BAB 2 TINJAUAN PUSTAKA

2.1. Kriptografi

Kriptografi diambil dari 2 kata Yunani, *crypto* dan *graphia*. *Crypto* yang memiliki arti kata rahasia dan *graphia* yang juga berarti tulisan. Secara istilah enkripsi adalah ilmu yang menjamin kerahasiaan dan keamanan pesan yang dikirim dari satu lokasi ke lokasi lain.

Kriptografi adalah ilmu yang mempelajari faktor keamanan informasi: tingkat kepercayaan, integritas data, otentikasi entitas, dan teknik matematika yang terkait dengan otentikasi keaslian data. Karena seni didefinisikan dengan cara yang berbeda untuk setiap individu untuk melindungi data, pesannya memiliki makna estetisnya sendiri dan ada juga hubungan antara seni dan budaya. Jika dicermati, dalam kriptografi, grafis berarti seni. Keamanan harus menggunakan teknologi dan teknologi serta keamanan data dan file. Keandalan keamanan sangat bergantung pada bagaimana anda memahami pentingnya data yang anda amankan.

2.2. Enkripsi

Kriptografi merupakan bagian integral dari dunia kriptografi. Enkripsi adalah metode yang paling umum digunakan untuk melindungi data untuk memastikan kerahasiaan data. Pesan asli disebut teks biasa, yang menerjemahkan pesan asli ke dalam kode yang tidak dapat dipahami. Kriptografi dapat diartikan dalam kriptografi atau kode. Cari kata atau frasa dalam kamus atau glosarium, seperti yang Anda lakukan ketika Anda tidak memahami kalimat atau kata dalam sebuah pesan. Tidak seperti enkripsi, mengubah pesan sederhana menjadi kode memerlukan penggunaan algoritme yang dapat menyandikan pesan yang ingin Anda ubah atau lindungi.

Enkripsi adalah proses mengubah kode atau pesan yang dapat dipahami menjadi kode atau pesan rahasia yang tidak dapat dipahami. Dari definisi di atas dapat kita simpulkan bahwa enkripsi adalah teknik mengacak kode biasa menjadi kode anomali sehingga kode aslinya tidak dapat terbaca. Kriptografi sudah ada sejak Perang Romawi, tetapi secara teknis, berbeda dengan kemajuan kriptografi saat ini.

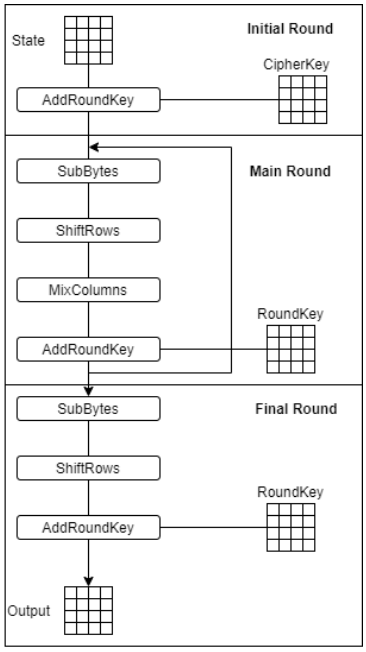
2.3. Algoritma AES (*Advanced Encryption Standard*)

A. Representasi Data

Input dan output dari algoritma AES terdiri dari string data 128-bit. Sebuah string data yang terdiri dari sekelompok 128 bit, juga dikenal sebagai blok data atau teks biasa, dienkripsi dengan ciphertext. Kunci enkripsi AES mencakup kunci 128-bit, 192-bit, dan 256-bit. Urutan bit diberi nomor secara berurutan dari 0 hingga n1. di mana n adalah nomor seri. Urutan 8 bit data berurutan, yang disebut byte, adalah unit dasar operasi yang dilakukan pada blok data.

B. Enkripsi

Proses enkripsi algoritma AES mencakup empat jenis transformasi byte, yaitu SubBytes, ShiftRows, MixColumns dan AddRoundKey. Pada awal proses enkripsi, entri yang disalin ke status mengalami transformasi byte AddRoundKey. State kemudian mengalami transformasi SubBytes, ShiftRows, MixColumns dan AddRoundKey berulang-ulang hingga N. Proses algoritma AES ini disebut fungsi bulat. Babak terakhir sedikit berbeda dengan babak sebelumnya yang merupakan keadaan babak terakhir, yang tidak mengalami transformasi MixColumns.



Gambar 2.1 Proses Enkripsi AES

Proses dari enkripsi algoritma AES adalah sebagai berikut :

a. Transformasi *SubBytes*()

SubBytes merupakan transformasi byte dimana setiap elemen pada state akan dipetakan dengan menggunakan sebuah tabel subtitusi (*S-Box*).

hex	y															
x	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

S-BOX

Gambar 2.2 Tabel S-Box

b. Transformasi *ShiftRows*()

Transformasi ShiftRows pada dasarnya adalah proses pergesaran bit dimana bit paling kiri akan dipindahkan menjadi bit paling kanan (rotasi bit). Transformasi ini juga melakukan pergeseran pada tiga baris terakhir dari *array state*. Jumlah pergeseran bergantung pada nilai baris (*r*). Baris *r* = 1 digeser sejauh 1 *byte*, baris r = 2 digeser sejauh 2 *byte*, dan baris r = 3 digeser sejauh 3 *byte*. Baris r = 0 tidak digeser.

d4	e0	b8	1e
27	bf	b4	41
11	98	5d	52
ae	f1	e5	30

Geser baris r = 1

d4	e0	b8	1e
bf	b4	41	27
11	98	5d	52
ae	f1	e5	30

Hasil pergeseran baris r = 1 dan geser baris r = 2

d4	e0	b8	1e
bf	b4	41	27
5d	52	11	98
ae	f1	e5	30

Hasil pergeseran baris r = 2 dan geser baris r = 3

d4	e0	b8	1e
bf	b4	41	27
5d	52	11	98
30	ae	f1	e5

Hasil pergeseran baris r = 3

Gambar 2.3 Hasil Transformasi Shiftrows

c. Transformasi *MixColumns*()

Transformasi *MixColumns* mengalikan setiap kolom dari array state dengan polinom $a(x)$ mod (x^4+1). Setiap kolom diperlakukan sebagai polinom 4-suku pada GF(2^8). $a(x)$ yang ditetapkan adalah :

$$a(x) = \{03\}x^3+\{01\}x^2+\{01\}x+\{02\}$$

Transformasi dinyatakan sebagai perkalian matriks sebagai berikut :

$$s'(x) = a(x) \otimes s(x)$$

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

$$s'_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$$

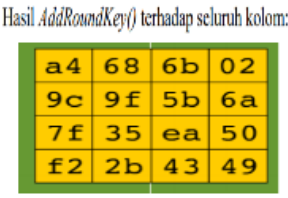
$$s'_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c}$$

$$s'_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{3,c})$$

$$s'_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{3,c})$$

d. Transformasi *AddRoundKey*()

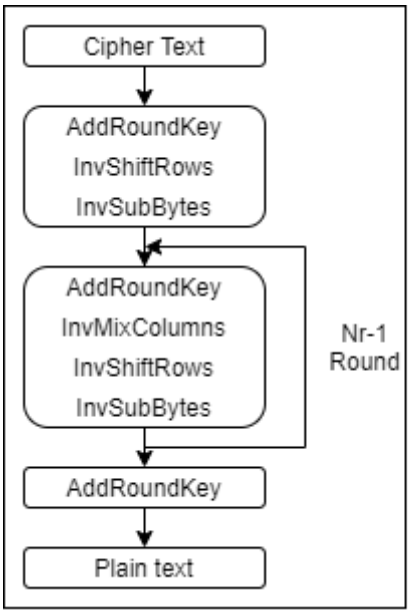
Transformasi melakukan operasi XOR terhadap sebuah round key dengan *array state*, dan hasilnya disimpan di *array state*. Gambar (angka) merupakan dari transformasi around key.



Gambar 2.4 Transformasi AddRoundKey

C. Dekripsi

Transformasi Anda dapat mengimplementasikan cipher dalam arah terbalik dan berlawanan untuk membuat cipher terbalik atau inverse cipher yang mudah dipahami untuk algoritma AES.. Transformasi byte yang digunakan pada invers *cipher* adalah *InvShiftRows*, *InvSubBytes*, *InvMixColums*, dan *InvAddRoundKey*.

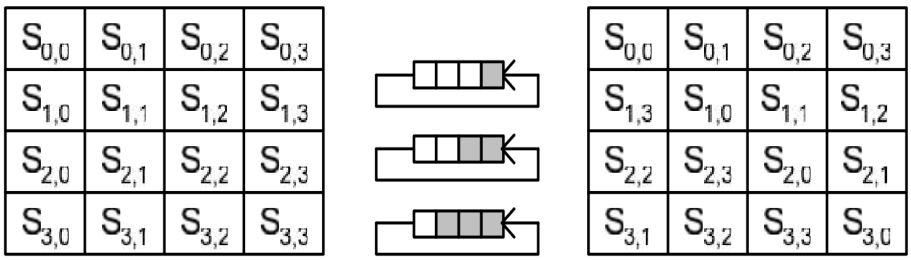


Gambar 2.5 Proses Dekripsi AES

Proses dekripsi dari algoritma AES :

a. Transformasi *InvShiftRows*()

InvShiftRows adalah transformasi byte yang bertentangan dengan transformasi *ShiftRows.* Di *InvShiftRows,* pemindahan bit dilakukan ke kanan sedangkan di *ShiftRows,* pemindahan bit dilakukan ke kiri. Pada baris kedua dilakukan pemindahan bit sebanyak 3 kali, pada baris ketiga dilakukan pemindahan bit sebanyak 2 kali dan pada baris keempat dilakukan sebanyak 1 kali.



Gambar 2.6 Transformasi Shiftrows

b. Transformasi *InvSubBytes*()

InvSubBytes bukan transformasi SubBytes, tetapi transromasi Byte. Di InvSubBytes, setiap elemen dipetakan menggunakan tabel SBox terbalik atau inverse SBox. Tabel ini berbeda dengan tabel SBox, dimana hasil yang didapat dari tabel ini merupakan hasil dari dua proses dengan urutan yang berbeda. Yaitu, pertama transformasi affine, kemudian perkalian terbalik di GF(2^8).

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	82	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	89	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	84	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	87	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	38	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

Gambar 2.7 Tabel nverse S-Box

c. Transformasi *InvMixColumns*()

Pada *nvMixColumns*, kolom – kolom pada tiap state (*word*) akan dipandang sebagai polinom atas GF(2^8) dan mengalikan dengan modulo x^4+1 dengan polinom tetap $a^{-1}(x)$ yang diperoleh dari :

$$a^{-1}(x) = \{0B\}x^3+\{0D\}x^2 + \{09\}x + \{0E\}.$$

Atau dalam matriks :

$$s'(x) = a(x) \otimes s(x)$$

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

Gambar 2.8 Matriks nvMixColumns

Hasil dari perkalian di atas adalah :

$$s'_{0,c} = (\{0E\} \bullet s_{o,c}) \oplus (\{0B\} \bullet s_{1,c}) \oplus (\{0D\} \bullet s_{2,c}) \oplus (\{09\} \bullet s_{3,c})$$
$$s'_{1,c} = (\{09\} \bullet s_{o,c}) \oplus (\{0E\} \bullet s_{1,c}) \oplus (\{0B\} \bullet s_{2,c}) \oplus (\{0D\} \bullet s_{3,c})$$
$$s'_{2,c} = (\{0D\} \bullet s_{o,c}) \oplus (\{09\} \bullet s_{1,c}) \oplus (\{0E\} \bullet s_{2,c}) \oplus (\{0B\} \bullet s_{3,c})$$
$$s'_{3,c} = (\{0B\} \bullet s_{o,c}) \oplus (\{0D\} \bullet s_{1,c}) \oplus (\{09\} \bullet s_{2,c}) \oplus (\{0E\} \bullet s_{3,c})$$

Gambar 2.9 Hasil Perkalian Matriks

d. Transformasi *nvAddRoundKey*()

Transformasi AddRoundKey sama dengan Transformasi AddRoundKey. Transformasi ini hanya melakukan operasi penjumlahan sederhana menggunakan operasi XOR bitwise.

2.4. Kompresi

Kompresi, yang memiliki definisi kompresi atau pengurangan ukuran, kompresi data adalah proses mengubah informasi menggunakan bit atau informasi yang lebih rendah daripada informasi asli yang tidak dikodekan atau dikodekan menggunakan sistem pengkodean tertentu. Selain itu, kompresi data dapat mengurangi ukuran untuk menyimpan informasi ini, atau teknik kompresi data dapat lebih efisien karena ukuran informasi atau data dapat lebih kecil dari ukuran informasi aslinya. Keuntungan dari kompresi data adalah menghemat ruang disk dan penggunaan bandwidth saat mengirim informasi. Namun, kompresi memiliki kelemahan. Sebelum Anda dapat membuka kembali informasi tersebut, Anda harus mengekstraknya terlebih dahulu.

Kompresi data adalah metode yang digunakan untuk memperkecil ukuran data atau informasi dari data aslinya. Kompresi data sering terjadi pada mesin, terutama mesin komputer. Hal ini bisa terjadi karena nilai bit dari simbol yang ditampilkan di komputer berbeda. Misalnya, dalam ASCII, setiap simbol yang ditampilkan memiliki panjang bit 8. Misalnya, kode ASCII A, ketika dikonversi ke biner, memiliki nilai desimal 65, yaitu 010000001. Kompresi data mengurangi jumlah bit yang terkandung dalam setiap simbol yang ditampilkan. Dengan kompresi yang diinginkan untuk memperkecil ukuran data agar dapat disimpan lebih efisien.

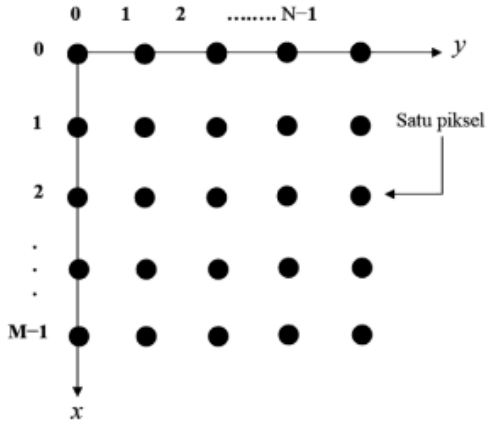
2.5. Algoritma LZW (*Lempel-Ziv-Welch*)

LZW adalah singkatan dari Lempel Ziv Welch. Abraham Lempel, Jacob Jib, Terry Welch. Pencipta algoritma kompresi lossless universal ini. Kelebihan dari algoritma ini adalah dapat diimplementasikan dengan cepat. Kekurangannya adalah tidak optimal karena keterbatasan analisis data. Algoritma ini melakukan kompresi kamus dengan mengganti fragmen teks dengan indeks yang diambil dari "kamus". Pendekatan ini adaptif dan efisien karena memungkinkan penyandian banyak karakter dengan mereferensikan string yang ditampilkan sebelumnya dalam teks. Prinsip kompresi dicapai ketika referensi tipe pointer dapat disimpan dengan bit yang lebih sedikit daripada string asli.

2.6. Citra Digital

A. Representasi Citra Digital

Hasil pemindaian dan kuantifikasi citra diketahui merupakan bilangan real yang membentuk matriks M x N. Artinya ukuran gambar adalah M x N. Sistem koordinat umum yang digunakan dalam teori pemrosesan gambar untuk mewakili gambar. Citra digital diwakili oleh matriks baris M dan kolom, di mana perpotongan baris dan kolom disebut piksel. Piksel memiliki dua parameter: koordinat dan intensitas atau warna. Nilai yang terdapat pada koordinat (x,y) adalah f(x,y), yang merupakan intensitas atau warna piksel pada titik tersebut.



Gambar 2.10 Sistem Koordinat Citra

Artinya sebuah citra digital dapat ditulis dalam bentuk matriks berikut :

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

Gambar 2.11 Matriks Citra Digital

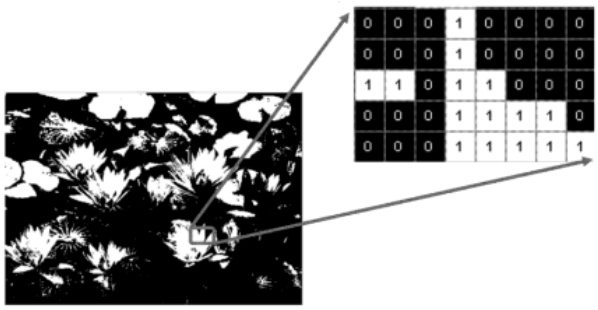
Berdasarkan uraian di atas, sebuah citra digital dapat dideskripsikan secara matematis sebagai fungsi dari besaran f(x,y). Dimana nilai x (baris) dan y (kolom) adalah koordinat posisi dan f (x,y) adalah nilai fungsi pada setiap titik (x,y), intensitas citra atau keabuan piksel. Tentukan tingkat atau warna isian.

B. Tipe Citra Digital

Ada beberapa tipe citra digital yang sering digunakan untuk penelitian, di antaranya adalah :

a. Citra Biner

Gambar biner hanya membutuhkan satu bit memori per piksel. Oleh karena itu, setiap piksel hanya memiliki dua kemungkinan nilai intensitas: 1 atau 0.

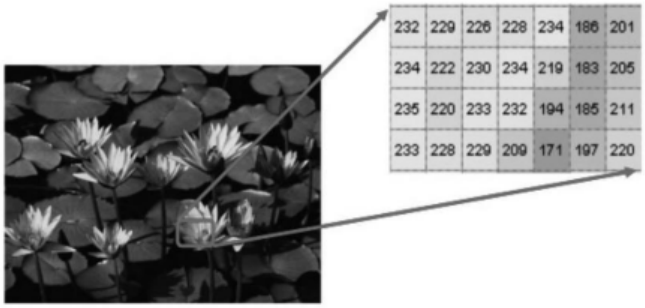


Gambar 2.12 Citra Biner

b. Citra *Grayscale*

Citra skala abu-abu adalah matriks data yang nilainya mewakili intensitas setiap piksel dalam kisaran 0255. Setiap piksel membutuhkan memori sekitar 8 bit. Gambar 2.13 menunjukkan

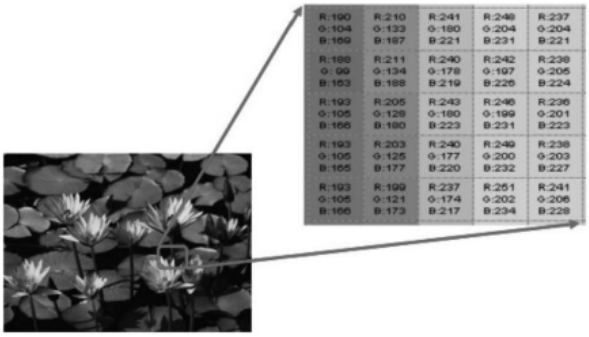
gambar skala abu-abu close-up dengan beberapa nilai intensitas piksel.



Gambar 2.13 Citra Grayscale

c. Citra Warna

"Gambar Berwarna" adalah gambar yang setiap pikselnya memiliki tiga elemen: merah (Merah), hijau (Hijau), dan biru (Biru). Warna setiap piksel ditentukan dengan menggabungkan intensitas warna merah, hijau, dan biru yang disimpan dalam bidang warna di lokasi piksel. Format file grafik yang menyimpan gambar berwarna sebagai gambar 24-bit. Setiap gambar berisi memori 8-bit. Dengan demikian, gambar berwarna memiliki 24 juta kemungkinan warna. Gambar 2.15 menunjukkan citra warna dilihat dari dekat dengan beberapa nilai intensitas piksel.



Gambar 2.14 Citra RGB

C. Pengertian JPG dan PNG

a. JPG

.jpg adalah format yang sangat umum saat ini, terutama untuk transisi gambar. Format ini digunakan untuk menyimpan gambar terkompresi menggunakan metode JPEG.

b. PNG (*Portable Network Graphics*)

Format .png adalah format yang sangat umum saat ini, terutama untuk transisi gambar. Format ini digunakan untuk menyimpan gambar terkompresi. Format ini dapat digunakan untuk gambar skala abu-abu, gambar palet warna, dan gambar penuh warna. Format ini memiliki penyimpanan 1 hingga 16 bit per piksel dan juga dapat menyimpan informasi hingga alfa piksel.

2.7. Python

Python adalah bahasa pemrograman yang ringan dan kuat yang memberikan kekuatan dan kompleksitas bahasa kompilasi tradisional, serta skrip yang mudah dan bahasa yang mudah diinterpretasikan. Anda akan kagum dengan seberapa cepat Anda dapat mempelajari bahasa ini dan apa yang dapat Anda lakukan dengan Python. Tak perlu dikatakan, satu-satunya batasan saat menggunakan Python adalah imajinasi Anda.

A. *Pycrypto*

Pycrypto adalah toolkit *Python* dengan pilihan mplementasi dari algortima kriptografi termasuk generator nomor acak. ni sangat menghemat waktu dan meningkatkan keamanan. Terdapat banyak subpaket dari *Pycrypto*, yaitu *Crypto.Cipher*, *Crypto.Hash*, *Crypto.Protocol*, *Crypto.PublicKey*, *Crypto.Signature*, *Crypto.Uti*