

4

IMAGE FILTERING

Pada pokok bahasan ini, mahasiswa akan mempelajari tentang pemfilteran citra dengan *Low Pass Filter* maupun *High Pass Filter*.

Pokok Bahasan :

1. Smoothing filter
2. Sharphening filter
3. Low Pass Filter
4. High Pass Filter

Latihan :

1. Membuat aplikasi untuk menambah kehalusan citra dengan *smoothing filter*
2. Membuat aplikasi untuk menambah ketajaman citra dengan *sharpening filter*
3. Membuat aplikasi untuk mereduksi noise pada citra dengan *low pass filter*
4. Membuat aplikasi untuk mereduksi noise pada citra dengan *high pass filter*

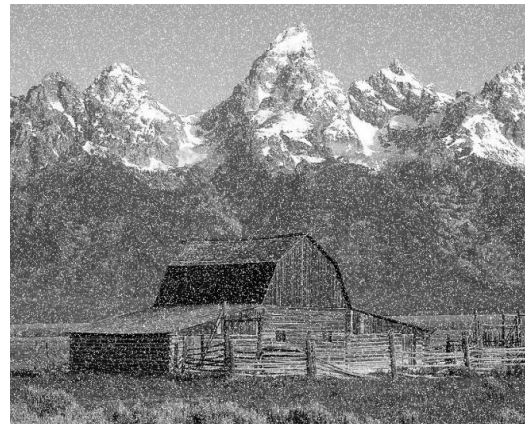
3.1 Smoothing Filter

Capaian pembelajaran : mampu memahami dan mengaplikasikan smoothing filter

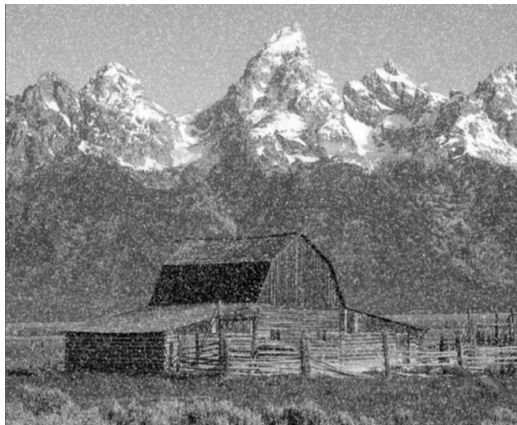
Smoothing filter adalah sejenis low pass filter yang digunakan untuk mengurangi transisi intensitas yang terlalu tajam sehingga intensitas warna menjadi halus (smooth). Fungsi dari smoothing filter adalah : mereduksi noise, mengaburkan edge dan menghaluskan citra.. Gambar 3.1 menunjukkan sebuah citra asli, bernoise dan hasil filter smoothing.



a. Citra original



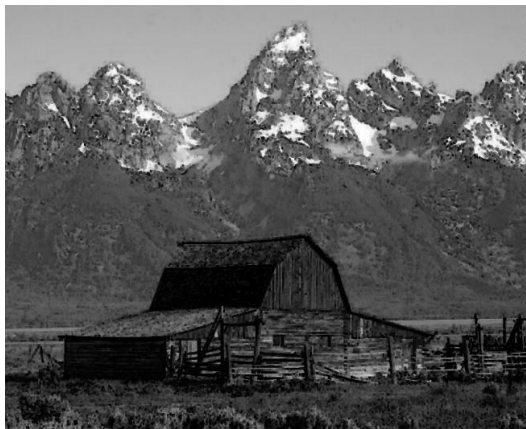
b. Citra bernoise



c. Mean filter



d. Median filter



e. Minimum filter



f. Maksimum filter

Gambar 3.1. Citra hasil pemrosesan smoothing filter

a. Mean Filter

Mean filter adalah filter yang mengambil rata-rata dari sekelompok pixel dalam image. Biasanya sekelompok pixel ini dinamakan Kernel. Untuk kernel 3x3 maka mean filter didefinisikan sebagai :

X1	X2	X3
X4	X5	X6
X7	X8	X9

Dengan X1-X9 adalah nilai intensitas citra.

Maka mean didefinisikan sebagai:

$$X_{mean} = \frac{1}{9} \sum_{i=1}^9 X_i$$

Koordinat kernel itu sendiri didefinisikan sebagai berikut dengan melihat koordinat titik tengah sebagai koordinat (x, y).

(x-1,y-1)	(x,y-1)	(x+1,y-1)
(x-1,y)	(x,y)	(x+1,y)
(x-1,y+1)	(x,y+1)	(x+1,y+1)

Sehingga untuk pembacaan citra di kernel tersebut adalah :

```
Color X1 = Image.GetPixel(x - 1, y - 1);  
Color X2 = Image.GetPixel(x - 1, y);  
Color X3 = Image.GetPixel(x - 1, y + 1);  
Color X4 = Image.GetPixel(x, y - 1);  
Color X5 = Image.GetPixel(x, y);  
Color X6 = Image.GetPixel(x, y + 1);  
Color X7 = Image.GetPixel(x + 1, y - 1);  
Color X8 = Image.GetPixel(x + 1, y);  
Color X9 = Image.GetPixel(x + 1, y + 1);
```

b. Median Filter

Median filter adalah filter yang mengambil nilai tengah dari sekelompok pixel dalam image setelah nilai pixel tersebut diurutkan.

Contoh : median dari kernel di bawah setelah diurutkan adalah 1, 2, 3, 4, 5, 6, 7, 8, 9.

Maka mediannya adalah 5.

3	2	4
5	1	7
9	6	8

c. Minimum Filter

Minimum filter adalah filter yang mengambil nilai minimum dari sekelompok pixel dalam suatu image.

Contoh : nilai minimum dari kernel di bawah adalah 1.

3	2	4
5	1	7
9	6	8

d. Maksimum Filter

Maximum filter adalah filter yang mengambil nilai maksimum dari sekelompok pixel dalam suatu image.

Contoh : nilai maksimum dari kernel di bawah adalah 9.

3	2	4
5	1	7
9	6	8

3.2 Low Pass Filter

Capaian pembelajaran : mampu memahami dan mengaplikasikan low pass filter.

Low Pass filter adalah filter yang meloloskan objek di frekuensi rendah dan mereduksi objek di frekuensi tinggi. Karena noise biasanya berada di frekuensi tinggi, maka low pass filter ini dapat digunakan untuk mereduksi noise. Kernel untuk Low Pass Filter jika dijumlahkan maka akan bernilai 1. Berikut kernel untuk low pass filter di atas:

1/6	0	1	0
	1	2	1
	0	1	0

a. Low pass Filter type 1

1/10	1	1	1
	1	2	1
	1	1	1

b. Low pass Filter type 2

1/9	1	1	1
	1	1	1
	1	1	1

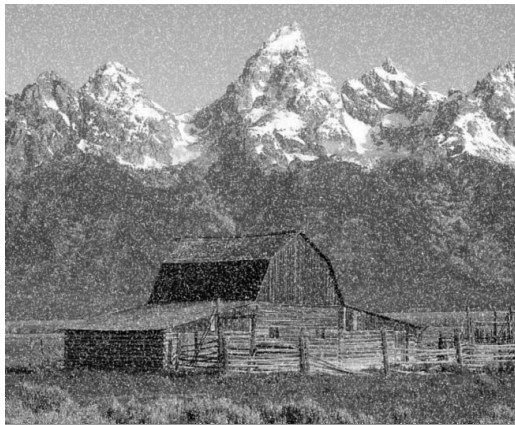
c. Low pass Filter type 3

1/16	1	2	1
	2	4	2
	1	2	1

d. Low pass Filter type 4

Gambar 3.3 Low Pass Filter Kernel

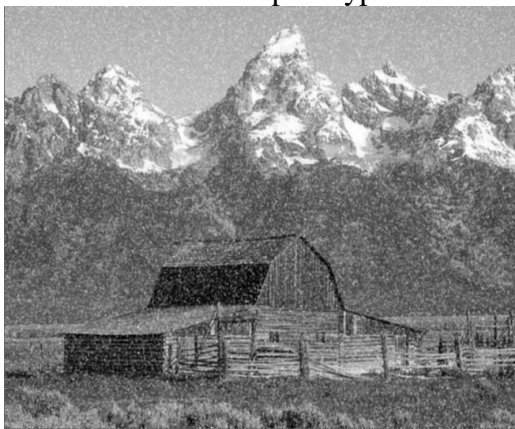
Gambar 3.2 menunjukkan sebuah citra hasil low pas filter.



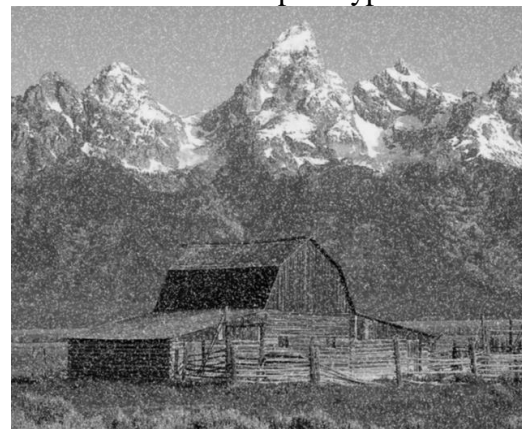
a. Low pass type 1



b. Low pass type 2



c. Low pass type 3



d. Low pass type 4

Gambar 3.2. Citra hasil pemrosesan low pass filter

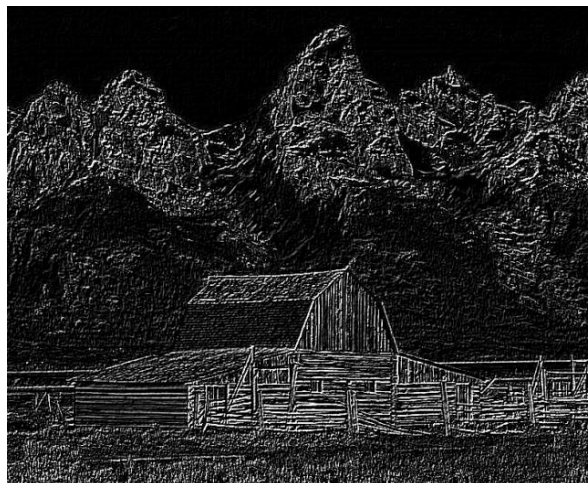
3.3 Sharpening Filter

Capaian pembelajaran : mampu memahami dan mengaplikasikan sharpening filter

Sharpening filter adalah sejenis high pass filter yang digunakan untuk meng-*highlight*/ meng-*enhance* suatu citra yang telah blur sehingga menjadi lebih tajam(*sharp*). Fungsi dari smoothing filter adalah : mendeteksi tepi (*edge*), dan menajamkannya citra.. Gambar 3.3 menunjukkan sebuah citra asli, bernoise dan hasil filter smoothing.



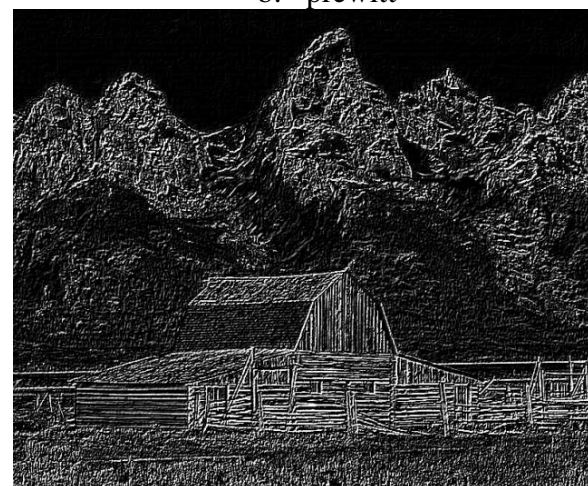
a. robert



b. prewitt



c. sobel



d. laplacian

Gambar 3.3. Citra hasil deteksi tepi/edge

Berikut kernel untuk filter diatas:

-1	1
1	-1

Robert

1	-2	1
-2	4	-2
1	-2	1

Laplacian

-1	-1	-1
0	0	0
1	1	1

Prewitt Horizontal

-1	0	1
-1	0	1
-1	0	1

Prewitt Vertikal

-1	-2	-1
0	0	0
1	2	1

Sobel Horizontal

-1	0	1
-2	0	2
-1	0	1

Sobel Vertikal

3.4 High Pass Filter

Capaian pembelajaran : mampu memahami dan mengaplikasikan high pass filter.

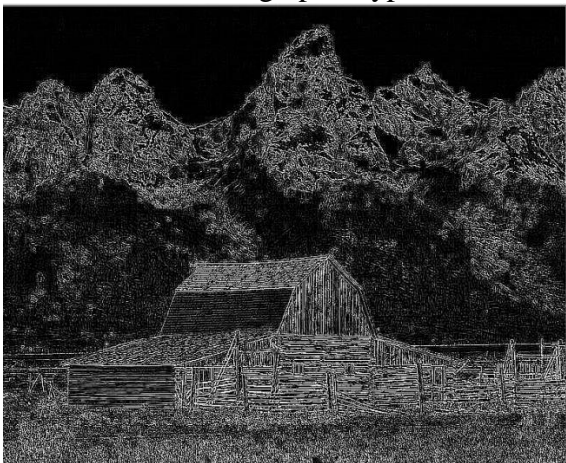
Low Pass filter adalah filter yang meloloskan objek di frekuensi tinggi dan mereduksi objek di frekuensi rendah. Biasaya high pass filter ini dapat digunakan untuk mendeteksi tepian (*edge*). Gambar 3.4 menunjukan sebuah citra hasil low pas filter.



a. High pass type 1



b. High pass type 2



c. Type 3



d. Type 4

Gambar 3.4. Citra Hasil High Pass Filter

Kernel untuk Low Pass Filter jika dijumlahkan maka akan bernilai 0. Berikut kernel untuk high pass filter di atas:

0	1	0
1	-4	1
0	1	0

a. High pass Filter type 1

0	-1	0
-1	4	-1
0	-1	0

b. High pass Filter type 2

1	1	1
1	-8	1
1	1	1

c. High pass Filter type 3

-1	-1	-1
-1	8	-1
-1	-1	-1

d. High pass Filter type 4

3.5 Latihan

Tujuan

Memahami cara membaca file gambar dengan microsoft visual C#

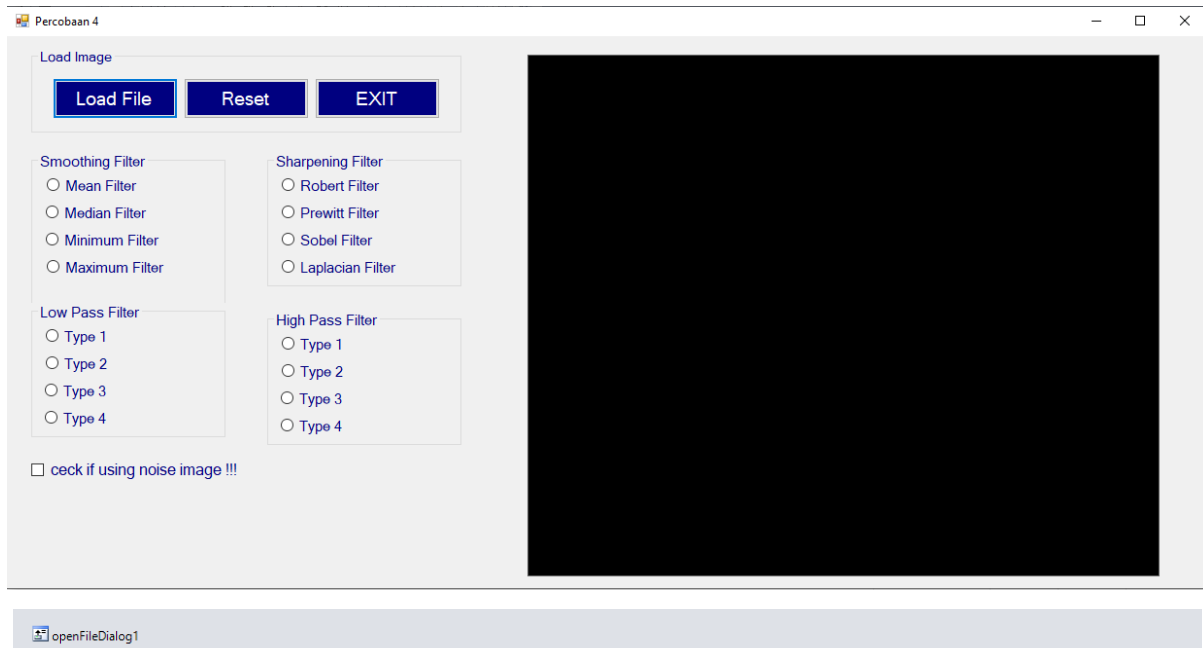
Mampu memanipulasi data citra

Prosedur

1. Tambahkah proyek baru dengan toolbox sebagai berikut :

- 1 buah **pictureBox** dan 1 buah **label** dengan nama : **original Image**
- 1 buah **openFileDialog1**
- 3 buah **button** : **Load File**, **Reset**, dan **Exit**
- 5 buah **groupBox** : **Load Image**, **Smoothing Filter**, **Low Pass Filter**, **Sharpening Filter**, **High Pass Filter**.
- 16 buah **radioButton**
- 1 buah **checkBox**

sehingga tampilannya seperti yang terlihat pada gambar 3.5.



Gambar 3.5. . Tampilan VC# setelah membuat proyek baru

2. Setting properties dari **pictureBox** seperti percobaan 1
3. Setting properties dari **openFileDialog1** seperti percobaan 1
4. Tambahkan global variable sebagai berikut :

```
//global variable
Bitmap sourceImage; //RGB Image
Bitmap grayImage;    //gray image without noise
Bitmap noiseImage;   //gray image with noise

int filterSmoothingType; //1 : meanFilter, 2: MedianFilter, 3:MinimumFilter,
4:MaximumFilter
int filterSharpeningType; //1 : RoberFilter, 2: PrewittFilter, 3 : SobelFilter,
4:LaplacianFilter
int lowPassType;
int highPassType;
```

5. Tambahkan juga fungsi berikut untuk mereset kondisi semua toolbox ke posisi awal

```
private void resetRadioButtonSmoothing()
{
    radioButton1.Checked = false;
    radioButton2.Checked = false;
    radioButton3.Checked = false;
    radioButton4.Checked = false;
}

private void resetRadioButtonLowPass()
{
    radioButton5.Checked = false;
    radioButton10.Checked = false;
    radioButton11.Checked = false;
```

```

        radioButton12.Checked = false;
    }

    private void resetRadioButtonHighPass()
    {
        radioButton13.Checked = false;
        radioButton14.Checked = false;
        radioButton15.Checked = false;
        radioButton16.Checked = false;
    }

    private void resetRadioButtonSharpening()
    {
        radioButton6.Checked = false;
        radioButton7.Checked = false;
        radioButton8.Checked = false;
        radioButton9.Checked = false;
    }

    private void resetCheckBox()
    {
        checkBox1.Checked = false;
    }

```

6. Double klik tombol **Load File**, kemudian tuliskan sub rutin program berikut :

```

private void button1_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        //loading source image
        sourceImage = (Bitmap)Bitmap.FromFile(openFileDialog1.FileName);

        //menkonversi ke gray image
        grayImage = grayImaging(sourceImage);

        //menambahkan noise ke gray image
        noiseImage = noiseImaging(grayImage);

        //original image ditampilkan dalam bentuk gray image
        pictureBox1.Image = grayImage;

        //resetting radio button
        resetRadioButtonSmoothing();
        resetRadioButtonSharpening();
        resetRadioButtonLowPass();
        resetRadioButtonHighPass();
        resetCheckBox();
    }
}

```

7. Tambahkan fungsi berikut untuk melakukan proses gray imaging :

```
private Bitmap grayImaging(Bitmap image)
{
    Bitmap tempImage = new Bitmap(image);
    //grayscale conversion
    for (int x = 0; x < sourceImage.Width; x++)
        for (int y = 0; y < sourceImage.Height; y++)
        {
            Color w = image.GetPixel(x, y);
            int r = w.R; int g = w.G; int b = w.B;
            int xg = (int)((r + g + b) / 3);
            Color wb = Color.FromArgb(xg, xg, xg);

            tempImage.SetPixel(x, y, wb);
        }
    return tempImage;
}
```

8. Tambahkan fungsi berikut untuk menambahkan noise pada gray image :

```
private Bitmap noiseImaging(Bitmap image)
{
    noiseImage = new Bitmap(grayImage);
    int noiseProb = 10;
    Random r = new Random();
    for (int x = 0; x < grayImage.Width; x++)
        for (int y = 0; y < grayImage.Height; y++)
        {
            Color w = image.GetPixel(x, y);
            int xg = w.R;
            int xb = xg;

            //generate random number (0-100)
            int nr = r.Next(0, 100);

            //generationg 20% gaussian noise
            if (nr < noiseProb) xb = 255;
            Color wb = Color.FromArgb(xb, xb, xb);
            noiseImage.SetPixel(x, y, wb);
        }
    return noiseImage;
}
```

9. Tambahkan fungsi berikut untuk melakukan proses **smoothing filter** :

```
//smoothing filter
private Bitmap smoothingfilter(int filterType)
{
    Bitmap filteredImage = new Bitmap (noiseImage);

    int[] xt = new int[10];
    int xb = 0;

    for (int x = 1; x < noiseImage.Width - 1; x++)
        for (int y = 1; y < noiseImage.Height - 1; y++)
        {
            Color w1 = noiseImage.GetPixel(x - 1, y - 1);
            Color w2 = noiseImage.GetPixel(x - 1, y);
            Color w3 = noiseImage.GetPixel(x - 1, y + 1);
```

```

Color w4 = noiseImage.GetPixel(x, y - 1);
Color w5 = noiseImage.GetPixel(x, y);
Color w6 = noiseImage.GetPixel(x, y + 1);
Color w7 = noiseImage.GetPixel(x + 1, y - 1);
Color w8 = noiseImage.GetPixel(x + 1, y);
Color w9 = noiseImage.GetPixel(x + 1, y + 1);

xt[1] = w1.R; xt[2] = w2.R; xt[3] = w3.R;
xt[4] = w4.R; xt[5] = w5.R; xt[6] = w6.R;
xt[7] = w7.R; xt[8] = w8.R; xt[9] = w9.R;

if (filterType == 1) //mean filter
{
    //tambahkan coding
}
else if (filterType == 2) //median filter
{
    //looking for median
    for (int i = 1; i < 9; i++)
        for (int j = 1; j < 9; j++)
        {
            if (xt[j] > xt[j + 1])
            {
                int a = xt[j];
                xt[j] = xt[j + 1];
                xt[j + 1] = a;
            }
        }
    //the median
    xb = xt[5];
}
else if (filterType == 3) //minimum filter
{
    int xMinimum = xt[1]; //initialization
    //looking for minimum
    for (int i = 2; i < 10; i++)
    {
        if (xt[i] < xMinimum)
        {
            xMinimum = xt[i];
        }
    }
    xb = xMinimum;
}
else if (filterType == 4) //maximum filter
{
    //tambahkan coding
}
Color wb = Color.FromArgb(xb, xb, xb);
filteredImage.SetPixel(x, y, wb);
}

return filteredImage;
}

```

10. Double klik tombol **Median Filter** dan tambahkan rutin berikut :

```
private void radioButton2_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton2.Checked == false) return;

    //resetting radio button
    resetRadioButtonSharpening();
    resetRadioButtonLowPass();
    resetRadioButtonHighPass();

    if (noiseImage == null) return;
    Bitmap tempImage = new Bitmap(noiseImage);

    filterSmoothingType = 2;
    tempImage = smoothingfilter(filterSmoothingType);

    pictureBox1.Image = tempImage;
}
```

11. Double klik tombol **Minimum Filter** dan tambahkan rutin berikut :

```
private void radioButton3_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton3.Checked == false) return;

    //resetting radio button
    resetRadioButtonSharpening();
    resetRadioButtonLowPass();
    resetRadioButtonHighPass();

    if (noiseImage == null) return;
    Bitmap tempImage = new Bitmap(noiseImage);

    filterSmoothingType = 3;
    tempImage = smoothingfilter(filterSmoothingType);

    pictureBox1.Image = tempImage;
}
```

12. Tambahkan fungsi berikut untuk melakukan proses **Low Pass Filter** ::

```
private Bitmap lowPassFilter(int lowPassType)
{
    Bitmap filteredImage = new Bitmap(noiseImage);

    int[] xt = new int[10];
    int xb = 0;

    for (int x = 1; x < noiseImage.Width - 1; x++)
        for (int y = 1; y < noiseImage.Height - 1; y++)
        {
            Color w1 = noiseImage.GetPixel(x - 1, y - 1);
            Color w2 = noiseImage.GetPixel(x - 1, y);
            Color w3 = noiseImage.GetPixel(x - 1, y + 1);
            Color w4 = noiseImage.GetPixel(x, y - 1);
            Color w5 = noiseImage.GetPixel(x, y);
            Color w6 = noiseImage.GetPixel(x, y + 1);
            Color w7 = noiseImage.GetPixel(x + 1, y - 1);
```

```

Color w8 = noiseImage.GetPixel(x + 1, y);
Color w9 = noiseImage.GetPixel(x + 1, y + 1);

xt[1] = w1.R; xt[2] = w2.R; xt[3] = w3.R;
xt[4] = w4.R; xt[5] = w5.R; xt[6] = w6.R;
xt[7] = w7.R; xt[8] = w8.R; xt[9] = w9.R;

// low pass filter type 1
//      0  1  0
//  1/6  * 1  2  1
//      0  1  0
//
// low pass filter type 2
//      1  1  1
//  1/10 * 1  2  1
//      1  1  1
//
// low pass filter type 3
//      1  1  1
//  1/9  * 1  1  1
//      1  1  1
//
// low pass filter type 4
//      1  2  1
//  1/16 * 2  4  2
//      1  2  1
//

//calculation of low pass filter
if (lowPassType == 1)
{
    xb = (int)(0 * xt[1] + 1 * xt[2] + 0 * xt[3] +
               1 * xt[4] + 2 * xt[5] + 1 * xt[6] +
               0 * xt[7] + 1 * xt[8] + 0 * xt[9]) / 6;
    if (xb < 0) xb = 0;
    if (xb > 255) xb = 255;
}
else if (lowPassType == 2)
{
    //tambahkan koding
}
else if (lowPassType == 3)
{
    //tambahkan koding
}
else if (lowPassType == 4)
{
    //tambahkan koding
}

Color wb = Color.FromArgb(xb, xb, xb);
filteredImage.SetPixel(x, y, wb);
}

return filteredImage;
}

```

13. Double klik radioButton **Low Pass Filter Type 1** dan tambahkan rutin berikut:

```
private void radioButton5_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton5.Checked == false) return;

    if (noiseImage == null) return;
    Bitmap tempImage = new Bitmap(noiseImage);

    resetRadioButtonSmoothing();
    resetRadioButtonSharpening();
    resetRadioButtonHighPass();

    lowPassType = 1;
    tempImage = lowPassFilter(lowPassType);

    pictureBox1.Image = tempImage;
}
```

14. Tambahkan fungsi **sharpening filter** berikut :

```
//sharpening filter
private Bitmap sharpeningFilter(int filterType)
{
    noiseImage = grayImage;
    Bitmap filteredImage = new Bitmap(noiseImage);

    int[] xt = new int[10];
    int xb = 0;

    for (int x = 1; x < noiseImage.Width - 1; x++)
        for (int y = 1; y < noiseImage.Height - 1; y++)
        {
            Color w1 = noiseImage.GetPixel(x - 1, y - 1);
            Color w2 = noiseImage.GetPixel(x - 1, y);
            Color w3 = noiseImage.GetPixel(x - 1, y + 1);
            Color w4 = noiseImage.GetPixel(x, y - 1);
            Color w5 = noiseImage.GetPixel(x, y);
            Color w6 = noiseImage.GetPixel(x, y + 1);
            Color w7 = noiseImage.GetPixel(x + 1, y - 1);
            Color w8 = noiseImage.GetPixel(x + 1, y);
            Color w9 = noiseImage.GetPixel(x + 1, y + 1);

            xt[1] = w1.R; xt[2] = w2.R; xt[3] = w3.R;
            xt[4] = w4.R; xt[5] = w5.R; xt[6] = w6.R;
            xt[7] = w7.R; xt[8] = w8.R; xt[9] = w9.R;

            // Robert filter
            //      -1  1
            //      1  -1
            //
            // Prewit vertical filter
            //      -1  0  1
            //      -1  0  1
            //      -1  0  1
            //
            // Prewit horizontal filter
            //      -1  -1  -1
            //      0  0  0
            //      1  1  1
            //
            // Sobel horizontal filter
```

```

//      -1 -2 -1
//      0  0  0
//      1  2  1
//
// Sobel vertical filter
//      -1  0  1
//      -2  0  2
//      -1  0  1
//
// Laplacian filter
//      1 -2  1
//      -2  4 -2
//      1 -2  1
//

if (filterType == 1) //Robert filter
{
    //calculation of mean
    xb = xt[5] - xt[2] + xt[5] - xt[4];
    if (xb < 0) xb = 0;
    if (xb > 255) xb = 255;
}
else if (filterType == 2) //Prewitt filter
{
    int xh = -1 * xt[1] - 1 * xt[2] - 1 * xt[3] +
             0 * xt[4] + 0 * xt[5] + 0 * xt[6] +
             1 * xt[7] + 1 * xt[8] + 1 * xt[9];
    int xv = -1 * xt[1] + 0 * xt[2] + 1 * xt[3] +
             -1 * xt[4] + 0 * xt[5] + 1 * xt[6] +
             -1 * xt[7] + 0 * xt[8] + 1 * xt[9];

    xb = xh + xv;
    if (xb < 0) xb = 0;
    if (xb > 255) xb = 255;
}
else if (filterType == 3) //Sobel filter
{
    //tambahkan coding
}
else if (filterType == 4) //Laplacian filter
{
    // tambahkan coding
}

Color wb = Color.FromArgb(xb, xb, xb);
filteredImage.SetPixel(x, y, wb);
}

return filteredImage;
}

```


15. Double klik radioButton **Robert Filter** dan tambahkan rutin berikut:

```
private void radioButton9_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton9.Checked == false) return;
    if (noiseImage == null) return;

    resetCheckBox();
    resetRadioButtonSmoothing();
    resetRadioButtonLowPass();
    resetRadioButtonHighPass();

    Bitmap tempImage = new Bitmap(noiseImage);

    filterSharpeningType = 1;
    tempImage = sharpeningFilter(filterSharpeningType);

    pictureBox1.Image = tempImage;
}
```

16. Double klik radioButton **Prewitt Filter** dan tambahkan rutin berikut:

```
private void radioButton10_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton10.Checked == false) return;
    if (noiseImage == null) return;
    Bitmap tempImage = new Bitmap(noiseImage);

    resetCheckBox();
    resetRadioButtonSmoothing();
    resetRadioButtonLowPass();
    resetRadioButtonHighPass();

    filterSharpeningType = 2;
    tempImage = sharpeningFilter(filterSharpeningType);

    pictureBox1.Image = tempImage;
}
```

17. Tambahkan fungsi **High Pass Filter** berikut:

```
private Bitmap highPassFilter(int highPassType)
{
    noiseImage = grayImage;
    Bitmap filteredImage = new Bitmap(noiseImage);

    int[] xt = new int[10];
    int xb = 0;

    for (int x = 1; x < noiseImage.Width - 1; x++)
        for (int y = 1; y < noiseImage.Height - 1; y++)
        {
            Color w1 = noiseImage.GetPixel(x - 1, y - 1);
            Color w2 = noiseImage.GetPixel(x - 1, y);
            Color w3 = noiseImage.GetPixel(x - 1, y + 1);
            Color w4 = noiseImage.GetPixel(x, y - 1);
            Color w5 = noiseImage.GetPixel(x, y);
            Color w6 = noiseImage.GetPixel(x, y + 1);
            Color w7 = noiseImage.GetPixel(x + 1, y - 1);
            Color w8 = noiseImage.GetPixel(x + 1, y);
        }
}
```

```

Color w9 = noiseImage.GetPixel(x + 1, y + 1);

xt[1] = w1.R; xt[2] = w2.R; xt[3] = w3.R;
xt[4] = w4.R; xt[5] = w5.R; xt[6] = w6.R;
xt[7] = w7.R; xt[8] = w8.R; xt[9] = w9.R;

// high pass filter type 1
//      0  1  0
//      1 -4  1
//      0  1  0
//
// high pass filter type 2
//      0  -1  0
//     -1  4 -1
//      0  -1  0
//
// high pass filter type 3
//      1  1  1
//      1 -8  1
//      1  1  1
//
// high pass filter type 4
//     -1  -1  -1
//     -1  8  -1
//     -1  -1  -1
//

//calculation of low pass filter
if (highPassType == 1)
{
    xb = (int)(0 * xt[1] + 1 * xt[2] + 0 * xt[3] +
              1 * xt[4] - 4 * xt[5] + 1 * xt[6] +
              0 * xt[7] + 1 * xt[8] + 0 * xt[9]);
    if (xb < 0) xb = 0;
    if (xb > 255) xb = 255;
}
else if (highPassType == 2)
{
    //tambahkan koding
}
else if (highPassType == 3)
{
}
else if (highPassType == 4)
{
}

Color wb = Color.FromArgb(xb, xb, xb);
filteredImage.SetPixel(x, y, wb);
}

return filteredImage;
}

```

18. Double klik radioButton **High Pass Filter type 1** dan tambahkan rutin berikut

```
private void radioButton13_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton13.Checked == false) return;

    if (noiseImage == null) return;

    resetCheckBox();
    resetRadioButtonSmoothing();
    resetRadioButtonLowPass();
    resetRadioButtonSharpening();

    Bitmap tempImage = new Bitmap(noiseImage);

    highPassType = 1;
    tempImage = highPassFilter(highPassType);

    pictureBox1.Image = tempImage;
}
```

19. Double klik **checkbox** dan tambahkan fungsi berikut:

```
private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
    //resetting radio button
    resetRadioButtonSmoothing();
    resetRadioButtonSharpening();

    if (checkBox1.Checked == true)
    {
        Bitmap tempImage = noiseImaging(grayImage);
        noiseImage = tempImage;

        //menampilkan noise image
        pictureBox1.Image = noiseImage;
    }
    else
    {
        Bitmap tempImage = grayImaging(sourceImage);
        grayImage = tempImage;
        noiseImage = grayImage;

        //menampilkan gray image
        pictureBox1.Image = grayImage;
    }
}
```

20. Jalankan program dan load salah satu file gambar.

3.6 Tugas dan Pertanyaan

1. Tambahkan kode program untuk radioButton **Mean Filter** dan **Maximum Filter**.
2. Tambahkan kode program pada fungsi `private Bitmap smoothingfilter(int filterType)` untuk menampilkan citra hasil **Mean Filter** dan **Maximum Filter**.
3. Tambahkan kode program untuk radioButton **Low Pass Filter type 4**
4. Tambahkan kode program pada fungsi `private Bitmap lowPassFilter(int lowPassType)` untuk menampilkan citra hasil **Low Pass Filter type 4**.
5. Tambahkan kode program untuk radioButton **Sobel Filter** dan **Laplacian Filter**.
6. Tambahkan kode program pada fungsi `private Bitmap sharpeningFilter(int filterType)` untuk menampilkan citra hasil **Sobel Filter** dan **Laplacian Filter**..
7. Tambahkan kode program untuk radioButton **High Pass Filter type 4**
8. Tambahkan kode program pada fungsi `private Bitmap highPassFilter(int lowPassType)` untuk menampilkan citra hasil **High Pass Filter type 4**.