

# IF5152 VISI KOMPUTER

## LAPORAN TUGAS INDIVIDU 1



Disusun Oleh:

Maximilian Sulistiyo (13522061)

Kelas:

K01

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

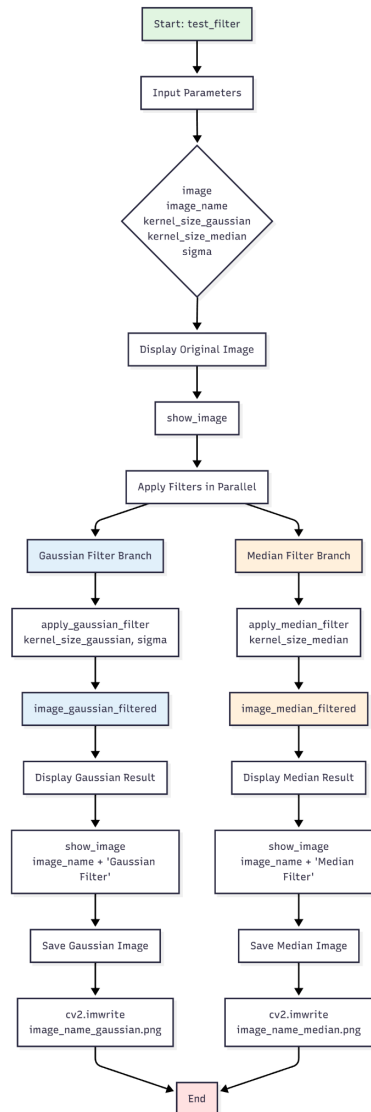
2025

## DAFTAR ISI

1. Workflow	3
1.1. Filtering	3
1.2. Edge	4
1.3. Feature Points	5
1.4. Geometry	6
2. Proses & Hasil Tiap Fitur	7
2.1. Filtering	7
2.2. Edge	10
2.3. Feature Points	13
2.4. Geometry	14
3. Komparasi dan Refleksi Pribadi	15
4. Lampiran	16

# 1. Workflow

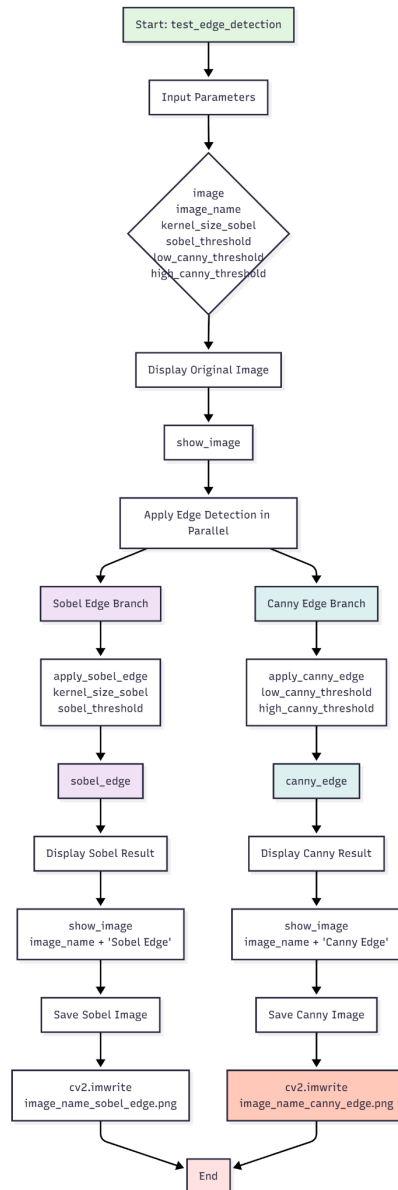
## 1.1. Filtering



Workflow dari filtering adalah:

1. Pertama menunjukkan gambar awal
2. Kemudian memanggil fungsi filtering gaussian dan median
3. Menampilkan gambar setelah di filter
4. Menyimpan gambar hasil filter

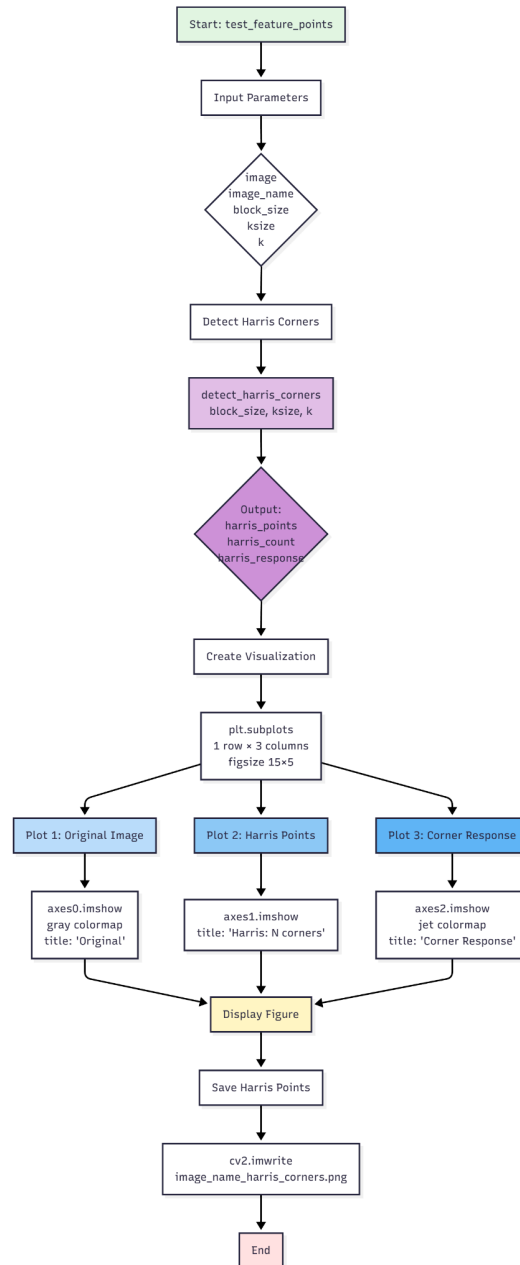
## 1.2. Edge



Workflow dari edge detection:

1. Pertama menunjukkan gambar awal
2. Kemudian memanggil fungsi edge detection sobel dan canny
3. Menampilkan gambar setelah di filter
4. Menyimpan gambar hasil filter

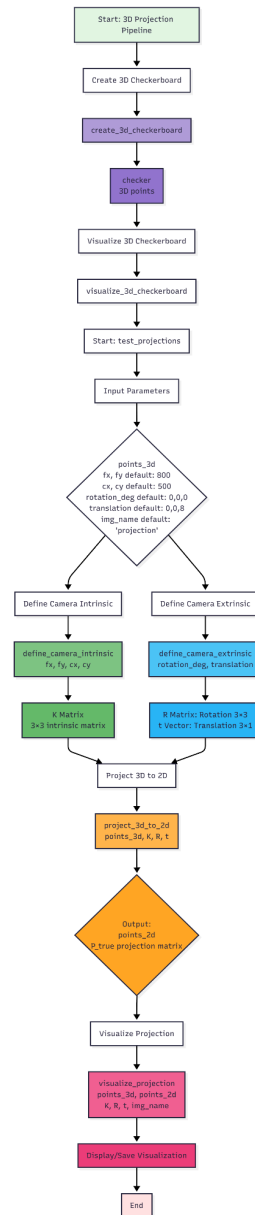
### 1.3. Feature Points



Workflow dari feature points adalah:

1. Pertama panggil fungsi harris corner
2. Tunjukkan hasil
3. Menyimpan gambar hasil marking

## 1.4. Geometry



Workflow dari geometry:

1. Membuat 3D object checkerboard
2. Menampilkan 3D object
3. Membuat matrix intrinsik dan ekstrinsik
4. Memproyeksikan 3D ke 2D
5. Melihatkan hasil
6. Menyimpan hasil

## 2. Proses & Hasil Tiap Fitur

### 2.1. Filtering

#### 2.1.1. Gaussian Filter

##### 2.1.1.1. Penjelasan

Gaussian filter merupakan *low-pass filter* yang digunakan untuk mereduksi noise dengan membaurkan (smoothing) intensitas piksel secara terkontrol menggunakan fungsi Gaussian. Nilai piksel keluaran diperoleh dari rata-rata berbobot tetangga piksel, di mana bobot yang lebih besar diberikan pada piksel yang lebih dekat dengan pusat kernel.

Karena sifatnya yang melakukan konvolusi dengan kernel berbobot, Gaussian cenderung menghilangkan detail frekuensi tinggi (misal, tepi halus dan tekstur halus), tetapi hasil blur-nya halus dan tidak menimbulkan artefak.

##### 2.1.1.2. Parameter

Menggunakan kernel size yang menentukan area tetangga dan sigma yang mengontrol besar kecilnya blur

##### 2.1.1.3. Hasil

Gambar awal:

Cameraman



Hasil (kernel 15, sigma 5):

Cameraman Gaussian Filter



## 2.1.2. Median Filter

### 2.1.2.1. Penjelasan



Median filter adalah *non-linear filter* yang mengganti nilai piksel dengan nilai median dari tetangga di sekitarnya. Berbeda dengan Gaussian yang menggunakan rata-rata berbobot, median mengambil nilai tengah (ranking) sehingga sangat efektif dalam menghilangkan *salt-and-pepper noise* tanpa mengaburkan tepi secara signifikan. Karena tidak melakukan smoothing berbasis rata-rata, struktur tepi dapat lebih terjaga.

#### 2.1.2.2. Parameter

Menggunakan kernel size yang menentukan banyaknya piksel tetangga yang dipakai untuk mencari nilai median

#### 2.1.2.3. Hasil

Gambar awal:

Cameraman



Hasil (kernel 7):

Camerman Median Filter



## 2.2. Edge

### 2.2.1. Sobel

#### 2.2.1.1. Penjelasan

Sobel adalah metode deteksi tepi berbasis turunan pertama (*gradient-based operator*). Operator ini menghitung perubahan intensitas piksel di arah horizontal ( $G_x$ ) dan vertikal ( $G_y$ ), kemudian menggabungkannya menjadi magnitude gradien. Tepi terdeteksi ketika perubahan intensitas cukup besar.

Sobel relatif sederhana dan cepat, tetapi peka terhadap noise karena tidak ada proses threshold adaptif atau pemurnian hasil.

#### 2.2.1.2. Parameter

Kernel size yang mengatur area tetangga, threshold untuk membatasi edge yang masuk berdasarkan nilai gradiennya

#### 2.2.1.3. Hasil

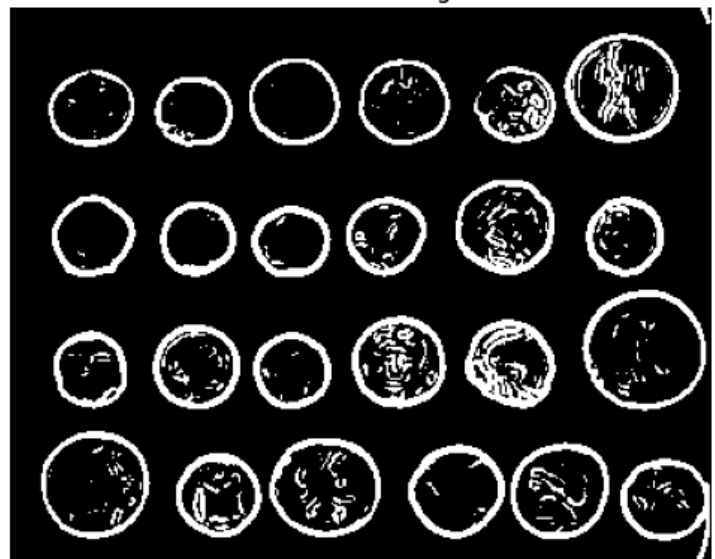
Gambar awal

Coins



Hasil sobel (kernel 7, threshold 50):

Coins Sobel Edge



## 2.2.2. Canny

### 2.2.2.1. Penjelasan

Canny merupakan metode deteksi tepi yang lebih kompleks dan stabil karena melalui beberapa tahap:

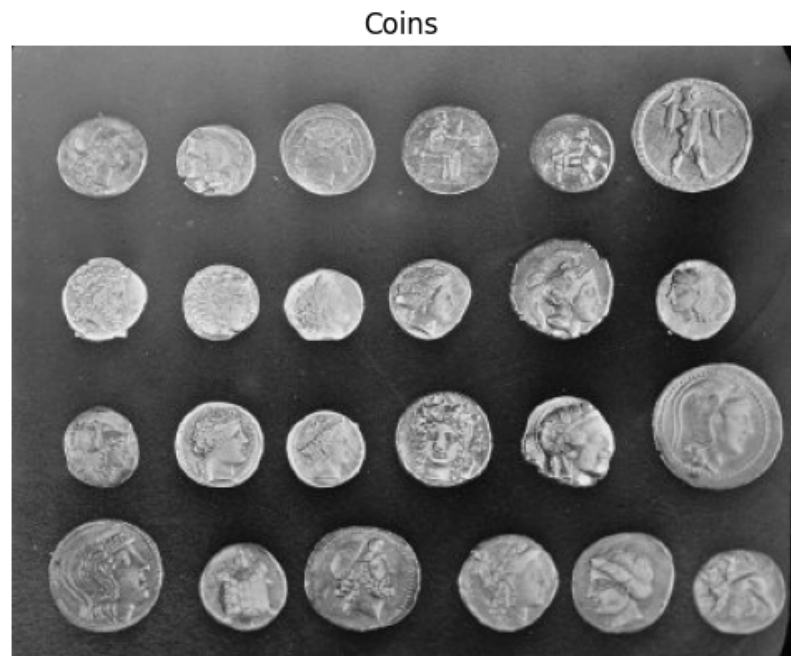
1. Gaussian smoothing → reduksi noise awal
2. Gradient + NMS (Non-Maximum Suppression) → mempertajam kandidat edge
3. Double threshold (low, high) → klasifikasi tepi lemah/kuat
4. Hysteresis tracking → menyeleksi edge akhir berdasarkan konektivitas

#### 2.2.2.2. Parameter

Low dan high threshold untuk menentukan candidate edge mana yang menjadi edge

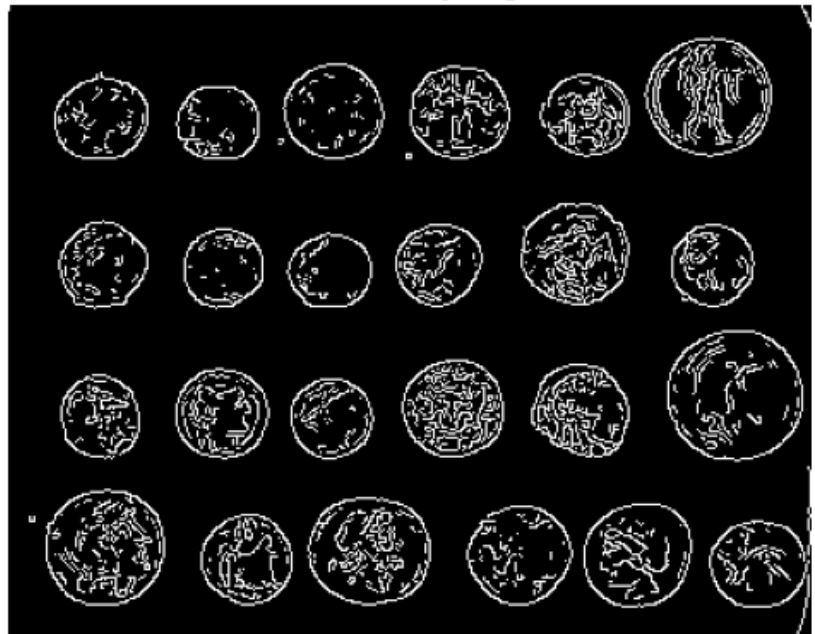
#### 2.2.2.3. Hasil

Gambar awal:



Hasil canny ( low\_t = 150, high\_t = 200):

Coins Canny Edge



2.2.2.4.

## 2.3. Feature Points

### 2.3.1. Penjelasan Harris

Harris Corner mendeteksi titik sudut dengan menganalisis variasi intensitas pada jendela kecil di berbagai arah. Sebuah “sudut” adalah lokasi di mana pergeseran kecil jendela ke arah mana pun menghasilkan perubahan intensitas yang besar. Secara matematis, dihitung *structure tensor* MMM dari gradien  $(I_x, I_y)$  lalu skor respons:

$$R = \det(M) - k \cdot (\text{trace}(M))^2$$

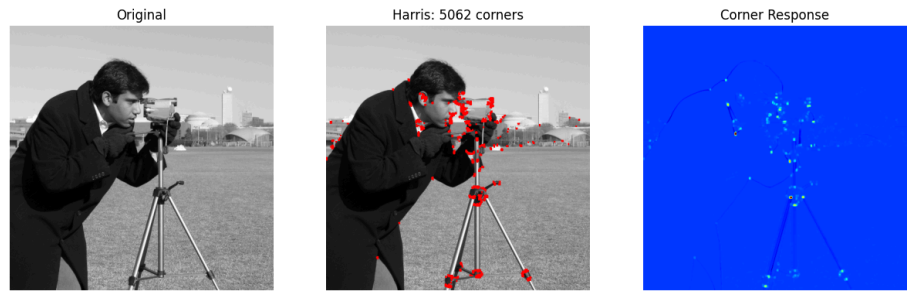
Titik dengan R tinggi dianggap sudut.

### 2.3.2. Parameter

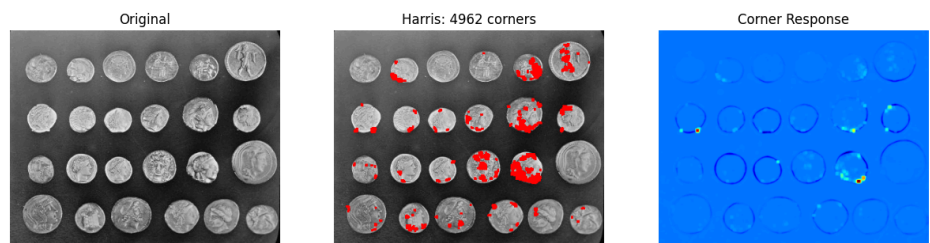
Block\_size merupakan ukuran jendela lokal yang dipakai untuk menghitung structure tensor, ksize merupakan aperture size sobel untuk gradien. K merupakan koefisien empiris pada rumus harris

### 2.3.3. Hasil

Hasil cameraman:



Hasil coins:



## 2.4. Geometry

### 2.4.1. Penjelasan

Model kamera *pinhole* memetakan titik 3D dunia  $X = [X, Y, Z]^T$  ke titik piksel  $x = [u, v, 1]^T$   $x=[u,v,1]^T$  melalui:

$$x = K[R|t]X$$

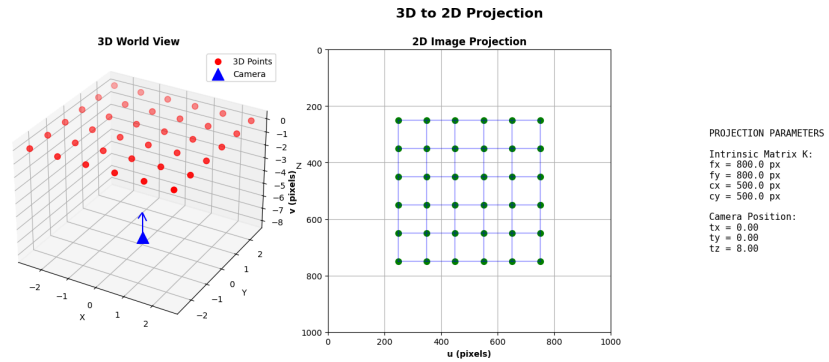
Dengan K matriks intrinsik (fokus  $f_x, f_y$ , pusat proyeksi  $c_x, c_y$ ), sedangkan R,t adalah posisi kamera relatif ke dunia (ekstrinsik). Distorsi lensa (radial  $k_1, k_2, k_3$ , tangensial  $p_1, p_2$ ) menyimpangkan proyeksi ideal

### 2.4.2. Parameter

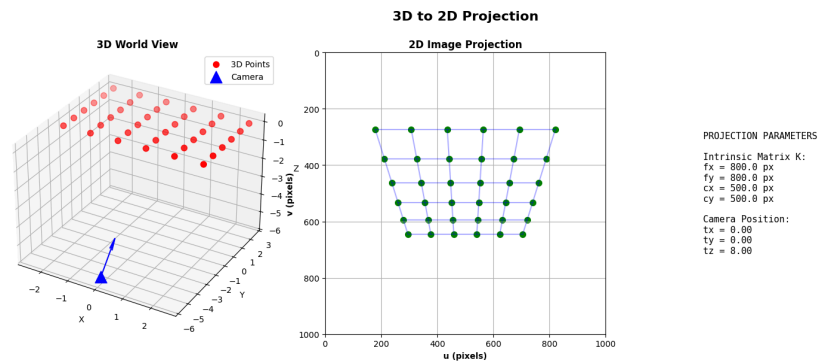
Points 3d merupakan 3D object yang akan diproyeksikan, K merupakan matrix intrinsik, R merupakan matriks rotasi, dan t merupakan matriks translasi

### 2.4.3. Hasil

Hasil  $f_x=800$ ,  $f_y=800$ ,  $c_x=500$ ,  $c_y=500$ ,  $rotation\_deg=(0, 0, 0)$ ,  $translation=(0, 0, 8)$ :



Hasil  $f_x=800$ ,  $f_y=800$ ,  $c_x=500$ ,  $c_y=500$ ,  $\text{rotation\_deg}=(45, 0, 0)$ ,  $\text{translation}=(0, 0, 8)$ :



### 3. Komparasi dan Refleksi Pribadi

#### 3.1. Filtering

##### 3.1.1. Gaussian

Pada gaussian kernel size dan sigma memengaruhi seberapa smooth image akhir menghilangkan noise tapi juga menghilangkan edge

##### 3.1.2. Median

Kernel size jika dibesarkan maka akan membuat image menjadi smooth tapi juga menghilangkan edges

#### 3.2. Edge

##### 3.2.1. Sobel

Jika kernel dibesarkan maka noisy edges berkurang tapi edges yang weak bisa hilang

##### 3.2.2. Canny

Low threshold jika ditingkatkan maka akan menghilangkan weak linkage dan jika high threshold dinaikkan maka pemilihan edges akan lebih selective

### 3.3. Feature Points

Block size jika diperbesar akan mengurangi noise tapi sudut kecil bisa hilang, ksize akan membuat gradien lebih halus tapi tepi tipis melemah, jika k dinaikkan maka lebih selektif

### 3.4. Refleksi

Dalam menjalani tugas ini saya kesusahan untuk mengerjakan no 4 karena detailnya kurang spesifik, apakah mengetes kalibrasi atau mencoba proyeksi, akhirnya saya memilih mencoba kalibrasi

## 4. Lampiran

Link github:

[https://github.com/riyorax/MaximilianSulistiyo\\_13522061\\_IF5152\\_TugasIndividuCV](https://github.com/riyorax/MaximilianSulistiyo_13522061_IF5152_TugasIndividuCV)