

---

---

# Visi Komputer dan Pengolahan Citra

Soal Ujian Tengah Semester

— 1223800005 - Silfiana Nur Hamida —

---

---

## Soal 2

2. Jelaskan tentang histogram equalization, kemudian selesaikan kasus berikut dengan salah satu contoh perhitungan detailnya !

# Jawaban dengan program C

```
#include<stdio.h>

void main(){
    int p[1000];int k[1000];
    int wnew[1000];
    int kk = 0;
    p[1] = 2; p[2] = 4;p[3] = 3;p[4] = 1;p[5] = 3;p[6] = 6;
    p[7] = 4;p[8] = 3;p[9] = 1;p[10] = 0;p[11] = 3;p[12] = 2;
    for (int i = 1; i <=12; i++)
    {
        kk = kk + p[i];
        k[i] = kk;
    }
    for (int i = 1; i <=12; i++)
    {
        wnew[i] = (k[i]*12)/32;
    }
    printf("w | Cw | w-baru\n");

    for (int i = 1; i <=12; i++)
    {
        printf("%d | %d | %d\n", p[i], k[i], wnew[i]);
    }
}
```

# Output Program

```
C:\Users\ASUS\Documents\PascaSarjana\MatrikulasiPemerograman\jawabanUTS_nomer2_Silfiana\bin\Debug\jawabanUTS_nomer2_Silfiana.exe
w | Cw | w-baru
2 | 2 | 0
4 | 6 | 2
3 | 9 | 3
1 | 10 | 3
3 | 13 | 4
6 | 19 | 7
4 | 23 | 8
3 | 26 | 9
1 | 27 | 10
0 | 27 | 10
3 | 30 | 11
2 | 32 | 12

Process returned 12 (0xC)   execution time : 0.028 s
Press any key to continue.
```

# Jawaban Program dengan Python

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

img = cv2.imread('mybest.jpg', cv2.IMREAD_GRAYSCALE)

hist, bins = np.histogram(img.flatten(), 256, [0, 256])

# Normalisasi histogram
normalisasi = hist / sum(hist)

# Hitung CDF
cdf = np.cumsum(normalisasi)

proses_equalized = np.interp(img.flatten(), range(256), 255 * cdf).astype(np.uint8)

result_equalized = proses_equalized.reshape(img.shape)

hist_eq, bins_eq = np.histogram(result_equalized.flatten(), 256, [0, 256])

plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.imshow(img, cmap='gray')
plt.title('Citra Asli')

plt.subplot(1, 2, 2)
plt.imshow(result_equalized, cmap='gray')
plt.title('Hasil Citra Setelah Histogram Equalization')

plt.show()
```

# Output Program



## Soal 3

3. Hasil gambar yang di filter Y dari gambar asal X dan filter dengan kernel H dapat dihitung dengan menggunakan persamaan konvolusi. Selesaikan perhitungan konvolusi berikut :

$$H = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Karena ukuran H adalah 3x3 agar simetri terhadap 0, maka batas perhitungan adalah -1,0 dan 1 untuk setiap posisi u dan v, maka berapa nilai  $Y(2,3)$  ?

# Jawaban Program dengan Python

```
import numpy as np

# matriks 3x3 dan 4x4
H = np.array([[1, 1, 1],
              [1, 4, 1],
              [1, 1, 1]])

X = np.array([[1, 0, 0, 0],
              [1, 1, 1, 0],
              [1, 1, 1, 0],
              [1, 0, 0, 0]])

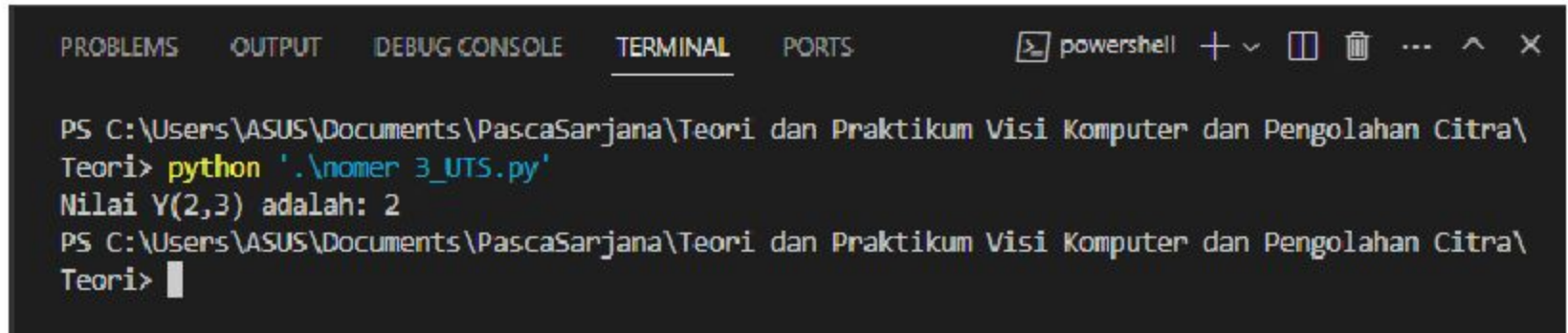
NilaiY = 0

# Perhitungan konvolusi
for m in range(-1, 2):
    for n in range(-1, 2):
        if 0 <= (2 - m) < X.shape[0] and 0 <= (3 - n) < X.shape[1]:
            NilaiY += X[2 - m, 3 - n] * H[m + 1, n + 1]

# Cetak nilai Y(2,3)
print("Nilai Y(2,3) adalah:", NilaiY)
```



# Output Program



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  powershell + - [ ] [ ] ... ^ X

PS C:\Users\ASUS\Documents\PascaSarjana\Teori dan Praktikum Visi Komputer dan Pengolahan Citra\
Teori> python '.\nomer 3_UTS.py'
Nilai Y(2,3) adalah: 2
PS C:\Users\ASUS\Documents\PascaSarjana\Teori dan Praktikum Visi Komputer dan Pengolahan Citra\
Teori> |
```

## Soal 4

4. Jelaskan cara perhitungan pada metode sobel dengan menggunakan kernel berikut

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \text{dan} \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

# Jawaban Program dengan C

```
#include<stdio.h>
```

```
void main(){
```

```
    int X[100][100];
    int V_awal, V_akhir, Qawal, Qakhir;
    int x, y; int Yx, Yy, Ytotal;
```

```
    //Matriks
```

```
    X[1][1] = 1; X[1][2] = 0;
    X[2][1] = 1; X[2][2] = 1;
    X[3][1] = 1; X[3][2] = 1;
    X[4][1] = 1; X[4][2] = 0;
```

```
    X[1][3] = 0; X[1][4] = 0;
    X[2][3] = 1; X[2][4] = 0;
    X[3][3] = 1; X[3][4] = 0;
    X[4][3] = 0; X[4][4] = 0;
```

```
    for(int s = 1; s<=10; s++)
```

```
    {
        X[10][s] = 0;
        X[s][10] = 0;
    }
```

```
    printf("Hasil perhitungan konvolusi dengan metode sobel pada:\n");
    printf("Sebutkan Baris Matriks: ");
    scanf("%d", &x);
    printf("Sebutkan Kolom Matriks : ");
    scanf("%d", &y);
```

```
    printf("Hasil perhitungan konvolusi dengan metode sobel pada:\n");
    printf("Sebutkan Baris Matriks: ");
    scanf("%d", &x);
    printf("Sebutkan Kolom Matriks : ");
    scanf("%d", &y);
```

```
    V_awal = x-1;
    V_akhir = x+1;
    Qawal = y-1;
    Qakhir = y+1;
```

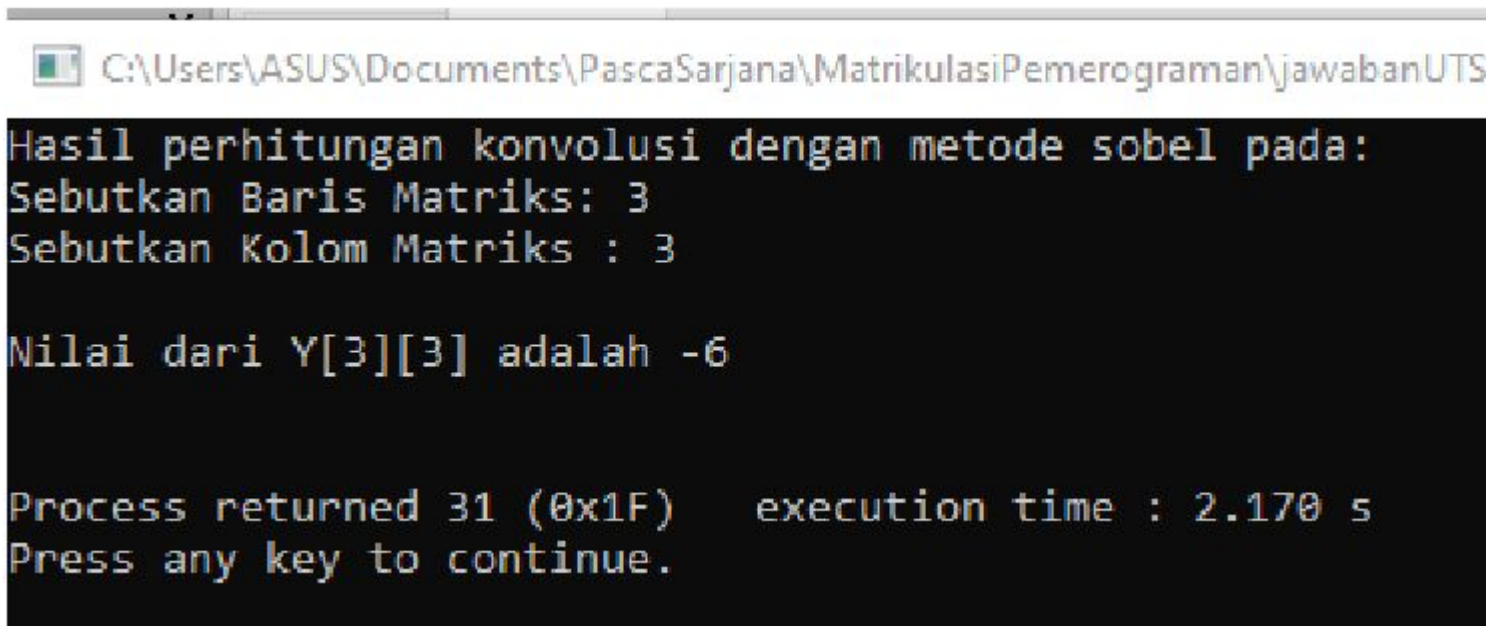
```
    if(V_awal<1) V_awal = 10;
    if(V_akhir>4) V_akhir = 10;
    if(Qawal<1) Qawal = 10;
    if(Qakhir>4) Qakhir = 10;
```

```
    //implementasi rumus
```

```
    Yx = -X[V_awal][Qawal]-2*X[V_awal][y]-X[V_awal][Qakhir]+X[V_akhir][Qawal]+2*X[V_akhir][y]+X[V_akhir][Qakhir];
    Yy = -X[V_awal][Qawal]+X[V_awal][Qakhir]-2*X[x][Qawal]+2*X[x][Qakhir]-X[V_akhir][Qawal]+X[V_akhir][Qakhir];
    Ytotal = Yx+Yy;
```

```
    printf("\nNilai dari Y[%d][%d] adalah %d\n\n", x,y,Ytotal);
```

# Output Program



```
C:\Users\ASUS\Documents\PascaSarjana\MatrikulasiPemerograman\jawabanUTS
Hasil perhitungan konvolusi dengan metode sobel pada:
Sebutkan Baris Matriks: 3
Sebutkan Kolom Matriks : 3

Nilai dari Y[3][3] adalah -6

Process returned 31 (0x1F)    execution time : 2.170 s
Press any key to continue.
```

# Jawaban Program dengan Python

```
import cv2
import numpy as np

gambar = cv2.imread('bestie.png', cv2.IMREAD_GRAYSCALE)
result_sobel_vertikal = np.array([[ -1,  0,  1],
                                   [ -2,  0,  2],
                                   [ -1,  0,  1]])

result_sobel_horizontal = np.array([[ -1, -2, -1],
                                     [  0,  0,  0],
                                     [  1,  2,  1]])

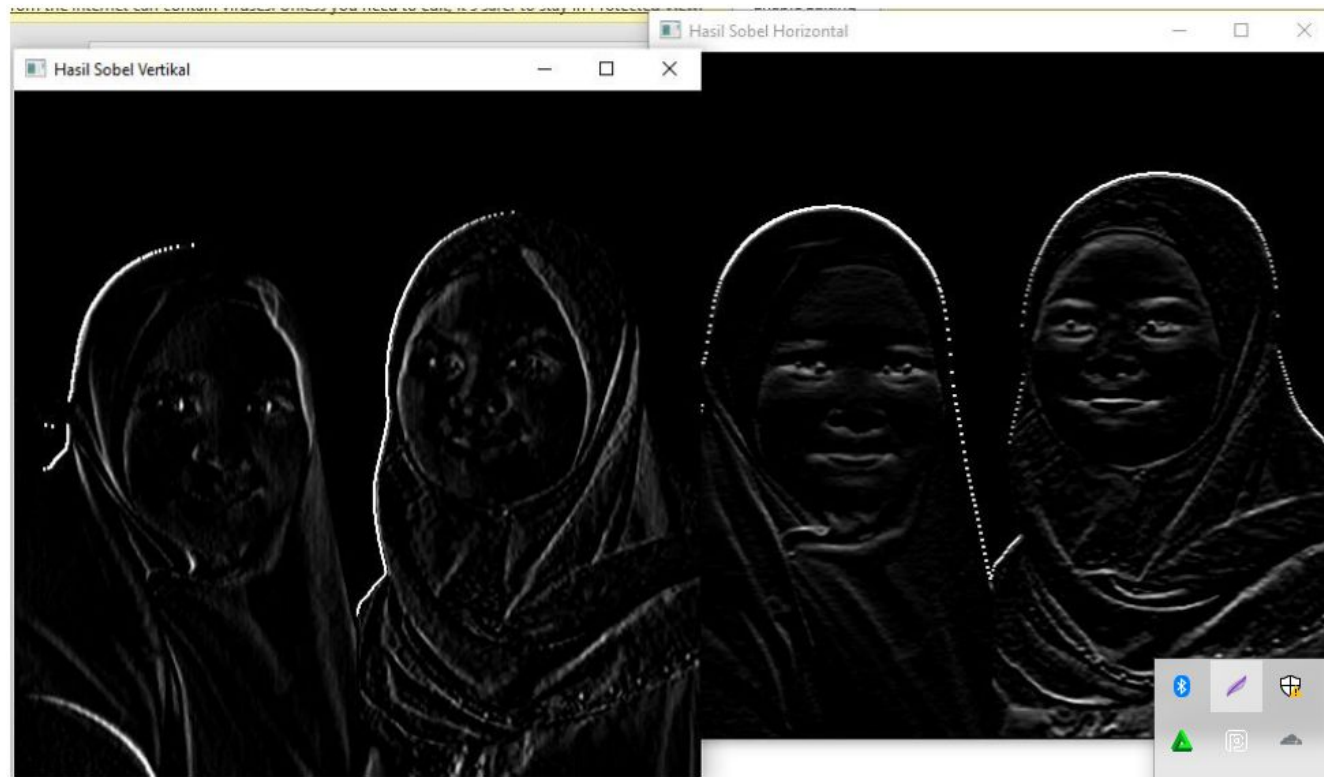
hasil_vertikal = cv2.filter2D(gambar, -1, result_sobel_vertikal)

hasil_horizontal = cv2.filter2D(gambar, -1, result_sobel_horizontal)

cv2.imshow('Hasil Sobel Vertikal', hasil_vertikal)
cv2.imshow('Hasil Sobel Horizontal', hasil_horizontal)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

# Output Program



**Terimakasih**