

Nama : Muhammad Fikry  
NIM : 2018104204

```
#include <iostream>
#include "opencv2/highgui.hpp"
#include "opencv2/imgproc.hpp"
#include "opencv2/core.hpp"
#include <fstream>

using namespace cv;
using std::cout;
using std::endl;

Point anchor;
int kernel_size = 11;

int main(int argc, char** argv) {
    Mat src = imread("media/me.jpg");
    resize(src, src, Size(440, 540));
    if (src.empty()) {
        cout << "Gambar tidak ditemukan !" << endl;
        return -1;
    }

    imshow("Original", src);

    Mat homogeneous, gaussian, median, bilateral;
    // Homogeneous Filter
    anchor = Point(-1, 1);
    blur(src, homogeneous, Size(kernel_size, kernel_size), anchor);

    // Gaussian Filter
    GaussianBlur(src, gaussian, Size(kernel_size, kernel_size), 0, 0);

    // Median Filter
    medianBlur(src, median, 11);

    // Bilateral Filter
    // Parameter (src, dest, diameter, sigma_color, sigma_space)
    bilateralFilter(src, bilateral, 25, 25 * 2, 25 / 2);

    imshow("homogenous Filter", homogeneous);
    imshow("Gaussian Filter", gaussian);
    imshow("Median Filter", median);
    imshow("Bilateral Filter", bilateral);

    // SHARPEN

    // Filter 2D
    Mat sharp;
    Mat sharpening_kernel = (Mat_<double>(3, 3) << -1, -1, -1,
        -1, 9, -1,
        -1, -1, -1);
    filter2D(src, sharp, -1, sharpening_kernel);
    imshow("Sharpening Filter", sharp);
```

```

        // addWeighted
        Mat sharp_sub;
        addWeighted(src, 4, gaussian, -4, 0, sharp_sub);
        imshow("Weighted Sharpening Filter", sharp_sub);

        waitKey(0);
        return 0;
    }

```

#### Analisis :

- a. Menambahkan / menyisipkan isi file "Header" ke dalam kode program saat program dicompile dengan menggunakan #include, pada code diatas menggunakan <iostream>, <fstream>, "opencv2/highgui.hpp", "opencv2/core.hpp", dan "opencv2/imgproc.hpp"
- b. Using namespace untuk mendeklarasikan/ memberitahukan kepada compiler bahwa kita akan menggunakan semua fungsi/class/file yang terdapat dalam namespace, pada code diatas menggunakan namespace cv.
- c. std::cout yang memungkinkan mengirim data ke konsol untuk dicetak sebagai teks. cout adalah singkatan dari "character output". std::endl Menyisipkan karakter baris baru ke dalam urutan keluaran os dan membersihkannya seolah-olah dengan memanggil os.
- d. Imread digunakan untuk membaca gambar pada variable Matriks, pada code diatas variabelnya adalah src dan disertai direktori file gambar tersebut.
- e. Resize window menjadi 440x540
- f. Kondisi if untuk memberitahu jika gambar tidak ada didirektori maka akan mengeluarkan output "gambar tidak ditemukan"
- g. Membuat matriks homogeneous, gaussian, median, bilateral, sharp, sharp\_sub
- h. Homogeneous Blur adalah metode paling sederhana untuk menghaluskan gambar. Dalam teknik ini, setiap nilai piksel dihitung sebagai nilai rata-rata dari lingkungan piksel yang ditentukan oleh kernel. Pada code diatas menyeting anchor point bernilai -1,1, lalu memanggil function homogeneous filter yaitu blur dengan keterangan source matriks gambar, matriks tujuan untuk disimpan, size sesuai kernel\_size yang telah dideklarasikan dan nilai anchor.
- i. Gaussian blur adalah teknik penghalusan yang paling umum digunakan untuk menghilangkan noise pada gambar dan video. Dalam teknik ini, gambar harus dibulatkan dengan kernel Gaussian untuk menghasilkan gambar yang dihaluskan. Lalu memanggil function Gaussian filter yaitu GaussianBlur dengan keterangan source matriks gambar, matriks tujuan untuk disimpan, size sesuai kernel\_size yang telah dideklarasikan dan nilai anchor 0,0.
- j. medianBlur mengambil median dari semua piksel di bawah area kernel dan elemen tengah diganti dengan nilai median ini. Efektif melawan noise salt and paper pada gambar. Menariknya, pada filter di atas, elemen sentral adalah nilai yang baru dihitung yang dapat berupa nilai piksel pada gambar atau nilai baru. Namun pada pemburaman median, elemen sentral selalu diganti dengan beberapa nilai piksel pada gambar. Ini mengurangi kebisingan secara efektif. Lalu memanggil function Median filter yaitu medianBlur dengan keterangan source matriks gambar, matriks tujuan untuk disimpan, dan nilai ukuran aperture linier harus ganjil dan lebih besar dari 1.
- k. bilateralFilter sangat efektif dalam menghilangkan noise sekaligus menjaga ketajaman tepi. Tetapi operasinya lebih lambat dibandingkan dengan filter lain. Dimana bahwa filter Gaussian mengambil lingkungan di sekitar piksel dan menemukan rata-rata tertimbang Gaussiannya. Filter Gaussian ini adalah fungsi ruang saja, yaitu piksel

terdekat dipertimbangkan saat memfilter. Lalu memanggil function Bilateral filter yaitu `bilateralFilter` dengan keterangan source matriks gambar, matriks tujuan untuk disimpan, diameter, `sigmaColor` dan `sigmaSpace`.

- l. `filter2D` digunakan untuk melakukan konvolusi. Parameter ketiga (`depth`) diset ke -1 yang berarti kedalaman bit gambar keluaran sama dengan gambar masukan. Jadi jika citra masukan berjenis `CV_8UC3` maka citra keluaran juga akan berjenis sama. Lalu memanggil function `filter2D` yaitu `filter2D` dengan keterangan source matriks gambar, matriks tujuan untuk disimpan, `depth`, matriks rumus `sharpening_kernel` yang telah dideklarasikan diatas.
- m. `addWeighted` adalah untuk melakukan penjumlahan berbobot dari dua gambar dengan bobot lebih banyak ke gambar asli dan bobot lebih sedikit ke gambar yang dihaluskan. Parameter untuk kedua fungsi tersebut adalah sesuatu yang dapat Anda sesuaikan untuk melihat apakah Anda mendapatkan hasil yang lebih baik untuk gambar Anda. Lalu memanggil function `addWeighted` yaitu `addWeighted` dengan keterangan source matriks gambar, nilai `alpha`, source matriks gaussian, nilai `beta`, 0 dan matriks tujuan untuk disimpan.
- n. `Imshow` digunakan untuk menampilkan data matriks menjadi sebuah gambar, digunakan untuk memanggil matriks yang telah berisi sesuai dengan filter.
- o. Fungsi `waitkey(0)` untuk mempertahankan window agar tetap menampilkan gambar.

Output :



