BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi digital saat ini telah membawa kemajuan signifikan dalam berbagai sektor, termasuk komunikasi, bisnis, dan hiburan. Salah satu media digital yang paling banyak digunakan adalah gambar yang banyak digunakan dalam berbagai aplikasi seperti fotografi, media sosial, kesehatan, serta keamanan dan pengawasan. Namun, dengan meningkatnya penggunaan gambar digital, muncul pula ancaman serius terhadap privasi dan keamanan data, terutama dalam hal pencurian, manipulasi, atau penyebaran gambar tanpa izin.

Pengamanan citra digital menjadi hal yang sangat penting untuk melindungi informasi sensitif yang terkandung dalam gambar. Salah satu teknik umum yang digunakan untuk menjaga kerahasiaan data adalah dengan kriptografi. Kriptografi merupakan ilmu ataupun seni mengamankan data dengan cara mengenkripsinya sehingga tidak bisa diakses oleh pihak yang tidak berwenang (Budiman et al., 2023).

Di antara berbagai algoritma kriptografi yang ada, Advanced Encryption Standar atau yang disingkat dengan AES merupakan salah satu algoritma yang diakui secara luas karena tingkat keamanannya yang tinggi, efisiensi, dan kecepatan dalam memproses data. AES telah menjadi standar enkripsi yang disetujui oleh NIST (National Institute of Standards and Technology) dan digunakan dalam berbagai aplikasi, termasuk komunikasi data, pengamanan jaringan, dan pengolahan citra digital.

AES bekerja dengan mengenkripsi blok data berukuran 128-bit dan menggunakan kunci sepanjang 128, 192, atau 256-bit, yang memberikan

fleksibilitas dalam tingkat keamanan. Penggunaan AES dalam pengamanan citra digital menawarkan berbagai keuntungan, termasuk kemampuannya untuk menjaga kualitas gambar serta melindungi informasi di dalam gambar dari akses yang tidak sah.

Namun, meskipun algoritma AES telah terbukti andal untuk enkripsi data umum, implementasinya pada citra digital memerlukan penelitian lebih lanjut untuk memastikan bahwa proses enkripsi tidak merusak kualitas citra dan tetap efisien secara komputasi. Oleh karena itu, penelitian ini bertujuan untuk mengimplementasikan algoritma AES pada citra digital dan menganalisis performanya dalam menjaga keamanan serta mempertahankan kualitas gambar.

Dengan semakin tingginya resiko terhadap privasi dan keamanan dalam dunia digital, solusi enkripsi yang efektif seperti AES sangat diperlukan, terutama untuk melindungi citra digital yang seringkali mengandung informasi pribadi atau rahasia.

1.2. Rumusan Masalah

Adapun rumusan masalah pada penelitian ini yaitu : Bagaimana cara mengimplementasikan algoritma Advanced Encryption Standard (AES) untuk mengenkripsi dan mendekripsi citra digital secara efektif?

1.3. Batasan Masalah

Adapun batasan masalah pada penelitian ini yaitu :

1. Penelitian ini hanya menggunakan algoritma Advanced Encryption Standard (AES) dengan variasi ukuran kunci 128-bit, 192-bit, dan 256-bit.
2. Jenis Data yang digunakan pada penelitian ini yaitu pada citra digital dengan format umum seperti JPEG, PNG, dan JPG.

3. Implementasi algoritma AES untuk pengamanan citra digital dilakukan menggunakan Bahasa pemrograman Python.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut :

1. Mengimplementasikan algoritma Advanced Encryption Standard (AES) untuk mengenkripsi dan mendekripsi citra digital guna meningkatkan keamanan data visual.
2. Menguji dan membandingkan performa AES dengan berbagai ukuran kunci (128-bit, 192-bit, dan 256-bit) terhadap kecepatan pemrosesan dan tingkat keamanan citra digital.

1.5. Metode Penelitian

Dalam pelaksanaan penelitian ini penulis melakukan eksplorasi terhadap konsep keamanan data gambar yang akan dienkripsi dan dekripsi.

a. Studi Literatur

   Penulis melakukan studi literatur terkait algoritma AES dan teknik enkripsi citra digital. Penulis juga akan mencakup pemahaman tentang bagaimana AES bekerja pada data berbentuk gambar serta perbandingan dengan algoritma kriptografi lainnya

b. Pengumpulan Data Citra Digital

   Pada penelitian ini data yang digunakan adalah data citra digital dengan format JPG, JPEG, dan PNG dengan ukuran variasi resolusi (128x128, 256x256, dan 512x512 piksel). Gambar yang dipilih akan diujikan dalam proses enkripsi dan dekripsi.

c. Implementasi Algoritma AES

Algoritma AES akan diimplementasikan pada citra digital menggunakan Bahasa pemrograman Python.

1.6. Sistematika Penulisan

Sistematika penyusunan skripsi ini dibagi menjadi lima bab, sesuai dengan sistematika/ketentuan dalam pembuatan skripsi, adapun pembagian bab-bab tersebut adalah:

BAB I : PENDAHULUAN

Pada bab ini diuraikan secara ringkas pembahsan tentang Latar Belakang, Identifikasi Masalah, Ruang Lingkup Masalah, Maksud dan Tujuan, Metode penelitian , dan Sistematika Penulisan.

BAB II : LANDASAN TEORI

Didalam bab ini diuraikan sekilas tentang pengertian Kriptografi, Algoritma AES, Sekilas Tentang Python.

BAB III : METODE PENELITIAN

Dalam bab ini membahas tentang analisis terhadap permasalahan yang ada, melakukan Perancangan Sistem berdasarkan hal analisis tersebut dan mengadakan testing terhadap sistem tersebut.

BAB IV : ANALISA DAN PEMBAHASAN

Bab ini membahas analisis serta pembahasan Implementasi Algoritma Advanced Encryption Standard (Aes) Untuk Pengamanan Citra Digital dengan menggunakan Bahasa pemrograman python.

BAB V : KESIMPULAN DAN SARAN

Pada bab ini membuat Kesimpulan yang merupakan rangkuman dari hasil analisis kerja pada bagian sebelumnya dan saran yang perlu diperhatikan berdasarkan keterbatasan yang ditentukan dan asumsi-asumsi yang dibuat selama pembuatan aplikasi ini.

BAB II

LANDASAN TEORI

2.1 Kriptografi

Kriptografi adalah cabang ilmu komputer yang berhubungan dengan keamanan informasi digital. Kriptografi mengelola data dengan cara menjaga kerahasiaan dan integritas selama dan setelah proses transmisi. Algoritma kriptografi dibagi menjadi dua bagian berdasarkan kuncinya kriptografi simetris dan kriptografi kunci publik. Kriptografi simetris memiliki kunci yang sama untuk enkripsi dan dekripsi data, sedangkan kriptografi asimetris terdiri dari dua kunci yaitu kunci publik untuk enkripsi data dan kunci privat untuk dekripsi data (Budiman et al., 2023).

Kriptografi berkaitan dengan merancang dan menggunakan kode (atau *cipher*) yang memungkinkan kedua belah pihak mengirim pesan secara tersembunyi dari seorang hacker yang dapat memantau semua komunikasi antara mereka. Dalam bahasa modern, kode disebut skema enkripsi dan begitulah terminologi kriptografi. Keamanan dari skema enkripsi klasik mengandalkan rahasia dan juga kunci yang dibagikan oleh pihak yang berkomunikasi sebelumnya dan tidak diketahui oleh penyadap.

Dalam konteks enkripsi, dua pihak berbagi kunci dan menggunakan kunci itu ketika mereka ingin berkomunikasi. Satu pihak dapat mengirim pesan atau data dengan menggunakan kunci bersama untuk mengenkripsi pesan dan dengan demikian mendapatkan *ciphertext* yang dikirimkan ke penerima. Penerima menggunakan kunci yang sama untuk mendekripsi atau mengurai *ciphertext* dan mengubah *plaintext*. Kunci yang sama digunakan untuk mengonversi *plaintext* menjadi *ciphertext* dan sebaliknya (Katz & Lindell, 2021).

Ada 4 tujuan utama pada kriptografi yaitu :

1. *Confidentiality or privacy*, yaitu menjaga isi informasi dari siapapun kecuali orang yang memiliki hak atau kunci rahasia untuk membuka informasi yang telah di enkripsi.
2. *Integrity*, yaitu memberikan jaminan bahwa setiap bagian pesan tidak akan mengalami perubahan dari saat data telah dibuat ataupun dikirim oleh pengirim sampai saat dengan data tersebut dibuka oleh orang yang berhak menerima data.
3. *Authentication*, yaitu mengidentifikasi kebenaran pihak-pihak yang berkomunikasi maupun mengidentifikasi kebenaran sumber pesan.
4. *Non-repudiation*, yaitu mencegah suatu entitas menyagkal tindakan sebelumnya (Paul & Maitra, 2011).

Pada penelitian ini penulis fokus pada *Confidentiality* or *privacy* yaitu menjaga isi pesan ataupun data dari siapapun kecuali orang yang berhak menerimanya.

2.2 Algoritma *Advance Encryption Standart* (AES)

Algoritma *Advance Encryption Standart* atau yang disingkat dengan AES merupakan metode enkripsi blok yang menggunakan panjang blok 128 bit. AES menggantikan Algoritma DES (*Data Encryption Standard*) dan dikenal karena keamanannya yang lebih tinggi. Proses enkripsi dan dekripsi yang digunakan AES melibatkan beberapa *round* transformasi, yang mencakup operasi seperti substitusi *byte*, pergeseran baris, pencampuran kolom, dan operasi XOR dengan kunci (Agita et al., 2024).

Kriptografi *Advanced Encryption Standard* (AES), khususnya AES-256, digunakan untuk melindungi data melalui proses enkripsi dan dekripsi, menjamin

kerahasiaan serta integritas informasi. Sebagai algoritma *block cipher* simetris dengan panjang kunci 256 bit, AES-256 menawarkan tingkat keamanan yang lebih tinggi dibandingkan pendahulunya seperti Data *Encryption* Standard (DES) (Papilaya & Pradana, 2024).

Algoritma AES adalah blok *chipertext* simetrik yang dapat mengenkripsi (*encipher*) dan dekripsi (*decipher*) informasi. Enkripsi merubah data yang tidak dapat lagi dibaca disebut ciphertext; sebaliknya dekripsi adalah merubah *ciphertext* data menjadi bentuk semula yang kita kenal sebagai *plaintext*. Algoritma AES menggunakan kunci kriptografi 128, 192, dan 256 bits untuk mengenkripsi dan dekripsi data. Setiap blok dienkripsi dalam sejumlah putaran tertentu, sebagaimana halnya DES. Berikut adalah banyaknya putaran kunci pada algoritma AES.

Tabel 2. 1 Putaran Kunci Algoritma AES

AES (Bits)	Panjang Kunci (NK Words)	Ukuran Bloks (Nb Words)	Jumlah Putaran (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Table 2.1 menunjukkan besaran bits, Panjang kunci, ukuran blok dan jumlah putaran pada algoritma AES. AES menetapkan panjang kunci adalah 128, 192 dan 256 maka dikenal sebagai AES-128, AES-192 dan AES-256. AES memiliki panjang kunci paling sedikit yaitu 128 bits, namun AES tetap tahan terhadap serangan *exhaustive key search* dengan teknologi saat ini. Dengan panjang kunci 128 bits maka terdapat sebanyak $2^{128}= 3,4 \times 10^{38}$ kemungkinan kunci. AES menggunakan substitusi dan permutasi dalam sejumlah putaran atau *cipher* berulang. Setiap putaran menggunakan kunci internal yang berbeda. Empat proses

utama algoritma AES yaitu sebagai berikut :

1. *SubBytes* (Transformasi Subtitusi *Byte*)
2. *ShiftRow* (Transformasi Pergeseran Baris)
3. *MixColumns* (Transformasi Pencampuran Kolom)
4. *Addroundkey* (Transformasi Penambahan Kunci)

Contoh Perhitungan Algoritma AES Pada Citra

Misalkan kita ingin melakukan enkripsi Gambar :

Gambar 2. 1 Contoh Gambar Yang Akan Dienkripsi

Gambar 2.1 menunjukkan sebuah contoh gambar yang akan dienkripsi dengan menggunakan algoritma AES. Gambar tersebut memiliki dimensi 1280 pixels x 720 pixels dengan ukuran file 114kb. Gambar tersebut akan dienkripsi dengan menggunakan algoritma AES sebagai berikut :

1. **Konversikan gambar kedalam biner :**
    - Ubah data pixel gambar menjadi format binner.
    - Format tersebut akan dijadikan inputan algoritma AES.
2. **Pilih ukuran kunci AES :** (128, 192, atau 256 bit) dalam contoh ini saya menggunakan kunci 192 bit.

3. **Langkah enkripsi:**

   - Langkah ini ini melibatkan :
     - ***Subbytes* :** Mengganti setiap *byte* menggunakan table subsitusi (S-box)
     - ***ShiftRows* :** Menggeser baris-baris matriks data.
     - ***MixColumns* :** Mengalikan matriks dengan matriks khusus.
     - ***AddRoundKey* :** Meng-XOR data dengan subkunci.

4. **Input Gambar :**

   - Data gambar (16 Byte) = **"3A, B0. D5, 37, AE, D4, 32, AB, D0, 2B, A8, CC, 28, A6, CC 2D".**
   - Pada contoh ini saya menggunakan kunci **"abcdefghijklmnopqrstuvwx"** (24 karakter ASCII).

5. **Perhitungan Manual :**

   Matriks *Plaintext* Gambar :

3A	B0	D5	37
AE	D4	32	AB
D0	28	A8	CC
28	A6	CC	2D

   Matriks kunci awal 192:

61	62	63	64	65	66
67	68	69	6A	6B	6C
6D	6E	6F	70	71	72
73	74	75	76	77	78

6. Add*Roundkey*

Blok *Plaintext* di-XOR dengan bagian pertama dari kunci (16 *byte* pertama) :

34 ⊕ 61 = 5B

B0 ⊕ 62 = D2

D5 ⊕ 63 = B6

37 ⊕ 64 = 53

AE ⊕ 65 = CB

D4 ⊕ 66 = B2

32 ⊕ 67 = 55

AB ⊕ 68 = C3

D0 ⊕ 69 = B9

2B ⊕ 6A = 41

A8 ⊕ 6B = C3

CC ⊕ 6C = A0

28 ⊕ 6D = 05

A6 ⊕ 6E = C8

CC ⊕ 6F = A3

2D ⊕ 70 = 5D

Hasil *AddRoundkey :*

5B	D2	B6	53
CB	B2	55	C3
B9	41	C3	A0
05	C8	A3	5D

7. SubBytes

Setiap *byte* diganti menggunakan table substitusi S-Box (lihat table S-box AES untuk detail).

Tabel 2. 2 S-box

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	63	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	4	C7	23	C3	1B	9	D5	9A	07	12	B0	E2	EB	27	B2	75
4	09	C0	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	8A	20	7C	B1	5B	6A	CB	BE	ED	4A	7A	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	D2	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	43	32	3A	4E	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	B1	37	6D	BD	D5	CB	A9	6C	19	F4	EA	65	7A	AE	08
C	BA	78	25	7C	1C	A6	B4	C6	1F	DD	74	6D	4B	BD	8B	8A
D	B5	3E	70	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	BE	94	9B	1E	B7	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Setiap byte diganti menggunakan tabel substitusi S-Box (lihat tabel S-Box AES untuk detail). Sebagai contoh Byte pertama (5B):

Baris ke-5 (hex 5) dan kolom ke-11 (hex B) dalam S-Box.

Substitusi menghasilkan: ED

Baris D kolom 2 menghasilkan 70.

Adapun hasil *subbytes* adalah sebagai berikut :

ED 70 CB 8A

6D B1 7C 3E

19 C0 3E 43

63 1F 4E 7A

8. *ShiftRows*

Setiap baris matriks digeser ke kiri :

- Baris 0 : Tidak Digeser.
- Baris 1 : Geser 1 Posisi ke kiri.
- Baris 2 : Geser 2 Posisi ke kiri.
- Baris 3 : Geser 3 Posisi ke kiri.

Hasil setelah *ShiftRows* :

ED	70	CB	8A
B1	7C	3E	6D
7C	43	19	C0
7A	63	1F	4E

9. *MixColumns*

Kolom matriks dikalikan dengan matriks tetap AES dalam ruang GF(2⊕8).

Contoh untuk kolom pertama:

ED

B1

7C

7A

Kalikan dengan matriks tetap :

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

Proses perhitungan dilakukan dengan operasi $\oplus$ dan shift :

$ED \times 02 \oplus B1 \times 03 \oplus 7C \times 01 \oplus 7A \times 01.$

Lakukan perhitungan yang sama untuk seluruh kolom tersebut.

Hasil *MixColumns(Ciphertext)* :

47	20	14	85
BC	99	34	2E
A1	67	22	6C
8F	F4	98	11

Hasil akhir dari *MixColumns* tersebut merupakan Hasil enkripisi (*Ciphertext)*

10. Proses Dekripsi

Untuk melakukan dekripsi maka kita harus mengetahui hasil enkripsi *Ciphertext.*

Hasil Enkripsi (*Ciphertext)* :

47	20	14	85
BC	99	34	2E
A1	67	22	6C
8F	F4	98	11

Langkah 1 : AddroundKey

Ciphertex di XORkan dengan subkunci dari ronde terakhir. Subkunci terakhir diperoleh dari ekspansi kunci AES.

Misalkan subkunci terakhir adalah :

6B	77	81	94
A5	D2	C8	38
C3	F0	11	5F
2A	B4	9E	0D

Proses XOR untuk setiap *byte* :

47 ⊕ 6B = 2C

20 ⊕ 77 = 57

14 ⊕ 81 = 95

85 ⊕ 94 = 11

BC ⊕ A5 = 19

99 ⊕ D2 = 4B

34 ⊕ C8 = FC

2E ⊕ 3B = 15

A1 ⊕ C3 = 62

67 ⊕ F0 = 97

22 ⊕ 11 = 33

6C ⊕ 5F = 33

8F ⊕ 2A = A5

F4 ⊕ B4 = 40

98 ⊕ 9E = 06

11 ⊕ 0D = 1C

Hasil setelah dilakukan AddroundKey :

2C	57	95	11
19	4B	FC	15
62	97	33	33
A5	40	06	1C

Langkah 2 *Inverse ShiftRows :*

2C	**57**	**95**	**11**
19	**4B**	**FC**	**15**
62	**97**	**33**	**33**
A5	**40**	**06**	**1C**

Proses Inverse ShiftRows :

- Baris 0 : Tidak digeser.
- Baris 1 : Geser 1 Posisi Kanan.
- Baris 2 : Geser 2 Posisi Kanan.
- Baris 3 : Geser 3 Posisi Kanan.

Hasil setelah Inverse ShiftRows :

2C	57	95	11
15	19	4B	FC
33	33	62	9
1C	A5	40	06

Langkah 3 Inverse SubBytes :

Setiap *byte* diganti dengan menggunakan table *Inverse S-Box.* Proses ini kebalikan dari *SubBytes.* Adapun table S-Box dapat dilihat pada table 2.3 berikut ini :

Tabel 2. 3 Tabel S-Box Invers SubBytes

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	63	7C	77	7B	F2	6B	5E	C5	30	01	67	2B	FE	D7	AB	76
1	CA	F8	C9	7D	FA	1E	47	F0	AD	2D	A2	AF	3D	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	7B	D8	31	15
3	4	C7	23	4C	1B	9	D5	9A	07	12	B0	E2	EB	27	B2	75
4	27	83	2C	1A	1B	6E	5A	A0	52	3B	D6	0A	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	D4	6A	CB	BE	ED	4A	4C	58	CF

6	D0	EF	78	FB	43	4D	33	85	45	F9	D2	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	9B	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	6F	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	BD	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	70	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	BE	94	9B	1E	B7	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	3F	54	BB	16

Contoh :

Byte dari baris pertama : 2C

Pada table *Inverse S-Box,* baris ke-2 (hexadecimal) dan kolom ke-12 (hexadecimal) memberikan *byte* 7B

7B	D4	E1	F8
1E	2D	0A	3F
4C	A5	78	9B
3D	6F	27	5E

Langkah 4 *Invers MixColumns*

Setiap kolom matriks dikalikan dengan matriks *Invers MixColumns* diruang GF(2^8). Matriks *Inverse MixColumns* adalah:

0E	0B	0D	09
09	0E	0B	0D
0D	09	0E	0B
0E	09	0D	0B

Contoh untuk kolom pertama yang akan dilakukan *Inverse MixColumns:*

7B

1E

4C

3D

Dikalikan dengan matriks *Inverse MixColumns :*

0E × 7B ⊕ 0B × 1E ⊕ 0D × 4C ⊕ 09 × 3D

09 × 7B ⊕ 0E × 1E ⊕ 0B × 4C ⊕ 0D × 3D

0D × 7B ⊕ 09 × 1E ⊕ 0E × 4C ⊕ 0B × 3D

0B × 7B ⊕ 0D × 1E ⊕ 09 × 4C ⊕ 0E × 3D

Lakukan operasi ini untuk semua kolom. Hasil setelah *Inverse MixColumns :*

3A	B0	D5	37
AE	D4	32	AB
D0	2B	A8	CC
28	A6	CC	2D

setelah proses *Inverse MixColumns* diulang sebanyak 11 ronde, maka didapatlah hasil akhir ataupun *plaintext* asli yaitu:

3A	**B0**	**D5**	**37**
AE	**D4**	**32**	**AB**
D0	**2B**	**A8**	**CC**
28	**A6**	**CC**	**2D**

2.3 Citra Digital

Citra Digital (*Digital Image Processing*) adalah disiplin ilmu yang mempelajari teknik dalam mengolah gambar diam ataupun gambar yang bergerak (Ratna, 2020). Pengolahan citra mempunyai keterikatan yang erat dengan disiplin ilmu yang jika sebuah disiplin ilmu dinyatakan dalam bentuk proses suatu input menjadikan output,

maka pengolahan citra memiliki input berupa citra serta output berupa citra (Rilo Pambudi et al., 2020).

Suatu citra dapat didefinisikan sebagai fungsi dua dimensi, f (x,y), dengan x dan y adalah koordinat spasial (bidang), dan amplitudo f pada sembarang pasangan koordinat (x,y) disebut intensitas atau tingkat abu-abu dari gambar pada saat itu. Ketika x,y, dan nilai intensitas f semuanya berhingga, besaran diskrit, kita menyebut bayangan itu sebagai bayangan digital.

Bidang pemrosesan gambar digital mengacu pada pemrosesan gambar digital dengan menggunakan komputer digital (Gonzalez & Woods, 2018).

Pada umumnya citra digital menjadi 3 macam, diantaranya adalah :

1. *Binary Image*

   Binary Image adalah jenis citra digital yang hanya terdiri dari warna hitam dan putih. Disebut binary karena hanya ada dua warna untuk setiap pikselnya, maka hanya perlu 1 bit untuk masing – masing pikselnya (0 dan 1).

2. *Black and White*

   Black and White atau citra hitam putih atau grayscale adalah citra dimana setiap pikselnya memiliki gradiasi mulai dari warna putih hingga hitam. Dengan demikian masing masing piksel dapat diwakili oleh 8 bit atau 1 byte.

3. *Colour Image* atau RGB (*Red, Green, Blue*)

   Colour Image atau RGB (*Red, Green, Blue*) atau citra berwarna yang dimana setiap piksel memiliki warna tertentu dimana warna tersebut merupkan representasi dari warna merah (*Red*), hijau (*Green*), biru (*Blue*). Setiap masing masing warna tersebut memiliki range antara 0 - 255, dengan total yaitu 2553 = 16.581.375 variasi warna berbeda pada sebuah gambar. Pada dasarnya color

image atau citra berwarna ini terdiri dari tiga buah matriks yang mewakili nilai R, G dan B atau merah, hijau dan biru untuk masing masing pikselnya (Mahesa et al., 2019).

Citra digital adalah data dalam bentuk gambar yang disimpan dalam format tertentu, seperti JPEG, PNG, atau BMP. Pengamanan citra digital dengan menggunakan algoritma AES bertujuan untuk mengubah citra tersebut menjadi bentuk yang tidak dapat dikenali tanpa kunci dekripsi yang tepat. Enkripsi citra menggunakan AES sangat penting dalam konteks pengamanan data, terutama di era digital yang marak dengan ancaman keamanan.

Enkripsi citra dapat dilakukan dengan membagi citra menjadi blok-blok piksel yang lebih kecil. Setiap blok ini kemudian dienkripsi secara terpisah menggunakan algoritma AES. AES akan mengenkripsi informasi citra (seperti warna, intensitas piksel, dan metadata) menggunakan kunci yang telah ditentukan. Hasil enkripsi ini adalah citra yang tampak seperti noise atau acak, yang tidak dapat dipahami tanpa proses dekripsi yang benar.

Implementasi AES dalam pengamanan citra digital melibatkan beberapa langkah, yaitu:

1. **Pemeriksaan Gambar:** Gambar yang akan dienkripsi dibaca dan diubah menjadi representasi digital (biasanya dalam bentuk array piksel).
2. **Pengolahan Gambar:** Gambar dibagi menjadi blok-blok piksel (biasanya blok 128 bit atau lebih kecil) untuk mempersiapkan data untuk proses enkripsi.

3. **Enkripsi dengan AES:** Setiap blok gambar dienkripsi menggunakan algoritma AES dengan kunci yang telah dipilih. Proses enkripsi ini akan menghasilkan citra yang terenkripsi.
4. **Dekripsi:** Untuk memulihkan gambar asli, dekripsi dilakukan menggunakan kunci yang sama untuk mengembalikan ciphertext ke bentuk plaintext (gambar asli).

2.4 *Flowchart*

Flowchart adalah salah satu visualisasi aliran logis yang berlaku untuk sebagian besar disiplin ilmu. Ini cukup standar. Sintaksnya sangat sederhana, yang membuat bahasanya unik itu sangat kuat sehingga Anda bisa idealnya membuat konsep aliran logis apa pun dengannya.

Sebagaimana dinyatakan di atas, diagram alur dapat digunakan untuk membangun aliran logis dari program perangkat lunak. *Flowchart* memudahkan pengembang untuk memahami, men-debug, dan memeriksa validitas kode untuk pengembang. Ada banyak paket perangkat lunak untuk menghasilkan kode dari diagram *Flowchart* .

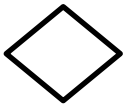
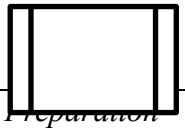
Tapi tidak ada cara untuk membuat diagram alur dari kode sumber. Bagaimana jika kita memodelkan solusi untuk mengonversi kode sumber menjadi diagram alur yang dapat memudahkan pengembang perangkat lunak dalam memahami, men-debug, dan memvalidasi kode.

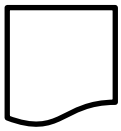
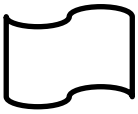
Meskipun *Flowchart* adalah alat diagram, itu dapat direpresentasikan dalam sintaks seperti grafik terarah. Masalah yang dihadapi adalah semacam masalah terjemahan/transformasi. Masalah terjemahan bahasa pemrograman biasanya diselesaikan menggunakan kompiler. Karena masalah yang dihadapi adalah

menerjemahkan kode sumber program ke diagram alur, kami mengusulkan solusi untuk meniru arsitektur kompiler dalam teori compiler (Pan et al., n.d.).

Adapun arti dari symbol *Flowchart* adalah sebagai berikut :

Tabel 2. 4 Simbol Flowchart

Simbol	Arti
Input / Output 	Mempresentasikan *input* data atau *output* data yang diproses atau informasi
Proses 	Mempresentasikan Operasi
Penghubung 	Keluar ke atau masuk dari bagian lain *Flowchart* khususnya halaman yang sama
Anak Panah 	Mempresentasikan alur kerja
Keputusan 	Keputusan dalam program
Predefined Process 	Rincian operasi berada di tempat lain
Preparation 	Pemberian harga awal
Punched Card 	Input / Output yang menggunakan kartu berlubang

Dokumen 	Input / Output dalam format yang dicetak
Magnetic Disk 	*Input* / *Output* yang menggunakan *Disk Magnetic*
Magnetic Drum 	*Input* / *Output* yang menggunakan *Drum Magnetic*
Punched Tape 	*Input* / *Output* yang menggunakan pita kertas berlubang
Manual Input 	*Input* yang dimasukkan secara manual dari *keyboard*
Display 	*Output* yang ditampilkan pada terminal
Manual Operation 	Operasi Manual

2.4 Python

Python adalah bahasa pemrograman interpretative multiguna dengan filosofi perancagan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif (Sinaga, 2017).

Python diciptakan pertama kali oleh Guido Van Rossum di belanda pada tahun 1990 dan namanya diambildari acara televisikesukaanGuido Monty Python's Flying Circus. Van Rossum mengembangkan Python sebagaihobi, kemudian Python menjadi bahasa pemrograman yang dipakai secara luas dalam industri dan pendidikan karena sederhana, ringkas, sintak sintuitif dan memiliki pustaka yang luas (Romzi & Kurniawan, 2020).