

# **LAPORAN TUGAS INDIVIDU**

## **IF5152 COMPUTER VISION**



**Oleh**

Hayya Zuhailii Kinasih

NIM : 13522102

Kelas: K01

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO & INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2025**

## DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>2</b>
<b>WORKFLOW.....</b>	<b>3</b>
1. Pendahuluan.....	3
2. Struktur Program.....	4
<b>PROSES DAN HASIL.....</b>	<b>6</b>
1. Filtering.....	6
1.1. Gaussian Filter.....	6
1.2. Median Filter.....	6
1.3. Pengujian.....	6
2. Edge Detection.....	8
2.1. Canny Edge Detection.....	8
2.2. Sobel Edge Detection.....	8
2.3. Pengujian.....	9
3. Feature Detection.....	10
3.1. FAST Feature Detection.....	10
3.2. Pengujian.....	10
4. Transformasi.....	12
4.1. Transformasi.....	12
4.2. Pengujian.....	12
<b>KOMPARASI DAN REFLEKSI.....</b>	<b>14</b>
1. Komparasi Hasil Gambar.....	14
1.1. Filtering.....	14
1.2. Edge Detection.....	14
2. Refleksi Pribadi.....	14
2.1. Tantangan dan Solusi.....	14
<b>LAMPIRAN.....</b>	<b>16</b>

# WORKFLOW

## 1. Pendahuluan

Salah satu langkah penting dalam Computer Vision adalah tahap pemrosesan gambar, yaitu tahap yang bertujuan untuk meningkatkan kualitas gambar dan mempersiapkan gambar untuk tahap analisis berikutnya. Dalam tugas ini, akan dilakukan eksplorasi beberapa teknik dasar dalam pemrosesan gambar yang merupakan dasar untuk membangun sistem computer vision yang lebih kompleks. Fitur-fitur yang dieksplorasi meliputi filtering, edge detection, feature point detection, dan transformasi geometri.

Proses filtering dilakukan dengan tujuan mengurangi noise dan memperhalus gambar agar fitur penting pada gambar bertahan sedangkan yang kurang penting dihilangkan. Dua jenis filtering yang digunakan dalam tugas ini adalah metode Gaussian dan metode median. Gaussian filter melakukan perataan dengan pembobotan berbasis distribusi normal yang membuat gambar menjadi blur namun tetap mempertahankan tepi dengan baik. Median filter melakukan filtering dengan menggantikan setiap piksel dengan median dari tetangganya, yang cocok untuk menghilangkan noise yang bersifat salt-and-pepper dan tidak mengaburkan tepi.

Proses edge detection bertujuan untuk mengekstraksi struktur utama dari sebuah gambar. Dua metode edge detection dalam tugas ini adalah sobel operator dan canny edge detector. Cara kerja sobel edge detector adalah dengan mendeteksi perubahan intensitas piksel pada arah horizontal dan vertikal, kemudian digabungkan untuk menemukan bentuk dasar objek. Canny edge detector memanfaatkan filtering Gaussian secara bertahap, perhitungan gradien, dan thresholding untuk menghasilkan edge detection yang lebih rapi dan bersih.

Proses feature point detection bertujuan untuk menemukan titik-titik sudut yang terdapat dalam gambar. Algoritma yang digunakan dalam tugas ini adalah FAST (Features from Accelerated Segment Test).

Fitur terakhir adalah transformasi geometri, yang merupakan simulasi kalibrasi kamera sederhana dan proyeksi citra dengan pola grid. Tujuannya adalah untuk memahami hubungan

antara koordinat dunia nyata yang bersifat tiga dimensi dan proyeksinya ke bidang dua dimensi, yang dimodelkan dengan matriks transformasi.

Secara keseluruhan, fitur-fitur ini membentuk pipeline dasar dalam pengolahan citra. Dimulai dari pembersihan data hingga ekstraksi dan representasi spasial dari elemen penting dalam gambar.

## 2. Struktur Program

Berikut adalah struktur folder dalam program, terpisah sesuai masing-masing fitur.

```
Nama_NIM_IF5152_TugasIndividuCV/
```

```
|
```

```
|— 01_filtering/
```

```
|
```

```
| |— experiments.ipynb
```

```
|
```

```
| |— gaussian.py
```

```
|
```

```
| |— median.py
```

```
|
```

```
| |— util.py
```

```
|
```

```
|— 02_edge/
```

```
|
```

```
| |— experiments.ipynb
```

```
|
```

```
| |— sobel.py
```

```
|
```

```
| |— canny.py
```

```
|
```

```
| |— util.py
```

```
|
```

```
|— 03_featurepoints/
```

```
|
```

```
| |— experiments.ipynb
```

```
|
```

```
| |— fast.py
```

```
|
```

```
| |— util.py
```

```
|  
|— 04_geometri/  
| |— experiments.ipynb  
| |— transformasi.py  
| |— util.py  
|  
|— 05_laporan.pdf/  
|— README.md
```

Proses pengujian dan hasil pengujian masing-masing fitur dilakukan dalam file experiments.ipynb sesuai dengan foldernya. Gambar yang digunakan pada setiap fitur didapatkan dari library skimage, yaitu gambar camera, coins, checkerboard, dan chelsea. Tiga gambar pertama adalah gambar-gambar hitam-putih yang sering digunakan dalam pengujian pemrosesan gambar, sedangkan gambar chelsea adalah gambar berwarna yang juga sering digunakan untuk keperluan yang serupa.

## PROSES DAN HASIL

### 1. Filtering

#### 1.1. Gaussian Filter

Gaussian filter adalah filter yang menggunakan fungsi Gaussian untuk meredam frekuensi tinggi pada gambar, seperti noise atau tepi yang tajam. Kernel Gaussian didefinisikan sebagai berikut:



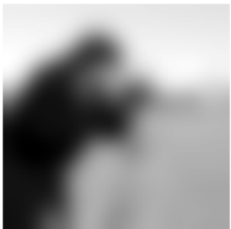

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

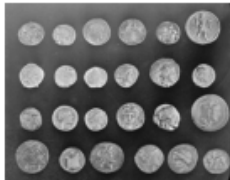
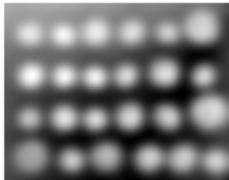

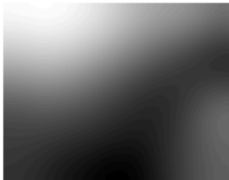
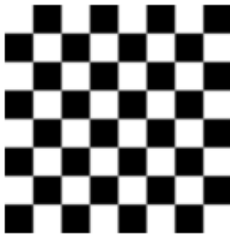

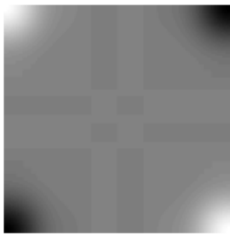
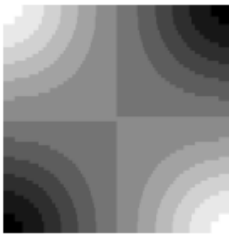







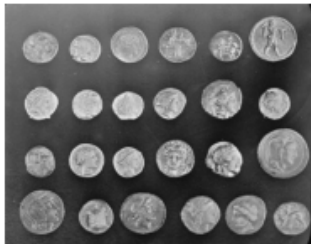
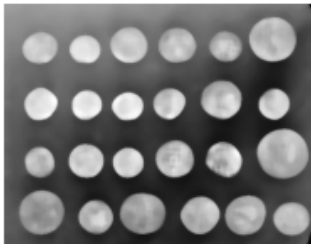
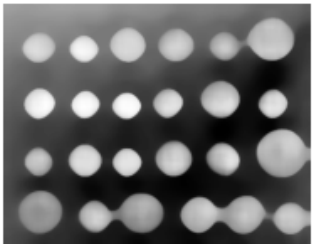
dengan  $x$  dan  $y$  sebagai jarak dari pusat dan  $\sigma$  sebagai standar deviasi dari distribusi Gaussian. Nilai  $\sigma$  tersebut mengatur tingkat blur, semakin tinggi, maka hasilnya akan semakin buram.

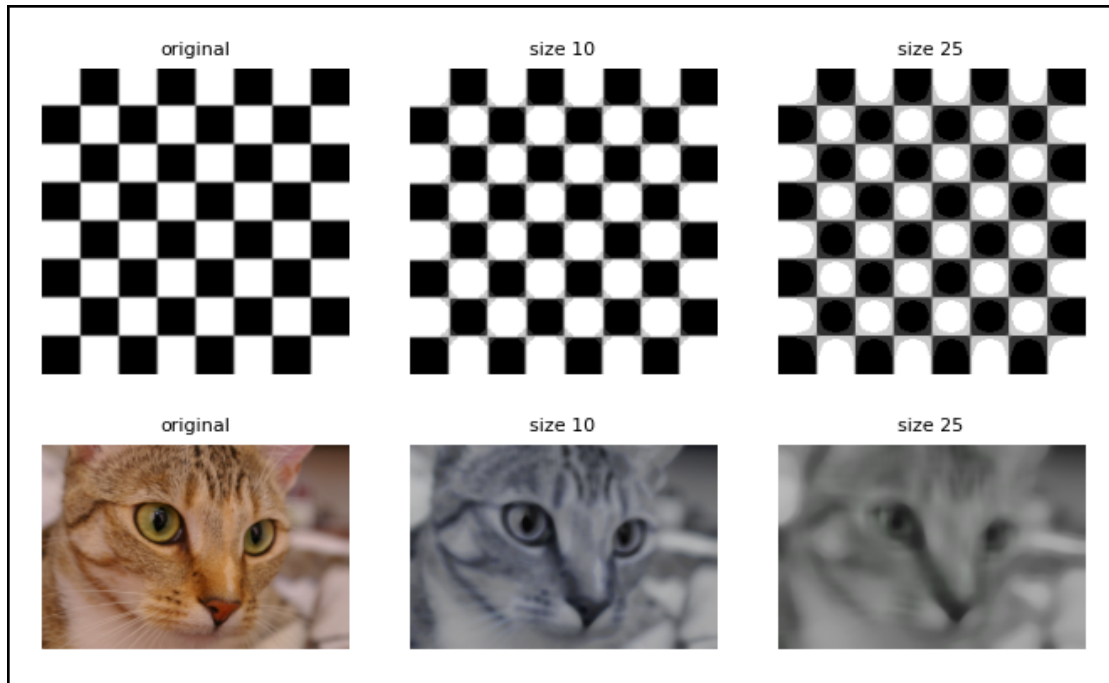
#### 1.2. Median Filter

Median filter adalah filter non-linier yang menggantikan setiap piksel dengan median dari nilai piksel-piksel tetangganya dengan ukuran kernel tertentu. Median filter lebih cocok untuk menghapus salt-and-pepper noise, namun memiliki kelemahan yaitu memakan waktu yang lebih lama untuk dieksekusi.

#### 1.3. Pengujian

No	Metode Filtering	Parameter
1	Gaussian	Sigma = [10, 25, 50]
<div><div>original</div><div></div><div>sigma 10</div><div></div><div>sigma 25</div><div></div><div>sigma 50</div><div></div></div>		

original				sigma 10				sigma 25				sigma 50			
															
original				sigma 10				sigma 25				sigma 50			
															
original				sigma 10				sigma 25				sigma 50			
															
2	Median						Size = [10, 25]								
original				size 10				size 25							
															
original				size 10				size 25							
															



## 2. Edge Detection

### 2.1. Canny Edge Detection

Metode Canny disebut sebagai detektor yang optimal dalam fungsinya. Langkah-langkah yang dilalui antara lain filter Gauss untuk mengurangi noise, menghitung arah dan besar gradien, supresi piksel yang nilainya kurang dari maksimum, dan thresholding dengan dua nilai untuk menyambung tepi yang terputus.


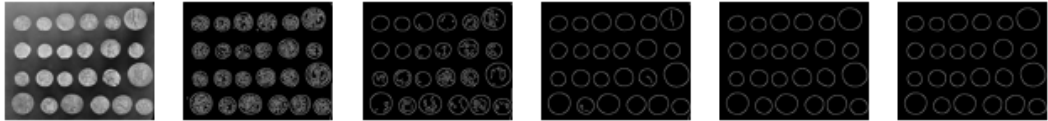
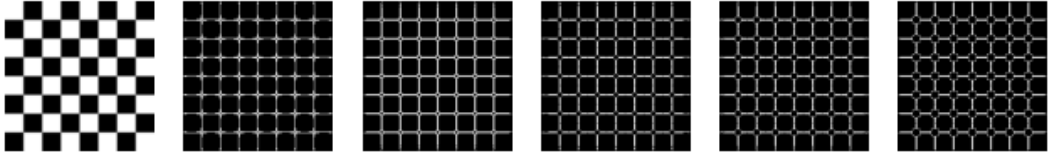
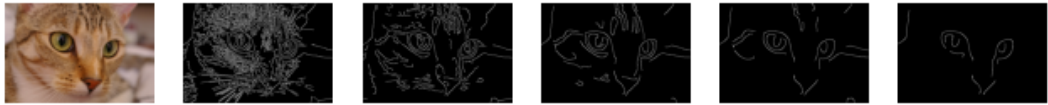
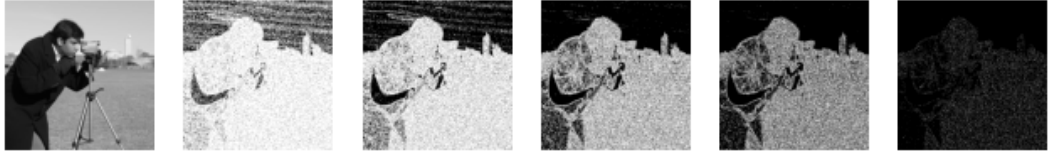
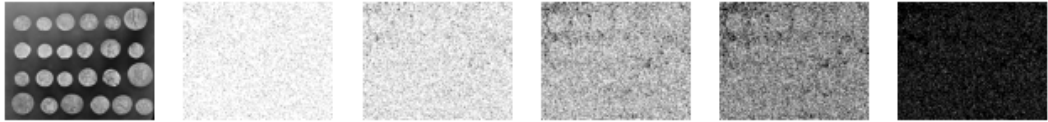
Parameter penting dalam implementasi ini adalah  $\sigma$  pada Gaussian smoothing. Semakin besar nilainya, semakin banyak noise yang dihilangkan, namun hal itu berisiko menghilangkan tepi.

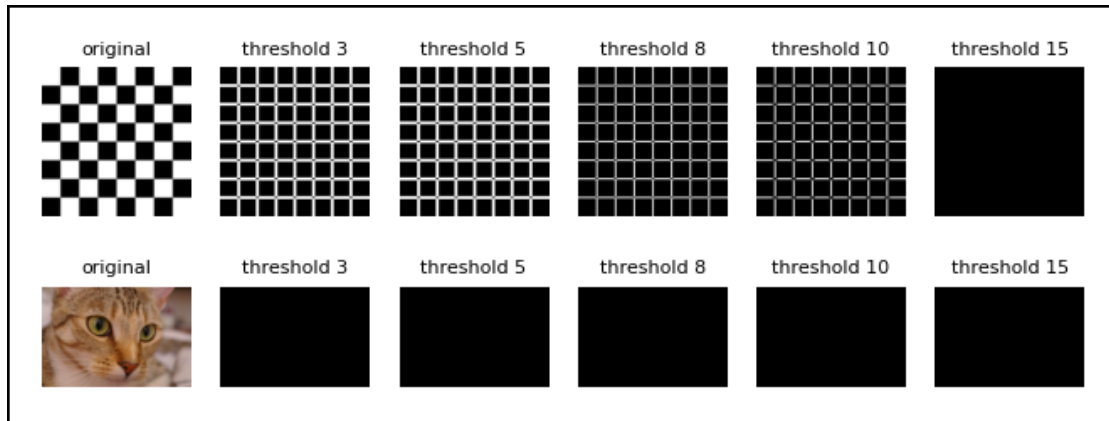
### 2.2. Sobel Edge Detection

Sobel menghitung perkiraan turunan pertama pada arah horizontal dan vertikal. Setelah itu, kedua turunan tersebut digabungkan dengan menghitung nilai akar dari penjumlahan kuadrat kedua nilai tersebut. Setelah itu, dilakukan thresholding untuk mengubah hasil ke citra biner tepi. Semakin tinggi threshold, maka akan lebih sedikit noise, namun juga dapat menghilangkan tepi.



### 2.3. Pengujian

No	Metode Edge Detection	Parameter
1	Canny	Sigma = [1, 2, 3, 4, 5]
<div> <div>originalsigma 1sigma 2sigma 3sigma 4sigma 5</div>  </div> <div> <div>originalsigma 1sigma 2sigma 3sigma 4sigma 5</div>  </div> <div> <div>originalsigma 1sigma 2sigma 3sigma 4sigma 5</div>  </div> <div> <div>originalsigma 1sigma 2sigma 3sigma 4sigma 5</div>  </div>		
2	Sobel	Threshold= [3, 5, 8, 10, 15]
<div> <div>originalthreshold 3threshold 5threshold 8threshold 10threshold 15</div>  </div> <div> <div>originalthreshold 3threshold 5threshold 8threshold 10threshold 15</div>  </div>		

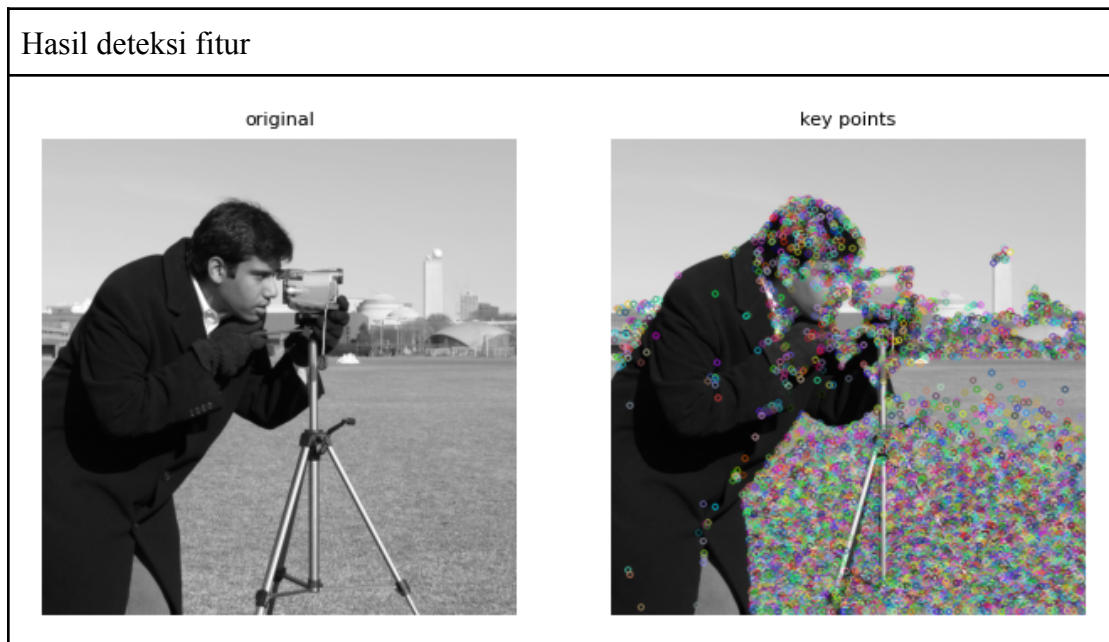


### 3. Feature Detection

#### 3.1. FAST Feature Detection

Algoritma FAST dieksekusi dengan memilih piksel kandidat  $p$  dan mengambil 16 piksel di sekitarnya dalam bentuk lingkaran. Intensitas piksel dalam lingkaran dibandingkan dengan piksel pusat. Jika terdapat  $n$  piksel berurutan yang berbeda jauh dengan  $p$ , maka  $p$  dianggap sebagai corner. Setelah itu, dilakukan juga supresi nilai yang bukan maksimum untuk mempertahankan titik-titik yang signifikan.

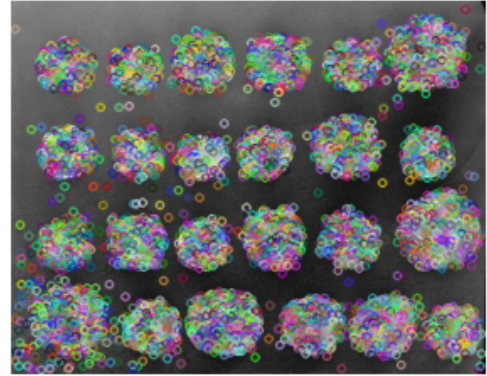
#### 3.2. Pengujian



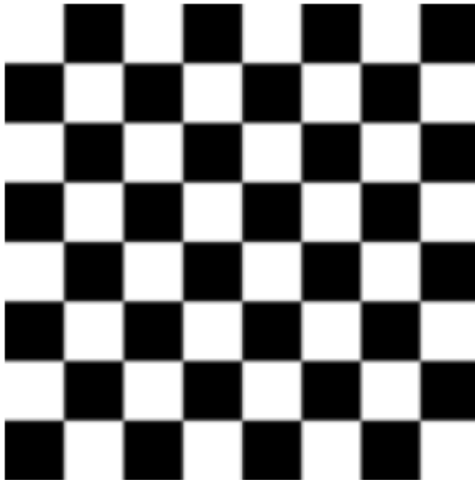
original



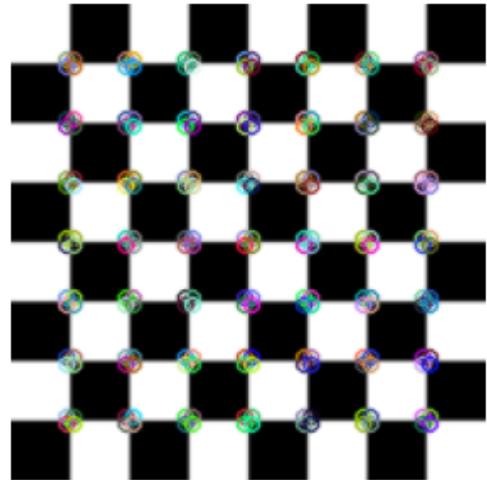
key points



original



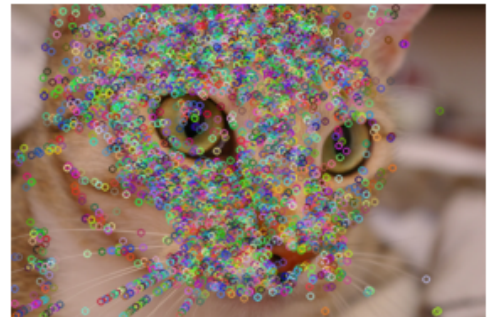
key points



original



key points




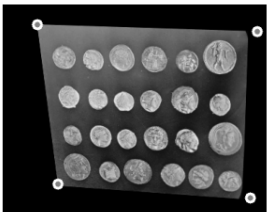

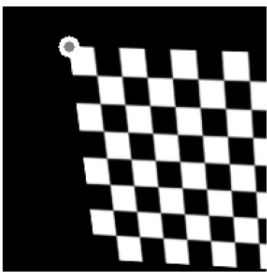




## 4. Transformasi

### 4.1. Transformasi

Jenis transformasi yang dilakukan pada tugas ini adalah projective transform, yaitu untuk memetakan satu bidang ke bidang yang lain. Perhitungan dilakukan dari empat pasang titik korespondensi dengan metode Direct Linear Transform (DLT).

### 4.2. Pengujian

No	Matriks	Hasil
1	$\begin{bmatrix} 7.52474658e-01 & 1.38046704e-01 & 5.00000000e+01 \\ 1.97080238e-02 & 8.91795600e-01 & 3.00000000e+01 \\ -1.33865822e-04 & 4.71403194e-04 & 1.00000000e+00 \end{bmatrix}$	<div><div>original</div></div> <div><div>transformed</div></div>
2	$\begin{bmatrix} 7.52474658e-01 & 1.38046704e-01 & 5.00000000e+01 \\ 1.97080238e-02 & 8.91795600e-01 & 3.00000000e+01 \\ -1.33865822e-04 & 4.71403194e-04 & 1.00000000e+00 \end{bmatrix}$	<div><div>original</div></div> <div><div>transformed</div></div>
3	$\begin{bmatrix} 7.52474658e-01 & 1.38046704e-01 & 5.00000000e+01 \\ 1.97080238e-02 & 8.91795600e-01 & 3.00000000e+01 \\ -1.33865822e-04 & 4.71403194e-04 & 1.00000000e+00 \end{bmatrix}$	<div><div>original</div></div> <div><div>transformed</div></div>

4	<pre>[[ 7.52474658e-01  1.38046704e-01  5.00000000e+01]  [ 1.97080238e-02  8.91795600e-01  3.00000000e+01]  [-1.33865822e-04  4.71403194e-04  1.00000000e+00]]</pre>	<div>original</div>  <div>transformed</div> 
---	--	---

# KOMPARASI DAN REFLEKSI

## 1. Komparasi Hasil Gambar

### 1.1. Filtering

Gaussian dan Median filter telah menunjukkan hasil sesuai teori. Gaussian meratakan secara global, sedangkan median meratakan secara lokal. Gaussian akan lebih cocok pada citra dengan detail tekstur kecil yang tidak terlalu penting, sedangkan median lebih efektif untuk citra nyata dengan noise tidak teratur.

### 1.2. Edge Detection

Canny memberikan hasil edge yang lebih bersih dan dengan kontras tinggi dibandingkan dengan Sobel. Sobel memiliki lebih banyak noise dan gradasi. Canny lebih bagus jika diterapkan pada citra dengan pencahayaan yang tidak terlalu ekstrim, sedangkan sobel lebih cocok pada kondisi-kondisi tersebut, walaupun hasilnya tetap tidak sebanding dengan hasil dari Canny ketika kondisinya terkendali.

## 2. Refleksi Pribadi

### 2.1. Tantangan dan Solusi

No	Aspek	Tantangan	Keputusan Desain
1	Struktur Program	Perlu menata letak fungsi dengan jelas dan bersifat modular	Membuat file python terpisah untuk setiap jenis metode, dan pengujian dilakukan dalam sebuah Jupyter Notebook.
2	Edge detection	Gambar berwarna menyebabkan error pada pada fungsi Sobel dan Canny	Mengecek apakah gambar berwarna atau tidak. Jika ya, konversi terlebih dahulu menjadi hitam-putih.

### 2.2. Kesimpulan

1. Parameter sangat mempengaruhi kualitas hasil image processing.
2. Image preprocessing adalah kunci untuk meningkatkan hasil dari tahap-tahap pemrosesan citra berikutnya.
3. Tahap-tahap image processing dapat terhubung di antaranya.

## **LAMPIRAN**

Link program:

[https://github.com/hayyazk/Hayya-Zuhailii-Kinasih\\_13522102\\_IF5152\\_TugasIndividuCV](https://github.com/hayyazk/Hayya-Zuhailii-Kinasih_13522102_IF5152_TugasIndividuCV)