APLIKASI ENKRIPSI DAN DEKRIPSI DATA GAMBAR MENGGUNAKAN ALGORITMA *ADVANCED ENCRYPTION STANDARD*

OLEH

MUH FAJRIL A. RABI'U

T3120009

SKRIPSI

Untuk memenuhi salah satu syarat ujian
guna memperoleh gelar Sarjana

PROGRAM SARJANA

TEKNIK INFORMATIKA

UNIVERSITAS ICHSAN GORONTALO

GORONTALO

2024

PERSETUJUAN SKRIPSI

APLIKASI ENKRIPSI DAN DEKRIPSI DATA GAMBAR MENGGUNAKAN ALGORTIMA *ADVANCED ENCRYPTION STANDARD*

OLEH

MUH FAJRIL A. RABI'U

T3120009

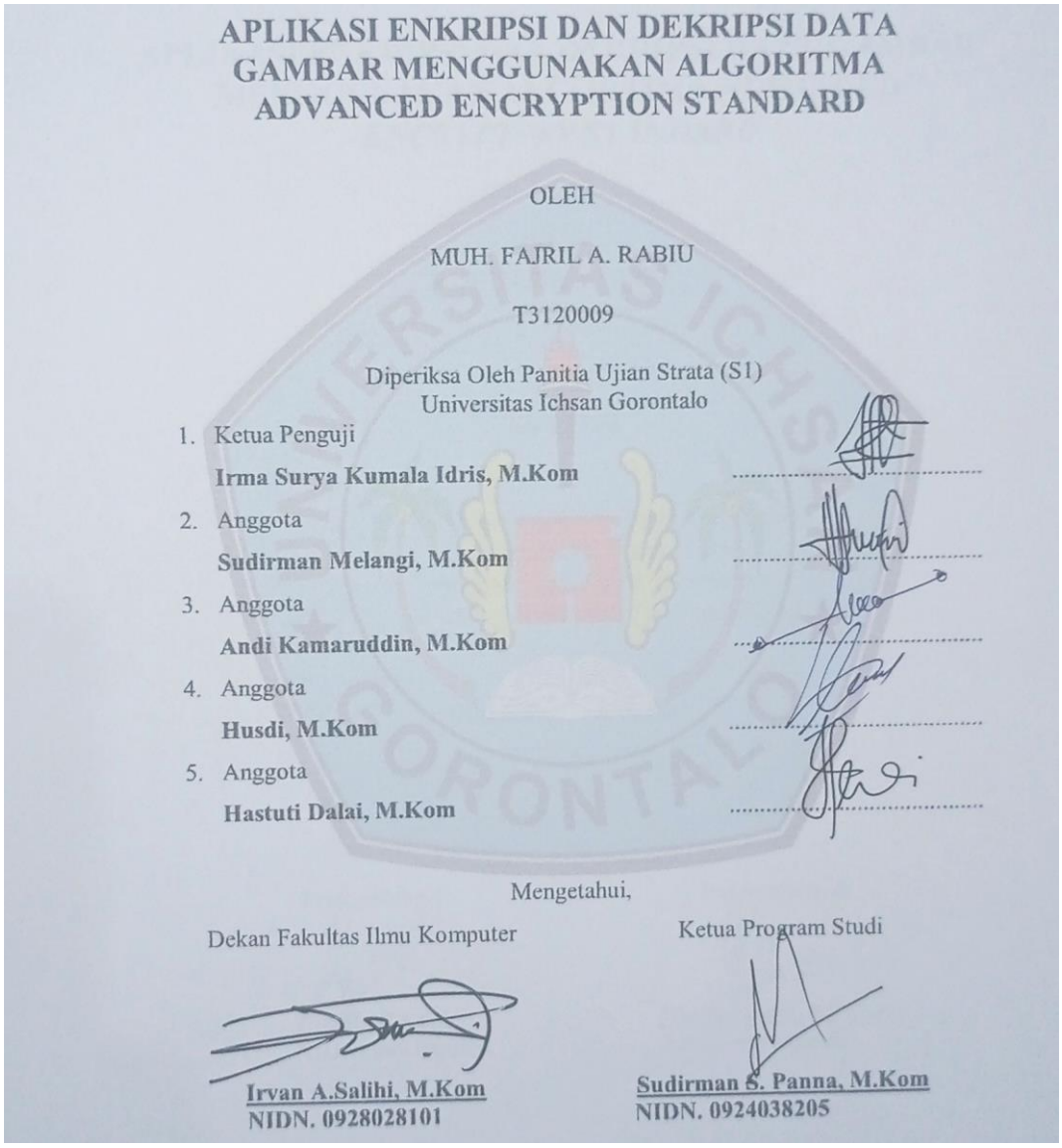
SKRIPSI

Untuk memenuhi salah satu syarat ujian
guna memperoleh gelar Sarjana
Program Studi Teknik Informatika,
ini telah disetujui oleh Tim Pembimbing

Gorontalo, Juni 2024

Pembimbing I	Pembimbing II
Huzdi, M.Kom	**Hastuti Dalai, M.Kom**
NIDN: 0907108701	**NIDN: 0918038803**

PENGESAHAN SKRIPSI



APLIKASI ENKRIPSI DAN DEKRIPSI DATA GAMBAR MENGGUNAKAN ALGORITMA ADVANCED ENCRYPTION STANDARD

OLEH

MUH. FAJRIL A. RABIU

T3120009

Diperiksa Oleh Panitia Ujian Strata (S1)
Universitas Ichsan Gorontalo

1. Ketua Penguji
   Irma Surya Kumala Idris, M.Kom .....
2. Anggota
   Sudirman Melangi, M.Kom .....
3. Anggota
   Andi Kamaruddin, M.Kom .....
4. Anggota
   Husdi, M.Kom .....
5. Anggota
   Hastuti Dalai, M.Kom .....

Mengetahui,

Dekan Fakultas Ilmu Komputer

Irvan A.Salihi, M.Kom
NIDN. 0928028101

Ketua Program Studi

Sudirman S. Panna, M.Kom
NIDN. 0924038205

PERNYATAAN SKRIPSI

Dengan ini saya menyatakan bahwa:

1. Karya tulis (Skripsi) saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar
   Akademik (sarjana) baik di Universitas Ichsan Gorontalo maupun pergururan tinggi lainnya.
2. Karya tulis (Skripsi) saya ini adalah murni gagasan, rumusan dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan tim pembimbing.
1. Dalam karya tulis (Skripsi) saya ini tidak terdapat karya yang telah dipublikasikan orang lain, kecuali secara tertulis di cantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
2. Pernyataan ini saya buat dengan sesungguhnya dan apa bila kemudian hari terdapat penyimpangan dan ketidak benaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena skripsi ini, serta sanksi lainnya sesuai dengan norma yang berlaku di perguruan tinggi ini.

Gorontalo, Juni 2024

Yang membuat pernyataan,

Muh Fajril A. Rabiu

ABSTRACT

MUH FAJRIL A. RABIU. T3120009. IMAGE DATA ENCRYPTION AND DECRYPTION APPLICATION USING ADVANCED ENCRYPTION STANDARD ALGORITHM

Digital data security is becoming increasingly important in today's information age, including image data protection. This research aims to develop an image data encryption and decryption application using the Advanced Encryption Standard (AES) algorithm to improve data security. To further strengthen the security of private keys, the RSA algorithm is used in securing the private keys used in the encryption and decryption process. The application of the AES algorithm on imagedata shows that this method effectively maintains data confidentiality and integrity.The research results confirm that the AES algorithm can be effectively applied for image data security. The RSA algorithm used to secure the private key is performedto strengthen the security, making it more difficult for cryptanalysts to obtain the private key from the AES algorithm used. Application testing is conducted using White-Box and Base Path methods which resulted in a value of V(G) = CC = 6, indicating well-managed complexity. In addition, Black-Box testing shows the correctness of the application logic, proving that the logic flowchart used in the application is correct and reliable. It means that the application developed can be operated effectively and, in practice, is feasible to protect image data from securitythreats.

Keywords: encryption, decryption, image data security, AES algorithm, RSA Algorithm.

ABSTRAK

MUH FAJRIL A. RABIU. T3120009. APLIKASI ENKRIPSI DAN DEKRIPSI DATA GAMBAR MENGGUNAKAN ALGORITMA ADVANCED ENCRYPTION STANDARD

Keamanan data digital menjadi semakin penting dalam era informasi saat ini, termasuk dalam perlindungan data gambar. Penelitian ini bertujuan untuk mengembangkan aplikasi enkripsi dan dekripsi data gambar menggunakan algoritma Advanced Encryption Standard (AES) untuk meningkatkan keamanan data. Untuk lebih memperkuat keamanan kunci pribadi, algoritma RSA digunakan dalam mengamankan kunci pribadi yang digunakan dalam proses enkripsi dan dekripsi. Penerapan algoritma AES pada data gambar menunjukkan bahwa metode ini efektif dalam menjaga kerahasiaan dan integritas data. Hasil penelitian ini menegaskan bahwa algoritma AES dapat diterapkan secara efektif untuk keamanan data gambar. Penggunaan algoritma RSA untuk mengamankan kunci pribadi terbukti memperkuat keamanan, membuatnya lebih sulit bagi kriptanalis untuk mendapatkan kunci pribadi dari algoritma AES yang digunakan. Uji coba aplikasi dilakukan dengan menggunakan metode White Box dan Basis Path yang menghasilkan nilai V(G) = CC = 6, menunjukkan kompleksitas yang dikelola dengan baik. Selain itu, pengujian Black Box menunjukkan kebenaran logika aplikasi, membuktikan bahwa flowchart logika yang digunakan dalam aplikasi adalah benar dan dapat diandalkan. Hasil ini menunjukkan bahwa aplikasi yang dikembangkan dapat dioperasikan dengan efektif dan layak digunakan dalam praktik untuk melindungi data gambar dari ancaman keamanan.

Kata kunci: Enkripsi, Dekripsi, Keamanan Data Gambar, Algoritma AES, Algoritma RSA.

KATA PENGANTAR

Alhamdulillah, segala puji bagi Allah SWT atas rahmat, dan hidayah nya sehingga penulis dapat menyelesaikan skripsi ini dengan judul: "Aplikasi Enkripsi dan Dekripsi Data Gambar Menggunakan Algoritma Advanced Encryption Standard". Shalawat serta salam selalu tercurah kepada Nabi Muhammad SAW yang telah membawa umat manusia dari zaman kegelapan dan kebodohan menuju zaman yang penuh cahaya dan kemajuan ilmu pengetahuan dan teknologi. Skripsi ini disusun untuk memenuhi salah satu syarat penyusunan Skripsi Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo

Penulis menyadari sepenuhnya bahwa skripsi ini tidak mungkin terwujud tanpa bantuan dan dorongan dari berbagai pihak, baik bantuan moral maupun material. Untuk itu, dengan segala keikhlasan dan kerendahan hati, penulis mengucapkan banyak terima kasih dan penghargaan yang setinggi-tingginya kepada:

1. Ibu Dr. Yuriko Abdulsamad, M.Si, selaku Ketua Yayasan Pengembangan Ilmu Pengetahuan dan Teknologi (YPIPT) Ichsan Gorontalo;
2. Bapak Dr. Abdul Gaffar La Tjokke, M.Si, selaku Rektor Universitas Ichsan Gorontalo;
3. Bapak Irvan Abrahman Salihi S.Kom.,M.Kom, selaku Dekan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
4. Bapak Sudirman Melangi, M.Kom, selaku Pembantu Dekan I Bidang Akademik Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
5. Ibu Irma Surya Kumala Idris, M.Kom, selaku Pembantu Dekan II Bidang Administrasi Umum dan Keuangan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
6. Bapak Sudirman S. Panna, S.Kom, M.Kom, selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
7. Bapak Husdi, M.kom, selaku Pembimbing utama yang telah membimbing penulis selama penyusunan penelitian ini.

8. Ibu Hastuti Dalai, M.Kom, selaku Pembimbing pendamping yang telah membimbing penulis selama penyusunan penelitian ini.
9. Bapak dan Ibu Dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan berbagai disiplin ilmu kepada penulis;
10. Kedua Orang Tua saya yang tercinta, atas segala kasih sayang, jerih payah dan doa restunya dalam membesarkan dan mendidik penulis;
11. Rekan-rekan seperjuangan yang telah banyak memberikan bantuan dan dukungan moral ya ng sangat besar kepada penulis;
12. Kepada semua pihak yang ikut membantu dalam penyelesaian skripsi ini yang tak sempat penulis sebutkan satu-persatu.

Semoga Allah, SWT melimpahkan balasan atas jasa-jasa mereka kepada kami. Penulis menyadari sepenuhnya bahwa apa yang telah dicapai ini masih jauh dari kesempurnaan dan masih banyak terdapat kekurangan. Oleh karena itu, penulis sangat mengharapkan adanya kritik dan saran yang konstruksi. Akhirnya penulis berharap semoga hasil yang telah dicapai ini dapat bermanfaat bagi kita semua, Aamiin.

Gorontalo, /Juni/2024

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dalam era digital yang semakin maju ini, keamanan data telah menjadi isu yang sangat penting. Dalam lingkungan yang terhubung secara digital, data yang disimpan dan ditransmisikan rentan terhadap ancaman dan serangan yang dapat mengakibatkan kerugian signifikan [1], seperti kerugian finansial dan privasi. Agar keamanannya terjaga maka ilmu yang dikembangkan untuk menjaga keamanan data tersebut adalah kriptografi. Kriptografi adalah suatu ilmu yang mempelajari bagaimana cara menjaga agar data atau pesan tetap aman saat dikirimkan, dari pengirim ke penerima tanpa mengalami gangguan dari pihak ketiga [2].

Berdasarkan jenis kunci yang digunakan, kriptografi terbagi atas dua metode, yaitu kriptografi kunci simetris dan kriptografi kunci asimetris. Perbedaan dari kedua kriptografi ini terletak pada penggunaan kunci [3]. Untuk kriptografi simetris menggunakan kunci yang sama pada saat melakukan enkripsi dan dekripsi, salah satu algoritma simetris yaitu AES (*Advanced Encryption Satndard*) [4]. Oleh sebab itu harus benar-benar dijaga kerahasian kunci tersebut, namun berbeda halnya dengan kriptografi asimetris, kunci pada saat enkripsi berbeda dengan kunci yang digunakan pada saat melakukan dekripsi, hal ini menjadi salah satu faktor kriptografi asimetris lebih aman dibandingkan dengan kriptografi simetris [5], salah satu algoritma asimetris yaitu RSA.

AES digunakan sebagai standar algoritma kriptografi terbaru yang dipublikasikan oleh NIST (*National Institute of Standard and Technology*) sebagai pengganti algoritma DES (*Data Encryption Standard*) yang sudah berakhir masa penggunaannya [6]. Algoritma AES adalah algoritma kriptografi yang dapat mengenkripsi dan mendekripsi data dengan panjang kunci yang

bervariasi, yaitu 128 bit, 192 bit, dan 256 bit [4]. Sedangkan Algoritma asimetris yang paling populer adalah RSA . Algoritma RSA dibuat oleh 3 orang peneliti dari MIT (*Massachussets institute of Technology*) pada tahun 1976, yaitu Rivets, Shamir, dan Adleman. Algoritma ini menggunakan pemfaktoran bilangan yang sangat besar, Oleh karena alasan tersebut RSA dianggap aman [7].

Pada penelitian yang dilakukan oleh Faturungi Muharram, Huzain Azis, dan Abdul Rachman Manga dengan judul **"**Analisis Algoritma pada Proses Enkripsi dan Dekripsi File Menggunakan Advanced Encryption Standard (AES)**"**, dengan tujuan mengamankan data gambar menggunakan algoritma AES. Dari penelitian ini gambar berhasil dienkripsi dan dikembalikan ke bentuk aslinya, waktu yang dibutuhkan saat enkripsi dan dekripsi dipengaruhi oleh besarnya ukuran data. Penelitian yang dilakukan oleh Steven Richardo Siburian [8], tentang kombinasi AES dan RSA untuk enkripsi teks pesan, pada penelitian ini di buat sebuah program python untuk mengimplementasikan kombinasi AES dan RSA untuk enkripsi teks. Selanjutnya penelitian yang dilakukan oleh Muhamad Andra Fahreza dan Arif Harbani dengan judul Aplikasi Keamanan Data Gambar Menggunakan Algoritma RSA (Rivest Shamir Adleman) Berbasis Desktop. Algortima RSA digunakan untuk mengamankan file gambar.

Algoritma AES memiliki kelemahan Dari segi jenis kunci yang simetris dimana pengirim dan penerima data memiliki kunci yang sama untuk setiap proses pengiriman-penerimaan data, hal ini akan menyebabkan kunci mudah bocor meskipun dalam waktu yang lama [9]. Untuk meningkatkan keamanan dalam proses enkripsi dan dekripsi data gambar maka dilakukan kombinasi algortima simetris (AES) dan asimetris (RSA). AES di gunakan untuk mengenkripsi data atau gambar dengan kunci simetris yang cepat dan efisien, sedangkan pada RSA di gunakan untuk mengenkripsi kunci simetris tersebut dengan kunci public penerima. Dengan kombinasi ini gambar dapat diamankan dengan enkripsi yang kuat saat pengiriman dan hanya pemegang

kunci privat yang dapat mengakses dan mendekripsi gambar tersebut [8]. Dengan kombinasi algortima AES dan RSA, suatu informasi akan menjadi lebih sulit untuk diketahui oleh orang yang tidak berhak. Kemanan tersebut di perlukan untuk menghindari adanya penyadapan atau pembajakan gambar yang mengandung informasi penting bagi penggunanya. Keamanan diperlukan untuk menjaga integritas gambar tersebut agar tetap aman [7].

Berdasarkan berbagai pemaparan di atas, maka peneliti tertarik melakukan penelitian dengan judul **"APLIKASI ENKRIPSI DAN DEKRIPSI DATA GAMBAR MENGGUNAKAN ALGORITMA *ADVANCED ENCRYPTION STANDARD*"**. Pada penelitian ini aplikasi dibuat dalam bentuk website dan file gambar yang akan digunakan dengan format JPG,JPEG dan PNG.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah disebutkan diatas, dapat diidentifikasi masalah yaitu, algoritma AES keamanannya masih lemah sehingga perlu ditingkatkan keamanannya menggunakan algoritma RSA untuk enkripsi dan dekripsi data gambar.

1.3 Rumusan Masalah

Berdasarkan identifikasi masalah yang telah diuraikan diatas, maka dapat dirumuskan permasalahan dalam penelitian ini adalah Bagaimana penerapan algoritma AES dan RSA untuk keamanan data gambar.

1.4 Tujuan Penelitian

Sesuai dengan rumusan masalah diatas, maka tujuan yang akan dicapai dalam penelitian ini adalah Menerapkan algoritma AES dan RSA untuk enkripsi dan dekripsi data gambar.

1.5 Manfaat Penelitian

1.5.1 Manfaat Teoritis

Hasil penelitian ini diharapkan mampu memberikan masukan bagi perkembangan ilmu pengetahuan dan teknologi khususnya ilmu komputer, berupa manfaat dalam mengamankan data.

1.5.2 Manfaat Praktis

Diharapkan hasil penelitian yang dibuat ini dapat memberikan masukan atau reverensi serta menambah pengetahuan bagi mahasiswa pada umumnya dan khususnya bagi program Studi Fakultas Ilmu Komputer.

BAB II

LANDASAN TEORI

2.1 Tinjauan Studi

Tabel 2. 1 Rangkuman Tinjauan Studi

No	Peneliti	Judul	Tahun	Metode	Hasil
1	F. Muharram, H. Azis, dan A. R. Manga	Analisis Algoritma pada Proses Enkripsi dan Dekripsi File Menggunakan Advanced Encryption Standard (AES)	2018	Advanced Encryption Standard (AES)	Dari hasil uji coba pada proses enkripsi dan dekripsi bahwa file yang melalui uji coba enkripsi akan berubah bentuk menjadi file yang tak bias dibaca, file dapat kembali kebentuk asli jika melalui proses dekripsi dengan menggunakan kunci yang sama saat enkripsi. Dan waktu proses hasil enkripsi-dekripsi data dapat dipengaruhi oleh besar ukuran data yang akan di uji.
2	Muhamad Andra Fahreza dan Arif Harbani	Aplikasi Keamanan Data Gambar Menggunakan Algoritma RSA (Rivest Shamir Adleman) Berbasis Desktop	2019	RSA	Aplikasi keamanan data gambar menggunakan algortima RSA mendapatkan presentase kelayakan sebesar 84.1% yang di berikan para responden

No	Peneliti	Judul	Tahun	Metode	Hasil
3	S. R. Siburian, R. Alek, S. Sinaga, dan F. Yudistira	Kriptosistem Hybrid Menggunakan Kombinasi Aes Dan Rsa Untuk Enkripsi Teks Pesan	2023	Advanced Encryption Standard (AES) dan RSA	AES digunakan untuk mengenkripsi pesan menggunakan kunci simetris yang cepat dan efisien, sedangkan RSA digunakan untuk mengenkripsi kunci simetris tersebut dengan kunci penerima publik. Dengan adanya Kriptosistem Hybrid, pesan dapat diamankan dengan enkripsi yang kuat selama pengiriman, dan hanya pemegang kunci privat yang dapat mengakses dan mendekripsi pesan tersebut.

2.2 Tinjauan Pustaka

2.2.1 Advanced Encryption Standard (AES)

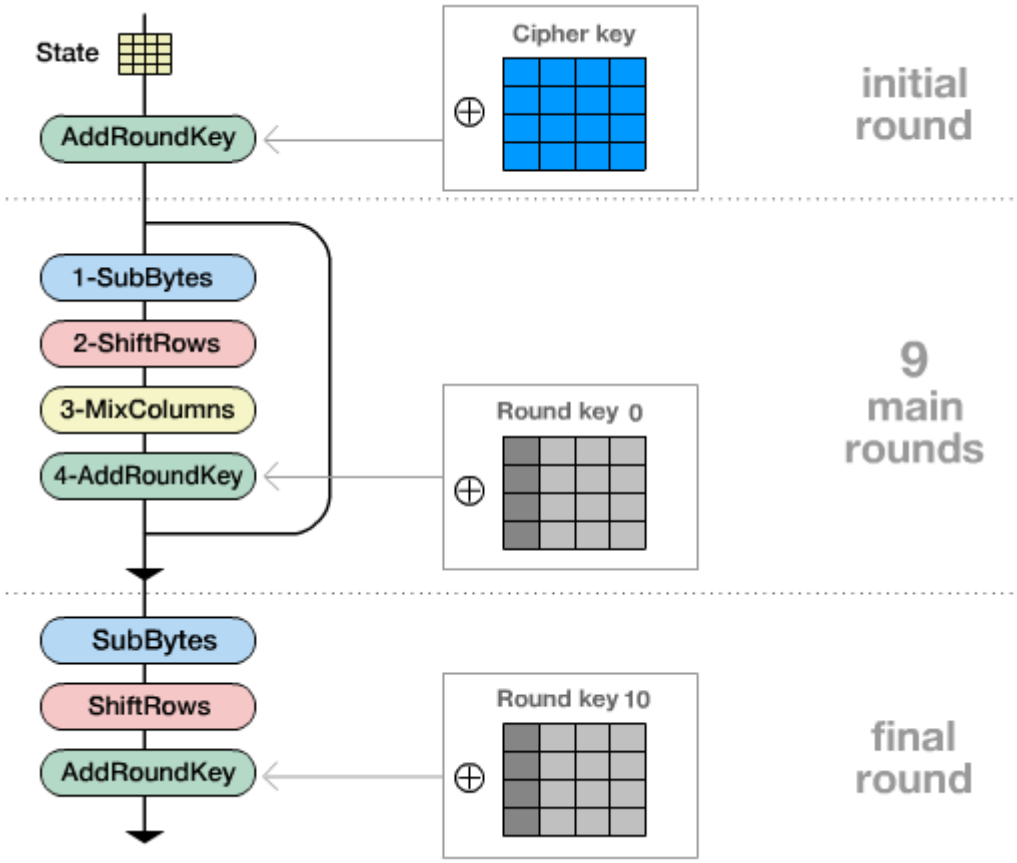
Algoritma AES merupakan algoritma simetris yaitu mengunakan kunci yang sama untuk proses enkripsi dan dekripsi. Algoritma AES memiliki tiga pilihan kunci yaitu tipe: AES-128, AES-192 dan AES-256 [7]. *Advanced Encryption Standard* (AES) dipublikasikan oleh NIST (*National Institute of Standard and Tchnology*) pada tahun 2001. AES merupakan blok kode simetris untuk mengantikan DES (*Data Encryption Standard*) [4]. AES adalah algoritma kriptografi yang dapat mengenkripsi dan mendekripsi data dengan panjang kunci yang bervariasi, yaitu 128 bit, 192 bit, dan 256 bit [7]. Perbedaan panjang kunci ini yang nantinya mempengaruhi jumlah putaran pada algoritma AES ini. Jumlah putaran yang digunakan algoritma ini ada tiga macam seperti pada Tabel 2.2.

Tabel 2. 2: Parameter AES [4].

Algoritma	Jumlah Key (Nk)	Ukuran Block (Nb)	Jumlah Putaran (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

2.2.1.1 Proses Enkripsi AES

Proses enkripsi algoritma AES terdiri dari 4 jenis transformasi bytes, yaitu SubBytes, ShiftRows, Mixcolumns dan AddRoundKey [7] . Ilustrasi proses enkripsi AES dapat digambarkan seperti pada Gambar 2.1.



Gambar 2. 1: Ilustrasi Proses Enkripsi AES [10].

1) *AddRoundKey*

Pada proses AES AddRoundKey, sebuah round key ditambahkan pada state dengan operasi XOR. Setiap key terdiri dari Nb word dimana tiap word tersebut akan dijumlahkan dengan word atau kolom yang bersesuaian dari state sehingga menjadi persamaan (1) .

[S'o, c, s' 1,c, s'2,c, s'3,c] = [s0,c, s1,c,s3,c] Φ [W round*Nb+c] (1)

Untuk $0 \leq c \leq Nb$

. [W_1] adalah world dari key yang bersesuaian dimana i = round * Nb+c. Transformasi AddRoundKey pada proses enkripsi pertama kali pada round = 0 untuk round selanjutnya = round + 1, pada proses dekripsi pertama kali pada round = 14 untuk round selanjutnya round = round -1.

2) *SubBytes*

SubBytes merupakan transformasi byte dimana setiap elemen pada state akan dipetakan dengan menggunakan sebuah tabel substitusi (S-Box). Tabel substitusi S-Box akan dipaparkan dalam gambar 2.

hex	y															
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

x

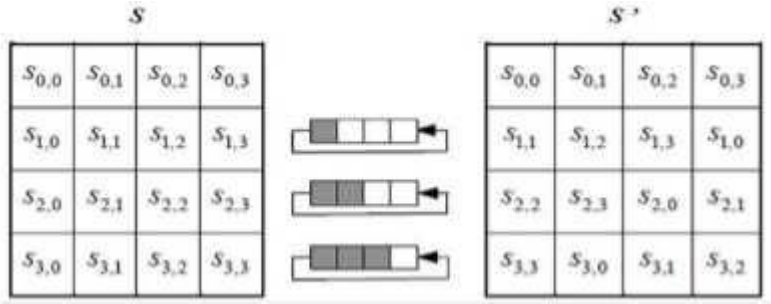
Gambar 2. 2: Tabel S-Box SubBytes [7]**.**

3) *ShiftRows*

Transformasi Shiftrows pada dasarnya adalah proses pergeseran bit dimana bit paling kiri akan dipindahkan menjadi bit paling kanan (rotasi bit).

4) *MixColumns*

Mixcolumns mengoperasikan setiap elemen yang berada dalam satu kolom pada state. Secara lebih jelas, perkalian matriks transformasi mixcolumns ditunjukkan pada Gambar 2.3.



Gambar 2. 3: Perkalian matriks transformasi MixColumns [7]**.**

$$[S'_{0,c}, S'_{1,c}, S'_{2,c}, S'_{3,c}] = \begin{matrix} 02 & 03 & 01 & 01 & S0,c \\ 01 & 02 & 03 & 01 & S1,c \\ 01 & 01 & 02 & 03 & S2,c \\ 03 & 01 & 01 & 02 & S3,c \end{matrix} \quad (1)$$

Berdasarkan persamaan (1), hasil dari perkalian matriks diatas dapat dianggap seperti perkalian pada persamaan (2).

$$S'_{0,c} = (\{02\}.S0,c) \oplus (\{03\}.S1) \oplus S2,c \oplus S3,c$$

$$S'_{1,c} = S0,c \oplus (\{02\}.S1,c) \oplus (\{03\}.S2,c) \oplus S3,c$$

$$S'_{2,c} = S0,c \oplus S1,c \oplus (\{02\}.S2,c) \oplus (\{03\}.S3,c) \quad (2)$$

$$S'_{3,c} = (\{03\}.S1,c) \oplus S1,c \oplus S2,c \oplus (\{02\}.S3,c)$$

2.2.1.2 Proses Dekripsi AES

Transformasi chipher dapat dibalikkan dan diimlementasikan dalam arah yang berlawanan untuk menghasilkan inverse cipher yang mudah dipahami untuk algoritma AES. Trasformasi byte yang digunakan pada invers cipher adalah InvShiftRows, InvSubBytes, InvMixColumns, dan AddRoundKey.

1) InvShiftRows

InvShiftRows adalah transformasi byte yang berkebalikandengan transformasi ShiftRows. Pada transformasi InvShiftRows, dilakukan pergeseran bit ke kanan sedangkan pada ShiftRows dilakukan pergeseran bit ke kiri.

2) InvSubBytes

InvSubBytes juga merupakan transformasi bytes yang berkebalikan dengan transformasi SubBytes. Pada InvSubBytes, tiap elemen pada state dipetakan dengan menggunakan tabel Inverse S-Box.

3) InvMixColumns

Setiap kolom dalam state dikalikan dengan matrik perkalian dalam AES [7].

2.2.1.3 Kunci Ekspansi

Algoritma AES melaksanakan kunci kode dan membuat suatu kunci ekspansi untuk menghasilkan kunci skedul. Kunci ekspansi yang diperlukan AES Nb(Nr+1) kata sehingga bisa digunakan AES 128 bit sehingga 4(10+1) = 40 kata = 44 x 32 bit = 1408 bit upa-kunci. Ekspansi dari 128 menjadi 1408 bit upa-kunci. Proses ini disebut dengan kunci skedul. Upa-kunci ini diperlukan karena setiap putaran merupakan suatau inisial dari Nb kata untuk Nr = 0 dan 2 Nb untuk = 1,3 Nb untuk Nr = 2, …., 11 Nb untuk Nr = 10. Dari operasi ini didapatkan kunci skedul yang berisi larik linear empat byte kata (Wi), $0=I \leq Nb\ (Nr + 1)$ [4].

2.2.2 Rivest Shamir Adleman (RSA)

Algoritma RSA dibuat oleh Ron Rivest, Adi Shamir, dan Leonard Adleman, tiga orang peneliti dari MIT (*Massachussets Institute of Technology*) pada tahun 1976. RSA termasuk ke dalam skema kriptografi asimetris. Kriptografi asimetris merupakan algoritma kriptografi yang menggunakan kunci yang berbeda yaitu kunci publik (*public key*) dan kunci rahasia (*private key*) pada proses enkripsi dan dekripsinya. Kunci publik dapat mengenkripsi pesan namun tidak dapat mendekripsi pesan [11].

Misalnya, Bob ingin mengirim pesan kepada Alice menggunakan kriptografi
asimetris, hal yang harus dilakukan oleh Bob dan Alice adalah:

1. Bob membangkitkan kunci publik dan kunci *privat*.
2. Bob memberikan kunci publik kepada Alice.
3. Alice mengenkripsi pesan menggunakan kunci publik dari Bob, kemudian mengirimkan pesan yang terenkripsi kepada Bob.
4. Bob mendekripsi pesan menggunakan kunci *privat* yang dimilikinya [4].

Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima. Pemfaktoran ini dilakukan untuk memperoleh kunci *privat* yang dapat mendekripsi pesan. Oleh karena itu, RSA dianggap aman. Namun, RSA juga memiliki kelemahan. Proses enkripsi dan dekripsi dalam RSA cenderung memakan waktu yang lama, terutama ketika digunakan untuk mengenkripsi atau mendekripsi data dengan ukuran yang besar. Oleh karena itu, RSA umumnya digunakan untuk pertukaran kunci dan pengamanan data yang relatif kecil, sedangkan algoritma simetris seperti AES digunakan untuk mengenkripsi data dalam skala besar. Properti algoritma RSA sesuai dengan yang tercantum pada tabel 2.3.

Tabel 2. 3: Properti algoritma RSA [11]**.**

Properti	Properti Algoritma	Kerahasiaan
p	bilangan prima	rahasia
q	bilangan prima	rahasia
n	$n = p \times q$	tidak rahasia
ϕ	$\phi(n) = (p - 1)(q - 1)$	rahasia
e	Kunci enkripsi, yang memenuhi $FPB(e, \phi(n)) = 1$	tidak rahasia

Pada tahap pembangkitan kunci, dilakukan beberapa langkah sebagai berikut.

1. Pilih dua bilangan prima p dan q.
2. Hitung

$$n = p \times q \qquad (1)$$

($p \neq q$ agar n tidak mudah difaktorkan)

3. Hitung

$$\phi(n) = (p - 1)(q - 1) \qquad (2)$$

4. Pilih kunci publik e, yang relatif prima terhadap $\phi(n)$.
5. Bangkitkan d yang memenuhi

$$e.d = 1 \ mod \ \phi(n) \qquad (3)$$

Kemudian, proses enkripsinya adalah sebagai berikut.

1. Ambil kunci publik penerima pesan , dan modulus .
2. Nyatakan *plainteks* m menjadi blok-blok m_1 , m_2 , …, sedemikian sehingga setiap blok merepresentasikan nilai di dalam selang [0,n - 1] .
3. Setiap blok m_i dienkripsi menjadi blok c_i dengan rumus

$$c_i = m_i^e \ mod \ n \qquad (4)$$

Pada proses dekripsi, setiap blok *cipherteks* didekripsi kembali menjadi blok m_i dengan rumus $m_i = c_i^d \ mod \ n$ [11].

2.2.2.1 Contoh enkripsi dan dekripsi algoritma RSA

Diberikan teks-asli : TAKEHOMETES [4].

Teks-asli	Heksadesimal	Biner
T	54	01010100
A	41	01000001
K	4B	01001011
E	45	01000101
H	48	01001000
O	4F	01001111
M	4D	01001101
E	45	01000101
T	54	01010100
E	45	01000101
S	53	01010011

Teks-asli biner:

01010100010000010100101101000101010010000100111101001101010001010101010001000101010010011

Teks-asli biner di bagi dalam 11 bit per blok sehingga di[eroleh 8 blokbit teks-asli beserta interpretasi desimalnya:

Teks-asli	Desimal	1024	512	256	128	64	32	16	8	4	2	1
Blok 1	674	0	1	0	1	0	1	0	0	0	1	0
Blok 2	82	0	0	0	0	1	0	1	0	0	1	0
Blok 3	1674	1	1	0	1	0	0	0	1	0	1	0
Blok 4	1156	1	0	0	1	0	0	0	0	1	0	0

Teks-asli	Desimal	1024	512	256	128	64	32	16	8	4	2	1
Blok 5	1958	1	1	1	1	0	1	0	0	1	1	0
Blok 6	1301	1	0	1	0	0	0	1	0	1	0	1
Blok 7	648	0	1	0	1	0	0	0	1	0	0	0
Blok 8	1363	1	0	1	0	1	0	1	0	0	1	1

1. Enkripsi RSA

Ali akan mengirim abu pesan:

Teks-asli : TAKEHOMETES

Ali memilih p = 42, q = 53 dan b = 623

Rumus enkripsi RSA :

$Y = E(X) = \text{Teks-kode\_desimal} = (X)^b \bmod n$

Di mana:

Y : Teks-kode_desimal

X : Teks-asli_desimal

b : kunci umum ali = 623

$n = p \times q$ = 2173

$\phi(n) = (p-1) \times (q-1)$ = 2080

$\gcd(\phi(n), b) = \gcd(2080,623) = 1$

Blok	X	Y =E(X)	Teks-kode_biner (11 bit)
1	674	1650	11001110010
2	82	1230	10011001110
3	1674	1655	11001110111
4	1156	143	00010001111

Blok	X	Y =E(X)	Teks-kode_biner (11 bit)
5	1958	353	00101100001
6	1301	1495	10111010111
7	648	349	00101011101
8	1363	590	01001001110

Teks-kode biner:

11001110010100110011101100111011100010001111001011000011011101011100101011101010010011110

Teks-kode_biner 8 bit	Heksadesimal	Karakter
1100 1110	CE	
0101 0011	53	S
0011 1011	3B	;
0011 1011	3B	;
1000 1000	88	ê
1111 0010	F2	≥
1100 0011	C3	
0111 0101	75	u
1110 1010	EA	Ω
0100 1110	4E	N

Menentukan eksponen dekripsi rahasia (a)

a.b = k.mod $\phi(n)$ + 1

a.623 = k.mod 2080 + 1 atau b^{-1} = 207 mod 2080

gcd(2080,270) = 1

jadi eksponen dekripsi rahasia Ali a = 207

2. Dekripsi RSA

Rumus dekripsi RSA:

$X = D(Y) = \text{Teks-asli\_desimal} = (Y)^a \bmod n$

Di mana:

X : Teks-asli_desimal

Y : Teks-kode_desimal

a : Kunci rahasia Ali = 207

$n = p \times q$ = 2173

$\phi(n) = (p-1) \times (q-1)$ = 2080

$\gcd(\phi(n),a) = \gcd(2080,207)$ = 1

Blok	Y	X = D(Y)	Teks-asli_desimal (11 bit)
1	1650	674	01010100010
2	1230	82	00001010010
3	1655	1674	11010001010
4	143	1156	10010000100
5	353	1958	11110100110
6	1495	1301	10100010101
7	349	648	01010001000
8	590	1363	10101010011

Teks-asli biner:

01010100010000010100101101000101010010000100111101001101010001010101010001000101010010011

Teks-asli biner 8 bit	Heksadesimal	Karakter
0101 0100	54	T
0100 0001	42	A
0100 1011	4B	K
0100 0101	45	E
0100 1000	48	H
0100 1111	4F	O
0100 1101	4D	M
0100 0101	45	E
0101 0100	54	T
0100 0101	45	E
0101 0011	53	S

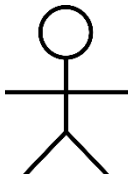
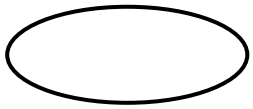
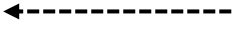
2.2.3 Analisis Sistem

2.2.3.1 Diagram Unifield Modeling Language (UML)

1. *Use Case Diagram.*

Menggambarkan interaksi sistem informasi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Berikut adalah contoh diagram use case:

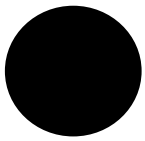
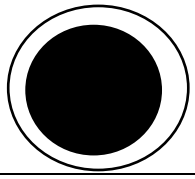
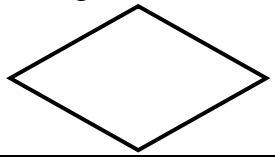
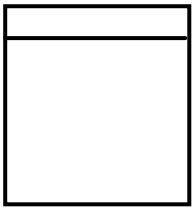
Tabel 2. 4: Simbol dari Use Case Diagram [12]**.**

SIMBOL	KETERANGAN
	Aktor: Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan use case
	*Use case*: Abstraksi dan interaksi antara Sistem Pakar
	Association: Abstraksi dari penghubung antara aktor dengan use case
	Generealisasi: Menunjukan Spesialis aktor untuk dapat berpartipasi dengan use case
<<include>>	Menunjukan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya
<<extend>> 	Menunjukan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi

2. *Activity Diagram*

Diagram aktivitas menggambarkan work flow (aliran kerja) dari sebuah sistem. Berikut tampilan simbol dari *Activity Diagram*:

Tabel 2. 5 Simbol Activity [13]**.**

SIMBOL	KETERANGAN
Status awal 	Sebuah diagram aktifitas memiliki sebuah status awal
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktifitas memliki sebuah status akhir
Aktivitas 	Merupakan suatu aktivitas atau proses dari sebuah sistem
Penggambungan 	Merupakan suatu simbol yang menampilkan suatu aktivitas yang lebih dari satu dan digabungkan menjadi satu
Percabangan 	Simbol yang menampilkan tanda berupa pengambilan keputusan contoh, *true* atau *false*
Swimlane 	Merupakan model pembagian activity diagram, serta menjadikan model ini sebagai model diagram yang bertanggung jawab terhadap aktivitas yang terjadi

2.2.4 Pengujian Sistem

Pengujian Sistem yaitu proses untuk menguji sistem yang dibangun apakah sudah sesuai dengan yang di inginkan. Tahap pengujian sistem sebagai berikut:

2.2.4.1 White Box Testing

White Box adalah pengujian masukan untuk menguji sistem dan keluaran perangkat lunak pada alur kode program [14]. Jika hasilnya tidak sesuai dengan apa yang diharapkan maka akan di kompilasi dan cek kembali kode-kode tersebut [15]. Langkah penyelasaian White Box sebagai berikut:

1. Menganalisa sistem berdasarkan alur flowchart.
2. Membuat flow graph berdasarkan alur flowchart.
3. Menentukan jalur independen berdasarkan gambar floe graph.
4. Menghitung komlpeksitas berdasarkan jalur independen yang dilalui.
   Untuk menghitung kompleksitas siklimatis ada 3 cara yaitu:
   a. Jumlah region grafik alir sesuai dengan kompleksitas siklomatis
   b. Kompleksitas siklomatis V(G) untuk grafik alir G ditentukan sebagai V(G)=EN+2 dimana E adalah jumlah edge grafik alir dan N adalah Jumlah simpul grafik alir
   c. Kompleksitas siklomatis V(G) untuk grafik alir G ditentukn sebagai V(G)=P+1 dimana P adalah jumlah simpul predikat yang diisikan dalam grafik alir G.
5. Melakukan *test case* [15].

2.2.4.2 Black Box Testing

Black Box adalah metode pengujian perangkat lunak dari spesifikasi fungsional tanpa menguji desain dan kode program [16]. Pengujian ini hanya berfokus untuk mengetaui fungsi masukan dan keluaran apakah sudah sesuai dengan yang diinginkan. Salah satu cara dalam menguji Black Box Testing adalah menggunakan teknik Equivalence Partitions (EP).Equivalence Partitions (EP) merupakan sebuah pengujian inputan pada setiap menu yang melalui proses pengelompokan berdasarkan fungsinya baik itu bernilai valid ataupun tidak valid [17].

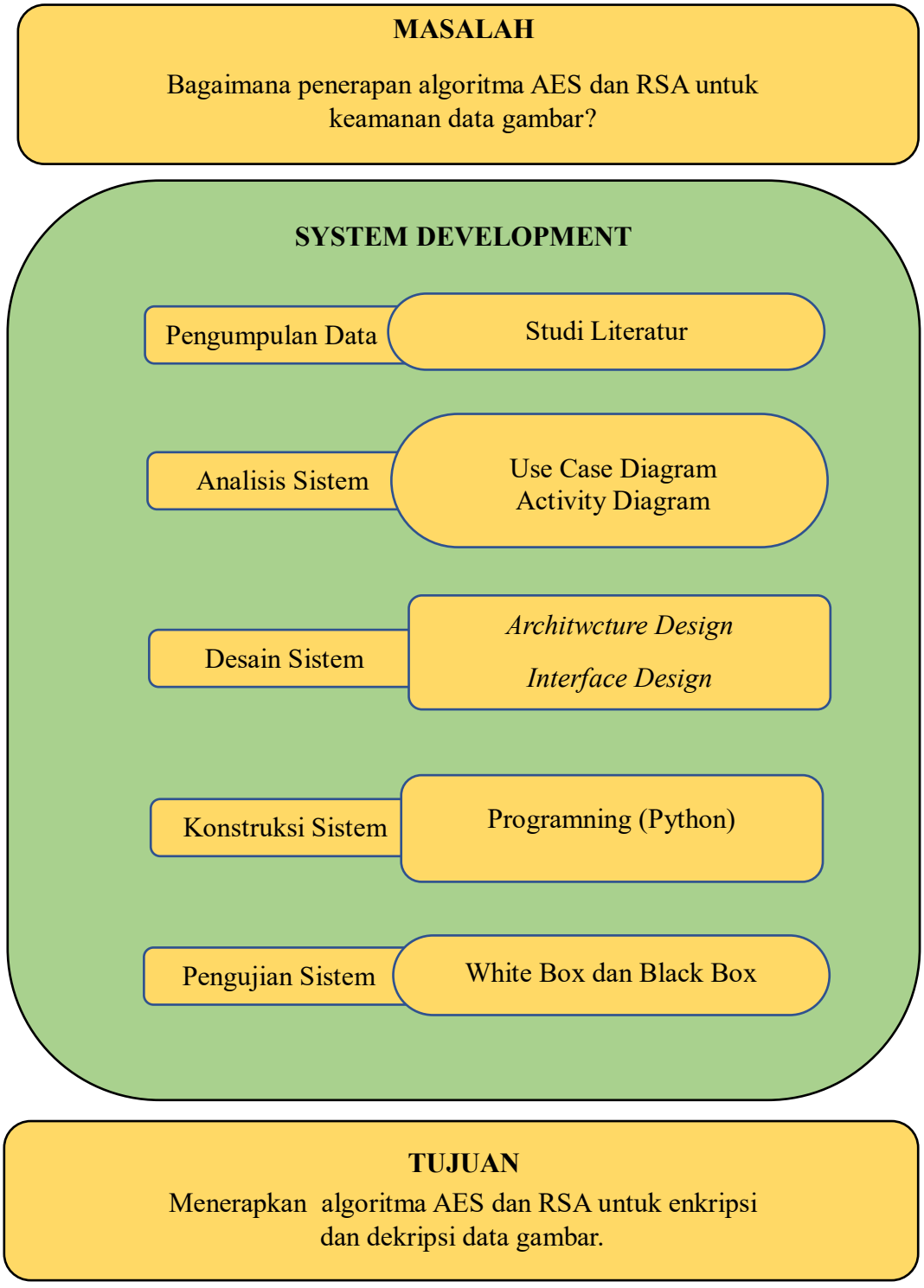
2.2.5 Perangkat lunak Pendukung

Perangkat lunak pendukung yang akan digunakan penulis dalam membangun aplikasi ini yaitu

Tabel 2. 6: Perangkat Lunak Pendukung

NO	PERANKAT LUNAK	KEGUNAAN
1	Python	Bahasa pemrograman yang dipakai buat menciptakan program
2	Visual Studio Code	Kode editor untuk membuat aplikasi web

2.3 Kerangka Pikir



BAB III
METODOLOGI PENELITIAN

3.1 Jenis, Metode, Subjek, Objek, Waktu, dan Lokasi Penelitian

Penelitian ini menggunakan metode *eksperimen*. Dengan demikian jenis penelitian ini adalah penelitian *eksperimental*.

Subjek penelitian ini adalah Aplikasi enkripsi dan dekripsi data gambar menggunakan algoritma Advanced Encryption Standard pada objek data gambar, penelitian ini dimulai dari Januari 2024 sampai dengan Mei 2024.

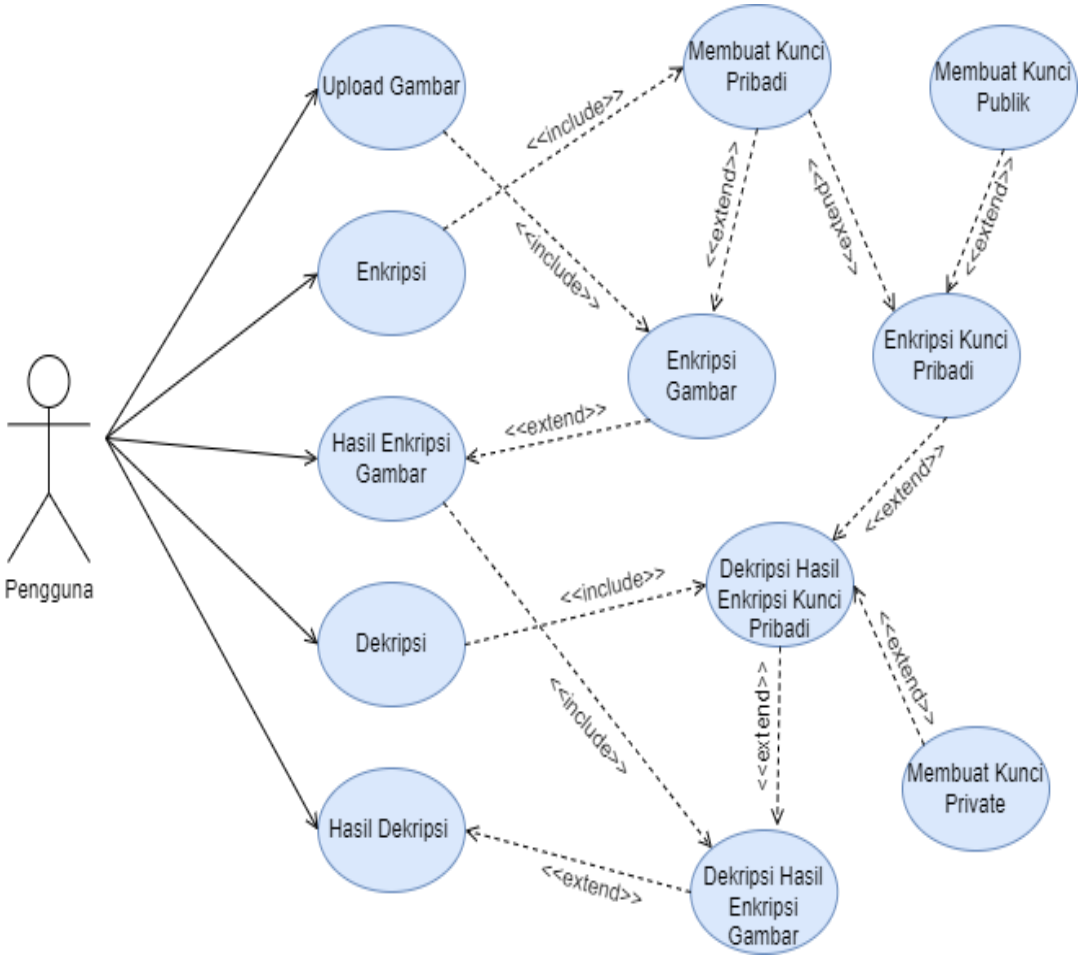
3.2 Pengumpulan Data

Pada penelitian ini pengumpulan data yang dikumpulkan menggunakan *Studi Literatur*. Dengan mengumpulkan barbagai macam referensi seperti jurnal terkait dari internet yang berhubungan dengan kombinasi algoritma AES dan RSA untuk enkripsi dan dekripsi data gambar.

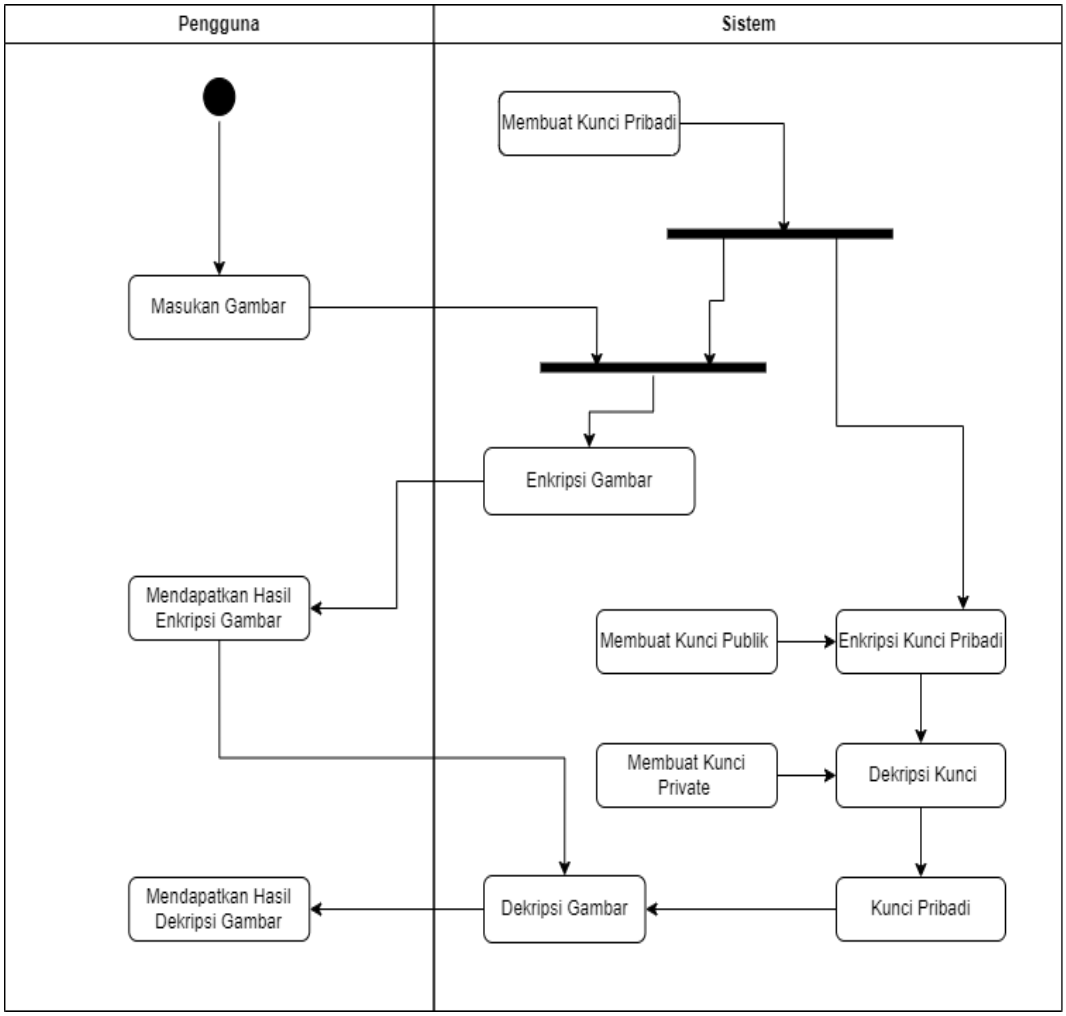
3.2.1 Analisis Sistem

Analisis sistem menggunakan pendekatan berorientasi objek yang digambarkan dalam bentuk:

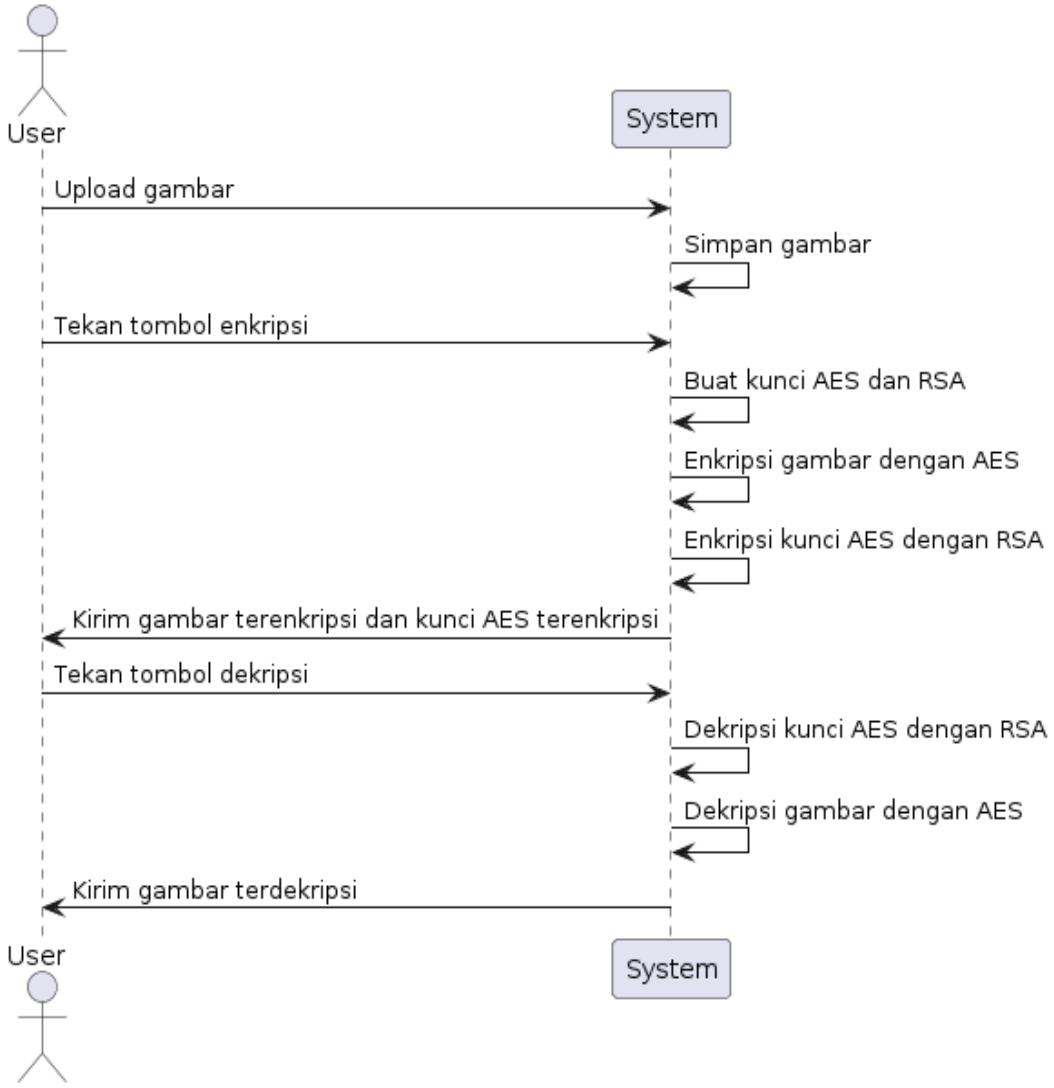
a. *Functional* Modeling, menggunakan alat bantu UML, dalam bentuk:
   - *Use Case Diagram*.
   - *Activity Diagram*.
   - *Sequance Diagram*



Gambar 3. 1 : Use Case Diagram



Gambar 3. 2 : Activity Diagram



Gambar 3. 3: Sequence Diagram.

3.2.2 Desain Sistem

Desain sistem menggunakan pendekatan berorientasi objek yang digambarkan dalam bentuk:

a. *Architecture Design*, menggunakan alat bantu *Visual Studio Code*, dalam bentuk:
   - Model jaringan dari sistem *client server*.
   - Spesifikasi Hardware dan Software yang direkomendasikan.

b. *Interface Design*
   - Mekanisme User.
   - Mekanisme Navigasi.
   - Mekanisme Input
   - Mekanisme Output

3.2.3 Konstruksi Sistem

pada tahap ini menerjamahkan hasil pada tahap anlisis dan desain ke dalam bentuk kode program komputer kemudian membangun sistemnya. Alat bantu yang digunakan pada tahap ini adalah *Visual Studio Code* dengan bahasa *Phyton*.

3.2.4 Pengujian Sistem

a. White Box Testing

Software yang telah direkayasa kemudian dengan metode *White Box Testing* pada kode program proses penerapan metodenya/modelnya. Kode program tersebut dibuatkan *flowchart* programnya, kemudian dipetakan kedalambentuk *flowgraph*, ditentukan jumlah *Region* dan *Cyclomatic Complexity* (CC).apabila independent path = V(G) = (CC) = *Region*, di mana setiap path hanya dieksekusi sekalidan sudah benar, maka sistem dinyatakan efisien dari segi kelayakan logika pemrograman.

b. Black Box

Selanjutnya software diuji pula dengan metode Black Box Testing yang fokus pada keperluan fungsional dari software dan berusaha untuk

menemukan kesalahan dalam beberapa kategori, diantaranya: (1) Fungsi-fungsi yang salah atau hilang; (2) Kesalahan interface; (3) Kesalahan dalam struktur data atau akses basis data eksternal; (4) Kesalahan performa; (5) Kesalahan inisialisasi dan terminasi. Jika sudah tidak ada kesalahan-kesalahan tersebut, maka sistem dinyatakan efisien dari segi kesalahan komponen-komponen sistem.

BAB IV
HASIL PENELITIAN

4.1 Arsitektur Sistem

Sistem yang dibuat menggunakan model jaringan client server. Sedangkan spesifikasi hardware dan software yang direkomendasikan, yaitu:

1. Minimum Processor intel Cleron N4000
2. Minimum RAM 4 GB
3. Minimum Hardisk 500 HDD
4. Operating System Windwos 10
5. Tools yang digunakan yaitu Visual Studio Code

4.2 Interface Design

4.2.1 Mekanisme User

Tabel 4. 1: Mekanisme User

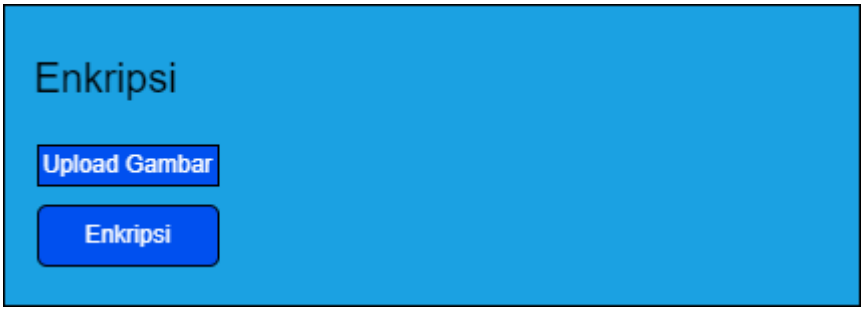
User	Kategori	Akses Input	Akses Output
User	User	All	All

4.2.2 Mekanisme Navigasi.

Tabel 4. 2: Mekanisme Navigasi.

NAVIGASI MENU	**NAVIGASI INPUT**
	NAVIGASI OUTPUT

4.2.3 Mekanisme Input Gambar



Gambar 4. 1: Mekanisme Input Gambar.

4.2.3 Mekanisme Output Enkripsi

Tabel 4. 3: Mekanis Output Enkripsi.

Nama File	Ekstensi File	Nama File Enkripsi	Nama File Bin	Kunci Enkripsi AES	Ukuran File	Waktu Eksekusi	Memori di gunakan	Aksi

4.2.4 Mekanisme Output Dekripsi

Tabel 4. 4: Mekanisme Output Dekripsi

Nama File Enkripsi	Ekstensi File	Nama File Dekripsi	Ukuran File Dekripsi	Waktu Eksekus Dekripsi	Memori digunakan	Keterangan	Aksi

4.3 Analisis dan Implementasi Sistem

4.3.1 Analisis kinerja Enkripsi AES

Tabel 4. 5: Kinerja Enkripsi AES

Nama File	Ukuran File	Waktu Enkripsi	File Hasil Enkripsi	Ukuran File(byte)
KlngkpnBrks.png	178kb	15 ms	enc_20240606_191118.enc	182kb
Uji2.jpeg	65kb	14 ms	enc_20240606_222704.enc	67kb
Uji3.jpg	49kb	5ms	enc_20240606_223546.enc	51kb

4.3.2 Analisis Kinerja Dekripsi AES

Tabel 4. 6: Kinerja Derkipsi AES

Nama File	Ukuran File(byte)	Waktu Dekripsi	File Hasil Dekripsi
enc_20240606_191118.enc	182kb	18 ms	desk_20240606_191239.png
enc_20240606_222704.enc	67kb	6 ms	desk_20240606_222728.jpeg
enc_20240606_223546.enc	51kb	6 ms	desk_20240606_223551.jpg

Berdasarkan kinerja ekripsi dan dekripsi dapat disimpulkan bahwa sistem dapat melakukan proses enkripsi dan dekripsi data gambar menggunakan algoritma AES. Sedangkan waktu pada proses enkripsi dan dekripsi di pengaruhi oleh besarnya ukuran file gambar dan file hasil enkripsi.

4.3.3 Analisis Kinerja Enkripsi AES dan RSA

Tabel 4. 7: Kinerja Enkripsi AES dan RSA

Nama File	Ukuran File	Waktu Enkripsi	File Hasil Enkripsi	Ukuran File(byte)
KlngkpnBrks.png	178kb	4 s	enc_20240606_191353.enc	182kb
Uji2.jpeg	65kb	66 ms	enc_20240606_222742.enc	67kb
Uji3.jpg	49kb	2 s	enc_20240606_223613.enc	51kb

4.3.4 Analisis Kinerja Dekripsi AES dan RSA

Tabel 4. 8: Kinerja Dekripsi AES dan RSA

Nama File	Ukuran File(byte)	Waktu Enkripsi	File Hasil Dekripsi
enc_20240606_191353.enc	182kb	8 ms	desk_20240606_191544.png
enc_20240606_222742.enc	67kb	3 ms	desk_20240606_222755.jpeg
enc_20240606_223613.enc	51kb	3 ms	desk_20240606_223627.jpg

Berdasarkan kinerja enkripsi dan dekripsi dapat disimpulkan bahwa sistem dapat melakukan proses enkripsi dan dekripsi gambar mengggunakan algoritma AES dan melakukan enkripsi dan dekripdi kunci pribadi menggunakan algoritma RSA. Sedangkan waktu proses enkripsi lebih lambat dibandingkan hanya menggunakan algoritma AES, lambatnya proses enkripsi di karenakan adanya proses enkripsi kunci pribadi menggunakan algoritma RSA.

4.4 Pengujian Sistem

4.4.1 Pengujian White Box

if request.method == 'POST:…….....1

if 'file' not in request.files:……..2

flash(‘No. file part’)……...3

return redirect(request.url)……..3

file = request.files['file']……..4

if file.filename == '':……...5

flash('No selected file')……...6

return redirect(request.url)……..6

if file:……..7

start_time = datetime.now()……8

original_file_path = os.path.join(UPLOAD_FOLDER, original_filename)…...8

file.save(original_file_path)……....8

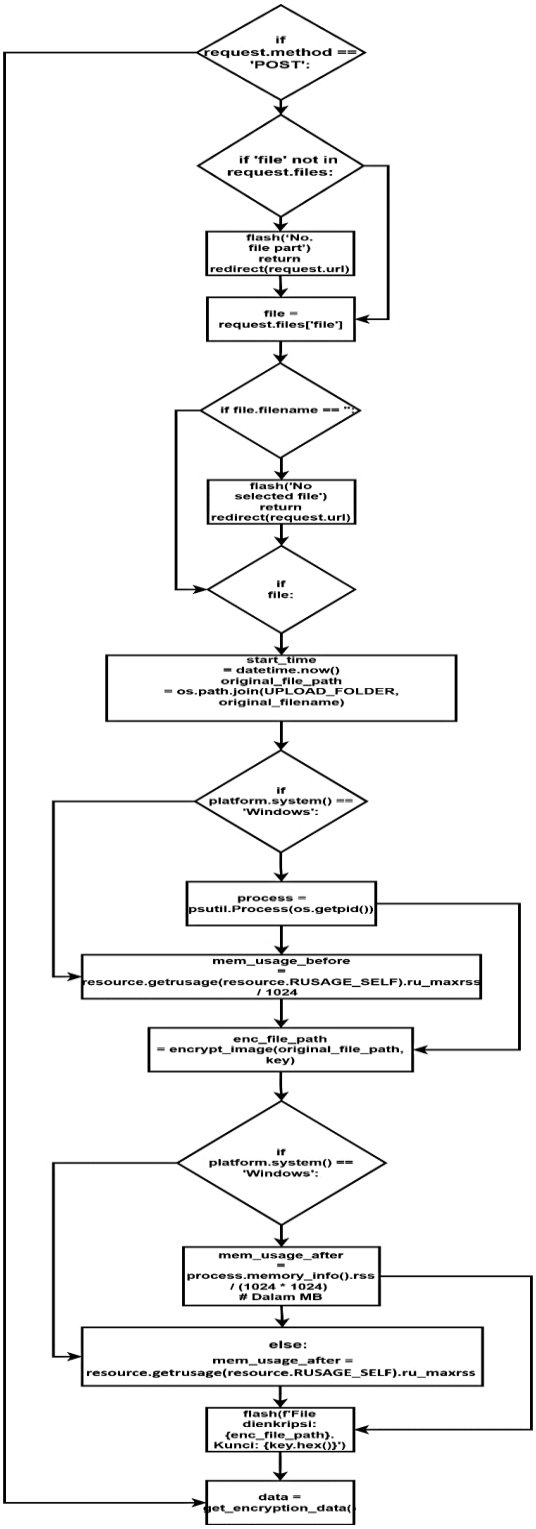
new_filename = f"aes-{formatted_time}.bin"

encrypted_file_path = os.path.join(UPLOAD_FOLDER, new_filename)……8

key = get_random_bytes(16) # Kunci 128-bit……..8

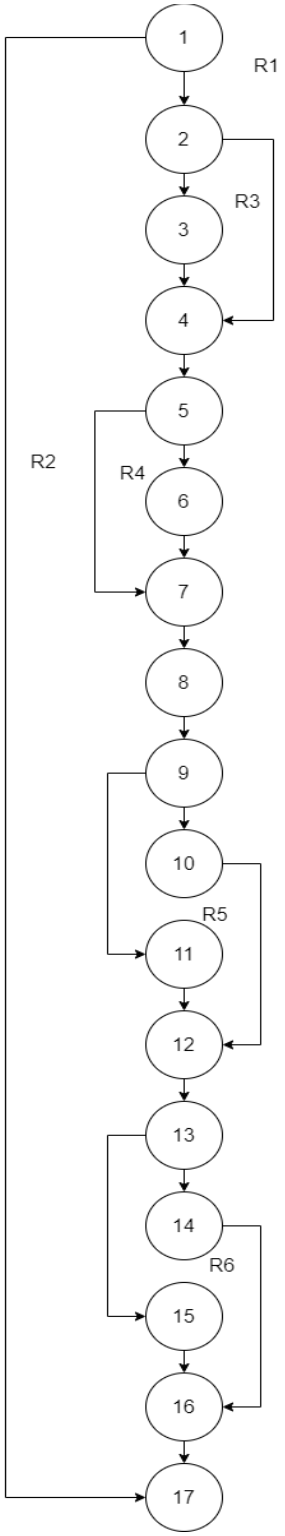
if platform.system() == 'Windows': ……..9

4.4.1.1 *Flowchart* White Box



Gambar 4. 2: Flowchart White Box.

4.4.1.2 *Flowgraph* White Box



Gambar 4. 3: Flowgraph White Box.

4.4.1.3 Perhitungan CC Pada Pengujian *White Box*

Dari *Flowgraph* tesebut didapatkan:

Diketahui Edge(E) = 21

Node(N) = 17

Predikat Node(P) = 5

Region(R) = 6

Penyelesain V(G)1 = E-N+2

= 21 – 17 + 2

= 6

V(G)2 = P + 1

= 5 + 1

= 6

4.4.1.4 Path Pada Pengujian *White Box*

Tabel 4. 9: Path pada pengujian *White Box*.

No	Path	Ket
1	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17	Ok
2	1-17	Ok
3	1-2-4-..-17	Ok
4	1-2-4-5-7-..-17	Ok
5	1—2-4-5-7-8-9-11-..-17	Ok
6	1-2-4-5-7-8-9-11-12-13-15-..-17	Ok

4.4.2 Pengujian *Black Box*

Tabel 4. 10: Pengujian *Black Box*.

NO	INPUT/EVENT	FUNGSI	Hasil Yang Diharapkan	Hasil Uji
1	Home	Halaman Home	Menampilkan Halaman Home	Sesuai
2	Enkripsi AES	Mengenkripsi gambar menggunakan AES	Menampilkan Halaman Enkripsi Gambar	Sesuai
3	Dekripsi AES	Mendekripsi hasil enkripsi gambar menggunakan AES	Menampilkan Halaman Dekripsi Gambar	Sesuai
4	Enkripsi AES dan RSA	Mengenkripsi gambar menggunakan AES dan RSA	Menampilkan Halaman Enkripsi Gambar	Sesuai
5	Dekripsi AES dan RSA	Mendekripsi hasil enkripsi gambar menggunakan AES dan RSA	Menampilkan Halaman Dekripsi Gambar	Sesuai
6	Klik Menu Tentang	Menampilkan Tentang	Menampilkan Halaman Tentang	Sesuai
7	Klik Menu Kontak	Menampilkan Kontak	Menampilkan Halmana Kontak	Sesuai
8	Klik Tombol Upload Gambar	Mengupload Gambar	Menampilkan Gambar	Sesuai
9	Klik Tombol Enkripsi	Mengenkripsi	Menampilkan Hasil Enkripsi	Sesuai
10	Klik Tombol Dekripsi	Mendekripsi	Menampilkan Hasil Dekripsi	Sesuai
11	Klik Tombol Hapus	Menghapus	Menghapus proses enkripsi dan dekripsi	Sesuai

4.5 Penerapan Algoritma AES dan RSA

4.5.1 Proses Gambar menjadi biner

Untuk gambar kita menggunakan contoh gambar sederhana dalam format grayscale. Gambar yang akan digunakan yaitu 2x2 piksel.

A. Mengonversi gambar menjadi grayscale.

Tabel 4. 11: RGB

R	G	B
255	0	0
0	255	0
0	0	255
255	255	255

Untuk mengonversi ke grayscale, kita bisa menggunakan rumus umum: Grayscale=0.299×R+0.587×G+0.114×B\text{Grayscale} = 0.299 \times R + 0.587 \times G + 0.114 \times BGrayscale=0.299×R+0.587×G+0.114×B.

Menghitung nilai grayscale untuk setiap piksel:

1. Piksel (255, 0, 0):
   Grayscale=0.299×255+0.587×0+0.114×0=76.245\text{Grayscale} = 0.299\times 255 + 0.587 \times 0 + 0.114 \times 0 = 76.245Grayscale=0.299×255+0.587×0+0.114×0= 76.245
2. Piksel (0, 255, 0):
   Grayscale=0.299×0+0.587×255+0.114×0=149.685\text{Grayscale} = 0.299 \times 0 + 0.587 \times 255 + 0.114 \times 0 = 149.685Grayscale=0.299×0+0.587×255+0.114×0= 149.685
3. Piksel (0, 0, 255):
   Grayscale=0.299×0+0.587×0+0.114×255=29.07\text{Grayscale} = 0.299 \times 0 + 0.587 \times 0 + 0.114 \times 255 = 29.07Grayscale=0.299×0+0.587×0+0.114×255= 29.07

4. Piksel (255, 255, 255):

   Grayscale=0.299×255+0.587×255+0.114×255=255\text{Grayscale} = 0.299 \times 255 + 0.587 \times 255 + 0.114 \times 255 = 255Grayscale=0.299×255+0.587×255+0.114×255= 255

5. Jadi gambar grayscale-nya adalah: 75, 150 ,29 dan 255.

B. mengonversi nilai grayscale menjadi biner.

   Setiap nilai grayscale (0-255) dapat diubah menjadi biner (8-bit):

   76 = 01001100

   150 = 10010110

   29 = 00011101

   255 = 11111111

C. Konversi teks biner ke blok byte .

   Biner digabungkan menjadi satu blok 16 byte (128 bit).

   01001100 10010110 00011101 11111111 01001100 10010110 00011101 11111111 01001100 10010110 00011101 11111111 01001100 10010110 00011101 11111111

4.5.2 Proses Enkripsi AES

Diketahui Plaintext: 4C 96 1D FF 4C 96 1D FF 4C 96 1D FF 4C 96 1D FF

Kunci = 4B 52 49 50 54 4F 47 52 41 46 49 41 45 53 4B 55

1. Initial Round.

4C	4C	4C	4C		4B	54	41	45
96	96	96	96	⊕	52	4F	46	53
1D	1D	1D	1D		49	47	49	4B
FF	FF	FF	FF		50	52	41	55

Hasil XOR:

07	18	0D	09
C4	D9	D0	C5
54	5A	54	56
AF	AD	BE	AA

2. Pembangkitan Kunci.

Tabel 4. 12: Pembangkitan Kunci

4B	54	41	45	A7	F3	B2	F7	4F	BC	0E	F9
52	4F	46	53	E1	AE	E8	BB	6D	C3	2B	90
49	47	49	4B	B5	F2	BB	F0	09	FB	40	B0
50	52	41	55	3E	6C	2D	78	56	3A	17	6F
Round 0				Round 1				Round 2			

2B	97	99	60	AA	3D	A4	C4	AC	91	35	F1
8A	49	62	F2	26	6F	0D	FF	61	0E	03	FC
A1	5A	1A	AA	FC	A6	BC	16	6B	CD	71	67
CF	F5	E2	8D	1F	EA	08	85	03	E9	E1	64
Round 3				Round 4				Round 5			

3C	AD	98	69	25	88	10	79	B3	3B	2B	52
E4	EA	E9	15	E9	03	EA	FF	14	17	FD	02
28	E5	94	F3	A3	46	D2	21	31	77	A5	84
A2	4B	AA	CE	5B	10	BA	74	ED	FD	47	33
Round 6				Round 7				Round 8			

DF	E4	CF	9D	E3	07	C8	55
4B	5C	A1	A3	02	5E	FF	5C
F2	85	20	A4	B1	34	14	B0
ED	10	57	64	B3	A3	F4	90
Round 9				Round 10			

3. Subbytes.

07	18	0D	09		C5	AD	D7	01
C4	D9	D0	C5		1C	35	70	A6
54	5A	54	56	=	20	BE	20	B1
AF	AD	BE	AA		79	95	AE	AC

4. Shiftrows

C5	AD	D7	01		C5	AD	D7	01
1C	35	70	A6	=>	35	70	A6	1C
20	BE	20	B1		20	B1	20	BE
79	95	AE	AC		AC	79	95	AE

5. Mixcolumn.

02 03 01 01		C5	AD	D7	01		42	19	F1	36
01 02 03 01		35	70	A6	1C		63	FC	75	4E
01 01 02 03	x	20	B1	20	BE	=	5F	2F	95	93
03 01 01 02		AC	79	95	AE		02	DF	D5	E6

6. AddRound Key.

42	19	F1	36		A7	F3	B2	F7		E5	EA	43	C1
63	FC	75	4E	⊕	E1	AE	E8	BB	=	82	52	9D	F5
5F	2F	95	93		B5	F2	BB	F0		EA	DD	2E	63
02	DF	D5	E6		3E	6C	2D	78		3C	B3	F8	9E

Tabel 4. 13: Hasil Round.

Round	Mulai Round	Hasil Subbytes	Hasil Shiftrows	Hasil Mixcolom	Hasil Add Round Key
2	E5 EA 43 C1 82 52 9D F5 EA DD 2E 63 3C B3 F8 9E	D9 87 1A 78 13 00 5E E6 87 C1 31 FB EB 6D 41 0B	D9 87 1A 78 00 5E E6 13 31 FB 87 C1 0B EB 6D 41	93 E7 EF 45 81 C6 32 47 A6 12 5E 31 57 FA 95 D8	DC 5B E1 BC EC 05 19 D7 AF E9 1E 81 01 C0 82 B7
3	DC 5B E1 BC EC 05 19 D7 AF E9 1E 81 01 C0 82 B7	86 39 F8 65 CE 6B D4 0E 79 1E 72 0C 7C BA 13 A9	86 39 F8 65 6B D4 0E CE 72 0C 79 1E A9 7C BA 13	71 65 3A 8E 6F E2 D5 D3 E9 71 D1 A2 C1 6B 0B 59	5A F2 A3 EE E5 AB B7 21 48 2B CB 08 0E 9E E9 D4
4	5A F2 A3 EE E5 AB B7 21 48 2B CB 08 0E 9E E9 D4	BE 89 0A 28 D9 62 A9 FD 52 F1 1F 30 AB 0B 1E 48	BE 89 0A 28 62 A9 FD D9 1F 30 52 F1 48 AB 0B 1E	96 72 51 CF 13 3B 16 97 3A A6 4E 2A 34 54 A7 6C	3C 4F F5 0B 35 54 1B 68 C6 00 F2 3C 2B BE AF E9
5	3C 4F F5 0B 35 54 1B 68 C6 00 F2 3C 2B BE AF E9	EB 84 E6 2B 96 20 AF 45 B4 63 89 EB F1 AE 79 1E	EB 84 E6 2B 20 AF 45 96 89 EB B4 63 1E F1 AE 79	3A E3 02 ED 35 16 05 C0 E0 EE 39 F0 B3 2A 87 7A	96 72 37 1C 54 18 06 3C 8B 23 48 97 B0 C3 66 1E
6	96 72 37 1C 54 18 06 3C 8B 23 48 97 B0 C3 66 1E	90 40 9A 9C 20 AD 6F EB 3D 26 52 88 E7 2E 33 72	90 40 9A 9C AD 6F EB 20 52 88 3D 26 72 E7 2E 33	F7 5E 1A 56 55 FA 3E 85 0F 16 79 A5 B0 F2 3F DF	CB F3 82 3F B1 10 D7 90 27 F3 ED 56 12 B9 95 11
7	CB F3 82 3F B1 10 D7 90 27 F3 ED 56 12 B9 95 11	1F 0D 13 75 C8 CA 0E 60 CC 0D 55 B1 C9 56 2A 82	1F 0D 13 75 CA 0E 60 C8 55 B1 CC 0D 82 C9 56 2A	AC 70 1C 8E ED 10 CA C3 E2 3A 0A D9 A1 21 35 0E	89 F8 0C F7 04 13 20 3C 41 7C D8 F8 FA 31 8F 7A
8	89 F8 0C F7 04 13 20 3C 41 7C D8 F8 FA 31 8F 7A	A7 41 FE 68 F2 7D B7 EB 83 10 61 41 2D C7 73 DA	A7 41 FE 68 7D B7 EB F2 61 41 83 10 DA 2D C7 73	69 2C 85 BE 24 DA 6A D4 6D 03 5A 2F 41 6F E4 BC	DA 17 AE EC 30 CD 97 D6 5C 74 FF AB AC 92 A3 8F

Round	Mulai Round	Hasil Subbytes	Hasil Shiftrows	Hasil Mixcolom	Hasil Add Round Key
9	DA 17 AE EC 30 CD 97 D6 5C 74 FF AB AC 92 A3 8F	57 F0 E4 CE 04 BD 88 F6 4A 92 16 62 91 4F 0A 73	57 F0 E4 CE BD 88 F6 04 16 62 4A 92 73 91 4F 0A	17 8B D7 13 7F CC 82 61 53 14 57 EB B4 D8 15 CB	C8 6F 18 8E 34 90 23 C2 A1 91 77 4F 59 C8 42 AF
10	C8 6F 18 8E 34 90 23 C2 A1 91 77 4F 59 C8 42 AF	E8 A8 AD 19 18 60 26 25 32 81 F5 84 CB E8 2C 79	E8 A8 AD 19 60 26 25 18 F5 84 32 81 79 CB E8 2C		0B AF 65 4C 62 78 DA 44 44 B0 26 31 CA 68 1C BC

Hasil akhir proses enkripsi yaitu:

Tabel 4. 14: Hasil enkripsi AES.

Hexa	0B	62	44	CA	AF	78	B0	68	65	DA	26	1C	4C	44	31	BC
Dec	13	98	69	202	175	120	176	104	101	218	38	28	76	68	49	188
Chiper	♪	b	E	╩	»	X	▒	H	e	┌	&	└	L	D	1	╝

4.5.3 Proses Enkripsi dan Dekripsi RSA

Pada proses ini kunci dari algoritma AES di enkripsi dan dekripsi

Dik. $p = 3, q = 5$

$N = 3 \times 5 = 15$

Mencari Nilai $\phi(n) = (3-1) \times (5-1) = 2 \times 4 = 8$

Plih kunci $e = 3$

Menghitung kunci privat d dengan ke kongruenan

$e.d = 1(\text{mod}(\phi(n)))/e$

penyelesaian: $d = 1 + (K*(\phi(n))/e$

$= 1 + (4 * 8)/3$

$= (32 + 1)/3 = 33/3 = 11$

Maka didapat nilai *d* yang bulat adalah 11

Mengkonversi plainteks hexadesimal “4B” ke dalam bentuk desimal sehingga diperoleh:

4B = 75

Plainteks: 75

Proses Enkripsi:

Rumus: Ci = mi^e mod n

$C1 = 75^3 \bmod 8$

= 3

Chipertext = 3

Proses Dekripsi:

Rumus: mi = Ci^d mod n

$m1 = 3^5 \bmod 11$

= 75

Plaintext m1 = 75 yang dimana dalam bentuk Hexadesimal yaitu “4B” yang berasal dari kunci pribadi.

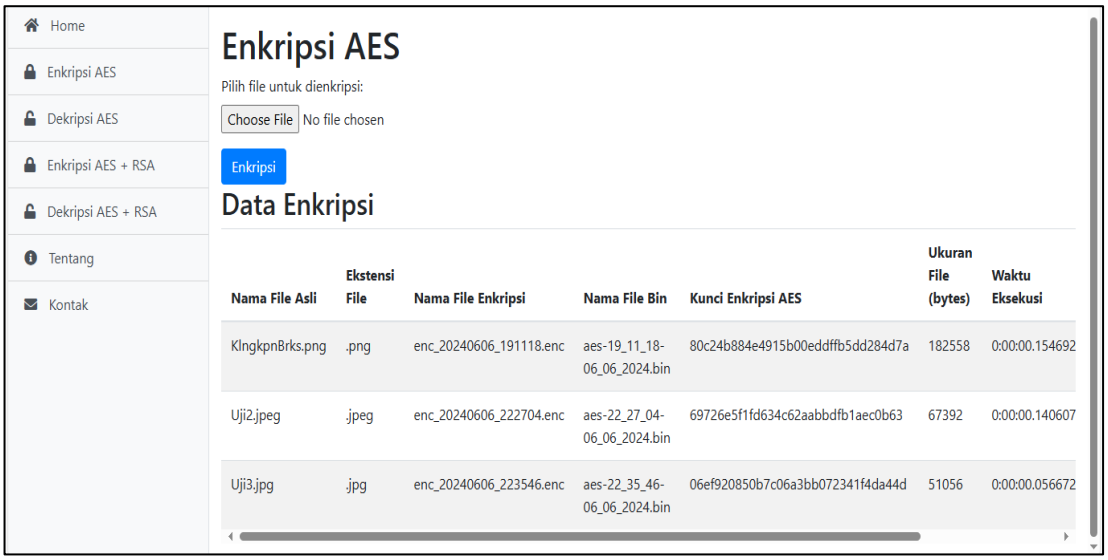
Berdasarkan penerapan algoritma AES dan RSA disimpulkan bahwa keamanan gambar akan semakin kuat karena seorang kriptanalis harus mengetahui kunci private dan kunci pribadi untuk mendapatkan data gambar yang di inginkan.

BAB V

PEMBAHASAN

5.1 Pembahasan Sistem

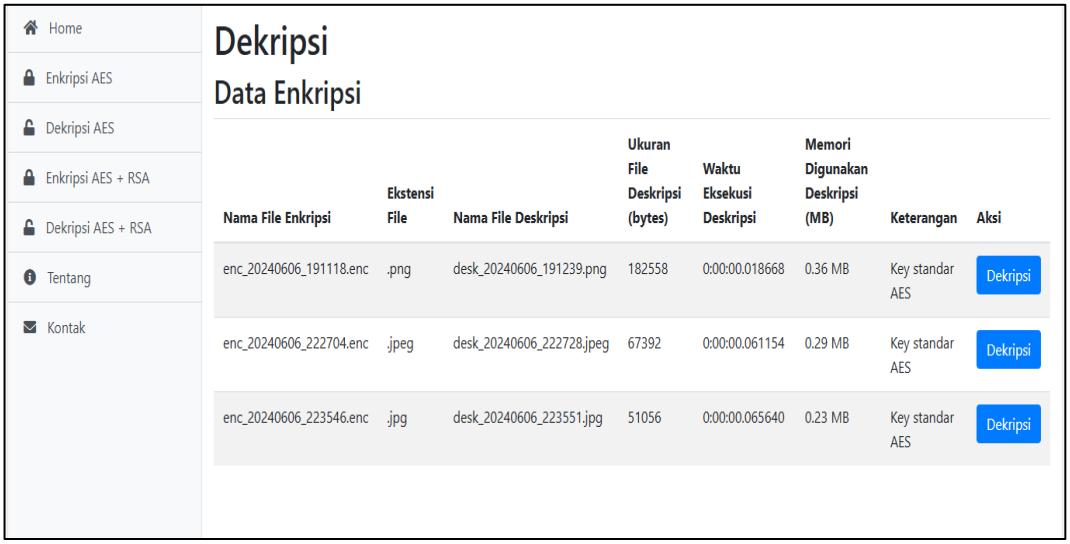
5.1.1 Tampilan Enkripsi AES



Gambar 5. 1: Tampilan Menu Enkripsi AES

Pada menu ini dilakukan enkrispis pada gambar menggunkan algorrtima AES.

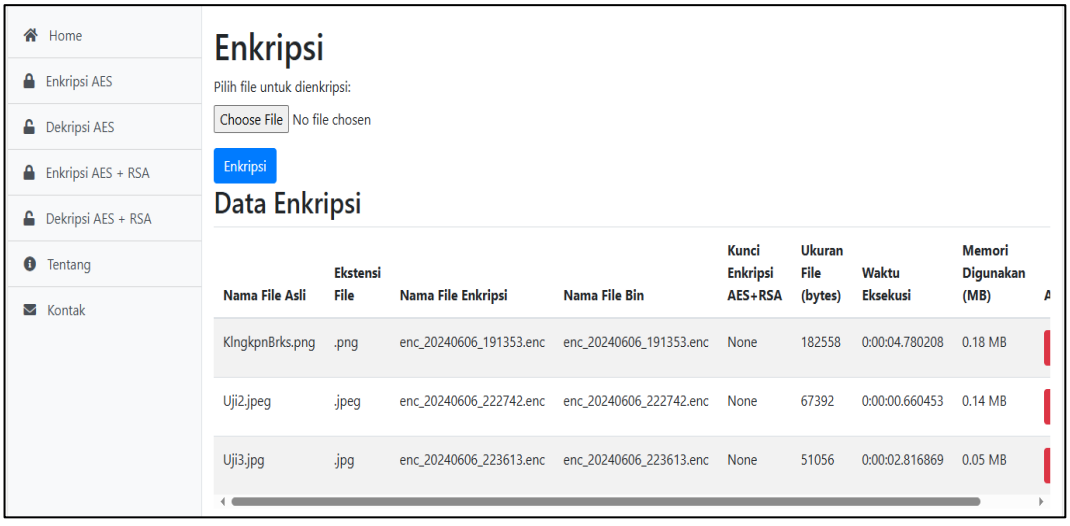
5.1.2 Tampilan Dekripsi AES



Gambar 5. 2: Tampilan Menu Dekripsi AES

Pada menu ini dilakukan dekripsi pada hasil enkripsi gambar enggunakan algortima AES.

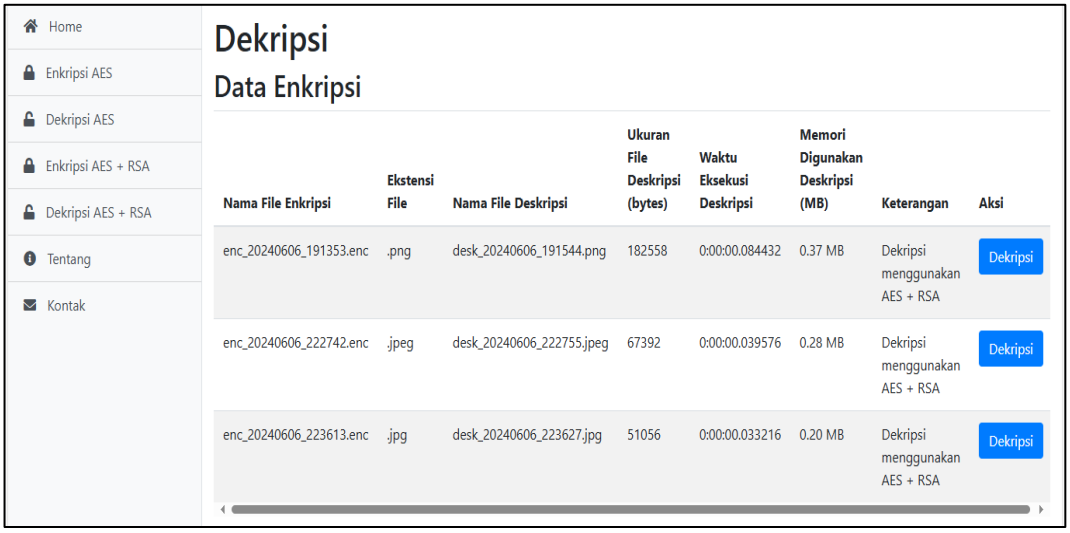
5.1.3 Tampilan Enkripsi AES dan RSA



Gambar 5. 3: Tampilan Menu Enkripsi AES dan RSA

Pada menu ini dilakukan enkripsi gambar menggunakan algortima AES kemudian kunci AES di enkripsi menggunakan algortima RSA.

5.1.4 Tampilan Dekripsi AES dan RSA



Gambar 5. 4 : Tampilan Menu Dekripsi Menggunakan AES dan RSA

Pada menu ini dilakukan dekripsi kunci AES menggunakan algortima RSA kemudian barulah gambar hasil enkripsi di dekrpisi menggunakan Algortima AES.

BAB VI
PENUTUP

6.1 Kesimpulan

Berdasarkan hasil penelitian yang dilakukan dan pembahasan yang telah diuraikan sebelumnya, maka kesimpulan yaitu Algoritma AES dapat diterapkan untuk keamanan data gambar. algortima RSA yang digunakan untuk mengamankan kunci pribadi dapat memperkuat keamanan kunci pribadi sehingga kriptanalis lebih sulit untuk mendapatkan kunci pribadi dari algoritma AES yang digunakan untuk mengenkripsi dan dekripsi data gambar. Dapat diketahui bahwa aplikasi yang dibuat dapat digunakan. Hal ini dibuktikan dengan hasil pengujian yang dilakukan dengan metode *White Box* dan *Basis Path* yang menghasilkan nilai V(G) = CC = 6, serta pengujian *Black Box* yang menggambarkan kebenaran sebuah logika sehingga didapat bahwa logika *flowchart* benar dan menghasilkan aplikasi yang dapat digunakan.

6.2 Saran

Setelah melakukan penelitian, ada beberapa saran yang perlu diperhatikan untuk mencapai tujuan yang diharapkan, yaitu data gambar yang di enkripsi bisa dalam bentuk lain seperti SVG dan EPS.

DAFTAR PUSTAKA

[1] A. Pratama, M. N. Arif, M. Nazir, and Z. Dannaun, "Algoritma Des (Data Encryption Standard) Untuk," *J. Siteba*, vol. 2, no. 1, pp. 15–18, 2023.

[2] Azlin, F. Musadat, and J. Nur, "Aplikasi Kriptografi Keamanan Data Menggunakan Algoritma Base64," *J. Inform.*, vol. 7, no. 2, pp. 1–5, 2018.

[3] T. Limbong, U. Katolik, S. Thomas, and S. Utara, "Pengujian kriptografi klasik caesar chipper menggunakan matlab," no. September 2015, 2017.

[4] D. Ariyus, *Pengantar Ilmu Kriptografi Teori, Analisis dan Implementasi.* Yogyakarta: Penerit Andi, 2008.

[5] N. E. Saragih, D. Universitas, and P. Utama, "Implementasi Algoritma One Time Pad," no. 3, pp. 31–40, 1924.

[6] S. H. Putra, E. Santoso, and L. Muflikhah, "Implementasi Algoritma Kriptografi Advanced Encryption Standard (AES) Pada Kompresi Data Teks," *J. Ilmu Komput.*, pp. 1–14, 2013.

[7] F. Muharram, H. Azis, and A. R. Manga, "Analisis Algoritma pada Proses Enkripsi dan Dekripsi File Menggunakan Advanced Encryption Standard (AES)," *Proc. Semin. Nas. Ilmu Komput. dan Teknol. Inf.*, vol. 3, no. 2, pp. 112–115, 2018.

[8] S. R. Siburian, R. Alek, S. Sinaga, and F. Yudistira, "Kriptosistem Hybrid Menggunakan Kombinasi Aes Dan Rsa Untuk Enkripsi Teks Pesan," *J. JOCOTIS - J. Sci. Inform. Robot.*, vol. 1, no. 1, pp. 22–31, 2023, [Online]. Available: https://jurnal.ittc.web.id/index.php/jct/

[9] Asriyanik, "Studi Terhadap Advanced Encryption Standard (Aes) Dan," *J. Ilm. Sains dan Teknol.*, vol. 7, no. 1, pp. 553–561, 2017.

[10] Asiyanik, "Studi Terhadap Advanced Encryption Standard (Aes) DanAlgoritma Knapsack Dalam Pengamanan Data," *Santika*, vol. 7, no. J.

Ilm. Sains dan Teknol., pp. 553–561, 2017.

[11] P. E-mail, “Program Studi Matematika, FPMIPA, Universitas Pendidikan Indonesia e-mail:,” vol. 3, no. 2, pp. 92–101, 2023.

[12] R. Juliarto, “Contoh Use Case Diagram Lengkap dengan Penjelasannya,” Dicoding Blog. Accessed: Mar. 26, 2024. [Online]. Available: https://www.dicoding.com/blog/contoh-use-case-diagram/

[13] Rendi Juliarto, “Apa itu Activity Diagram? Beserta Pengertian, Tujuan, Komponen.” Accessed: Mar. 26, 2024. [Online]. Available: https://www.dicoding.com/blog/apa-itu-activity-diagram/

[14] C. T. Pratala, E. M. Asyer, I. Prayudi, and A. Saifudin, “Pengujian White Box pada Aplikasi Cash Flow Berbasis Android Menggunakan Teknik Basis Path,” *J. Inform. Univ. Pamulang*, vol. 5, no. 2, p. 111, 2020, doi: 10.32493/informatika.v5i2.4713.

[15] D. Suprapti, M. Kamisutara, and P. Artaya, “Analisa Pengujian Sistem Informasi Penjualan,” *Anal. Penguji. Sist. Inf. Penjualan Menggunakan Metod. White Box*, pp. 1–12, 2017.

[16] W. N. Cholifah, Y. Yulianingsih, and S. M. Sagita, “Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap,” *STRING (Satuan Tulisan Ris. dan Inov. Teknol.*, vol. 3, no. 2, p. 206, 2018, doi: 10.30998/string.v3i2.3048.

[17] A. A. Arwaz, T. Kusumawijaya, R. Putra, K. Putra, and A. Saifudin, “Pengujian Black Box pada Aplikasi Sistem Seleksi Pemenang Tender Menggunakan Teknik Equivalence Partitions,” *J. Teknol. Sist. Inf. dan Apl.*, vol. 2, no. 4, p. 130, 2019, doi: 10.32493/jtsi.v2i4.3708.

LAMPIRAN

Lampiran 1: Program.

Proses Enkripsi AES

```
def encrypt():
    if request.method == 'POST':
        if 'file' not in request.files:
            flash('No file part')
            return redirect(request.url)
        file = request.files['file']
        if file.filename == '':
            flash('No selected file')
            return redirect(request.url)
        if file:
            start_time = datetime.now()

            now = datetime.now()
            formatted_time = now.strftime("%H_%M_%S-%d_%m_%Y")

            original_filename = file.filename
            original_ext = os.path.splitext(original_filename)[1]

            # Simpan file asli
            original_file_path = os.path.join(UPLOAD_FOLDER,
original_filename)
            file.save(original_file_path)

            # Generate new filename for encryption
            new_filename = f"aes-{formatted_time}.bin"
            encrypted_file_path = os.path.join(UPLOAD_FOLDER,
new_filename)

            key = get_random_bytes(16)  # Kunci 128-bit

            if platform.system() == 'Windows':
                process = psutil.Process(os.getpid())
                mem_usage_before = process.memory_info().rss / (1024
* 1024)  # Dalam MB
            else:
                mem_usage_before =
resource.getrusage(resource.RUSAGE_SELF).ru_maxrss / 1024  # Dalam
MB
```

```
            enc_file_path = encrypt_image(original_file_path, key)

            if platform.system() == 'Windows':
                mem_usage_after = process.memory_info().rss / (1024
* 1024)  # Dalam MB
            else:
                mem_usage_after =
resource.getrusage(resource.RUSAGE_SELF).ru_maxrss / 1024  # Dalam
MB

            memory_used = mem_usage_after - mem_usage_before
            end_time = datetime.now()
            execution_time = end_time - start_time

            # Simpan data enkripsi ke database
            conn = sqlite3.connect(DB_PATH)
            cursor = conn.cursor()
            cursor.execute('''INSERT INTO data_enkripsi
                              (nama_file_asli, ekstensi_file,
nama_file_enc, nama_file_bin, key_enc, ukuran_file, waktu_eksekusi,
memori_digunakan)
                              VALUES (?, ?, ?, ?, ?, ?, ?, ?)''',
                           (original_filename, original_ext,
os.path.basename(enc_file_path),
os.path.basename(encrypted_file_path), key.hex(),
                            os.path.getsize(original_file_path),
str(execution_time), f"{memory_used:.2f} MB"))
            conn.commit()
            conn.close()

            flash(f'File dienkripsi: {enc_file_path}. Kunci:
{key.hex()}')
            return redirect(url_for('encrypt'))
    data = get_encryption_data()
    return render_template('encrypt.html', data=data)
```

Proses Dekripsi AES

```
def decrypt():
    if request.method == 'POST':
        enc_filename = request.form['enc_filename']

        conn = sqlite3.connect(DB_PATH)
        cursor = conn.cursor()
        cursor.execute('SELECT key_enc, key_aes_rsa, ekstensi_file,
file_desk FROM data_enkripsi WHERE nama_file_enc = ?',
(enc_filename,))
        row = cursor.fetchone()

        if row:
            key_enc, key_aes_rsa, file_ext, file_desk = row
            if file_desk:  # Check if file_desk is already filled
                flash(f'Data telah dilakukan Deskripsi: {file_desk}.
Tidak bisa diproses ulang.')
                conn.close()
                return redirect(url_for('decrypt'))

            if key_enc:
                key = bytes.fromhex(key_enc)
                keterangan = "Key standar AES"
            elif key_aes_rsa:
                key = bytes.fromhex(key_aes_rsa)
                keterangan = "Key AES+RSA"
            else:
                flash('Kunci enkripsi tidak ditemukan.')
                conn.close()
                return redirect(url_for('decrypt'))

            start_time = datetime.now()

            file_path = os.path.join(UPLOAD_FOLDER, enc_filename)

            if platform.system() == 'Windows':
                process = psutil.Process(os.getpid())
                mem_usage_before = process.memory_info().rss / (1024
* 1024)  # In MB
            else:
                mem_usage_before =
resource.getrusage(resource.RUSAGE_SELF).ru_maxrss / 1024  # In MB

            dec_file_path = decrypt_image(file_path, key, file_ext)
```

```
            if platform.system() == 'Windows':
                mem_usage_after = process.memory_info().rss / (1024
* 1024)  # In MB
            else:
                mem_usage_after =
resource.getrusage(resource.RUSAGE_SELF).ru_maxrss / 1024  # In MB

            memory_used = mem_usage_after - mem_usage_before
            end_time = datetime.now()
            execution_time = end_time - start_time

            # Update decryption data in the database
            cursor.execute('''UPDATE data_enkripsi SET
                              file_desk = ?, ukuran_file_desk = ?,
waktu_eksekusi_desk = ?, memori_digunakan_desk = ?, keterangan = ?
                              WHERE nama_file_enc = ?''',
                           (os.path.basename(dec_file_path),
os.path.getsize(dec_file_path), str(execution_time),
f"{memory_used:.2f} MB", keterangan, enc_filename))
            conn.commit()
            conn.close()

            flash(f'File didekripsi: {dec_file_path}')
            return redirect(url_for('decrypt'))
    data = get_decryption_data()
    return render_template('decrypt.html', data=data)
```

Proses Enkripsi RSA

```
def encryptrsa():
    if request.method == 'POST':
        if 'file' not in request.files:
            flash('No file part')
            return redirect(request.url)
        file = request.files['file']
        if file.filename == '':
            flash('No selected file')
            return redirect(request.url)
        if file:
            start_time = datetime.now()

            now = datetime.now()
            formatted_time = now.strftime("%H_%M_%S-%d_%m_%Y")

            original_filename = file.filename
            original_ext = os.path.splitext(original_filename)[1]

            # Save the original file
            original_file_path = os.path.join(UPLOAD_FOLDER,
original_filename)
            file.save(original_file_path)

            # Generate RSA key pair
            rsa_key = RSA.generate(2048)
            public_key = rsa_key.publickey()
            cipher_rsa = PKCS1_OAEP.new(public_key)

            # Generate AES key
            aes_key = get_random_bytes(16)  # 128-bit key

            # Encrypt AES key with RSA public key
            encrypted_aes_key = cipher_rsa.encrypt(aes_key)

            # Save the RSA private key
            private_key_path = os.path.join(UPLOAD_FOLDER,
f'private_key_{formatted_time}.pem')
            with open(private_key_path, 'wb') as f:
                f.write(rsa_key.export_key())

            # Save the encrypted AES key to a file
            encrypted_aes_key_path = os.path.join(UPLOAD_FOLDER,
f'encrypted_aes_key_{formatted_time}.bin')
```

```
            with open(encrypted_aes_key_path, 'wb') as f:
                f.write(encrypted_aes_key)

            if platform.system() == 'Windows':
                process = psutil.Process(os.getpid())
                mem_usage_before = process.memory_info().rss / (1024
* 1024)  # In MB
            else:
                mem_usage_before =
resource.getrusage(resource.RUSAGE_SELF).ru_maxrss / 1024  # In MB

            # Encrypt the file using AES key (implement
encrypt_image function accordingly)
            enc_file_path = encrypt_image(original_file_path,
aes_key)

            if platform.system() == 'Windows':
                mem_usage_after = process.memory_info().rss / (1024
* 1024)  # In MB
            else:
                mem_usage_after =
resource.getrusage(resource.RUSAGE_SELF).ru_maxrss / 1024  # In MB

            memory_used = mem_usage_after - mem_usage_before
            end_time = datetime.now()
            execution_time = end_time - start_time

            # Save encryption data to the database
            conn = sqlite3.connect(DB_PATH)
            cursor = conn.cursor()
            cursor.execute('''INSERT INTO data_enkripsi
                              (nama_file_asli, ekstensi_file,
nama_file_enc, nama_file_bin, key_enc, ukuran_file, waktu_eksekusi,
memori_digunakan, private_key_path, encrypted_aes_key_path)
                              VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?,
?)''',
                           (original_filename, original_ext,
os.path.basename(enc_file_path), os.path.basename(enc_file_path),
None,
                            os.path.getsize(original_file_path),
str(execution_time), f"{memory_used:.2f} MB", private_key_path,
encrypted_aes_key_path))
            conn.commit()
            conn.close()
```

```
            flash(f'File dienkripsi: {enc_file_path}. Kunci telah
dienkripsi dengan RSA.')
            return redirect(url_for('encrypt'))
    data = get_encryption_datarsa()
    return render_template('encrypt_rsa.html', data=data)
```

Proses Dekripsi RSA

```
def decryptrsa():
    if request.method == 'POST':
        enc_filename = request.form['enc_filename']

        conn = sqlite3.connect(DB_PATH)
        cursor = conn.cursor()
        cursor.execute('SELECT private_key_path,
encrypted_aes_key_path, ekstensi_file, file_desk FROM data_enkripsi
WHERE nama_file_enc = ?', (enc_filename,))
        row = cursor.fetchone()

        if row:
            private_key_path, encrypted_aes_key_path, file_ext,
file_desk = row
            if file_desk:  # Check if file_desk is already filled
                flash(f'Data telah dilakukan Deskripsi: {file_desk}.
Tidak bisa diproses ulang.')
                conn.close()
                return redirect(url_for('decrypt'))

            # Read the RSA private key
            with open(private_key_path, 'rb') as f:
                private_key = RSA.import_key(f.read())

            # Read the encrypted AES key
            with open(encrypted_aes_key_path, 'rb') as f:
                encrypted_aes_key = f.read()

            # Decrypt the AES key
            cipher_rsa = PKCS1_OAEP.new(private_key)
            aes_key = cipher_rsa.decrypt(encrypted_aes_key)

            start_time = datetime.now()

            file_path = os.path.join(UPLOAD_FOLDER, enc_filename)

            if platform.system() == 'Windows':
                process = psutil.Process(os.getpid())
                mem_usage_before = process.memory_info().rss / (1024
* 1024)  # In MB
            else:
                mem_usage_before =
resource.getrusage(resource.RUSAGE_SELF).ru_maxrss / 1024  # In MB
```

```
            dec_file_path = decrypt_imagersa(file_path, aes_key,
file_ext)

            if platform.system() == 'Windows':
                mem_usage_after = process.memory_info().rss / (1024
* 1024)  # In MB
            else:
                mem_usage_after =
resource.getrusage(resource.RUSAGE_SELF).ru_maxrss / 1024  # In MB

            memory_used = mem_usage_after - mem_usage_before
            end_time = datetime.now()
            execution_time = end_time - start_time

            # Update decryption data in the database
            cursor.execute('''UPDATE data_enkripsi SET
                              file_desk = ?, ukuran_file_desk = ?,
waktu_eksekusi_desk = ?, memori_digunakan_desk = ?, keterangan = ?
                              WHERE nama_file_enc = ?''',
                           (os.path.basename(dec_file_path),
os.path.getsize(dec_file_path), str(execution_time),
f"{memory_used:.2f} MB", "Dekripsi menggunakan AES + RSA",
enc_filename))
            conn.commit()
            conn.close()

            flash(f'File didekripsi: {dec_file_path}')
            return redirect(url_for('decrypt'))
    data = get_decryption_datarsa()
    return render_template('decrypt.html', data=data)
```

Proses Hapus

```
def delete(enc_filename):
    # Hapus data dari database dan file dari direktori
    conn = sqlite3.connect(DB_PATH)
    cursor = conn.cursor()
    cursor.execute('SELECT nama_file_enc, nama_file_bin, file_desk,
nama_file_asli FROM data_enkripsi WHERE nama_file_enc = ?',
(enc_filename,))
    row = cursor.fetchone()
    if row:
        # Hapus file dari direktori
        enc_file_path = os.path.join(UPLOAD_FOLDER, row[0])
        bin_file_path = os.path.join(UPLOAD_FOLDER, row[1])
        # desk_file_path = os.path.join(UPLOAD_FOLDER, row[2])
        nama_file_asli = os.path.join(UPLOAD_FOLDER, row[3])

        if os.path.exists(enc_file_path):
            os.remove(enc_file_path)
        if os.path.exists(bin_file_path):
            os.remove(bin_file_path)
        # if os.path.exists(desk_file_path) and row[2]:  # Hanya
hapus jika file_desk tidak kosong
        #     os.remove(desk_file_path)
        if os.path.exists(nama_file_asli) and row[3]:
            os.remove(nama_file_asli)

        # Hapus data dari tabel
        cursor.execute('DELETE FROM data_enkripsi WHERE
nama_file_enc = ?', (enc_filename,))
        conn.commit()
        flash(f'Data dan file terkait {enc_filename} telah
dihapus.')
    conn.close()
    return redirect(url_for('encrypt'))
```

turnitin

Similarity Report ID: oid:25211:61203287

PAPER NAME
Skripsi_T3120009_Muh_Fajril_A_Rabiu.pdf

AUTHOR
Muh Fajril A. Rabiu
muhammadfajrilfadil@gmail.com

WORD COUNT
6445 Words

CHARACTER COUNT
31370 Characters

PAGE COUNT
46 Pages

FILE SIZE
1.9MB

SUBMISSION DATE
Jun 11, 2024 4:16 PM GMT+8

REPORT DATE
Jun 11, 2024 4:17 PM GMT+8

28% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- **28% Internet database**
- **3% Publications database**
- **Crossref database**
- **Crossref Posted Content database**
- **1% Submitted Works database**

Excluded from Similarity Report

- **Bibliographic material**
- **Quoted material**
- **Cited material**
- **Small Matches (Less then 30 words)**

Summary

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS ICHSAN GORONTALO
FAKULTAS ILMU KOMPUTER
SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001
Jl. Achmad Najamuddin No. 17 Telp. (0435) 829975 Fax (0435) 829976 Gorontalo

SURAT REKOMENDASI BEBAS PLAGIASI
No. **163/FIKOM-UIG/R/VI/2024**

Yang bertanda tangan di bawah ini :

Nama	: Irvan Abraham Salihi, M.Kom
NIDN	: 0928028101
Jabatan	: Dekan Fakultas Ilmu Komputer

Dengan ini menerangkan bahwa :

Nama Mahasiswa	:	Muh. Fajril A. Rabiu
NIM	:	T3120009
Program Studi	:	Teknik Informatika (S1)
Fakultas	:	Fakultas Ilmu Komputer
Judul Skripsi	:	Aplikasi Enkripsi Dan Deskripsi Data Gambar Menggunakan Algoritma Advanced Standard

Sesuai hasil pengecekan tingkat kemiripan skripsi melalui aplikasi **Turnitin** untuk judul skripsi di atas diperoleh hasil *Similarity* sebesar **28%**, berdasarkan Peraturan Rektor No. 32 Tahun 2019 tentang Pendeteksian Plagiat pada Setiap Karya Ilmiah di Lingkungan Universitas Ichsan Gorontalo dan persyaratan pemberian surat rekomendasi verifikasi calon wisudawan dari LLDIKTI Wil. XVI, bahwa batas kemiripan skripsi maksimal 30%, untuk itu skripsi tersebut di atas dinyatakan **BEBAS PLAGIASI** dan layak untuk diujiankan.

Demikian surat rekomendasi ini dibuat untuk digunakan sebagaimana mestinya.

Gorontalo, 19 Juni 2024

Mengetahui
Dekan,

Irvan Abraham Salihi, M.Kom
NIDN. 0928028101

Tim Verifikasi,

Zulfrianto Y. Lamasigi, M.Kom
NIDN. 0914089101

Terlampir :
Hasil Pengecekan Turnitin

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS ICHSAN GORONTALO
FAKULTAS ILMU KOMPUTER
UPT. PERPUSTAKAAN FAKULTAS
SK. MENDIKNAS RI NO. 84/D/0/2001
Jl. Achmad Nadjamuddin No.17 Telp(0435) 829975 Fax. (0435) 829976 Gorontalo

SURAT KETERANGAN BEBAS PUSTAKA

No : 034/Perpustakaan-Fikom/VI/2024

Perpustakaan Fakultas Ilmu Komputer (FIKOM) Universitas Ichsan Gorontalo dengan ini menerangkan bahwa :

Nama Anggota	:	Muh. Fajril A. Rabiu
No. Induk	:	T3120009
No. Anggota	:	M202454

Terhitung mulai hari, tanggal : Rabu, 19 Juni 2024, dinyatakan telah bebas pinjam buku dan koleksi perpustakaan lainnya.

Demikian keterangan ini di buat untuk di pergunakan sebagaimana mestinya.

Gorontalo, 19 Juni 2024
Mengetahui,
Kepala Perpustakaan

Apriyanto Alhamad, M.Kom
NIDN : 0924048601

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS ICHSAN GORONTALO
LEMBAGA PENELITIAN
Kampus Unisan Gorontalo Lt.3 - Jln. Achmad Nadjamuddin No. 17 Kota Gorontalo
Telp: (0435) 8724466, 829975 E-Mail: lembagapenelitian@unisan.ac.id

SURAT KETERANGAN

Nomor : 5129/SK/LEMLIT-UNISAN/GTO/V/2024

Yang bertanda tangan di bawah ini :

Nama	: Dr. Rahmisyari, ST.,SE.,MM
NIDN	: 0929117202
Jabatan	: Ketua Lembaga Penelitian

Menerangkan bahwa :

Nama Mahasiswa	: Nuh. Fajril A Rabiu
NIM	: T3120009
Fakultas	: Fakultas Ilmu Komputer
Program Studi	: Teknik Informatika
Judul Penelitian	: APLIKASI ENKRIPSI DAN DEKRIPSI DATA GAMBAR MENGGUNAKAN ALGORITMA ADVANCED ENCRYPTION STANDARD

Adalah benar telah melakukan pengambilan data penelitian dalam rangka Penyusunan Proposal/Skripsi pada KETUA PROGRAM STUDI TEKNIK INFORMATIKA UNIVERSITAS ICHSAN GORONTALO.

Gorontalo, 15 Mei 2024
Ketua,

Dr. Rahmisyari, ST.,SE.,MM
NIDN 0929117202

BIODATA MAHASISWA

Nama	: Muh Fajril A. Rabiu
Tempat/tgl lahir	: Gorontalo, 28 Mei 2002
Jenis kelamin	: Pria
Alamat	: Kel. Lamo, Kec. Batui
Agama	: Islam
Status perkawinan	: Belum menikah
Pekerjaan	: Mahasiswa
Kewarganegaraan	: WNI

RIWAYAT PENDIDIKAN

SD	: SD INPRES LAMO
SMP	: SMP N 6 BATUI
SMA	: SMA N 1 BATUI