

**LAPORAN TUGAS INDIVIDU**  
**IF5152 Visi Komputer**



**Disusun oleh:**

**Angelica Kierra Ninta Gurning**                           **(13522048)**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**  
**INSTITUT TEKNOLOGI BANDUNG**

**2025**

## DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>1</b>
<b>BAB I.....</b>	<b>5</b>
<b>Workflow Pipeline.....</b>	<b>5</b>
<b>BAB II.....</b>	<b>7</b>
<b>Proses &amp; Hasil Tiap Fitur.....</b>	<b>7</b>
<b>2.1. Image Filtering.....</b>	<b>7</b>
2.1.1. Pengertian Image Filtering.....	7
2.1.2. Jenis Filter yang Digunakan.....	7
2.1.3. Jenis Eksperimen Yang Dilakukan.....	9
2.1.4. Hasil.....	9
2.1.4.1. Eksperimen 1 : Gaussian dengan kernel tetap.....	9
2.1.4.2. Eksperimen 2 : Gaussian dengan sigma tetap.....	12
2.1.4.3. Eksperimen 3 : Median.....	14
<b>2.2. Edge Detection.....</b>	<b>17</b>
2.2.1. Pengertian Edge Detection.....	17
2.2.2. Jenis Edge Detection Yang Digunakan.....	17
2.2.3. Jenis Eksperimen Yang Dilakukan.....	19
2.2.4. Hasil.....	20
2.2.4.1. Eksperimen 1 : Sobel.....	20
2.2.4.2. Eksperimen 2: Canny.....	26
<b>2.3. Feature Points.....</b>	<b>33</b>
2.3.1. Pengertian Feature Points.....	33
2.3.2. Jenis Corner Detector Yang Digunakan.....	34
2.3.3. Jenis Eksperimen Yang Dilakukan.....	35
2.3.4. Hasil.....	36
2.3.4.1. Eksperimen 1 : Harris Corner Detector.....	36
2.3.4.2. Eksperimen 2: SIFT.....	41
2.3.4.3. Eksperimen 3 : FAST Feature Detector.....	44
<b>2.4. Transformasi Geometri.....</b>	<b>48</b>
2.4.1. Pengertian Transformasi Geometri.....	48
2.4.2. Jenis Transformasi Geometri Yang Digunakan.....	49
2.4.3. Jenis Eksperimen Yang Dilakukan.....	50
2.4.4. Hasil.....	50
Analisis.....	55
<b>BAB III.....</b>	<b>56</b>
<b>Komparasi &amp; Refleksi Pribadi.....</b>	<b>56</b>
<b>3.1. Image Filtering.....</b>	<b>56</b>
<b>3.2. Edge Detection.....</b>	<b>57</b>

<b>3.3. Feature Points.....</b>	<b>58</b>
<b>3.4. Transformasi Geometri.....</b>	<b>59</b>

## DAFTAR GAMBAR

Gambar 2.1 Hasil Foto ‘Camera’ Gaussian I.....	10
Gambar 2.2 Hasil Foto ‘Camera’ Gaussian II.....	10
Gambar 2.3 Hasil Foto ‘Coin’ Gaussian I.....	10
Gambar 2.4 Hasil Foto ‘Coin’ Gaussian II.....	11
Gambar 2.5 Hasil Foto Pribadi Gaussian I.....	11
Gambar 2.6 Hasil Foto Pribadi Gaussian II.....	11
Gambar 2.7 Hasil Foto “Checkerboard” Gaussian I.....	12
Gambar 2.8 Hasil Foto “Checkerboard” Gaussian II.....	12
Gambar 2.9 Hasil Foto “Astronaut” Gaussian I.....	13
Gambar 2.10 Hasil Foto “Astronaut” Gaussian II.....	13
Gambar 2.11 Hasil Foto Pribadi Gaussian III.....	14
Gambar 2.12 Hasil Foto Pribadi Gaussian IV.....	14
Gambar 2.13 Hasil Foto “Camera” Median I.....	15
Gambar 2.14 Hasil Foto “Camera” Median II.....	15
Gambar 2.15 Hasil Foto “Camera” Median III.....	15
Gambar 2.16 Hasil Foto Pribadi Median I.....	16
Gambar 2.17 Hasil Foto Pribadi Median II.....	16
Gambar 2.18 Hasil Foto Pribadi Median III.....	16
Gambar 2.19 Hasil Foto “Camera” Sobel I.....	20
Gambar 2.20 Hasil Foto “Camera” Sobel II.....	20
Gambar 2.21 Hasil Foto “Camera” Sobel III.....	21
Gambar 2.22 Hasil Foto “Coin” Sobel I.....	21
Gambar 2.23 Hasil Foto “Coin” Sobel II.....	21
Gambar 2.24 Hasil Foto “Coin” Sobel III.....	21
Gambar 2.25 Hasil Foto “Checkerboard” Sobel I.....	22
Gambar 2.26 Hasil Foto “Checkerboard” Sobel II.....	22
Gambar 2.27 Hasil Foto “Checkerboard” Sobel III.....	22
Gambar 2.28 Hasil Foto “Astronaut” Sobel I.....	23
Gambar 2.29 Hasil Foto “Astronaut” Sobel II.....	23
Gambar 2.30 Hasil Foto “Astronaut” Sobel III.....	23
Gambar 2.31 Hasil Foto “Chelsea” Sobel I.....	24
Gambar 2.32 Hasil Foto “Chelsea” Sobel II.....	24
Gambar 2.33 Hasil Foto “Chelsea” Sobel III.....	24
Gambar 2.34 Hasil Foto Pribadi Sobel I.....	25
Gambar 2.35 Hasil Foto Pribadi Sobel II.....	25
Gambar 2.36 Hasil Foto Pribadi Sobel III.....	25
Gambar 2.37 Hasil Foto “Camera” Canny I.....	27
Gambar 2.38 Hasil Foto “Camera” Canny II.....	27

Gambar 2.39 Hasil Foto “Camera” Canny III.....	27
Gambar 2.40 Hasil Foto “Coins” Canny I.....	28
Gambar 2.41 Hasil Foto “Coins” Canny II.....	28
Gambar 2.42 Hasil Foto “Coins” Canny III.....	28
Gambar 2.43 Hasil Foto “Checkerboard” Canny I.....	29
Gambar 2.44 Hasil Foto “Checkerboard” Canny II.....	29
Gambar 2.45 Hasil Foto “Checkerboard” Canny III.....	29
Gambar 2.46 Hasil Foto “Astronaut” Canny I.....	30
Gambar 2.47 Hasil Foto “Astronaut” Canny II.....	30
Gambar 2.48 Hasil Foto “Astronaut” Canny III.....	30
Gambar 2.49 Hasil Foto “Chelsea” Canny I.....	31
Gambar 2.50 Hasil Foto “Chelsea” Canny II.....	31
Gambar 2.51 Hasil Foto “Chelsea” Canny III.....	31
Gambar 2.52 Hasil Foto Pribadi Canny I.....	32
Gambar 2.53 Hasil Foto Pribadi Canny II.....	32
Gambar 2.54 Hasil Foto Pribadi Canny III.....	32
Gambar 2.55 Hasil Foto “Camera” Harris (Block) I.....	36
Gambar 2.56 Hasil Foto “Camera” Harris (Block) II.....	37
Gambar 2.57 Hasil Foto “Camera” Harris (Kernel) I.....	38
Gambar 2.58 Hasil Foto “Camera” Harris (Kernel) II.....	38
Gambar 2.59 Hasil Foto “Camera” Harris (K) I.....	39
Gambar 2.60 Hasil Foto “Camera” Harris (K) II.....	39
Gambar 2.61 Hasil Foto “Camera” Harris (Threshold) I.....	40
Gambar 2.62 Hasil Foto “Camera” Harris (Threshold) II.....	40
Gambar 2.63 Hasil Foto “Coins” SIFT I.....	41
Gambar 2.64 Hasil Foto “Coins” SIFT II.....	41
Gambar 2.65 Hasil Foto “Coins” SIFT III.....	42
Gambar 2.66 Hasil Foto Pribadi SIFT I.....	42
Gambar 2.67 Hasil Foto Pribadi SIFT II.....	42
Gambar 2.68 Hasil Foto Pribadi SIFT III.....	43
Gambar 2.69 Hasil Foto “Chelsea” FAST I.....	44
Gambar 2.70 Hasil Foto “Chelsea” FAST II.....	45
Gambar 2.71 Hasil Foto “Chelsea” FAST III.....	45
Gambar 2.72 Hasil Foto Pribadi FAST I.....	45
Gambar 2.73 Hasil Foto Pribadi FAST II.....	45
Gambar 2.74 Hasil Foto Pribadi FAST III.....	46
Gambar 2.75 Hasil Foto “Chelsea” FAST (suppression) I.....	47
Gambar 2.76 Hasil Foto “Chelsea” FAST (suppression) II.....	47
Gambar 2.77 Hasil Foto Pribadi FAST (suppression) I.....	48

Gambar 2.78 Hasil Foto Pribadi FAST (suppression) II.....	48
Gambar 2.79 Hasil Foto “Checkerboard” Affine.....	51
Gambar 2.80 Hasil Foto “Checkerboard” Affine Matrix.....	51
Gambar 2.81 Hasil Foto “Checkerboard” Perspective.....	52
Gambar 2.82 Hasil Foto “Checkerboard” Perspective Matrix.....	52
Gambar 2.83 Hasil Foto “Astronaut” Affine.....	53
Gambar 2.84 Hasil Foto “Astronaut” Affine Matrix.....	53
Gambar 2.85 Hasil Foto “Astronaut” Perspective.....	53
Gambar 2.86 Hasil Foto “Astronaut” Perspective Matrix.....	53
Gambar 2.87 Hasil Foto Pribadi Affine.....	54
Gambar 2.88 Hasil Foto Pribadi Affine Matrix.....	54
Gambar 2.89 Hasil Foto Pribadi Perspective.....	54
Gambar 2.90 Hasil Foto Pribadi Perspective Matrix.....	55

## **DAFTAR TABEL**

Tabel 2.1 Jenis Eksperimen Filtering.....	10
Tabel 2.2 Jenis Eksperimen Edge Detection.....	20
Tabel 2.3 Analisis Karakteristik Sobel.....	27
Tabel 2.4 Analisis Karakteristik Canny.....	34
Tabel 2.5 Jenis Eksperimen Feature Points.....	36
Tabel 2.6 Hasil Statistik SIFT.....	44
Tabel 2.7 Hasil Statistik FAST.....	47
Tabel 2.8 Jenis Eksperimen Transformasi Geometri.....	51

# BAB I

## Workflow Pipeline

Dalam bidang pengolahan citra digital (digital image processing), proses analisis citra tidak dilakukan secara langsung pada gambar mentah, melainkan melalui serangkaian tahapan yang terstruktur dan saling berkaitan. Setiap tahap memiliki fungsi spesifik yang berkontribusi terhadap peningkatan kualitas data visual serta kemudahan dalam ekstraksi informasi penting dari citra.

Tahap-tahap pipeline adalah sebagai berikut:

### 1. Input Image

Tahapan awal dalam workflow pemrosesan citra adalah menyiapkan citra masukan (input image). Biasanya citra yang digunakan masih dalam format berwarna (RGB), di mana setiap piksel memiliki tiga kanal warna: merah, hijau, dan biru.

Namun, untuk keperluan analisis visual seperti filtering, edge detection, dan feature extraction, citra terlebih dahulu dikonversi ke bentuk grayscale (hitam-putih). Proses konversi ini dilakukan untuk:

- a. Mengurangi kompleksitas komputasi dari tiga kanal warna menjadi satu kanal intensitas.
- b. Memastikan fokus analisis pada perbedaan tingkat kecerahan (intensity), bukan perbedaan warna.
- c. Menghindari gangguan warna yang dapat mempengaruhi hasil deteksi tepi dan fitur.

### 2. Filtering

Setelah citra dikonversi ke grayscale, langkah berikutnya adalah filtering atau penyaringan. Tahapan ini berfungsi sebagai *pre-processing* yang bertujuan untuk mengurangi gangguan (noise) dan memperhalus variasi intensitas agar struktur utama citra lebih mudah dikenali. Metode yang digunakan adalah sebagai berikut:

- a. Gaussian Filter
- b. Median Filter

### 3. Edge Detection

Tahap ketiga dalam pipeline adalah deteksi tepi, yaitu proses untuk menemukan area yang menunjukkan perubahan intensitas piksel secara signifikan. Tepi (edge) menggambarkan batas antara dua wilayah berbeda dalam citra, misalnya antara objek dan latar belakang. Metode yang digunakan adalah sebagai berikut:

- a. Sobel Operator
- b. Canny Edge Detector

### 4. Feature Points Detection

Setelah struktur tepi objek diketahui, tahap berikutnya adalah deteksi titik fitur (feature points detection). Titik fitur adalah bagian citra yang memiliki karakteristik unik dan stabil terhadap perubahan pencahayaan, rotasi, atau skala, sehingga dapat digunakan sebagai acuan dalam proses pencocokan (matching) antar-citra. Metode yang digunakan adalah sebagai berikut:

- a. Harris Corner Detector
- b. SIFT
- c. FAST

### 5. Geometric Transformation

Tahap terakhir dalam pipeline adalah geometric transformation, yaitu proses untuk memanipulasi posisi, orientasi, atau bentuk citra berdasarkan titik-titik fitur yang telah diperoleh sebelumnya.

Metode yang digunakan adalah sebagai berikut:

- a. Affine Transformation
- b. Perspective Transformation

## BAB II

### Proses & Hasil Tiap Fitur

#### 2.1. Image Filtering

##### 2.1.1. Pengertian Image Filtering

*Filtering* merupakan operasi yang memodifikasi suatu sinyal terukur dalam bentuk 1D, 2D, ataupun multidimensi. Dalam pemrosesan citra digital, filtering bertujuan untuk mengubah nilai intensitas piksel berdasarkan nilai piksel di sekitarnya (neighborhood operation). Secara umum, operasi filtering dapat digunakan untuk:

1. Menghilangkan komponen yang tidak diinginkan (misalnya noise).
2. Mengubah sinyal menjadi bentuk yang lebih diinginkan.
3. Mengekstraksi komponen tertentu dari citra.

Secara umum, operasi filter dua dimensi dapat dinyatakan sebagai:

$$h[m, n] = \sum_{k,l} f[k, l] \cdot I[m + k, n + l]$$

dengan

$I$  = citra masukan,

$f$  = kernel filter,

$h$  = hasil citra terfilter,

$m, n$  = koordinat piksel,

$k, l$  = koordinat kernel

##### 2.1.2. Jenis Filter yang Digunakan

Pada eksperimen ini, akan digunakan dua jenis filter utama, yaitu:

1. Gaussian Filter (Linear Filter)

Gaussian filter merupakan *low-pass filter* berbasis distribusi Gaussian yang digunakan untuk menghaluskan citra (reduksi noise dan detail tajam).

Kernel Gaussian berbentuk simetris dan dikendalikan oleh parameter:

- a. Kernel size (ksize): menentukan ukuran jendela (misalnya  $3 \times 3$ ,  $5 \times 5$ ,  $9 \times 9$ ). Semakin besar ukuran kernel, semakin kuat efek smoothing.

- b. Sigma ( $\sigma$ ): standar deviasi fungsi Gaussian yang mengontrol tingkat penyebaran (blur). Nilai  $\sigma$  besar akan menghasilkan blur dengan jangkauan lebih luas.

Secara matematis, fungsi Gaussian 2D dapat didefinisikan sebagai:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Filter ini bersifat separable, artinya konvolusi 2D dapat dipecah menjadi dua konvolusi 1D (arah x dan y), sehingga lebih efisien secara komputasi. Dalam implementasi, kode yang digunakan adalah sebagai berikut:

```
cv2.GaussianBlur(img, (ksize, ksize), sigma)
```

## 2. Median Filter (Non-Linear Filter)

Median filter termasuk non-linear filter yang menggantikan setiap piksel dengan nilai median dari piksel di sekitarnya (berdasarkan ukuran jendela). Berbeda dengan mean atau Gaussian filter yang menghitung rata-rata, median filter mempertahankan tepi (*edges*) lebih baik dan sangat efektif untuk menghilangkan salt and pepper noise. Dalam implementasi, kode yang digunakan adalah sebagai berikut:

```
cv2.medianBlur(img, ksize)
```

Parameter utama:

1. ksize: ukuran jendela (harus bilangan ganjil, misalnya 3, 5, 9).

Pada median filter, ksize menentukan jumlah piksel yang dipertimbangkan untuk mencari median. Semakin besar ukuran ksize, maka semakin kuat efek “pembersihan noise”

- ksize kecil ( $3\times 3$ ) → menghilangkan noise kecil tanpa banyak mengubah struktur gambar.
- ksize besar ( $9\times 9$ ) → menghilangkan noise lebih efektif, tetapi juga bisa menghapus detail halus atau membuat tepi objek melebar.

### 2.1.3. Jenis Eksperimen Yang Dilakukan

Berikut merupakan jenis eksperimen yang dilakukan untuk melihat hasil *filtering* menggunakan Gaussian dan Median:

Tabel 2.1 Jenis Eksperimen Filtering

Jenis Eksperimen	Filter	Parameter Tetap	Parameter Variasi	Tujuan
Eksperimen 1	Gaussian	ksize	sigma	Melihat pengaruh perubahan sigma terhadap tingkat blur.
Eksperimen 2	Gaussian	sigma	ksize	Melihat pengaruh ukuran kernel terhadap hasil smoothing.
Eksperimen 3	Median	–	ksize	Membandingkan hasil median filter terhadap Gaussian filter.

### 2.1.4. Hasil

#### 2.1.4.1. Eksperimen 1 : Gaussian dengan kernel tetap

Data Uji : Camera	
Parameter	Hasil
kernel =5 Sigma = 2.0	 <b>Gaussian (k=5, sigma=2.0)</b>

	Gambar 2.1 Hasil Foto ‘Camera’ Gaussian I
kernel =5 Sigma = 10.0	 <p><b>Gaussian (k=5, sigma=10.0)</b></p>

Gambar 2.2 Hasil Foto ‘Camera’ Gaussian II

### Analisis

Pada gambar pertama, ketika  $\sigma = 2.0$ , detail wajah, tripod, dan bangunan di latar belakang masih terlihat jelas namun tepi (*edges*) mulai melembut. Namun, pada  $\sigma = 10.0$ , efek blur meningkat drastis, seluruh kontur objek melebur, batas antara tubuh dan latar belakang menjadi tidak tajam, serta kontras tepi berkurang signifikan.

Data Uji : Coin	
Parameter	Hasil
kernel =7 Sigma = 0.5	 <p><b>Gaussian (k=7, sigma=0.5)</b></p>

Gambar 2.3 Hasil Foto ‘Coin’ Gaussian I

kernel =7 Sigma = 5.0	 <b>Gaussian (k=7, sigma=5.0)</b>	
Gambar 2.4 Hasil Foto ‘Coin’ Gaussian II		

### Analisis

Ketika  $\sigma = 0.5$ , efek Gaussian hampir tidak terlihat, detail pada permukaan koin tetap tajam, hanya sedikit perataan pada noise kecil. Pada  $\sigma = 5.0$ , tepi antar koin melebur, bentuk koin masih dapat dikenali tetapi relief dan ukiran di permukaan benar-benar hilang.

Foto Pribadi	
Parameter	Hasil
kernel =5 Sigma = 6.0	 <b>Gaussian (k=5, sigma=6.0)</b>
kernel =5 Sigma = 36.0	 <b>Gaussian (k=5, sigma=36.0)</b>

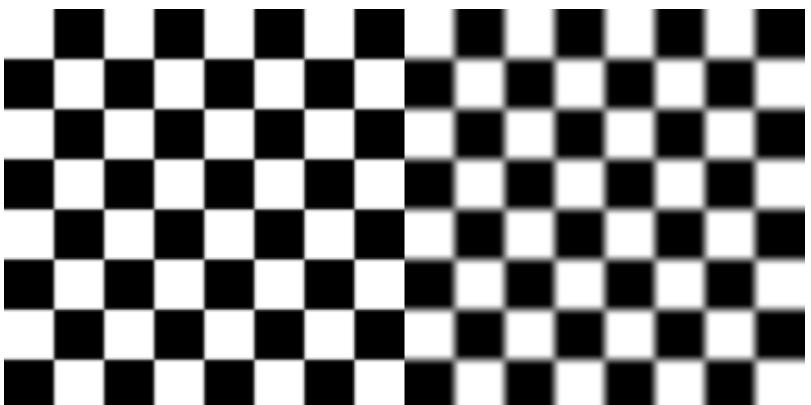
Gambar 2.5 Hasil Foto Pribadi Gaussian I

Gambar 2.6 Hasil Foto Pribadi Gaussian II

## Analisis

Pada  $\sigma = 6.0$ , filter menghaluskan tekstur genteng dan awan, namun bentuk bangunan tetap terdefinisi. Ketika  $\sigma = 36.0$ , seluruh citra tampak seperti bayangan kabur, hanya area besar yang tersisa.

### 2.1.4.2. Eksperimen 2 : Gaussian dengan sigma tetap

Data Uji : Checkerboard	
Parameter	Hasil
kernel =5 Sigma = 5.0	 <b>Gaussian (sigma=5.0, k=5)</b>
kernel =15 Sigma = 5.0	 <b>Gaussian (sigma=5.0, k=15)</b>

Gambar 2.7 Hasil Foto “Checkerboard” Gaussian I

Gambar 2.8 Hasil Foto “Checkerboard” Gaussian II

## Analisis

Pada  $k = 5$ , efek blur masih terbatas di sekitar tepi tiap kotak. Pola kotak hitam-putih tetap jelas, hanya sedikit lembut di batasnya. Namun pada  $k = 15$ , area blur meluas dan perbedaan antara kotak hitam dan putih melebur sehingga menghasilkan transisi abu-abu yang lebih halus di antara keduanya.

Data Uji : Astronaut	
Parameter	Hasil
kernel =7 Sigma = 7.0	 <p><b>Gaussian (sigma=7.0, k=7)</b></p> <p>Gambar 2.9 Hasil Foto “Astronaut” Gaussian I</p>
kernel =21 Sigma = 7.0	 <p><b>Gaussian (sigma=7.0, k=21)</b></p> <p>Gambar 2.10 Hasil Foto “Astronaut” Gaussian II</p>

### Analisis

Dengan  $k = 7$ , citra masih mempertahankan bentuk utama seperti wajah, helm, dan logo pada baju tetapi dapat dikenali, meskipun tekstur halus seperti rambut dan lipatan baju sedikit melembut. Ketika  $k = 21$ , blur meningkat signifikan, fitur wajah dan tepi tubuh menjadi sangat kabur, citra tampak lebih homogen tanpa tekstur detail.

Foto Pribadi	
Parameter	Hasil
kernel =3 Sigma = 1.0	 <p><b>Gaussian (sigma=1.0, k=3)</b></p>
kernel =21 Sigma = 1.0	 <p><b>Gaussian (sigma=1.0, k=21)</b></p>

### Analisis

Nilai sigma yang kecil ( $\sigma = 1.0$ ) menghasilkan blur yang ringan, sehingga perubahan akibat variasi kernel size terlihat lebih halus. Pada  $k = 3$ , citra hanya sedikit melembut di tepian genteng dan awan. Namun ketika  $k = 21$ , efek blur lebih terasa meskipun sigma sama, detail genteng dan tekstur dinding melebur, menunjukkan bahwa kernel yang lebih besar tetap memperkuat efek smoothing karena jangkauan area rata-rata menjadi lebih luas.

#### 2.1.4.3. Eksperimen 3 : Median

Data Uji : Camera	
Parameter	Hasil

kernel =3		Gambar 2.13 Hasil Foto “Camera” Median I
kernel =15		Gambar 2.14 Hasil Foto “Camera” Median II
Kernel = 21		Gambar 2.15 Hasil Foto “Camera” Median III

### Analisis

Pada  $k = 3$ , efek smoothing ringan dan detail masih terlihat jelas. Saat  $k = 15$ , detail halus mulai hilang. Permukaan lapangan terlihat lebih rata, wajah dan tangan

menjadi lebih homogen. Namun, bentuk utama objek tetap jelas. Saat  $k = 21$ , seluruh citra tampak seperti *posterized*, dengan bidang besar berwarna abu-abu rata. Detail kecil, seperti tekstur baju dan struktur tripod menghilang.

<b>Foto Pribadi</b>	
<b>Parameter</b>	<b>Hasil</b>
kernel =5	 <p><b>Median (k=5)</b></p>
Gambar 2.16 Hasil Foto Pribadi Median I	
kernel =13	 <p><b>Median (k=13)</b></p>
Gambar 2.17 Hasil Foto Pribadi Median II	
Kernel = 29	 <p><b>Median (k=29)</b></p>
Gambar 2.18 Hasil Foto Pribadi Median III	

### **Analisis**

Saat  $k = 5$ , noise dan tekstur kecil di dinding dan genteng sedikit dihaluskan. Gambar masih tampak natural. Saat  $k = 13$ , transisi antar bidang (atap, langit,

tembok) mulai terlihat rata, awan menjadi lebih solid. Saat  $k = 29$ , efek smoothing terlihat dengan sangat jelas, seluruh detail arsitektur hilang, hanya bentuk besar bangunan yang tersisa. Area terang dan gelap bercampur menjadi blok besar.

## 2.2. Edge Detection

### 2.2.1. Pengertian Edge Detection

*Edge detection* adalah proses untuk mengidentifikasi bagian dari citra digital yang mengalami perubahan intensitas secara tajam (*discontinuity in image intensity*). Bagian-bagian ini biasanya menandai batas antara dua area yang berbeda warna, tekstur, atau iluminasi. Tujuan utama edge detection adalah untuk menyederhanakan representasi citra sehingga lebih mudah diolah untuk:

1. Pengenalan objek (*object recognition*)
2. Rekonstruksi bentuk atau adegan (*scene reconstruction*)
3. Analisis geometri dan orientasi permukaan
4. Efek artistik dan penyuntingan gambar.

*Edge detection* juga sering digunakan untuk:

1. *Depth discontinuity*: perbedaan jarak antar objek.
2. *Surface color discontinuity*: perubahan warna permukaan.
3. *Illumination discontinuity*: perubahan pencahayaan.
4. *Surface orientation discontinuity*: perubahan arah bidang permukaan.

### 2.2.2. Jenis Edge Detection Yang Digunakan

1. Sobel

Sobel merupakan metode berbasis *gradient filter* yang menggunakan dua kernel konvolusi untuk mendeteksi perubahan intensitas pada arah horizontal dan vertikal. Sobel menghasilkan suatu peta tepi yang berdasarkan gradien lokal. Secara matematis, kernel Sobel dapat diilustrasikan sebagai:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Operator ini menghitung derivatif dan sekaligus melakukan perataan (smoothing) untuk mengurangi noise. Dalam implementasi, kode yang digunakan adalah sebagai berikut:

```
sobelx = cv2.Sobel(img, cv2.CV_64F, 1, 0,  
ksize=ksize)  
sobely = cv2.Sobel(img, cv2.CV_64F, 0, 1,  
ksize=ksize)  
sobel = np.sqrt(sobelx**2 + sobely**2)
```

Parameter:

1. Ksize = ukuran kernel (3,5,7)

Semakin besar ksize, efek smoothing akan semakin mendekat sehingga noise berkurang, namun detail tepi yang halus juga dapat hilang

2. Canny

Canny Edge Detector adalah metode yang lebih kompleks dan optimal yang dikembangkan oleh John Canny (1986). Tujuannya adalah mencapai tiga kriteria:

1. **Good detection:** menemukan semua tepi nyata dan mengabaikan noise.
2. **Good localization:** posisi tepi sedekat mungkin dengan tepi sebenarnya.
3. **Minimal response:** hanya satu piksel per tepi.

Berikut merupakan langkah-langkah untuk algoritma Canny:

1. Smoothing: citra difilter dengan *Gaussian filter* untuk mengurangi noise.
2. Gradient computation: hitung gradien  $G_x$ ,  $G_y$ , serta magnitude dan arah.
3. Non-maximum suppression: mempertahankan hanya piksel yang merupakan puncak lokal pada arah gradien.
4. Hysteresis thresholding menggunakan gunakan dua *threshold* (lower dan upper).
  - $G > \text{upper}$ : tepi kuat (*strong edge*).

- $G < \text{lower}$ : noise akan diabaikan.
- Di antara keduanya, *weak edge* akan disambungkan jika bersebelahan dengan tepi kuat.

Dalam implementasi, kode yang digunakan adalah sebagai berikut:

```
def apply_canny(img, lower=50, upper=150):
    return cv2.Canny(img, lower, upper)
```

Parameter:

1. lower dan upper : *thresholds* untuk tahap hysteresis
  - Nilai kecil → lebih sensitif terhadap detail halus, tapi bisa memunculkan *noise*.
  - Nilai besar → hanya mendeteksi tepi utama yang kuat.

### 2.2.3. Jenis Eksperimen Yang Dilakukan

Berikut merupakan jenis eksperimen yang dilakukan untuk melihat hasil *edge detection* menggunakan Sobel dan Canny:

Tabel 2.2 Jenis Eksperimen Edge Detection

Jenis Eksperimen	Metode (Filter)	Parameter Tetap	Parameter Variasi	Tujuan Eksperimen
Eksperimen 1	Sobel Edge Detection	Citra grayscale	ksize = [3, 7, 21]	Menganalisis pengaruh ukuran kernel Sobel terhadap ketajaman dan kehalusan hasil tepi.
Eksperimen 2	Canny Edge Detection	Citra grayscale, filter Gaussian internal Canny	threshold (lower, upper) = (30,100), (100,150), (20,400)	Melihat pengaruh kombinasi ambang bawah–atas terhadap sensitivitas deteksi tepi.

## 2.2.4. Hasil

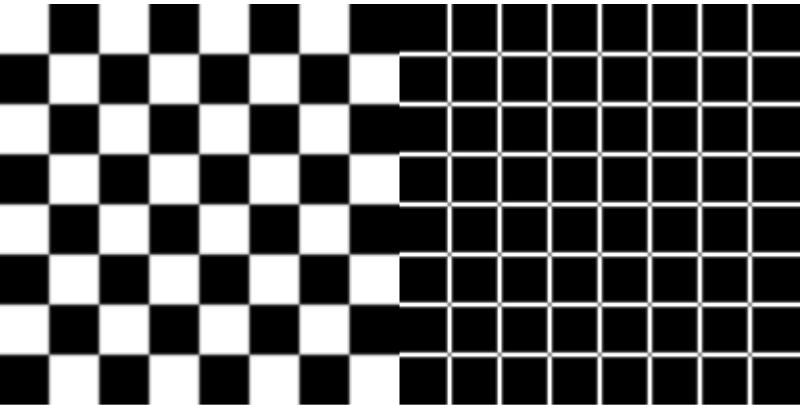
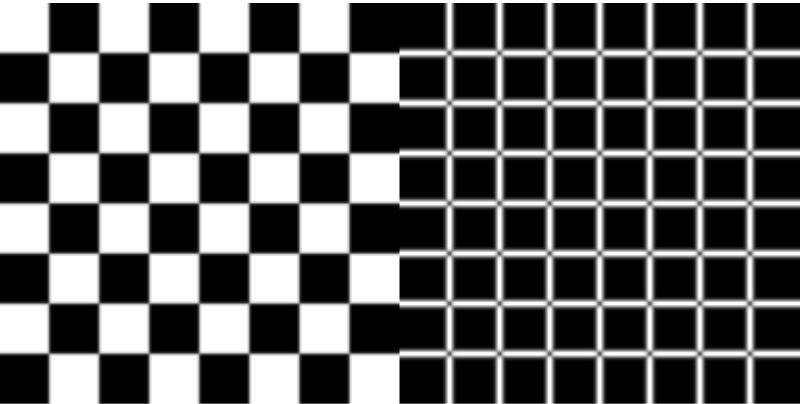
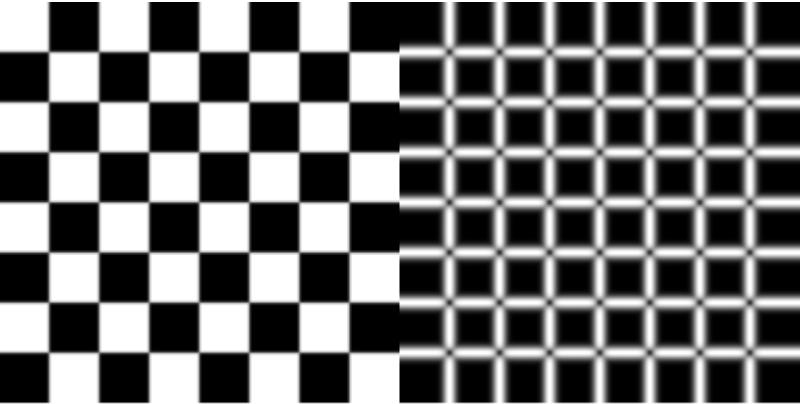
### 2.2.4.1. Eksperimen 1 : Sobel

Data Uji : Camera	
Parameter	Hasil
kernel =3	  <b>Sobel Edge (ksize=3)</b>
	<p>Gambar 2.19 Hasil Foto “Camera” Sobel I</p>
kernel =7	  <b>Sobel Edge (ksize=7)</b>
	<p>Gambar 2.20 Hasil Foto “Camera” Sobel II</p>
Kernel = 21	  <b>Sobel Edge (ksize=21)</b>

	Gambar 2.21 Hasil Foto “Camera” Sobel III
--	---

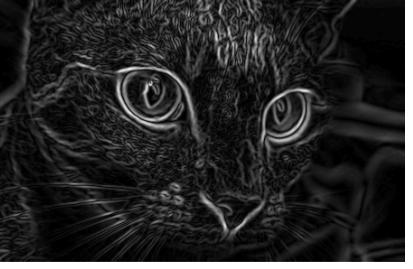
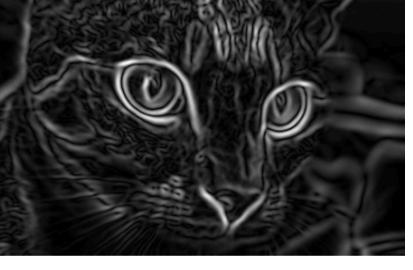
Data Uji : Coins	
Parameter	Hasil
kernel =3	 <p><b>Sobel Edge (ksize=3)</b></p> <p>Gambar 2.22 Hasil Foto “Coin” Sobel I</p>
kernel =7	 <p><b>Sobel Edge (ksize=7)</b></p> <p>Gambar 2.23 Hasil Foto “Coin” Sobel II</p>
Kernel = 21	 <p><b>Sobel Edge (ksize=21)</b></p> <p>Gambar 2.24 Hasil Foto “Coin” Sobel III</p>

#### Data Uji : Checkerboard

Parameter	Hasil
kernel =3	 <p><b>Sobel Edge (ksize=3)</b></p> <p>Gambar 2.25 Hasil Foto “Checkerboard” Sobel I</p>
kernel =7	 <p><b>Sobel Edge (ksize=7)</b></p> <p>Gambar 2.26 Hasil Foto “Checkerboard” Sobel II</p>
Kernel = 21	 <p><b>Sobel Edge (ksize=21)</b></p> <p>Gambar 2.27 Hasil Foto “Checkerboard” Sobel III</p>

**Data Uji : Astronaut**

Parameter	Hasil
kernel =3	 <p><b>Sobel Edge (ksize=3)</b></p> <p>Gambar 2.28 Hasil Foto “Astronaut” Sobel I</p>
kernel =7	 <p><b>Sobel Edge (ksize=7)</b></p> <p>Gambar 2.29 Hasil Foto “Astronaut” Sobel II</p>
Kernel = 21	 <p><b>Sobel Edge (ksize=21)</b></p> <p>Gambar 2.30 Hasil Foto “Astronaut” Sobel III</p>

<b>Data Uji : Chelsea</b>	
<b>Parameter</b>	<b>Hasil</b>
kernel =3	  <p><b>Sobel Edge (ksize=3)</b></p>
Gambar 2.31 Hasil Foto “Chelsea” Sobel I	
kernel =7	  <p><b>Sobel Edge (ksize=7)</b></p>
Gambar 2.32 Hasil Foto “Chelsea” Sobel II	
Kernel = 21	  <p><b>Sobel Edge (ksize=21)</b></p>
Gambar 2.33 Hasil Foto “Chelsea” Sobel III	

<b>Foto Pribadi</b>	
<b>Parameter</b>	<b>Hasil</b>

kernel =3	 	<b>Sobel Edge (ksize=3)</b>	Gambar 2.34 Hasil Foto Pribadi Sobel I
kernel =7	 	<b>Sobel Edge (ksize=7)</b>	Gambar 2.35 Hasil Foto Pribadi Sobel II
Kernel = 21	 	<b>Sobel Edge (ksize=21)</b>	Gambar 2.36 Hasil Foto Pribadi Sobel III

## Analisis

Pada ukuran kernel yang kecil (misalnya  $3 \times 3$ ), hasil deteksi tepi umumnya tampak tajam. Tepi-tepi halus dan detail kecil pada citra dapat terdeteksi dengan baik. Pada kernel size = 7, efek perataan (smoothing) pada citra juga meningkat. Deteksi tepi menjadi lebih lembut dan stabil, di mana tepi besar atau area dengan kontras kuat tetap teridentifikasi dengan jelas, sementara detail kecil mulai melebur. Pada kernel = 21, garis tepi semakin halus. Ukuran kernel yang besar menyebabkan proses perataan intensitas sebelum perhitungan gradien menjadi lebih kuat. Akibatnya,

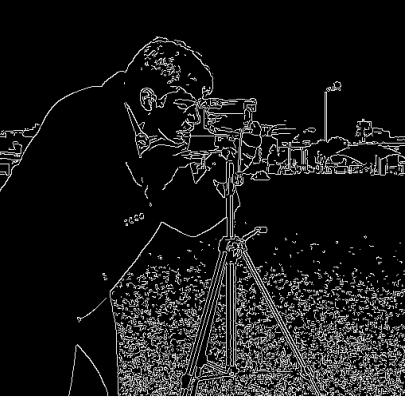
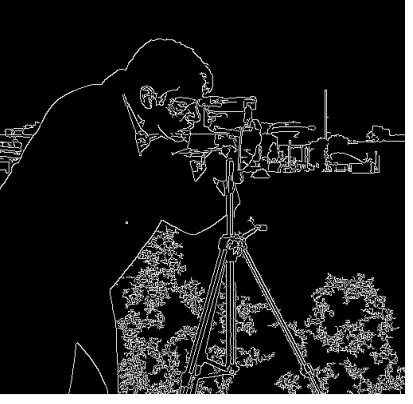
hanya struktur besar dan tepi dengan kontras tinggi yang masih terdeteksi secara jelas, sedangkan detail halus menghilang.

Tabel 2.3 Analisis Karakteristik Sobel

<b>Ukuran Kernel (ksize)</b>	<b>Karakteristik Hasil</b>	<b>Kelebihan</b>	<b>Kekurangan</b>	<b>Cocok Untuk</b>
Kecil (mis. 3×3)	Deteksi tepi sangat detail dan tajam; sensitivitas tinggi terhadap perubahan intensitas piksel.	Menangkap tepi halus dan detail kecil secara akurat.	Rentan terhadap noise dan tekstur halus ikut terdeteksi.	Analisis fitur kecil, citra beresolusi tinggi, atau tahap awal ekstraksi fitur.
Sedang (mis. 5×5 – 7×7)	Hasil deteksi seimbang antara ketajaman dan kehalusan.	Mengurangi noise tanpa menghilangkan terlalu banyak detail.	Beberapa tepi halus mulai melebur.	Citra umum dengan tingkat noise sedang.
Besar (mis. 15×15 – 21×21)	Tepi tampak halus dan kontur global lebih dominan; detail kecil hilang.	Menonjolkan batas besar dan struktur utama.	Detail kecil dan tekstur halus hilang akibat smoothing kuat.	Segmentasi objek besar atau analisis kontur global.

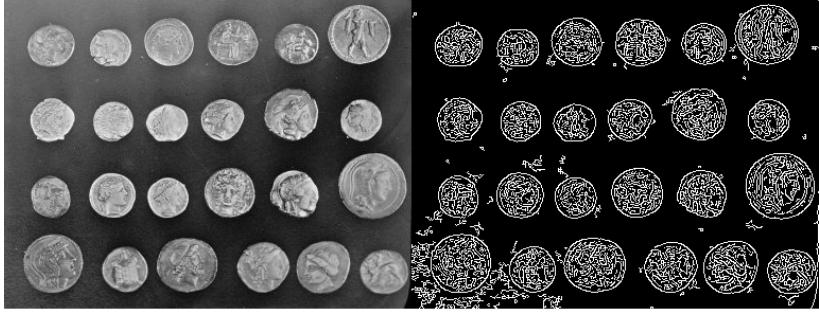
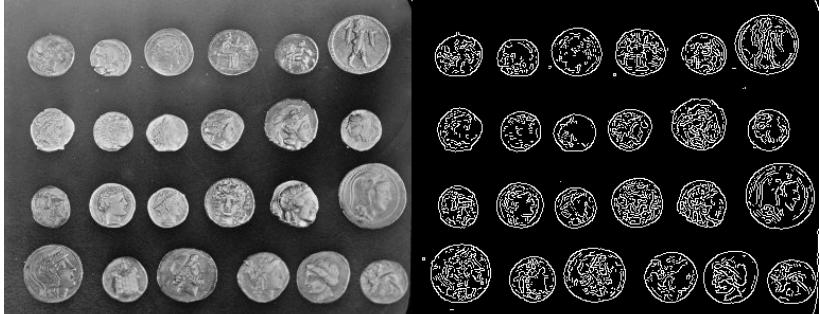
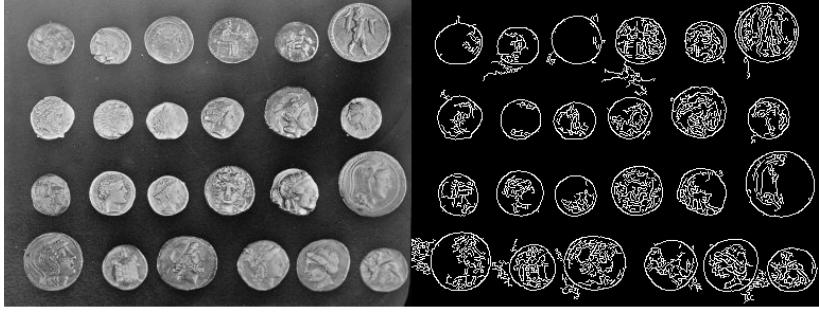
#### 2.2.4.2. Eksperimen 2: Canny

<b>Data Uji : Camera</b>	
<b>Parameter</b>	<b>Hasil</b>

Threshold = 30-100	 
Threshold = 100-150	 
Threshold = 20-400	 

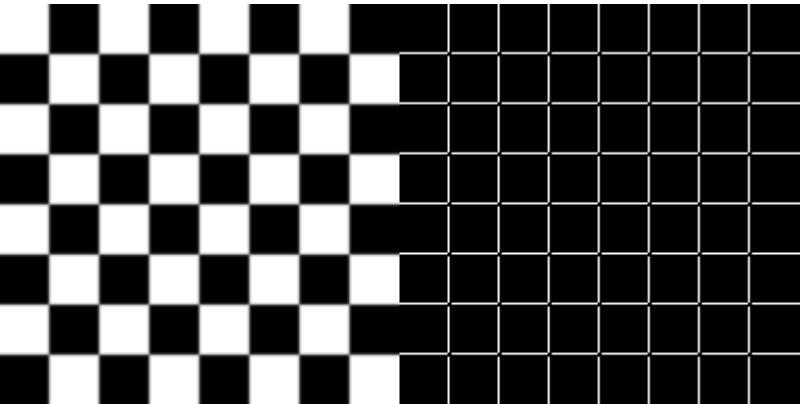
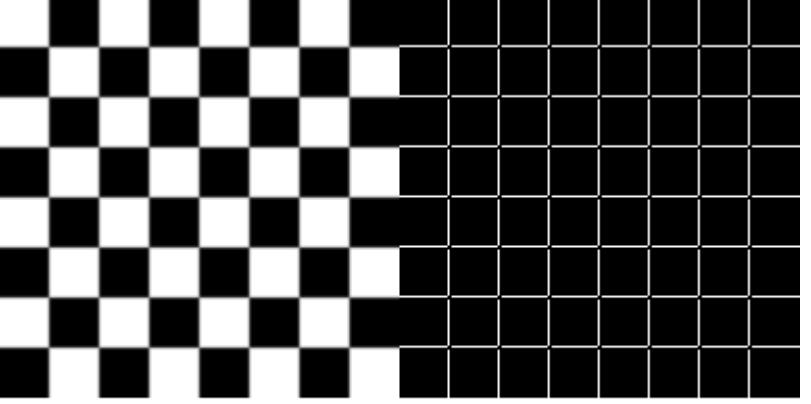
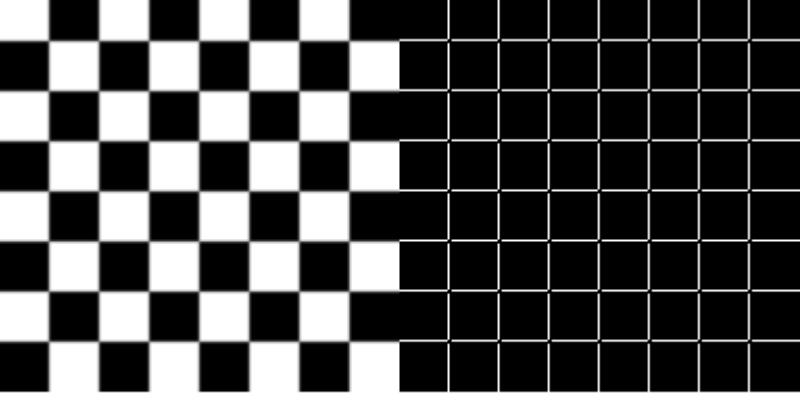
#### Data Uji : Coins

Parameter	Hasil

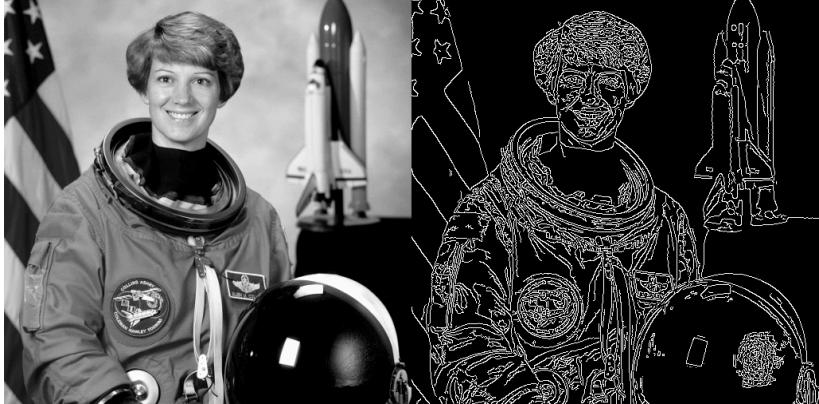
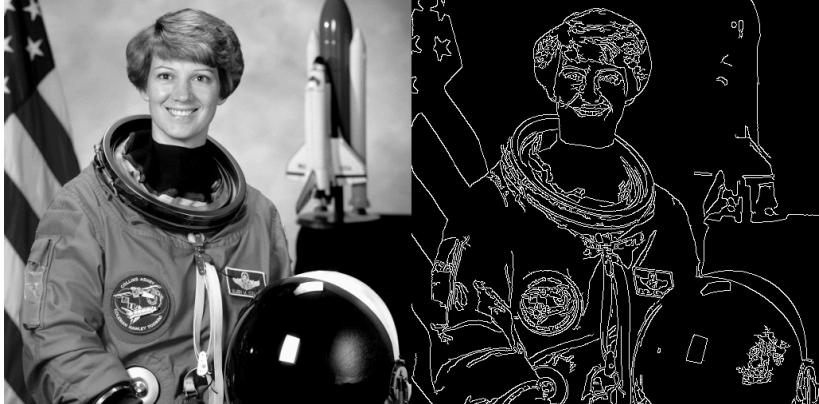
Threshold = 30-100	 <b>Canny Edge (lower=30, upper=100)</b>
Threshold = 100-150	 <b>Canny Edge (lower=100, upper=150)</b>
Threshold = 20-400	 <b>Canny Edge (lower=20, upper=400)</b>

Gambar 2.40 Hasil Foto "Coins" Canny I  
Gambar 2.41 Hasil Foto "Coins" Canny II  
Gambar 2.42 Hasil Foto "Coins" Canny III

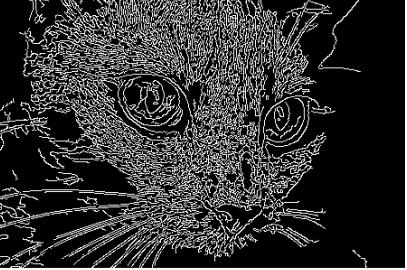
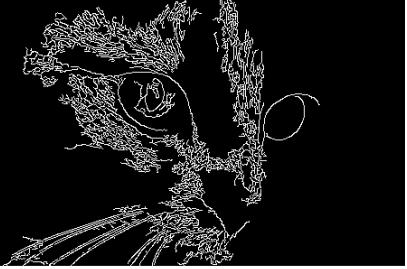
Data Uji : Checkerboard	
Parameter	Hasil

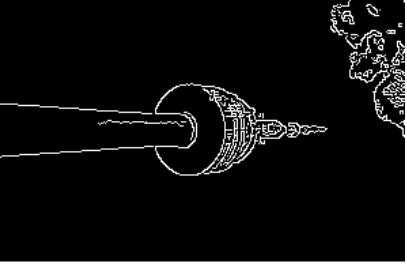
Threshold = 30-100	
	<p><b>Canny Edge (lower=30, upper=100)</b></p> <p>Gambar 2.43 Hasil Foto “Checkerboard” Canny I</p>
Threshold = 100-150	
	<p><b>Canny Edge (lower=100, upper=150)</b></p> <p>Gambar 2.44 Hasil Foto “Checkerboard” Canny II</p>
Threshold = 20-400	
	<p><b>Canny Edge (lower=20, upper=400)</b></p> <p>Gambar 2.45 Hasil Foto “Checkerboard” Canny III</p>

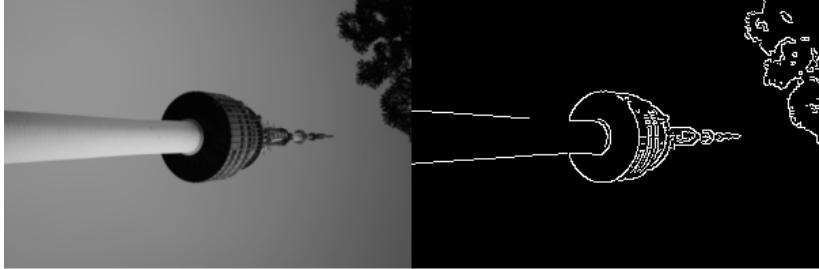
Data Uji : Astronaut
----------------------

Parameter	Hasil
Threshold = 30-100	 <p>Canny Edge (lower=30, upper=100)</p> <p>Gambar 2.46 Hasil Foto "Astronaut" Canny I</p>
Threshold = 100-150	 <p>Canny Edge (lower=100, upper=150)</p> <p>Gambar 2.47 Hasil Foto "Astronaut" Canny II</p>
Threshold = 20-400	 <p>Canny Edge (lower=20, upper=400)</p> <p>Gambar 2.48 Hasil Foto "Astronaut" Canny III</p>

Data Uji : Chelsea

Parameter	Hasil
Threshold = 30-100	  <p>Canny Edge (lower=30, upper=100)</p> <p>Gambar 2.49 Hasil Foto "Chelsea" Canny I</p>
Threshold = 100-150	  <p>Canny Edge (lower=100, upper=150)</p> <p>Gambar 2.50 Hasil Foto "Chelsea" Canny II</p>
Threshold = 20-400	  <p>Canny Edge (lower=20, upper=400)</p> <p>Gambar 2.51 Hasil Foto "Chelsea" Canny III</p>

Data Uji : Camera	
Parameter	Hasil
Threshold = 30-100	  <p>Canny Edge (lower=30, upper=100)</p>

	Gambar 2.52 Hasil Foto Pribadi Canny I
Threshold = 100-150	 <p><b>Canny Edge (lower=100, upper=150)</b></p>
Threshold = 20-400	 <p><b>Canny Edge (lower=20, upper=400)</b></p>
	Gambar 2.54 Hasil Foto Pribadi Canny III

### Analisis

Ketika nilai threshold rendah (misalnya lower=20, upper=400), sistem menjadi lebih sensitif terhadap perubahan intensitas kecil, sehingga tepi yang dihasilkan lebih banyak dan detail halus ikut terdeteksi. Namun, sensitivitas tinggi ini juga dapat menyebabkan munculnya tepi palsu (*false edges*) akibat noise atau variasi pencahayaan.

Pada nilai threshold tinggi (misalnya lower=100, upper=150), hanya perubahan intensitas yang kuat yang terdeteksi. Hasilnya menampilkan garis tepi yang lebih bersih, halus, dan terfokus pada kontur utama, tetapi beberapa detail kecil atau tepi dengan kontras rendah bisa terabaikan.

Sementara itu, penggunaan threshold menengah (misalnya lower=30, upper=100) memberikan hasil yang seimbang antara ketajaman dan kestabilan, di mana tepi utama masih jelas terlihat namun jumlah noise sudah berkurang.

Tabel 2.4 Analisis Karakteristik Canny

<b>Threshold (Lower–Upper)</b>	<b>Sensitivitas Deteksi</b>	<b>Kelebihan</b>	<b>Kekurangan</b>	<b>Cocok Untuk</b>
Rendah (mis. 20–400)	Sangat tinggi	Menangkap banyak tepi termasuk detail halus	Banyak tepi palsu dan noise	Citra dengan kontras rendah atau ingin eksplorasi detail kecil
Sedang (mis. 30–100)	Seimbang	Menjaga keseimbangan antara jumlah tepi dan kebersihan hasil	Beberapa tepi halus bisa melebur	Pengolahan citra umum dengan tingkat kontras sedang
Tinggi (mis. 100–150)	Rendah	Menghasilkan kontur utama yang bersih dan jelas	Kehilangan detail kecil dan tepi kontras rendah	Segmentasi bentuk utama atau analisis struktur besar

## 2.3. Feature Points

### 2.3.1. Pengertian Feature Points

*Feature points* atau *interest points* adalah titik-titik khas dalam citra yang memiliki karakteristik lokal yang unik dan mudah dikenali kembali, bahkan ketika citra mengalami transformasi seperti rotasi, translasi, perubahan skala, atau pencahayaan. Dalam visi komputer, fitur lokal digunakan untuk berbagai aplikasi seperti:

1. Penyambungan panorama (image stitching)
2. 3D reconstruction
3. Object recognition (pra-deep learning)
4. Motion tracking

Salah satu jenis *feature points* adalah *corner* (titik sudut). Titik sudut adalah area yang mengalami perubahan intensitas pada dua arah sekaligus. Untuk mengukur perubahan tersebut, digunakan pendekatan *auto-correlation function* yang menghitung perbedaan antara patch citra yang digeser sedikit dalam berbagai arah.

### 2.3.2. Jenis Corner Detector Yang Digunakan

1. Harris Corner Detector

Harris Corner Detector adalah metode klasik untuk mendeteksi titik-titik corner yang stabil. Berikut merupakan langkah-langkah Harris Corner Detector:

1. Hitung turunan intensitas  $I_x$  dan  $I_y$ .

2. Bentuk matriks struktur tensor:

$$M = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

3. Nilai eigen dari M ( $\lambda_1, \lambda_2$ ) menunjukkan tingkat perubahan pada dua arah:

- a. Keduanya besar  $\rightarrow$  corner
- b. Salah satu besar  $\rightarrow$  edge
- c. Keduanya kecil  $\rightarrow$  flat region

4. Skor “cornerness” dihitung dengan rumus:

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

di mana  $k \approx 0.04-0.06$

5. *Threshold* digunakan untuk memilih titik dengan nilai R tinggi (corner kuat), lalu dilakukan *non-maximal suppression* untuk menyimpan puncak lokal.

2. SIFT (Scale-Invariant Feature Transform)

SIFT adalah algoritma fitur yang invariant terhadap skala, rotasi, dan iluminasi. Berikut merupakan langkah-langkah SIFT:

1. Deteksi titik ekstrem (DoG): mencari titik maksimum dan minimum dalam ruang skala (Difference of Gaussian).
2. Eliminasi noise: menghapus titik dengan kontras rendah dan tepi tipis.
3. Estimasi orientasi: menentukan arah dominan gradien di sekitar titik fitur.
4. Deskripsi fitur: membuat deskriptor 128-dimensi dengan menggabungkan histogram orientasi gradien dalam  $4 \times 4$  sub-region.

Hasilnya berupa kumpulan keypoints dan deskriptor, yang bisa digunakan untuk mencocokkan titik antara dua gambar berbeda.

### 3. FAST (Features from Accelerated Segment Test)

FAST merupakan metode deteksi fitur yang sangat cepat dibandingkan Harris atau SIFT. Berikut merupakan langkah-langkah FAST:

1. Setiap piksel dibandingkan dengan lingkaran 16 piksel di sekitarnya (radius 3).
2. Jika terdapat  $\geq 12$  piksel berturut-turut yang lebih terang atau lebih gelap dari pusat (berdasarkan threshold), maka titik tersebut dianggap sebagai corner.
3. FAST tidak menghitung gradien atau skala, sehingga jauh lebih efisien untuk aplikasi real-time seperti video tracking.

#### 2.3.3. Jenis Eksperimen Yang Dilakukan

Berikut merupakan jenis eksperimen yang dilakukan untuk melihat hasil *feature points* menggunakan Harris, SIFT, dan FAST:

Tabel 2.5 Jenis Eksperimen Feature Points

Jenis Eksperimen	Metode (Feature Detector)	Tujuan Eksperimen
Eksperimen 1	Harris Corner Detector	Melihat pengaruh ambang deteksi (threshold) terhadap jumlah dan sebaran titik corner yang terdeteksi.

Eksperimen 2	SIFT (Scale-Invariant Feature Transform)	Mengevaluasi kemampuan SIFT dalam mendeteksi fitur dengan ukuran dan orientasi berbeda, serta menilai <i>response</i> rata-rata dan ukuran keypoint.
Eksperimen 3	FAST Feature Detector	Menganalisis sensitivitas FAST terhadap nilai threshold dan jumlah keypoint yang terdeteksi.

### 2.3.4. Hasil

#### 2.3.4.1. Eksperimen 1 : Harris Corner Detector

##### 1. Perbedaan Block Size

Data Uji : Camera	
Parameter	Hasil
Block size = 2 Kernel = 3 K = 0.04 Threshold = 0.01	 <p>Harris (1010 pts)</p>

Gambar 2.55 Hasil Foto “Camera” Harris (Block) I

Block size = 5 Kernel = 3 K = 0.04 Threshold = 0.01	 <b>Harris (5419 pts)</b>
Gambar 2.56 Hasil Foto “Camera” Harris (Block) II	

## Analisis

Pada percobaan dengan metode Harris Corner Detector, pengaruh perubahan nilai block size terlihat cukup signifikan terhadap jumlah dan kualitas titik sudut yang terdeteksi. Ketika block size bernilai kecil, yaitu 2, jumlah titik sudut yang terdeteksi relatif lebih sedikit (sekitar 1010 titik). Sebaliknya, pada block size yang lebih besar, yaitu 5, jumlah titik meningkat tajam hingga mencapai 5419 titik. Hal ini terjadi karena semakin besar nilai block size, semakin luas area yang digunakan untuk menghitung matriks kovarians gradien, sehingga lebih banyak variasi lokal yang dianggap sebagai sudut oleh algoritma.

Block size kecil (2) lebih sensitif terhadap perubahan intensitas halus dan mampu mendeteksi detail kecil pada citra, meskipun berpotensi melewatkannya beberapa sudut dengan kontras lemah. Sementara itu, block size besar (5) mendeteksi lebih banyak titik, termasuk titik-titik semu yang muncul di area bertekstur atau tepi yang kurang signifikan.

Pada block size 2, titik-titik merah hasil deteksi hanya muncul di area dengan perubahan intensitas yang benar-benar kuat. Namun, pada block size 5, titik-titik tersebut tampak jauh lebih banyak dan menyebar, bahkan di area yang secara visual tampak halus atau tidak memiliki perubahan intensitas signifikan.

### 2. Perbedaan Kernel Size

<b>Data Uji : Camera</b>
--------------------------

Parameter	Hasil
Kernel = 3 Block = 2 K = 0.04 Threshold = 0.01	 <p>Harris (1010 pts)</p> <p>Gambar 2.57 Hasil Foto “Camera” Harris (Kernel) I</p>
Kernel = 7 Block = 2 K = 0.04 Threshold = 0.01	 <p>Harris (909 pts)</p> <p>Gambar 2.58 Hasil Foto “Camera” Harris (Kernel) II</p>

### Analisis

Ketika digunakan kernel berukuran kecil (3), Harris mampu mendeteksi lebih banyak titik sudut, yaitu sekitar 1010 titik. Namun, saat ukuran kernel diperbesar menjadi 7, jumlah titik yang terdeteksi menurun menjadi 909 titik. Hal ini menunjukkan bahwa semakin besar ukuran kernel, semakin sedikit jumlah sudut yang berhasil dikenali, karena proses perataan (*smoothing*) yang lebih luas mengurangi sensitivitas terhadap perubahan intensitas lokal.

Kernel berukuran kecil (3) berfokus pada area lokal yang sempit sehingga lebih sensitif terhadap perubahan intensitas kecil dan mampu menangkap sudut-sudut tajam serta detail halus. Sebaliknya, kernel besar (7) melakukan perataan pada area

yang lebih luas sebelum menghitung gradien, sehingga hasilnya menjadi lebih stabil terhadap noise, tetapi sensitivitas terhadap detail kecil berkurang.

### 3. Perbedaan Nilai K

Data Uji : Camera	
Parameter	Hasil
Kernel = 3 Block = 2 K = 0.04 Threshold = 0.01	 <p>Harris (1010 pts)</p>
Kernel = 3 Block = 2 K = 0.06 Threshold = 0.01	 <p>Harris (771 pts)</p>

Gambar 2.59 Hasil Foto “Camera” Harris (K) I

Gambar 2.60 Hasil Foto “Camera” Harris (K) II

### Analisis

Ketika digunakan  $K = 0.04$ , jumlah titik sudut yang terdeteksi mencapai sekitar 1010 titik, sedangkan saat  $K$  meningkat menjadi 0.06, jumlah tersebut menurun menjadi 771 titik. Penurunan ini menunjukkan bahwa semakin besar nilai  $K$ , detektor Harris menjadi lebih ketat atau selektif dalam menentukan suatu titik sebagai sudut, karena hanya area dengan perbedaan kontras yang kuat yang akan diakui sebagai corner.

Nilai K kecil (0.04) membuat sistem menjadi lebih sensitif, sehingga lebih banyak titik yang dianggap sebagai corner, termasuk titik-titik lemah atau ambigu yang mungkin berasal dari noise. Sebaliknya, K besar (0.06) membuat hanya mendeteksi sudut yang benar-benar kuat dan stabil terhadap variasi pencahayaan maupun gangguan kecil pada citra.

#### 4. Perbedaan Threshold Ratio

Data Uji : Camera	
Parameter	Hasil
Kernel = 3 Block = 2 K = 0.04 Threshold = 0.001	 <p>Harris (6652 pts)</p>
Kernel = 3 Block = 2 K = 0.04 Threshold = 0.5	 <p>Harris (6 pts)</p>

Gambar 2.61 Hasil Foto “Camera” Harris (Threshold) I

Gambar 2.62 Hasil Foto “Camera” Harris (Threshold) II

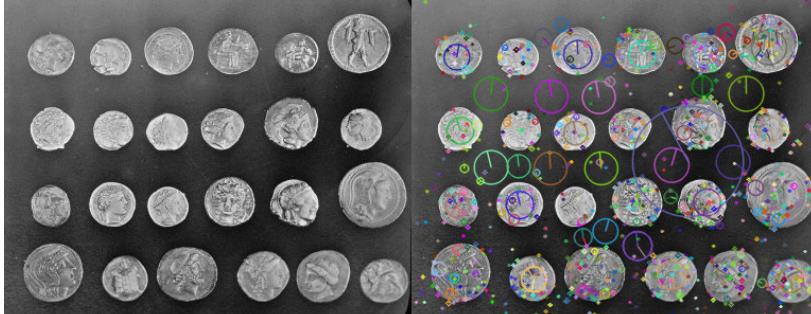
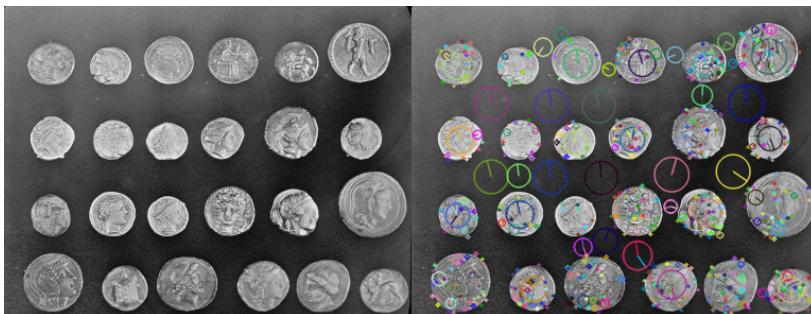
#### Analisis

Ketika threshold bernilai rendah (0.001), jumlah titik yang terdeteksi meningkat drastis hingga mencapai 6652 titik, dan titik-titik tersebut tersebar hampir di seluruh area gambar. Sebaliknya, saat threshold dinaikkan menjadi 0.5, jumlah titik

sudut yang terdeteksi menurun secara ekstrem hingga hanya 6 titik saja. Hal ini menunjukkan bahwa semakin kecil nilai threshold, semakin banyak piksel yang dianggap sebagai corner, bahkan termasuk titik-titik yang sebenarnya bukan fitur signifikan.

Threshold rendah (0.001) hampir seluruh area dengan sedikit perubahan intensitas terdeteksi sebagai corner, menghasilkan fenomena over-detection. Banyak titik palsu (false corners) muncul di area rumput, jas, atau bayangan yang sebenarnya tidak memiliki struktur sudut nyata. Sebaliknya, pada threshold tinggi (0.5) hanya area dengan kontras tinggi yang lolos. Namun hal ini menyebabkan hilangnya banyak titik relevan, sehingga terjadi under-detection.

#### 2.3.4.2. Eksperimen 2: SIFT

Data Uji : Coins	
Parameter	Hasil
contrastThresh old= 0.01 edgeThreshold = 10 nOctaveLayers = 3 sigma=1.6	 <p><b>SIFT (926 pts)</b></p>
contrastThresh old= 0.04 edgeThreshold = 10 nOctaveLayers = 3 sigma=1.6	 <p><b>SIFT (655 pts)</b></p>

Gambar 2.63 Hasil Foto “Coins” SIFT I

Gambar 2.64 Hasil Foto “Coins” SIFT II

<pre> contrastThresh old= 0.08 edgeThreshold = 10 nOctaveLayers = 3 sigma=1.6 </pre>	<p><b>SIFT (316 pts)</b></p>
Gambar 2.65 Hasil Foto "Coins" SIFT III	

<b>Foto Pribadi</b>	
<b>Parameter</b>	<b>Hasil</b>
<pre> contrastThresh old= 0.01 edgeThreshold = 10 nOctaveLayers = 3 sigma=1.6 </pre>	<p><b>SIFT (123 pts)</b></p>
Gambar 2.66 Hasil Foto Pribadi SIFT I	
<pre> contrastThresh old= 0.04 edgeThreshold = 10 nOctaveLayers = 3 sigma=1.6 </pre>	<p><b>SIFT (98 pts)</b></p>
Gambar 2.67 Hasil Foto Pribadi SIFT II	

contrastThresh old= 0.08 edgeThreshold = 10 nOctaveLayers = 3 sigma=1.6	<p><b>SIFT (46 pts)</b></p>
Gambar 2.68 Hasil Foto Pribadi SIFT III	

### Analisis

Ketika nilai contrastThreshold rendah, yaitu 0.01, jumlah fitur yang terdeteksi sangat banyak, mencapai 926 titik, dan tersebar hampir di seluruh permukaan koin. Saat nilai contrastThreshold dinaikkan menjadi 0.04, jumlah fitur menurun menjadi 655 titik, dan ketika dinaikkan lagi menjadi 0.08, hanya tersisa sekitar 316 titik fitur. Hal ini menunjukkan bahwa semakin tinggi nilai contrastThreshold, semakin sedikit titik fitur yang dianggap signifikan karena detektor menjadi lebih selektif terhadap perubahan kontras.

Pada threshold rendah (0.01) terlihat banyak keypoints yang muncul bahkan pada area datar atau bayangan, sehingga terjadi over-detection, di mana beberapa titik muncul di bagian koin yang tidak terlalu memiliki struktur jelas. Pada threshold sedang (0.04), titik-titik mulai lebih terfokus di area yang memiliki pola dan tekstur kuat pada permukaan koin. Sementara pada threshold tinggi (0.08), hanya fitur-fitur utama seperti tepi ukiran yang tetap terdeteksi, menandakan terjadinya under-detection, namun dengan keunggulan stabil terhadap gangguan noise dan variasi pencahayaan.

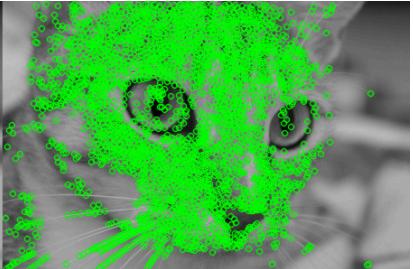
Tabel 2.6 Hasil Statistik SIFT

Image	Threshold	Num Features	Mean Response	Mean Size	Mean Angle	Total Response
coins	<b>0.01</b>	<b>926</b>	0.0229	4.01	181.38	21.25

coins	<b>0.04</b>	<b>655</b>	0.0297	4.48	181.44	19.44
coins	<b>0.08</b>	<b>316</b>	0.0400	5.98	190.38	12.64
personal-2	<b>0.01</b>	<b>123</b>	0.0242	4.69	173.34	—
personal-2	<b>0.04</b>	<b>98</b>	0.0279	5.19	172.82	2.73
personal-2	<b>0.08</b>	<b>46</b>	0.0362	5.79	171.72	1.66

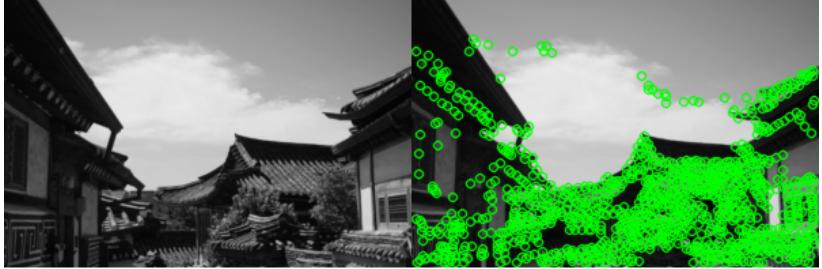
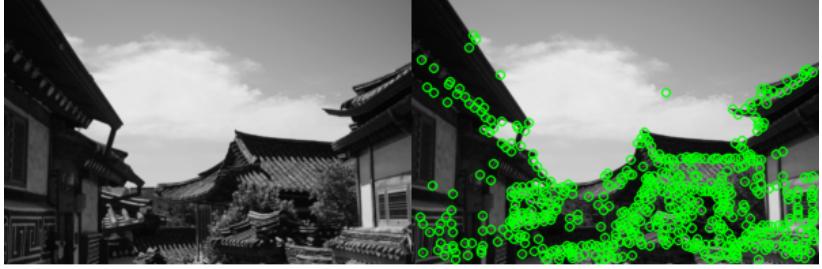
### 2.3.4.3. Eksperimen 3 : FAST Feature Detector

#### 1. Perbedaan Threshold

Data Uji : Chelsea	
Parameter	Hasil
Threshold = 10 NonMax Suppresion = yes	  <p><b>FAST (3037 pts)</b></p>
Threshold = 25 NonMax Suppresion = yes	  <p><b>FAST (469 pts)</b></p>

	Gambar 2.70 Hasil Foto "Chelsea" FAST II
Threshold = 50 NonMax Suppresion = yes	 <p>FAST (19 pts)</p>

Gambar 2.71 Hasil Foto "Chelsea" FAST III

Foto Pribadi	
Parameter	Hasil
Threshold = 10 NonMax Suppresion = yes	 <p>FAST (1086 pts)</p>
Threshold = 25 NonMax Suppresion = yes	 <p>FAST (553 pts)</p>

Gambar 2.72 Hasil Foto Pribadi FAST I

Gambar 2.73 Hasil Foto Pribadi FAST II



### Analisis

Pada threshold rendah (10), jumlah titik fitur yang muncul sangat banyak pada kedua gambar. Hampir seluruh area bertekstur atau memiliki tepi berhasil terdeteksi. Hal ini menunjukkan bahwa FAST memiliki sensitivitas yang sangat tinggi terhadap perubahan intensitas kecil ketika nilai threshold rendah, sehingga menghasilkan deteksi fitur yang padat namun berpotensi mengandung banyak titik tidak relevan.

Ketika threshold dinaikkan menjadi 25, jumlah titik fitur menurun drastis, hanya tersisa sekitar 15–25% dari jumlah awal. Titik-titik yang terdeteksi mulai terkonsentrasi pada area dengan kontras tinggi. Pada threshold tinggi (50), jumlah titik fitur berkurang secara ekstrem, hanya menyisakan puluhan atau bahkan belasan titik saja. Deteksi hanya terjadi pada area dengan kontras yang benar-benar tajam, seperti tepi keras atau sudut bangunan. Meskipun hasilnya tampak lebih bersih dan terfokus, peningkatan threshold yang terlalu besar menyebabkan kehilangan banyak detail penting yang seharusnya juga dapat menjadi fitur signifikan.

Tabel 2.7 Hasil Statistik FAST

Image	Thre shold	Num Features	Mean Response	Mean Size	Mean Angle	Total Response
chelsea	10	3037	17.73	7.0	-1.0	53846.0

chelsea	25	469	32.20	7.0	-1.0	15100.0
chelsea	50	19	59.84	7.0	-1.0	1137.0
personal	10	1780	0.00	7.0	-1.0	0.0
personal	25	553	48.28	7.0	-1.0	26700.0
personal-	50	177	78.15	7.0	-1.0	13832.0

## 2. NonMax Suppression

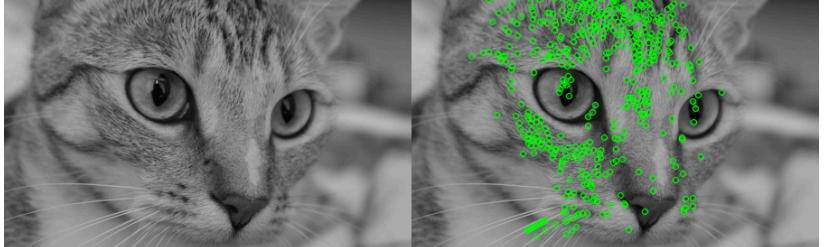
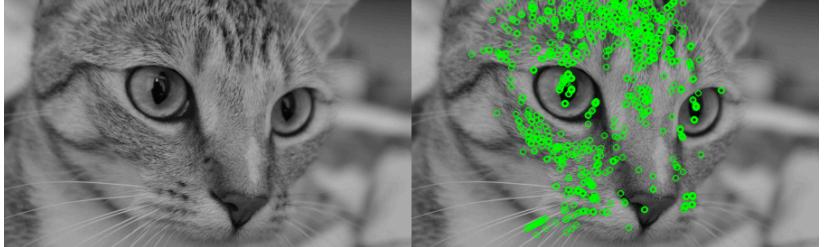
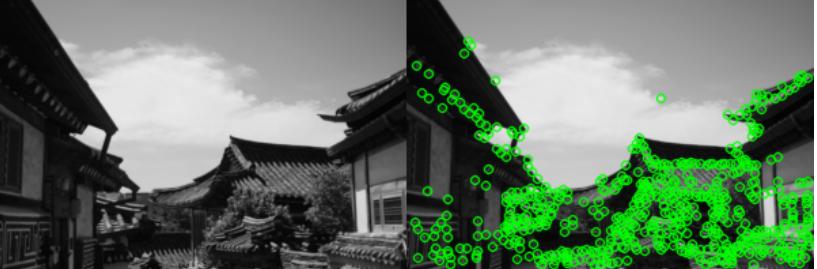
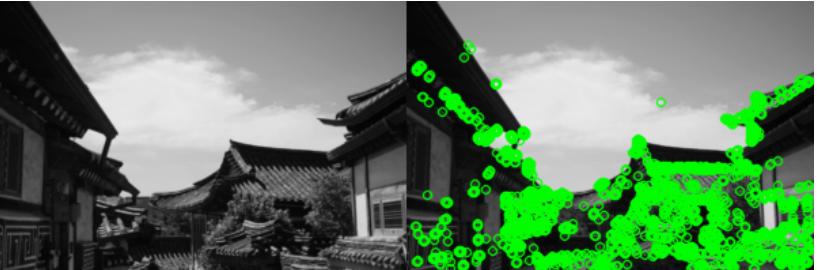
Data Uji : Astronaut	
Parameter	Hasil
Threshold = 25 NonMax Supresion = yes	 <b>FAST (469 pts)</b> Gambar 2.75 Hasil Foto "Chelsea" FAST (suppression) I
Threshold = 25 NonMax Supresion = no	 <b>FAST (898 pts)</b> Gambar 2.76 Hasil Foto "Chelsea" FAST (suppression) II

Foto Pribadi	
Parameter	Hasil

Threshold = 25 NonMax Supresion = yes	 <p><b>FAST (553 pts)</b></p>
	<p>Gambar 2.77 Hasil Foto Pribadi FAST (suppression) I</p>
Threshold = 25 NonMax Supresion = no	 <p><b>FAST (1780 pts)</b></p>
	<p>Gambar 2.78 Hasil Foto Pribadi FAST (suppression) II</p>

## Analisis

Saat NMS diaktifkan (Yes), jumlah titik fitur yang dihasilkan jauh lebih sedikit, sekitar setengah atau bahkan kurang dari jumlah titik ketika NMS tidak digunakan. Pada citra Chelsea, detektor hanya menemukan sekitar 469 titik, sedangkan pada citra Temple sebanyak 553 titik. Sebaliknya, ketika NMS dinonaktifkan (No), jumlah titik meningkat drastis hingga hampir dua sampai tiga kali lipat, yaitu 898 titik pada Chelsea dan 1780 titik pada Temple. Hal ini menunjukkan bahwa NMS berfungsi sebagai proses penyaringan yang menyaringkkan titik-titik berdekatan dengan respons lemah, hanya mempertahankan titik dengan nilai corner tertinggi pada area tertentu.

## 2.4. Transformasi Geometri

### 2.4.1. Pengertian Transformasi Geometri

Transformasi geometri adalah proses mengubah posisi, orientasi, atau bentuk suatu citra tanpa mengubah nilai intensitas pikselnya .Dalam konteks *Computer Vision*, transformasi ini digunakan untuk memetakan koordinat titik pada citra asal ke koordinat baru pada citra hasil. Secara matematis transformasi geometri dapat dituliskan sebagai:

$$p' = T(p)$$

di mana  $T$  adalah fungsi transformasi dan  $p'$  adalah posisi hasil dari titik awal  $p$ .

Menurut teori *Camera Geometry*, kamera dapat dipandang sebagai sistem transformasi proyektif yang memproyeksikan dunia 3D ke bidang gambar 2D. Dengan demikian, transformasi geometri seperti affine dan perspective menjadi dasar dalam mensimulasikan proyeksi kamera nyata, rotasi, translasi, dan distorsi pandangan.

#### 2.4.2. Jenis Transformasi Geometri Yang Digunakan

##### 1. Transformasi Affine

Affine transformation merupakan bentuk umum dari transformasi linier yang banyak digunakan dalam pengolahan citra. Transformasi ini mempertahankan garis lurus dan paralelisme antar garis, namun tidak selalu mempertahankan jarak maupun sudut. Dengan kata lain, bentuk citra dapat berubah ukuran atau orientasi, tetapi susunan spasialnya tetap konsisten. Secara matematis, affine transformation dinyatakan dalam bentuk matriks:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Transformasi ini merupakan kombinasi dari translasi, rotasi, scaling, dan shear, di mana:

- a. Translasi menggeser posisi citra
- b. Rotasi memutar citra terhadap titik tertentu
- c. Scaling memperbesar atau memperkecil ukuran citra
- d. Shear menggeser bagian citra secara miring

##### 2. Transformasi Perspektif

Transformasi Perspektif, atau sering disebut juga *projective transformation (homography)*, adalah jenis transformasi geometri yang meniru efek pandangan kamera nyata terhadap dunia tiga dimensi (3D). Transformasi ini memungkinkan suatu bidang citra mengalami perubahan perspektif, sehingga garis-garis paralel pada dunia nyata tampak bertemu di satu titik

(vanishing point) ketika diproyeksikan ke bidang gambar. Transformasi Perspektif tetap mempertahankan setiap garis lurus setelah transformasi. Secara matematis, perspektif dinyatakan dalam bentuk matriks  $3 \times 3$  yang bekerja dalam sistem koordinat homogen:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Hasil akhir kemudian dinormalisasi menjadi koordinat kartesian:

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}, \quad y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

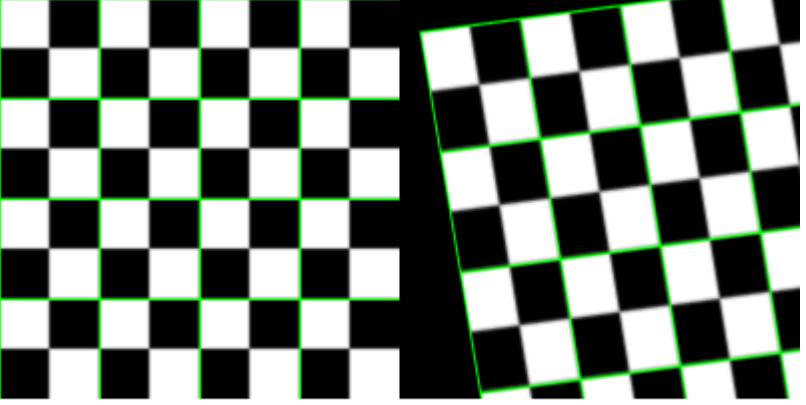
#### 2.4.3. Jenis Eksperimen Yang Dilakukan

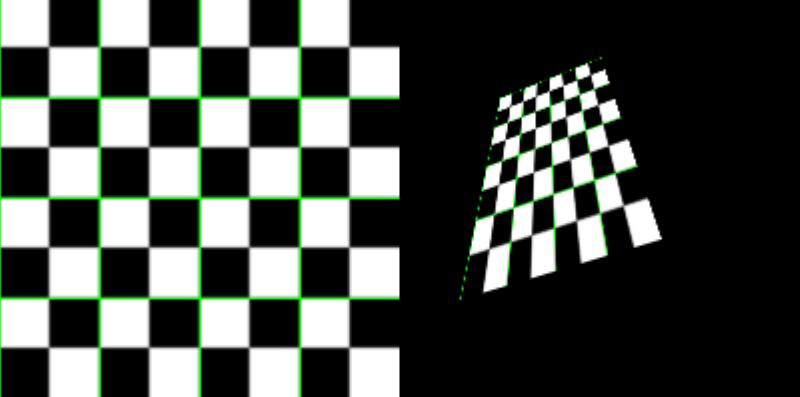
Berikut merupakan jenis eksperimen yang dilakukan untuk melihat hasil transformasi geometri affine dan perspektif:

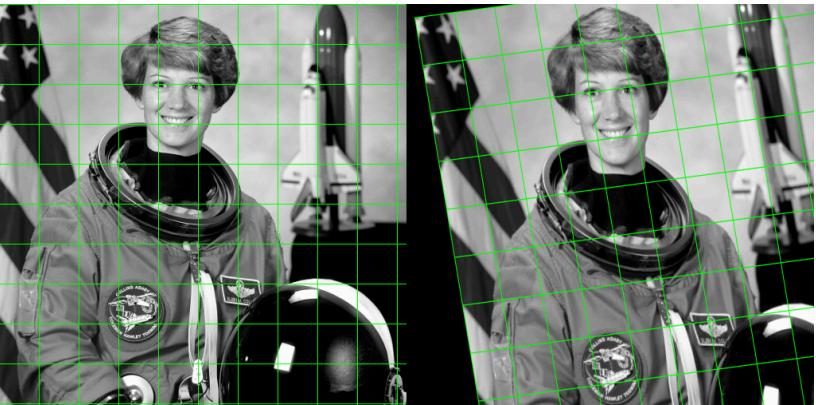
Tabel 2.8 Jenis Eksperimen Transformasi Geometri

Jenis Eksperimen	Metode (Transformasi)	Parameter Tetap	Tujuan Eksperimen
Eksperimen 1	Affine Transformation	Titik asal: pts1 = [[50,50],[200,50],[50,200]]	Mengamati efek kombinasi rotasi, translasi, dan scaling pada citra serta memastikan garis paralel tetap terjaga.
Eksperimen 2	Perspective Transformation	Titik awal: empat sudut gambar ((0,0), (w,0), (0,h), (w,h))	Melihat efek proyeksi kamera (hilangnya paralelisme dan munculnya vanishing point).

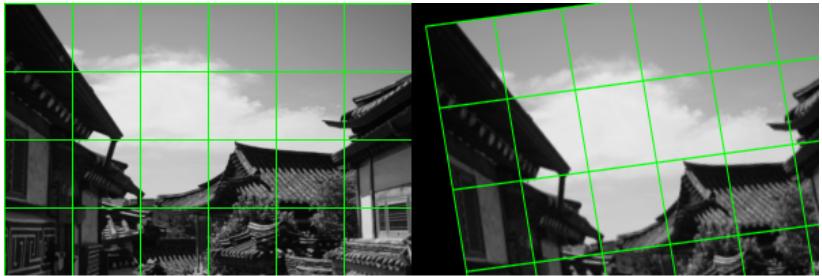
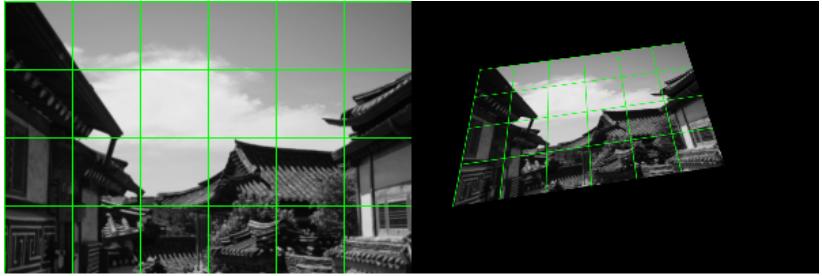
#### 2.4.4. Hasil

Data Uji : Checkerboard							
Parameter	Hasil						
Affine	 <p><b>checkerboard: Affine Transform (Before)</b></p> <p>Gambar 2.79 Hasil Foto “Checkerboard” Affine</p>						
<p><b>Affine Matrix – chelsea</b></p> <table style="margin-left: auto; margin-right: auto;"> <tr> <td>1.0000</td> <td>0.2000</td> <td>10.0000</td> </tr> <tr> <td>-0.1333</td> <td>1.2000</td> <td>16.6667</td> </tr> </table>		1.0000	0.2000	10.0000	-0.1333	1.2000	16.6667
1.0000	0.2000	10.0000					
-0.1333	1.2000	16.6667					
<p>Gambar 2.80 Hasil Foto “Checkerboard” Affine Matrix</p>							

Perspective	 <p><b>checkerboard: Perspective Transform (Before   After)</b></p> <p>Gambar 2.81 Hasil Foto “Checkerboard” Perspective</p>									
	<p><b>Perspective Matrix – checkerboard</b></p> <table style="margin-left: 40px;"> <tbody> <tr> <td>0.2259</td> <td>-0.1736</td> <td>50.0000</td> </tr> <tr> <td>-0.1081</td> <td>0.1370</td> <td>50.0000</td> </tr> <tr> <td>-0.0003</td> <td>-0.0024</td> <td>1.0000</td> </tr> </tbody> </table> <p>Gambar 2.82 Hasil Foto “Checkerboard” Perspective Matrix</p>	0.2259	-0.1736	50.0000	-0.1081	0.1370	50.0000	-0.0003	-0.0024	1.0000
0.2259	-0.1736	50.0000								
-0.1081	0.1370	50.0000								
-0.0003	-0.0024	1.0000								

<b>Data Uji : Astronaut</b>	
<b>Parameter</b>	<b>Hasil</b>
Affine	 <p><b>astronaut: Affine Transform (Before   After)</b></p>

	Gambar 2.83 Hasil Foto “Astronaut” Affine									
	<p><b>Affine Matrix – astronaut</b></p> <table> <tbody> <tr> <td>1.0000</td> <td>0.2000</td> <td>10.0000</td> </tr> <tr> <td>-0.1333</td> <td>1.2000</td> <td>16.6667</td> </tr> </tbody> </table>	1.0000	0.2000	10.0000	-0.1333	1.2000	16.6667			
1.0000	0.2000	10.0000								
-0.1333	1.2000	16.6667								
	Gambar 2.84 Hasil Foto “Astronaut” Affine Matrix									
Perspective	 <p><b>astronaut: Perspective Transform (Before   After)</b></p>									
	Gambar 2.85 Hasil Foto “Astronaut” Perspective									
	<p><b>Perspective Matrix – astronaut</b></p> <table> <tbody> <tr> <td>0.7211</td> <td>-0.0463</td> <td>50.0000</td> </tr> <tr> <td>-0.0382</td> <td>0.6955</td> <td>50.0000</td> </tr> <tr> <td>0.0000</td> <td>-0.0002</td> <td>1.0000</td> </tr> </tbody> </table>	0.7211	-0.0463	50.0000	-0.0382	0.6955	50.0000	0.0000	-0.0002	1.0000
0.7211	-0.0463	50.0000								
-0.0382	0.6955	50.0000								
0.0000	-0.0002	1.0000								
	Gambar 2.86 Hasil Foto “Astronaut” Perpective Matrix									

Foto Pribadi	
Parameter	Hasil
Affine	 <p><b>personal-1: Affine Transform (Before   After)</b></p>
<b>Affine Matrix – personal-1</b>	
	$\begin{matrix} 1.0000 & 0.2000 & 10.0000 \\ -0.1333 & 1.2000 & 16.6667 \end{matrix}$
Perspective	 <p><b>personal-1: Perspective Transform (Before   After)</b></p>
	<p>Gambar 2.87 Hasil Foto Pribadi Affine</p> <p>Gambar 2.88 Hasil Foto Pribadi Affine Matrix</p> <p>Gambar 2.89 Hasil Foto Pribadi Perspective</p>

Perspective Matrix – personal-1
0.5194 -0.1388 50.0000
-0.0642 0.3111 50.0000
0.0001 -0.0013 1.0000

Gambar 2.90 Hasil Foto Pribadi Perspective Matrix

### Analisis

Pada transformasi Affine, perubahan yang terjadi bersifat linear sehingga garis lurus tetap lurus dan kesejajaran antar garis tetap terjaga, meskipun ukuran dan sudutnya dapat berubah. Hal ini dapat dilihat pada hasil uji citra Coins, di mana pola grid masih tampak sejajar dan teratur meskipun citra mengalami sedikit rotasi, translasi, penskalaan, serta shear. Efek yang muncul tampak seperti kemiringan halus tanpa adanya distorsi ruang, sehingga affine cocok digunakan untuk koreksi posisi atau rotasi sederhana.

Pada transformasi Perspective, perubahan yang terjadi lebih kompleks karena mempertimbangkan proyeksi pandangan kamera terhadap bidang objek. Transformasi ini memungkinkan garis sejajar pada citra asli tampak bertemu pada titik hilang (vanishing point), sehingga memberikan efek seolah-olah citra memiliki kedalaman atau dimensi tiga. Pada contoh citra Coins, hasil perspective menunjukkan bagian kanan atas tampak lebih jauh dan mengalami penyempitan, meniru efek perubahan sudut pandang kamera.

## BAB III

### Komparasi & Refleksi Pribadi

#### 3.1. Image Filtering

Pada fitur Image Filtering, percobaan menunjukkan bahwa pemilihan jenis filter dan parameter sangat memengaruhi hasil smoothing serta ketajaman tepi citra.

##### 1. Gaussian Filter

Filter Gaussian menghasilkan efek blur yang halus dan natural, di mana peningkatan nilai sigma dan kernel size memperluas area perataan intensitas sehingga detail kecil menghilang.

- Pada  $\sigma$  kecil ( $0.5\text{--}2$ ), efek blur ringan; detail citra seperti tekstur permukaan atau garis halus masih terlihat jelas.
- Pada  $\sigma$  besar ( $\geq 5$ ), area blur meluas, detail halus hilang, dan batas antar objek melebur.

Parameter kernel size juga berperan penting. Meskipun sigma tetap, kernel yang lebih besar (mis.  $15\times 15$  atau  $21\times 21$ ) memperluas area rata-rata, menyebabkan smoothing lebih kuat dan citra tampak lembut tanpa tepi tajam.

**Hasil Refleksi:** Sigma berpengaruh terhadap intensitas blur, sedangkan kernel menentukan area penyebaran blur. Kombinasi besar keduanya dapat menyebabkan hilangnya detail penting.

##### 2. Median Filter

Median filter bekerja secara non-linear dengan mengganti nilai piksel menggunakan median di sekitarnya.

- Pada kernel kecil ( $3\times 3\text{--}5\times 5$ ), noise kecil seperti *salt and pepper* dapat dihilangkan tanpa mengubah bentuk objek.
- Pada kernel besar ( $\geq 13\times 13$ ), citra menjadi terlalu rata, tekstur hilang, dan muncul efek *posterization* (bidang datar abu-abu besar).

**Hasil Refleksi:** Median efektif menghilangkan noise impulsif tanpa kehilangan tepi, tetapi parameter terlalu besar menyebabkan hilangnya detail dan ketajaman visual.

Median filter lebih efektif dalam menghapus *salt and pepper noise* tanpa banyak merusak tepi objek, meskipun pada kernel besar citra menjadi tampak seperti blok-blok datar. Gaussian lebih unggul untuk menghaluskan noise global dan menjaga tekstur.

### 3.2. Edge Detection

Fitur Edge Detection memperlihatkan hubungan langsung antara parameter detektor dengan sensitivitas hasil tepi.

#### 1. Sobel

Metode Sobel menggunakan gradien intensitas dalam arah horizontal dan vertikal.

- Kernel kecil ( $3 \times 3$ ) menghasilkan tepi yang tajam dan detail tinggi, cocok untuk mendeteksi garis halus atau struktur kecil.
- Kernel sedang ( $5 \times 5 - 7 \times 7$ ) memberikan hasil seimbang antara ketajaman dan stabilitas, mengurangi noise tanpa kehilangan banyak detail.
- Kernel besar ( $\geq 15 \times 15$ ) menyebabkan smoothing kuat sehingga hanya tepi besar yang terlihat, sedangkan detail halus menghilang.

**Hasil Refleksi:** Semakin besar kernel, semakin kuat perataan yang dilakukan sebelum perhitungan gradien, sehingga sensitivitas menurun tetapi hasil lebih bersih.

#### 2. Canny

Metode Canny memberikan hasil paling optimal karena menggabungkan smoothing Gaussian, perhitungan gradien, dan hysteresis thresholding. Parameter threshold bawah (lower) dan threshold atas (upper) sangat memengaruhi hasil:

- Threshold rendah (mis. 20–400): sistem sangat sensitif, mendeteksi banyak detail halus tetapi rawan noise dan tepi palsu.

- Threshold sedang (30–100): menghasilkan keseimbangan antara jumlah tepi dan kebersihan hasil.
- Threshold tinggi (100–150): hanya tepi kuat yang terdeteksi, hasil bersih namun beberapa detail kontras rendah hilang.

**Hasil Refleksi:** Canny memberikan kontrol lebih besar terhadap sensitivitas deteksi tepi melalui dua ambang batas. Pengaturan threshold menentukan apakah fokus analisis lebih pada struktur besar atau detail kecil.

### 3.3. Feature Points

Pada eksperimen *Feature Points*, terdapat tiga metode yang diuji, Harris, SIFT, dan FAST.

#### 1. Harris Corner Detector

Harris sensitif terhadap perubahan intensitas dua arah dan parameter-parameter berikut berpengaruh besar:

- Block size: semakin besar nilainya (mis. 5 dibanding 2), area analisis semakin luas dan jumlah titik meningkat, namun beberapa di antaranya menjadi *false corners* di area bertekstur.
- Kernel size: kernel besar menurunkan sensitivitas karena perataan yang berlebihan, sehingga sudut halus terlewati.
- Nilai K (0.04–0.06): K kecil membuat detektor lebih sensitif terhadap perubahan intensitas, sedangkan K besar membuatnya lebih selektif dan stabil.
- Threshold ratio: nilai rendah (0.001) menimbulkan *over-detection*, sedangkan nilai tinggi (0.5) menimbulkan *under-detection*.

**Hasil Refleksi :** Harris efektif untuk mendeteksi struktur kuat, tetapi memerlukan penyesuaian parameter hati-hati agar seimbang antara sensitivitas dan keakuratan.

#### 2. SIFT

Parameter utama yang berpengaruh adalah contrastThreshold.

- Nilai rendah (0.01): mendeteksi banyak keypoints, termasuk area datar yang tidak signifikan (*over-detection*).

- Nilai tinggi ( $\geq 0.08$ ): hanya fitur dominan yang terdeteksi, hasil lebih bersih tetapi jumlah titik berkurang drastis (under-detection).

### 3. FAST

Parameter penting pada FAST adalah threshold dan Non-Max Suppression (NMS).

- Threshold rendah (10): jumlah titik fitur sangat banyak, namun banyak yang tidak relevan.
- Threshold tinggi (50): hanya fitur dengan kontras kuat yang tersisa.
- NMS aktif: menyaring fitur duplikat dan menjaga persebaran titik lebih rapi.

**Hasil Refleksi:** FAST sangat cepat dan efisien untuk aplikasi real-time, namun harus diimbangi dengan NMS agar tidak terjadi deteksi berlebih.

#### 3.4. Transformasi Geometri

Fitur *Transformasi Geometri* menunjukkan perbedaan mendasar antara Affine dan Perspective transformation. Affine mempertahankan garis lurus dan paralelisme, sehingga hasil transformasi terlihat proporsional dan stabil terhadap rotasi atau translasi. Sebaliknya, Transformasi Perspektif mengubah persepsi kedalaman dengan meniru proyeksi kamera 3D, di mana garis sejajar dapat tampak bertemu di satu titik hilang (*vanishing point*).

Transformasi Affine cocok digunakan untuk koreksi orientasi dan penyelarasan citra (alignment), sementara Perspective dapat cocok digunakan untuk simulasi tampilan kamera, rekonstruksi ruang, atau *homography mapping*. Jika disimpulkan, transformasi bukan sekadar manipulasi posisi piksel, tetapi juga representasi model matematis dari hubungan antara ruang dunia nyata dan bidang gambar.

## **Lampiran**

Repository Github :

[https://github.com/angiekiera/Angelica\\_13522048\\_IF5152\\_TugasIndividuCV.git](https://github.com/angiekiera/Angelica_13522048_IF5152_TugasIndividuCV.git)