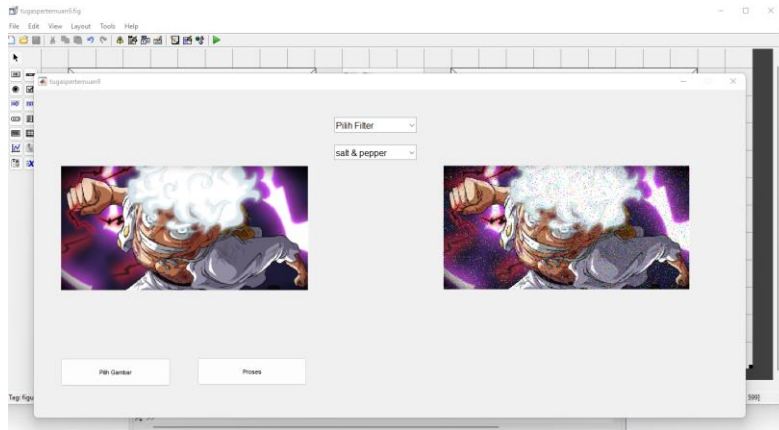


**Nama : Muhammad Umar Mansyur**

**NIM : 2020520018**

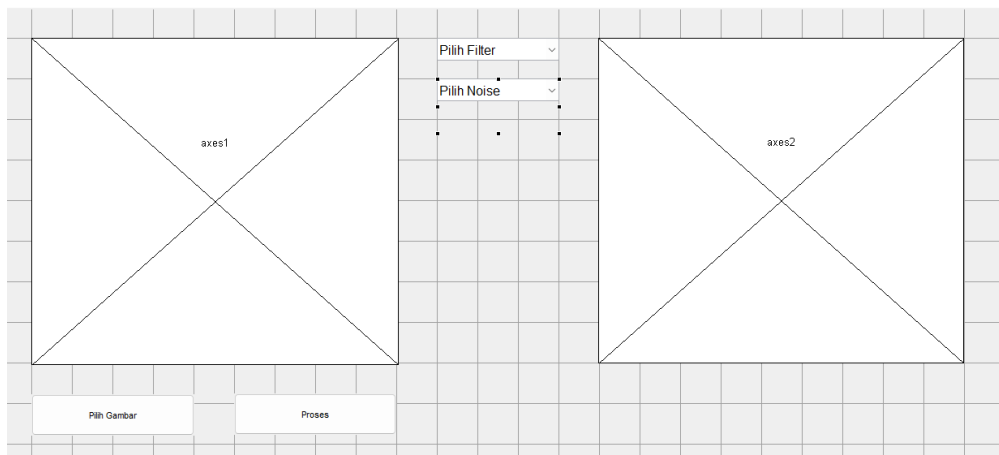
**Kelas : B**

### **Spatial Filtering Matlab**



*Gambar 1 Program Spatial Filtering Matlab*

Spatial Filtering adalah proses penyaringan atau filtering langsung terhadap pixel. Teknik spatial filtering biasanya digunakan untuk menghilangkan noise pada gambar. Di matlab ada 8 metode filtering. Kedelapan metode tersebut antara lain, average, disk, gaussian, Laplacian, log, motion, prewitt, sobel dan median filter. Sedangkan untuk noise, di matlab juga tersedia sebanyak 5 function. Kelima fungsi tersebut adalah gaussian, localvar, poissond, salt & pepper dan speckle. Untuk membuat program spatial filtering di matlab, terlebih dahulu perhatikan interface berikut:



*Gambar 2 Interface Program Spatial Filtering Matlab*

Pada gambar 2, terdapat 2 axes, 2 tombol (Pilih Gambar, dan Proses), 2 popupmenu (pilih noise dan pilih filter). Axes1 digunakan untuk menampilkan gambar yang terpilih dan Axes 2 digunakan untuk menampilkan hasil dari filtering noise. Hasil akan ditampilkan pada axes 2 setelah menekan tombol proses. Adapun program dari interface pada gambar 2, sebagaimana berikut:

```
% --- Executes on button press in browse.  
function browse_Callback(hObject, eventdata, handles)  
% hObject    handle to browse (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
[file, path] =uigetfile('*.jpg;*.png;*.jpeg;');  
image = imread(fullfile(path, file));  
handles.object = image;  
guidata(hObject, handles);  
axes(handles.axes1);  
imshow(image);
```

Gambar 3 Program pada Tombol Pilih Gambar

Pada tombol Pilih Gambar, gunakan fungsi `uigetfile` untuk menangkap lokasi dan nama file. Karena file yang mau digunakan hanya berupa gambar maka berikan validasi dengan ekstensi gambar seperti `*.jpg` dan dipisah dengan titikkoma dalam tanda petik yang sama. Fungsi `uigetfile` mengembalikan dua nilai, yaitu nama file dan lokasi file. Dari dua nilai yang kembalikan akan disimpan pada variable *image*. Agar semua gambar dapat dimodifikasi maka dibuatkan varibel global yaitu *handles.object*. dimana *handles.object* merupakan variable global untuk menampung gambar(variable *image*). Karena Variable global di matlab membutuhkan update struktur dimatlab dengan menggunakan perintah `guidata(hObject, handles)`. Kemudian tampilkan di axes 1 dengan perintah `axes(handles.axes1)` dan `imshow(image)`.

Selanjutnya, pada bagian inti yaitu proses. Di dalam proses, pilihan filter dan noise yang dipilih oleh user akan diproses. Kode programnya adalah sebagaiberikut:

```
% --- Executes on button press in proccess.
function proccess_Callback(hObject, eventdata, handles)
% hObject    handle to proccess (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
options = handles.noise.String;
selected = handles.noise.Value;
selectedNoise = options{selected};

filterItem = handles.filter.String;
choicest = handles.filter.Value;
choicest = filterItem{choicest};
image = handles.object;
```

```

if (string(selectedNoise) == 'Pilih Noise')
    switch (string(choicest))
        case 'Pilih Filter'
            axes(handles.axes2);
            imshow(image);
        case 'median filter'
            red = image(:,:,1);
            green = image(:,:,2);
            blue = image(:,:,3);
            getRed = medfilt2(red);
            getGreen = medfilt2(green);
            getBlue = medfilt2(blue);
            image = cat(3, getRed, getGreen, getBlue);
            axes(handles.axes2);
            imshow(image);
        otherwise
            filter = fspecial(string(choicest));
            axes(handles.axes2);
            imshow(imfilter(image,filter));
    end

elseif (string(selectedNoise) == 'localvar')
    image = imnoise(image, 'localvar',0.05*rand(size(image)));
    switch (string(choicest))
        case 'Pilih Filter'
            axes(handles.axes2);
            imshow(image);
        case 'median filter'
            image = imnoise(image, 'localvar',0.05*rand(size(image)));
            red = image(:,:,1);
            green = image(:,:,2);
            blue = image(:,:,3);
            getRed = medfilt2(red);
            getGreen = medfilt2(green);
            getBlue = medfilt2(blue);
            image = cat(3, getRed, getGreen, getBlue);
            axes(handles.axes2);
            imshow(image);
        otherwise
            filter = fspecial(string(choicest));
            axes(handles.axes2);
            imshow(imfilter(image,filter));
    end
end

else
    switch (string(choicest))
        case 'Pilih Filter'
            gambar = imnoise(image, selectedNoise);
            axes(handles.axes2);
            imshow(gambar);
        case 'median filter'
            image = imnoise(image,selectedNoise);
            red = image(:,:,1);
            green = image(:,:,2);
            blue = image(:,:,3);
            getRed = medfilt2(red);
            getGreen = medfilt2(green);
            getBlue = medfilt2(blue);
            image = cat(3, getRed, getGreen, getBlue);
            axes(handles.axes2);
            imshow(image);
        otherwise
            image = imnoise(image, selectedNoise);
            filter = fspecial(string(choicest));
            axes(handles.axes2);
            imshow(imfilter(image,filter));
    end
end

end

```

Kode ini ditulis pada tombol proses, pertama ambil dulu pilihan filter dan noise. Untuk mengambil pilihan perhatikan variable selected dan choicest. Selected noise menyimpan pilihan noise dan choicest untuk menyimpan pilihan filter. Variable image digunakan untuk menyimpan gambar. Perhatikan gambar di bawah ini

```
% --- Executes on button press in process.
function process_Callback(hObject, eventdata, handles)
% hObject      handle to process (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
options = handles.noise.String;
selected = handles.noise.Value;
selectedNoise = options{selected};

filterItem = handles.filter.String;
choicest = handles.filter.Value;
choicest = filterItem{choicest};
image = handles.object;
```

*Gambar 4 Get Selected Item*

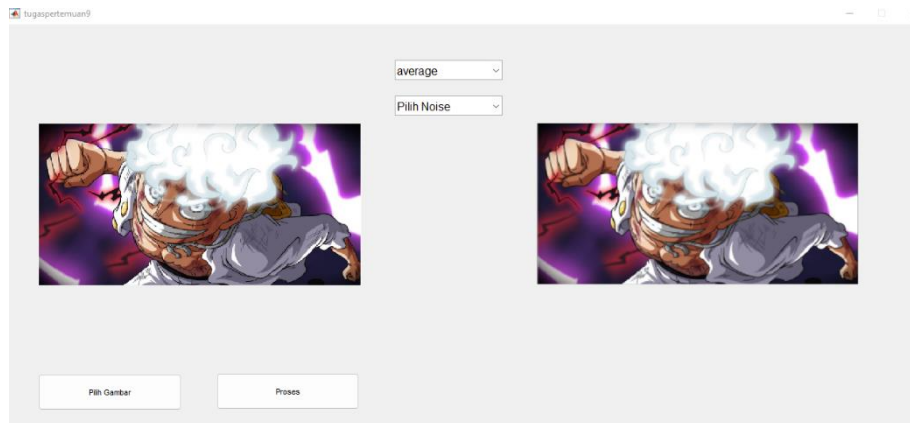
Pilihan yang sudah tersimpan dilakukan pengecekan dan diberikan pengkondisian. Pengkondisian ini dimaksudkan untuk menerapkan filter dan noise berbeda pada saat user memilih dan memproses gambar. Pengecekan pertama dilakukan untuk mengecek pilihan noise, kemudian mengecek nilai filter. Jika tidak memilih maka yang akan ditampilkan adalah gambar semula. Namun jika memilih filter maka akan ditampilkan sebagaimana fitur dari filter yang telah disediakan.

Hanya saja ada beberapa yang perlu diperhatikan yaitu Ketika case filter berada di median filter datersebut gambar akan dipecah terlebih dahulu dan diberikan fungsi medfilt2 pada setiap layernya. Kemudian pecahan layer tersebut akan disatukan dengan menggunakan perintah cat dan ditampilkan pada axes2.

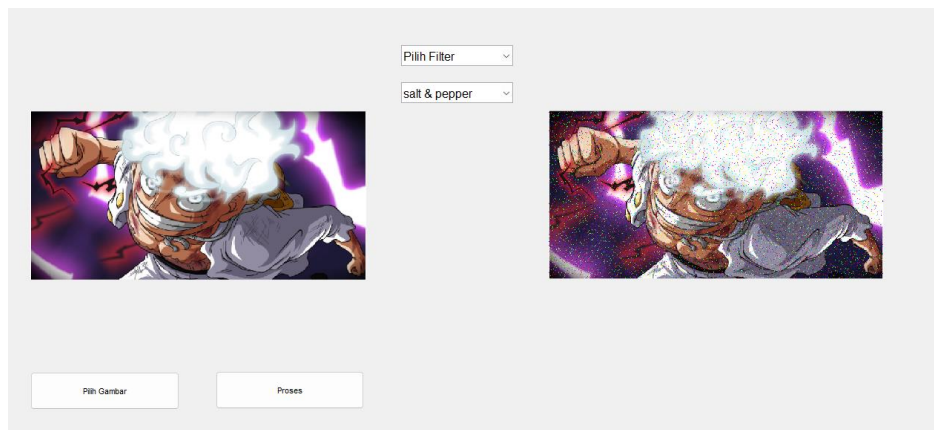
Ketika user mengisi filter dan noise maka gambar akan dinoisekan dulu baru diberikan filter. Perhatikan kode berikut:

```
switch(string(choicest))
    case 'Pilih Filter'
        gambar = imnoise(image, selectedNoise);
        axes(handles.axes2);
        imshow(gambar);
    case 'median filter'
        image = imnoise(image, selectedNoise);
        red = image(:,:,1);
        green = image(:,:,2);
        blue = image(:,:,3);
        getRed = medfilt2(red);
        getGreen = medfilt2(green);
        getBlue = medfilt2(blue);
        image = cat(3, getRed, getGreen, getBlue);
        axes(handles.axes2);
        imshow(image);
    otherwise
        image = imnoise(image, selectedNoise);
        filter = fspecial(string(choicest));
        axes(handles.axes2);
        imshow(imfilter(image, filter));
end
```

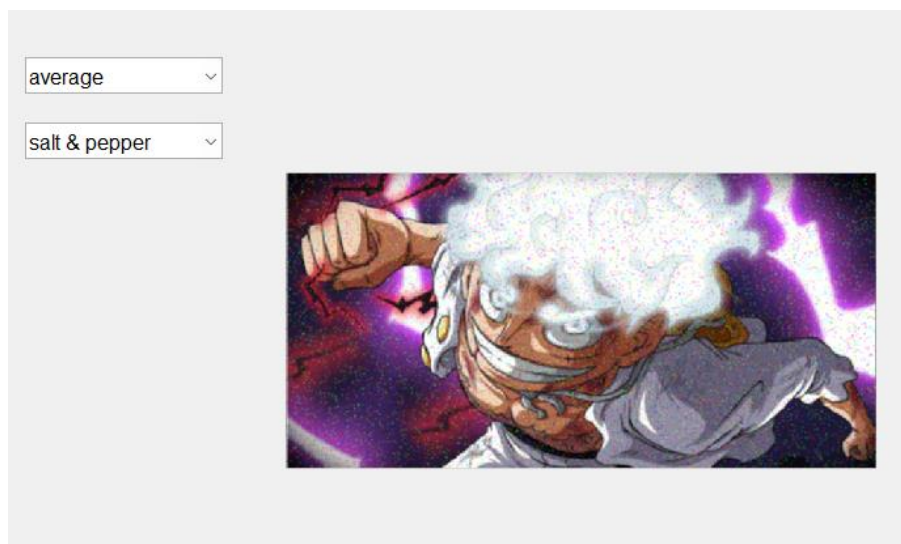
Pada otherwise gambarkan diberikan noise dengan menggunakan perintah `imnoise` dengan parameter gambar dan pilihan noise yang dipilih user. Kemudian untuk menyimpan filter dan akan menggunakannya maka digunakan fungsi `fspecial` dengan parameter pilihan user. Langkah terakhir yaitu menampilkan gambar namun diperintah `imshow` dikombinasikan langsung dengan menggunakan perintah `imfilter`. `Imfilter` memiliki parameter gambar dan filter yang ingin digunakan. Namun hal tersebut tidak berlaku pada median filter. Karena median filter memecah gambar noisenya langsung dan langsung diberikan fungsi `medfilt` pada setiap pecahan layernya.



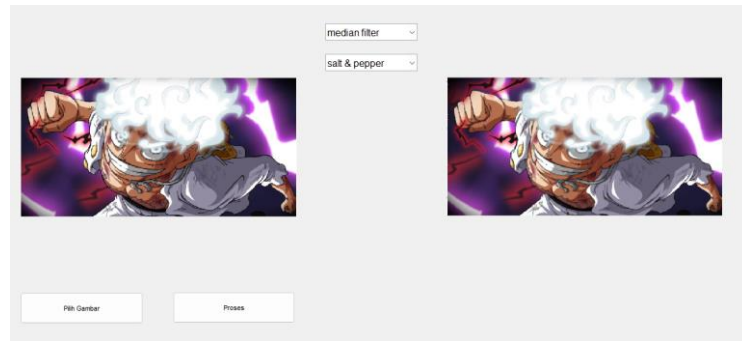
*Gambar 5 mode average tanpa noise*



*Gambar 6 Gambar dengan noise salt & pepper*



*Gambar 7 Gambar dengan filter average dan noise salt & pepper*



*Gambar 8 Metode Median Filter(noise mulai hilang)*

Untuk lebih lengkapnya dapat dijalankan pada program yang akan dilampirkan.