IMPLEMENTASI ALGORITMA *RUBIK'S CUBE* DAN ALGORITMA *RIVEST-SHAMIR-ADLEMAN* (RSA) PADA PENGAMANAN CITRA DIGITAL IRIS MATA

SKRIPSI

OLEH
SITI HABIBATUL MA'RIFAH
NIM. 18610075

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2022

IMPLEMENTASI ALGORITMA *RUBIK'S CUBE* DAN ALGORITMA *RIVEST-SHAMIR-ADLEMAN* (RSA) PADA PENGAMANAN CITRA DIGITAL IRIS MATA

SKRIPSI

Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Matematika (S.Mat)

Oleh
Siti Habibatul Ma'rifah
NIM. 18610075

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2022

IMPLEMENTASI ALGORITMA *RUBIK'S CUBE* DAN ALGORITMA *RIVEST-SHAMIR-ADLEMAN* (RSA) PADA PENGAMANAN CITRA DIGITAL IRIS MATA

SKRIPSI

Oleh
Siti Habibatul Ma'rifah
NIM. 18610075

Telah Disetujui untuk Diuji
Tanggal 16 Juni 2022

Dosen Pembimbing I	Dosen Pembimbing II
Muhammad Khudzaifah, M.Si	Erna Herawati, M.Pd
NIDT. 19900511 20160801 1 057	NIDT. 19760723 20180201 2 222

Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc
NIP. 19741129 200012 2 005

IMPLEMENTASI ALGORITMA *RUBIK'S CUBE* DAN ALGORITMA *RIVEST-SHAMIR-ADLEMAN* (RSA) PADA PENGAMANAN CITRA DIGITAL IRIS MATA

SKRIPSI

Oleh
Siti Habibatul Ma'rifah
NIM. 18610075

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)

Tanggal 21 Juni 2022

Ketua Penguji	: Juhari, M.Si	.....
Angota Penguji 1	: Hisyam Fahmi, M.Kom	.....
Anggota Penguji 2	: Muhammad Khudzaifah, M.Si	.....
Anggota Penguji 3	: Erna Herawati, M.Pd	.....

Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc
NIP. 19741129 200012 2 005

PERNYATAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama	:	Siti Habibatul Ma'rifah
NIM	:	18610075
Program Studi	:	Matematika
Fakultas	:	Sains dan Teknologi
Judul Skripsi	:	Implementasi Algoritma *Rubik's Cube* dan Algoritma *Rivest-Shamir-Adleman* (RSA) Pada Pengamanan Citra Digital Iris Mata

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya sendiri, bukan merupakan pengambilan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan dan pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan atau daftar rujukan. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 13 Juni 2022
Yang membuat pernyataan,

Siti Habibatul Ma'rifah

MOTO DAN PERSEMBAHAN

"Apa yang sudah ditakdirkan untukmu tak akan jadi milik orang lain dan begitu juga sebaliknya."

Skripsi ini penulis persembahkan untuk:

Kedua Orang tua penulis, kakak penulis, keluarga penulis, dan sahabat penulis yang selalu menjadi motivasi dan semangat bagi penulis dalam menuntut ilmu, mengabdi, dan berjuang di tanah rantau.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Segala puji bagi Allah SWT atas rahmat, taufik serta hidayah-Nya, sehingga penulis mampu menyelesaikan penyusunan skripsi dengan judul Implementasi Algoritma *Rubik's Cube* dan Algoritma *Rivest-Shamir-Adleman* (RSA) Pada Pengamanan Citra Digital Iris Mata, sebagai salah satu syarat untuk memperoleh gelar sarjana dalam bidang matematika di Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pada proses penyusunan skripsi ini, penulis banyak mendapat arahan dan bimbingan dari berbagai pihak. Untuk itu ucapan terima kasih yang sebesar-besarnya dan penghargaan yang setinggi-tingginya penulis sampaikan terutama kepada:

1. Prof. Dr. H. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim.
2. Dr. Sri Harini, M.Si, selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim.
3. Dr. Elly Susanti, S.Pd., M.Sc, selaku ketua Program Studi Matematika, Universitas Islam Negeri Maulana Malik Ibrahim.
4. Muhammad Khudzaifah, M.Si, selaku dosen pembimbing I yang telah memberikan arahan dan berbagi ilmunya kepada penulis.
5. Erna Herawati, M.Pd, selaku dosen pembimbing II yang telah banyak memberikan arahan, nasihat, motivasi, dan berbagi banyak pengalaman kepada penulis.

6. Juhari, M. Si, selaku dosen penguji utama yang telah memberikan arahan dan saran yang membangun kepada penulis.
7. Hisyam Fahmi, M.Kom, selaku dosen penguji yang telah banyak memberikan nasihat dan berbagi banyak pengalaman kepada penulis.
8. Mohammad Nafie Jauhari, M.Si, selaku dosen wali yang senantiasa memberikan arahan kepada penulis untuk dapat menyelesaikan studi.
9. Seluruh dosen Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim.
10. Kedua orang tua saya yaitu Bapak Mochamad Mas'ud dan Ibu Nur Khasanah dan seluruh keluarga yang senantiasa memberikan motivasi dan dukungan penuh bagi penulis untuk menempuh studi sampai selesai.
11. Seluruh mahasiswa angkatan 2018 Program Studi Matematika Universitas Islam Negeri Maulana Malik Ibrahim Malang yang senantiasa memberikan support dan energi positif untuk membantu penulis menyelesaikan skripsi ini.

Semoga Allah SWT melimpahkan rahmat dan karunia-Nya kepada kita semua. Akhirnya penulis berharap semoga dengan rahmat dan izin-Nya mudah-mudahan skripsi ini dapat bermanfaat bagi penulis dan pembaca. Amiin.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Malang, 14 Juni 2022

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR SIMBOL

M	:	Lebar citra
N	:	Tinggi citra
(m,n)	:	Posisi pada citra digital
Kr	:	Kunci untuk pergeseran baris pada algoritma *Rubik's Cube*
Kc	:	Kunci untuk pergeseran kolom pada algoritma *Rubik's Cube*
ITER	:	Iterasi maksimum pada algoritma *Rubik's Cube*
$a(i)$	:	Jumlah elemen pada baris
$M\alpha(i)$	:	Nilai modulo 2 dari $a(i)$
$\beta(j)$	:	Jumlah elemen pada kolom
$M\beta(j)$	:	Nilai modulo 2 dari $\beta(j)$
$\oplus$	:	Operasi XOR
p dan q	:	Bilangan prima untuk proses pembangkitan kunci algoritma RSA
n	:	Modulo pembagi untuk algoritma RSA
$\oplus(n)$	:	Nilai *totient euler* (n)
(e, n)	:	Pasangan kunci publik untuk algoritma RSA
(d, n)	:	Pasangan kunci privat untuk algoritma RSA
$l(a,b)$	:	Perbandingan yang digunakan untuk mengukur kemiripan nilai dari luminan
$c(a,b)$	:	Perbandingan nilai kontras yang diperoleh dari perbandingan standar deviasi
$s(a,b)$	:	Perbandingan struktur yang mengukur koefisien korelasi pada 2 citra
μ	:	Nilai luminasi
σ	:	Standar deviasi

DAFTAR LAMPIRAN

ABSTRAK

Ma'rifah, Siti Habibatul. 2022. **Implementasi Algoritma *Rubik's Cube* dan Algoritma *Rivest-Shamir-Adleman* (RSA) pada Pengamanan Citra Digital Iris Mata**. Skripsi. Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing (1): Muhammad Khudzaifah, M.Si, Pembimbing (2): Erna Herawati, M.Pd

Kata Kunci: Citra Digital, Iris Mata, Enkripsi, Dekripsi, Algoritma *Rubik's Cube*, Algoritma *Rivest-Shamir-Adleman* (RSA)

Perkembangan teknologi terutama pada *smartphone* menyebabkan adanya proses autentikasi yang harus dikembangkan. Salah satunya adalah penggunaan iris mata dalam proses autentikasi. Penggunaan iris mata dapat meningkatkan keamanan dalam proses autentikasi dikarenakan struktur iris mata yang unik dan berbeda antara individu satu dengan individu lainnya. Autentikasi membutuhkan proses enkripsi dan dekripsi untuk mengamankan data. Pada penelitian ini menggunakan gabungan dari dua algoritma, yaitu algoritma *Rubik's Cube* dan algoritma *Rivest-Shamir-Adleman* (RSA). Tujuan dari penelitian ini adalah untuk mendapatkan hasil akurasi dan efisiensi waktu yang digunakan pada proses enkripsi dan dekripsi. Pada penelitian ini dilakukan proses enkripsi dengan menggunakan algoritma *Rubik's Cube* kemudian dilanjutkan enkripsi dengan menggunakan algoritma *Rivest-Shamir-Adleman* (RSA). Pada proses dekripsi didapatkan dengan menggunakan algoritma *Rivest-Shamir-Adleman* (RSA) kemudian dilanjutkan dekripsi dengan menggunakan algoritma *Rubik's Cube.* Hasil uji coba yang didapatkan menunjukkan bahwa hasil gambar terenkripsi berbeda dengan citra awal. Dengan hasil rata-rata *Structural Similarity Index Metrics* (SSIM) adalah 0,01 dan rata-rata *Mean Square Error* (MSE) adalah 37620,59. Selanjutnya dilakukan juga evaluasi terkait waktu proses enkripsi dan dekripsi saat menggunakan kunci publik RSA yaitu (197,403), kunci privat RSA (53,403), dan iterasi maksimum algoritma Rubik's Cube adalah 1. Pada proses enkripsi didapatkan rata-rata waktu 0,796 detik dan pada proses dekripsi didapatkan rata-rata waktu 0,652 detik. Pada penelitian ini memiliki kontribusi dalam mendapatkan prosedur baru dalam pengamanan citra digital yang dapat dikembangkan pada penelitian-penelitian selanjutnya.

ABSTRACT

Ma'rifah, Siti Habibatul. 2022. **The Implementation of Rubik's Cube Algorithm and Rivest-Shamir-Adleman (RSA) Algorithm on Iris Digital Image Security**. Thesis. Mathematics Study Program, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Supervisor (1): Muhammad Khudzaifah, M.Si, Supervisor (2): Erna Herawati, M.Pd

Keywords: Digital Image, Eye's Iris, Encryption, Decryption, Rubik's Cube Algorithm, Rivest-Shamir-Adleman (RSA) Algorithm

Technological developments, especially on smartphones, have led to an authentication process that must be developed. One of them is the use of the iris in the authentication process. The use of the iris can increase security in the authentication process because the structure of the iris is unique and differs from individual to individual. Authentication requires encryption and decryption processes to secure data. This research uses a combination of two algorithms, namely the Rubik's Cube algorithm and the Rivest-Shamir-Adleman (RSA) algorithm. The purpose of this study was to obtain the results of accuracy and time efficiency used in the encryption and decryption process. In this study, the encryption process was carried out using the Rubik's Cube algorithm and then followed by encryption using the Rivest-Shamir-Adleman (RSA) algorithm. In the decryption process, it was obtained using the Rivest-Shamir-Adleman (RSA) algorithm, then continued with the decryption using the Rubik's Cube algorithm. The experimental results obtained indicate that the encrypted image results are different from the initial image. With the results the average Structural Similarity Index Metrics (SSIM) is 0,01 and the average Mean Square Error (MSE) is 37620,59. Furthermore, evaluations related to the encryption and decryption process time were also carried out when using the RSA public key (197,403), RSA private key (53,403), and the maximum iteration of the Rubik's Cube algorithm was 1. In the encryption process, the average time was 0,796 seconds and the decryption process obtained an average time of 0,652 seconds. In this study, it has a contribution in obtaining new procedures in securing digital images that can be developed in further studies.

مستخلص البحث

المعرفة ، سيتي حبيبة. ٢٠٢٢. **تطبيق خوارزمية *Rubik's Cube* وخوارزمية *Rivest-Shamir-Adleman* (RSA) على أمن الصورة الرقمية على قزحية العين**. البحث الجامعي. قسم الرياضيات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية,مالانج. المشرف الأول: محمد حديفة, الماجستير، المشرف الثاني: إيرنا هيراواتي, الماجستير

الكلمات المفتاحية: الصورة الرقمية، قزحية البحث، التشفير، فك التشفير، خوارزمية *Rivest-Shamir-Adleman* (RSA), خوارزمية *Rubik's Cube* .

تطور التكنولوجيا خاصة على الهاتف الذكي يؤدي إلى تطوير عملية المصادقة الذي يجب تطورها منها هو استخدام القزحية في عملية المصادقة. ويستطيع أن يؤدي استخدام القزحية إلى ترقية الأمان في عملية المصادقة لأن بنية القزحية فريدة من نوعها وتختلف من فرد لآخر. وتحتاج المصادقة إلى عملية التشفير وفك التشفير لتأمين البيانات. يستخدم هذا البحث مجموعة من خوارزميتين هما خوارزمية مكعب روبيك وخوارزمية *Rivest-Shamir-Adleman* (RSA). الغرض من هذه الدراسة هو الحصول على نتائج الدقة وكفاءة الوقت المستخدمة في عملية التشفير وفك التشفير. في هذه الدراسة ، تم تنفيذ عملية التشفير باستخدام خوارزمية مكعب روبيك ثم تبعها التشفير باستخدام خوارزمية *Rivest-Shamir-Adleman* (RSA). في عملية فك التشفير ، تم الحصول عليها باستخدام خوارزمية *Rivest-Shamir-Adleman* (RSA)، ثم استمرت في فك التشفير باستخدام خوارزمية *Rubik's Cube*. النتائج التجريبية التي تم الحصول عليها تشير إلى أن نتائج الصورة المشفرة مختلفة عن الصورة الأولية. مع النتائج ، بلغ متوسط مقاييس مؤشر التشابه الهيكلي (SSIM) ٠,٠١ ومتوسط الخطأ المربع (MSE) هو ٣٧,٦٢٠,٥٩. علاوة على ذلك ، تم إجراء التقييمات المتعلقة بوقت عملية التشفير وفك التشفير أيضًا عند استخدام المفتاح العام RSA (١٩٧,٤٠٣) ، والمفتاح الخاص RSA (٥٣,٤٠٣) ، وكان الحد الأقصى لتكرار خوارزمية مكعب روبيك هو ١. في عملية التشفير ،كان متوسط الوقت ٠,٧٩٦ ثانية وحصلت عملية فك التشفير على متوسط زمن قدره ٠,٦٥٢ ثانية. في هذه الدراسة ، لها مساهمة في الحصول علىإجراءات جديدة في تأمين الصور الرقمية يمكن تطويرها في مزيد من الدراسات.

BAB I
PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi yang terus berkembang memiliki dampak besar dalam segala aspek kehidupan. *Smartphone* merupakan salah satu bukti nyata dari cepatnya perkembangan teknologi pada saat ini. Terdapat berbagai macam inovasi terhadap teknologi *smartphone* yang semakin meningkatkan fungsionalitas dan efisiensinya. *Smartphone* juga merupakan perangkat keras yang dirancang sedemikian ringkas untuk mengutamakan fleksibilitas pengguna dan sistem komputer yang canggih. Salah satu fitur penting pada *smartphone* yaitu sistem autentikasi. Autentikasi penyimpanan data dan autentikasi kunci layar *smartphone* sering didapati menggunakan pola atau *pattern* dan *password* berupa *text* atau angka yang sudah dianggap kuno atau kurang efektif. Hal tersebut juga memiliki tingkat keamanan yang rendah karena mudah untuk diketahui orang lain (Nader, dkk., 2015).

Berkaitan dengan adanya pandemi COVID-19 yang mengakibatkan segala aktivitas masyarakat disarankan untuk menggunakan masker. Untuk menangani permasalahan tersebut, salah satu bentuk pengembangan pada *smartphone* yang sedang berkembang yaitu sistem autentikasi dengan biometrik. Dengan adanya sistem autentikasi biometrik, keamanan data serta fleksibilitas dalam melakukan proses transaksi melalui *smartphone* dapat lebih terjaga. Disampaikan bahwa penggunaan kata kunci (*password*) dalam keamanan sistem komputer bisa diretas dengan mudah, sehingga dilakukan peningkatan keamanan sistem komputer berupa penggunaan identifikasi menggunakan biometrik (Kristanto, dkk., 2013). Beberapa

autentikasi biometrik yang bisa digunakan pada *smartphone* yaitu autentikasi dengan sidik jari dan iris mata. Penggunan autentikasi biometrik dapat memberikan tingkat keamanan yang lebih tinggi karena memiliki keunikan dan tingkat kesalahan yang rendah (Vatsal & Dwivedi, 2018). Seperti yang kita ketahui bahwa sidik jari dan iris mata adalah bagian unik dari tubuh manusia yang berbeda pada tiap manusia. Dalam mempertimbangkan kondisi kesehatan masyarakat saat ini yang mengharuskan penggunaan masker setiap saat, maka diperlukan inovasi terhadap proses autentikasi *smartphone* dengan metode biometrik yang dapat dilakukan tanpa melepas masker. Salah satu solusi yang dimiliki ialah menggunakan iris mata. Hal tersebut dikarenakan sidik jari dapat menjadi sulit di identifikasi akibat adanya kerusakan tekstur akibat aktivitas luar, sedangkan iris mata merupakan organ dalam yang terlindungi dari kerusakan luar karena memiliki selaput lapisan yang disebut kornea (Vatsal & Dwivedi, 2018). Sehingga penelitian ini menggunakan citra dari iris mata sebagai data yang akan diuji.

Upaya pengamanan data *smartphone* bertujuan untuk melindungi privasi dari pemilik *smartphone*. *Smartphone* bisa dikatakan sebagai buku harian bagi penggunanya, karena dengan *smartphone* manusia bisa melakukan segala kegiatan mulai dari *chatting*, transaksi keuangan, pendidikan, dsb. Dari penjelasan sebelumnya, bisa disimpulkan bahwa *smartphone* memiliki berbagai rekam jejak baik rahasia ataupun tidak rahasia dari penggunanya. Sehingga diperlukan cara untuk menjaga kerahasiaan tersebut. Dalam Al-Qur'an Surat An-Nur /24:27, telah dijelaskan terkait anjuran untuk melindungi privasi atau rahasia orang lain.

يٰٓاَيُّهَا الَّذِيْنَ اٰمَنُوْا لَا تَدْخُلُوْا بُيُوْتًا غَيْرَ بُيُوْتِكُمْ حَتّٰى تَسْتَأْنِسُوْا وَتُسَلِّمُوْا عَلٰٓى اَهْلِهَاۗ ذٰلِكُمْ خَيْرٌ لَّكُمْ لَعَلَّكُمْ تَذَكَّرُوْنَ ۝

"Wahai orang-orang yang beriman! Janganlah kamu memasuki rumah yang bukan

rumahmu sebelum meminta izin dan memberi salam kepada penghuninya. Yang demikian itu lebih baik bagimu, agar kamu (selalu) ingat." (QS. An-Nur/24:27).

Penjelasan ayat tersebut dalam Kitab Tafsir Al-Misbah (Shihab, 2000) yakni, Allah SWT memerintahkan kaum muslimin untuk menghindari tempat dan sebab-sebab yang dapat menimbulkan kecurigaan dan prasangka buruk. Karena itu, di sini diperintahkan untuk meminta izin sebelum masuk ke rumah. Hal tersebut juga memiliki kaitan dengan menghormati batas-batas privasi orang lain. Upaya autentikasi dapat meminimalisir adanya pelanggaran privasi pada setiap individu. Maka diperlukan solusi untuk menjaga keamanan data atau rahasia yang berada dalam masing-masing *smartphone* pengguna.

Adanya proses autentikasi tentunya memerlukan tingkat keamanan data yang tinggi. Hal tersebut memerlukan ilmu kriptografi yang berfungsi dalam proses enkripsi dan dekripsi data. Pada penelitian sebelumnya oleh Julus (2022) dilakukan penelitian dengan menggunakan algoritma *Rubik's Cube* untuk melakukan proses pengacakan dan penggunaan kunci dari algoritma Rivest-Shamir-Adleman (RSA) yang diterapkan pada operasi XOR. Kemudian Belkaid (2015) melakukan penelitian dengan menggunakan algoritma enkripsi *hybrid* dengan menggunakan algoritma *Advanced Encryption Standard* (AES) dan Algoritma *Rivest-Shamir-Adleman* (RSA) untuk mengenkripsi kunci yang dimiliki algoritma AES. Algoritma *Rivest-Shamir-Adleman* (RSA) sering digunakan karena dapat memberikan tingkat keamanan yang tinggi berdasarkan proses pembangkitan kunci yang sulit dengan dasar perhitungan modulo dan pemfaktoran bilangan prima yang besar (Belkaid, 2015). Kemudian Loukhaoukha, dkk. (2012) melakukan penelitian untuk menerapkan proses enkripsi dan dekripsi pada gambar dengan menggunakan algoritma *Rubik's Cube* dan menjelaskan bahwa Algoritma *Rubik's Cube*

diperlukan sebagai algoritma yang berfungsi untuk melakukan pengacakan piksel pada citra yang akan digunakan. Algoritma *Rubik's Cube* menunjukkan skema enkripsi dan kemampuan pengacakan yang baik dan dapat menahan serangan baik statistik maupun diferensial.

Berdasarkan pada penelitian sebelumnya, didapatkan bahwa terdapat penelitian yang menggunakan algoritma *Rubik's Cube* dan algoritma *Rivest-Shamir-Adleman* (RSA) dengan menerapkan algoritma enkripsi *hybrid*. Oleh karena itu, penelitian ini dilakukan untuk mendapatkan inovasi baru dengan menerapkan gabungan dua algoritma tersebut untuk melakukan proses enkripsi dan dekripsi tanpa menerapkan algoritma enkripsi *hybrid* seperti yang telah dilakukan pada penelitian sebelumnya. Penelitian ini perlu dilakukan untuk mengetahui bagaimana tingkat akurasi yang diperoleh berdasarkan pada citra awal dengan citra terdekripsi yang di proses menggunakan algoritma *Rubik's Cube* dan algoritma *Rivest-Shamir-Adleman* (RSA) untuk mengetahui bahwa algoritma yang digunakan dapat menghasilkan hasil enkripsi dan dekripsi yang baik. Selain itu, penelitian ini perlu dilakukan untuk mengetahui efisiensi waktu yang digunakan untuk menjalankan proses enkripsi dan dekripsi dengan gabungan algoritma *Rubik's Cube* dan algoritma *Rivest-Shamir-Adleman* (RSA) pada citra iris mata karena pada penelitian sebelumnya terkait penggunaan algoritma enkripsi *hybrid* yang berdasar pada algoritma *Rubik's Cube* dan algoritma *Rivest-Shamir-Adleman* (RSA) belum didapatkan terkait waktu yang diperlukan dalam melakukan proses enkripsi dan dekripsi.

1.2 Rumusan Masalah

Dari latar belakang yang tertulis sebelumnya, maka hal yang dapat diidentifikasi pada penelitian ini yaitu bagaimana hasil akurasi dan efisiensi waktu penggunaan algoritma *Rubik's Cube* yang dikombinasikan dengan algoritma *Rivest-Shamir-Adleman* (RSA) dalam pemanfaatan enkripsi dan dekripsi citra digital iris mata?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah mengetahui hasil akurasi dan efisiensi waktu pada penerapan algoritma *Rubik's Cube* yang dikombinasikan dengan algoritma *Rivest-Shamir-Adleman* (RSA) dalam pemanfaatan enkripsi dan dekripsi citra digital iris mata.

1.4 Manfaat Penelitian

Manfaat yang dapat diperoleh dari penelitian ini adalah sebagai berikut:

1. Menambah pengetahuan dan keilmuan terkait algoritma kriptografi untuk proses pengamanan citra digital.
2. Dapat menjadi rujukan bagi penelitian-penelitian yang akan dikembangkan selanjutnya terutama pada bidang pengamanan citra digital.
3. Dengan program komputer, didapatkan prosedur baru dalam pengamanan citra digital.

BAB II
KAJIAN TEORI

2.1 Teori Pendukung

2.1.1 Citra Digital

Menurut kamus Webster, citra merupakan hasil imitasi atau sebuah representasi dari suatu objek. Menurut KBBI, citra berarti gambar atau rupa (Sulistiyanti, dkk., 2016). Citra dibagi menjadi 2, yaitu citra kontinu yang merupakan citra yang mendapatkan sinyal analog, misalnya kamera analog dan mata manusia dan citra diskrit yang merupakan hasil dari proses digitalisasi terhadap citra kontinu yang biasa disebut juga sebagai citra digital. Citra digital atau istilah lain dari gambar digital menjadi salah satu bagian penting dari segi informasi visual karena memiliki karakteristik yang tidak dipunya oleh data teks (Sutoyo, dkk., 2009).

Pada penggunaan citra digital, tentu dibutuhkan teknik pengolahan gambar sehingga gambar yang diperoleh dapat sesuai dengan syarat-syarat yang diperlukan selama proses penelitian. Menurut KBBI, pengolahan ialah suatu proses yang mengupayakan sesuatu untuk menjadi lebih baik atau menjadi yang lain sesuai dengan kebutuhan. Sehingga dapat disimpulkan bahwa pengolahan citra merupakan pemrosesan citra untuk menghasilkan *output* berupa citra yang dikehendaki (Sulistiyanti, dkk., 2016). Proses merepresentasikan citra dari fungsi kontinu menjadi nilai-nilai diskrit disebut sebagai proses digitalisasi. Setelah itu, akan terbentuk citra digital dan memiliki dimensi ukuran yang dinyatakan dengan lebar (M) x tinggi (N) (Cahyanti, dkk., 2016).

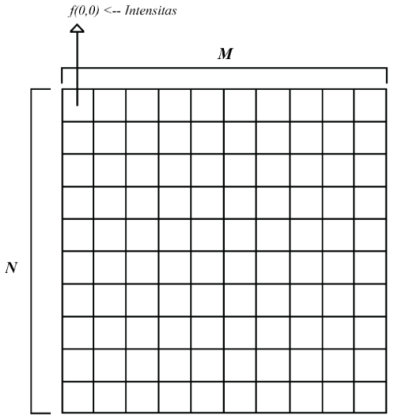
Terdapat sebanyak *M* x *N* posisi diskrit dan setiap posisi *(m,n)* memiliki unsur gambar (*picture element*) yang biasa disebut *pixel*. Berikut merupakan notasi citra digital *M* x *N* dari gambar di atas:

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

Ruas kanan pada matriks di atas merupakan definisi dari citra digital. Berikut merupakan notasi matriks lain yang menotasikan citra digital beserta elemennya.

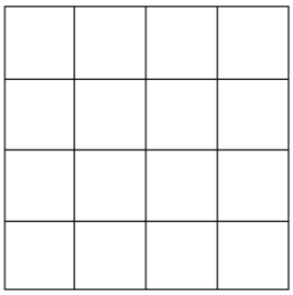
$$\mathbf{A} = \begin{bmatrix} \alpha_{0,0} & \alpha_{0,1} & \cdots & \alpha_{0,N-1} \\ \alpha_{1,0} & \alpha_{1,1} & \cdots & \alpha_{1,N-1} \\ \cdots & \cdots & \cdots & \cdots \\ \alpha_{M-1,0} & \alpha_{M-1,1} & \cdots & \alpha_{M-1,N-1} \end{bmatrix}$$

Notasi di atas menunjukkan bahwa, $\alpha_{j,i} = f(x = i, y = j) = f(i,j)$. Sehingga matriks $f(x,y)$ dan matriks A merupakan matriks yang sama. Dengan indeks baris (i) dan indeks kolom (j) menyatakan koordinat titik dari citra digital yang dapat direpresentasikan pada Gambar 2.1



Gambar 2.1 Koordinat Citra Digital

Untuk mempermudah pemahaman terkait matriks citra digital, berikut kumpulan persegi untuk merepresentasikan setiap *pixel* yang ada pada citra digital seperti pada Gambar 2.2.



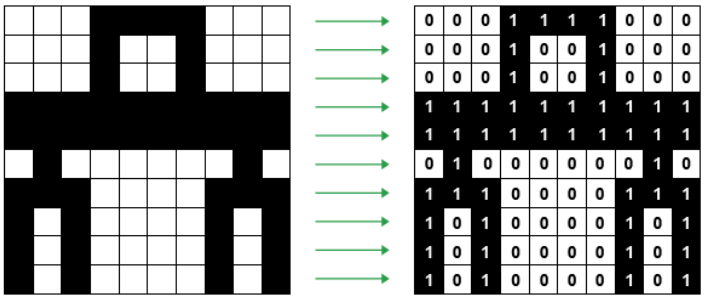
Gambar 2.2 Koordinat Citra Digital Berukuran 4 X 4 dalam Visualisasi

Citra digital memiliki dua bagian utama yaitu koordinat dan tingkat keabuan atau *gray-level*. Gambar di atas menunjukkan bahwa setiap titik koordinat memiliki nilai-nilai yang akan menentukan warna pada tiap *pixel*. Jumlah baris atau kolom *pixel* menyatakan resolusi dari sebuah citra. Resolusi citra berfungsi untuk menunjukkan tingkat kedetailan suatu citra. Resolusi citra dirumuskan berbanding lurus dengan detail citra sehingga dapat disimpulkan bahwa semakin tinggi resolusi dari suatu citra maka semakin detail pula citra tersebut (Sulistiyanti, dkk., 2016).

Sedangkan tingkat keabuhan pada sebuah citra berhubungan dengan *binary digit* atau bit yang terdiri atas bilangan 0 dan 1 dengan semua warna diperoleh dari tiga warna dasar yaitu merah, hijau, dan biru. Nilai *pixel* dari suatu citra merupakan bilang biner dengan panjang k. Sedangkan nilai k merupakan representasi dari kedalaman bit atau *bit depth* dari sebuah citra. Jenis citra akan mempengaruhi susunan dari skala bitnya. Pada umumnya citra digital dibagi menjadi 3, yaitu *binary image*, *color image*, dan *black and white image* (Kusumanto & Tompunu, 2011). Berikut merupakan penggambaran matriks citra digital dengan masing- masing jenis citra.

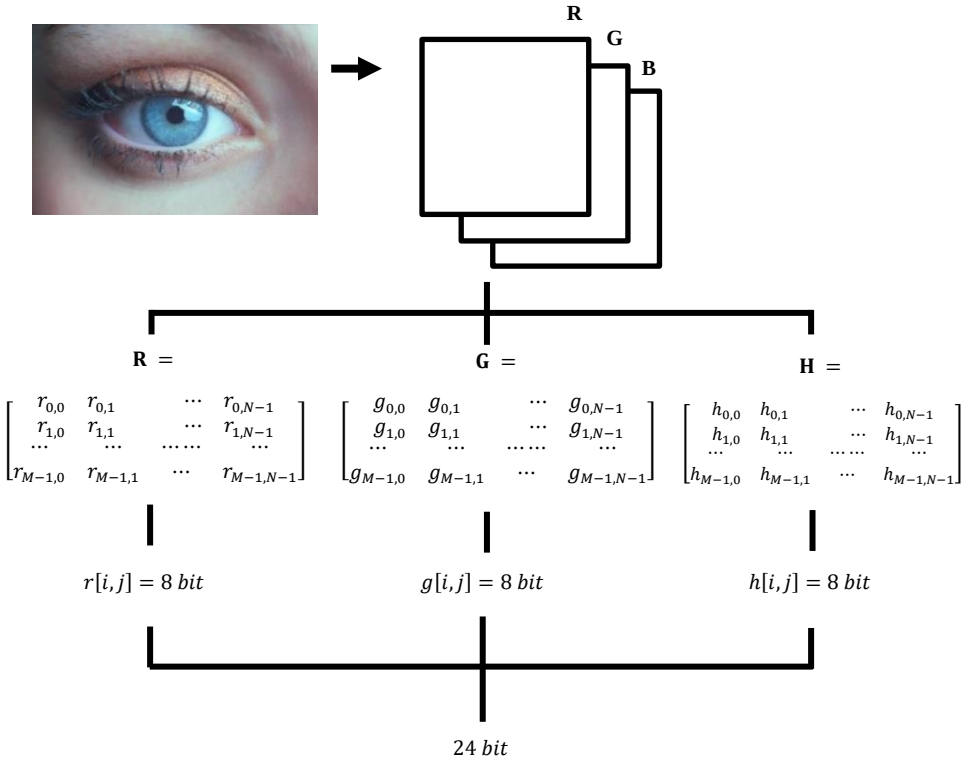
Pada gambar 2.3 ditunjukkan bahwa matriks di *binary image* hanya terdiri dari angka 0 dan 1. Dengan demikian, warna yang diperoleh hanya hitam dan

putih. *Binary image* seringkali digunakan pada *image processing* yang membutuhkan proses *edge detection* guna memperoleh bentuk objek yang lebih jelas.



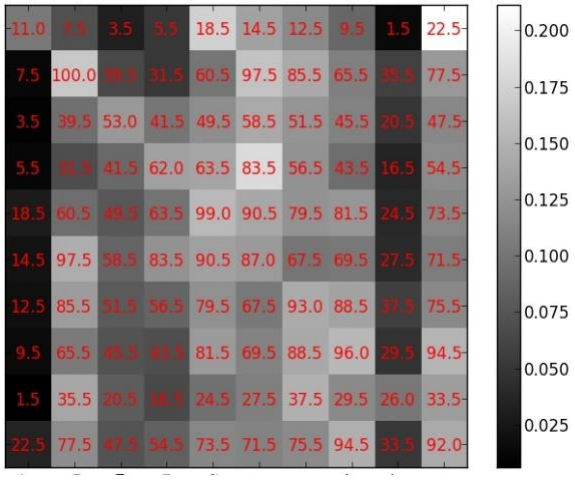
Gambar 2.3 Matriks dalam Citra Digital *Binary Image*

Selanjutnya pada Gambar 2.4 menunjukkan bahwa pada *color image* memiliki kedalaman bit sebanyak 24bit dengan rentang warna [0,255] dan terdapat 3 kanal/*layer*. Sehingga terdapat 255^3 atau sekitar 16 ribu varian warna yang dapat ditampilkan pada citra tersebut.



Gambar 2.4 Matriks dalam Citra *Color Image*

Gambar 2.5 menunjukkan bahwa pada *grayscale image* memiliki intensitas 0 sampai 255 dengan 1 kanal/*layer*. Citra grayscale merupakan citra dengan nilai R=G=B. Sehingga warna yang dihasilkan hanya warna-warna yang didapatkan oleh gradasi antara putih sampai hitam.



Gambar 2.5 Matriks dalam Citra *Grayscale Image*

2.1.2 Kriptografi

Kriptografi atau *Cryptography* berasal dari bahasa Yunani. Terdiri dari dua kata dasar yaitu *kryptos* yang berarti rahasia atau tersembunyi dan *graphein* yang berarti menulis. Secara umum kriptografi diartikan sebagai proses penyampaian pesan secara rahasia atau tersembunyi. Ginting, dkk., (2015) menyampaikan bahwa kriptografi diartikan sebagai bidang ilmu yang mempelajari berbagai macam teknik matematis dalam melakukan proses pengamanan informasi baik berupa data ataupun gambar dalam aspek kerahasiaan, integritas, keabsahan, dan autentikasi. Arius (2008) mengatakan bahwa berdasarkan catatan sejarah, kriptografi diperkenalkan oleh orang mesir

lewat *hieroglyph* sejak 4000 tahun lalu. Jenis tulisan itu bukan bentuk *standart* yang digunakan untuk menulis pesan.

Pada dasarnya, kriptografi digunakan untuk menjaga proses pengiriman pesan kepada penerima secara aman tanpa mendapatkan gangguan dari pihak lain. Proses tersebut dilakukan dengan cara melakukan pengacakan suatu pesan agar tidak diketahui makna dari pesan sebenarnya. Kemudian, penerima pesan menjalankan proses berdasarkan algoritma yang sama guna mendapatkan kembali makna dari pesan sebenarnya.

Berikut beberapa prinsip - prinsip dasar kriptografi, yaitu (Julus, 2022):

1. *Data integrity*

   Data integrity atau integritas data merupakan fitur dimana akurasi data dapat dijamin karena perubahan data hanya bisa dilakukan atas sepengetahuan kedua belah pihak.

2. *Confidentially*

   Confidentially merupakan fitur perlindungan informasi dari pihak luar selain pengirim dan penerima sehingga pihak luar tidak dapat menjalankan sistem atau terkendala dalam hak akses sistem.

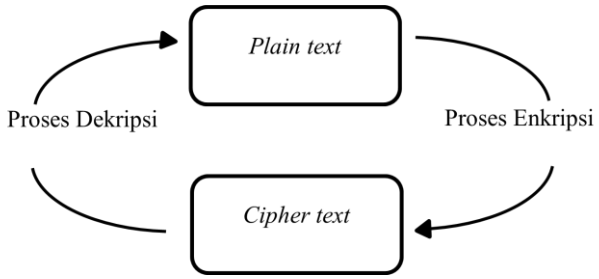
3. *Authentication*

   Authentication atau autentikasi merupakan fitur dimana pihak pengirim dan penerima saling mengetahui asal identitas data atau informasi berasal.

4. *Non-repudiation*

   Non-repudiation merupakan fitur dimana kedua belah pihak tidak dapat menyangkal tujuan untuk melakukan perubahan atau menciptakan informasi.

Terdapat dua proses dasar yang ada pada ilmu kriptografi, yaitu enkripsi dan dekripsi yang digambarkan pada Gambar 2.6. Proses enkripsi merupakan proses pengacakan *plaintext* (naskah asli) menjadi chipertext (naskah acak) yang sulit untuk dibaca apabila tidak memiliki kunci yang sesuai untuk melakukan proses dekripsi (Kromodimoeljo, 2009). Sedangkan proses dekripsi proses untuk menemukan *plaintext* (naskah asli) dari proses enkripsi yang telah dilakukan sebelumnya dengan menggunakan kunci dekripsi. Proses dekripsi dilakukan dengan runtutan proses kebalikan dari proses enkripsinya.



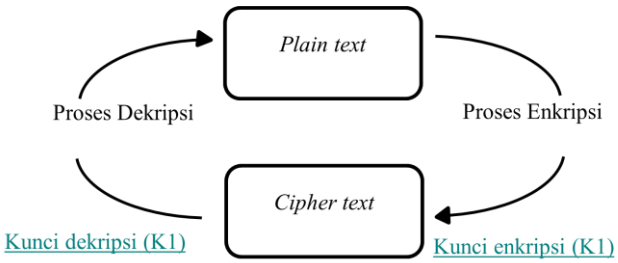
Gambar 2.6 Proses Enkripsi dan Dekripsi

Beberapa aspek yang memiliki peran penting pada proses enkripsi dan dekripsi data yaitu algoritma untuk mendapatkan sistem yang aman dan kunci untuk menjaga kerahasiaan dari *ciphertext* yang telah dibuat agar tidak mudah dilakukan pembobolan atau adanya kebocoran data.

Secara umum algortima kriptografi merupakan metode atau susunan langkah-langkah yang sistematis untuk melakukan proses enkripsi dan dekripsi. Jenis algoritma yang ada pada sistem kriptografi yaitu algoritma simetris dan algoritma asimetris.

Gambar 2.7 merupakan representasi algoritma simetris. Dengan pengertian algoritma simetris (*symmetric algorithm*) ialah salah satu algoritma

yang memiliki kunci enkripsi dan kunci dekripsi yang sama sehingga algoritma ini sering disebut sebagai *single-key algorithm.*



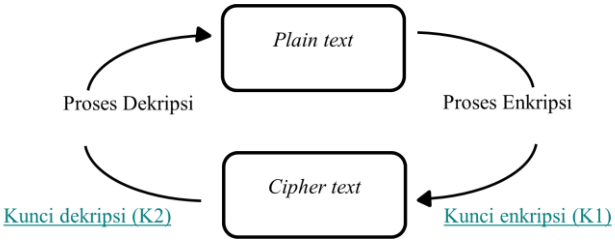
Gambar 2.7 Proses Enkripsi dan Dekripsi Algoritma Simetris

(Widiasari, 2014) mengatakan bahwa terdapat kelebihan dan kekurangan yang dimiliki algoritma simetris, yaitu:

Tabel 2.1 Kelebihan dan Kekurangan Algoritma Simetris

No.	Keterangan	Rincian
1.	Kelebihan	Kecepatan operasi sistem yang lebih tinggi daripada algoritma asimetrik.
		Dapat digunakan pada sistem *real-time* karena kecepatan operasi yang tinggi.
2.	Kekurangan	Kesulitan manajemen kunci karena kebutuhan kunci yang berbeda pada tiap pengiriman pesan.

Sedangkan algoritma asimetris (*asymmetric algorithm*) merupakan algoritma yang berkebalikan dengan algoritma simetri. Gambar 2.8 merupakan representasi algoritma asimetris. Pada algoritma ini digunakan kunci yang berbeda antara proses enkripsi dan dekripsi. Pada proses enkripsi digunakan kunci publik (*public-key*) yang memang disebarkan secara umum dan pada proses dekripsi digunakan kunci privat (*private-key)* yang disimpan secara rahasia oleh pengguna. Pada algoritma asimetris, kunci privat tetap sukar ditemukan walaupun kunci publik sudah diketahui.



Gambar 2.8 Proses Enkripsi dan Dekripsi Algoritma Asimetris

Pada tabel 2.2 ditunjukkan terkait beberapa kelebihan dan kekurangan yang dimiliki algoritma asimetris (Widiasari, 2014).

Tabel 2.2 Kelebihan dan Kekurangan Algoritma Asimetris

No.	Keterangan	Rincian
1.	Kelebihan	Keamanan pada distribusi kunci lebih aman dan lebih baik
		Jumlah kunci lebih sedikit sehingga manajemen kunci dapat lebih baik.
2.	Kekurangan	Kecepatan operasi lebih rendah dibandingkan algoritma asimetris.
		Kunci yang digunakan lebih panjang daripada algoritma asimetris pada tingkat keamanan sama.

2.1.3 Algoritma *Rubik's Cube*

Rubik's Cube merupakan permainan mekanik tiga dimensi yang ditemukan oleh seorang professor arsitektur dan pemahat bernama Erno Rubik pada tahun 1974 (Haekal Al-Fadillah & Djiwadukusumah, 2019). Pada umumnya, sebuah rubik terdiri dari 26 kubus kecil dengan ukuran dimensi 3 x 3 x 3 dan memiliki 6 permukaan dengan 6 warna yang berbeda-beda. Algoritma *Rubik's Cube* menggunakan prinsip dari permainan kubus rubik dengan membuat pergeseran-pergeseran terhadap matriks dari citra digital dengan dua kunci yaitu kunci Kr dan kunci Kc. Kunci Kr berfungsi untuk melakukan pergeseran baris matriks dari kiri ke kanan dan kunci Kc berfungsi untuk melakukan pergeseran

kolom matriks ke atas atau ke bawah. Digunakan pula iterasi maksimum untuk menentukan jumlah iterasi pengacakan citra yang akan dilakukan. Dua kunci tersebut akan dibangkitkan dengan menggunakan operator Boolean Exclusive-OR (Haekal Al-Fadillah & Djiwadukusumah, 2019). Operator XOR dapat disimbolkan dengan $\oplus$ yang berfungsi untuk melakukan operasi pembandingan suatu nilai. Dengan nilai input yang sama akan mendapatkan 0 dan apabila nilai input yang berbeda akan mendapatkan 1.

Beberapa parameter yang digunakan pada algoritma *Rubik's Cube* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Parameter Pada Algoritma *Rubik's Cube*

No.	Nama Parameter	Keterangan
1.	I_0 (Nilai matriks dari citra awal)	Rahasia
2.	*M* (Lebar citra)	Tidak rahasia
3.	*N* (Tinggi citra)	Tidak rahasia
4.	*Kr* (Kunci untuk pergeseran baris)	Rahasia
5.	*Kc* (Kunci untuk pergeseran kolom)	Rahasia
6.	ITER (Iterasi maksimum)	Rahasia

Penggunaan algoritma *Rubik's Cube* pada fungsi enkripsi dilakukan dengan urutan sebagai berikut (Loukhaoukha, dkk., 2012)

1. Inisialisasi secara acak pada bilangan kunci untuk membentuk vektor *Kr* dan *Kc* dengan panjang *N* dan *M*. Elemen *Kr(i)* dan *Kc(j)* akan mengambil nilai pada himpunan $\mathrm{A} = \{ 0, 1, 2, \dots , 2\alpha - 1\}$. Nilai *Kr* dan *Kc* tidak harus memiliki nilai yang konstan.
2. Inisialisasi jumlah dari iterasi dan iterasi maksimum dan memberikan nilai iterasi dengan nilai 0.

3. Inkremen setiap iterasi dengan satu sehingga menghasilkan nilai:

   iterasi = iterasi + 1.

4. Setiap baris *i* pada matriks I_0 akan diproses dengan urutan sebagai berikut:

   a. Melakukan perhitungan jumlah elemen pada baris *i* yang akan dinotasikan sebagai $a(i)$, dengan persamaan:

$$a(i) = \sum_{j=0}^{M-1} I_0\,(i,j), i = 0, 1, \dots, N-1 \tag{2.1}$$

   b. Menghitung nilai $M\alpha(i)$dengan fungsi:

$$M\alpha(i) = a(i) mod\ 2 \tag{2.2}$$

   c. Baris i bergerak ke kiri, kanan, atau melingkar bergeser oleh posisi *Kr(i)* yang diartikan dengan piksel citra akan digeser oleh posisi *Kr(i)* kearah kanan atau kiri, dan piksel pertama akan berpindah ke piksel terakhir.

      Jika $M\alpha(i) = 0$ maka rotasi ke kanan dan selain itu rotasi ke kiri.

5. Setiap kolom *j* pada matriks I_0 akan diproses dengan urutan sebagai berikut:

   a. Melakukan perhitungan jumlah elemen pada kolom *j* yang akan dinotasikan sebagai $\beta(j)$, dengan persamaan:

$$\beta(j) = \sum_{i=0}^{N-1} I_0\,(j,i), j = 0, 1, \dots, M-1 \tag{2.3}$$

   b. Menghitung nilai $M\beta(j)$ dengan persamaan:

$$M\beta(j) = \beta(j) mod\ 2 \tag{2.4}$$

   c. Kolom j bergerak ke atas, bawah, atau melingkar bergeser oleh posisi *Kc(j)* yang diartikan dengan piksel citra akan digeser oleh posisi *Kc(j)* kearah atas atau bawah. Sehingga diperoleh:

      Jika $M\beta(j) = 0$ maka rotasi keatas dan selain itu rotasi kebawah.

6. Pada langkah d dan e akan didapatkan matriks gambar acak yang disebut I_{scr}. Kemudian pada matriks I_{scr} dilakukan operasi XOR dengan vektor *Kc* dengan persamaan:

$$I_1(2i-1,j) = I_{scr}(2i-1,j) \oplus Kc(j) \quad (2.5)$$

$$I1(2i,j) = I_{scr}(2i,j) \oplus rot180(Kc(j)) \quad (2.6)$$

Dimana $\oplus$ dan $rot180(Kc)$ merupakan proses operasi XOR dan pergeseran vektor Kc dari atas kebawah atau sebaliknya.

7. Pada matriks $I1$ dilakukan operasi XOR dengan Kr dengan persamaan:

$$I_{enc}(i,2j-1) = I1(i,2j-1) \oplus Kr(i) \quad (2.7)$$

$$I_{enc}(i,2j) = I1(i,2j) \oplus rot180Kr(i) \quad (2.8)$$

Dimana $rot180(Kc)$ merupakan proses pergeseran dari kiri ke kanan pada vektor Kr.

8. Jika ITER = Imax maka matriks citra terenkripsi akan tersimpan pada I_{ENC} dan jika tidak sama maka akan terjadi pengulangan kembali dari langkah ketiga.

Proses dekripsi pada algoritma *Rubik's Cube* dapat dilakukan sesuai dengan kebalikan dari proses enkripsinya.

2.1.4 Algoritma *Rivest-Shamir-Adleman* (RSA)

RSA kriptosistem merupakan algoritma kriptografi asimetris yang ditemukan pada tahun 1977 oleh tiga penemu yaitu Ron Rivest, Adi Shamir, Len Adleman. Penamaan dari algoritma ini juga diambil dari nama ketiga penemunya, sehingga dapat disebut sebagai algoritma *Rivest-Shamir-Adleman* (RSA). Dalam penggunaannya, algoritma RSA termasuk dalam algoritma kriptografi kunci publik. Sebagai algoritma kriptografi kunci publik, algoritma RSA memiliki dua

kunci yaitu kunci publik dan kunci privat. Kunci publik merupakan kunci yang digunakan dalam melakukan proses enkripsi. Dalam hal ini, kunci publik tidak dirahasiakan kepada umum. Kunci privat merupakan kunci yang digunakan dalam proses dekripsi dan dirahasiakan dari pihak luar selain pengirim dan penerima.

Pada algoritma RSA dilakukan tiga proses utama yaitu pembangkitan kunci, proses enkripsi, dan proses dekripsi. Dasar proses enkripsi dan dekripsi algoritma RSA adalah aritmetika modulo dan konsep bilangan prima (Ginting, dkk., 2015). Kunci enkripsi dan kunci dekripsi algoritma RSA merupakan bilangan bulat. Kunci enkripsi disebut juga sebagai kunci umum (*public key*), sedangkan kunci dekripsi disebut sebagai kunci rahasia (*private key*). Untuk mendapatkan *private key*, perlu dilakukan pemfaktorkan bilangan bulat menjadi faktor-faktor primanya. Hal tersebut bukan hal yang mudah untuk dilakukan karena belum ditemukan algoritma yang efisien untuk melakukan pemfaktoran. Pohon faktor merupakan salah satu cara untuk mendapatkan hasil dari proses pemfaktoran. Namun pada algoritma RSA digunakan bilangan prima yang besar sehingga proses pemfaktoran akan semakin sulit sulit dan membutuhkan waktu yang lama. Hal tersebut menyebabkan semakin kuat pula algoritma RSA yang dimiliki. Oleh karena itu, RSA dianggap aman dan dapat membuat proses enkripsi dan dekripsi yang sulit untuk di pecahkan (Belkaid, 2015).

Beberapa parameter yang digunakan pada algoritma RSA dapat dilihat pada tabel 2.4.

Tabel 2.4 Parameter Pada Algoritma RSA

No.	Nama Parameter	Keterangan
1.	*p* dan *q* bilangan prima	Rahasia
2.	$n = p \cdot q$ (modulo pembagi)	Tidak rahasia
3.	$\phi(n) = (p-1)(q-1)$	Rahasia
4.	*e* (kunci enkripsi)	Tidak rahasia
5.	*d* (kunci dekripsi)	Rahasia
6.	X (plainteks)	Rahasia
7.	Y (chiperteks)	Tidak rahasia

Proses yang dilakukan untuk melakukan pembangkitan kunci RSA ialah sebagai berikut (Khairil Azhar & Yuliany, 2019):

1. Tentukan nilai *p* dan *q*.

   p dan *q* adalah bilangan prima dengan $p \neq q$.

2. Menghitung nilai modulus (*n*).

   Dengan nilai $n = p.q$

3. Menentukan nilai $\phi(n)$ atau nilai *totient euler (n).*

   Dengan $\phi(n)$ merupakan banyaknya bilangan bulat positif yang lebih kecil atau sama dengan n dan relative prima terhadap n.

   $\phi(n)$ dapat ditentukan dengan rumus:

   $$\phi(n) = (p-1)(q-1) \tag{2.9}$$

4. Menentukan nilai bilangan bulat *e* (kunci publik).

   Dengan nilai *e* antara satu dan $\phi(n)(1<e<\phi(n))$ yang tidak memiliki faktor pembagi dari $\phi(n)$ sehingga gcd (e, $\phi(n)$) = 1. Dengan demikian dapat dikatakan bahwa *e* relatif prima terhadap $\phi(n)$.

Greatest Common Divisor (gcd) dari dua bilangan merupakan irisan dari himpunan faktor bilangan prima dari kedua bilangan tersebut.

5. Menentukan nilai bilangan bulat *d*.

   Dengan nilai *d* adalah $1<d<\bigoplus(n)$, sehingga $e.d = 1(mod \bigoplus(n))$. Menurut persamaan *Chinese Remainder Theorem* (CRT) yang menyatakan bahwa $a.b(mod\ d)$ ekuivalen dengan $a=b+k.m$, atau dapat dinyatakan dengan persamaan:

   $$e.d = 1\ (mod\ \bigoplus(n)) \tag{2.10}$$

   Dengan percobaan nilai $d = 0, 1, 2, 3, \ldots$ sehingga memenuhi persaman tersebut.

6. Terbentuk kunci publik dan privat berdasarkan proses pembangkitan kunci di atas dengan hasil sebagai berikut:

   Kunci publik dengan pasangan (e, n)

   Kunci privat dengan pasangan (d, n)

Penggunaan algoritma RSA pada fungsi enkripsi dan dekripsi dilakukan dengan urutan sebagai berikut:

1. Ambil pasangan kunci publik pada proses pembangkitan kunci sebelumnya yaitu pasangan (e, n).
2. Pada proses enkripsi menggunakan fungsi di bawah ini untuk menghasilkan matriks dari gambar terenkripsi.

   $$X = Y^e mod\ n \tag{2.11}$$

3. Ambil pasangan kunci privat pada proses pembangkitan kunci sebelumnya yaitu pasangan (d, n).

4. Pada proses dekripsi menggunakan fungsi di bawah ini untuk menghasilkan matriks awal dari gambar yang telah terenkripsi.

$$Y = X^d mod\ n \tag{2.12}$$

2.1.5 *Structural Similarity Index Metrics* (SSIM)

Structural Similarity Index Metrics atau SSIM adalah salah satu metode yang digunakan untuk mengetahui dan mengukur tingkat kemiripan dari 2 citra gambar (Sumarna, dkk., 2020). Algoritma *Structural Similarity Index Metrics* dilakukan dengan cara membandingkan fitur struktural. Dengan perbandingan lurus antara kesamaan structural dengan kualitas gambar. Sehingga dapat disimpulkan bahwa semakin tinggi kesamaan, maka semakin tinggi pula kualitas citra, begitu juga sebaliknya. Algoritma *Structural Similarity Index Metrics* memiliki 3 pembanding utama, yaitu *luminance distortion*, *contrass distortion*, dan *correlation*. Fitur – fitur tersebut dibandingkan dan menghasilkan tiga hasil perbandingan.

(Sumarna, dkk., 2020) menjelaskan bahwa persamaan *Structural Similarity Index Metrics* dapat dituliskan dengan persamaan berikut:

$$SSIM(a.b) = l(a.b)c(a.b)s(a.b) \tag{2.13}$$

Dengan masing-masing pembanding/fitur sebagai berikut :

$$\textit{Luminance}: \quad l(a,b) = \frac{2\mu_a\mu_b + C_1}{u_x^2 + u_y^2 + C_1} \tag{2.14}$$

$$\textit{Contrast}: \quad c(a,b) = \frac{2\sigma_a\sigma_b + C_2}{\sigma_a^2 + \sigma_b^2 + C_2} \tag{2.14}$$

$$\textit{Structure}: \quad s(a,b) = \frac{\sigma_{ab} + C_3}{\sigma_a\sigma_b + C_3} \tag{2.15}$$

Dimana C_1, C_2, C_3 merupakan konstanta untuk menghindari *error* akibat penyebut = 0. Persamaan $l(a,b)$ merupakan perbandingan yang digunakan untuk mengukur kemiripan nilai dari luminan (μ) kedua citra yang diuji. Dengan nilai maksimum dari $l(a,b)$ adalah 1. Nilai maksimum akan terpenuhi apabila nilai $\mu_a = \mu_b$.

Persamaan $c(a,b)$ merupakan perbandingan nilai kontras yang diperoleh dari perbandingan standar deviasi (σ) citra yang diuji. Sama dengan persamaan $l(a,b)$ dimana nilai maksimum yang dapat diperoleh adalah 1, dengan syarat $\sigma_a = \sigma_b$.

Persamaan $s(a,b)$ merupakan perbandingan struktur yang mengukur koefisien korelasi pada 2 citra yang diuji. Dengan σ_{ab} adalah nilai kovarian antara *a* dan *b*.

Algoritma *Structural Similarity Index Metrics* menghasilkan nilai dari rentang 0 sampai dengan 1(Sara, dkk., 2019). Dimana nilai "0" menunjukkan bahwa kedua citra tidak berkorelasi ataupun tidak sama. Sedangkan nilai "1" menunjukkan bahwa kedua gambar yang diuji adalah gambar yang mirip atau sama persis. Sehingga dapat disimpulkan bahwa semakin besar nilai *Structural Similarity Index Metrics* maka semakin mirip citra yang diuji, dan begitu juga sebaliknya. Algoritma ini berhubungan dengan topik yang dikaji karena dapat mengukur keakuratan gambar yang telah didekripsi.

2.1.6 *Mean Square Error* (MSE)

Mean Square Error merupakan salah satu perhitungan untuk mengetahui besarnya error yang dimiliki pada proses penyisipan (Male, dkk., 2012). Dalam proses kriptografi, MSE bisa berfungsi untuk mengetahui apakah proses enkripsi

dan dekripsi memiliki keakuratan yang baik. Hal tersebut dapat diperoleh dengan membandingkan nilai *pixel* dari citra awal dan citra yang telah melalui proses dekripsi. Perhitungan matematika dari *Mean Square Error* dapat dijabarkan dengan persamaan berikut:

$$MSE = \frac{1}{m\ x\ n}\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}(A_{i,j} - B_{i,j}) \tag{2.16}$$

Tabel 2.5 merupakan beberapa parameter yang digunakan untuk proses mencari nilai *Mean Square Error* (MSE).

Tabel 2.5 Parameter untuk Mencari Nilai *Mean Square Error* (MSE)

No.	Nama Parameter	Keterangan
1.	MSE	Nilai *Mean Square Error* Citra
2.	m	Panjang citra terdekripsi
3.	n	Lebar citra terdekripsi
4.	$A_{i,j}$	Representasi satu *pixel* dari *plain image*
5.	$B_{i,j}$	Representasi satu *pixel* dari citra terdekripsi
6.	m* n	Dimensi citra

2.1.7 Autentikasi *Smartphone*

Autentikasi merupakan suatu proses untuk mendapatkan pembuktian (validasi) terhadap sesuatu. Pada *smartphone*, proses autentikasi digunakan untuk mendapatkan akses kedalam bagian atau aplikasi yang dirasa memiliki tingkat kerahasiaan tertentu. Sehingga autentikasi dapat disebut sebagai salah satu bagian pada proses pengamanan data. Dalam implementasinya, autentikasi tidak hanya digunakan saat seseorang ingin membuka *smartphone*, namun juga dapat digunakan untuk proses verifikasi akun *bank*, verifikasi transaksi pembelian, verifikasi *cloud* berupa *google*, dll. Saat ini terdapat empat macam cara autentikasi

pada *smartphone* yang sering dijumpai, yaitu *password, fingerprint scanner, face recognition*, dan *iris recognition.*

2.2 Kajian Integrasi Topik Dengan Al-Quran/Hadits

Setiap individu berhak memiliki privasi dan rahasia di kehidupannya. Dalam Al-Qur'an dijelaskan anjuran untuk menghormati batas-batas privasi orang lain dalam Surat An-Nur ayat 27. Berdasarkan ayat terebut, dalam Kitab Tafsir Al-Misbah (Shihab, 2000) yakni, Allah SWT memerintahkan kaum muslimin untuk menghindari tempat dan sebab-sebab yang dapat menimbulkan kecurigaan dan prasangka buruk. Karena itu, di sini diperintahkan untuk meminta izin sebelum masuk ke rumah. Oleh karena itu, setiap individu juga memiliki kewajiban untuk menghormati dan menghargai batas-batas privasi yang dimiliki individu lainnya. Dalam kasus ini, diibaratkan bahwa *smartphone* adalah rumah seseorang. Kemudian Rasulullah SAW mengajarkan untuk tidak berdiri di depan pintu saat bertamu dan melarang untuk menengok (mengintip) ke dalam rumah sesorang (HR.Muslim No.4016).

Rasulullah SAW bersabda:

حَدَّثَنِي زُهَيْرُ بْنُ حَرْبٍ حَدَّثَنَا جَرِيرٌ عَنْ سُهَيْلٍ عَنْ أَبِيهِ عَنْ أَبِي هُرَيْرَةَ عَنْ النَّبِيِّ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ قَالَ مَنْ اطَّلَعَ فِي بَيْتِ قَوْمٍ بِغَيْرِ إِذْنِهِمْ فَقَدْ حَلَّ لَهُمْ أَنْ يَفْقَئُوا عَيْنَهُ

"Barang siapa menengok ke dalam rumah seseorang tanpa izin pemiliknya, maka sungguh mereka boleh mencongkel mata orang itu." (HR. Muslim no. 4016).

Berdasarkan hadist di atas, disebutkan bahwa tidak diperbolehkannya untuk melihat catatan atau mendengar pembicaraan orang lain tanpa sepengetahuan dan kerelaannya. Hal tersebut tentu mendukung adanya proses pengamanan data agar

batas-batas privasi senantiasa terjaga dan tidak menimbulkan fitnah-fitnah yang disebabkan oleh omongan dari orang ke orang.

2.3 Kajian Topik Dengan Teori Pendukung

Penelitian ini dilakukan untuk menangani permasalahan aktivitas masyarakat pada saat pandemi COVID-19, dimana para pengguna *smartphone* harus melakukan proses autentikasi untuk membuka dan mengakses *smartphone*. Sehingga, dari permasalahan tersebut, salah satu bentuk pengembangan pada *smartphone* yang sedang berkembang yaitu sistem autentikasi dengan biometrik iris mata. Dengan adanya sistem autentikasi biometrik iris mata, keamanan data serta fleksibilitas dalam melakukan proses transaksi melalui *smartphone* dapat lebih terjaga. Disampaikan bahwa penggunaan kata kunci (*password*) dalam keamanan sistem komputer bisa diretas dengan mudah, sehingga dilakukan peningkatan keamanan sistem komputer berupa penggunaan identifikasi menggunakan biometrik. Dengan adanya proses autentikasi *smartphone*, tentunya memerlukan keamanan data yang tinggi dan untuk mengoptimalisasi hal tersebut dapat dinggunakan ilmu kriptografi yang berfungsi dalam proses enkripsi dan dekripsi data.

Proses kriptografi untuk mengamankan data iris mata pengguna *smartphone* memiliki dua proses yaitu enkripsi dan dekripsi. Sesuai dengan teori sebelumnya, bahwa proses enkripsi merupakan proses pengacakan *plaintext* (naskah asli) menjadi *chipertext* (naskah acak) yang sulit untuk dibaca apabila tidak memiliki kunci yang sesuai untuk melakukan proses dekripsi. Sedangkan proses dekripsi proses untuk menemukan *plaintext* (naskah asli) dari proses enkripsi yang telah dilakukan sebelumnya dengan menggunakan kunci dekripsi.

Pada penelitian sebelumnya oleh Julus (2022) dilakukan penelitian dengan menggunakan algoritma *Rubik's Cube* untuk melakukan proses pengacakan dan penggunaan kunci dari algoritma Rivest-Shamir-Adleman (RSA) yang diterapkan pada operasi XOR. Namun, pada penelitian ini belum diteliti terkait waktu yang digunakan untuk melakukan proses enkripsi dan dekripsi. Kemudian (Belkaid, 2015) melakukan penelitian dengan menggunakan algoritma enkripsi *hybrid* dengan menggunakan algoritma *Advanced Encryption Standard* (AES) dan Algoritma *Rivest-Shamir-Adleman* (RSA) untuk mengenkripsi kunci yang dimiliki algoritma AES. Penggunaan Algoritma *Rivest-Shamir-Adleman* (RSA) digunakan karena dapat memberikan tingkat keamanan yang tinggi berdasarkan proses pembangkitan kunci yang sulit dengan dasar perhitungan modulo dan pemfaktoran bilangan prima yang besar (Belkaid, 2015). Kemudian Loukhaoukha, dkk. (2012) melakukan penelitian untuk menerapkan proses enkripsi dan dekripsi pada gambar dengan menggunakan algoritma *Rubik's Cube* dan menjelaskan bahwa Algoritma *Rubik's Cube* diperlukan sebagai algoritma yang berfungsi untuk melakukan pengacakan piksel pada citra yang akan digunakan. Algoritma *Rubik's Cube* menunjukkan skema enkripsi dan kemampuan pengacakan yang baik dan dapat menahan serangan baik statistik maupun diferensial (Loukhaoukha, dkk., 2012). Namun, pada penelitian sebelumnya hanya menggunakan algoritma *Rubik's Cube* dalam melakukan enkripsi dan dekripsi data.

Berdasarkan pada penelitian sebelumnya, didapatkan bahwa terdapat penelitian yang menggunakan algoritma *Rubik's Cube* dan algoritma *Rivest-Shamir-Adleman* (RSA) dengan menerapkan algoritma enkripsi *hybrid*. Penelitian ini dilakukan untuk mendapatkan inovasi baru dengan menerapkan gabungan dua

algoritma tersebut untuk melakukan proses enkripsi dan dekripsi tanpa menerapkan algoritma enkripsi *hybrid* seperti yang telah dilakukan pada penelitian sebelumnya. Dikarenakan penggunaan algoritma *Rivest-Shamir-Adleman* (RSA) yang juga memiliki tingkat keamanan yang tinggi serta adanya skema enkripsi dan kemampuan pengacakan yang baik oleh algoritma *Rubik's Cube.*

Dengan adanya proses kriptografi tentunya gambar akan melewati banyak proses dan melakukan pengubahan posisi piksel, warna, dll. Maka dari itu diperlukan tahap evaluasi yang berfungsi untuk melakukan analisis terkait citra yang telah di dekripsi dan citra awal. Dengan melakukan evaluasi, peneliti dapat mengetahui tingkat akurasi hasil dekripsi dari metode yang diteliti. Pada penelitian ini, metode evaluasi yang dipillih ialah metode *Structural Similarity Index Metrics* (SSIM) dan *Mean Square Error* (MSE). Kedua metode tersebut melakukan proses perbandingan oleh 2 citra yang berkaitan dengan kemiripan dan tingkat kesamaan citra tersebut.

BAB III
METODE PENELITIAN

3.1 Jenis Penelitian

Jenis penelitian dalam penelitian ini menggunakan metode eksperimen. Penelitian eksperimen merupakan pendekatan penelitian yang objektif, sistematis, dan terkontrol dalam memprediksi suatu kejadian. Dalam arti lain, metode eksperimen juga digunakan untuk menguji hubungan sebab akibat. Sehingga pada penelitian ini akan digunakan metode eksperimen, karena hal tersebut sesuai dengan tujuan yang ada pada penelitian ini untuk mengetahui hasil akurasi dan efisiensi waktu penggunaan algoritma *Rubik's Cube* yang dikombinasikan dengan algoritma *Rivest-Shamir-Adleman* (RSA) dalam pemanfaatan enkripsi dan dekripsi citra digital iris mata.

3.2 Data dan Sumber Data

Data yang digunakan pada penelitian ini ialah data gambar (citra digital) *RGB* dengan ukuran M x N yaitu 320 x 240 *pixel* dengan format file *.bmp yang terdiri dari 5 gambar iris masing-masing kiri dan kanan dari 46 orang, dengan total 460 gambar. Data yang digunakan pada penelitian ini merupakan MMU *Iris Dataset* yang dapat diperoleh secara gratis pada website Kaggle yang telah terlisensi oleh *Multimedia University Iris Database for Biometric Attendance System* pada link *https://www.kaggle.com/naureenmohammad/mmu-iris-dataset.*

Data yang dimiliki akan diproses dengan menggunakan program komputer dengan Bahasa pemrograman yaitu *python* dengan bantuan library *Pillow*.

3.3 Teknik Analisis Data

Proses enkripsi dan dekripsi pada penelitian ini merupakan gabungan dari algoritma *Rubik's Cube* dan algoritma *Rivest-Shamir-Adleman* (RSA). Tahapan yang digunakan yaitu diawali dengan enkripsi menggunakan algoritma *Rubik's Cube* kemudian dilanjutkan dengan enkripsi menggunakan algoritma *Rivest-Shamir-Adleman* (RSA). Pada proses dekripsi dilakukan dengan menggunakan algoritma *Rivest-Shamir-Adleman* (RSA) kemudian dilanjutkan dengan dekripsi menggunakan algoritma *Rubik's Cube.*

3.3.1 Proses Enkripsi

Proses enkripsi dilakukan dengan menerapkan dua algoritma yaitu algoritma Rubik's Cube dan algoritma *Rivest-Shamir-Adleman* (RSA). Berikut beberapa tahapan yang perlu dilakukan pada proses enkripsi menggunakan algoritma *Rubik's Cube* dengan rincian:

1. Menyiapkan *plain image* yaitu citra digital iris mata berdimensi M x N yaitu 320 x 240 piksel.
2. Mengubah *plain image* ke dalam bentuk matriks dan menginisialisasi tiap entri matriks *plain teks* dengan nilai dari tingkat keabuan warna R(*Red*), G(*Green*), dan B(*Blue*).
3. Mendapatkan vektor Kr dengan panjang *N* dan vektor Kc dengan panjang *M* dengan berisi kumpulan bilangan acak dan tidak konstan.
4. Menentukan iterasi maksimum.
5. Memberikan nilai iterasi = 0 dan menambahkan nilai iterasi = iterasi +1.
6. Mengoperasikan vektor Kr pada baris matriks *plain image* dan mengoperasikan vektor Kc pada kolom matriks *plain image.*

7. Berdasar pada langkah f akan diperoleh citra acak yang dilambangkan dengan matriks I_{scr}.
8. Kemudian akan diimplementasikan operator XOR pada baris I_{scr} dengan melakukan penggeseran vektor Kc dari atas ke bawah atau sebaliknya untuk menghasilkan baris citra baru yang disebut matriks I_1.
9. Kemudian akan diimplementasikan operator XOR pada kolom I_1 dengan melakukan pergeseran vektor Kr dari kiri ke kanan untuk menghasilkan kolom citra baru yang disebut matriks I_{enc}.
10. Apabila nilai iterasi = iterasi maksimum maka gambar terenkripsi akan disimpan pada matriks I_{enc} dan bisa dilakukan transformasi matriks menjadi *cipher image* pertama (gambar terenkripsi oleh algoritma *Rubik's Cube*). Apabila iterasi ≠ iterasi maksimum maka akan diulang kembali dari langkah e hingga terpenuhi untuk nilai iterasi = iterasi maksimum.

Selanjutnya adalah beberapa tahapan yang perlu dilakukan pada proses enkripsi menggunakan algoritma *Rivest-Shamir-Adleman* (RSA) dengan rincian:

1. Mengambil *cipher image* pertama dari algoritma *Rubik's Cube*.
2. Mengubah *cipher image* pertama ke dalam bentuk matriks dan menginisialisasi tiap entri matriks *plain teks* dengan nilai dari tingkat keabuan warna R(*Red*), G(*Green*), dan B(*Blue*) yang disebut dengan matriks I_{enc}.
3. Menentukan bilangan prima *p* dan *q*.
4. Memproses bilangan prima p dan q untuk menentukan pasangan kunci publik dan kunci privat.

5. Mengimplementasikan pasangan kunci publik RSA pada matriks I_{enc} dan menghasilkan matriks I_{enc2}.
6. Transformasi matriks I_{enc2} menjadi citra digital untuk mendapatkan *cipher image* kedua (gambar terenkripsi oleh algoritma RSA).

3.3.2 Proses Dekripsi

Proses dekripsi dilakukan dengan menerapkan dua algoritma yaitu algoritma *Rivest-Shamir-Adleman* (RSA) dan algoritma *Rubik's Cube*. Berikut beberapa tahapan yang perlu dilakukan pada proses dekripsi menggunakan algoritma *Rivest-Shamir-Adleman* (RSA) dengan rincian:

1. Mengambil *encrypt image* kedua dari proses enkripsi menggunakan algoritma *Rubik's Cube* dan algoritma *Rivest-Shamir-Adleman* (RSA).
2. Mengubah *encrypt image* kedua menjadi matriks *encrypt image* kedua yang dinotasikan dengan *(I_{enc2}).*
3. Menyiapkan pasangan kunci privat untuk melakukan proses dekripsi.
4. Mengoperasikan pasangan kunci privat pada matriks *encrypt image* kedua *(I_{enc2})* yang kemudian menghasilkan matriks *decrypt image* pertama yang dinotasikan dengan *(I_{dec}).*
5. Transformasi matriks *(I_{dec})* menjadi *decrypt image* pertama.

Selanjutnya adalah beberapa tahapan yang perlu dilakukan pada proses dekripsi menggunakan algoritma *Rubik's Cube* dengan rincian:

1. Masukkan *decrypt image* pertama dari proses dekripsi menggunakan algoritma *Rivest-Shamir-Adleman* (RSA).
2. Mengubah *decrypt image* pertama menjadi matriks *decrypt image* pertama (I_{dec1}).

3. Masukkan vektor Kr, vektor Kc, dan nilai iterasi maksimum.
4. Kemudian Inisialisasi nilai iterasi=0 dan inkremen nilai
    a. iterasi = iterasi + 1.
5. Mengoperasikan operator XOR pada vektor Kr dari kolom matriks (I_{dec1}) untuk mendapatkan kembali matriks I_1.
6. Mengoperasikan operasi XOR pada vektor Kc dari baris matriks (I_{dec1}) untuk mendapatkan kembali matriks I_{scr1}.
7. Mengoperasikan vektor Kr pada baris matriks I_{scr1} dan mengoperasikan vektor Kc pada kolom matriks I_{scr1}.
8. Apabila nilai iterasi = iterasi maksimum maka gambar terenkripsi akan disimpan pada matriks I_{dec2} dan bisa dilakukan transformasi matriks menjadi *decrypt image* kedua (gambar terdekripsi oleh algoritma *Rubik's Cube*). Apabila iterasi ≠ iterasi maksimum maka akan diulang kembali dari langkah d hingga terpenuhi untuk nilai iterasi = iterasi maksimum.

3.3.3 Tahap Evaluasi

Tahap evaluasi merupakan tahap untuk mengkaji metode yang sudah diterapkan. Dimana proses ini berfungsi untuk mengetahui tingkat keberhasilan terkait algoritma yang digunakan pada penelitian ini. Evaluasi pada penelitian ini akan menggunakan metode *Structural Similarity Index Metrics* (SSIM) dan *Mean Square Error* (MSE). Dimana kedua nya merupakan metode untuk melakukan analisis terkait tingkat kemiripan citra terdekripsi yang telah melalui proses enkripsi dengan citra awal sebelum proses enkripsi. Sehingga, dapat mengetahui dan memperoleh bagaimana tingkat akurasi dan keberhasilan dari proses enkripsi dan dekripsi yang telah diteliti.

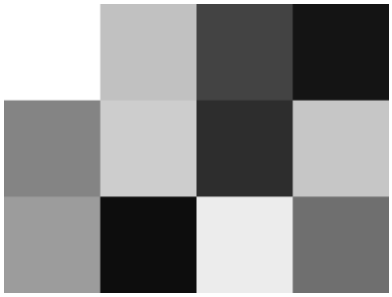
BAB IV
HASIL DAN PEMBAHASAN

4.1 Proses Enkripsi dan Dekripsi Citra Digital

4.1.1 Proses Enkripsi Citra Digital dengan Algoritma *Rubik's Cube*

Berikut merupakan contoh penerapan proses enkripsi dengan menggunakan algoritma *Rubik's Cube* pada citra digital yang berukuran 4 x 3 piksel dengan tipe gambar *grayscale*.

1. *Plain image* berukuran *M* x *N* yaitu 4 x 3 piksel yang dimisalkan sebagai berikut.



Gambar 4.1 *Plain Image* ukuran 4x3 piksel

2. Selanjutnya *plain-image* dibaca sebagai matriks *plain-image* (I_0) sebagai berikut.

$$I_0 = \begin{bmatrix} 255 & 193 & 67 & 20 \\ 132 & 205 & 45 & 198 \\ 156 & 13 & 236 & 111 \end{bmatrix}$$

3. Inisialisasi secara acak pada bilangan kunci untuk membentuk vektor Kr dengan panjang *N* dan vektor Kc dengan panjang M.

   Dimisalkan sebagai berikut:

   $Vektor\ Kr = [1 \quad 3 \quad 2]$

   $Vektor\ Kc = [0 \quad 3 \quad 1 \quad 1]$

4. Inisialisasi jumlah iterasi maksimum = 1.

5. Setiap baris *i* pada matriks I_0 akan diproses sebagai berikut:

   a. Menghitung jumlah elemen pada baris I yang dinotasikan sebagai $a(i)$ sesuai dengan persamaan (2.1) dengan $i = 0, 1, \dots, N-1$ sehingga didapatkan hasil berikut:

      $a(0) = 535$,

      $a(1) = 580$,

      $a(2) = 516$.

   b. Menghitung nilai $Ma(i)$ sesuai dengan persamaan (2.2) sehingga didapatkan hasil sebagai berikut:

      $Ma(0) = 1$,

      $Ma(1) = 0$,

      $Ma(2) = 0$.

      Jika $Ma(i) = 0$ maka baris *i* di rotasi ke kanan dan selain itu baris *i* akan di rotasi ke kiri sebanyak vektor *Kr(i)*.

   c. Sehingga menghasilkan matriks yang dinotasikan sebagai matriks I_R dengan entri sebagai berikut.

$$I_R = \begin{bmatrix} 193 & 67 & 20 & 255 \\ 205 & 45 & 198 & 132 \\ 236 & 111 & 156 & 13 \end{bmatrix}$$

6. Setiap kolom *j* pada matriks I_0 akan diproses sebagai berikut:

   a. Menghitung jumlah elemen pada kolom *j* yang dinotasikan sebagai $\beta(j)$ sesuai dengan persamaan (2.3) dengan $j = 0, 1, \dots, M-1$ sehingga didapatkan hasil berikut:

      $\beta(0) = 634$,

      $\beta(1) = 223$,

$\beta(2) = 374$,

$\beta(3) = 400$.

b. Menghitung nilai $M\beta(j)$ sesuai dengan persamaan (2.4) sehingga didapatkan hasil sebagai berikut:

$M\beta(0) = 0$,

$M\beta(1) = 1$,

$M\beta(2) = 0$,

$M\beta(3) = 0$.

Jika $M\beta(j) = 0$ maka kolom *j* di rotasi ke atas dan selain itu kolom *j* akan di rotasi ke bawah sebanyak vektor *Kc(j)*.

Sehingga menghasilkan matriks yang dinotasikan sebagai matriks I_{scr} dengan entri sebagai berikut:

$$I_{scr} = \begin{bmatrix} 193 & 67 & 198 & 132 \\ 205 & 45 & 156 & 13 \\ 236 & 111 & 20 & 255 \end{bmatrix}$$

c. Kemudian pada baris i matriks I_{scr} dilakukan operasi XOR dengan vektor Kc dengan persamaan (2.5) dan (2.6) sehingga diperoleh matriks I_1 dengan entri sebagai berikut:

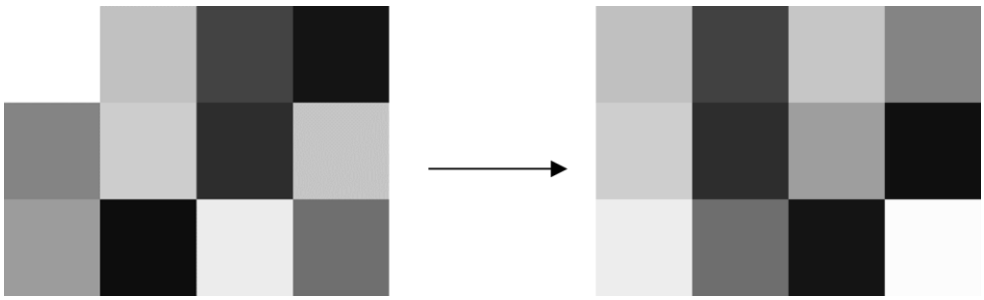
$$I_1 = \begin{bmatrix} 193 & 64 & 199 & 133 \\ 205 & 46 & 157 & 12 \\ 236 & 108 & 21 & 254 \end{bmatrix}$$

d. Kemudian pada kolom j matriks I_1 dilakukan operasi XOR dengan vektor Kr dengan persamaan (2.7) dan (2.8) sehingga diperoleh matriks I_{enc} dengan entri sebagai berikut:

$$I_{enc} = \begin{bmatrix} 192 & 65 & 198 & 132 \\ 206 & 45 & 158 & 15 \\ 237 & 110 & 20 & 252 \end{bmatrix}$$

e. Karena iterasi maksimum = 1 maka proses pengacakan pixel hanya dilakukan satu kali, sehingga matriks yang diperoleh untuk proses enkripsi dengan menggunakan algoritma *Rubik's cube* adalah matriks I_{enc}.

Dengan demikian matriks I_{enc} dapat dinyatakan sebagai *ciphertext* dari proses enkripsi menggunakan algortima *Rubik's Cube*. Selanjutnya adalah membaca kembali nilai dari setiap piksel yang telah dienkripsi dengan algoritma algortima *Rubik's Cube* untuk diubah ke dalam bentuk citra digital. Dengan citra hasil enkripsi sebagai berikut:



Gambar 4.2 Hasil Enkripsi Citra Digital Algoritma *Rubik's Cube*

4.1.2 Proses Enkripsi Citra Digital dengan Algoritma RSA

Pada proses enkripsi menggunakan algoritma *Rivest-Shamir-Adleman* (RSA), dilakukan dua tahapan utama yaitu proses pembangkitan kunci dan proses enkripsi. Pada penelitian ini, algoritma *Rivest-Shamir-Adleman* (RSA) digunakan sebagai algoritma enkripsi kedua setelah algoritma *Rubik's Cube.*

Berikut merupakan proses pembangkitan kunci untuk algoritma *Rivest-Shamir-Adleman* (RSA) dengan tahapan sebagai berikut:

1. Menentukan nilai p dan nilai q.

   Misalkan nilai p = 127 dan nilai q = 59.

2. Menghitung nilai modulus (*n*).

$n = p\ x\ \ q = 127\ x\ 59 = 7493$

3. Menentukan nilai $\phi(n)$ atau nilai *totient euler (n).*

   $\phi(n) = (p-1)\ x\ (q-1)$

   $= (127-1)\ x\ (59-1)$

   $=\ 126\ x\ 58$

   $= 7308$

4. Ambil sembarang nilai e dengan syarat $GCD(e, \phi(n)) = 1$.

   Misal e = 4327, apakah $GCD(\ 4327, 7308\ ) = 1$. Dengan menggunakan algoritma Euclidian, diperoleh:

   7308= 4327 x 1 + 2981

   4327 = 2981 x 1 + 1346

   2981 = 1346 x 2 + 289

   1346 = 289 x 4 + 190

   289 = 190 x 1 + 99

   190 = 99 x 1 + 91

   99 = 91 x 1 + 8

   91 = 8 x 11 + 3

   8 = 3 x 2 + 2

   3 = 2 x 1 + **1**

   2 = 1 x 2 + 0

   Maka dapat dikatakan bahwa $GCD(\ 4327\ ,7308\ ) = 1$, sehingga dapat digunakan e = 4327.

5. Mencari nilai d yang sesuai dengan persamaan (2.10) dengan hasil sebagai berikut:

Dengan d = 0 maka $(0\ x\ 4327) mod\ 7308 = 0$,

d = 1 maka $(1\ x\ 4327) mod\ 7308 = 4327$,

d = 2 maka $(2\ x\ 4327) mod\ 7308 = 1346$,

d = 3 maka $(3\ x 4327) mod\ 7308 = 5673$,

d = 4 maka $(4\ x\ 4327) mod\ 7308 = 2692$,

⋮

d = 2731 maka $\boldsymbol{(2731\ x\ 4327) mod\ 7308 = 1}$.

Sehingga didapat nilai d = 2731.

6. Setelah itu, didapatkan pasangan kunci sebagai berikut:

   Kunci publik (e, n) = (4327, 7308)

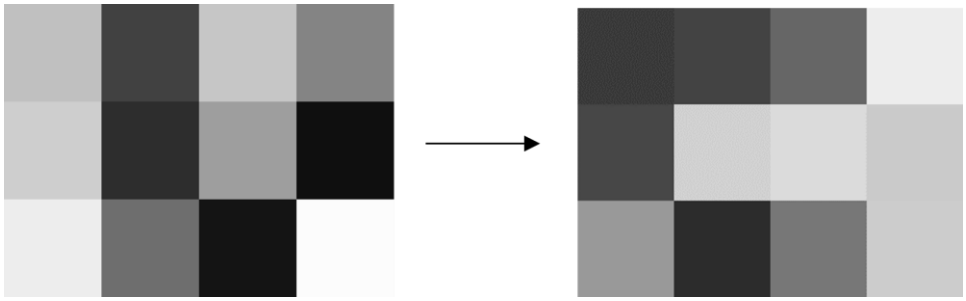
   Kunci privat (d, n) = (2731, 7308)

Selanjutnya dilakukan proses enkripsi dengan algoritma *Rivest-Shamir-Adleman* (RSA) pada matriks I_{enc} menggunakan kunci publik dengan menggunakan persamaan (2.11).

Kemudian diperoleh matriks yang dinotasikan dengan matriks I_{enc2} dengan entri sebagai berikut.

$$I_{enc2} = \begin{bmatrix} 59 & 67 & 102 & 237 \\ 71 & 210 & 219 & 202 \\ 153 & 44 & 119 & 204 \end{bmatrix}$$

Dengan demikian matriks I_{enc2} dapat dinyatakan sebagai *ciphertext* dari proses enkripsi menggunakan algortima *Rubik's Cube* dan algoritma *Rivest-Shamir-Adleman* (RSA). Selanjutnya adalah membaca kembali nilai dari setiap piksel yang telah dienkripsi untuk diubah ke dalam bentuk citra digital. Maka dapat dilihat bahwa citra digital telah berubah dari citra digital aslinya.



Gambar 4.3 Hasil Enkripsi Citra Digital Algoritma RSA

4.1.3 Proses Dekripsi Citra Digital dengan Algoritma RSA

Berikut merupakan contoh penerapan proses dekripsi dengan menggunakan algoritma *Rivest-Shamir-Adleman* (RSA) pada citra digital yang berukuran 4 x 3 piksel dengan tipe gambar *grayscale*.

1. Menentukan citra yang akan dienkripsi
2. Menentukan kunci privat yang digunakan sesuai pada saat proses pembangkitan kunci, yaitu:

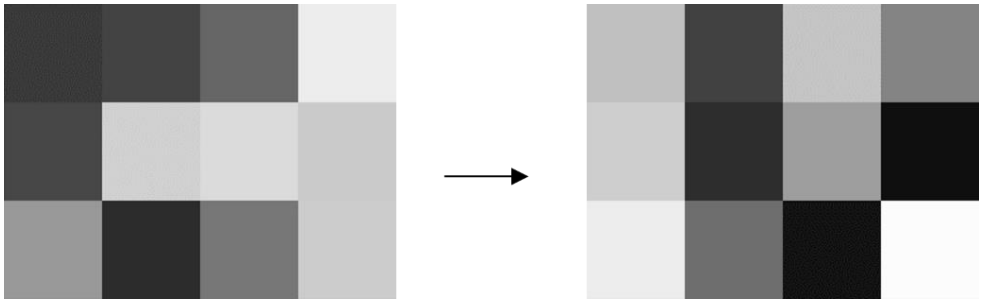
   Kunci privat (d, n) = (2731, 7308).
3. Selanjutnya dilakukan proses dekripsi dengan algoritma *Rivest-Shamir-Adleman* (RSA) pada matriks I_{enc2} menggunakan kunci privat dengan menggunakan persamaan (2.12).

   Sehingga diperoleh matriks I_{dec} dengan entri sebagai berikut.

$$I_{dec} = \begin{bmatrix} 192 & 65 & 198 & 132 \\ 206 & 45 & 158 & 15 \\ 237 & 110 & 20 & 252 \end{bmatrix}$$

Dengan demikian matriks I_{dec} dapat dinyatakan sebagai *plaintext* dari proses dekripsi menggunakan algoritma *Rivest-Shamir-Adleman* (RSA). Selanjutnya adalah membaca kembali nilai dari setiap piksel yang telah

dienkripsi untuk diubah ke dalam bentuk citra digital. Dengan citra hasil enkripsi sebagai berikut:



Gambar 4.4 Hasil Dekripsi Citra Digital Algoritma RSA

4.1.4 Proses Dekripsi Citra Digital dengan Algoritma *Rubik's Cube*

Pada penelitian ini, algoritma *Rubik's Cube* digunakan sebagai algoritma dekripsi kedua setelah algoritma *Rivest-Shamir-Adleman* (RSA). Terdapat beberapa tahapan pada proses dekripsi oleh algoritma *Rubik's Cube* dengan rincian sebagai berikut:

1. Kemudian pada kolom *j* matriks I_{dec} dilakukan operasi XOR dengan vektor Kr dan menghasilkan matriks I_{dec1} dengan menggunakan persamaan (2.7) dan (2.8) sehingga diperoleh matriks I_{dec1} dengan entri sebagai berikut:

$$I_{dec1} = \begin{bmatrix} 193 & 64 & 199 & 133 \\ 205 & 46 & 157 & 12 \\ 236 & 108 & 21 & 254 \end{bmatrix}$$

2. Kemudian pada baris i matriks I_{dec1} dilakukan operasi XOR dengan vektor Kc dan menghasilkan matriks I_{C1} dengan persamaan (2.5) dan (2.6) sehingga diperoleh matriks I_1 dengan entri sebagai berikut:

$$I_{C1} = \begin{bmatrix} 193 & 67 & 198 & 132 \\ 205 & 45 & 156 & 13 \\ 236 & 111 & 20 & 255 \end{bmatrix}$$

3. Setiap kolom *j* pada matriks I_0 akan diproses sebagai berikut:

a. Menghitung jumlah elemen pada kolom *j* yang dinotasikan sebagai $\beta(j)$ sesuai dengan persamaan (2.3) dengan $j = 0, 1, \dots, M-1$.

Sehingga didapatkan hasil sebagai berikut:

$\beta(0) = 634$,

$\beta(1) = 223$,

$\beta(2) = 374$,

$\beta(3) = 400$.

b. Menghitung nilai $M\beta(j)$ sesuai dengan persamaan (2.4) sehingga didapatkan hasil sebagai berikut:

$M\beta(0) = 0$,

$M\beta(1) = 1$,

$M\beta(2) = 0$,

$M\beta(3) = 0$.

Jika $M\beta(j) = 0$ maka kolom *j* di rotasi ke bawah dan selain itu kolom *j* akan di rotasi ke atas sebanyak vektor *Kc(j)*.

Sehingga menghasilkan matriks yang dinotasikan sebagai matriks I_{R1} dengan entri sebagai berikut.

$$I_{R1} = \begin{bmatrix} 193 & 67 & 20 & 255 \\ 205 & 45 & 198 & 132 \\ 236 & 111 & 156 & 13 \end{bmatrix}$$

4. Setiap baris *i* pada matriks I_0 akan diproses sebagai berikut:

a. Menghitung jumlah elemen pada baris I yang dinotasikan sebagai $a(i)$ sesuai dengan persamaan (2.1) dengan $i = 0, 1, \dots, N-1$.

$a(0) = 535$,

$a(1) = 580$,

$a(2) = 516$.

b. Menghitung nilai $Ma(i)$ sesuai dengan persamaan (2.2) sehingga didapatkan hasil sebagai berikut:

$Ma(0) = 1$,

$Ma(1) = 0$,

$Ma(2) = 0$.

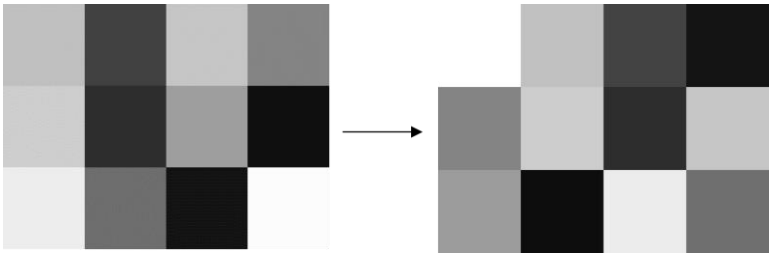
Jika $Ma(i) = 0$ maka baris *i* di rotasi ke kiri dan selain itu baris *i* akan di rotasi ke kanan sebanyak vektor *Kr(i)*.

Sehingga menghasilkan matriks yang dinotasikan sebagai matriks I_{dec2} dengan entri sebagai berikut.

$$I_{dec2} = \begin{bmatrix} 255 & 193 & 67 & 20 \\ 132 & 205 & 45 & 198 \\ 156 & 13 & 236 & 111 \end{bmatrix}$$

Dengan demikian matriks I_{dec2} dapat dinyatakan sebagai *plaintext* dari proses dekripsi menggunakan algortima *Rubik's Cube* dan algoritma *Rivest-Shamir-Adleman* (RSA).

Dengan demikian matriks I_{dec2} dapat dinyatakan sebagai *plaintext* dari proses dekripsi menggunakan algoritma *Rivest-Shamir-Adleman* (RSA) dan algoritma *Rubik's Cube.* Selanjutnya adalah membaca kembali nilai dari setiap piksel yang telah dienkripsi dengan algoritma *Rivest-Shamir-Adleman* (RSA) dan algoritma *Rubik's Cube* untuk diubah ke dalam bentuk citra digital. Dengan citra hasil dekripsi sebagai berikut:



Gambar 4.5 Hasil Dekripsi Citra Digital Algoritma *Rubik's Cube*

4.2 Pembahasan Hasil Enkripsi dan Dekripsi Citra Digital Iris Mata

Pada bagian ini, sebelumnya telah dilakukan pengujian terkait proses enkripsi dan dekripsi menggunakan program *python* dengan bantuan *text editor* yaitu *visual studio code.* Pengujian dilakukan dengan melakukan input citra digital dan kunci pada text editor untuk memperoleh hasil citra digital yang telah diproses dan waktu yang dibutuhkan untuk melakukan proses enkripsi dan dekripsi.

Berikut pengujian yang dilakukan menggunakan algoritma *Rubik's Cube* dan algoritma *Rivest-Shamir-Adleman* (RSA) terhadap 3 citra digital iris mata yang terdapat pada dataset dan diambil secara acak dan menggunakan variasi kunci dari algoritma RSA dan iterasi maksimum dari algoritma *Rubik's Cube* yang berbeda.

Tabel 4.1 Hasil Pengujian Waktu Proses Enkripsi dan Dekripsi

No.	Nama Percobaan	Kunci Publik Algoritma RSA	Kunci Privat Algoritma RSA	Iterasi Maks. Algoritma *Rubik's Cube*	Waktu Proses Enkripsi (Detik)	Waktu Proses Dekripsi (Detik)
1.	aeval1_1	(197,403)	(53,403)	1	0,794	0,656
2.	aeval1_2	(2741,3233)	(461,3233)	1	11,542	1,936
3.	aeval1_3	(9551,15151)	(671,15151)	1	73,833	3,806
4.	aeval1_4	(197,403)	(53,403)	2	1,201	1,125

5.	aeval1_5	(197,403)	(53,403)	3	1,558	1,446
6.	aeval1_6	(197,403)	(53,403)	4	1,977	1,813
7.	bryanl1_1	(197,403)	(53,403)	1	0,793	0,651
8.	bryanl1_2	(2741,3233)	(461,3233)	1	11,344	1,843
9.	bryanl1_3	(9551,15151)	(671,15151)	1	73,798	3,770
10.	bryanl1_4	(197,403)	(53,403)	2	1,145	1,027
11.	bryanl1_5	(197,403)	(53,403)	3	1,579	1,498
12.	bryanl1_6	(197,403)	(53,403)	4	1,945	1,902
13.	chingycl1_1	(197,403)	(53,403)	1	0,801	0,649
14.	chingycl1_2	(2741,3233)	(461,3233)	1	11,162	1,886
15.	chingycl1_3	(9551,15151)	(671,15151)	1	75,822	3,943
16.	chingycl1_4	(197,403)	(53,403)	2	1,176	1,039
17.	chingycl1_5	(197,403)	(53,403)	3	1,535	1,513
18.	chingycl1_6	(197,403)	(53,403)	4	1,940	1,816

Dapat dilihat dari hasil pengujian yang dilakukan didapatkan rata-rata waktu 0,796 detik untuk proses enkripsi dan 0,652 detik untuk proses dekripsi menggunakan pasangan kunci publik (197,403) dan pasangan kunci privat (53,403) dan dengan iterasi maksimum = 1 untuk algoritma *Rubik's Cube*. Kemudian, dengan pasangan kunci publik dan kunci privat yang sama namun dengan iterasi maksimum = 2 algoritma *Rubik's Cube* didapatkan rata-rata waktu 1,174 detik untuk proses enkripsi dan 1,063 detik untuk proses dekripsi. Selanjutnya, dengan iterasi maksimum = 3 didapatkan rata-rata waktu 1,557 detik untuk proses enkripsi dan 1,485 detik untuk proses dekripsi. Dengan iterasi maksimum = 4 didapatkan rata-rata waktu 1,954 detik untuk proses enkripsi dan 1,843 detik untuk proses dekripsi. Kemudian, didapatkan rata-rata waktu 11,349 detik untuk proses enkripsi

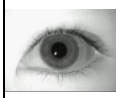
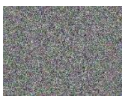
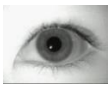
dan 1,845 detik untuk proses dekripsi menggunakan pasangan kunci publik (2741,3233) dan pasangan kunci privat (461,3233) dan dengan iterasi maksimum = 1 untuk algoritma *Rubik's Cube*. Dengan menggunakan pasangan kunci publik (9551,15151) dan pasangan kunci privat (671,15151) dan dengan iterasi maksimum = 1 didapatkan rata-rata waktu 74,484 detik untuk proses enkripsi dan 3,839 detik untuk proses dekripsi.

Penggunan besar kunci pada algoritma *Rivest-Shamir-Adleman (RSA)* dan iterasi maksimum pada algoritma *Rubik's Cube* dapat mempengaruhi waktu yang dibutuhkan oleh sistem untuk melakukan proses enkripsi dan dekripsi. Seperti yang tertera pada Tabel 4.1, bahwa semakin kecil nilai kunci dan iterasi maksimum yang ditentukan maka waktu yang dibutuhkan untuk proses enkripsi maupun dekripsi akan relatif lebih cepat. Dengan penambahan nilai pada kunci dan iterasi maksimum akan membutuhkan tambahan waktu sehingga proses enkripsi dan dekripsi relatif lebih lama.

Selanjutnya merupakan pengujian yang dilakukan menggunakan metode *Structural Similarity Index Metrics* (SSIM) dan *Mean Square Error* (MSE) untuk mengukur tingkat akurasi dari struktural citra antara citra awal sebelum dilakukan proses enkripsi dan dekripsi dengan citra yang telah melalui proses enkripsi dan dekripsi.

Tabel 4.2 Hasil Pengujian Tingkat Akurasi Proses Enkripsi dan Dekripsi

No.	Citra Digital Awal	Hasil Enkripsi	Nilai SSIM Enkripsi	Nilai MSE Enkripsi	Hasil Dekripsi	Nilai SSIM Dekripsi	Nilai MSE Dekripsi
1.	aeval1_1 		0,01	43840.11		1	0

2.	aeval1_2 		0,01	38397.26		1	0
3.	aeval1_3 		0,01	40326.03		1	0
4.	aeval1_4 		0,01	43832		1	0
5.	aeval1_5 		0,01	44110.59		1	0
6.	aeval1_6 		0,01	43936.16		1	0

Nilai *Structural Similarity Index Metrics* (SSIM) yang didapatkan dapat menjadi tolak ukur dari tingkat kesamaan *structural* citra awal dan citra yang telah terenkripsi. Semakin kecil nilai *Structural Similarity Index Metrics* (SSIM) maka semakin baik pula hasil citra terenkripsi karena citra yang dihasilkan memiliki *structural* yang berbeda dengan citra awal. Namun pada proses dekripsi diperlukan nilai *Structural Similarity Index Metrics* (SSIM) maksimum yaitu 1 untuk menunjukkan bahwa proses dekripsi berhasil karena citra terdekripsi memiliki *structural* yang sama dengan citra awal. Selanjutnya, untuk nilai *Structural Similarity Index Metrics* (SSIM) yang dihasilkan dari proses enkripsi dan dekripsi pada pengujian menggunakan citra digital iris mata yang sama tetapi nilai kunci dan iterasi maksimum berbeda menghasilkan nilai SSIM yang sama.

Pada tabel 4.2, dapat dilihat hasil pengujian yang dilakukan dengan total 6 kali percobaan pada 1 citra digital iris mata dengan menggunakan besar kunci publik dan kunci privat yang berbeda untuk algoritma *Rivest-Shamir-Adleman*

(RSA) dan iterasi maksimum yang berbeda untuk algoritma *Rubik's Cube*. Kemudian, percobaan dengan menggunakan 3 gambar dapat dilihat pada Lampiran 1. Pada proses enkripsi didapatkan nilai rata-rata *Structural Similarity Index Metrics* (SSIM) adalah 0,01 pada 18 pengujian dengan 3 digital iris mata yang sama. Hal tersebut menunjukkan bahwa citra digital iris mata yang dihasilkan pada proses enkripsi tidak identik atau sama dengan citra digital iris mata awal. Kemudian pada proses dekripsi didapatkan nilai rata-rata *Structural Similarity Index Metrics* (SSIM) adalah 1 untuk semua pengujian. Hal tersebut menunjukkan bahwa citra digital iris mata yang telah melalui proses enkripsi dan dekripsi identik atau sama dengan citra digital iris mata awal.

Selanjutnya dalam proses kriptografi, *Mean Square Error* (MSE) bisa berfungsi untuk mengetahui apakah proses enkripsi dan dekripsi memiliki keakuratan yang baik. Untuk nilai *Mean Square Error* (MSE) didapatkan hasil yang bervariasi pada proses enkripsi dengan nilai rata-rata adalah 37620,59 pada 18 pengujian dengan nilai kunci dan iterasi maksimum yang berbeda. Pada proses enkripsi, nilai kunci yang semakin besar dapat menghasilkan nilai *Mean Square Error* (MSE) yang semakin kecil. Tetapi, nilai yang ditunjukkan tetap menunjukkan nilai *Mean Square Error* (MSE) yang tinggi. Kemudian, dengan menambahkan nilai iterasi maksimum pada proses enkripsi, tidak terlalu menunjukkan perbedaan yang signifikan pada nilai *Mean Square Error* (MSE) yang dihasilkan. Pada proses dekripsi didapatkan nilai *Mean Square Error* (MSE) dengan rata- rata adalah 0. Hal tersebut menunjukkan bahwa citra yang telah terdekripsi dan citra digital iris mata awal adalah identik atau sama dan tidak memiliki nilai *error*.

4.3 Kajian Keislaman

Proses autentikasi pada smartphone membutuhkan tahapan untuk melakukan pengamanan data. Pengamanan data di dapatkan dengan memanfaatkan algoritma-algoritma yang ada pada ilmu kriptografi. Proses pengamanan data tersebut dinamakan dengan proses enkripsi dan dekripsi. Hal tersebut digunakan untuk melindungi privasi yang dimiliki oleh setiap individu agar tidak digunakan oleh individu-individu yang tidak bertanggung jawab untuk hal-hal yang tidak diinginkan. Dengan demikian hal tersebut berkaitan dengan anjuran untuk menghormati dan menghargai batas-batas privasi yang dimiliki individu lainnya.

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan pembahasan terhadap pengujian hasil enkripsi dan dekripsi citra digital iris mata sebelumnya:

1. Pada proses enkripsi dan dekripsi menggunakan algoritma *Rubik's Cube* dan algoritma *Rivest-Shamir-Adleman (RSA)* didapatkan bahwa besaran nilai kunci dan iterasi maksimum berbanding lurus dengan waktu yang dibutuhkan pada proses enkripsi dan dekripsi. Semakin besar nilai dari pasangan kunci pada algoritma *Rivest-Shamir-Adleman (RSA)* dan nilai iterasi maksimum dari algoritma *Rubik's Cube* maka waktu yang dibutuhkan untuk melakukan proses enkripsi dan dekripsi juga akan lebih lama. Dari hasil pengujian didapatkan rata-rata waktu 0,796 detik untuk proses enkripsi dan 0,652 detik untuk proses dekripsi menggunakan pasangan kunci publik (197,403) dan pasangan kunci privat (53,403) dan dengan iterasi maksimum = 1 untuk algoritma *Rubik's Cube.*
2. Pada proses enkripsi dan dekripsi menggunakan algoritma *Rubik's Cube* dan algoritma *Rivest-Shamir-Adleman (RSA)* didapatkan bahwa hasil citra digital terenkripsi yang acak dan tidak identik dengan citra digital iris mata awal. Dengan rata-rata nilai *Similarity Index Metrics* (SSIM) adalah 0,01 dan rata-rata nilai *Mean Square Error* (MSE) adalah 37620,59 pada proses enkripsi. Selanjutnya untuk hasil dari proses dekripsi, diperoleh nilai rata-rata *Similarity Index Metrics* (SSIM) adalah 1 dan nilai rata-rata *Mean Square Error* (MSE) adalah 0 yang membuktikan bahwa pada proses

dekripsi yang dilakukan berhasil mendapatkan citra digital iris mata yang identik dengan citra digital iris mata awal.

5.2 Saran untuk Penelitian Lanjutan

Melihat hasil yang telah diperoleh pada penelitian ini, terdapat saran yaitu melakukan pengembangan pada penelitian selanjutnya terkait bagaimana mendapatkan hasil enkripsi dengan menggunakan nilai kunci yang lebih besar, serta dapat dilakukan uji coba dengan menggunakan data citra digital yang lebih banyak. Selain itu, dapat dikembangkan terkait penelitian yang dapat menerapkan sistem yang telah dibuat pada perangkat yang dituju yaitu *smartphone*.

DAFTAR PUSTAKA

Arius, D. (2008). *PENGANTAR ILMU KRIPTOGRAFI Teori Analisis dan Implementasi* (S. Suyantoro, Ed.; I). CV ANDI OFFSET.

Belkaid, B. M. (2015). Meteosat Images Encryption based on AES and RSA Algorithms Meteosat Image Encryption. In *IJACSA) International Journal of Advanced Computer Science and Applications* (Vol. 6, Issue 6).

Cahyanti, M., Salim, R. A., & Wisuda, M. (2016). IMPLEMENTASI PENGOLAHAN CITRA UNTUK PENGENALAN CITRA BENDERA NEGARA BERDASARKAN WARNA. In *Seminar Riset Teknologi Informasi (SRITI) tahun*.

Ginting, A., Isnanto, R. R., & Windasari, I. P. (2015). Implementasi Algoritma Kriptografi RSA untuk Enkripsi dan Dekripsi Email. *Jurnal Teknologi Dan Sistem Komputer*, *3*, 253–258.

Hadits riwayat Muslim no. 4016.

Haekal Al-Fadillah, R., & Djiwadukusumah, F. (2019). *IMPLEMENTASI KRIPTOGRAFI RUBIK CUBE PADA MEDIA GAMBAR PNG*.

Julus, L. J. (2022). *A Secured Image Encryption Algorithm Based On Rubik's Cube Principle with RSA Encryption*. www.ijcrt.org

Kementrian Agama Republik Indonesia. (2022). *Al-Qur'an dan terjemahan*.

Khairil Azhar, J., & Yuliany, S. (2019). *Implementasi Algoritma RSA (Rivest, Shamir dan Adleman) untuk Enkripsi dan Dekripsi File .pdf*.

Kristanto, E. G., Rompas, E., & Wangko, S. (2013). Identifikasi Iris Opsi Identifikasi Iris. *Jurnal Biomedik*, *5*, S7-11.

Kromodimoeljo, S. (2009). *Teori dan Aplikasi Kriptografi*. SPK IT Consulting.

Kusumanto, R. D., & Tompunu, A. N. (2011). PENGOLAHAN CITRA DIGITAL UNTUK MENDETEKSI OBYEK MENGGUNAKAN PENGOLAHAN WARNA MODEL NORMALISASI RGB. In *Seminar Nasional Teknologi Informasi & Komunikasi Terapan*.

Loukhaoukha, K., Chouinard, J. Y., & Berdai, A. (2012). A secure image encryption algorithm based on Rubik's cube principle. *Journal of Electrical and Computer Engineering*. https://doi.org/10.1155/2012/173931

Male, G. M., Wirawan, & Setijadi, E. (2012). *ANALISA KUALITAS CITRA PADA STEGANOGRAFI UNTUK APLIKASI e-GOVERNMENT*.

Nader, J., Alsadoon, A., Prasad, P. W. C., Singh, A. K., & Elchouemi, A. (2015). Designing Touch-Based Hybrid Authentication Method for Smartphones. *Procedia Computer Science*, *70*, 198–204. https://doi.org/10.1016/j.procs.2015.10.072

Sara, U., Akter, M., & Uddin, M. S. (2019). Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study. *Journal of Computer and Communications*, *07*(03), 8–18. https://doi.org/10.4236/jcc.2019.73002

Shihab, Moh. Quraish. (2000). *Tafsir Al-Mishbah : pesan, kesan dan keserasian Al-Qur'an*. Lentera Hati.

Sulistiyanti, S. R., Setyawan, F. A., & Komarudin, M. (2016). *PENGOLAHAN CITRA; DASAR DAN CONTOH PENERAPANNYA*.

Sumarna, H. B., Utami, E., & Hartanto, A. D. (2020). Tinjauan Literatur Sistematik tentang Structural Similarity Index Measure untuk Deteksi Anomali Gambar Systematic Literature Review of Structural Similarity Index Measure for Image Anomaly Detection. *Citec Journal*, *7*(2).

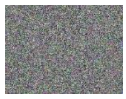
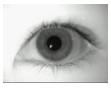
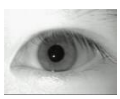
Sutoyo, T., Mulyanto, E., Suhartono, V., Nurhayati, O. D., & Wijanarto. (2009). *Teori Pengolahan Citra Digital*. Andi Offset.

Vatsal, S., & Dwivedi, Mr. S. S. (2018). *Advanced IRIS Recognition System: A Review*.

Widiasari, A. (2014). *Implementasi Algoritma Kriptografi RSA Pada Aplikasi Smart Card*.

LAMPIRAN

Lampiran 1 Tabel pengujian proses enkripsi dan dekripsi

No.	Citra Digital Awal	Hasil Enkripsi	Nilai SSIM Enkripsi	Nilai MSE Enkripsi	Hasil Dekripsi	Nilai SSIM Dekripsi	Nilai MSE Dekripsi
1.	aeval1_1 		0,01	43840,11		1	0
2.	aeval1_2 		0,01	38397,26		1	0
3.	aeval1_3 		0,01	40326,03		1	0
4.	aeval1_4 		0,01	43832		1	0
5.	aeval1_5 		0,01	44110,59		1	0
6.	aeval1_6 		0,01	43936,16		1	0
7.	bryanl1_1 		0,01	40240,59		1	0
8.	bryanl1_2 		0,01	35417,59		1	0
9.	bryanl1_3 		0,01	36803,85		1	0
10.	bryanl1_4 		0,01	40198,13		1	0

11.	bryanl1_5 		0,01	40361,82		1	0
12.	bryanl1_6 		0,01	40425,06		1	0
13.	chingycl1_1 		0,01	32192,64		1	0
14.	chingycl1_2 		0,01	29382,07		1	0
15.	chingycl1_3 		0,01	30542,28		1	0
16.	chingycl1_4 		0,01	32315,03		1	0
17.	chingycl1_5 		0,01	32459,71		1	0
18.	chingycl1_6 		0,01	32389,87		1	0

Lampiran 2 *Script* enkripsi algoritma *Rubik's Cube*

```
for iterations in range(ITER_MAX):
    for i in range(n):
        rTotalSum = sum(r[i])
        gTotalSum = sum(g[i])
        bTotalSum = sum(b[i])
        rModulus = rTotalSum % 2
        gModulus = gTotalSum % 2
        bModulus = bTotalSum % 2
        if(rModulus==0):
            r[i] = numpy.roll(r[i],Kr[i])
        else:
            r[i] = numpy.roll(r[i],-Kr[i])
        if(gModulus==0):
            g[i] = numpy.roll(g[i],Kr[i])
        else:
            g[i] = numpy.roll(g[i],-Kr[i])
        if(bModulus==0):
            b[i] = numpy.roll(b[i],Kr[i])
        else:
            b[i] = numpy.roll(b[i],-Kr[i])
    for j in range(m):
        rTotalSum = 0
        gTotalSum = 0
        bTotalSum = 0
        for i in range(n):
            rTotalSum += r[i][j]
            gTotalSum += g[i][j]
            bTotalSum += b[i][j]
        rModulus = rTotalSum % 2
        gModulus = gTotalSum % 2
        bModulus = bTotalSum % 2
        if(rModulus==0):
            upshift(r,j,Kc[j])
        else:
            downshift(r,j,Kc[j])
        if(gModulus==0):
            upshift(g,j,Kc[j])
        else:
            downshift(g,j,Kc[j])
        if(bModulus==0):
            upshift(b,j,Kc[j])
        else:
            downshift(b,j,Kc[j])
    for i in range(n):
        for j in range(m):
            if(i%2==1):
                r[i][j] = r[i][j] ^ Kc[j]
                g[i][j] = g[i][j] ^ Kc[j]
                b[i][j] = b[i][j] ^ Kc[j]
            else:
                r[i][j] = r[i][j] ^ rotate180(Kc[j])
                g[i][j] = g[i][j] ^ rotate180(Kc[j])
                b[i][j] = b[i][j] ^ rotate180(Kc[j])
    for j in range(m):
        for i in range(n):
            if(j%2==1):
                r[i][j] = r[i][j] ^ Kr[i]
```

Lampiran 3 *Script* enkripsi algoritma RSA

```
enc =[[0 for x in range(m)] for y in range(n)]
for i in range(n):
    for j in range(m):
        r,g,b = pix[j,i]
        r1 = enkripsiRSA(r,e,modn)
        g1 = enkripsiRSA(g,e,modn)
        b1 = enkripsiRSA(b,e,modn)
        enc[i][j] = [r1,g1,b1]
```

Lampiran 4 *Script* dekripsi algoritma RSA

```
dec = [[0 for x in range(m)] for y in range(n)]
for i in range(n):
    for j in range(m):
        r,g,b = enc[i][j]
        r1 = dekripsiRSA(r,d,modn)
        g1 = dekripsiRSA(g,d,modn)
        b1 = dekripsiRSA(b,d,modn)
        dec[i][j] = [r1,g1,b1]
```

Lampiran 5 *Script* dekripsi algoritma *Rubik's Cube*

```
for iterations in range(ITER_MAX):
    for j in range(m):
        for i in range(n):
            if(j%2==1):
                r[i][j] = r[i][j] ^ Kr[i]
                g[i][j] = g[i][j] ^ Kr[i]
                b[i][j] = b[i][j] ^ Kr[i]
            else:
                r[i][j] = r[i][j] ^ rotate180(Kr[i])
                g[i][j] = g[i][j] ^ rotate180(Kr[i])
                b[i][j] = b[i][j] ^ rotate180(Kr[i])
    for i in range(n):
        for j in range(m):
            if(i%2==1):
                r[i][j] = r[i][j] ^ Kc[j]
                g[i][j] = g[i][j] ^ Kc[j]
                b[i][j] = b[i][j] ^ Kc[j]
            else:
                r[i][j] = r[i][j] ^ rotate180(Kc[j])
                g[i][j] = g[i][j] ^ rotate180(Kc[j])
                b[i][j] = b[i][j] ^ rotate180(Kc[j])
    for j in range(m):
        rTotalSum = 0
        gTotalSum = 0
        bTotalSum = 0
        for i in range(n):
            rTotalSum += r[i][j]
            gTotalSum += g[i][j]
            bTotalSum += b[i][j]
        rModulus = rTotalSum % 2
        gModulus = gTotalSum % 2
        bModulus = bTotalSum % 2
        if(rModulus==0):
            downshift(r,j,Kc[j])
        else:
            upshift(r,j,Kc[j])
        if(gModulus==0):
            downshift(g,j,Kc[j])
        else:
            upshift(g,j,Kc[j])
        if(bModulus==0):
            downshift(b,j,Kc[j])
        else:
            upshift(b,j,Kc[j])
    for i in range(n):
        rTotalSum = sum(r[i])
        gTotalSum = sum(g[i])
        bTotalSum = sum(b[i])
        rModulus = rTotalSum % 2
        gModulus = gTotalSum% 2
        bModulus = bTotalSum % 2
        if(rModulus==0):
            r[i] = numpy.roll(r[i],-Kr[i])
        else:
            r[i] = numpy.roll(r[i],Kr[i])
        if(gModulus==0):
            g[i] = numpy.roll(g[i],-Kr[i])
        else:
            g[i] = numpy.roll(g[i],Kr[i])
        if(bModulus==0):
            b[i] = numpy.roll(b[i],-Kr[i])
        else:
            b[i] = numpy.roll(b[i],Kr[i])
```

RIWAYAT HIDUP

Siti Habibatul Ma'rifah lahir di Kota Gresik pada 29 Mei 2000. Memiliki nama panggilan Ifah. Tempat tinggal di Jalan Kapten Darmo Sugondo No.3 Gg.IV RT 002 RW 001, Desa Karangkiring, Kecamatan Kebomas, Kabupaten Gresik, Jawa Timur. Merupakan anak ketiga dari tiga bersaudara dari Bapak Mochamad Mas'ud dan Ibu Nur Khasanah.

Pendidikan yang pernah ditempuh yaitu TK Islam Desa Karangkiring. Kemudian melanjutkan sekolah di SDN Karangkiring dan lulus pada tahun 2012. Menempuh pendidikan SMP di Sekolah Menengah Pertama Negeri 3 Gresik dan lulus pada tahun 2015. Melanjutkan pendidikan di Sekolah Menengah Atas Negeri 1 Kebomas Gresik dan lulus pada tahun 2018.

Tahun 2018 melanjutkan studi ke jenjang pendidikan strata 1 di Universitas Islam Negeri Maulana Malik Ibrahim Malang dan mengambil program studi Matematika di Fakultas Sains dan Teknologi. Aktif mengikuti kegiatan intra kampus HMJ Integral Matematika UIN Malang (2019-2020).

Kegiatan yang pernah diikuti yaitu Tim IT Webinar Nasional Kompetisi Matematika (2020-2021), Tim Streaming Fakultas Sains dan Teknologi UIN Malang Tahun 2021, Praktek Kerja Lapangan (PKL) di Dinas Komunikasi dan Informatika Kota Malang bidang Aptika Tahun 2021.

KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

BUKTI KONSULTASI SKRIPSI

Nama : Siti Habibatul Ma'rifah
NIM : 18610075
Fakultas / Program Studi : Sains dan Teknologi / Matematika
Judul Skripsi : Implementasi Algoritma *Rubik's Cube* dan Algoritma *Rivest-Shamir-Adleman* (RSA) pada Pengamanan Citra Digital iris Mata
Pembimbing I : Muhammad Khudzaifah, M.Si
Pembimbing II : Erna Herawati, M.Pd

No	Tanggal	Hal	Tanda Tangan
1.	31 Desember 2021	Konfirmasi Bimbingan Proposal Skripsi	1.
2.	11 Januari 2022	Bimbingan Bab 1	2.
3.	13 Januari 2022	Bimbingan Bab 1 (Kajian Agama)	3.
4.	20 Januari 2022	Bimbingan Bab 2	4.
5.	1 Februari 2022	Bimbingan Bab 2 (Kajian Agama)	5.
6.	7 Februari 2022	Bimbingan Bab 2	6.
7.	10 Februari 2022	Bimbingan Bab 2	7.
8.	23 Februari 2022	Bimbingan Bab 3	8.
9.	25 Februari 2022	Bimbingan Bab 3	9.
10.	31 Maret 2022	Bimbingan Bab 3	10.
11.	14 April 2022	Acc Pendaftaran Seminar Proposal	11.
12.	22 April 2022	Bimbingan Bab 4 (Kajian Agama)	12.
13.	31 Mei 2022	Bimbingan Bab 4,5	13.
14.	6 Juni 2022	Acc Seminar Hasil	14.
15.	10 Juni 2022	Bimbingan Revisi Seminar Hasil (Kajian Agama)	15.
16.	14 Juni 2022	Bimbingan Revisi Seminar Hasil	16.
17.	16 Juni 2022	Acc Bab 4,5	17.

Malang, 24 Juni 2022
Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc
NIP.19741129200012200 5