

04 - Format Citra

IF4073 Pemrosesan Citra Digital

Oleh: Dr. Ir. Rinaldi, M.T



Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2025

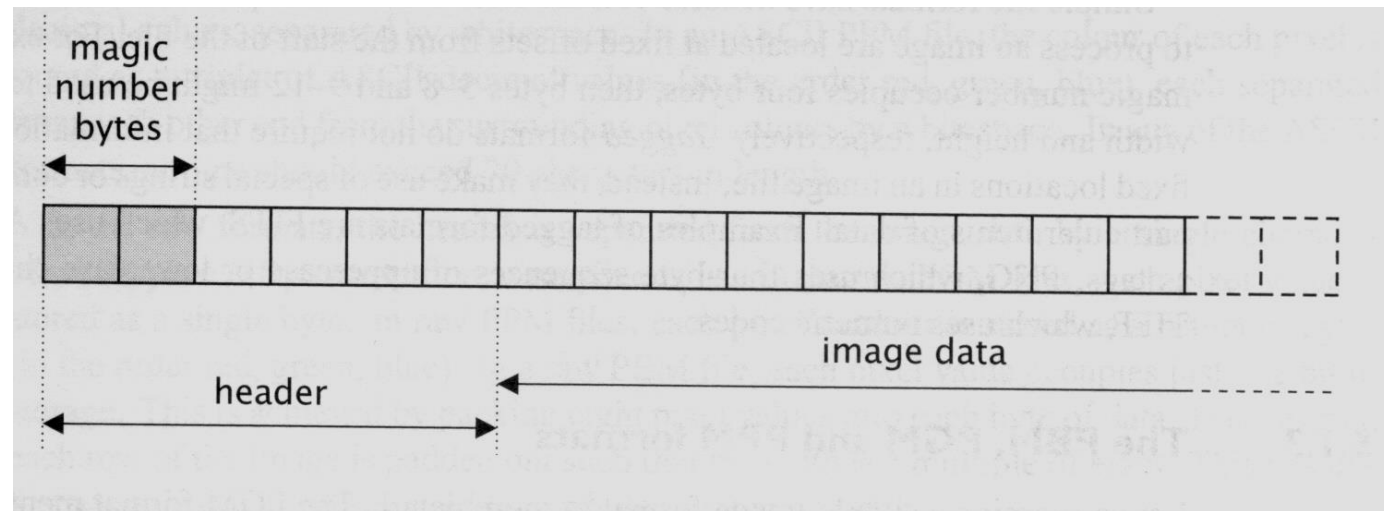
Format File Citra

Beberapa format file citra:

- PGM (Portable Gray Map), PBM (Portable Bitmap), PPM (Portable Pixelmap)
- BMP (Bitmap)
- GIF (Graphic Interchange Format) -
- PNG (Portable Network Graphics)
- JPEG (Joint Photographic Experts Group)
- TIFF (Tagged Image File Format)
- FITS (Flexible Image Transport System)
- Raw image
- dll

Pada kuliah ini hanya dibahas beberapa format citra yang penting saja

- File citra selalu diawali dengan *header* yang berisi informasi tentang data citra dan cara membaca data citra
- Kebanyakan *header* diawali dengan **signature** atau “magic number” (yaitu deretan *byte* yang mengidentifikasi format file)



PBM/PGM/PPM format

- PGM = *Portable Graymap*
- PGM merupakan format yang populer untuk citra *grayscale* (8 bits/pixel).
- Masih satu keluarga dengan:
 - PBM (*Portable Bitmap*), untuk citra biner (1 bit/pixel)
 - PPM (*Portable Pixelmap*), untuk citra berwarna (24 bits/pixel)

Type	Magic number		Extension	Colors
	ASCII	Binary		
Portable BitMap ^[1]	P1	P4	.pbm	0–1 (white & black)
Portable GrayMap ^[2]	P2	P5	.pgm	0–255 (gray scale)
Portable PixMap ^[3]	P3	P6	.ppm	0–255 (RGB)

Sumber: wikipedia

- Contoh format PBM

Ukuran citra



```
P1
# This is an example bitmap of the letter "J"
6 10
0 0 0 0 1 0
0 0 0 0 1 0
0 0 0 0 1 0
0 0 0 0 1 0
0 0 0 0 1 0
0 0 0 0 1 0
0 0 0 0 1 0
1 0 0 0 1 0
0 1 1 1 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```

Note: there is a newline character at the end of each line

Sumber: wikipedia



Citra yang terbentuk

- Contoh format PGM

Ukuran citra

Jumlah level
keabuan

```
P2
# Shows the word "FEEP" (example from Netpbm man page on PGM)
24 7
15
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 3 3 3 3 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 15 0
0 3 3 3 0 0 0 7 7 7 0 0 0 11 11 11 0 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 0 0
0 3 0 0 0 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```



Citra yang terbentuk

```
>> A = imread('balloons.ascii.pgm');  
>> imshow(A)  
>> size(A)
```

```
ans =  
  
    480    640
```



```

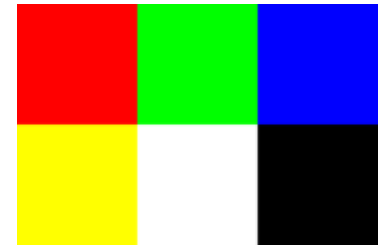
P2
# created by 'xv balloons_bw.tif'
640 480
255
232 227 220 216 212 209 207 206 205 205 205 207 208 209 210 211 212 211 211
213 212 211 210 209 210 210 211 212 211 210 210 210 210 211 210 210 210 210
209 210 209 208 209 208 209 210 209 208 210 209 209 208 208 208 209 208 208
208 207 207 207 206 207 207 207 207 207 207 207 207 207 207 205 204 206 205
205 204 204 204 203 202 203 202 201 201 201 200 199 199 200 199 198 198 198
197 197 198 197 196 195 195 194 193 192 192 191 191 190 190 190 190 189 189
190 188 188 188 187 187 187 186 186 186 186 187 186 186 187 188 188 187 186
186 186 185 186 186 186 187 186 186 186 185 185 187 186 185 186 185 185 186
185 184 185 186 185 186 186 186 185 186 185 185 185 184 183 184 184 183 184
184 183 183 183 184 183 183 184 183 184 185 185 184 184 183 183 184 183 184
182 183 182 183 183 183 183 183 183 184 183 183 184 182 183 182 183 183 183
183 183 182 184 184 183 183 183 183 184 185 185 184 185 184 185 185 185 185
186 186 185 184 185 184 184 184 186 185 183 181 176 161 150 149 150 153 156
163 166 159 154 156 158 158 158 159 159 160 160 159 159 159 159 160 160 160
160 162 162 163 162 163 164 164 166 170 172 174 175 175 175 175 175 174 176
176 176 176 177

```

..... •

- Contoh format PPM untuk citra berwarna RGB

```
P3
3 2
255
# The part above is the header
# "P3" means this is a RGB color image in ASCII
# "3 2" is the width and height of the image in pixels
# "255" is the maximum value for each color
# The part below is image data: RGB triplets
255 0 0 0 255 0 0 0 255
255 255 0 255 255 255 0 0 0
```



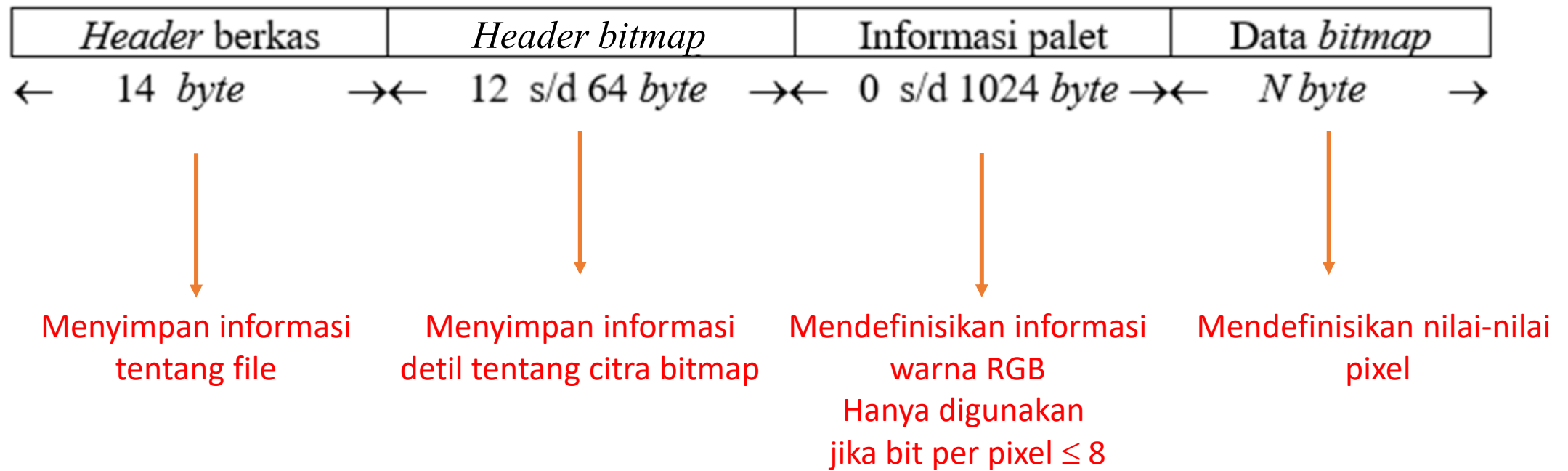
Citra yang terbentuk

Sumber: wikipedia

Format File BMP

- Dikenal juga dengan nama file citra bitmap atau *device independent bitmap (DIB) file format*.
- Merupakan format citra yang baku untuk sistem operasi Windows dan OS/2
- Umumnya tidak dimampatkan (*uncompressed image*)
- Kekurangan: membutuhkan memori yang besar
- Kelebihan: kualitas citranya lebih bagus daripada citra terkompresi (karena tidak ada informasi yang hilang)
- Citra format BMP ada tiga macam: citra biner, citra berwarna, dan citra hitam-putih (*grayscale*).

Struktur file citra BMP:



Tabel 1. *Header berkas bitmap (panjang = 14 byte)*

Byte ke-	Panjang (byte)	Nama	Keterangan
1 – 2	2	BmpType	Signature: BA = <i>bitmap array</i> , CI = <i>icon</i> BM = <i>bitmap</i> , CP = <i>color pointer</i> PT = <i>pointer</i>
3 – 6	4	<i>BmpSize</i>	Ukuran berkas <i>bitmap</i>
7 – 8	2	<i>XhotSpot</i>	Reserved
9 – 10	2	<i>YhotSpot</i>	Reserved
11 – 14	4	<i>OffBits</i>	Offset ke awal data <i>bitmap</i> (dalam <i>byte</i>)

Tabel 2. *Header bitmap versi lama dari Microsoft Windows (12 byte)*

Byte ke-	Panjang (byte)	Nama	Keterangan
1 – 4	4	<i>HdrSize</i>	Ukuran <i>header</i> dalam satuan <i>byte</i>
5 – 6	2	<i>Width</i>	Lebar <i>bitmap</i> dalam satuan <i>pixel</i>
7 – 8	2	<i>Height</i>	Tinggi <i>bitmap</i> dalam satuan <i>pixel</i>
9 – 10	2	<i>Planes</i>	Jumlah <i>plane</i> (umum- nya selalu satu)
11 – 12	2	<i>BitCount</i>	Jumlah bit per <i>pixel</i>

Tabel 3. *Header bitmap versi baru dari Microsoft Windows (40 byte)*

Byte ke-	Panjang (byte)	Nama	Keterangan
1 – 4	4	<i>HdrSize</i>	Ukuran <i>header</i> dalam satuan <i>byte</i>
5 – 8	4	<i>Width</i>	Lebar <i>bitmap</i> dalam satuan <i>pixel</i>
9 – 12	4	<i>Height</i>	Tinggi <i>bitmap</i> dalam satuan <i>pixel</i>
13 – 14	2	<i>Planes</i>	Jumlah <i>plane</i> (umum- nya selalu satu)
15 – 16	2	<i>BitCount</i>	Jumlah bit per <i>pixel</i>
17 – 20	4	<i>Compression</i>	0=tak dimampatkan, 1=dimampatkan
21 – 24	4	<i>ImgSize</i>	Ukuran <i>bitmap</i> dalam <i>byte</i>
25 – 28	4	<i>HorzRes</i>	Resolusi horizontal
29 – 32	4	<i>VertRes</i>	Resolusi vertikal
33 – 36	4	<i>ClrUsed</i>	Jumlah warna yang digunakan
37 – 40	4	<i>ClrImportant</i>	Jumlah warna yang penting

Tabel 4. Header bitmap versi baru dari IBM OS/2 (64 byte)

Byte ke-	Panjang (byte)	Nama	Keterangan
1 – 4	4	HdrSize	Ukuran <i>header</i> dalam satuan <i>byte</i>
5 – 8	4	Width	Lebar <i>bitmap</i> dalam satuan <i>pixel</i>
9 – 12	4	Height	Tinggi <i>bitmap</i> dalam satuan <i>pixel</i>
13 – 14	2	Planes	Jumlah <i>plane</i> (umumnya selalu satu)
15 – 16	2	BitCount	Jumlah bit per <i>pixel</i>
17 – 20	4	Compression	0 = tak dimampatkan, 1 = dimampatkan
21 – 24	4	ImgSize	Ukuran <i>bitmap</i> dalam <i>byte</i>
25 – 28	4	HorzRes	Resolusi horizontal
29 – 32	4	VertRes	Resolusi vertikal
33 – 36	4	ClrUsed	Jumlah warna yang digunakan
37 – 40	4	ClrImportant	Jumlah warna yang penting
41 – 42	2	Units	Satuan pengukuran yang dipakai
43 – 44	2	Reserved	Field Cadangan
45 – 46	2	Recording	Algoritma perekaman
47 – 48	2	Rendering	Algoritma <i>halftoning</i>
49 – 52	4	Size1	Nilai ukuran 1
53 – 56	4	Size2	Nilai ukuran 2
57 – 60	4	ClrEncoding	Pengkodean warna
61 – 64	4	Identifier	Kode yang digunakan aplikasi

- Informasi palet warna terletak sesudah *header bitmap*. Panjangnya 0 sampai 1024 bit. Palet hanya tersedia untuk citra dengan 8 bit per pixel atau kurang.
- Informasi palet warna dinyatakan dalam suatu tabel *RGB*.
- Setiap *entri* pada tabel terdiri atas tiga buah *field*, yaitu *R* (*red*), *G* (*green*), dan *B* (*blue*). Palet diindeks dari 0 sampai $2^n - 1$, n = ukuran bit per pixel
- Nilai setiap *R*, *G*, dan *B* di dalam palet dinormalisasi dari 0 sampai 1
- Untuk citra *grayscale*, nilai $R = G = B$.



	R	G	B
0			
1			
2			
...			
$2^n - 1$			

- Penyimpanan data *bitmap* di dalam berkas adalah sebagai berikut:
 - dari kiri ke kanan
 - disusun terbalik dari bawah ke atas dalam bentuk matriks yang berukuran *Height* $\times$ *Width*.

Baris ke-0 pada matriks data *bitmap* menyatakan data *pixel* di citra baris terbawah, sedangkan baris terakhir pada matriks menyatakan data *pixel* di citra baris teratas.

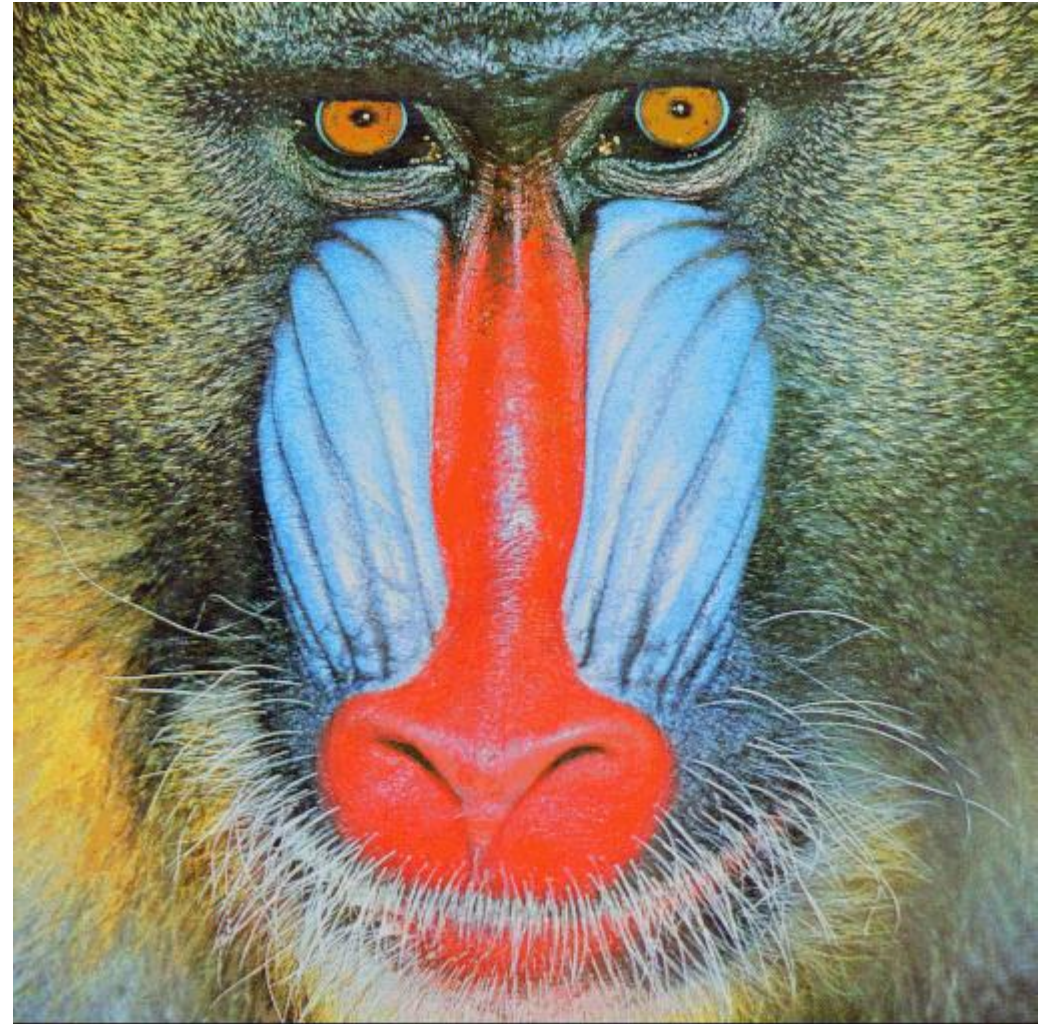
- Untuk citra berwarna dengan 8 bit/*pixel*, data *bitmap* menyatakan indeks ke palet warna. Jumlah kombinasi warna yang dihasilkan adalah $2^8 = 256$ warna.

<header berkas>			
<header bitmap>			
<palet warna RGB>			
	<i>R</i>	<i>G</i>	<i>B</i>
0	0.242	0.035	0.021
1	0.271	0.028	0.067
2	0.414	0.123	0.763
...			
255	0.543	0.234	0.045
<data bitmap>			
2 2 0 0 0 3 5 ...			

- Pada contoh di atas, warna *pixel* pertama dinyatakan oleh entri baris indeks ke-2 pada palet warna ($R = 0.414$, $G = 0.123$, $B = 0.763$).
- Warna setiap pixel ditentukan oleh kombinasi komponen R, G, dan B

Contoh citra berwarna dengan 8 bit/pixel

```
>> [A, map] = imread('baboon.bmp');  
>> size(map)  
  
ans =  
  
    256     3  
  
>> imshow(A, map)
```



- Tampilkan sebagian isi palet citra 'baboon.bmp')

```
>> map(1:15,1:3)
```

```
ans =
```

0.9137	0.4275	0.4314
0.8275	0.4353	0.4392
0.6941	0.4471	0.4510
0.3490	0.2784	0.2824
0.9216	0.3412	0.4196
0.9098	0.4784	0.5490
0.8314	0.3412	0.4275
0.6863	0.3373	0.4196
0.8275	0.5647	0.6314
0.8275	0.4549	0.5725
0.5608	0.3412	0.4196
0.7137	0.5804	0.6863
0.2353	0.1529	0.2196
0.1686	0.0902	0.1647
0.2941	0.2157	0.2902

- Nilai palet citra 'baboon.bmp' setelah dikonversi menjadi rentang nilai dari 0 sampai 255:

```
>> map2 = 255*map;  
>> map2(1:15, 1:3)
```

```
ans =
```

```
233 109 110  
211 111 112  
177 114 115  
89 71 72  
235 87 107  
232 122 140  
212 87 109  
175 86 107  
211 144 161  
211 116 146  
143 87 107  
182 148 175  
60 39 56  
43 23 42
```

- Tampilkan sebagian isi data bitmap citra 'baboon.bmp'

```
>> A(1:10, 1:10)
```

```
ans =
```

```
10×10 uint8 matrix
```

```
160 128 173 131 133 152 171 208 171 171
150 154 151 146 169 128 171 151 117 215
171 154 209 151 148 173 180 209 148 106
105 155 152 151 134 154 173 209 132 131
169 126 155 209 132 169 171 209 128 120
131 131 129 180 132 148 209 209 171 148
151 173 155 155 105 154 173 132 105 150
208 209 148 168 134 166 12 180 148 149
215 209 173 154 124 165 173 13 171 118
208 208 151 146 155 131 131 173 151 149
```

Contoh citra *grayscale* 8 bit/pixel

```
>> [A, map] = imread('cameraman.bmp');
```

```
>> size(map)
```

```
ans =
```

```
    256     3
```

```
>> imshow(A)
```



- Tampilkan sebagian isi palet citra 'cameraman.bmp'

```
>> map(1:15, 1:3)
```

```
ans =
```

0	0	0
0.0039	0.0039	0.0039
0.0078	0.0078	0.0078
0.0118	0.0118	0.0118
0.0157	0.0157	0.0157
0.0196	0.0196	0.0196
0.0235	0.0235	0.0235
0.0275	0.0275	0.0275
0.0314	0.0314	0.0314
0.0353	0.0353	0.0353
0.0392	0.0392	0.0392
0.0431	0.0431	0.0431
0.0471	0.0471	0.0471
0.0510	0.0510	0.0510
0.0549	0.0549	0.0549

- Nilai palet citra 'cameraman.bmp' setelah dikonversi menjadi rentang nilai 0 sampai 255:

```
>> map2 = 255*map;
```

```
>> map2(1:15, 1:3)
```

```
ans =
```

```
0  0  0
1  1  1
2  2  2
3  3  3
4  4  4
5  5  5
6  6  6
7  7  7
8  8  8
9  9  9
10 10 10
11 11 11
12 12 12
13 13 13
14 14 14
```

- Tampilkan sebagian isi data bitmap citra 'cameraman.bmp'

```
>> A(1:10, 1:10)
```

```
ans =
```

```
10×10 uint8 matrix
```

156	159	158	155	158	156	159	158	157	158
160	154	157	158	157	159	158	158	158	160
156	159	158	155	158	156	159	158	157	158
160	154	157	158	157	159	158	158	158	160
156	153	155	159	159	155	156	155	155	157
155	155	155	157	156	159	152	158	156	158
156	153	157	156	153	155	154	155	157	156
159	159	156	158	156	159	157	161	162	157
158	155	158	154	156	160	162	155	159	161
155	154	157	158	160	160	159	160	158	161

- Untuk citra *truecolor* (24-bit), tidak ada palet warna. Nilai *RGB* langsung diuraikan di dalam data *bitmap*.
- Setiap elemen data *bitmap* panjangnya 3 *byte*, masing-masing *byte* menyatakan komponen *R*, *G*, dan *B*.

```
<header berkas>
<header bitmap>

<data bitmap>
0.0341 0.4521 0.1264 0.645 0.3421 0.4529 ...
```

- Pada contoh di atas, *pixel* pertama mempunyai *R* = 0.0341, *G* = 0.4521, *B* = 0.1264, *pixel* kedua mempunyai *R* = 0.645, *G* = 0.3421, *B* = 0.4529. Demikian seterusnya.
- Citra *truecolor* disebut juga citra 16 juta warna, karena ia mampu menghasilkan $2^{24} = 16.777.216$ kombinasi warna.

Contoh citra *truecolor* 24 bit/pixel

```
>> [A, map] = imread('monarch.bmp');  
>> size(map)
```

```
ans =
```

```
      0      0
```

```
>> imshow(A)
```

- Tidak ada palet warna, ukuran 0 x 0



- Tampilkan sebagian isi data bitmap citra 'monarch.bmp'

```
>> size(A)
```

```
ans =
```

```
512 768 3
```

```
>> A(1:10, 1:10, 1) % komponen red (R)
```

```
ans =
```

```
10×10 uint8 matrix
```

107	108	109	111	114	118	121	123	124	125
121	122	122	123	123	123	123	123	126	126
121	122	125	126	127	128	127	126	127	127
117	118	121	123	125	127	128	128	126	126
125	124	123	123	123	124	125	125	124	124
126	125	124	123	122	123	124	125	124	124
120	120	121	122	123	124	125	126	126	127
120	121	123	124	125	124	124	123	129	129
121	122	124	125	126	127	127	126	127	128
120	121	123	125	127	127	128	128	128	129

```
>> A(1:10, 1:10, 2) % komponen green (G)
```

```
ans =
```

```
10×10 uint8 matrix
```

87	88	89	91	94	98	101	103	104	105
101	102	102	103	103	103	103	103	106	106
101	102	105	106	107	108	107	106	107	107
97	98	101	103	105	107	108	108	106	107
105	104	103	103	103	105	106	106	105	105
106	105	104	104	103	104	105	106	105	105
101	101	102	103	104	105	106	107	107	108
101	102	104	105	106	105	105	104	110	110
102	103	105	106	107	108	108	107	108	109
101	102	104	106	108	108	109	109	109	110

```
>> A(1:10, 1:10, 3) % komponen blue (B)
```

```
ans =
```

```
10×10 uint8 matrix
```

37	38	38	40	43	47	50	52	53	54
51	52	51	52	52	52	52	52	55	53
50	51	54	55	54	55	54	53	54	54
46	47	50	50	52	54	55	55	53	51
52	51	50	50	50	49	50	50	49	49
53	52	51	48	47	48	49	50	49	47
45	45	46	47	48	47	48	49	49	50
45	46	48	49	48	47	47	46	52	52
44	45	47	48	49	50	50	49	49	50
43	44	46	48	50	50	51	51	50	51

GIF

- Diperkenalkan oleh CompuServe tahun 1987
- GIF adalah citra terkompresi, menggunakan algoritma kompresi Lempel-Ziv-Welch (LZW).
- Citra GIF terdiri dari sebuah palet warna (*colormap*) RGB dan sebuah matriks yang nilai elemennya menyatakan indeks ke sebuah baris di dalam palet.
- Palet adalah matriks $m \times 3$ bertipe *floating-point* dengan nilai di dalam $[0, 1]$. Untuk citra dengan 256 warna, maka *map* berukuran 256×3 .
- Tiap baris di dalam palet menspesifikasikan komponen *red*, *green*, dan *blue* dari sebuah warna tunggal.
- GIF cocok untuk menyimpan grafik yang hanya memiliki beberapa warna seperti diagram, logo, dan kartun. Ada juga *animated GIF* yang disusun oleh beberapa *frame*.

JPEG

- JPEG adalah format citra terkompresi dengan metode kompresi yang *lossy*, yaitu DCT (*discrete cosine transform*).
- Ukuran file JPEG lebih kecil daripada file BMP tetapi dengan kompensasi penurunan kualitas secara visual.
- Ekstensi file: JPG atau JPEG
- Format citra umum untuk kamera digital
- Mendukung citra *grayscale* 8-bit dan citra berwarna 24-bit
- Varian dari JPEG adalah JPEG 2000 yang menggabungkan *lossless* dan *lossy compression*.
- JPEG akan dibahas dalam materi tersendiri (Pemampatan Citra)

PNG

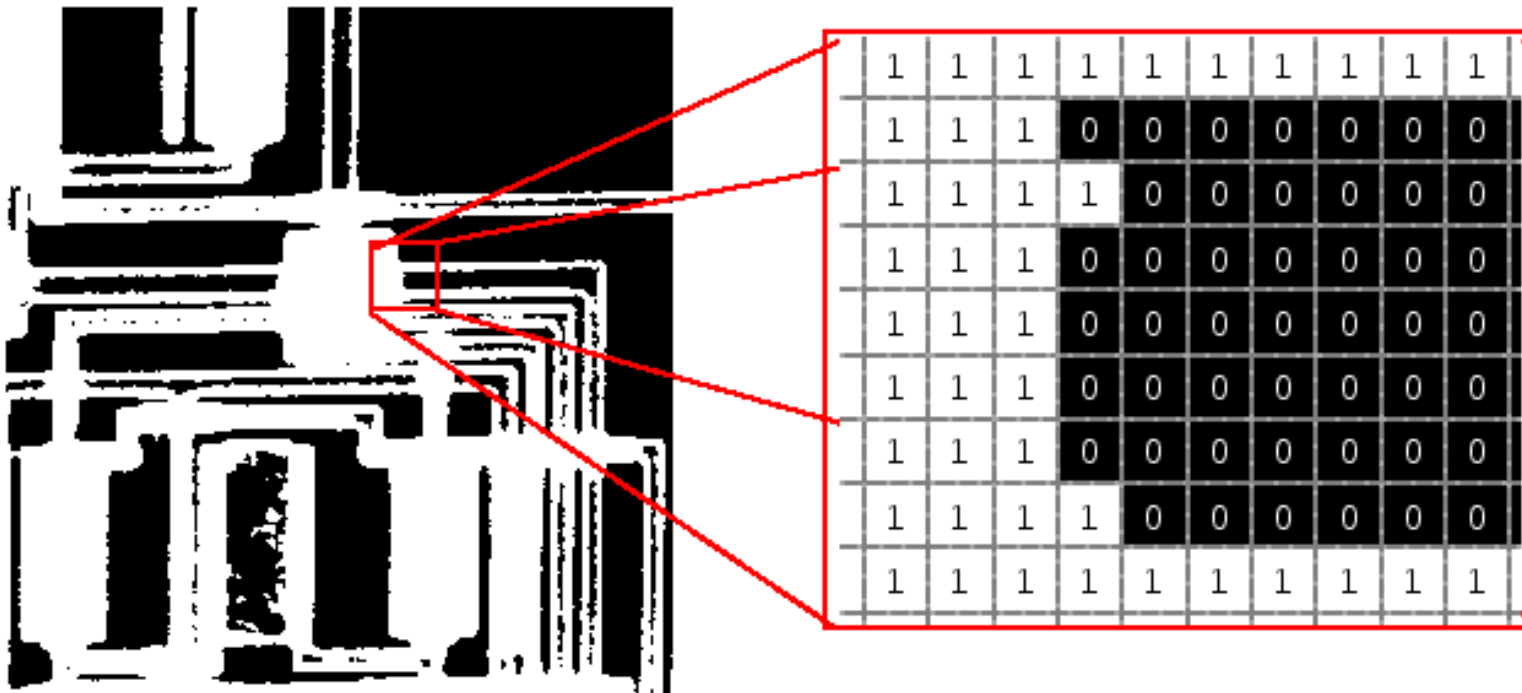
- PNG = Portable Network Graphics
- Merupakan format citra yang *free, open-source*, dan universal sehingga dapat digunakan secara luas di internet.
- Dibuat sebagai alternatif format GIF
- *Lossless compression*.
- Mendukung citra terindeks dengan palet 8-bit, 24-bit truecolor, dan 48-bit *truecolor*.

Jenis citra di dalam Matlab

1. Citra biner (*binary image*)
2. Citra terindeks (*indexed image*)
3. Citra hitam-putih (*grayscale image*)
4. Citra warna-sejati (*truecolor image*)

1. Citra biner

Hanya mempunyai nilai 0 dan 1, tipe logical, 0 = hitam, 1 = putih



(Sumber gambar: Matlab)

```
>> A = imread('ganesha.bmp');  
>> imshow(A)  
>> size(A)  
ans =
```

```
150 150
```

```
>> A(101:110, 101:110)  
ans =
```

10×10 logical array

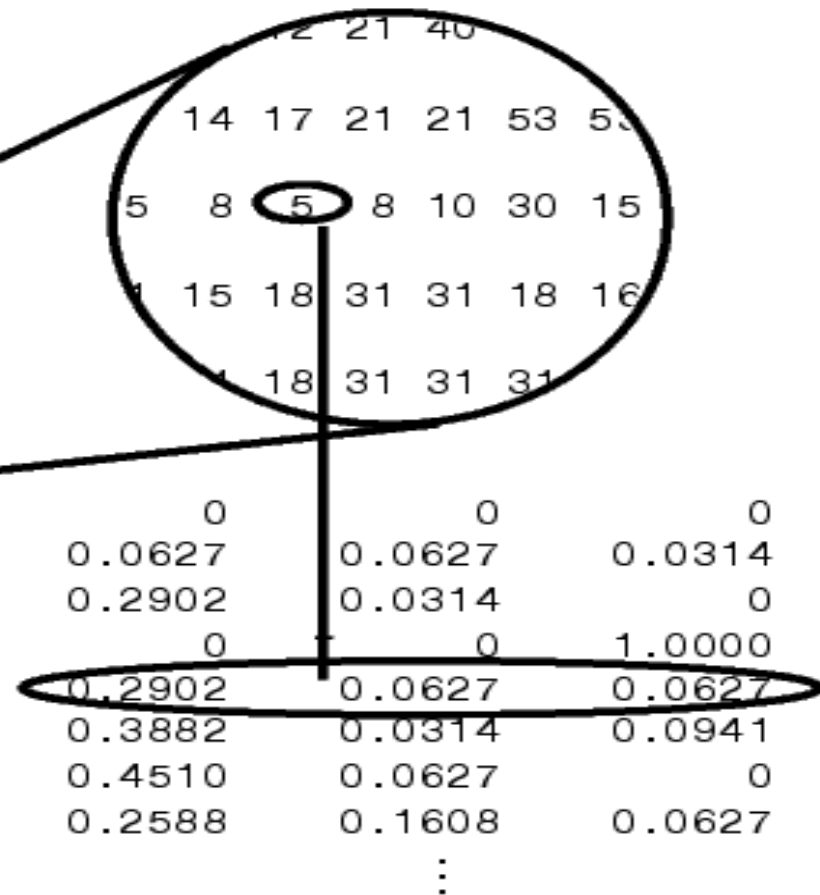
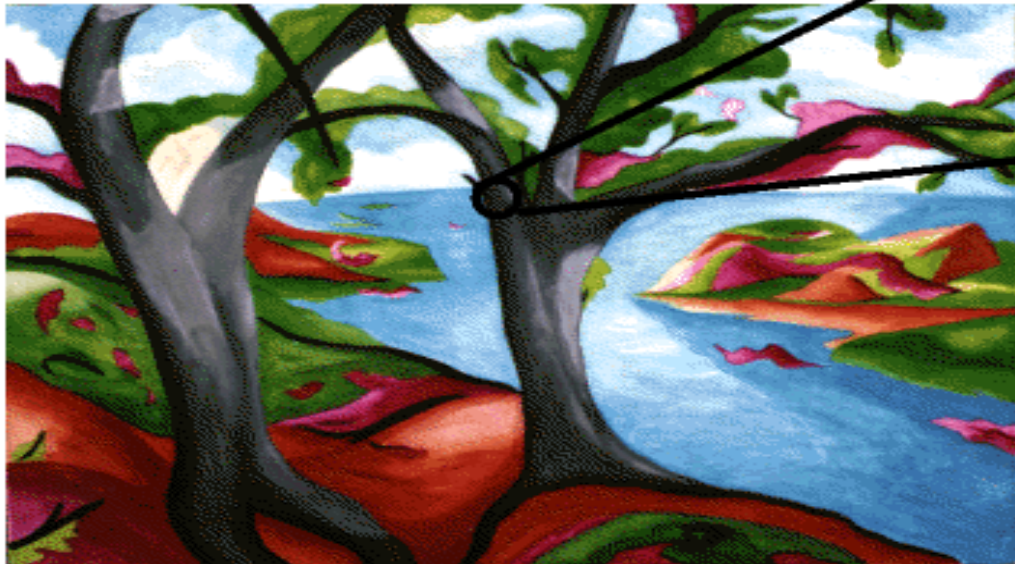
```
0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 1 1 1 1  
0 0 0 0 1 1 1 1 1 1  
1 0 0 0 0 1 1 1 1 1  
1 1 0 0 0 0 0 1 1 1  
1 1 1 0 0 0 0 0 0 1  
1 1 1 1 1 0 0 0 0 0  
1 1 1 1 1 1 1 1 0 0  
1 1 1 1 1 1 1 1 1 0  
0 0 0 0 0 0 0 0 0 0
```



2. Citra terindeks (*indexed image*)

- Citra terindeks terdiri dari sebuah palet warna (*colormap*) dan sebuah matriks yang nilai elemennya menyatakan indeks ke sebuah baris di dalam palet.
- Palet adalah matriks $m \times 3$ bertipe *floating-point* dengan nilai di dalam $[0, 1]$. Untuk citra dengan 256 warna, maka *map* berukuran 256×3 .
- Tiap baris di dalam palet menspesifikasikan komponen *red*, *green*, dan *blue* dari sebuah warna tunggal.
- Warna sebuah *pixel* adalah kombinasi setiap komponen *red* (R), *green* (G), dan *blue* (B) pada baris palet tersebut.
- Contoh citra terindeks adalah citra format GIF dan citra format BMP (untuk citra dengan 8 bit/pixel atau kurang)

(Sumber gambar: Matlab)



Pada gambar di atas, nilai *pixel* 5 menunjuk ke baris ke-5 dari palet. Nilai komponen warna pada baris ke-5 adalah $R = 0.2902$, $G = 0.0627$, dan $B = 0.0627$. Itu artinya warna pada *pixel* bernilai 5 merupakan kombinasi dari ketiga komponen R, G, dan B tersebut.

```
>> [A, map] = imread('kartun.gif');  
>> size(map)
```

```
ans =
```

```
256      3
```

```
>> imshow(A, map)
```



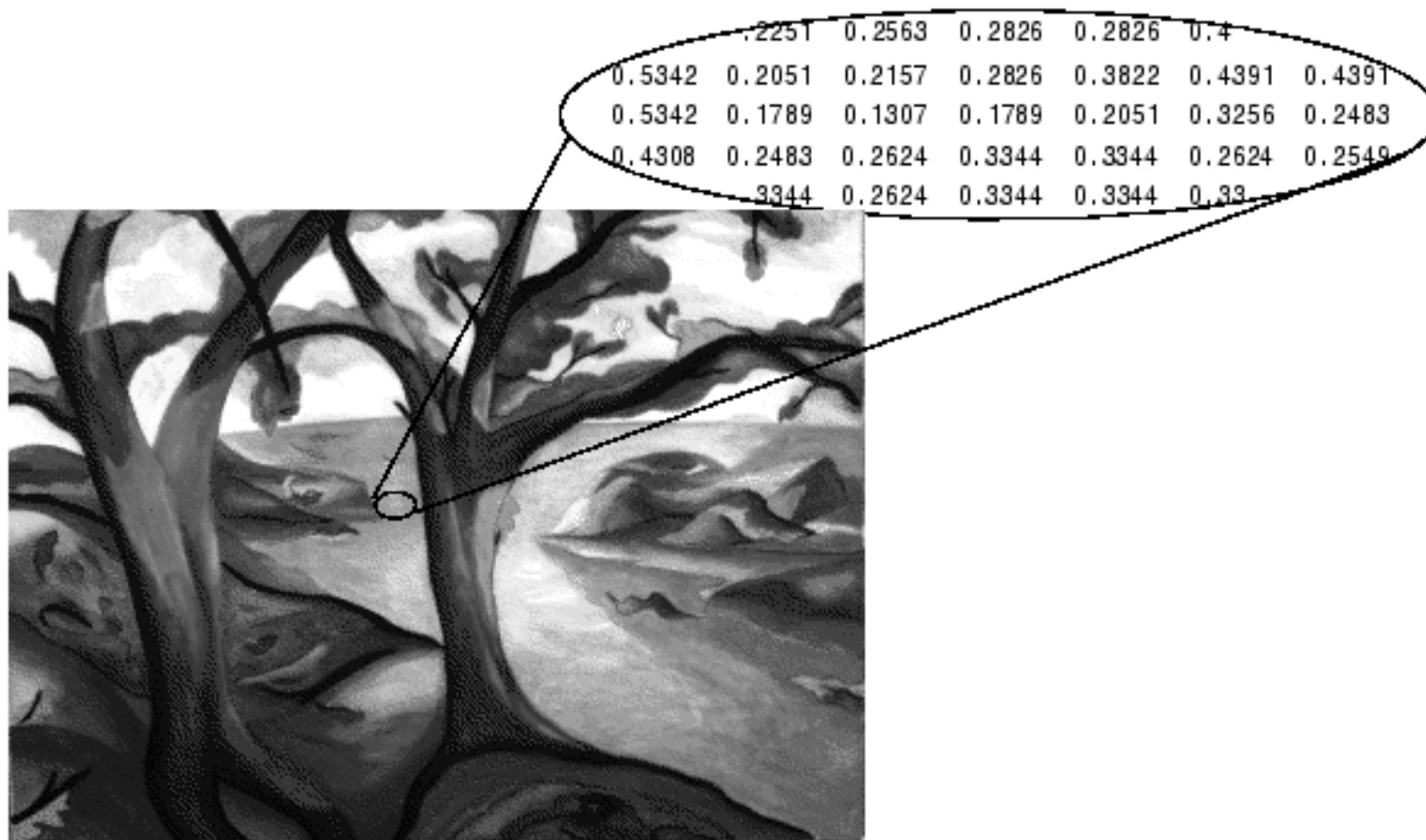
```
>> map(1:8, 1:3)
```

```
ans =
```

0.9137	0.9608	0.5412
0.0039	0	0
0.8471	0.3882	0.0863
0.9961	0.0039	0
0.6980	0.6980	0.6902
0.8078	0.5765	0.8039
0.6627	0.8392	0.8824
1.0000	0.8706	0.7843

3. Citra hitam-putih (*grayscale image*)

- Disebut juga *intensity image* atau *graylevel image*, setiap pixel menyatakan nilai intensitas.
- Direpresentasikan sebagai satu matriks Tunggal, setiap elemen matriks menyatakan satu pixel
- Untuk tipe data *single* atau *double*, nilai setiap pixel berada di dalam selang $[0, 1]$, 0 menyatakan hitam, 1 menyatakan putih, antara 0 dan 1 menyatakan degradasi dari hitam ke putih.
- Untuk tipe *uint8*, nilai setiap pixel di dalam selang $[0, 255]$, 0 menyatakan hitam, 255 menyatakan putih, antara 0 dan 255 menyatakan degradasi dari hitam ke putih.



(Sumber gambar: Matlab)

```
>> A = imread('boat.bmp');
```

```
>> size(A)
```

```
ans =
```

```
512    512
```

```
>> imshow(A)
```



```
>> A(125:135, 100:110)
```

```
ans =
```

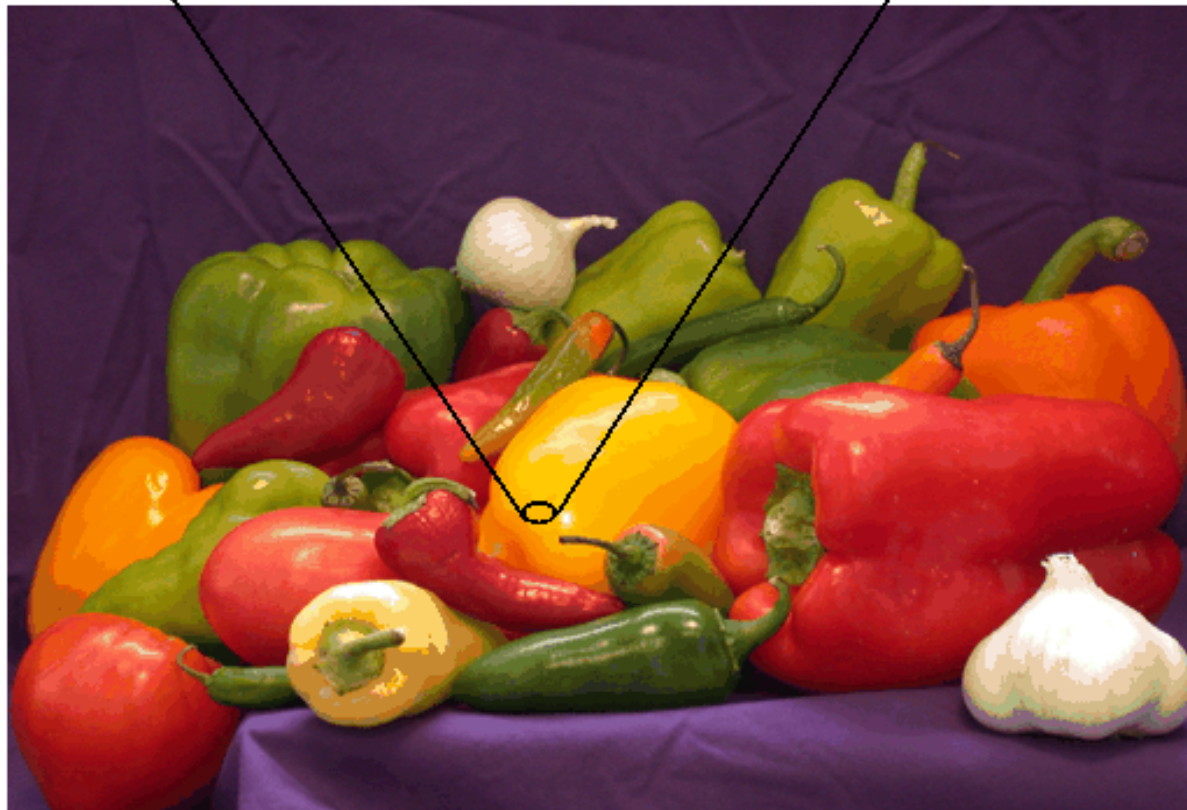
```
11×11 uint8 matrix
```

```
167 173 170 171 170 171 170 169 174 169 172
165 173 170 169 173 171 169 170 171 170 174
168 171 171 170 172 170 167 172 171 170 173
171 171 169 171 171 171 171 171 169 169 173
167 171 173 169 171 171 171 169 169 171 171
171 169 172 170 168 173 170 172 173 170 173
171 169 172 170 171 171 172 172 174 170 170
169 169 171 167 172 170 170 171 169 170 170
169 169 170 169 172 171 169 173 171 168 175
168 170 171 170 175 171 170 174 169 174 169
169 170 170 169 172 169 169 171 172 170 171
```

4. Citra warna-sejati (truecolor image)

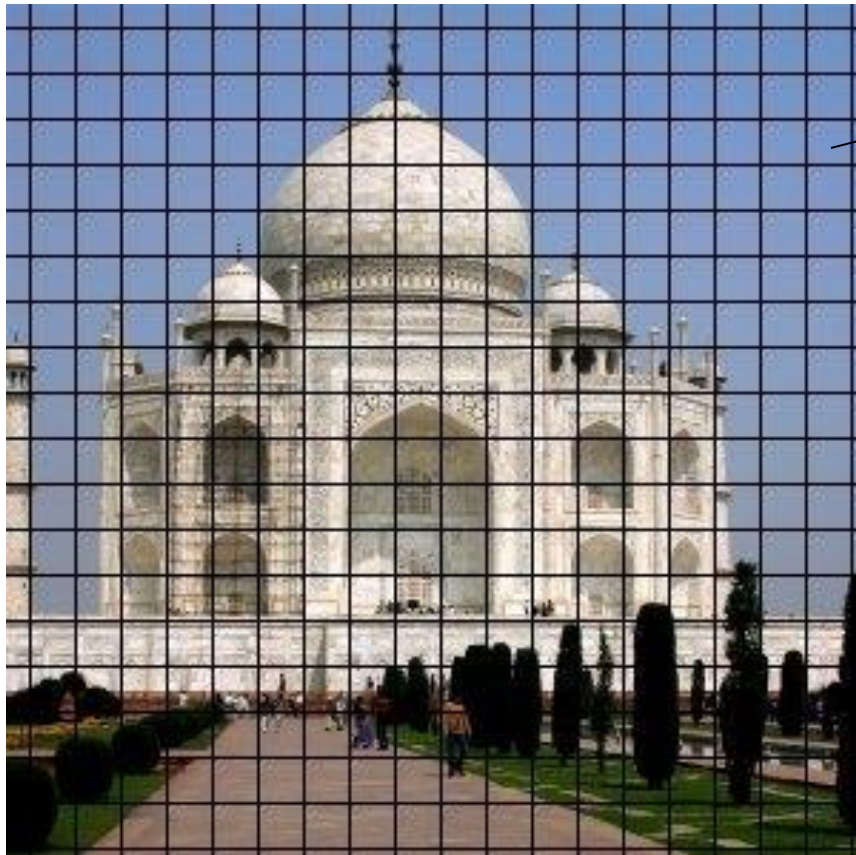
- Setiap pixel terdiri atas 3 komponen warna: red, green, dan blue. Warna setiap pixel ditentukan oleh kombinasi *red*, *green*, dan *blue*.
- Tidak ada palet warna (*colormap*) di dalam file citranya
- MATLAB menyimpan citra warna-sejati sebagai 3 buah matriks berukuran $M \times N$, setiap komponen warna sebagai satu matriks
- Untuk tipe single atau double, setiap komponen warna nilainya di dalam selang $[0, 1]$.
- Jika setiap komponen warna = 8 bit, maka citra warna-sejati adalah citra 24-bit.

0.5176	0.2235	0.1294	Blue	0.4190	0.2902	0.2902	0.4824
0.5804	0.2902	0.0627	0.2902	0.2902	0.2902	0.2902	0.4824
0.5804	0.0627	0.0627	0.0627	0.2235	0.2588	0.2588	0.2588
0.5176	0.1922	0.0627	Green	0.1922	0.2588	0.2588	0.2588
0.5176	0.1294	0.1608	0.1294	0.1294	0.2588	0.2588	0.2588
0.5176	0.1608	0.0627	0.1608	0.1922	0.2588	0.2588	0.2588
0.5490	0.2235	0.5490	Red	0.7412	0.7765	0.7765	0.902
0.5490	0.3882	0.5176	0.5804	0.5804	0.7765	0.7765	0.902
0.5490	0.2588	0.2902	0.2588	0.2235	0.4824	0.2235	0.2235
0.2235	0.1608	0.2588	0.2588	0.1608	0.2588	0.2588	0.2588
0.2588	0.1608	0.2588	0.2588	0.2588	0.2588	0.2588	0.2588



(Sumber gambar: Matlab)

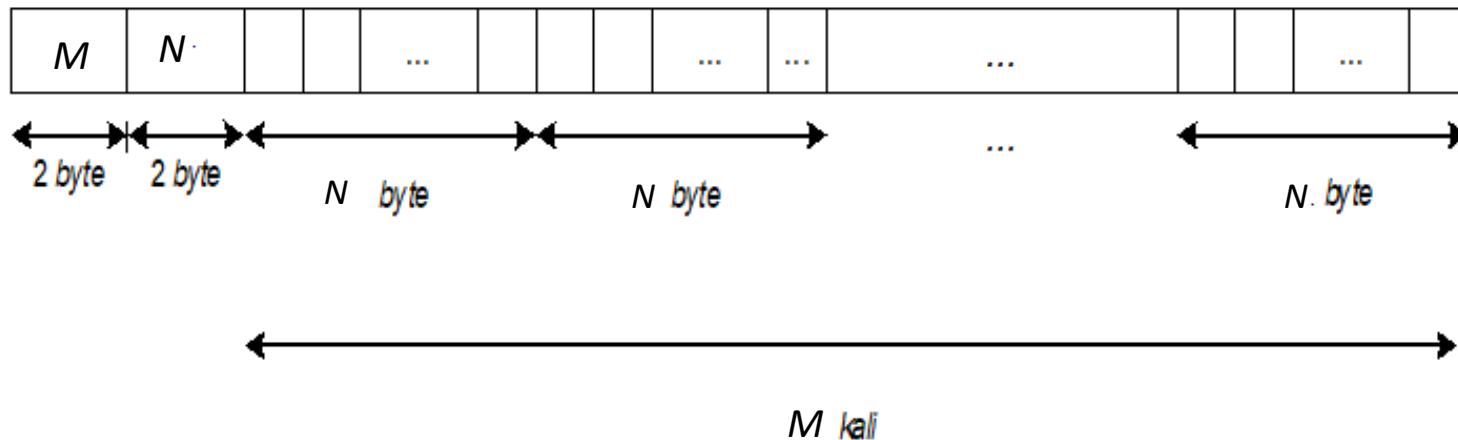
Pada citra 24-bit (*real image* atau *truecolor image*), 1 *pixel* = 24 bit, terdiri dari komponen RGB (*Red-Green-Blue*) sepanjang 24 bit, masing-masing 8 bit untuk setiap komponen.



100100111001010010001010
R G B

Format *Raw Image*

- *Raw image* atau citra mentah adalah format citra yang tidak mempunyai format spesifik.
- Hanya berisi informasi ukuran citra (tinggi x lebar atau $M \times N$) dan data *bitmap*



- Citra mentah juga dapat disimpan di dalam file teks dalam format ASCII. Umumnya untuk citra *grayscale* 8-bit/*pixel*.
- Baris pertama menyatakan ukuran citra (M x N), baris-baris berikutnya menyatakan nilai-nilai *pixel*. Setiap nilai *pixel* dipisahkan dengan spasi.

9	11									
148	162	175	182	189	194	195	193	195	195	197
148	164	174	176	185	189	191	191	196	194	195
144	159	167	176	178	185	188	191	196	194	197
128	147	157	168	173	179	182	184	191	191	192
119	134	148	160	164	170	179	176	181	189	185
145	124	142	151	160	168	169	174	180	182	183
172	120	140	153	157	169	171	178	180	182	182
196	120	129	144	152	158	167	170	177	176	178
204	144	116	134	142	149	155	165	165	170	171

Pengolahan Citra dengan Matlab

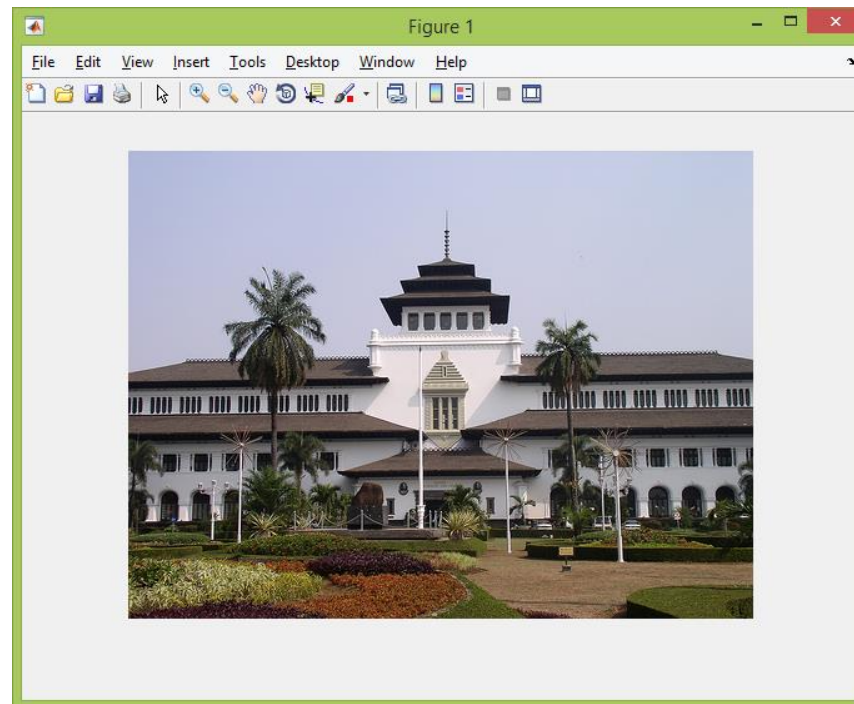
- Matlab memiliki *Image Processing Toolbox* yang dapat digunakan untuk pemrosesan citra digital.
- *Image processing toolbox* memiliki fungsi-fungsi *built-in* untuk pemrosesan citra.
- Di dalam Matlab, citra direpresentasikan dengan matriks M x N, namun indeks matriks dimulai dari 1, bukan dari nol. *Pixel* pada sudut kiri atas adalah pada posisi (1, 1).

$$f = \begin{bmatrix} f(1,1) & f(1,2) & \cdots & f(1,N) \\ f(2,1) & f(2,2) & \cdots & f(2,N) \\ \vdots & \vdots & \vdots & \vdots \\ f(M,1) & f(M,2) & \cdots & f(M,N) \end{bmatrix}$$

- Membaca citra dan menampilkan ke layar

```
>> I = imread('gedung-sate.jpg');  
>> imshow(I)  
>> whos
```

Name	Size	Bytes	Class	Attributes
I	374x500x3	561000	uint8	



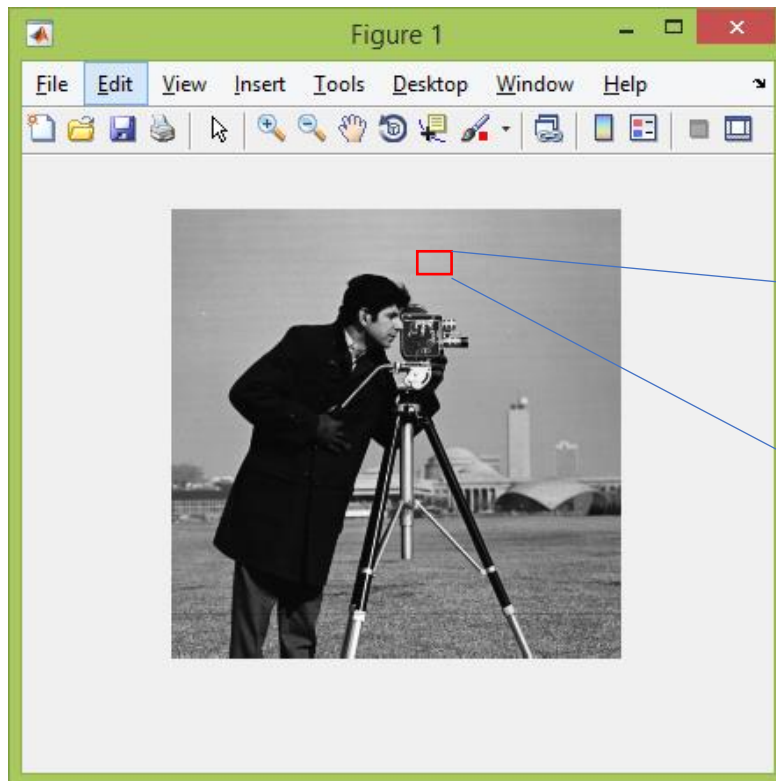
- Citra *grayscale*

```
>> img = imread('camera.bmp');
```

```
>> whos
```

Name	Size	Bytes	Class	Attributes
img	256x256	65536	uint8	

```
>> imshow(img)
```



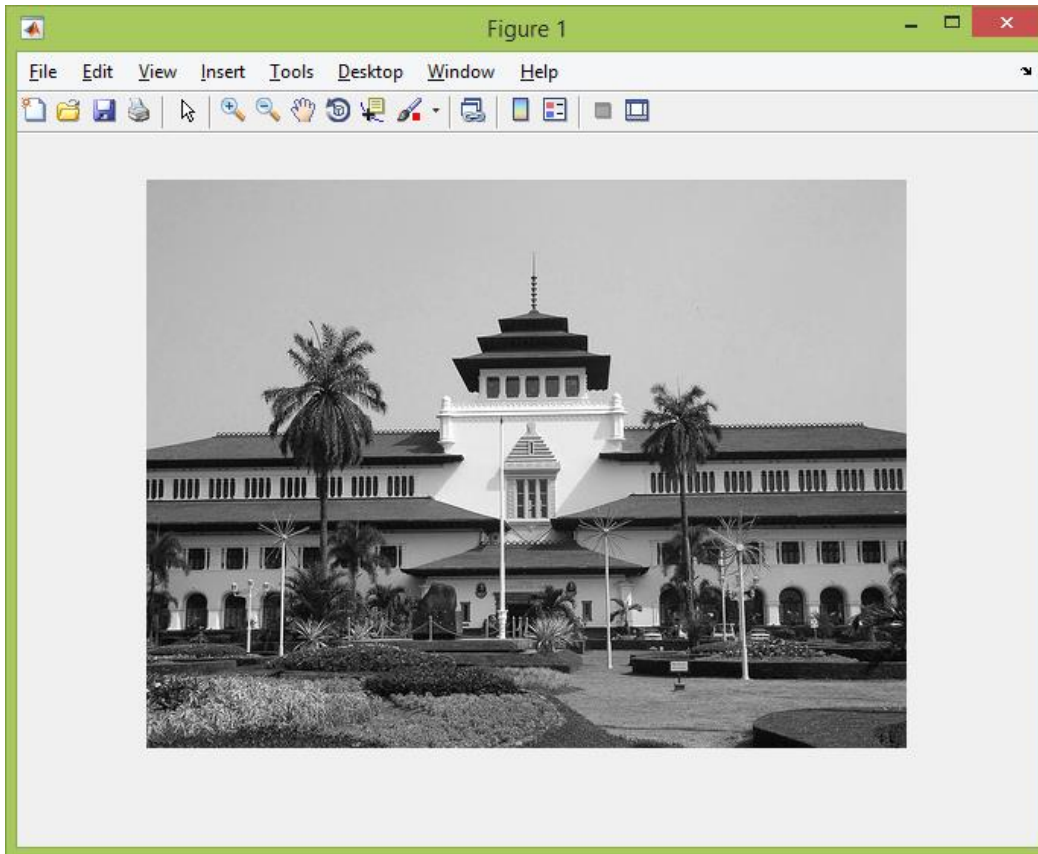
```
>> img(20:25, 100:110)
```

```
ans =
```

183	184	184	183	185	185	185	188	189	187	188
182	185	185	186	181	185	185	187	187	189	185
184	183	185	185	179	187	186	184	185	189	187
183	182	182	181	185	180	183	180	181	184	185
180	183	183	182	189	184	186	185	186	186	185
182	185	180	179	182	185	185	183	185	187	187

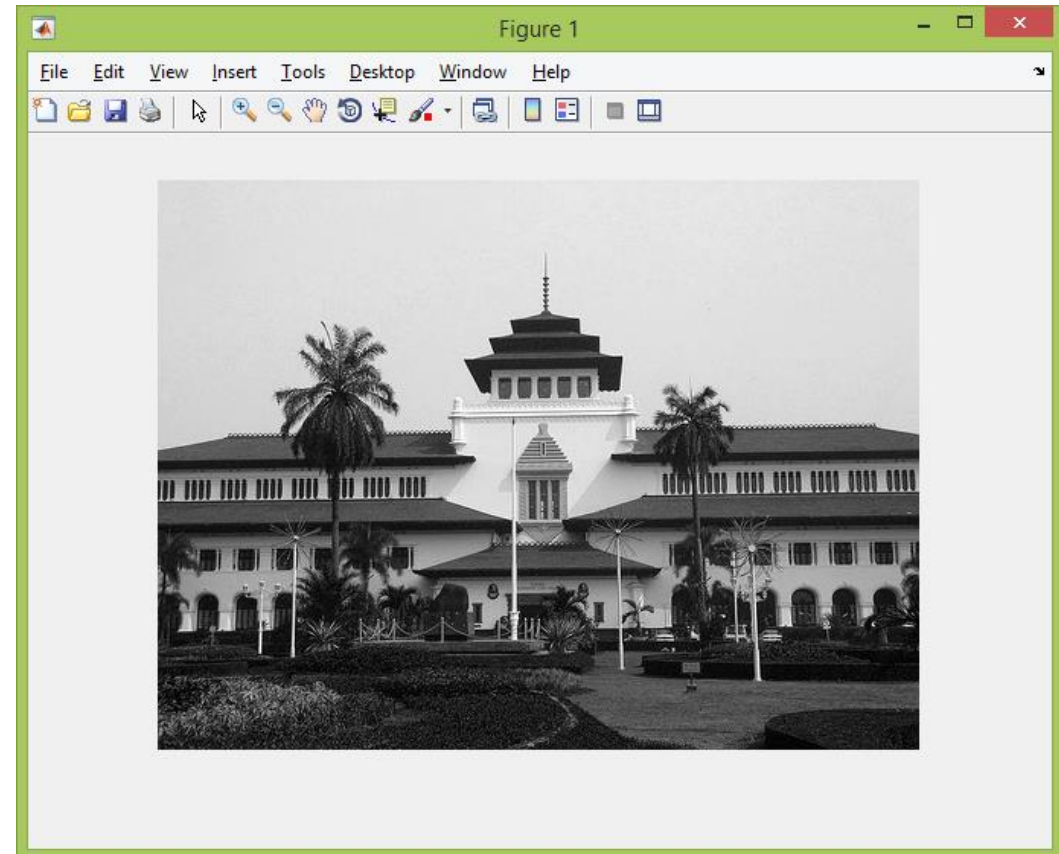
- Tampilkan hanya komponen *red* dari citra

```
>> imshow(I(:, :, 1))
```



- Tampilkan komponen *blue*

```
>> imshow(I(:, :, 3))
```



- Tampilkan data bitmap

```
>> I(:, :, 3)
```

The image shows the MATLAB R2016a interface. The Command Window displays the following output for the command `I(:, :, 3)`:

```
ans =  
  
Columns 1 through 38  
  
218 217 215 217 218 220 219 216 217 216 217 217 216 217 218 218 217 217 215 218 218 218 220 219 218 216 216 216 217 219 219 219 221 221 219 217 218  
217 218 216 219 219 220 218 216 217 217 216 217 217 217 217 218 217 217 217 217 218 218 218 218 216 216 217 216 216 216 217 217 218 221 221 219 219 218  
218 218 216 217 219 220 219 216 218 217 217 217 217 217 217 218 218 218 218 219 219 219 219 216 216 219 219 218 218 217 217 218 219 221 219 219 220  
218 218 216 215 218 218 220 219 218 218 219 220 220 218 218 218 218 219 219 220 220 219 214 217 219 219 220 218 217 215 219 219 222 221 220 220  
218 218 216 216 218 219 220 219 218 218 220 220 219 220 218 218 218 218 218 219 219 220 217 217 214 215 218 219 219 218 217 215 219 220 222 220 220 220  
218 219 217 217 220 219 219 217 217 218 220 220 219 217 218 218 219 218 218 218 219 217 217 217 214 215 217 219 219 219 217 217 219 220 223 220 220 220  
219 217 217 218 220 220 219 217 215 218 218 220 219 218 218 215 220 219 219 218 219 217 217 215 215 215 217 217 217 219 219 220 221 223 220 218 218  
219 216 216 217 220 220 220 218 216 219 218 220 220 219 216 215 219 219 218 218 218 216 217 215 217 217 215 215 217 217 219 219 220 221 221 221 218 218  
220 217 214 216 219 220 220 218 219 221 220 218 218 223 222 217 216 216 216 217 217 217 217 216 218 219 221 221 221 221 219 217 219 218 219 220 220 221  
215 219 220 219 217 215 218 220 221 216 216 220 220 218 219 221 216 218 218 219 219 217 217 216 216 219 219 219 219 219 217 220 220 220 220 220 220  
215 219 220 219 217 215 218 220 221 216 216 219 220 218 216 220 219 218 218 218 218 219 219 219 217 217 217 217 218 218 218 220 221 220 221 221 220  
220 217 215 216 220 219 218 216 218 220 218 215 216 221 220 214 218 218 218 218 219 217 217 217 217 217 218 216 216 216 218 220 222 219 220 219 218 218  
220 217 216 216 220 220 218 216 218 220 217 214 216 220 220 214 216 216 216 217 217 217 217 215 219 219 218 218 218 220 221 221 218 216 216 218  
217 220 221 220 216 216 218 220 221 215 213 220 221 218 216 218 215 215 217 218 218 218 217 217 219 219 220 220 220 220 220 220 222 217 216 216 216 216 218  
217 220 220 220 217 215 218 220 221 215 213 220 221 219 216 218 217 217 217 219 220 220 220 219 217 218 218 220 220 218 219 219 216 218 218 220 221 221  
221 218 216 216 219 220 218 216 218 219 219 217 220 222 220 214 217 219 218 221 221 221 221 221 214 215 217 217 217 216 215 218 220 223 223 223  
219 219 220 220 221 220 220 218 221 220 221 220 218 218 220 220 220 219 218 216 217 218 217 218 217 216 216 216 217 219 218 217 219 218 217 217 218 219  
220 220 220 220 219 219 219 217 221 221 221 219 218 218 220 220 218 219 218 216 217 217 216 217 218 216 216 216 218 217 217 220 219 218 218 218 219  
219 220 218 218 218 218 220 220 218 218 219 219 219 219 219 219 217 218 218 218 219 217 218 217 216 219 218 218 217 217 218 220 220 220 220 221 219 219  
219 219 217 216 217 219 222 222 217 218 216 217 219 219 219 221 217 218 218 218 218 220 219 219 216 217 218 218 217 217 218 220 220 220 220 221 219 219  
220 218 217 216 217 219 221 220 218 216 216 216 217 219 219 219 216 217 217 217 220 220 219 219 217 217 218 218 218 218 218 219 220 220 221 221 219  
218 218 217 218 218 219 220 220 218 218 219 217 217 218 218 218 216 217 217 219 220 220 219 219 218 218 218 218 218 218 217 217 218 219 221 221 219  
215 217 217 220 220 220 221 221 218 218 219 219 219 219 219 219 216 218 217 219 220 220 217 217 216 216 218 218 218 218 217 217 216 219 219 219 219 219  
215 218 218 220 219 219 220 219 219 220 218 219 219 220 220 220 217 218 219 219 220 219 216 216 214 215 216 218 218 218 218 218 216 216 219 219 218 219  
218 218 220 220 221 219 219 218 218 220 217 219 221 221 219 215 217 216 218 220 220 217 217 218 220 220 218 218 216 218 216 218 218 220 218 217 217 218 219 221  
217 218 218 220 220 218 218 219 218 218 221 222 221 219 216 217 216 217 220 220 218 217 218 220 218 218 216 216 217 219 219 217 216 216 217 219 219  
215 218 218 218 217 217 218 219 220 221 221 221 222 222 220 216 216 217 217 217 220 218 218 217 218 218 216 216 216 216 218 217 217 217 217 217 216  
215 216 218 218 217 217 218 218 221 219 219 221 221 219 217 214 217 217 217 220 218 220 219 220 218 217 215 216 218 218 218 218 216 217 217 216  
216 216 216 218 218 218 218 218 218 216 217 217 219 219 217 215 214 215 217 217 217 218 218 219 221 220 220 218 218 219 219 218 218 218 219 219 219  
216 216 216 218 218 218 218 218 218 216 217 217 219 219 217 215 214 215 217 217 217 218 218 219 221 220 220 218 218 219 219 218 218 218 219 219 219
```

• Membaca citra GIF

```
>> [C, map] = imread('kartun.gif');
```

```
>> whos
```

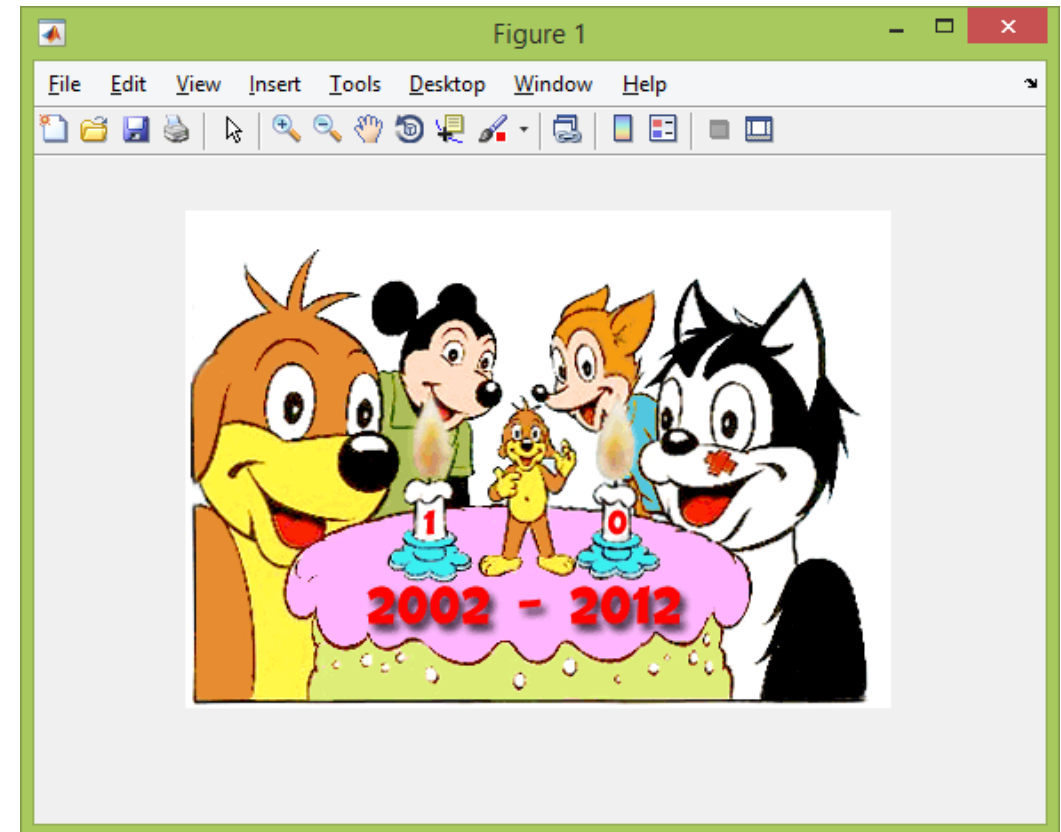
Name	Size	Bytes	Class	Attributes
C	280x397	111160	uint8	
map	256x3	6144	double	

```
>> imshow(C, map)
```

```
>> map
```

```
map =
```

0.9137	0.9608	0.5412
0.0039	0	0
0.8471	0.3882	0.0863
0.9961	0.0039	0
0.6980	0.6980	0.6902
0.8078	0.5765	0.8039
0.6627	0.8392	0.8824
1.0000	0.8706	0.7843
0.9647	0.6118	0.0627
0.6784	0.7412	0.3490
0.9608	0.9333	0.6902
...	...	...



- Menyimpan citra

```
>> imwrite(img, 'camera.jpg', 'jpg');
```

- Membaca citra *animated GIF*

```
[citra map]=imread('walk.gif', 'frames','all');
```

```
s = size(citra);  
numframes=s(4);
```

```
for n=1:numframes;  
    A = citra(:,:, :,n);  
    figure; imshow(A,map);  
end
```



- Frame-frame citra walk.gif

