

TUGAS MATA KULIAH PENGOLAHAN CITRA DIGITAL

“TRANSFORMASI INTENSITAS CITRA”



MUH. ALFIAN HANAFI

200209502028

PTIK B

PRODI PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER

FAKULTAS TEKNIK

UNIVERSITAS NEGERI MAKASSAR

TAHUN 2021

TRANSFORMASI INTENSITAS CITRA

❖ MEMBACA CITRA MENJADI SEBUAH DATA MATRIKS

Hal pertama yang harus dilakukan sebelum melakukan transformasi intensitas citra pada matlab yaitu membaca file gambar. Gambar yang akan digunakan sebaiknya mempunyai dimensi yang cukup kecil dan file gambar ditaruh pada path tempat file matlab agar dapat diakses dengan mudah. Caranya adalah dengan menggunakan fungsi `imread()`.

```
%Gambar
foto = imread('foto.jpeg');
figure(1); imshow(foto)
Untitled
```

Dapat kita lihat pada gambar di atas kita membuat sebuah variabel yaitu variabel “*foto*” yang diisikan dengan fungsi `imread` (‘foto.jpeg’) yang dimana citra ‘fotoku.jpg’ diubah menjadi sebuah data matriks. Kemudian jika kita ingin menampilkan matriksnya yaitu dengan menggunakan fungsi `imshow()`. Caranya yaitu pertama buat sebuah figure baru dengan menggunakan fungsi `figure()` lalu tampilkan hasil matriksnya menggunakan `imshow()` dengan parameter variable yang berisi matriks tersebut disini kita menggunakan variabel *foto*. Selanjutnya kita harus menjalankan atau RUN code yang telah kita buat apakah berhasil dan tidak terjadi error. Nah hasilnya dapat dilihat seperti gambar di bawah ini yaitu bentuknya citra RGB atau citra berwarna.

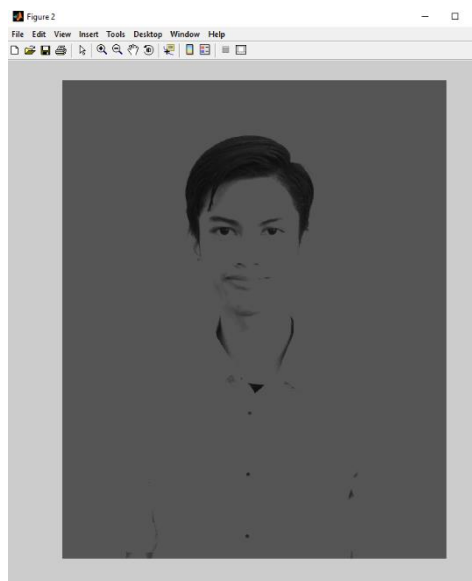


❖ Transformasi Citra Berwarna Ke Citra Keabuan

Selanjutnya untuk mengubah citra berwarna ke citra keabuan dapat dilakukan menggunakan dua cara yaitu dengan menggunakan rumus rata-rata dan rumus konstanta. Yang pertama yaitu Untuk mengubah citra berwarna ke citra keabuan dapat menggunakan nilai rata rata dari nilai intensitas ketiga komponen. Caranya dapat dilihat seperti pada gambar dibawah ini.

```
%Gambar Keabuan  
foto_gray= (foto(:,:,1) + foto(:,:,2) + foto(:,:,3)) / 3;  
figure(2); imshow(foto_gray)
```

Pada baris pertama kita membuat sebuah variable foto_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(foto_gray) untuk menampilkan hasil dari matriks pada variable I_gray_average tersebut . Lalu jika dijalankan, hasilnya akan seperti gambar dibawah ini.



❖ Transformasi Citra Berwarna Ke Citra Keabuan Menggunakan Nilai Konstanta

Selanjutnya jika kita ingin 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 (IRGB) ke citra keabuan (foto 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
foto_gray_const = foto(:,:,1) * 0.4 + foto(:,:,2) * 0.32 + foto(:,:,3) * 0.28;
figure(3); imshow(foto_gray_const)
```

Pada baris pertama kita membuat sebuah variable foto_gray_const yang diisikan dengan hasil dari perkalian ketiga nilai konstanta dengan nilai intensitas ketiga komponen dan semua nilai intensitas ditambahkan. Selanjutnya jika ingin melihat hasilnya kita tampilkan pada figure baru dengan menggunakan fungsi figure() dan fungsi imshow(foto_gray_const) untuk menampilkan hasil dari matriks pada variable foto_gray_const. Jika menggunakan nilai konstanta hasilnya akan lebih bagus dari transformasi citra berwarna ke citra keabuan dengan cara pertama yaitu menghitung rata rata. Jika dijalankan atau di RUN maka hasilnya akan seperti dibawah ini.



❖ Transformasi Citra Keabuan Ke Citra Biner

```
%Gambar Keabuan Menggunakan Biner
value = 120;
[kolom, baris] = size(foto_gray_const);
foto_gray_biner = zeros(kolom, baris);
```

Kemudian jika ingin mengubah citra keabuan ke citra biner kita dapat menggunakan sebuah nilai ambang (threshold). Disini saya menggunakan nilai ambang 120 dan disimpan pada variable value. Kemudian pada baris kedua kita menghitung size dari variable

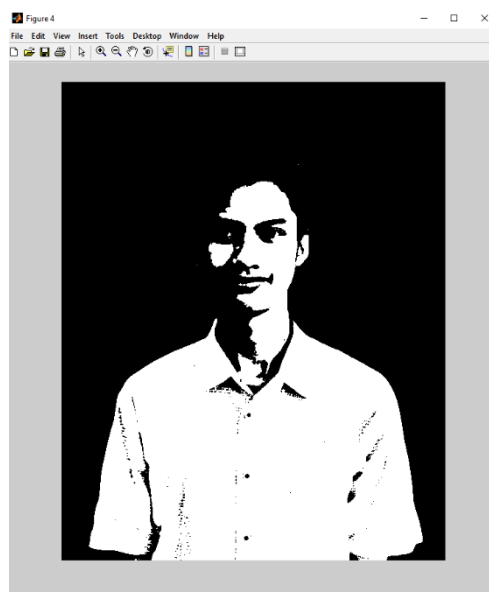
foto_gray_const yang dimana berisi matriks citra keabuan yang telah kita buat sebelumnya. Hasil sizenya kita taruh pada dua variable yaitu kolom dan baris. Selanjutnya pada baris ketiga kita membuat sebuah variable foto_gray_biner yang besarnya sama dengan variable foto_gray_const namun pada citra biner hanya berisi nilai 0 (kosong).

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

figure(4); imshow(foto_gray_biner)
```

Lalu kita akan mengisi variable foto_gray_biner agar nilainya berisi 1 atau 0. Caranya yaitu pertama, kita membuat sebuah perulangan dimulai dari angka 1 yang ditaruh pada variable x hingga jumlah nilai dari variable kolom. Selanjutnya didalam perulangan kolom tersebut kita juga membuat perulangan lagi untuk nilai baris yang ditaruh pada variable y. Didalam perulangan baris, dibuat sebuah kondisi dimana jika nilai foto_gray_const(x,y) atau nilai citra pada titik tersebut diatas dari variable value (120) maka nilai pada foto_gray_biner(x,y) akan bernilai 1 selain itu 0.

Terakhir kita akan dijalankan hasilnya akan kita tampilkan pada figure baru dengan menggunakan fungsi figure() dan fungsi imshow(foto_gray_biner) untuk menampilkan hasil dari matriks pada variable foto_gray_biner. Contohnya seperti pada gambar di bawah ini.



❖ Pengaturan Kontras

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  
foto_contrast = foto_gray_const * 1.3;  
figure(6); imshow(foto_contrast)
```

Dapat kita lihat pada baris pertama kita membuat sebuah variable yang bernama `foto_brightness` yang berisi nilai matriks pada variable `foto_gray_contrast` yang telah kita buat tadi kemudian dikalikan nilai yang sudah ditentukan. Yang dimana nilai yang ditentukan adalah 1.3. Nilai tersebut dapat diubah sesuai apa yang kita inginkan untuk membuat contrast gambar lebih tinggi ataupun lebih rendah.

Kemudian jika di jalankan atau di RUN maka hasilnya akan di tampilkan pada figure baru dengan menggunakan fungsi `figure()` dan fungsi `imshow(foto_brightness)` untuk menampilkan hasil dari matriks pada variable tersebut seperti pada gambar di bawah ini.



❖ Pengaturan 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  
foto_brightness = foto_gray_const + 20;  
figure(5); imshow(foto_brightness)
```

Seperti yang kita lihat pada baris pertama kita membuat sebuah variable baru lagi yang dinamakan variable brightness yang berisi nilai matriks pada variable foto_gray_const yang telah kita buat sebelumnya dan ditambahkan nilai yang sudah ditentukan, disini nilai yang ditentukannya adalah 20. Nilai tersebut dapat ditambahkan sesuai apa yang kita inginkan jika ingin membuat brightness atau kecahayaan gambar lebih terang ataupun lebih gelap.

Kemudian jika dijalankan maka hasilnya kita tampilkan pada figure baru lagi dengan menggunakan fungsi figure() dan fungsi imshow(foto_contrast) untuk menampilkan hasil dari matriks pada variable brightness. Contohnya dapat kita lihat pada gambar berikut

