

## PENGOLAHAN CITRA DIGITAL

Nama : Mitra

Kelas : PTIK B

NIM : 200209501019

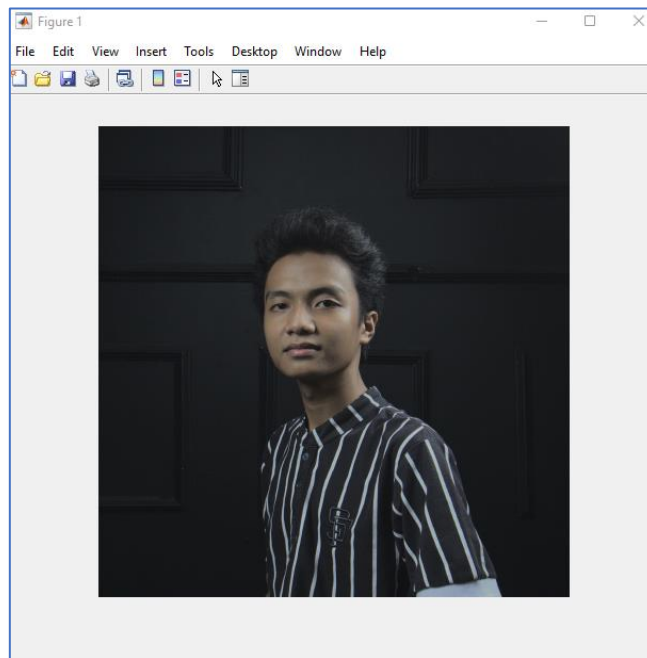
---

### MEMBACA CITRA MENJADI SEBUAH DATA MATRIKS

Hal pertama yang harus dilakukan sebelum melakukan transformasi intensitas citra pada matlab yaitu membaca file gambar. File gambar ditaruh pada path tempat file matlab disimpan agar dapat diakses dengan mudah. Caranya adalah dengan menggunakan fungsi `imread()`.

```
%GAMBAR
image = imread('image.png');
figure(1); imshow(image)
```

Pada Gambar diatas kita membuat sebuah variable *image* yang diisikan dengan fungsi `imread('image.png')` berarti citra 'image.png' diubah menjadi sebuah data matriks. Kemudian kita akan menampilkan hasil matriksnya dengan menggunakan fungsi `imshow()`. Caranya adalah pertama buat sebuah figure baru dengan menggunakan fungsi `figure()` kemudian tampilkan hasil matriksnya menggunakan `imshow()` dengan parameter variable yang berisi matriks tersebut disini kita menggunakan variable *image*. Hasilnya adalah seperti gambar dibawah :

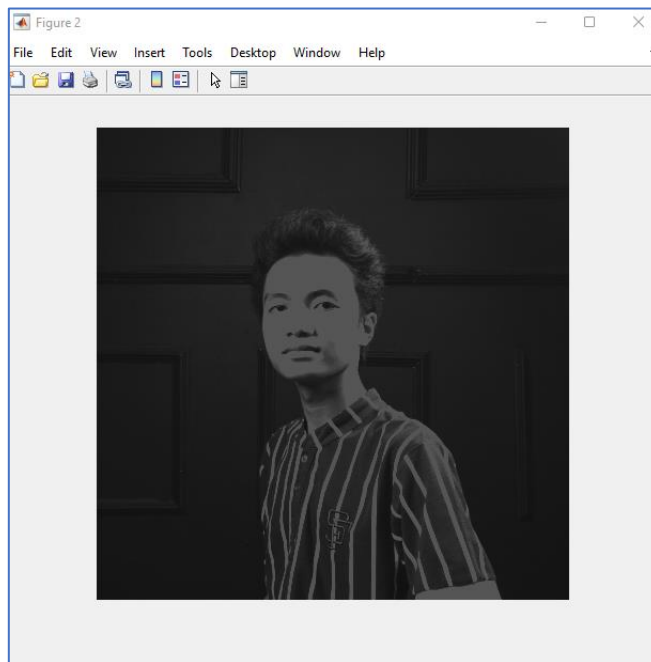


## TRANSFORMASI CITRA BERWARNA KE CITRA KEABUAN

Untuk mengubah citra berwarna ke citra keabuan dapat menggunakan nilai rata rata dari nilai intensitas ketiga komponen. Caranya dapat dilihat pada gambar dibawah

```
%GAMABR KEABUAN  
image_gray = (image(:,:,1) + image(:,:,2) + image(:,:,3)) / 3;  
figure(2); imshow(image_gray)
```

Pada baris pertama kita membuat sebuah variable image\_gray yang diisikan dengan nilai rata rata dari nilai intensitas ketiga komponen. Semua nilai intensitas ditambahkan kemudian dibagi dengan 3. Selanjutnya hasilnya kita tampilkan pada figure baru dengan menggunakan fungsi figure(2) dan fungsi imshow(image\_gray) untuk menampilkan hasil dari matriks pada variable image\_gray. Hasilnya akan seperti gambar dibawah

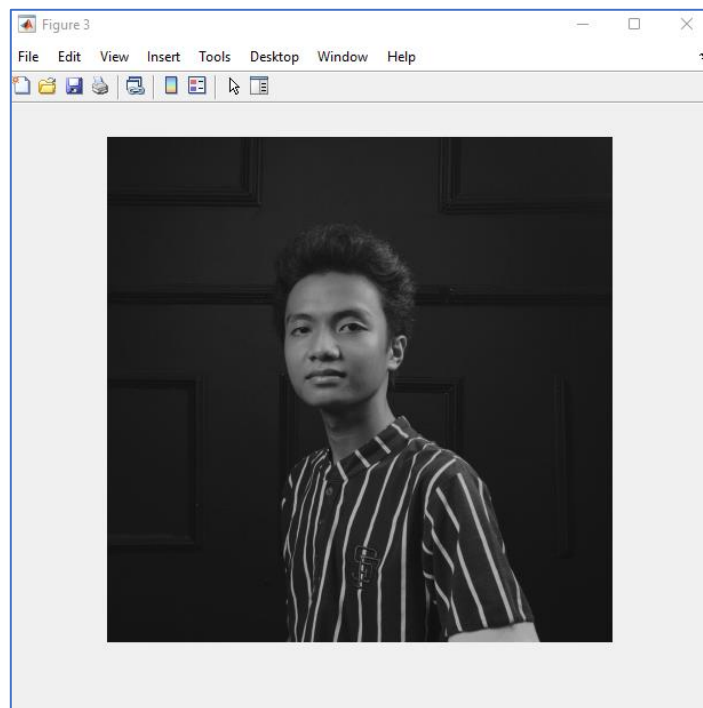


## TRANSFORMASI CITRA BERWARNA KE CITRA KEABUAN MENGUNAKAN NILAI KONSTANTA

Untuk mengubah citra berwarna ke citra keabuan menggunakan nilai konstanta digunakan suatu komposisi tertentu. Menurut penelitian, mata manusia lebih peka terhadap warna tertentu jadi pengubahan citra berwarna ( $I_{RGB}$ ) ke citra keabuan ( $I_{gray}$ ) dengan menggunakan suatu komposisi tertentu. Nilai konstanta yang dimaksud adalah 0.4 , 0.32 dan 0.28 ketiga nilai ini dikalikan dengan nilai intensitas ketiga komponen. Caranya dapat dilihat pada gambar dibawah

```
%GAMBAR KEABUAN MENGGUNAKAN KONSTANTA
image_gray_const = image(:,:,1) * 0.4 + image(:,:,2) * 0.32 + image(:,:,3) * 0.28;
figure(3); imshow(image_gray_const)
```

Pada baris pertama kita membuat sebuah variable `image_gray_const` yang diisi dengan hasil dari perkalian ketiga nilai konstanta dengan nilai intensitas ketiga komponen. Semua nilai intensitas ditambahkan. Selanjutnya hasilnya kita tampilkan pada figure baru dengan menggunakan fungsi `figure(3)` dan fungsi `imshow(image_gray_const)` untuk menampilkan hasil dari matriks pada variable `image_gray_const`. Hasilnya akan seperti dibawah ini, hasilnya lebih bagus dari transformasi citra bewarna ke citra keuan dengan cara pertama yaitu menghitung rata rata.



## TRANSFORMASI CITRA KEABUAN KE CITRA BINER

```
%GAMBAR KEABUAN MENGGUNAKAN BINNER
value = 100;
[kolom, baris] = size(image_gray_const);
image_gray_biner = zeros(kolom, baris);
```

Untuk mengubah citra keabuan ke citra biner kita dapat menggunakan sebuah nilai ambang (threshold) disini saya menggunakan nilai ambang 100 dan ditaruh pada variable value. Kemudian pada baris kedua kita menghitung size dari variable image\_gray\_const yang dimana berisi matriks citra keabuan yang tadi kita buat. Hasil sizenya kita taruh pada dua variable yaitu kolom dan baris. Pada baris ketiga kita membuat sebuah variable image\_gray\_biner yang besarnya sama dengan variable image\_gray\_const namun hanya berisi nilai 0.

```
for x = 1 : kolom
    for y = 1 : baris
        if(image_gray_const(x,y) > value)
            image_gray_biner(x,y) = 1;
        else
            image_gray_biner(x,y) = 0;
        end
    end
end

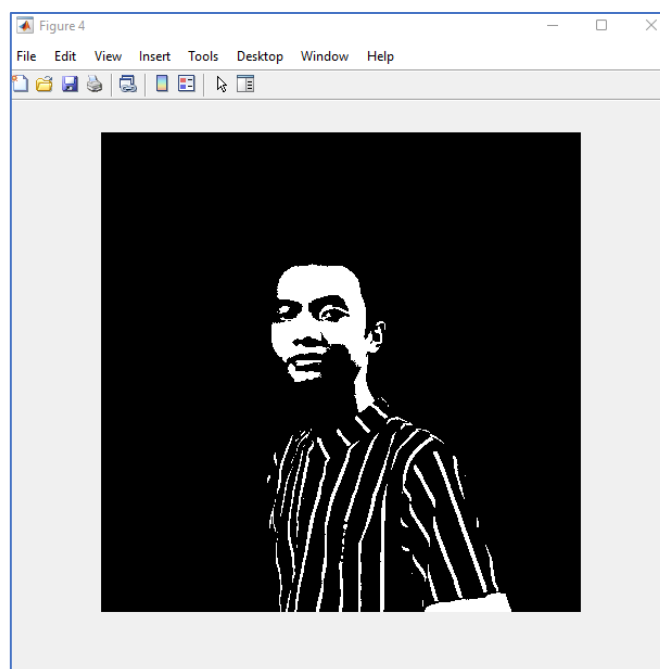
figure(4); imshow(image_gray_biner)
```

Kemudian kita akan mengisi variable image\_gray\_biner agar nilainya berisi 1 atau 0. Pertama buat sebuah perulangan dimulai dari angka 1 yang ditaruh pada variable x hingga jumlah nilai dari variable kolom. Selanjutnya

didalam perulangan kolom kita membuat perulangan lagi untuk nilai baris yang ditaruh pada variable y. Didalam perulangan baris, dibuat sebuah kondisi dimana jika nilai `image_gray_const(x,y)` atau nilai citra pada titik tersebut diatas dari variable value (100) maka nilai pada `image_gray_biner(x,y)` akan bernilai 1 selain itu 0.

Hal tersebut akan terus berulang hingga kolom dan baris pada variable `image_gray_const` telah dicek semua dan variable `image_gray_biner` akan berisi kolom dan baris yang sama namun dengan nilai 1 dan 0.

Terakhir hasilnya kita tampilkan pada figure baru dengan menggunakan fungsi `figure(4)` dan fungsi `imshow(image_gray_biner)` untuk menampilkan hasil dari matriks pada variable `image_gray_biner`.



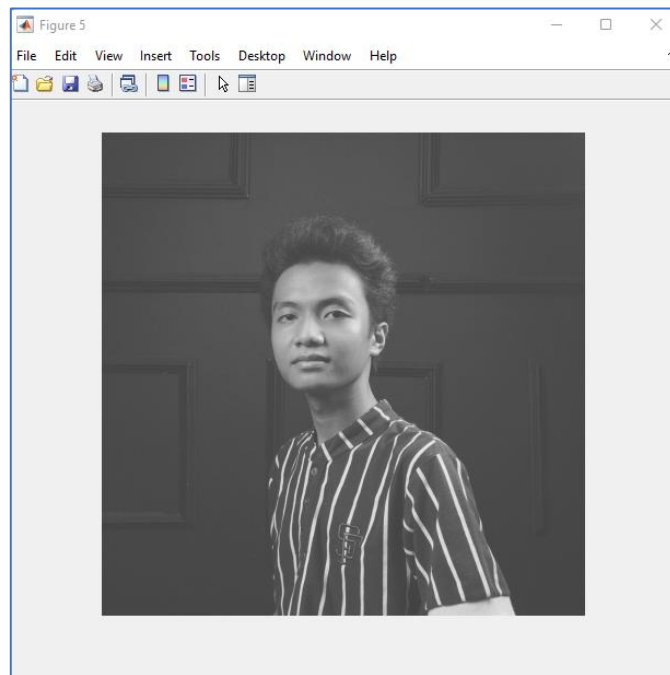
## PENGATURAN CITRA BRIGHTNESS

Proses pengaturan brightness adalah proses penambahan nilai derajat keabuan  $x$  dengan nilai perubahan brightness. Jadi semua nilai pada matriks akan ditambahkan dengan nilai yang sudah ditentukan.

```
%GAMBAR BRIGHTNESS  
image_brightness = image_gray_const + 50;  
figure(5); imshow(image_brightness)
```

Pada baris pertama kita membuat sebuah variable Bernama `image_brightness` yang berisi nilai matriks pada variable `image_gray_const` yang tadi kita buat ditambahkan nilai yang sudah ditentukan disini nilai yang ditentukanya adalah 50. Nilai tersebut dapat ditambahkan jika ingin membuat brightness atau kecahayaan gambar lebih terang atau lebih gelap.

Terakhir hasilnya kita tampilkan pada figure baru dengan menggunakan fungsi `figure(5)` dan fungsi `imshow(image_brightness)` untuk menampilkan hasil dari matriks pada variable `image_brightness`. Hasilnya akan seperti pada gambar berikut



## PENGATURAN CITRA CONTRAST

Proses pengaturan contrast adalah proses perkalian nilai derajat keabuan  $x$  dengan nilai perubahan contrast. Jadi semua nilai pada matriks akan dikalikan dengan nilai yang sudah ditentukan.

```
%GAMBAR CONTRAST  
image_contrast = image_gray_const * 1.1;  
figure(6); imshow(image_contrast)
```

Pada baris pertama kita membuat sebuah variable Bernama `image_contrast` yang berisi nilai matriks pada variable `image_gray_const` yang tadi kita buat dikalikan nilai yang sudah ditentukan disini nilai yang ditentukannya adalah 1.1. Nilai tersebut dapat diubah jika ingin membuat contrast gambar lebih tinggi atau lebih rendah.



Terakhir hasilnya kita tampilkan pada figure baru dengan menggunakan fungsi `figure(5)` dan fungsi `imshow(image_contrast)` untuk menampilkan hasil dari matriks pada variable `image_contrast`.

