PRAKTIKUM 3

IMAGE ENHANCEMENT DENGAN PENGOLAHAN TITIK

Materi

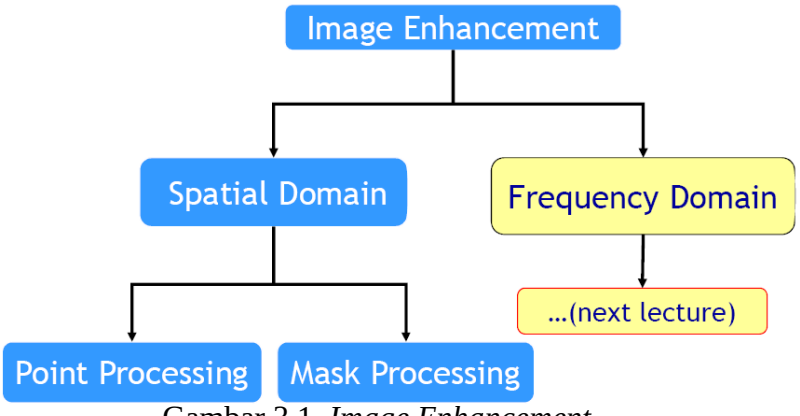
- Image Enhancement
- Pengolahan Titik (*Point Processing*)
- Arithmatic Processing
- Histogram Equalization

Tujuan Praktikum

- Mahasiswa dapat melakukan *enhancement* pada citra dengan *point processing* terutama dengan *arithmetic processing*.

A. Penyajian

Image enhancement adalah proses peningkatkan kualitas citra agar hasilnya dapat lebih **baik** dari citra awal untuk aplikasi tetentu. Kriteria **baik** tergantung pada aplikasi dan problem secara visual maupun secara otomatis (untuk aplikasi selanjutnya). Terdapat dua kategori pada image enhancement yang dapat dilihat pada Gambar 3.1.



Gambar 3.1. *Image Enhancement*

Point Processing

Point processing merupakan operasi yang paling sederhana, namun juga merupakan operasi pengolahan citra yang paling sering digunakan. *Point processing* dilakukan untuk mengubah nilai piksel tertentu yang hanya melibatkan satu piksel saja (tidak menggunakan jendela ketetanggaan). Beberapa contoh *point processing*:

- Arithmetic Processing,
- Logical Processing,
- Histogram Equalization,
- Histogram Specification,
- Contrast Stretching and,
- intensity Transformation.

Arithmetic Processing

Brightness dan kontras citra dapat dikendalikan oleh operasi aritmatika. Operasi aritmatika meliputi penambahan, pengurangan, perkalian, dan membagi nilai-nilai pixel dengan beberapa nilai konstan. Fungsi aritmatika pada nilai-nilai pixel RGB adalah:

- **Addition of a constant:** Increases Brightness
- **Subtraction of a constant**: Decreases Brightness
- **Multiplication by a constant:** Increases contrast
- **Division by a constant:** Decreases contrast

Pada OpenCV dikenal beberapa fungsi aritmatik sebagai berikut:

1. Add

```
void add(
     Mat src1,
     Mat src2,
     Mat dst
);
```

Atau

Mat dst = src1 + src2;

Menambahkan citra dengan konstanta (citra 1 channel)

Mat dst = src1 + value;

Menambahkan citra dengan konstanta (citra 3 channel)

Mat dst = src1 + Scalar(value1,value2,value3)

2. Subtraction

```
void subtract(
     Mat src1,
     Mat src2,
     Mat dst
);
```

Atau

Mat dst = src1 – src2;

Mengurangi citra dengan konstanta (citra 1 channel)

Mat dst = src1 – value;

Mengurangi citra dengan konstanta (citra 3 channel)

Mat dst = src1 – Scalar(value1,value2,value3);

3. Multiply

```
Mat::mul(InputArray src, double scale=1);
```

Mengalikan dengan konstanta

Mat dst = src1.mul(scale);

Atau

Mat dst = src1*scale;

Mengalikan dengan citra lain

Mat dst = src1.mul(src2);

Mengalikan dengan citra lain dan menggunakan konstanta

Mat dst = src1.mul(src2, scale);

4. Division

```
void divide(
     Mat src1,
     Mat src2,
     Mat dst,
     double scale = 1
);
```

Membagi citra dengan konstanta

Mat dst = src1/scale;

Membagi citra dengan citra lain

Mat dst = src1/src2;

Pada fungsi-fungsi aritmatik tersebut semua array harus memiliki tipe dan ukuran yang sama.

B. LATIHAN

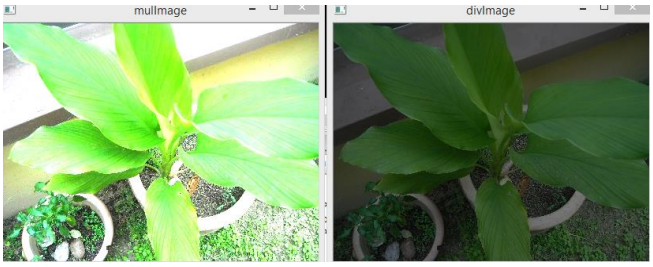
1. Membuat program penambahan dan pengurangan setiap nilai piksel pada citra dengan sebuah konstanta c dan program pengkalian dan pembagian setiap nilai piksel pada citra dengan sebuah konstanta s.

aritmatik.cpp

```
#include <cv.h>
#include <highgui.h>

using namespace cv;

int main()
```

```
{
	Mat original = imread("kunyit.jpg");
	int c = 50, s = 2;
	//proses penambahan
	Mat addIm = original + Scalar(c,c,c);

	//proses pengurangan
	Mat subIm = original – Scalar(c,c,c);

	//proses pengkalian
	Mat mulIm = original * s;

	//proses pembagian
	Mat divIm = original / s;

	//menampilkan citra
	imshow("original", original);
	imshow("addImage", addIm);
	imshow("subImage", subIm);
	imshow("mulImage", mulIm);
	imshow("divImage", divIm);

	waitKey(0);
	return 0;
}
```

Output aritmatik.cpp



2. Menambahkan dua buah citra dengan mengunakan operasi aritmatika.

addImage.cpp

```
#include <cv.h>
#include <highgui.h>
using namespace cv;

int main(){
	//baca citra
	Mat img1 = imread("gambar1.jpg");
	Mat img2 = imread("gambar2.jpg");
	Mat addIm;

	//penambahan dua citra
	addIm = img1 + img2;

	imshow("citra", addIm);
	waitKey(0);

	return 0;
}
```

output addImage.cpp

Nama :

NRP :

Nama Dosen :

Nama Asisten :

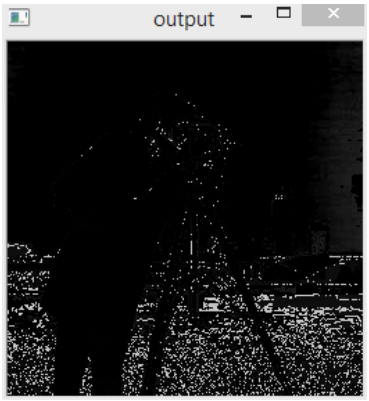
C. **Lembar Kerja Praktikum**

Buatlah sebuah fungsi imageDiff dengan algoritme sebagai berikut.

1. Baca dua citra (citra Cameraman.jpg dan citra Equalized.jpg) yang telah di sediakan di LMS.
2. Ubah dua citra tersebut menjadi *grayscale*
3. Masukkan kedua citra tersebut ke fungsi imageDiff
4. Kemudian hitung nilai rata-rata intensitas setiap citra tersebut
5. Setelah itu lakukan pengubahan tiap nilai intensitas piksel citra dengan ketentuan sebagai berikut:
   - Jika nilai intensitas piksel dibawah rata-rata, kalikan dengan angka 0.5
   - Jika nilai intensitas piksel diatas rata-rata atau sama dengan rata-rata, kalikan dengan angka 2
6. Lakukan proses pengurangan antara citra Cameraman dan citra Equalized
7. Tampilkan hasilnya

Setelah itu berikan penjelasan singkat terhadap citra yang dihasilkan.

Hasil yang diharapkan



Contoh program

```
#include <cv.h>
#include <highgui.h>

using namespace cv;

Mat imageDiff(Mat img1, Mat img2)
{
   Mat output;
   ...
   ...
   return output;
}

int main()
{
   Mat img1, img2, output;
   ...
   ...
   output = imageDiff(img1,img2);
   ...
   ...
   return 0;
}
```