMBAR PENGESAHAN SKRIPSI

IMPLEMENTASI *HYBRID CRYPTOSYSTEM* MENGGUNAKAN ALGORITMA SPRITZ DAN ALGORITMA RSA DALAM PENGAMANAN CITRA MEDIS

Disusun dan diajukan oleh

Husnul Khatimah Syahruddin
D121201026

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka
Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika
Fakultas Teknik Universitas Hasanuddin
Pada tanggal 28 November 2024
dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing Utama,

Dr. Eng. Ady Wahyudi Paundu, S.T., M.T.
NIP 19750313 200912 1 003

Ketua Program Studi,

Prof. Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys., IPM.ASEAN.Eng.
NIP 19750716 200212 1 004

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini ;

Nama	: Husnul Khatimah Syahruddin
NIM	: D121201026
Program Studi	: Teknik Informatika
Jenjang	: S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

Implementasi Hybrid Cryptosystem Menggunakan Algoritma Spritz dan Algoritma RSA dalam Pengamanan Citra Medis

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

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

Gowa, 29 November 2024

Yang Menyatakan

Husnul Khatimah Syahruddin

ABSTRAK

HUSNUL KHATIMAH SYAHRUDDIN. **Implementasi Hybrid Cryptosystem Menggunakan Algoritma Spritz dan RSA dalam Pengamanan Citra Medis** (dibimbing oleh Dr. Eng. Ady Wahyudi Paundu, S.T., M.T).

Citra medis seperti CT-Scan, MRI, dan X-Ray sering digunakan untuk diagnosis pasien, dan dengan berkembangnya teknologi digital, data tersebut memerlukan perlindungan yang kuat untuk mencegah penyalahgunaan. Penelitian ini mengimplementasikan *Hybrid Cryptosystem* yang mengombinasikan algoritma Spritz dan RSA untuk meningkatkan keamanan data citra medis. Algoritma Spritz mengenkripsi citra medis, sementara RSA mengamankan kunci Spritz.

Penelitian ini bertujuan untuk menguji performa algoritma Spritz dan RSA dalam mengamankan citra medis serta mengevaluasi kualitas citra setelah proses enkripsi dan dekripsi.

Tahapan penelitian meliputi pengumpulan citra CT Scan, MRI, dan X-Ray dalam format JPG/JPEG dari sumber terpercaya (Radiopaedia, TCIA, dan Kaggle), enkripsi dan dekripsi citra dengan Spritz, serta enkripsi dan dekripsi kunci Spritz menggunakan RSA. Evaluasi kualitas citra dilakukan dengan mengukur MSE (*Mean Squared Error*) dan PSNR (*Peak Signal-to-Noise Ratio*). Selain itu, dilakukan pengukuran entropi, pengujian waktu proses enkripsi dan dekripsi, analisis ukuran file dan konsumsi CPU.

Hasil penelitian menunjukkan bahwa sistem *hybrid* ini berhasil menjaga kualitas citra dengan nilai PSNR rata-rata di atas 60 dB untuk citra CT, MRI, dan X-Ray. Pada kategori XR_24b_HighRes, nilai PSNR mencapai 69,546 dB dengan MSE 0,0168094, menunjukkan kualitas citra dekripsi mendekati aslinya. Untuk citra MRI_24b_HighRes, diperoleh PSNR rata-rata 61,1106 dB dan MSE 0,051429. Nilai entropi mendekati 8 bits menunjukkan keamanan tinggi. Pengujian menunjukkan resolusi citra memengaruhi waktu proses, dengan rata-rata waktu enkripsi mencapai 15,079608 detik untuk XR_24b_HighRes. Sistem ini menghasilkan file citra terenkripsi dalam format .enc, yang tidak dapat dibuka oleh aplikasi standar, proses dekripsi ke JPG mempertahankan ukuran dan kualitas visual yang mendekati citra asli. Pengujian konsumsi CPU menunjukkan sistem bekerja optimal tanpa membebani sumber daya secara berlebihan. Waktu enkripsi-dekripsi kunci Spritz bervariasi, dengan kunci 64-bit lebih cepat dibandingkan 256-bit dan tetap efisien di bawah 0,005 detik. Sistem *Hybrid Cryptosystem* berbasis Spritz dan RSA ini mampu mengamankan citra medis dengan baik tanpa mengurangi kualitas citra secara signifikan.

Kata kunci: Kriptografi *hybrid;* keamanan data; citra medis; Spritz; RSA

ABSTRACT

HUSNUL KHATIMAH SYAHRUDDIN. **Implementation of Hybrid Cryptosystem Using Spritz and RSA Algorithms for Medical Image Security** (supervised by Dr. Eng. Ady Wahyudi Paundu, S.T., M.T).

Medical images like CT-Scan, MRI, and X-Ray are widely used for patient diagnosis. With advancing digital technology, require strong protection against misuse. This study implements a Hybrid Cryptosystem combining Spritz and RSA algorithms to enhance the security of medical image data. The Spritz algorithm encrypts the images, while RSA secures the Spritz key.

This study aims to evaluate the performance of Spritz and RSA algorithms in securing medical images and examines image quality after encryption and decryption.

Research involved collecting CT-Scan, MRI, and X-Ray images in JPG/JPEG format from trusted sources (Radiopaedia, TCIA, and Kaggle), encrypting and decrypting images with Spritz, and using RSA for the encryption and decryption the Spritz key. MSE (Mean Squared Error) and PSNR (Peak Signal-to-Noise Ratio) measured image quality. Additionally, entropy measurements, encryption and decryption time tests, file size analysis, and CPU consumption assessments are performed.

The results show that this hybrid system successfully maintains image quality, with an average PSNR value above 60 dB for CT, MRI, and X-Ray images. For the XR_24b_HighRes, PSNR reaches 69.546 dB with an MSE of 0.0168094, indicating that the decrypted image quality closely matches the original. MRI_24b_HighRes images achieved an average PSNR of 61.1106 dB and an MSE of 0.0514. Entropy nearing 8 bits demonstrates high security. Tests show that image resolution significantly impacts processing time, with average encryption times up to 15.08 seconds for XR_24b_HighRes. System generates encrypted image files in .enc format, which are unreadable by standard apps, while decryption to JPG preserves the original size and visual quality. CPU usage tests show the system operates optimally without overloading resources. The Spritz key encryption-decryption time varies, with the 64-bit key faster than the 256-bit key and remaining efficient under 0.005 seconds. This Spritz- and RSA-based Hybrid Cryptosystem effectively secures medical images without significantly reducing image quality.

Keywords: Hybrid cryptography; data security; medical images; Spritz; RSA

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR SINGKATAN DAN ARTI SIMBOL

Lambang/singkatan	Arti dan Penjelasan
RSA	*Rivest Shamir Adleman*
CT	*Computed Tomography* (pencitraan tomografi terkomputerisasi)
MRI	*Magnetic Resonance Imaging* (pencitraan resonansi magnetik)
RC4	*Rivest Cipher 4*
KSA	*Key Scheduling Algorithm*
PRGA	*Pseudo-Random Generation Algorithm*
XOR	*Exclusive OR* (operasi bitwise)
MAC	*Message Authentication Code*
ASCII	*American Standard Code for Information Interchange*
DICOM	*Digital Imaging and Communications in Medicine*
JPEG	*Joint Photographic Experts Group*
JPG	*Joint Photographic Group*
CPU	*Central Processing Unit*
MSE	*Mean Squared Error*
PSNR	*Peak Signal-to-Noise Ratio*
IL	*Intermediate Language*
TCIA	*The Cancer Imaging Archive*
GUI	*Graphical User Interface*
WinForms	*Windows Forms*
dB	*Decibel*
PEM	*Privacy-Enhanced Mail* (format berkas untuk sertifikat atau kunci kriptografi)
BIN	*Binary* (format berkas biner)
KB	*Kilobyte* (satuan ukuran data)
mod	*Modulo*
GCD	*Greatest Common Divisor* (Faktor Pembagi Terbesar)
USG	*Ultrasonography*
PET	*Positron Emission Tomography*
n	Modulus dalam kriptografi RSA
$S[i]$	Elemen *array S* dalam algoritma Spritz

Lambang/singkatan	Arti dan Penjelasan
$S[j]$	Elemen *array S* dalam algoritma Spritz
i	Indeks atau variabel dalam algoritma Spritz
j	Indeks atau variabel dalam algoritma Spritz
k	Indeks atau variabel dalam algoritma Spritz
z	Indeks atau variabel dalam algoritma Spritz
W	*Step size* (ukuran langkah yang menentukan perubahan indeks dalam state array di algoritma Spritz)
S	*state array*
$\varphi(n)$	*totient Euler*
e	Eksponen publik
e, n	Eksponen publik dan modulus dalam RSA (kunci publik)
d, n	Eksponen privat dan modulus dalam RSA (kunci privat)
C	*Ciphertext* (pesan yang sudah dienkripsi)
M	*Plaintext* (pesan asli sebelum enkripsi)
$I(i, j)$	nilai piksel pada posisi (i, j) dalam citra asli.
$K(i, j)$	nilai piksel pada posisi (i, j) dalam citra hasil dekripsi.
R	Rentang nilai intensitas piksel dalam citra.
H	Entropi
$p(i)$	Probabilitas kemunculan intensitas pada nilai ke-i
$i \bmod keylength$	Operasi modulo untuk siklus indeks dalam panjang kunci
N	Jumlah nilai kemungkinan piksel dalam citra

DAFTAR LAMPIRAN

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah SWT. yang telah melimpahkan nikmat, rahmat, dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi ini dengan judul "Implementasi *Hybrid Cryptosystem* Menggunakan Algoritma Spritz dan Algoritma RSA dalam Pengamanan Citra Medis" dengan baik sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu (S1) pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Dalam penyusunan skripsi ini, penulis sangat bersyukur atas bimbingan, petunjuk, dan dukungan yang diberikan oleh banyak pihak, sehingga laporan ini dapat diselesaikan tepat waktu. Oleh karena itu, dengan penuh rendah hati, penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Kedua orang tua penulis, Bapak Dr. Syahruddin, S.E., M.Si dan Ibu Kurniawan Harun Rasyid, S.T., M.T, beserta saudara-saudari penulis, yaitu kakak Miftahul Jannah, S.T, kakak Haris A, S.T, adik Nurumayyah Faturrachmah S, dan adik Muhammad Mifzal Mudhoffar S, bersama dengan seluruh keluarga, atas segala doa, dukungan, semangat, pengorbanan, perhatian, dan kasih sayang yang telah mereka berikan.
2. Kedua ponakan penulis, Eshan Rayyan Altair dan Arshan Zayyan Athaillah, yang telah menjadi *moodbooster* selama pengerjaan skripsi ini. Kehadiran mereka selalu memberi semangat dan kebahagiaan, sehingga penulis bisa melalui setiap tantangan dengan senyuman.
3. Bapak Prof. Dr. Ir. Jamaluddin Jompa, M.Sc. selaku Rektor Universitas Hasanuddin.
4. Bapak Prof. Dr. Eng. Ir. Muhammad Isran Ramli, S.T., M.T. selaku Dekan Fakultas Teknik Univeristas Hasanuddin.
5. Bapak Prof. Dr. Indrabayu, S.T., M.T., M.Bus.Sys., IPM., ASEAN.Eng. selaku Ketua Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas bimbingannya selama masa perkuliahan penulis.
6. Bapak Dr. Eng. Ady Wahyudi Paundu, S.T., M.T selaku Dosen Pembimbing yang telah memberikan bimbingan dan masukan yang sangat bermanfaat dalam

penyusunan laporan skripsi ini, serta waktu dalam membantu menyelesaikan skripsi ini.

7. Bapak Dr. Eng. Muhammad Niswar, S.T., M.IT. dan Bapak Iqra Aswad, S.T., M.T. selaku Dosen Penguji yang telah memberikan saran dan kritik kepada penulis dalam penyempurnaan skripsi ini.
8. Bapak Muhammad Alief Fadhal Imran Oemar, S.T., M.Sc. selaku Dosen Pembimbing Akademik yang telah banyak memberikan arahan, bantuan dan motivasi selama masa perkuliahan.
9. Seluruh Dosen dan Staf Pegawai di Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah berbagi ilmu, pengetahuan, dan pengalaman berharga sepanjang masa perkuliahan.
10. Para sahabat penulis, Atira Septiara, Sitti Zakiyah Nurkhalisah, Azzahra Beladina Shaff, Mutia Rahman, Nurza Zahira Faried, S.T, Tri Indah Wahyuningsi, beserta sahabat karib penulis Anggi Herlindia, S.Farm yang telah menemani selama masa perkuliahan dan senantiasa memberikan dukungan, nasehat, doa, dan semangat selama proses perkuliahan.
11. Muh. Arfan Bahrun dan Aldilah Rezki Rhamadani Syahsir, S.T yang telah memberi dukungan dan motivasi kepada penulis untuk terus berusaha menyelesaikan skripsi ini dengan baik.
12. Teman-teman seangkatan dari Program Studi Teknik Informatika angkatan 2020 (Rezolver 20) selaku rekan belajar sejak dari awal hingga akhir masa perkuliahan.
13. Seluruh keluarga besar Lab Ubicon yang telah menjadi wadah bagi penulis untuk berbagi pengalaman dan berkembang secara akademis. Terima kasih atas dukungan serta suasana kerja yang mendukung, sehingga penulis dapat menyelesaikan skripsi ini dengan lancar.
14. Warga A8 (Katrin, Ifah, Silmi, Pia, Salsa, Rara, Wina, Azmi, Wulan, Jauza, Ade, dan Winda), terima kasih atas kebersamaan, dukungan, dan keceriaan yang telah mereka bagi selama di Medan dan sampai sekarang. Mereka telah menjadi keluarga kedua bagi penulis, dan penulis sangat bersyukur memiliki mereka semua di sisi penulis selama ini.

15. Sepupu penulis, Fadli Lie, S.Ked, yang selalu menanyakan progres skripsi penulis dan sering menjadi sumber informasi seputar citra medis. Perhatiannya secara tidak langsung memotivasi penulis untuk terus maju dan lebih rajin dalam menyelesaikan skripsi ini. Terima kasih atas dukungan dan perhatian yang membuat penulis tetap semangat.
16. *Kpop group* "Stray Kids" (Bang Chan, Lee Minho, Seo Changbin, Hwang Hyungjin, Han Jisung, Lee Felix, Kim Seungmin, dan Yang Jeongin) yang secara tidak langsung telah menghibur dengan berbagai kontennya dan menemani penulis dalam proses penulisan skripsi melalui lagu-lagunya.
17. Serta seluruh pihak yang tak sempat penulis sebutkan satu persatu yang telah banyak meluangkan tenaga, waktu, dan pemikiran mereka selama penyusunan laporan skripsi ini dan tanpa sadar telah menjadi inspirasi dan membantu penulis selama penyelesaian skripsi.

Akhirnya dengan segala kerendahan hati, penulis menyadari masih banyak kekurangan dan kesalahan dalam penyusunan skripsi ini, baik dari isi maupun cara penyajiannya. Oleh karena itu, penulis mengharapkan adanya masukan dan kritik yang bersifat membangun demi kesempurnaan laporan ini. Semoga laporan skripsi ini dapat memberikan manfaat bagi pembaca secara umum, dan bagi penulis secara khusus.

Gowa, 18 September 2024

Penulis

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dalam konteks kemajuan teknologi informasi dan sistem kesehatan yang semakin terhubung, pengamanan data medis, termasuk citra medis, menjadi hal yang sangat penting. Citra medis seperti CT Scan, MRI, sinar-X, dan lainnya, mengandung informasi rinci tentang kondisi kesehatan seseorang. Keamanan data medis menjadi prioritas utama karena informasi tersebut sangat sensitif dan privasi pasien harus dilindungi dengan sangat baik. Penyimpanan dan pertukaran citra medis yang tidak aman dapat mengakibatkan pelanggaran privasi pasien, penyalahgunaan data, atau bahkan pencurian identitas medis. Pengamanan citra medis adalah langkah krusial yang perlu dilakukan agar data citra medis tidak disalahgunakan oleh pihak yang tidak berwenang seperti tertulis dalam Undang-Undang Republik Indonesia Nomor 29 Tahun 2004 tentang Praktik Kedokteran yang menyatakan bahwa data rekam medis harus dijaga kerahasiaannya. Pentingnya mengamankan data citra medis adalah untuk menciptakan rasa aman pada masyarakat sebagaimana tercatat bahwa sekitar 70% orang khawatir jika informasi mengenai kesehatan mereka mengalami kebocoran (Ningtyas & Lubis, 2018).

Mengutip *Kompas*, di awal tahun 2022 dokumen sebesar 720 GB yang berupa enam juta data rekam medis seperti hasil pemeriksaan radiologi, termasuk foto dan identitas pasien, serta hasil CT Scan, tes Covid-19, hingga hasil rontgen (X-Ray) lengkap dengan nama pasien, asal rumah sakit, dan waktu pengambilan gambar tersebut dibobol oleh hacker dan dijual di forum online Raidforums. Akun bernama *GOD User*, mengklaim bahwa data ini berasal dari server milik Kementerian Kesehatan yang berasal dari berbagai rumah sakit besar di seluruh Indonesia. Informasi sensitif yang disimpan menyoroti pentingnya citra medis dienkripsi dengan aman dan dapat dipulihkan sempurna setelah didekripsi. Mengantisipasi hal tersebut, dibutuhkan metode keamanan dengan menggunakan kriptografi untuk mengurangi risiko penggunaan yang tidak semestinya.

Adapun jenis citra medis yang menjadi fokus dalam penelitian ini adalah citra CT Scan, MRI, dan X-Ray. CT Scan atau *computerized tomography scan* adalah prosedur pemeriksaan medis yang menggunakan kombinasi teknologi sinar-X dan sistem komputer khusus untuk menghasilkan gambar organ, tulang, dan jaringan lunak di dalam tubuh (AloDokter, 2023). *Magnetic resonance imaging* (MRI) atau pencitraan resonansi magnetik adalah pemeriksaan yang memanfaatkan medan magnet dan energi gelombang radio untuk menampilkan gambar struktur maupun organ dalam tubuh dan menjadikannya salah satu modalitas pencitraan yang sangat penting dalam diagnosa medis (AloDokter, 2023). Pemeriksaan X-Ray atau rontgen adalah salah satu teknik pencitraan medis menggunakan radiasi sinar X untuk melihat gambar organ dalam tubuh (AloDokter, 2022). Pemilihan jenis citra medis yang akan dienkripsi dalam penelitian ini didasarkan pada tingkat kepentingan dan sensitivitas informasi yang terkandung dalam citra tersebut. CT Scan menghasilkan gambar yang lebih kompleks, memungkinkan pengujian enkripsi yang lebih menyeluruh pada data medis sensitif. Sementara itu, MRI dan X-Ray mewakili berbagai jenis pencitraan medis dengan tingkat kompleksitas yang berbeda, memungkinkan penelitian ini untuk menguji fleksibilitas enkripsi pada berbagai format citra medis yang umum digunakan. Pengujian ini dapat menunjukkan keefektifan enkripsi dalam melindungi data sensitif dan kompleks, serta fleksibilitasnya dalam berbagai aplikasi pencitraan medis.

Kriptografi adalah disiplin ilmu yang mempelajari teknik matematika yang berkaitan dengan keamanan informasi, termasuk menjaga kerahasiaan, integritas data, dan proses otentikasi. Sedangkan langkah untuk mengamankan data dalam kriptografi disebut dengan algoritma kriptografi yang terbagi menjadi dua jenis, yaitu simetris dan asimetris. Simetris menggunakan satu kunci untuk enkripsi dan dekripsi, sementara asimetris menggunakan pasangan kunci publik dan privat. Kunci publik digunakan untuk enkripsi, sementara kunci privat digunakan untuk dekripsi (Sulistiyorini & Agus, 2019). Meskipun kriptografi asimetris dianggap lebih aman karena panjang kunci yang lebih panjang, kriptografi simetris lebih cepat. Selain itu, teknik kriptografi simetris rentan terhadap serangan *brute-force*, sementara teknik kriptografi asimetris rentan terhadap serangan *man in the middle* (Al-Shabi, 2019).

Untuk mengatasi beberapa ancaman dan kelemahan yang terdapat pada masing-masing jenis kriptografi dapat dilakukan melalui penggunaan metodologi *Hybrid Cryptosystem. Hybrid Cryptosystem* menggabungkan kedua jenis kriptografi untuk menciptakan sistem yang efisien dan aman, dengan menggunakan algoritma asimetris untuk pertukaran kunci dan algoritma simetris untuk enkripsi dan dekripsi citra. Salah satu algoritma simetris yang akan digunakan adalah Algoritma Spritz yang merupakan varian dari algoritma RC4 yang dikembangkan oleh Ron Rivest dan Jacob Schultz pada tahun 2014. Algoritma ini menghasilkan *sponge-base construction* untuk menghasilkan kunci dalam proses enkripsi dan dekripsi. Penambahan elemen w yang relatif prima terhadap nilai N pada proses *pseudo- random generation algorithm* menjadi perbedaan antara algoritma Spritz dan RC4. Algoritma Spritz menambah tingkat kompleksitas dibandingkan dengan RC4, sehingga waktu yang dibutuhkan untuk mengenkripsi data menjadi lebih lama, namun keamanan yang ditawarkan oleh algoritma Spritz lebih unggul daripada RC4 (Rivest dan Schuldt, 2014).

Dalam algoritma Spritz, pesan yang dienkripsi akan diproses menggunakan metode *stream cipher*, di mana enkripsi dilakukan secara bertahap satu per satu. Kunci yang dihasilkan selanjutnya selalu bergantung pada rangkaian kunci yang dihasilkan sebelumnya. Tahapan inti dalam algoritma Spritz sebagai *stream cipher* terdiri dari tiga tahap, yaitu proses *Key Scheduling Algorithm* (KSA), *Pseudo-Random Generation Algorithm* (PRGA), dan proses enkripsi atau dekripsi.

Adapun algoritma asimetris yang akan digunakan adalah algoritma RSA (Rivest Shamir Adleman) yang diperkenalkan oleh tiga orang profesor MIT (Massachussets Institute of Technology) yaitu Ron Rivest, Adi Shamir, dan Leonard Adleman pada tahun 1978. RSA terdiri dari dua kunci, yaitu kunci publik (*public key*) dan kunci pribadi (*private key*). Keamanan algoritma RSA terletak pada tingkat kesulitan dalam memfaktorkan bilangan bulat n yang besar menjadi faktor-faktor prima (p dan q), yang dalam hal ini $n = p \times q$ (Rinaldi Munir, 2010).

Meskipun algoritma Spritz menawarkan keunggulan dalam hal kecepatan dan efisiensi, serta keamanan yang relatif tinggi, ada beberapa kelemahan yang perlu dipertimbangkan, seperti rentan terhadap serangan *plaintext recovery* dan serangan *brute-force* karena kunci enkripsi dan dekripsi sama. Sementara itu,

algoritma RSA memiliki keunggulan dalam hal keamanan karena kunci privatnya yang sulit dipecahkan, namun proses enkripsi dan dekripsinya cenderung lebih lambat dibandingkan dengan algoritma simetris. Dengan melihat permasalahan tersebut, penulis bermaksud untuk menggabungkan algoritma simetris Spritz dengan algoritma asimetri RSA sehingga memberikan perlindungan untuk kunci simetris serta meningkatkan kecepatan kriptografi kunci asimetris. Selain itu, masih sedikitnya penelitian terkait implementasinya maka dari itu penulis mengusulkan judul "**Implementasi Hybrid Cryptosystem Menggunakan Algoritma Spritz dan Algoritma RSA dalam Pengamanan Citra Medis**." Penelitian ini diharapkan dapat memberikan rasa aman bagi pasien dan memberikan solusi untuk melindungi citra medis sehingga informasi pasien tidak disalahgunakan oleh pihak yang tidak bertanggung jawab.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah pada penelitian ini adalah:

1. Bagaimana menerapkan metode Hybrid Cryptosystem menggunakan algoritma Spritz dan algoritma RSA dalam mengamankan citra medis?
2. Bagaimana kinerja Hybrid Cryptosystem algoritma Spritz dan RSA dalam mengamankan citra medis?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah diuraikan, maka tujuan penelitian ini adalah:

1. Untuk mengetahui penerapan metode Hybrid Cryptosystem menggunakan algoritma Spritz dan algoritma RSA dalam mengamankan citra medis.
2. Untuk menganalisis kinerja metode Hybrid Cryptosystem algoritma Spritz dan algoritma RSA dalam mengamankan citra medis.

1.4 Manfaat Penelitian

Adapun manfaat yang diharapkan dari penelitian ini adalah:

1. Penelitian ini diharapkan dapat membantu meningkatkan keamanan data citra medis dalam lingkungan pelayanan kesehatan, sehingga informasi pasien tidak disalahgunakan oleh pihak yang tidak bertanggung jawab.
2. Meningkatkan pemahaman tentang bagaimana algoritma Spritz dan RSA bekerja, serta menjadi referensi untuk penelitian selanjutnya dengan topik serupa.

1.5 Ruang Lingkup Penelitian

Ruang lingkup yang ditentukan dalam penelitian ini adalah:

1. Data yang dienkripsi adalah citra medis berupa gambar dengan format JPEG atau JPG dengan ukuran m × n pixel.
2. Jenis citra medis yang digunakan adalah jenis citra medis yang umum digunakan dalam praktik medis, yaitu citra CT Scan, MRI, dan X-Ray.

1.6 Sistematika Penulisan

Sistematika penulisan dalam penelitian ini terdiri dari lima bab, yaitu :

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan teori-teori yang menunjang penelitian yang dilakukan.

BAB III METODE PENELITIAN

Bab ini berisi waktu dan lokasi penelitian, tahapan penelitian, analisis kebutuhan sistem, pemodelan sistem, perancangan interface dan skenario pengujian.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi hasil penelitian dan pembahasan.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan hasil penelitian dan saran

BAB II
TINJAUAN PUSTAKA

2.1 Kriptografi

Kriptografi (*Cryptography*) diperoleh dari bahasa Yunani yang memadukan dua suku kata yaitu *kryptos* dan *graphein*. *Krypto* artinya tersembunyi atau rahasia, sedangkan *graphein* artinya tulisan. Arti dari istilah kriptografi ditafsirkan sebagai "tulisan yang tersembunyi" (Simarmata, 2019). Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang terkait aspek keamanan informasi, seperti menjaga kerahasiaan data, keabsahan data, integritas data, dan memverifikasi identitas data. (Hasugian, 2017).

Kriptografi dapat diartikan sebagai ilmu atau seni dalam menjaga kerahasiaan pesan. Ketika sebuah pesan dikirim dari satu tempat ke tempat lain, ada kemungkinan pihak yang tidak berwenang dapat menyadap dan mengetahui isinya. Untuk melindungi pesan ini, pesan tersebut diubah menjadi kode yang sulit dipahami oleh pihak luar (Hasugian, 2017).

Dalam kriptografi, sebuah pesan disebut plainteks (*plaintext)* atau terkadang *cleartext*. Proses untuk menyembunyikan pesan ini dikenal sebagai enkripsi (*encryption*). Pesan yang sudah dienkripsi disebut cipherteks (*ciphertext)*. Proses untuk mengubah cipherteks kembali menjadi plainteks disebut dekripsi (*decryption*) (Simarmata, 2019). Kunci yang digunakan untuk proses enkripsi dan dekripsi informasi disebut *key*, yang biasanya hanya dimiliki oleh pihak yang berwenang terhadap pesan tersebut.

2.2 Tujuan Kriptografi

Menurut Simarmata dkk (2019: 3-4), kriptografi memiliki empat tujuan dasar, yaitu:

1. Kerahasiaan (*confidentiality*) adalah layanan yang berfungsi untuk melindungi pesan agar tidak bisa diakses oleh pihak yang tidak berkepentingan. Dalam bidang kriptografi, layanan ini diwujudkan dengan menyandikan pesan menjadi *ciphertext*. Istilah lain dari kerahasiaan adalah *secrecy* dan *privacy*.

2. Autentikasi (*authentication*) adalah layanan yang berkaitan dengan proses identifikasi, baik untuk memastikan identitas asli pihak-pihak yang berkomunikasi (*user authentication* atau *entity authentication*) maupun untuk memverifikasi keaslian sumber pesan (*data origin authentication*). Dalam bidang kriptografi, layanan ini diimplementasikan dengan menggunakan tanda tangan digital (*digital signature)*.
3. Integritas data (*data integrity*) adalah layanan yang bertujuan memastikan bahwa pesan tetap asli dan tidak mengalami perubahan selama proses pengiriman. Untuk menjaga integritas data, sistem harus mampu mendeteksi perubahan yang dilakukan oleh pihak-pihak yang tidak berhak, seperti penghapusan, modifikasi, atau penyisipan data ke dalam pesan yang sebenarnya. Dalam bidang kriptografi, layanan ini diimplementasikan dengan menggunakan tanda tangan digital (*digital signature*).
4. Tidak ada penyangkalan (*non-repudiation*) adalah layanan yang berfungsi untuk mencegah penyangkalan oleh entitas yang berkomunikasi, sehingga pengirim pesan tidak dapat menyangkal bahwa mereka telah mengirim pesan, dan penerima tidak dapat menyangkal bahwa mereka telah menerima pesan.

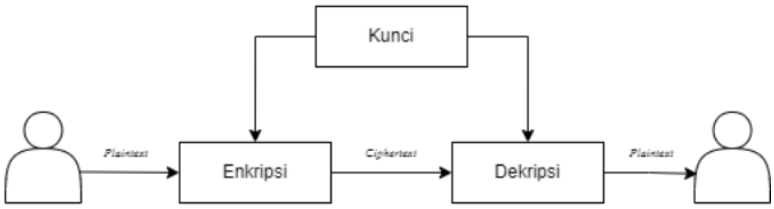
2.3 Jenis-Jenis Algoritma Kriptografi

Berdasarkan jenis kunci yang digunakan untuk enkripsi dan dekripsi, algoritma kriptografi dibagi menjadi dua, yaitu algoritma simetris (*symmetric cryptosystem)* dan algoritma asimetris (*asymmetric cryptosystem*).

1. Algoritma Simetris

Algoritma kriptografi simetris, yang juga dikenal sebagai algoritma kriptografi konvensional, adalah jenis algoritma yang menggunakan kunci yang sama untuk enkripsi dan dekripsi. Secara umum, algoritma ini terbagi menjadi dua tipe, yaitu algoritma aliran (*stream cipher*) dan algoritma blok (*block cipher*). Pada algoritma *stream cipher,* proses enkripsi atau dekripsi dilakukan pada *plaintext* atau *ciphertext* dalam satuan bit, di mana setiap bit dienkripsi atau didekripsi satu per satu. Sementara itu, pada algoritma *block cipher*, proses kriptografi dilakukan pada blok-bit, di mana rangkaian bit dibagi menjadi blok-blok dengan panjang yang telah

ditentukan sebelumnya, dan setiap blok dienkripsi satu per satu (Febrianto & Waluyo, 2023).



Gambar 1 Proses enkripsi dan dekripsi pada algoritma simetris

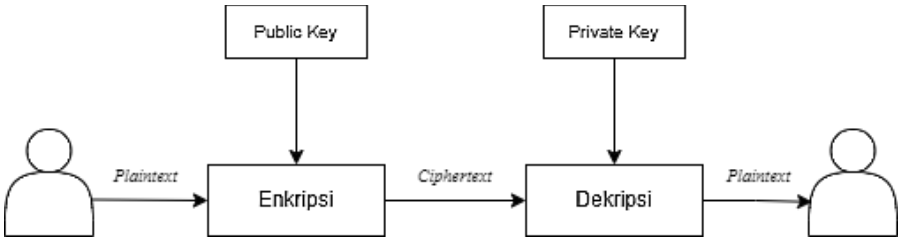
Algoritma kriptografi simetris adalah jenis kriptografi yang paling umum digunakan, di mana kunci yang digunakan untuk menyandi pesan sama dengan kunci yang digunakan untuk membuka pesan tersebut. Dalam hal ini, baik pengirim maupun penerima pesan harus memiliki kunci yang sama (Basri, 2016). Keamanan pesan yang dienkripsi dengan algoritma ini bergantung pada kerahasiaan kunci. Jika kunci tersebut diketahui oleh pihak lain, maka pihak tersebut dapat melakukan enkripsi dan dekripsi pada pesan tersebut. Beberapa contoh algoritma simetris, antara lain DES, AES, Blowfish, RC4, Spritz, Serpent, dan lainnya.

Beberapa kelebihan penggunaan kunci simetris adalah kecepatan operasi yang lebih tinggi dibandingkan dengan algoritma asimetris, meskipun hal ini berbanding lurus dengan peningkatan ukuran file. Kecepatan proses enkripsi dan dekripsi sangat bergantung pada ukuran file; semakin besar ukuran file, semakin banyak waktu yang dibutuhkan untuk enkripsi dan dekripsi (Basri, 2016 dalam **Syahputra dkk, 2021**). Namun, kelemahan dari algoritma simetris terletak pada tantangan distribusi kunci, di mana kunci yang digunakan harus tetap rahasia dan sulit untuk disampaikan dengan aman melalui jalur publik (Tampubolon, 2021).

2. Algoritma Asimetris

Algoritma asimetris, yang juga dikenal sebagai kriptografi *public key*, adalah sistem kriptografi yang menggunakan pasangan kunci, di mana satu kunci digunakan untuk enkripsi dan kunci lainnya untuk dekripsi. Dalam kriptografi kunci publik, kunci enkripsi (publik) dapat digunakan oleh siapa saja untuk mengenkripsi

pesan, sementara hanya pemilik kunci dekripsi (privat) yang dapat mendekripsinya. Proses enkripsi dilakukan dengan menggunakan kunci publik yang dipublikasikan untuk umum, sementara kunci dekripsi disimpan secara rahasia oleh pemiliknya. Dengan cara ini, siapa pun dapat mengirimkan pesan yang terenkripsi kepada pemilik kunci, namun hanya pemilik kunci yang dapat membukanya (Febrianto & Waluyo, 2023). Algoritma asimetris memiliki kelemahan pada kecepatan proses enkripsi dan dekripsi yang cukup lambat, namun tidak ada masalah dalam *key distribution* (Pljonkin, 2021). Beberapa contoh algoritma asimetris antara lain RSA, ECC, ElGamal, LUC, Rabin-P, dan lainnya.



Gambar 2 Proses enkripsi dan dekripsi pada algoritma asimetris

2.4 Metode *Hybrid Cryptosystem*

Hybrid Cryptosystem adalah pendekatan dalam kriptografi yang menggabungkan keunggulan dari dua jenis algoritma kriptografi utama, yaitu algoritma simetris dan algoritma asimetris. Dalam sistem ini, algoritma simetris digunakan untuk mengenkripsi data secara efisien, sedangkan algoritma asimetris digunakan untuk mengamankan kunci enkripsi simetris. Adapun skema enkripsi dan dekripsi dalam sistem kriptografi hibrida, prosesnya biasanya melibatkan kombinasi antara kriptografi asimetris dan kriptografi simetris. Berikut adalah penjelasan tentang skema proses enkripsi dan dekripsi dalam *hybrid cryptosystem*:

1. Proses Enkripsi

   Proses enkripsi dimulai dengan mengenkripsi pesan asli menggunakan kunci simetris, menghasilkan teks terenkripsi atau *ciphertext*. Kemudian, kunci simetris tersebut juga dienkripsi menggunakan kunci publik penerima, menciptakan kunci terenkripsi yang disebut *shared key*. *Ciphertext* dan *shared key* kemudian dikirimkan kepada penerima.

2. Proses Dekripsi:

   Dalam proses dekripsi, penerima menggunakan kunci privatnya untuk mendekripsi *shared key* yang diterima, mengembalikannya menjadi kunci simetris yang asli. Setelah itu, penerima menggunakan kunci simetris tersebut untuk mendekripsi *ciphertext* dan mendapatkan kembali pesan asli

2.5 Algoritma Spritz

Algoritma Spritz merupakan pembaruan atau varian dari algoritma RC4 yang dilakukan oleh Ron Rivest dan Jacob Schultz pada tahun 2014. Algoritma ini beroperasi dengan menggunakan konsep *stream cipher*, di mana proses enkripsi dilakukan secara bertahap satu demi satu. Penambahan elemen w yang relatif prima terhadap nilai N dalam proses *Pseudo-Random Generation Algorithm* menjadi perbedaan utama dibandingkan dengan algoritma RC4. Algoritma Spritz meningkatkan kompleksitas yang lebih tinggi dibandingkan dengan RC4, sehingga waktu yang digunakan untuk mengenkripsi sebuah data juga meningkat. Selain itu, tingkat keamanan yang ditawarkan oleh algoritma Spritz juga lebih baik dibandingkan dengan RC4 (Rivest dan Schuldt, 2014).

Salah satu kelebihan algoritma ini adalah proses pembangkitan kunci yang digunakan dalam proses enkripsi dan dekripsi. Kunci yang dibangkitkan berikutnya selalu bergantung pada aliran kunci sebelumnya. Selain digunakan sebagai *stream cipher,* algoritma Spritz juga dapat berfungsi sebagai *hash* dan *Message Authentication Code* (MAC) dengan memanfaatkan fungsi *sponge* untuk melindungi data. Proses utama dari algoritma Spritz sebagai *stream cipher* mencakup tiga tahap, yaitu *Key Scheduling Algorithm* (KSA), *Pseudo-Random Generation Algorithm* (PRGA), serta proses enkripsi atau dekripsi (Zebua, 2018).

2.6 *Key Scheduling Algorithm (KSA)*

Key Scheduling Algorithm, atau algoritma penjadwalan kunci, merupakan bagian penting dalam algoritma *stream cipher* seperti Spritz dan RC4. KSA berfungsi untuk membentuk tabel *S-Box* (*array S*) serta melakukan permutasi pada tabel dalam *array S*. Panjang *array* yang dibutuhkan adalah 256, dimulai dari indeks 0 hingga 255. Tujuan utama KSA adalah melakukan permutasi *array*

sebanyak 256 kali, yang diinisialisasi dengan variabel *i* dan *j* bertipe *integer*, masing-masing dimulai dengan nilai awal 0 (Nasution, 2020). Adapun *key* adalah kunci rahasia yang akan digunakan.

Rumus KSA adalah:
untuk i = 0 hingga N – 1
S[i] = i
selanjutnya ii, j = 0
untuk i = 0 hingga N – 1
j = (j + S[i] + *key*[i mod keylength]) mod N
swap(S[i], S[j])
j = j selanjutnya i

di mana N merupakan ukuran array yang akan dipermutasi, misalnya dari 0 hingga 255.

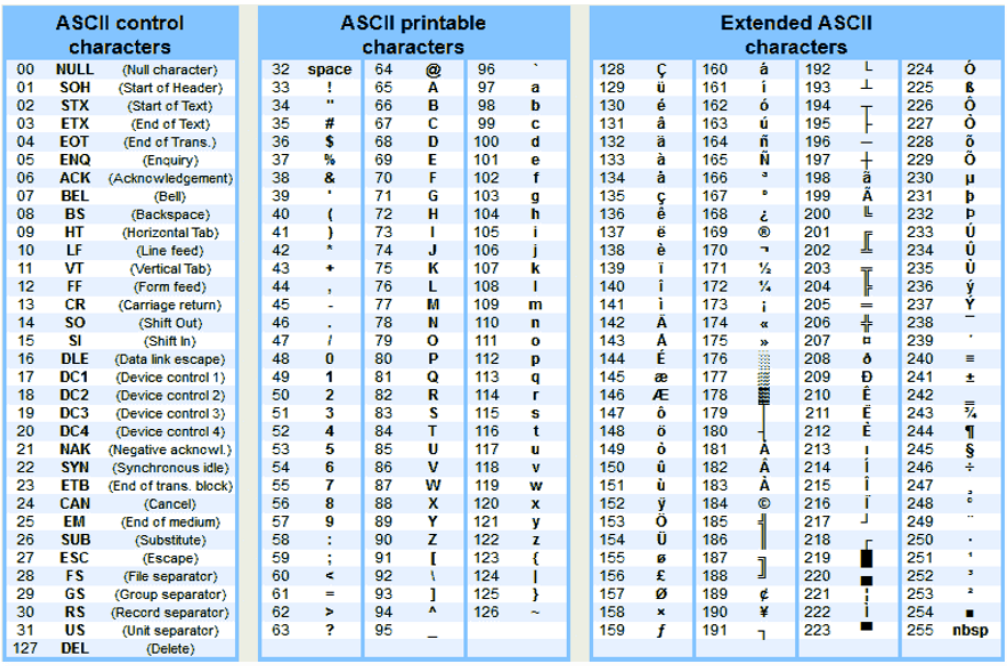
Sebagai contoh, proses enkripsi manual dengan menggunakan algoritma Spritz diterapkan pada citra CT *Scan grayscale*. Pada contoh ini, proses enkripsi manual menggunakan algoritma Spritz dilakukan untuk mengenkripsi citra CT *scan grayscale* dengan mengambil nilai desimal dari sampel 3x2 piksel dari gambar CT *scan grayscale* berukuran 512 x 512 dengan *bit depth* 8-bit, menggunakan kode Python dengan OpenCV yang terdapat pada **Lampiran 1.**

Adapun nilai dari masing-masing piksel sampel yang diambil adalah sebagai berikut:

Tabel 1. Nilai desimal dari setiap *pixel* sampel 3 x 2

Pixel	**Nilai Desimal**
P1 (153, 146)	36
P2 (154, 146)	52
P3 (155, 146)	77
P4 (153, 147)	61
P5 (154, 147)	82
P6 (155, 147)	101

Kunci yang diterapkan dalam enkripsi manual ini adalah "MINHO," dengan panjang kunci 5 *byte*. Kunci ini kemudian dikonversi ke dalam nilai ASCII. Kode ASCII tersebut dapat dilihat pada **Gambar 3** di bawah ini.



ASCII control characters

00	NULL	(Null character)
01	SOH	(Start of Header)
02	STX	(Start of Text)
03	ETX	(End of Text)
04	EOT	(End of Trans.)
05	ENQ	(Enquiry)
06	ACK	(Acknowledgement)
07	BEL	(Bell)
08	BS	(Backspace)
09	HT	(Horizontal Tab)
10	LF	(Line feed)
11	VT	(Vertical Tab)
12	FF	(Form feed)
13	CR	(Carriage return)
14	SO	(Shift Out)
15	SI	(Shift In)
16	DLE	(Data link escape)
17	DC1	(Device control 1)
18	DC2	(Device control 2)
19	DC3	(Device control 3)
20	DC4	(Device control 4)
21	NAK	(Negative acknowl.)
22	SYN	(Synchronous idle)
23	ETB	(End of trans. block)
24	CAN	(Cancel)
25	EM	(End of medium)
26	SUB	(Substitute)
27	ESC	(Escape)
28	FS	(File separator)
29	GS	(Group separator)
30	RS	(Record separator)
31	US	(Unit separator)
127	DEL	(Delete)

ASCII printable characters

32	space	64	@	96	`						
33	!	65	A	97	a						
34	"	66	B	98	b						
35	#	67	C	99	c						
36	$	68	D	100	d						
37	%	69	E	101	e						
38	&	70	F	102	f						
39	'	71	G	103	g						
40	(	72	H	104	h						
41	)	73	I	105	i						
42	*	74	J	106	j						
43	+	75	K	107	k						
44	,	76	L	108	l						
45	-	77	M	109	m						
46	.	78	N	110	n						
47	/	79	O	111	o						
48	0	80	P	112	p						
49	1	81	Q	113	q						
50	2	82	R	114	r						
51	3	83	S	115	s						
52	4	84	T	116	t						
53	5	85	U	117	u						
54	6	86	V	118	v						
55	7	87	W	119	w						
56	8	88	X	120	x						
57	9	89	Y	121	y						
58	:	90	Z	122	z						
59	;	91	[	123	{						
60	<	92	\	124	|					
61	=	93	]	125	}						
62	>	94	^	126	~						
63	?	95	_								

Extended ASCII characters

128	Ç	160	á	192	└	224	Ó				
129	ü	161	í	193	┴	225	ß				
130	é	162	ó	194	┬	226	Ô				
131	â	163	ú	195	├	227	Ò				
132	ä	164	ñ	196	─	228	õ				
133	à	165	Ñ	197	┼	229	Õ				
134	å	166	ª	198	ã	230	µ				
135	ç	167	º	199	Ã	231	þ				
136	ê	168	¿	200	╚	232	Þ				
137	ë	169	®	201	╔	233	Ú				
138	è	170	¬	202	╩	234	Û				
139	ï	171	½	203	╦	235	Ù				
140	î	172	¼	204	╠	236	ý				
141	ì	173	¡	205	═	237	Ý				
142	Ä	174	«	206	╬	238	¯				
143	Å	175	»	207	¤	239	´				
144	É	176	░	208	ð	240	≡				
145	æ	177	▒	209	Ð	241	±				
146	Æ	178	▓	210	Ê	242	‗				
147	ô	179	│	211	Ë	243	¾				
148	ö	180	┤	212	È	244	¶				
149	ò	181	Á	213	ı	245	§				
150	û	182	Â	214	Í	246	÷				
151	ù	183	À	215	Î	247	¸				
152	ÿ	184	©	216	Ï	248	°				
153	Ö	185	╣	217	┘	249	¨				
154	Ü	186	║	218	┌	250	·				
155	ø	187	╗	219	█	251	¹				
156	£	188	╝	220	▄	252	³				
157	Ø	189	¢	221	¦	253	²				
158	×	190	¥	222	Ì	254	■				
159	ƒ	191	┐	223	▀	255	nbsp				

Gambar 3 Kode ASCII

Sumber: https://theasciicode.com.ar/ascii-control-characters/backspace-ascii-code-8.html

M = 77

I = 73

N = 78

H = 72

O = 79

Jadi, kunci yang digunakan dalam perhitungan berdasarkan gambar tabel kode ASCII di atas adalah: [77, 73, 78, 72, 79].

Selanjutnya, dilakukan pengacakan tabel *state S* menggunakan *Key Scheduling Algorithm* (KSA). Pengacakan ini dilakukan sebanyak 256 kali, dengan urutan awal angka dari 0 hingga 255 yang diinisialisasi dalam *array S*. Proses dimulai dengan inisialisasi variabel $j = 0$. Setiap iterasi pengacakan melibatkan perhitungan nilai indeks j yang diperoleh dengan rumus $j = (j + S[i] + key[i \bmod keylength]) \bmod 256$, di mana i bergerak dari 0 hingga 255.

Perhitungan dimulai pada iterasi pertama dengan $i = 0$ dan $j = 0$

$j = (j + S[i] + key[i \bmod keylength]) \bmod 256$

$j = (0 + S[0] + key[0 \bmod 5]) \bmod 256$

$j = (0 + 0 + 77) \bmod 256$

$j = 77 \bmod 256 = 77$

$swap(S[i], S[j]) = swap(S[0], S[77])$

$S[0] = 77$

$S[77] = 0$

Perhitungan iterasi kedua dengan $i = 1$ dan $j = 77$

$j = (j + S[i] + key[i \bmod keylength]) \bmod 256$

$j = (77 + S[1] + key[1 \bmod 5]) \bmod 256$

$j = (77 + 1 + 73) \bmod 256$

$j = 151 \bmod 256 = 151$

$swap(S[i], S[j]) = swap(S[1], S[151])$

$S[1] = 151$

$S[151] = 1$

Perhitungan iterasi ketiga dengan $i = 2$ dan $j = 151$

$j = (j + S[i] + key[i \bmod keylength]) \bmod 256$

$j = (151 + S[2] + key[2 \bmod 5]) \bmod 256$

$j = (151 + 2 + 78) \bmod 256$

$j = 231 \bmod 256 = 231$

$swap(S[i], S[j]) = swap(S[2], S[231])$

$S[2] = 231$

$S[231] = 2$

Perhitungan iterasi keempat dengan $i = 3$ dan $j = 231$

$j = (j + S[i] + key[\text{i} \bmod keylength]) \bmod 256$

$j = (231 + \text{S}[3] + key[3 \bmod 5]) \bmod 256$

$j = (231 + 3 + 72) \bmod 256$

$j = 306 \bmod 256 = 50$

$swap(S[i], S[j]) = swap(S[3], S[50])$

$S[3] = 50$

$S[50] = 3$

Langkah ini diulangi sampai i = 255, dan hasil dari tahap *key scheduling* dapat dilihat pada **Tabel 2** di mana nilai i terletak pada baris yang ditandai dengan warna hijau tosca.

Tabel 2. Hasil perhitungan tahap *Key Scheduling Algorithm* 256 *byte*

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

2.7 *Pseudo-Random Generation Algorithm (PRGA)*

Pseudo-Random Generation Algoritma (PRGA) atau pengacakan adalah bagian penting dari algoritma *stream cipher*, yang menghasilkan stream kunci atau urutan bit acak untuk mengenkripsi atau mendekripsi data. PRGA digunakan untuk

menghasilkan kunci baru berdasarkan beberapa elemen teks asli. Nilai *W* merupakan variabel tambahan yang dimasukkan ke dalam algoritma Spritz, mengikuti konsep algoritma RC4. Nilai variabel *i, j, k,* dan *z* dimulai dari 0 dan akan berubah berdasarkan hasil dari setiap iterasi. Proses ini memanfaatkan nilai-nilai *array S* yang telah diproses dalam tahap KSA (Nasution, 2020).

Rumus PRGA adalah:
untuk i = 0 hingga *plain.length*
$i = (i + w) \bmod N$
$j = (k + S[j + S[i]]) \bmod N$
$k = (i + k + S[j]) \bmod N$
swap $S[i], S[j]$
$z = (S[j + S[i + S[z + k]]]) \bmod N$
output z selanjutnya *i*

di mana *w* adalah bilangan GCD atau relatif prima dengan panjang N dan nilai *i, j, k, z* dimulai dari 0.

Setelah KSA selesai, dilakukan proses *Pseudo-Random Generation Algorithm* (PRGA) untuk menghasilkan *keystream z* yang akan digunakan dalam enkripsi. Proses ini melibatkan perhitungan nilai *z* untuk setiap *byte* yang akan dienkripsi atau sebanyak panjang nilai *pixel plainimage*. Misalkan hasil dari PRGA untuk 6 *byte* yang dibutuhkan adalah $z1, z2, z3, z4, z5, z6$. Adapun nilai desimal dari sampel citra yang digunakan, yaitu: [36, 52, 77, 61, 82, 101].

Perhitungan nilai $z1$ untuk $i = 0$, $j = 0$, $k = 0$, $z = 0$, $w = 17$

$i = (i + w) \bmod 256 \quad = (0 + 17)\ mod\ 256 = 17$

$j = (k + S[j + S[i]]) \bmod N \quad = (0 + S[0 + S[17]]) \bmod 256$

$= (0 + S[0 + 100]) \bmod 256$

$= (0 + S[100]) \bmod 256$

$= (0 + 238) \bmod 256$

	= 238 mod 256 = 238
$k = (i + k + S[j])$ mod N	= (17 + 0 + S[238]) mod 256
	= (17 + 0 + 244) mod 256
	= 261 mod 256 = 5
$S[i], S[j]$	= S[i] = S[17] = 100, S[j] = S[238] = 244
$swap\ S[i], S[j]$	= S[i] = S[17] = 244, S[j] = S[238] = 100
$z = (S[j + S[i + S[z + k]]])$ mod N	= (S[238 + S[17 + S[0 + 5]]]) mod 256
	= (S[238 + S[17 + S[5]]]) mod 256
	= (S[238 + S[17 + 87]])mod 256
	= (S[238 + S[104]]) mod 256
	= (S[238 + 141]) mod 256
	= S[123] mod 256
	= 134 mod 256 = 134

Perhitungan nilai *z*2 untuk *i* = 17, *j* = 238, *k* = 5, *z* = 134, *w* = 17

$i = (i + w)$ mod 256	= (17 + 17) mod 256 = 34
$j = (k + S[j + S[i]])$ mod N	= (5 + S[238 + S[34]])mod 256
	= (5 + S[238 + 103]) mod 256
	= (5 + S[85]) mod 256
	= (5 + 86) mod 256
	= 91 mod 256 = 91
$k = (i + k + S[j])$ mod N	= (34 + 5 + S[91]) mod 256
	= (34 + 5 + 120) mod 256
	= 159 mod 256 = 159
$S[i], S[j]$	= S[i] = S[34] = 103, S[j] = S[91] = 120
$swap\ S[i], S[j]$	= S[i] = S[34] = 120, S[j] = S[91] = 103
$z = (S[j + S[i + S[z + k]]])$ mod N	=(S[91 + S[34 + S[134 + 159]]]) mod 256
	= (S[91 + S[34 + S[37]]]) mod 256
	= (S[91 + S[34 + 66]])mod 256
	= (S[91 + S[100]]) mod 256

= (S[91 + 238]) mod 256
= S[73] mod 256
= 234 mod 256 = 234

Perhitungan nilai *z*3 untuk *i* =34, *j* = 91, *k* = 159, *z* = 234, *w* = 17

$i = (i + w)$ mod 256 = $(34 + 17)\ mod\ 256 = 51$
$j = (k + S[j + S[i]])$ mod N = (159 + S[91 + S[51]])mod 256
= (159 + S[91 + 229]) mod 256
= (159 + S[64]) mod 256
= (159 + 40) mod 256
= 199 mod 256 = 199
$k = (i + k + S[j])$ mod N = (51 + 159 + S[199]) mod 256
= (51 + 159 + 83) mod 256
= 293 mod 256 = 37
$S[i], S[j]$ = S[i] = S[51] = 229, S[j] = S[199] = 83
$swap\ S[i], S[j]$ = S[i] = S[51] = 83, S[j] = S[199] = 229
$z = (S[j + S[i + S[z + k]]])$ mod N=(S[199 + S[51 + S[234 + 37]]]) mod 256
= (S[199 + S[51 + S[15]]]) mod 256
= (S[199 + S[51 + 54]])mod 256
= (S[199 + S[105]]) mod 256
= (S[199 + 130]) mod 256
= S[73] mod 256
= 234 mod 256 = 234

Perhitungan nilai *z4* untuk *i* = 51, *j* = 199, *k* = 37, *z* = 234, *w* = 17

$i = (i + w)$ mod 256 = $(51 + 17)\ mod\ 256 = 68$
$j = (k + S[j + S[i]])$ mod N = (37 + S[199 + S[68]])mod 256
= (37 + S[199 + 135]) mod 256
= (37 + S[78]) mod 256
= (37 + 148) mod 256

$= 185 \bmod 256 = 185$

$k = (i + k + S[j]) \bmod N$ $= (68 + 37 + S[185]) \bmod 256$

$= (68 + 37 + 199) \bmod 256$

$= 304 \bmod 256 = 48$

$S[i], S[j]$ $= S[i] = S[68] = 135, S[j] = S[185] = 199$

$swap\ S[i], S[j]$ $= S[i] = S[68] = 199, S[j] = S[185] = 135$

$z = (S[j + S[i + S[z + k]]]) \bmod N = (S[185 + S[68 + S[234 + 48]]]) \bmod 256$

$= (S[185 + S[68 + S[26]]]) \bmod 256$

$= (S[185 + S[68 + 4]]) \bmod 256$

$= (S[185 + S[72]]) \bmod 256$

$= (S[185 + 8]) \bmod 256$

$= S[193] \bmod 256$

$= 59 \bmod 256 = 59$

Perhitungan nilai *z*5 untuk *i* = 68, *j* = 185, *k* = 48, *z* = 59, *w* = 17

$i = (i + w) \bmod 256$ $= (68 + 17)\ mod\ 256 = 85$

$j = (k + S[j + S[i]]) \bmod N$ $= (48 + S[185 + S[85]]) \bmod 256$

$= (48 + S[185 + 86]) \bmod 256$

$= (48 + S[15]) \bmod 256$

$= (48 + 54) \bmod 256$

$= 102 \bmod 256 = 102$

$k = (i + k + S[j]) \bmod N$ $= (85 + 48 + S[102]) \bmod 256$

$= (85 + 48 + 36) \bmod 256$

$= 169 \bmod 256 = 169$

$S[i], S[j]$ $= S[i] = S[85] = 86, S[j] = S[102] = 36$

$swap\ S[i], S[j]$ $= S[i] = S[85] = 36, S[j] = S[102] = 86$

$z = (S[j + S[i + S[z + k]]]) \bmod N = (S[102 + S[85 + S[59 + 169]]]) \bmod 256$

$= (S[102 + S[85 + S[228]]]) \bmod 256$

$= (S[102 + S[85 + 169]]) \bmod 256$

$= (S[102 + S[254]]) \bmod 256$

$= (S[102 + 9]) \bmod 256$

= S[111] mod 256

= 246 mod 256 = 246

Perhitungan nilai *z*6 untuk *i* = 85, *j* = 102, *k* = 169, *z* = 246, *w* = 17

$i = (i + w)$ mod 256	= $(85 + 17)\ mod\ 256 = 102$
$j = (k + S[j + S[i]])$ mod N	= $(169 + S[102 + S[102]])$mod 256
	= $(169 + S[102 + 36])$ mod 256
	= $(169 + S[138])$ mod 256
	= $(169 + 146)$ mod 256
	= 315 mod 256 = 59
$k = (i + k + S[j])$ mod N	= $(102 + 169 + S[59])$ mod 256
	= $(102 + 169 + 121)$ mod 256
	= 392 mod 256 = 136
$S[i], S[j]$	= $S[i] = S[102] = 36, S[j] = S[59] = 121$
$swap\ S[i], S[j]$	= $S[i] = S[102] = 121, S[j] = S[59] = 36$
$z = (S[j + S[i + S[z + k]]])$ mod N	= $(S[59 + S[102 + S[246 + 136]]])$ mod 256
	= $(S[59 + S[102 + S[126]]])$ mod 256
	= $(S[59 + S[102 + 182]])$mod 256
	= $(S[59 + S[28]])$ mod 256
	= $(S[59 + 41])$ mod 256
	= S[100] mod 256
	= 238 mod 256 = 238

Maka, nilai *keystream z* yang dihasilkan adalah 134, 234, 234, 59, 246, 238.

2.8 Enkripsi dan Dekripsi Algoritma Spritz

Proses enkripsi dan dekripsi dilakukan dengan cara melakukan operasi XOR *biner* antara setiap output z dan masing-masing elemen asli dalam aliran data. Langkah-langkah dalam proses enkripsi dan dekripsi algoritma *stream cipher* biasanya melibatkan operasi-operasi XOR (eXclusive OR) antara data asli dan *stream* kunci.

Rumus enkripsi :

Ci = *Pi* XOR *zi* .....(1)

Rumus dekripsi :

Pi = *Ci* XOR *zi* .....(2)

Dimana:

Pi = elemen asli

Ci = elemen sandi

zi = elemen kunci (hasil dari proses PRGA)

Setelah mendapatkan *keystream z*, tahap enkripsi dilakukan dengan cara meng-XOR setiap nilai piksel citra dengan *byte keystream* yang sesuai. Nilai *output z* dari setiap perhitungan pada *Pseudo-Random Generation Algorithm* (PRGA) akan dikonversi ke bilangan biner. Proses enkripsi manual untuk setiap nilai desimal piksel adalah sebagai berikut.

Tabel 3. Hasil konversi nilai *P1* dan *keystream z* ke bilangan biner

Pixel	Nilai Desimal	Biner	Keystream (z)	Biner
P1	36	00100100	134	10000110
P2	52	00110100	234	11101010
P3	77	01001101	234	11101010
P4	61	00111101	59	00111011
P5	82	01010010	246	11110110
P6	101	01100101	238	11101110

Nilai *Pixel* 1 (P1)

Nilai P1 dengan nilai desimal 36 akan di-XOR dengan *output z* yang diperoleh dari perhitungan PRGA, yaitu 134

36 = 00100100

134 = 10000110 $\oplus$

= 10100010 = 162

Nilai *Pixel* 2 (P2)

Nilai P2 dengan nilai desimal 52 akan di-XOR dengan *output z* yang diperoleh dari perhitungan PRGA, yaitu 234

52 = 00110100

234 = 11101010 ⊕

= 11011110 = 222

Nilai *Pixel* 3 (P3)

Nilai P3 dengan nilai desimal 77 akan di-XOR dengan *output z* yang diperoleh dari perhitungan PRGA, yaitu 234

77 = 01001101

234 = 11101010 ⊕

= 10100111 = 167

Nilai *Pixel 4* (P4)

Nilai P4 dengan nilai desimal 61 akan di-XOR dengan *output z* yang diperoleh dari perhitungan PRGA, yaitu 59.

61 = 00111101

59 = 00111011 ⊕

= 00000110 = 6

Nilai *Pixel 5* (P5)

Nilai P5 dengan nilai desimal 82 akan di-XOR dengan *output z* yang diperoleh dari perhitungan PRGA, yaitu 246

82 = 01010010

246 = 11110110 ⊕

= 10100100 = 164

Nilai *Pixel 6* (P6)

Nilai P6 dengan nilai desimal 101 akan di-XOR dengan *output z* yang diperoleh dari perhitungan PRGA, yaitu 238

101 = 01100101

238 = 11101110 $\oplus$

= 10001011 = 139

2.9 Algoritma *Rivest Shamir Adleman* (RSA)

Algoritma RSA adalah sebuah algoritma *blok cipher* di mana seluruh informasi diubah menjadi sebuah integer. Algoritma RSA terdiri dari dua kunci, yaitu kunci publik yang dapat diakses oleh semua pihak, dan kunci privat yang hanya diketahui oleh pemilik data. Proses enkripsi menggunakan kunci publik, sedangkan proses dekripsi menggunakan kunci privat yang hanya dimiliki oleh pemilik data (Suhandinata et al, 2019).

Algoritma RSA dikembangkan oleh tiga peneliti dari MIT (*Massachusetts Institute of Technology*) pada tahun 1976, yaitu Ron Rivest, Adi Shamir, dan Leonard Adleman. Keamanan algoritma ini didasarkan pada kesulitan dalam memfaktorkan bilangan besar menjadi faktor-faktor primanya, sehingga semakin besar bilangan prima yang digunakan, semakin tinggi tingkat keamanan yang dihasilkan. Pemfaktoran dilakukan untuk memperoleh kunci privat. Algoritma RSA melibatkan tiga tahapan utama, yaitu proses pembangkitan kunci publik dan kunci privat, proses enkripsi, serta proses dekripsi (Dairi et al, 2023).

2.10 Pembangkit Kunci Algoritma RSA

Penggunaan RSA melibatkan dua kunci yang berbeda untuk melakukan enkripsi dan dekripsi. Kunci-kunci tersebut adalah bilangan prima yang besar, di mana faktorisasi bilangan tersebut merupakan hasil dari perkalian dua bilangan prima yang besar. Menemukan faktor-faktor tersebut akan sangat sulit. Algoritma pembangkitan kunci RSA adalah sebagai berikut:

1. Tentukan dua buah bilangan prima p dan q bernilai besar, acak dan dirahasiakan, $p \neq q$, p dan q memiliki ukuran yang sama.
2. Hitung $n = p\ x\ q$

   Besaran *n* tidak perlu dirahasiakan, sebaiknya $p \neq q$, sebab jika $p = q$ maka $n = p^2$ sehingga *p* dapat diperoleh dengan menarik akar pangkat dua dari *n*.
3. Hitung $\varphi(n) = (p-1)\ x\ (q-1)$ bilangan integer n disebut (RSA) modulus.

4. Pilih sebuah bilangan bulat untuk kunci publik yang disebut e, yang relatif prima terhadap $\varphi(n)$. Relatif prima terhadap $\varphi(n)$ artinya faktor pembagi keduanya adalah 1, secara matematis disebut GCD $(e, \varphi(n) = 1)$
5. Bangkitkan kunci privat atau kunci untuk dekripsi (d) dengan rumus $e.d = 1 \, mod \, \varphi(n)$.

   Maka hasil dari algoritma diatas, yaitu:

   a. Kunci publik adalah pasangan (e, n)
   b. Kunci privat adalah pasangan (d, n)

Catatan: n tidak bersifat rahasia, namun ia diperlukan pada perhitungan enkripsi/dekripsi.

2.11 Enkripsi Algoritma RSA

Dalam proses enkripsi, pesan yang akan dienkripsi m diubah menjadi bilangan bulat M (*plaintext*) yang berada dalam rentang 0 hingga n-1. Biasanya, ini dilakukan dengan menambahkan *padding* pada pesan sebelum enkripsi. Setelah pesan diubah menjadi M, enkripsi dilakukan dengan menggunakan kunci publik (e, n). *Ciphertext* C dihitung dengan rumus $\boldsymbol{C \equiv M^e \bmod n}$ (Ginting dkk, 2015), di mana M dipangkatkan dengan *e* dan kemudian diambil modulus *n*. Hasilnya adalah *ciphertext* yang hanya dapat didekripsi dengan kunci privat yang sesuai.

Sebagai contoh, jika *p* = 71 dan *q* = 67, maka $n = 71 \times 67 = 4757$ dan $\varphi(n) = (71 - 1) \times (67 - 1) = 4620$. Jika dipilih e = 19, maka d yang merupakan kebalikan modular dari 19 mod 4620 adalah 1459. Dengan demikian, kunci publik adalah (19, 4757) dan kunci privat adalah (1459, 4757). Jika pesan yang akan dienkripsi adalah *M* = 100, maka *ciphertext C* dihitung sebagai $C \equiv 100^{19} \bmod 4757$, yang menghasilkan *ciphertext* 2917. Proses ini menunjukkan bagaimana enkripsi dalam algoritma RSA mengubah pesan menjadi *ciphertext* yang aman.

2.12 Dekripsi Algoritma RSA

Proses dekripsi dalam algoritma RSA adalah kebalikan dari proses enkripsi dan menggunakan kunci privat untuk mengubah *ciphertext* kembali menjadi *plaintext.* Setelah pesan dienkripsi menggunakan kunci publik dan diubah menjadi *ciphertext* (*C*), dekripsi dilakukan dengan menggunakan kunci privat. Kunci privat terdiri dari pasangan (*d, n*), di mana d adalah kebalikan modular dari e (yang digunakan dalam enkripsi) modulo $\varphi(n)$. Proses dekripsi dimulai dengan mengambil *ciphertext* (*C*) yang telah diterima. Untuk mendekripsi *ciphertext*, algoritma RSA menggunakan rumus $M \equiv C^d \mod n$ (Ginting dkk, 2015), di mana *M* adalah pesan asli (*plaintext*) yang ingin dipulihkan.

Sebagai contoh, jika menggunakan nilai-nilai yang sama seperti pada proses enkripsi, di mana $p = 71$, $q = 67$, $n = 4757$, $\varphi(n) = 4620$, $e = 19$, dan $d = 1459$, serta *ciphertext* $C = 2917$, maka proses dekripsi akan menghitung $M \equiv 2917\text{^}1459 \mod 4757$. Perhitungan ini memanfaatkan eksponen d dan modulus n untuk mengembalikan *ciphertext* ke bentuk aslinya, yaitu *plaintext M*. Hasil dari perhitungan tersebut adalah nilai $M = 100$ yang merupakan pesan asli sebelum dienkripsi.

2.13 Citra Medis

Citra medis merupakan gambaran internal tubuh manusia yang banyak dimanfaatkan dalam penelitian kesehatan. Digunakan oleh para ahli kesehatan untuk mendeteksi dan menganalisis penyakit pasien. Jenis citra yang digunakan untuk mendiagnosa penyakit ini adalah jenis citra khusus yang dihasilkan dari peralatan medis, seperti *Magnetic Resonance Imaging* (MRI), X-Ray, *Ultrasonography* (USG), *Endoscopy*, *Computed Tomography* (CT-Scan), *Positron emission Tomography* (PET) dan *Nuclear Medicine* (Kusuma, 2020).

Citra medis memiliki karakteristik unik yang memengaruhi desain sistem kriptografi:

a. **Sensitivitas**, citra medis mengandung informasi sensitif tentang kondisi kesehatan pasien, sehingga menjadi target akses atau manipulasi yang tidak sah.
b. **Ukuran Besar**, citra medis dapat memiliki ukuran file yang cukup besar, sehingga menimbulkan tantangan untuk enkripsi dan dekripsi yang efisien.

c. **Keanekaragaman Format,** citra medis hadir dalam berbagai format, seperti DICOM, JPEG, dan PNG, masing-masing dengan karakteristiknya sendiri.

2.14 Citra *Computed Tomography Scan (*CT Scan)

CT scan, atau *Computed Tomography scan*, merupakan teknik medis canggih yang menggabungkan sinar-X dan komputer untuk menghasilkan gambar detail organ dan struktur internal tubuh. Berbeda dengan rontgen biasa yang hanya menghasilkan gambar dua dimensi, CT scan mampu menghasilkan gambar potongan melintang yang memberikan gambaran menyeluruh dari berbagai sudut. Kemampuan ini membuat CT scan sangat berharga bagi dokter untuk mendeteksi penyakit seperti kanker, infeksi, dan cedera, serta membantu merencanakan prosedur medis dan memantau respons pasien terhadap pengobatan (AloDokter, 2023).

CT scan menghasilkan gambar yang lebih detail dan tajam dibandingkan rontgen, memungkinkan visualisasi struktur kecil seperti pembuluh darah, tulang rawan, dan jaringan lunak. Dalam beberapa kasus, zat kontras mungkin digunakan untuk meningkatkan kualitas gambar. Secara keseluruhan, CT scan merupakan alat diagnostik penting dalam membantu dokter mendiagnosis dan mengobati berbagai kondisi medis berkat kemampuannya menghasilkan gambar yang detail (AloDokter, 2023).

2.15 Citra X-Ray

Citra X-ray atau rontgen merupakan representasi visual dua dimensi dari bagian dalam tubuh manusia yang dihasilkan melalui proses pencitraan medis menggunakan sinar-X. Sinar-X adalah gelombang elektromagnetik berenergi tinggi yang mampu menembus tubuh manusia dan direkam oleh detektor khusus. Prosedur ini menjadi bagian penting dalam pemeriksaan penunjang, melengkapi informasi yang diperoleh dari pemeriksaan fisik dan riwayat kesehatan. Dokter dapat melihat kondisi tulang dan sendi dengan detail, mendeteksi patah tulang, radang sendi, pembusukan gigi, osteoporosis, bahkan kanker tulang (AloDokter, 2022).

2.16 Citra *Magnetic Resonance Imaging* (MRI)

Magnetic Resonance Imaging (MRI) adalah metode diagnostik penting dalam kedokteran, terutama dalam radiologi, yang menghasilkan gambar potongan tubuh manusia menggunakan medan magnet, tanpa melibatkan sinar-X. MRI mampu menghasilkan gambar dalam berbagai orientasi seperti *coronal, sagittal, axial,* dan *oblique* dengan sedikit manipulasi pada tubuh pasien. Proses pencitraan MRI cukup kompleks, karena kualitas gambar yang dihasilkan sangat bergantung pada berbagai parameter. Dengan pemilihan parameter yang tepat, hasil gambar dapat menampilkan detail tubuh manusia dengan jelas, memungkinkan evaluasi anatomi dan patologi jaringan tubuh yang akurat (Nurhikmah, 2022).

2.17 Visual Basic .NET

Visual Basic .NET merupakan *object-oriented programming language* yang diciptakan oleh Microsoft untuk mengembangkan aplikasi Windows. Visual Basic .NET merupakan bahasa pemrograman yang menawarkan paradigma pemrograman berorientasi objek yang terstruktur, di mana setiap "objek" memiliki atribut dan metodenya sendiri. Diluncurkan pada tahun 2002, Visual Basic .NET hadir sebagai penyempurnaan signifikan dari pendahulunya, Visual Basic 6.0. Keunggulan utama VB .NET terletak pada pemanfaatan .NET *Framework* untuk eksekusinya. Hal ini menghasilkan aplikasi yang lebih andal, terukur, dan kompatibel dengan berbagai platform (Rozikin, 2022).

Kelebihan dari Visual Basic .NET antara lain :

a. Kode yang ditulis secara otomatis akan terformat dengan baik.
b. Menggunakan konsep pemrograman berbasis objek (*object-oriented*) yang memungkinkan pembuatan kode kelas *enterprise*.
c. Aplikasi web dapat dibuat dengan fitur-fitur modern menggunakan Visual Basic .NET.
d. Fitur *drag and drop* mempermudah proses pengembangan aplikasi.
e. Aplikasimu dapat dengan mudah diintegrasikan dengan aplikasi lain yang menggunakan .NET *framework.*
f. Menu editor yang disediakan sangat lengkap dan intuitif.

Sedangkan kekurangan Visual Basic .Net antara lain:

a. Penggunaan *pointer* cukup rumit dalam proses pemrograman.
b. Penambahan kode tambahan dapat memakan siklus CPU lebih banyak, sehingga memperlambat proses.
c. VB .NET mudah didekompilasi *(reverse engineer*) karena menggunakan kompilasi *Intermediate Language* (IL).
d. Meskipun *library* yang tersedia lengkap, hal ini dapat memperpanjang waktu komputasi.
e. Eksekusi aplikasi Visual Basic .NET dilakukan di .NET *Framework*, yang dapat menghasilkan aplikasi dengan kesamaan fitur dengan bahasa pemrograman lain seperti C++, Java, atau C#.

Dengan Visual Basic .NET, kita bisa merancang aplikasi dengan lebih cepat dan menggunakan fitur-fitur modern.

2.18 Parameter Pengujian

Ada beberapa parameter yang akan digunakan untuk mengevaluasi performa dan keamanan sistem enkripsi citra medis ini.

1. *Mean Squared Error* (MSE)

Mean Squared Error (MSE) adalah metrik yang digunakan untuk mengukur kualitas rekonstruksi citra dengan membandingkan citra asli dan citra hasil dekripsi. MSE dihitung dengan cara menjumlahkan kuadrat selisih antara nilai intensitas piksel citra asli dan citra hasil dekripsi, kemudian dibagi dengan jumlah piksel. Nilai MSE yang dihasilkan apabila semakin rendah dan mendekati 0 (nol), maka akan semakin mirip dengan citra aslinya dan bagus kualitas citra tersebut (Yustiantara et al, 2021).

2. *Peak Signal-to-Noise Ratio* (PSNR)

Peak Signal-to-Noise Ratio (PSNR) adalah metrik yang digunakan untuk menilai kualitas citra dengan membandingkan rasio antara sinyal (citra asli) dan noise (distorsi). PSNR dihitung berdasarkan MSE dan dinyatakan dalam satuan desibel (dB). Kualitas citra sudah dikatakan baik jika nilai PSNR di atas 30 dB

(Zakaria et al, 2019). Di bawah 30 itu dikatakan citra mengalami degradasi yang semakin besar dengan menurunnya PSNR (Munir, 2022).

3. Entropi

Entropi adalah ukuran acak atau ketidakpastian dalam distribusi intensitas piksel citra. Dalam konteks pengujian keamanan citra, entropi digunakan untuk mengevaluasi seberapa baik algoritma enkripsi menyembunyikan informasi. Entropi akan memiliki nilai tinggi saat citra menunjukkan ketidakseragaman atau keacakan, sebaliknya akan bernilai rendah ketika keacakan nilai *pixel* citra semakin seragam atau cenderung sama (Zulfikar, 2020).

Untuk kasus enkripsi citra, citra yang terenkripsi harusnya memiliki entropi mendekati nilai maksimum 8. Hal ini disebabkan oleh adanya 256 tingkat keabuan dalam citra (dari m0 = 0 hingga m255 = 255), dan setiap tingkat keabuan ini memiliki peluang tertentu yang dihitung berdasarkan histogram citra. Entropi maksimum akan tercapai ketika semua tingkat keabuan memiliki peluang yang sama, yaitu distribusi yang paling acak. (Munir, 2012).

4. Histogram

Histogram citra adalah representasi grafis dari distribusi intensitas piksel dalam citra. Histogram digunakan untuk menganalisis seberapa baik informasi tersebar dalam citra. Histogram diperoleh dengan menghitung jumlah piksel pada setiap tingkat intensitas dan menampilkannya dalam bentuk grafik. Pengujian menggunakan histogram membantu dalam mengevaluasi bagaimana algoritma enkripsi mempengaruhi distribusi intensitas piksel citra dan memberikan indikasi seberapa acak citra terenkripsi.

5. Waktu Proses Enkripsi dan Dekripsi

Pengukuran waktu proses enkripsi dan dekripsi adalah langkah penting untuk menilai efisiensi algoritma yang digunakan. Ini melibatkan pencatatan waktu sebelum dan setelah proses enkripsi atau dekripsi dilakukan. Hasil pengukuran ini memberikan informasi tentang seberapa cepat sistem dapat memproses citra medis dan kunci enkripsi.

6. Penggunaan CPU

Penggunaan CPU diukur selama proses enkripsi, dekripsi, dan pembangkitan kunci untuk mengamati efisiensi penggunaan sumber daya komputasi sistem. Hal ini bertujuan untuk memastikan bahwa algoritma yang digunakan, yakni Spritz dan RSA, serta proses pembangkitan kunci RSA, dapat berjalan optimal tanpa membebani CPU secara signifikan, sehingga memungkinkan penggunaan yang efisien bahkan pada perangkat dengan kapasitas komputasi terbatas. Pencatatan rata-rata penggunaan CPU dilakukan untuk setiap tahap proses dalam sistem pengamanan citra medis guna memberikan gambaran menyeluruh tentang efisiensi pemanfaatan sumber daya CPU.

7. Ukuran File

Pengukuran ukuran file dilakukan untuk menganalisis perubahan ukuran citra medis pada setiap tahap pemrosesan, yaitu citra asli, citra terenkripsi, dan citra hasil dekripsi. Tujuan dari pengujian ini adalah memastikan bahwa proses enkripsi-dekripsi tidak mengubah ukuran file secara signifikan yang dapat mempengaruhi efisiensi penyimpanan dan kualitas data. Ukuran file yang terjaga atau relatif stabil menunjukkan bahwa proses enkripsi dan dekripsi tidak menyebabkan redundansi data atau distorsi visual pada citra medis hasil dekripsi.