



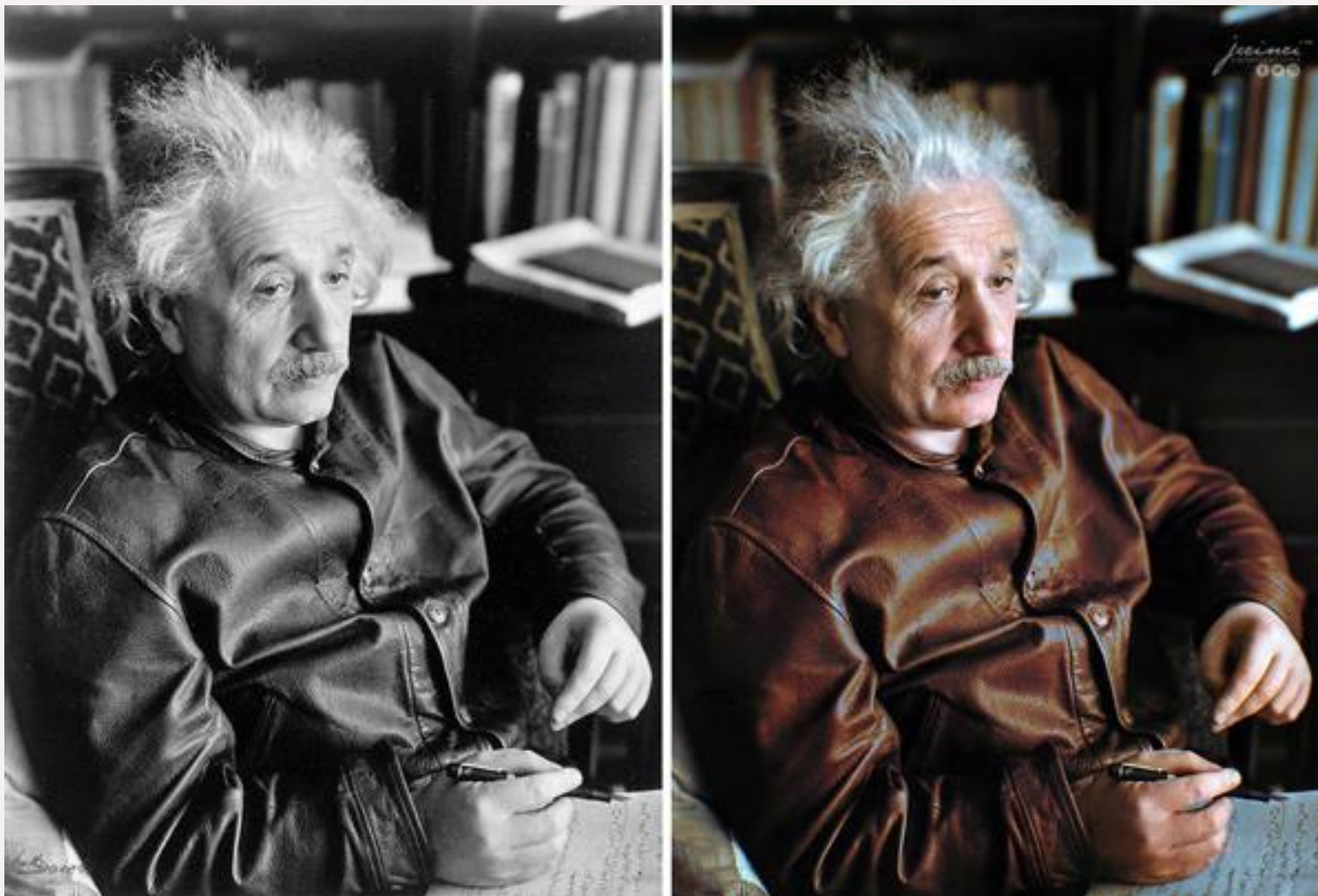
# Pengolahan Citra

## Pertemuan ke 3



[https://www.researchgate.net/publication/312511111/Pengolahan-Citra-Digital/](https://www.researchgate.net/publication/312511111/Pengolahan-Citra-Digital)

**APAKAH ANDA  
KENAL FOTO  
INI ?**

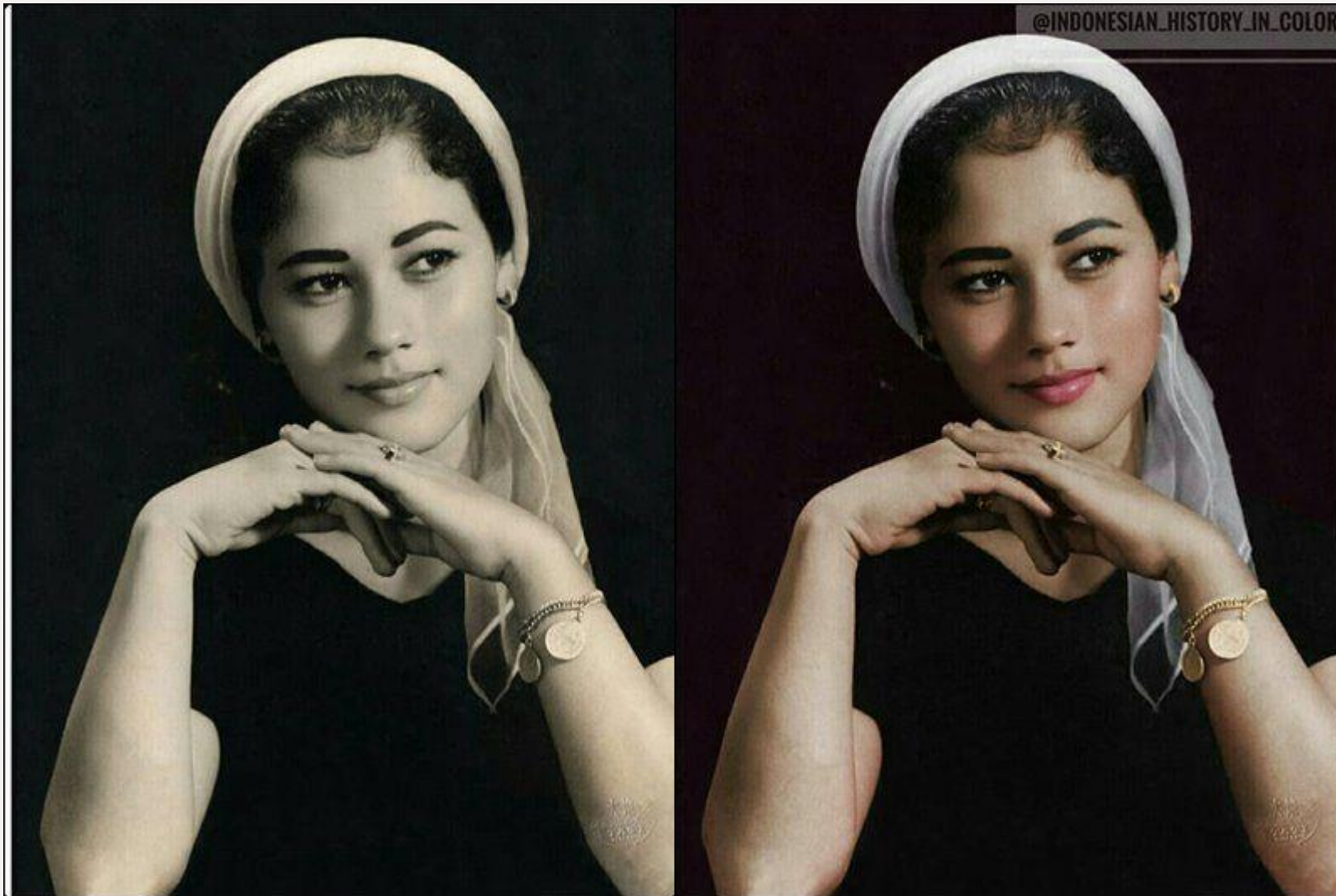


***Gambar 3.1 Albert Einstein, Dengan Jaket Kulit "Menlo Cossack" Tahun 1938***



**Gambar 3.2** Potret jadul siapa? Ini adalah Aminah  
Cendrakasih alias Mak Nyak semasa muda





***Gambar 3.3*** Potret Suzzanna semasa muda yang telah diedit berwarna

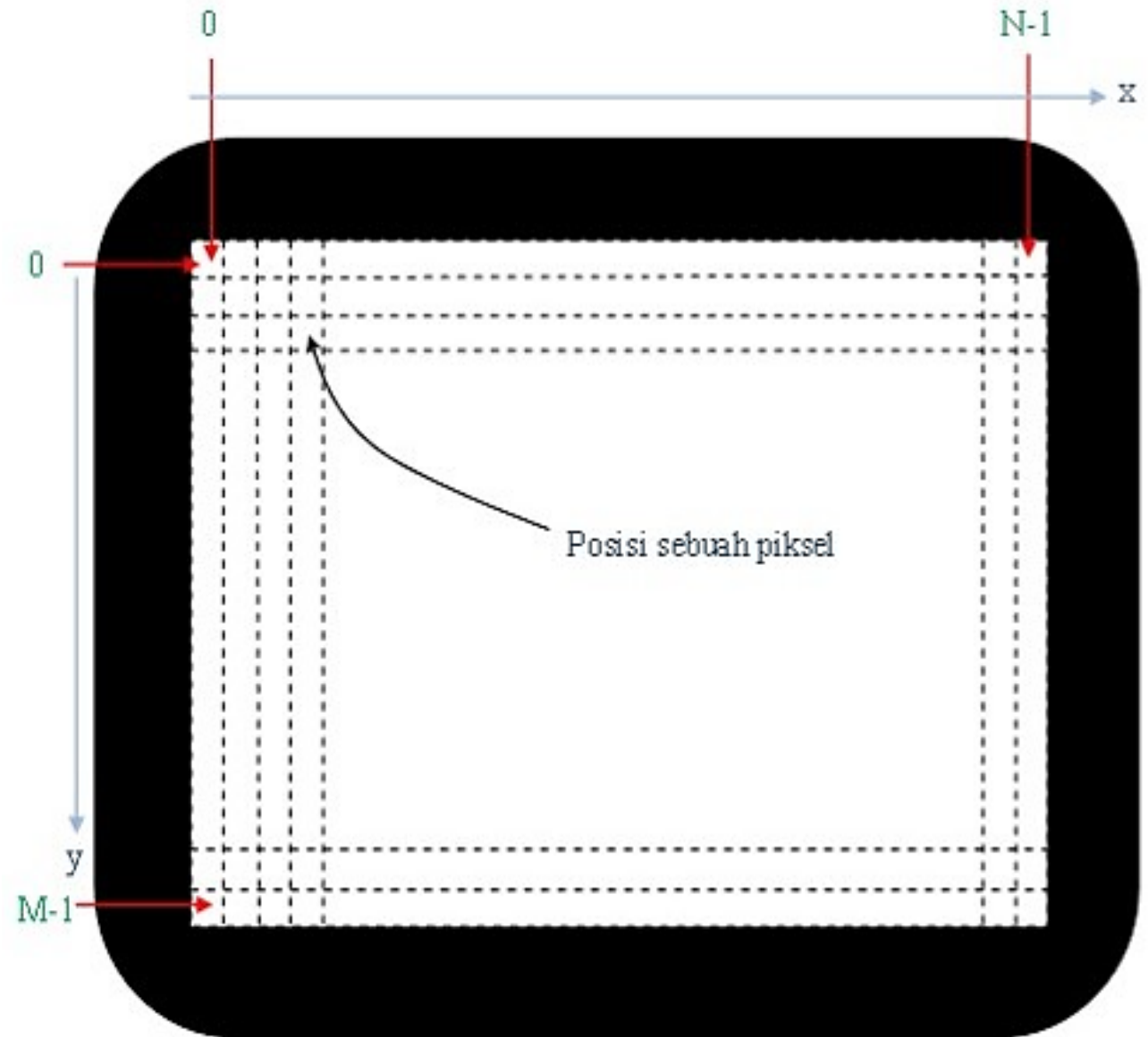
# Representasi Citra Digital

Citra digital dibentuk oleh kumpulan titik yang dinamakan piksel (*pixel* atau "*picture element*"). Setiap piksel digambarkan sebagai satu kotak kecil. Setiap piksel mempunyai koordinat posisi.

Dengan sistem koordinat yang mengikuti asas pemindaian pada layar TV standar, sebuah piksel mempunyai koordinat berupa  $(x, y)$ . Dalam hal ini;

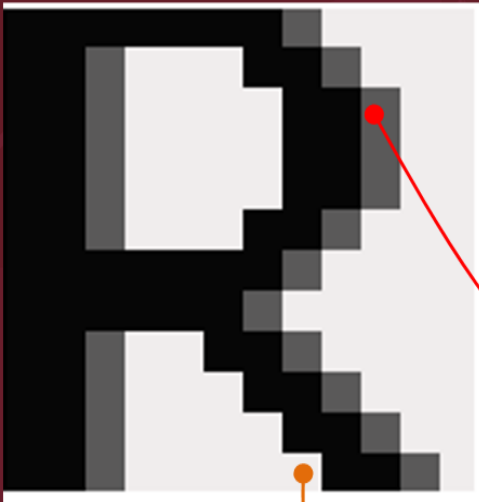
- $x$  menyatakan posisi kolom
- $y$  menyatakan posisi baris

Piksel pojok kiri-atas mempunyai koordinat  $(0, 0)$  dan piksel pada pojok kanan-bawah mempunyai koordinat  $(N-1, M-1)$ . Seperti ditunjukkan pada Gambar 3.4.



**Catatan :** untuk octav dan matlab tidak mempunyai koordinat  $(0,0)$  melainkan  $(1,1)$

Gambar 3.4 Sistem koordinat citra berukuran  $M \times N$  ( $M$  baris dan  $N$  kolom)

**R**

6	6	6	6	6	6	6	89	237	237	237	237
6	6	89	237	237	237	6	6	89	237	237	237
6	6	89	237	237	237	237	6	6	89	237	237
6	6	89	237	237	237	237	6	6	89	237	237
6	6	89	237	237	237	237	6	6	89	237	237
6	6	89	237	237	237	6	6	89	237	237	237
6	6	6	6	6	6	6	89	237	237	237	237
6	6	6	6	6	6	89	237	237	237	237	237
6	6	89	237	237	6	6	89	237	237	237	237
6	6	89	237	237	237	6	6	89	237	237	237
6	6	89	237	237	237	6	6	89	237	237	237
6	6	89	237	237	237	237	6	6	89	237	237

(a) Citra berukuran 12 x 12

(b) Data penyusun citra 12 x 12

Gambar 3.5 Citra dan nilai penyusun pixel

Didalam penggunaan program octave dan matlab, citra dinyatakan dengan  $f(y, x)$ . Dalam hal ini:

- x menyatakan posisi kolom
- y menyatakan posisi baris

Sebagai contoh, citra yang berukuran  $12 \times 12$  yang terdapat pada Gambar 3.5a memiliki susunan data seperti terlihat pada Gambar 3.5b.

$$f(2,1) = 6$$

	1	2	3	4	5	6	7	8	9	10	11	12
1	6	6	6	6	6	6	6	89	237	237	237	237
2	6	6	89	237	237	237	6	6	89	237	237	237
3	6	6	89	237	237	237	237	6	6	89	237	237
4	6	6	89	237	237	237	237	6	6	89	237	237
5	6	6	89	237	237	237	237	6	6	89	237	237
6	6	6	89	237	237	237	6	6	89	237	237	237
7	6	6	6	6	6	6	6	89	237	237	237	237
8	6	6	6	6	6	6	89	237	237	237	237	237
9	6	6	89	237	237	6	6	89	237	237	237	237
10	6	6	89	237	237	237	6	6	89	237	237	237
11	6	6	89	237	237	237	237	6	6	89	237	237
12	6	6	89	237	237	237	237	237	6	6	89	237

$$f(4,7) = 237$$

Gambar 3.6 menunjukkan contoh penotasian  $f(y,x)$ . Berdasarkan gambar tersebut maka:

- $f(2,1)$  bernilai 6
- $f(4,7)$  bernilai 237

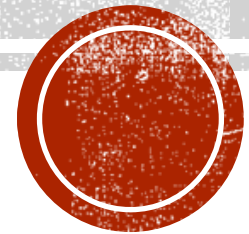
Pada citra berskala keabuan, nilai seperti 6 atau 237 dinamakan sebagai **intensitas**.

Gambar 3.6 Notasi pixel dalam citra

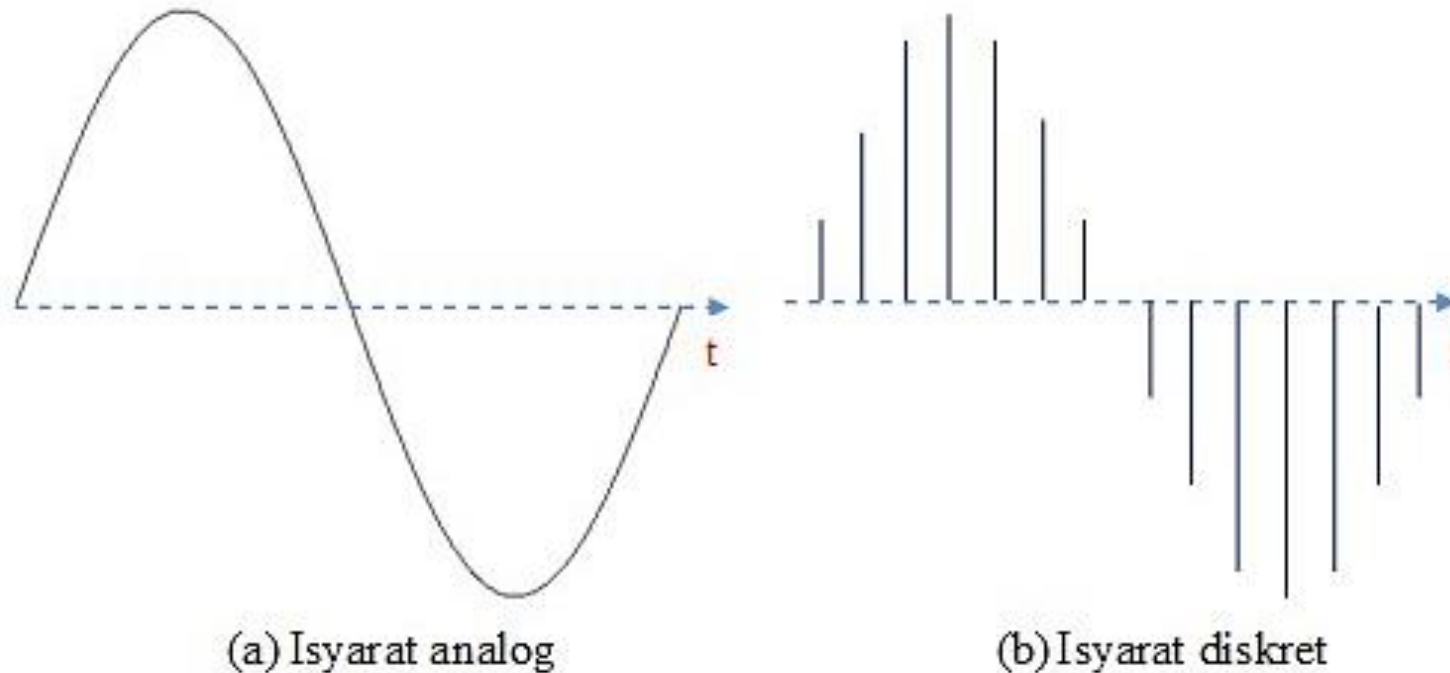


# KUANTISASI CITRA

**Kuantisasi** adalah prosedur yang dipakai untuk membuat suatu isyarat, yang bersifat kontinu kedalam bentuk diskrit. Seperti ditunjukkan pada gambar 3.4.



# ISYARAT ANALOG DENGAN DISKRIT



Gambar 3.7 Perbandingan isyarat analog dengan isyarat diskrit

Pada isyarat digital, nilai intensitas citra dibuat diskret atau terkuantisasi dalam sejumlah nilai bulat.

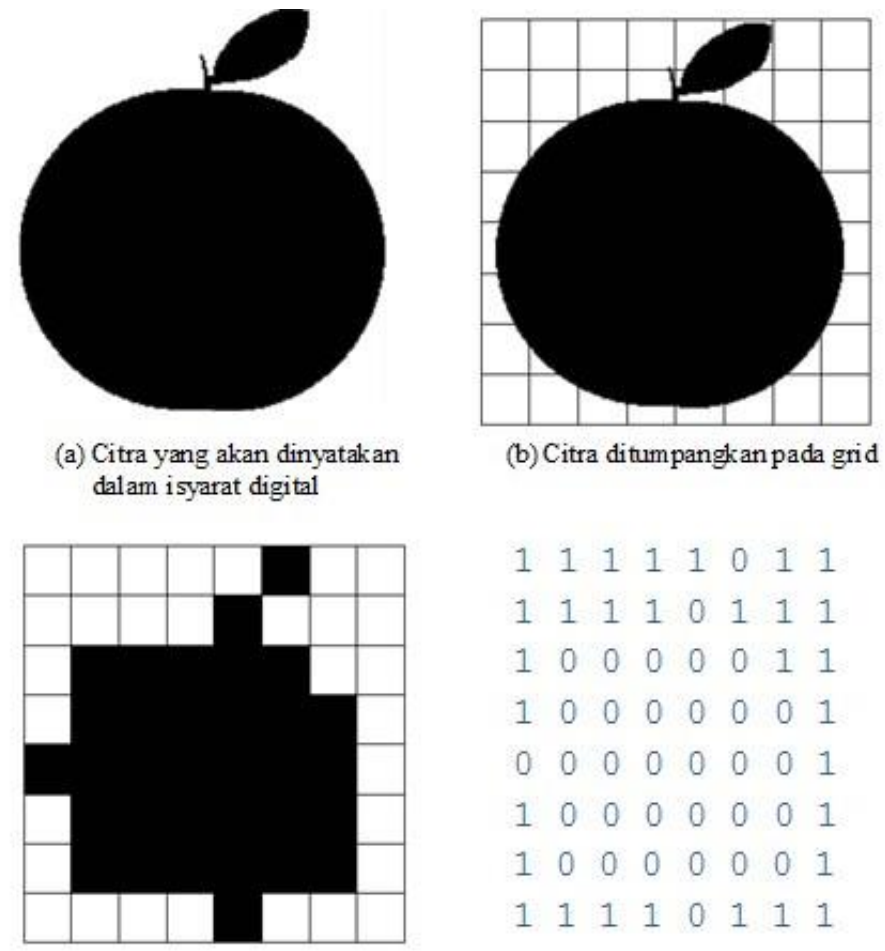




Gambar 3.8a menunjukkan contoh citra biner dua nilai intensitas berupa 0 (hitam) dan 1 (putih).

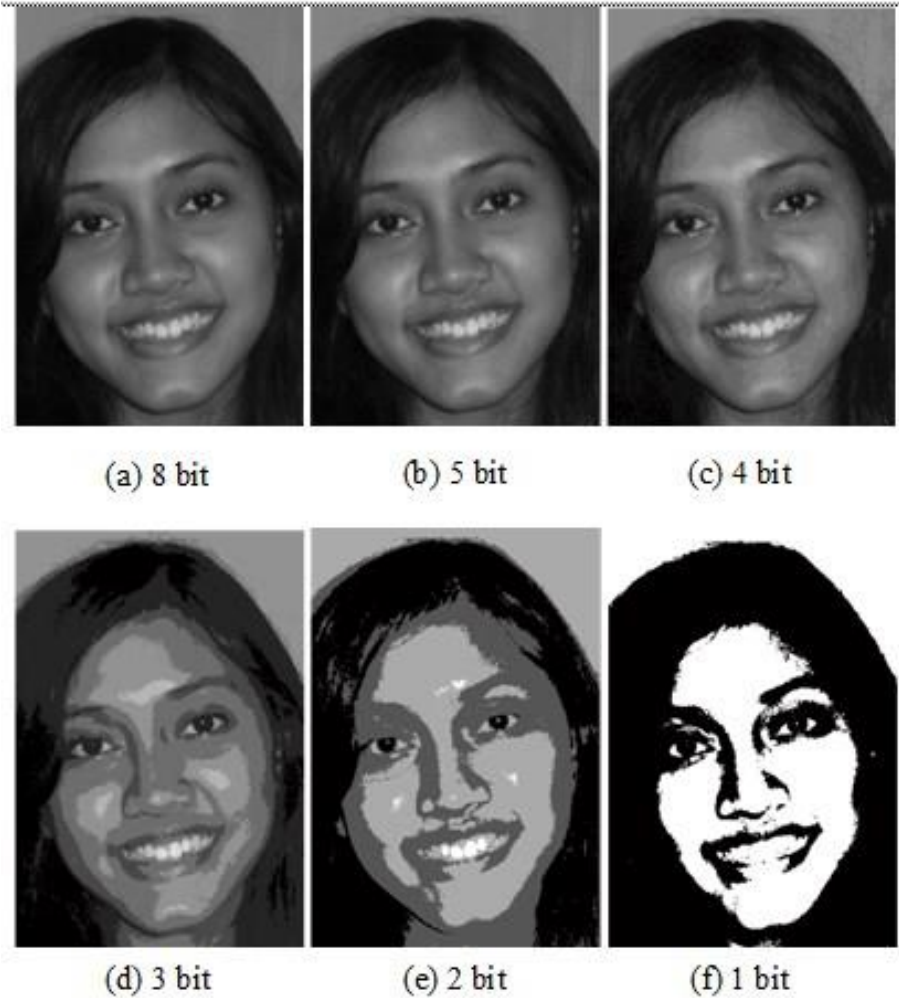
Gambar tersebut ditumpangkan pada grid 8x8 seperti yang diperlihatkan pada Gambar 3.8b.

Bagian gambar yang jatuh pada kotak kecil dengan warna hitam lebih sedikit dibanding latar belakang berwarna putih, maka seluruh isi kotak dibuat putih. Sebaliknya, jika isi kotak mayoritas hitam, maka seluruhnya kotak dibuat hitam.



Gambar 3.8 Digitalisasi citra biner 8 x 8 pixel

Hasil pengubahan ke citra digital seperti pada Gambar 3.8c. Adapun Gambar 3.8d memperlihatkan bilangan yang mewakili warna hitam (0) dan putih (1). Dengan demikian, citra digital lebih baik (lebih sesuai aslinya), jika jumlah piksel diperbanyak.



Gambar 3.9 Digitalisasi citra biner 8 x 8 pixel

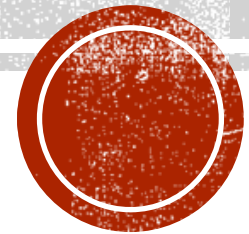
Dalam pengolahan citra, kuantisasi aras intensitas menentukan kecermatan hasilnya. Contoh Gambar 3.9 menunjukan citra yang dikuantisasi menggunakan 8, 5, 4, 3, 2, 1 bit. Kuantisasi dengan 1 bit, warna yang muncul berupa hitam dan putih saja. Penurunan jumlah aras pada tingkat tertentu membuat mata manusia masih bisa menerima citra dengan baik. Sebagai contoh, citra dengan 4 bit (Gambar 3.9c) dan citra dengan 8 bit (Gambar 3.9a) praktis terlihat sama. Hal seperti itulah yang mendasari gagasan **pemampatan data citra**, mengingat citra dengan jumlah bit lebih rendah tentu akan membutuhkan tempat dan transmisi yang lebih hemat.

\* Kuantisasi satu bit, merupakan jumlah level sebanyak 2 ( $2^1$ ), sehingga hanya terdiri dari 0 dan 1



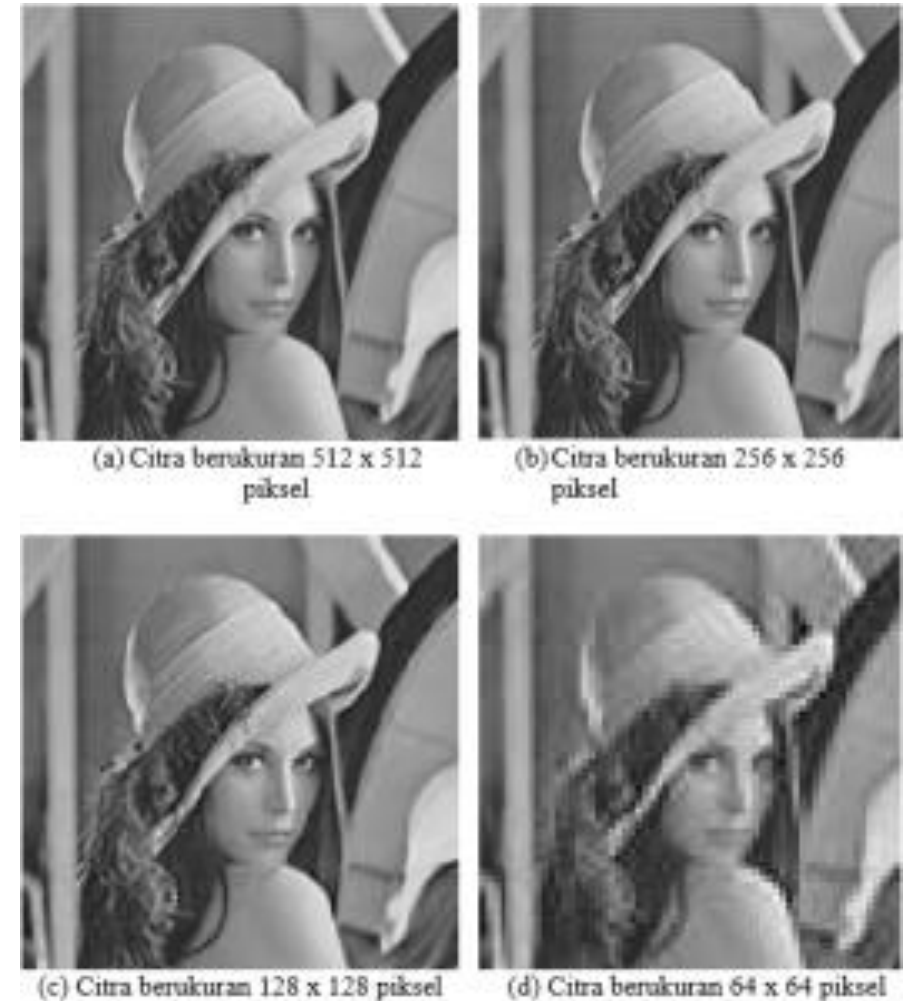
# KUALITAS CITRA

Di samping cacah intensitas kecerahan, jumlah piksel yang digunakan untuk menyusun suatu citra mempengaruhi kualitas citra.



# KUALITAS CITRA

Istilah resolusi citra biasa dinyatakan dengan jumlah piksel. Resolusi piksel biasa dinyatakan dengan notasi  $m \times n$ , dengan  $m$  menyatakan tinggi dan  $n$  menyatakan lebar dalam jumlah piksel. Gambar 3.10 menunjukkan efek resolusi berdasar jumlah pixel.



Gambar 3.10 Efek resolusi berdasar jumlah piksel pada citra

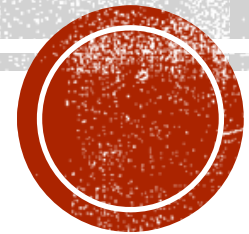




**Resolusi spasial** ditentukan oleh jumlah piksel per satuan panjang. Istilah seperti **dpi** (*dot per inch*) menyatakan jumlah piksel per inci. Misalnya citra 300 dpi, menyatakan bahwa citra akan dicetak dengan jumlah piksel sebanyak 300 piksel sepanjang satu inci.

# MENGENAL JENIS CITRA

Ada tiga jenis citra yang umum digunakan dalam pemrosesan citra. Ketiga jenis citra tersebut yaitu : **Citra berwarna, Citra berskala keabuan, dan Citra biner.**





# CITRA BERWARNA

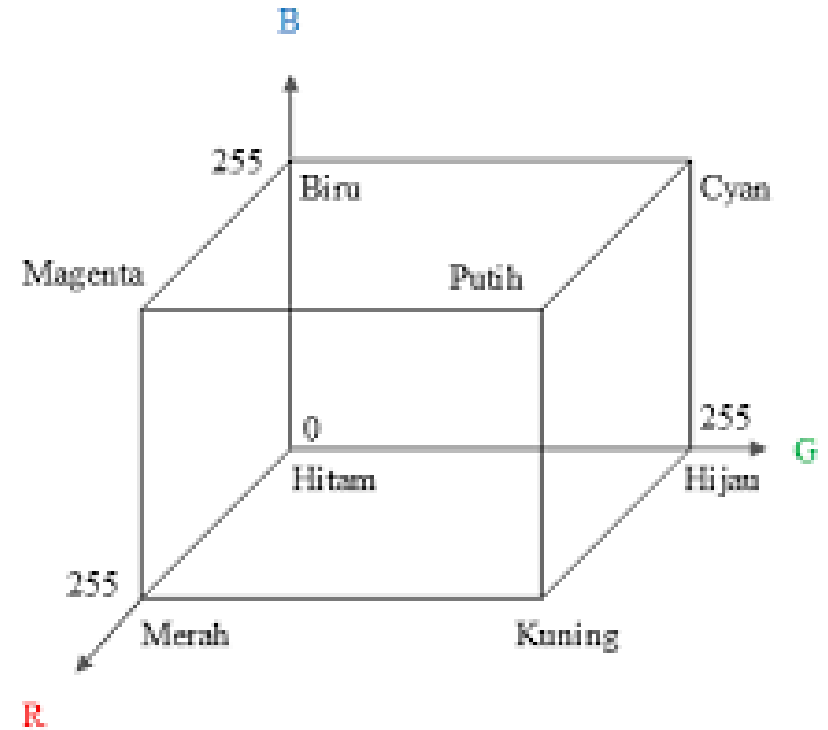
Citra berwarna atau biasa dinamakan citra RGB. Merupakan jenis citra yang menyajikan warna dalam bentuk R (Merah), G (Hijau), B (Biru).

Setiap komponen warna memiliki nilai berkisar antara 0 sampai 255. Tabel 3.1 Menunjukkan contoh warna dan nilai R, G, dan B.



Tabel 3.1 Warna dan nilai penyusun warna

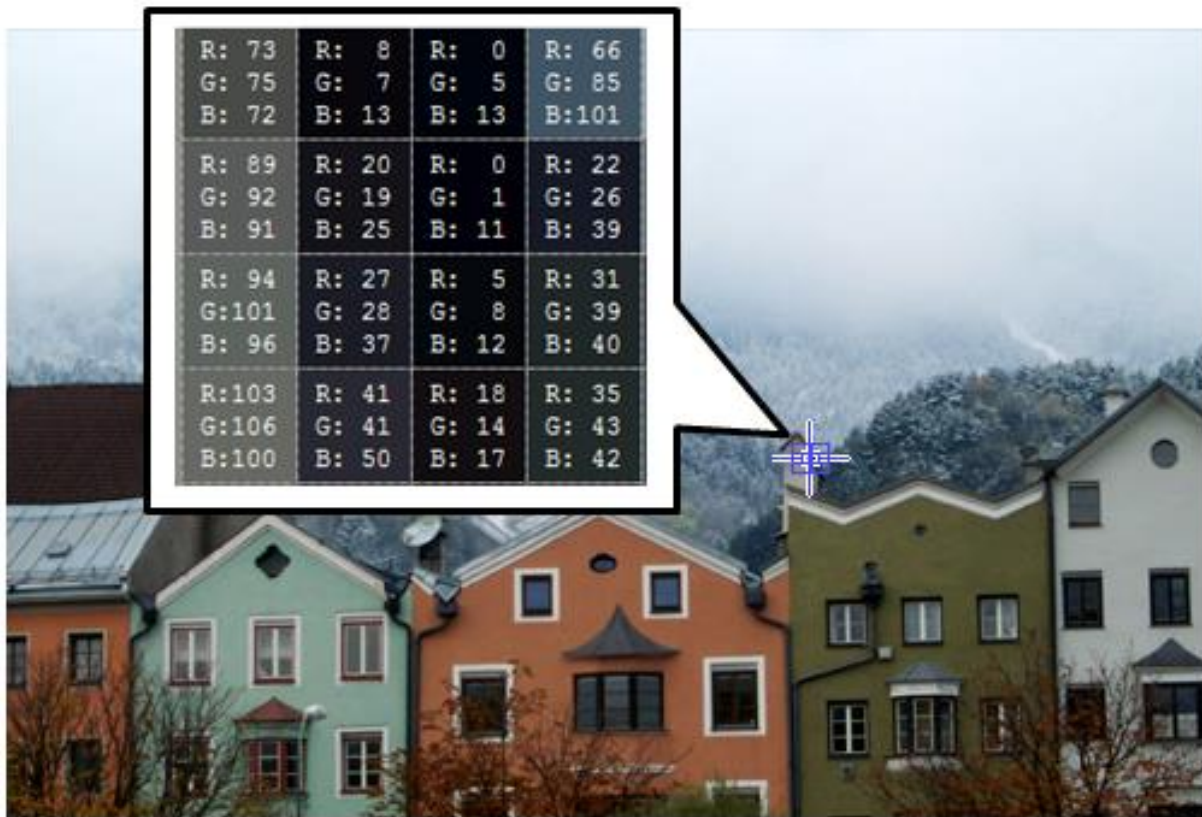
Warna	R	G	B
Merah	255	0	0
Hijau	0	255	0
Biru	0	0	255
Hitam	0	0	0
Putih	255	255	255
Kuning	0	255	255



Gambar 3.12 Warna RGB dalam ruang berdimensi 3

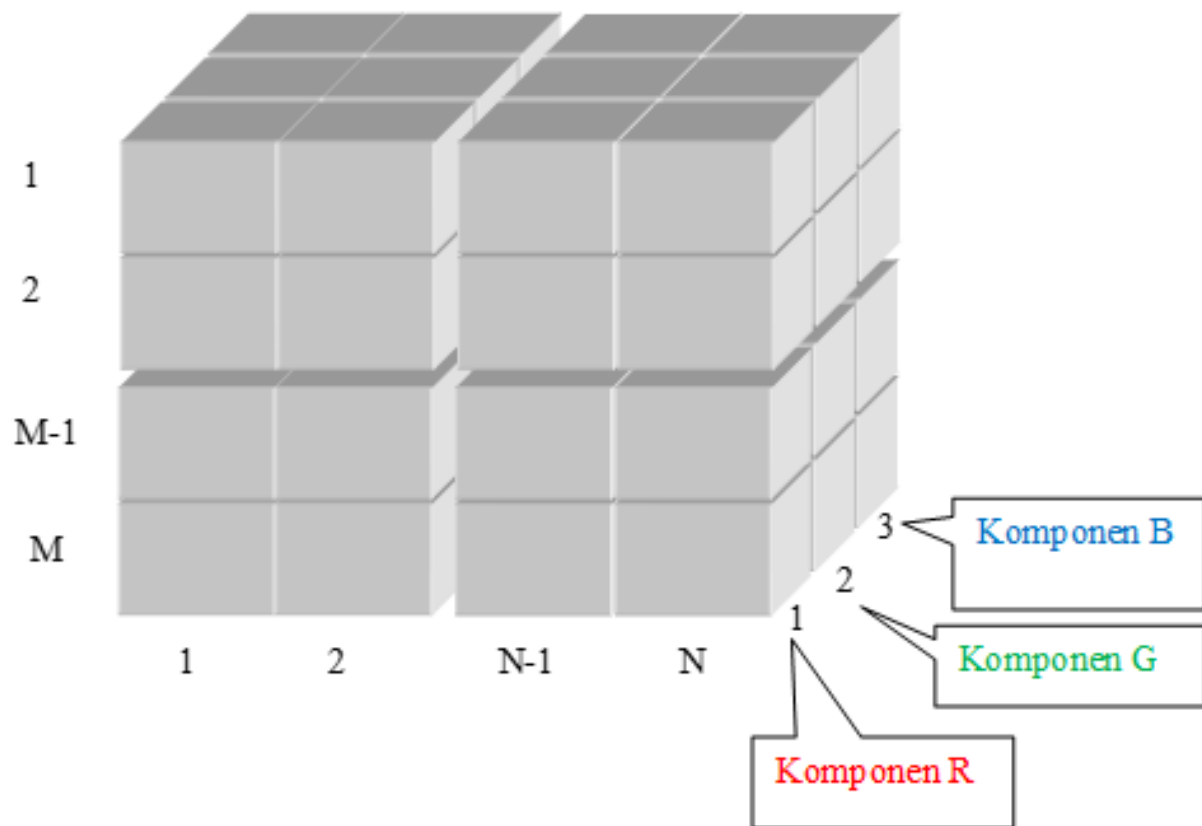
Dengan demikian warna yang dapat disajikan pada citra berwarna adalah :

255 x 255 x 255 yaitu mencapai 16.581.375



Gambar 3.13 Citra berwarna dan merepresai warnanya, setiap piksel dinyatakan dengan nilai R, B, G

Sebuah warna tidak hanya dinyatakan dengan komposisi R, G, dan B tunggal. Pada Tabel 3.1 terlihat bahwa warna merah mempunyai  $R = 255$ ,  $G = 0$ ,  $B = 0$ . Namun komposisi  $R = 254$ ,  $G = 1$ ,  $B = 1$ , juga berwarna merah



Gambar 3.14 Hasil pembacaan citra berwarna

Pada dimensi tiga menyatakan komponen R, G, B. Indeks pertama menyatakan komponen R, indeks kedua menyatakan komponen G, dan indeks ketiga menyatakan komponen B. Berikut adalah cara untuk mendapatkan komponen R, G, dan B pada larik Gambar 3.14.





Sesuai dengan nama yang melekat, citra jenis ini menangani gradasi warna hitam dan putih, yang tentu saja menghasilkan efek warna abu-abu. Pada jenis gambar ini, warna dinyatakan dengan intensitas. Dalam hal ini, intensitas berkisar antara 0 sampai dengan 255. Nilai 0 menyatakan hitam dan nilai 255 menyatakan putih.

# CITRA BERSKALA KEABUAN

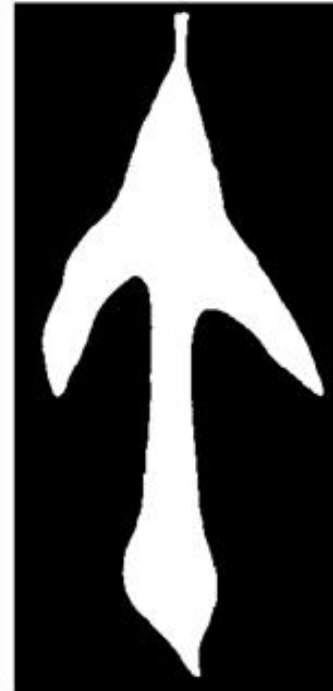
# CITRA BINER

Citra biner adalah citra dengan setiap piksel hanya dinyatakan dengan sebuah nilai dari dua buah kemungkinan (yaitu nilai 0 dan 1). Nilai 0 menyatakan warna hitam dan nilai 1 menyatakan warna putih.

Citra jenis ini banyak dipakai dalam pemrosesan citra, misalnya untuk kepentingan memperoleh tepi bentuk suatu objek. Sebagai contoh, perhatikan Gambar 3.15.



(a) Citra daun berskala keabuan



(b) Citra biner

Gambar 3.15 Citra kanan menyatakan bentuk dari citra kiri dengan mengabaikan komposisi warna







**APAKAH ADA YANG  
DITANYAKAN ?**





*Thank  
You*