

**Laporan Tugas 2 IF4073 Interpretasi dan Pengolahan Citra**  
**Semester I Tahun 2022/2023**

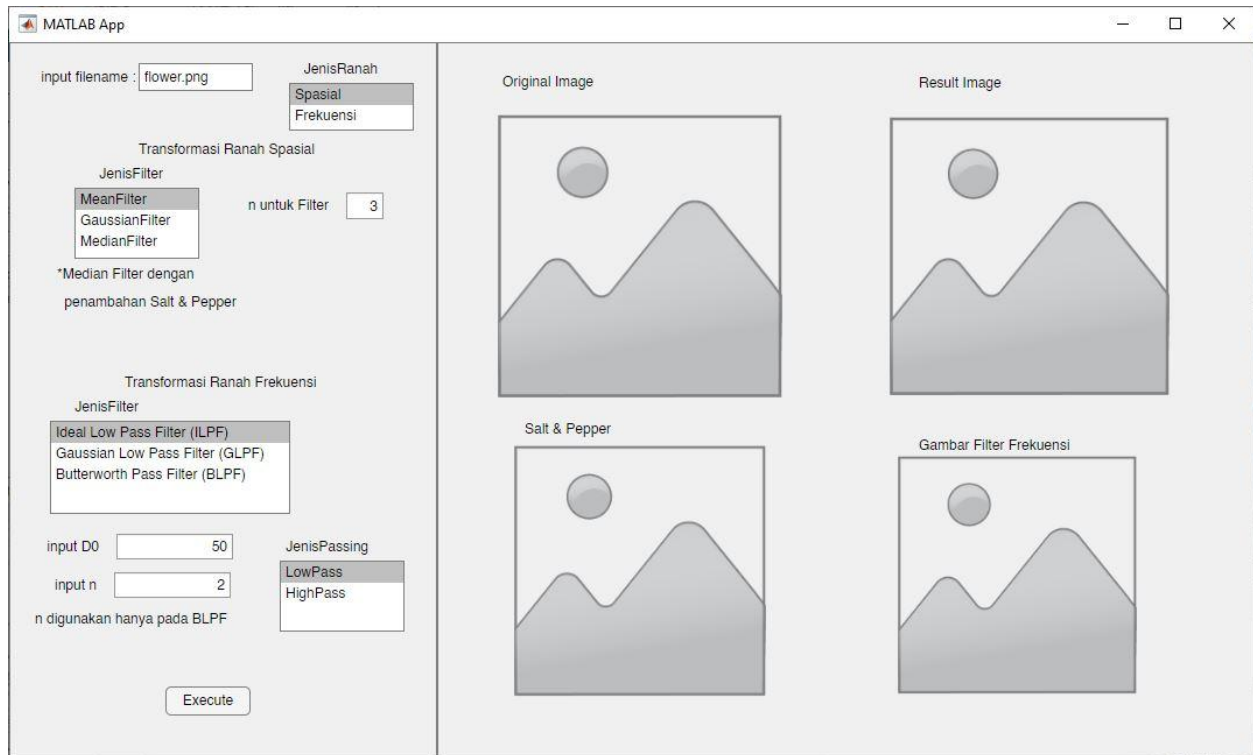


Dibuat oleh :

13519174 - Jusuf Junior Athala

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**  
**INSTITUT TEKNOLOGI BANDUNG**  
**2022**

## A. Screenshot GUI Program



User menjalankan file `app_tucil2.mlapp`. User kemudian dapat memasukkan nama file citra pada field input filename. Kemudian user dapat memilih modus program ranah antara ranah spasial dan frekuensi. Setelah user memilih modus program, user dapat memilih jenis filter, jenis passing, dan menginput konstanta-konstanta yang akan digunakan sebagai input program.

## B. Penjelasan masing-masing program

### 1. Program 1 : Melakukan konvolusi

#### a. Kode program

```
function num = convolu(mtxCut,mask)
if (ndims(mtxCut)==ndims(mask))
    num = 0;
    nMask = length(mask);
    for i=1:nMask
        for j=1:nMask
            num = num + mtxCut(i,j)*mask(i,j);
        end
    end
end
end
```

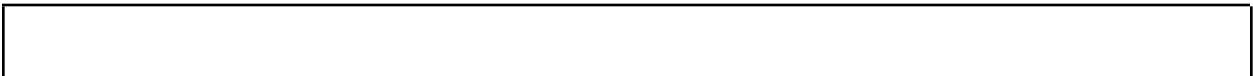
```

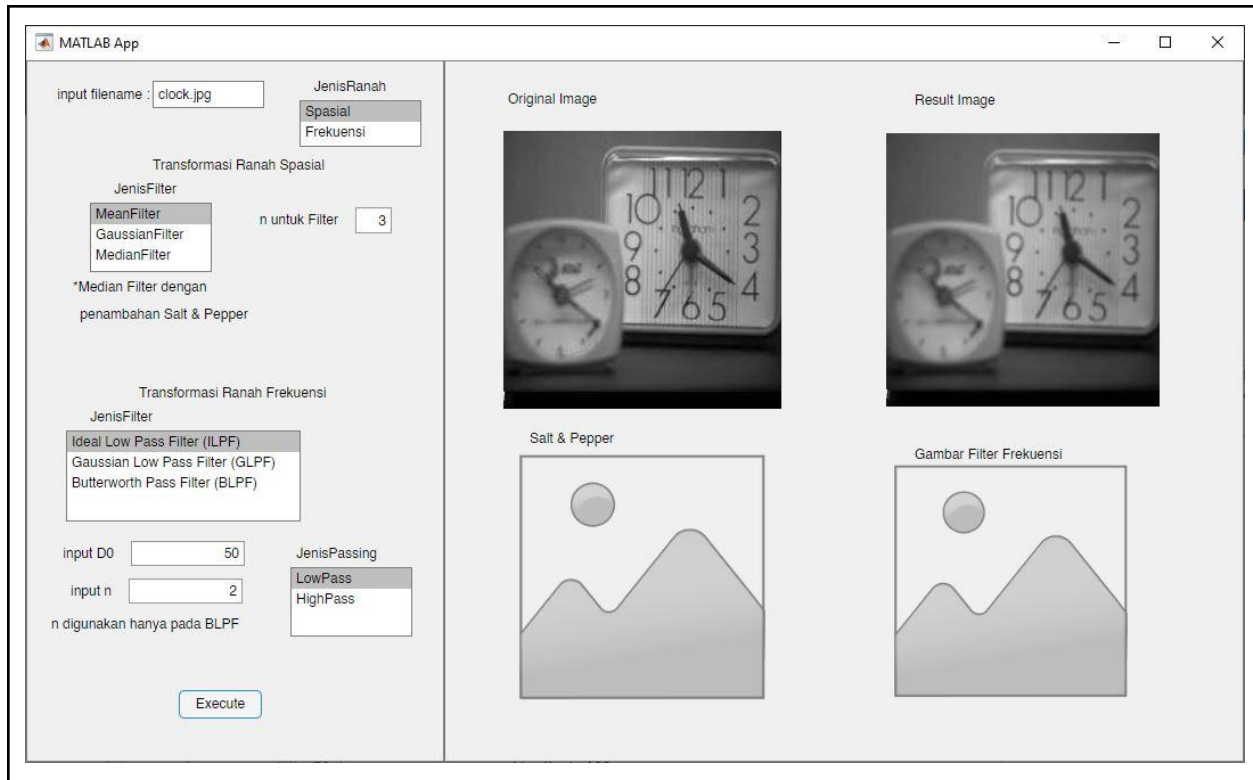
function mtxOut = mtxConvolution(mtxIn, maskIn, jenisFilter)
%mtxOut = mtxIn; mtxOut perlu diinisialisasi dahulu
mtxOut = mtxIn;
[nRow, nCol] = size(mtxIn);
nMask = length(maskIn); %n mask
maskCenter = int32(nMask / 2); % center dari mask
nRowCenter= nRow-(maskCenter);
nColCenter= nCol-(maskCenter);
const = (nMask - 3)/2;
%pixel pinggir tidak akan dikonvolusi
for i=1:(nRowCenter-const)
    for j=1:(nColCenter-const)
        mtxTemp = mtxIn(i:i+nMask-1, j:j+nMask-1);
        if jenisFilter=="Md" %jenis filter median;
            num = median(mtxTemp,'all');
        else %jenis filter mean atau gaussian
            num = convolu(mtxTemp,maskIn);
        end

        %clip jika hasil <0 atau >255
        if (num<0)
            num = 0;
        end
        if (num>255)
            num =255;
        end
        mtxOut(i+maskCenter,j+maskCenter)= num;
    end
end
end

```

b. Hasil eksekusi dengan contoh citra input





### c. Analisis cara kerja fungsi algoritma dan hasilnya

Program akan melakukan konvolusi pada citra masukan.

## 2. Program 2 : Melakukan penapisan ranah spasial dan frekuensi

### a. Kode program

```
% Penapisan dalam ranah frekuensi dengan Ideal Lowpass Filter (ILPF)
function mtxOut = transformFreq(chosenImage, D0, n, jenisFilter, jenisPassing)
% f=imread('cameraman.tif');
% imshow(f);
f= chosenImage;
[M,N] = size(f);
%Step 1: Tentukan parameter padding, biasanya untuk citra f(x,y)
% berukuran M x N, parameter padding P dan Q adalah P = 2M and Q = 2N.
P = 2*M;
Q = 2*N;
%Step 2: Bentuklah citra padding fp(x,y) berukuran P X Q dengan
```

```

% menambahkan pixel-pixel bernilai nol pada f(x, y).
f = im2double(f);
for i = 1:P
    for j = 1:Q
        if i <= M && j<= N
            fp(i,j) = f(i,j);
        else
            fp(i,j) = 0;
        end
    end
end
% imshow(f);title('original image');
% figure; imshow(fp);title('padded image');
% 15% Display the Fourier Spectrum
Fc=fftshift(fft2(fp)); % move the origin of the transform to the center of
% the frequency rectangle
S2=log(1+abs(Fc)); % use abs to compute the magnitude (handling imaginary)
% and use log to brighten display
% figure, imshow(S2,[]); title('Fourier spectrum'); GAMBAR fourier spectrum
%Step 3: Lakukan transformasi Fourier pada fpad(x, y)
F = fft2(double(fp));
%Step 4: Bangkitkan fungsi penapis H berukuran P x Q, misalkan penapis
%yang digunakan adalah Ideal Lowpass Filter (ILPF)
if D0<1
    D0 = 50; % cut-off frequency
end
% Set up range of variables.
u = 0:(P-1);
v = 0:(Q-1);
% Compute the indices for use in meshgrid
idx = find(u > P/2);
u(idx) = u(idx) - P;
idy = find(v > Q/2);
v(idy) = v(idy) - Q;
% 16% Compute the meshgrid arrays
[V, U] = meshgrid(v, u);
D = sqrt(U.^2 + V.^2);
H = 0.0;
% title_name = "";
if n<1
    n = 2; % default untuk Butterworth Low Pass Filter
end
if jenisFilter == "I"
    H = double(D <=D0);
    % title_name='LPF Ideal Mask';
elseif jenisFilter == "G"
    H = exp(-(D.^2)./(2*(D0^2)));
    % title_name='Gaussian Low Pass Filter';
elseif jenisFilter == "B"
    H = 1./(1 + (D./D0).^(2*n));
    % title_name='Butterworth Low Pass Filter';
end
if jenisPassing=="H" %L = low pass, H = high pass
    H = 1 -H ;

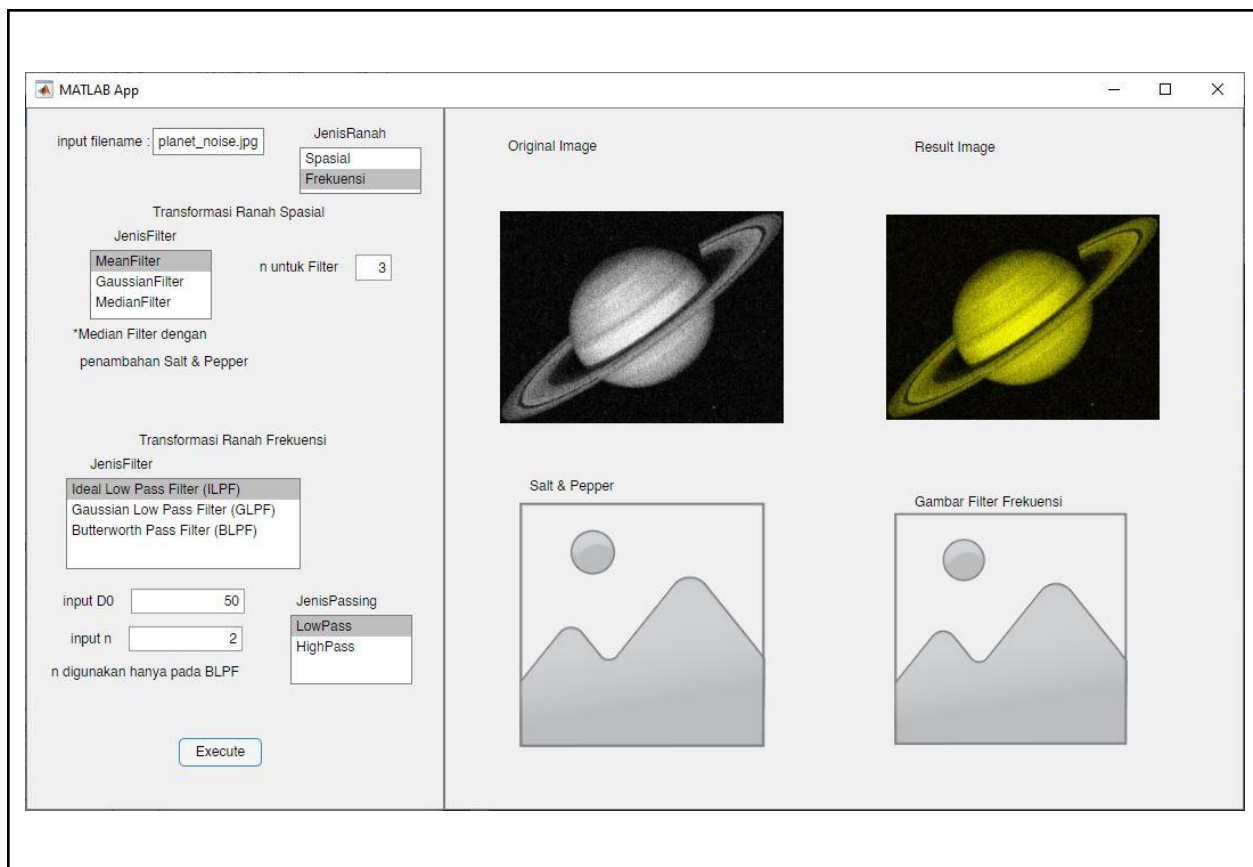
```

```

end
H = fftshift(H);
% figure;imshow(H);title(title_name);
% figure, mesh(H);
%Step 5: Kalikan F dengan H
H = ifftshift(H);
LPF_f = H.*F;
%Step 6: Ambil bagian real dari inverse FFT of G:
LPF_f2=real(ifft2(LPF_f)); % apply the inverse, discrete Fourier transform
% figure; imshow(LPF_f2);title('output image after inverse 2D DFT');
%Step 7: Potong bagian kiri atas sehingga menjadi berukuran citra semula
LPF_f2=LPF_f2(1:M, 1:N); % Resize the image to undo padding
% figure, imshow(LPF_f2); title('output image'); GAMBAR output
mtxOut = LPF_f2;
end

```

## b. Hasil eksekusi dengan contoh citra input



## c. Analisis cara kerja fungsi algoritma dan hasilnya

Program mengalami kegagalan dalam melakukan perubahan dalam ranah

frekuensi dengan low pass filter.

### 3. Program 3: Melakukan perubahan pada frekuensi dengan high pass filter

#### a. Kode program

```
% Penapisan dalam ranah frekuensi dengan Ideal Lowpass Filter (ILPF)
function mtXOut = transformFreq(chosenImage, D0, n, jenisFilter, jenisPassing)
% f=imread('cameraman.tif');
% imshow(f);
f= chosenImage;
[M,N] = size(f);
%Step 1: Tentukan parameter padding, biasanya untuk citra f(x,y)
% berukuran M x N, parameter padding P dan Q adalah P = 2M and Q = 2N.
P = 2*M;
Q = 2*N;
%Step 2: Bentuklah citra padding fp(x,y) berukuran P X Q dengan
% menambahkan pixel-pixel bernilai nol pada f(x, y).
f = im2double(f);
for i = 1:P
    for j = 1:Q
        if i <= M && j<= N
            fp(i,j) = f(i,j);
        else
            fp(i,j) = 0;
        end
    end
end
% imshow(f);title('original image');
% figure; imshow(fp);title('padded image');
% 15% Display the Fourier Spectrum
Fc=fftshift(fft2(fp)); % move the origin of the transform to the center of
% the frequency rectangle
S2=log(1+abs(Fc)); % use abs to compute the magnitude (handling imaginary)
% and use log to brighten display
% figure, imshow(S2,[]); title('Fourier spectrum'); GAMBAR fourier spectrum
%Step 3: Lakukan transformasi Fourier pada fpad(x, y)
F = fft2(double(fp));
%Step 4: Bangkitkan fungsi penapis H berukuran P x Q, misalkan penapis
% yang digunakan adalah Ideal Lowpass Filter (ILPF)
if D0<1
    D0 = 50; % cut-off frequency
end
% Set up range of variables.
u = 0:(P-1);
v = 0:(Q-1);
% Compute the indices for use in meshgrid
idx = find(u > P/2);
u(idx) = u(idx) - P;
idy = find(v > Q/2);
```

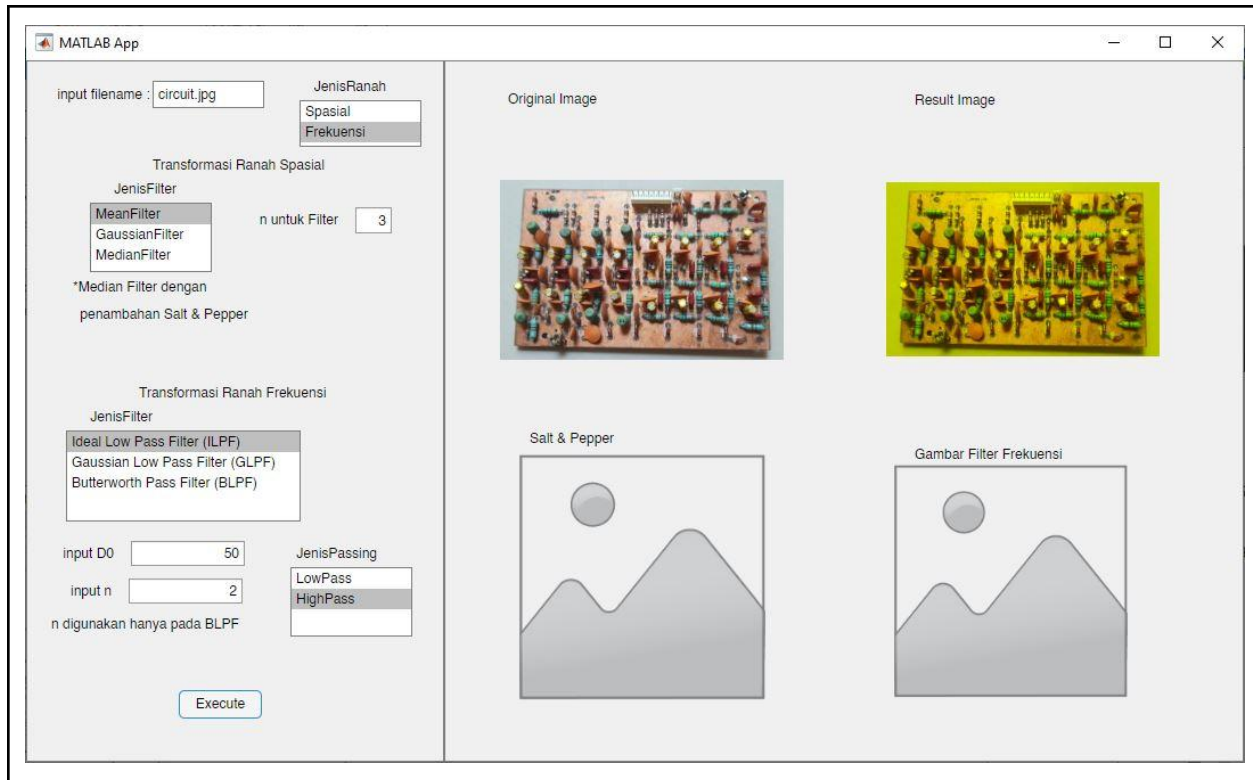
```

v(idy) = v(idy) - Q;
% 16% Compute the meshgrid arrays
[V, U] = meshgrid(v, u);
D = sqrt(U.^2 + V.^2);
H = 0.0;
% title_name = "";
if n<1
    n = 2; % default untuk Butterworth Low Pass Filter
end
if jenisFilter == "I"
    H = double(D <= D0);
%     title_name='LPF Ideal Mask';
elseif jenisFilter == "G"
    H = exp(-(D.^2)./(2*(D0^2)));
%     title_name='Gaussian Low Pass Filter';
elseif jenisFilter == "B"
    H = 1./(1 + (D./D0).^(2*n));
%     title_name='Butterworth Low Pass Filter';
end
if jenisPassing=="H" %L = low pass, H = high pass
    H = 1 - H ;
end
H = fftshift(H);
% figure;imshow(H);title(title_name);
% figure, mesh(H);
%Step 5: Kalikan F dengan H
H = ifftshift(H);
LPF_f = H.*F;
%Step 6: Ambil bagian real dari inverse FFT of G:
LPF_f2=real(ifft2(LPF_f)); % apply the inverse, discrete Fourier transform
% figure; imshow(LPF_f2);title('output image after inverse 2D DFT');
%Step 7: Potong bagian kiri atas sehingga menjadi berukuran citra semula
LPF_f2=LPF_f2(1:M, 1:N); % Resize the image to undo padding
% figure, imshow(LPF_f2); title('output image'); GAMBAR output
mtxOut = LPF_f2;
end

```

b. Hasil eksekusi dengan contoh citra input





### c. Analisis cara kerja fungsi algoritma dan hasilnya

Untuk high pass filter sama dengan low pass filter dengan sedikit perubahan yaitu  $H = 1 - H$ . Namun program juga mengalami kegagalan dalam melakukan high pass filter

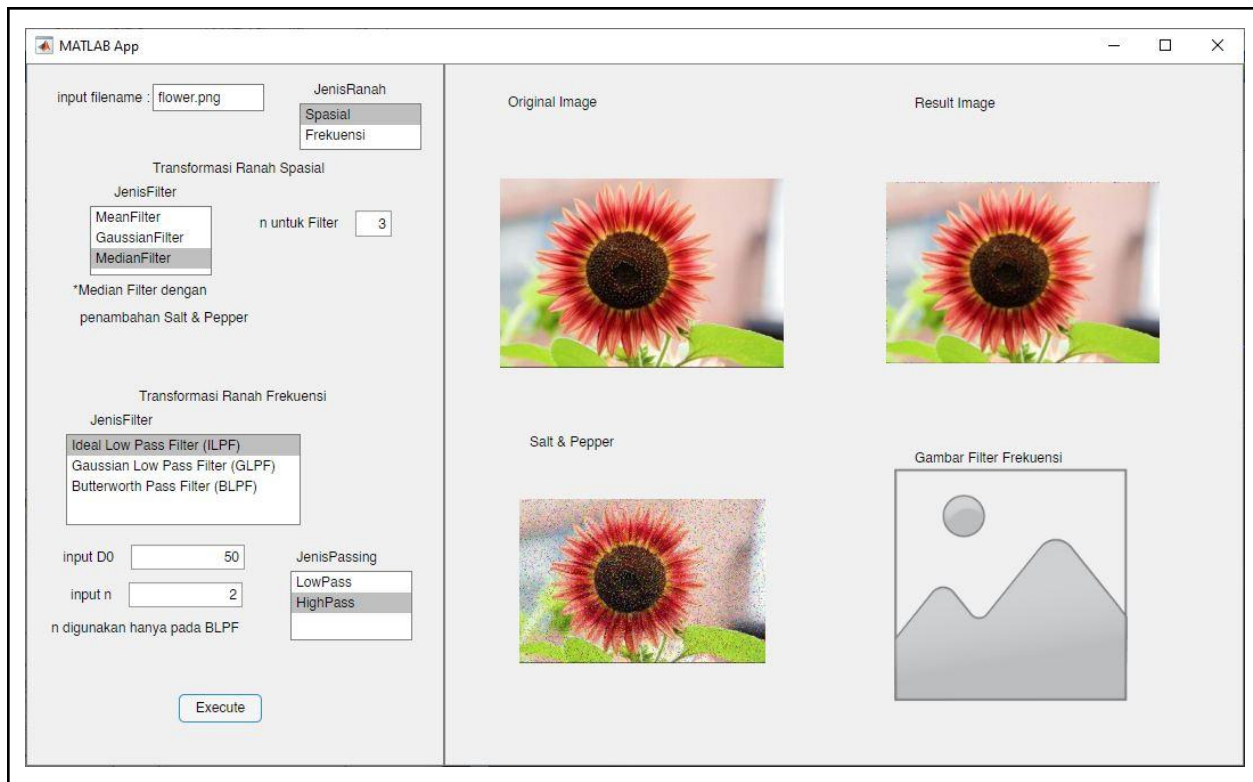
## 4. Program 5 : Menambahkan noise pada citra dan melakukan median filter

### a. Kode program

```
mtxTemp = mtxIn(i:i+nMask-1, j:j+nMask-1);
if jenisFilter=="Md" %jenis filter median;
    num = median(mtxTemp,'all');
% akan menghasilkan nilai median dari mtxTemp yaitu matrix berukuran n

if jenisFilter == "Md" %untuk median filter
    % diberi saltpepper terlebih dahulu
    imgOri = imnoise(imgOri, 'salt & pepper', 0.1);
    app.Image2.ImageSource = imgOri;
end
```

b. Hasil eksekusi dengan contoh citra input



c. Analisis cara kerja fungsi algoritma dan hasilnya

Program akan memberi derau pada gambar dan akan mengubah nilai pixel menjadi nilai median dari  $n \times n$  array potongan dari citra asli.

Beberapa program seperti program nomor 4 dan nomor 6 tidak sempat dikerjakan.

C. Alamat Github program

[jusufjathala/citra-tucil2 \(github.com\)](https://github.com/jusufjathala/citra-tucil2)