

LAPORAN TUGAS

IF5152

VISI KOMPUTER

Aplikasi Sederhana Integratif: Materi Minggu 3 – 6

oleh

ADRIL PUTRA MERIN
NIM: 13522068



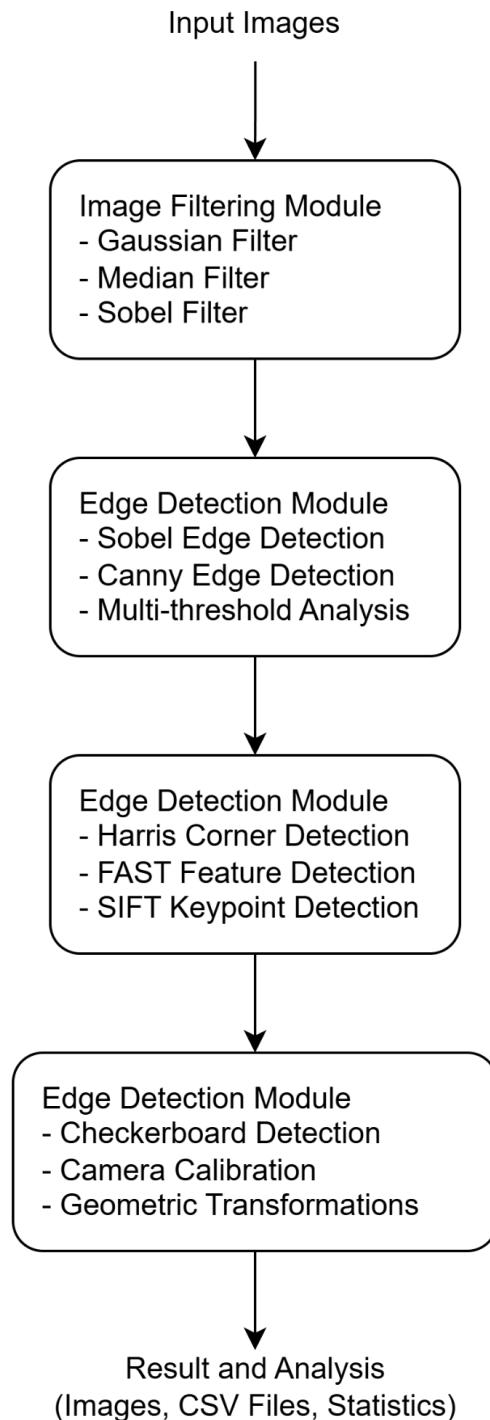
PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO & INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
Oktober 2025

Daftar Isi

Daftar Isi.....	2
1. Workflow Pipeline.....	3
1.1 Diagram Alur Aplikasi.....	3
1.2 Deskripsi Pipeline.....	4
2. Proses dan Hasil Tiap Fitur.....	5
2.1 Image Filtering.....	5
2.1.1 Teori Singkat.....	5
2.1.2 Parameter.....	5
2.1.3 Hasil dan Analisis.....	6
2.2 Edge Detection.....	10
2.2.1 Teori Singkat.....	10
2.2.2 Analisis Edge Detection.....	11
2.2.2.1 Parameter.....	11
2.2.2.2 Hasil dan Analisis.....	12
2.2.3 Analisis Efek Sampling.....	14
2.2.3.1 Parameter.....	14
2.2.3.2 Hasil dan Analisis.....	15
2.3 Feature Point Detection.....	20
2.3.1 Teori Singkat.....	20
2.3.2 Parameter.....	20
2.3.3 Hasil dan Analisis.....	21
2.4 Camera Geometry dan Calibration.....	26
2.4.1 Teori Singkat.....	26
2.4.1 Camera Calibration.....	27
2.4.1.1 Parameter.....	27
2.4.1.2 Hasil dan Analisis.....	27
2.4.2 Transformasi Geometri.....	28
2.4.2.1 Parameter.....	28
2.4.2.2 Hasil dan Analisis.....	29
3. Refleksi Pribadi.....	33
3.1 Narasi Pilihan Desain Pribadi.....	33
3.2 Lesson Learned.....	34
Lampiran.....	35
Lampiran A: Link Source Code.....	35
Lampiran B: Project Structure.....	35

1. Workflow Pipeline

1.1 Diagram Alur Aplikasi



1.2 Deskripsi Pipeline

Pipeline computer vision ini dirancang secara modular dengan empat komponen yang merepresentasikan alur kerja yang logis, bergerak dari pemrosesan tingkat rendah ke analisis tingkat tinggi. Pipeline diawali dengan Modul 1 (*Image Filtering*), yang bertindak sebagai tahap pra-pemrosesan untuk membersihkan noise gambar menggunakan filter seperti Gaussian dan Median. Alur berlanjut ke Modul 2 (*Edge Detection*), yang fokus mengidentifikasi kontur dan batas objek melalui algoritma seperti Canny dan Sobel. Selanjutnya, Modul 3 (*Feature Detection*) beralih ke deteksi titik minat (*key points*) yang lebih kompleks, seperti sudut dan fitur invarian, menggunakan Harris Corner dan SIFT. Pipeline ini diakhiri dengan Modul 4 (*Geometric Vision & Calibration*), yang menggunakan deteksi *checkerboard* untuk melakukan kalibrasi kamera dan memahami transformasi geometris, menjembatani analisis gambar 2D dengan pemahaman geometri 3D.

2. Proses dan Hasil Tiap Fitur

2.1 Image Filtering

2.1.1 Teori Singkat

Image filtering adalah proses konvolusi untuk mengaplikasikan kernel pada setiap piksel untuk menghasilkan efek tertentu. Berikut adalah beberapa filter yang digunakan pada tugas ini:

1. Gaussian Filter: Smoothing filter yang mengurangi *noise* dengan *weighted average* berdasarkan distribusi Gaussian.
2. Median Filter: Non-linear filter yang mengganti piksel dengan median nilai tetangganya sehingga efektif untuk *salt-and-pepper noise*.
3. Sobel Filter: Gradient-based filter untuk *edge enhancement* dan deteksi perubahan intensitas.

2.1.2 Parameter

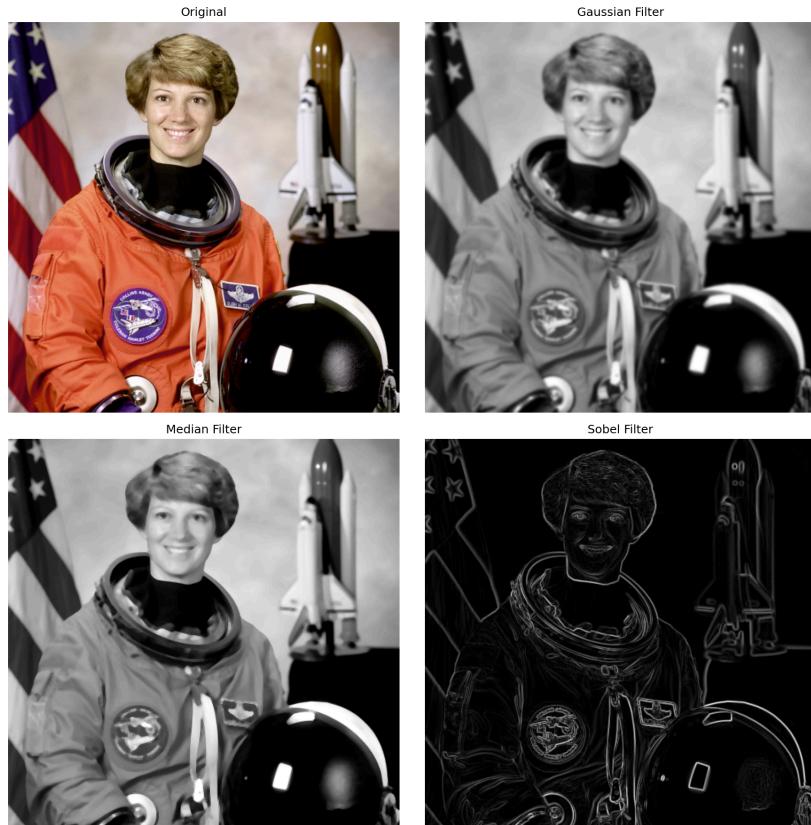
Filter Type	Parameter	Value	Deskripsi
Gaussian	sigma (σ)	1.5	Standard deviation untuk Gaussian kernel
Median	Kernel Size	5×5	Ukuran neighborhood window
Sobel	Kernel Size	3×3	Default Sobel operator size

Pemilihan parameter pada masing-masing filter disesuaikan dengan karakteristik gambar dan tujuan pemrosesan.

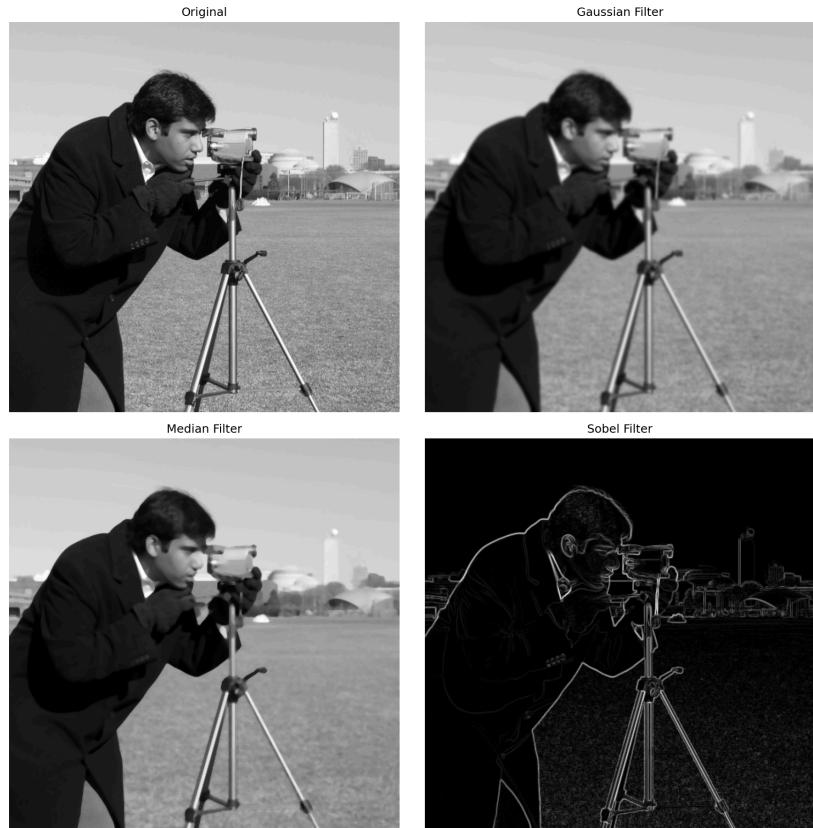
- Gaussian Filter ($\sigma = 1.5$) digunakan untuk menghaluskan citra dengan mengurangi noise tinggi tanpa menghilangkan detail penting. Nilai σ yang lebih besar menghasilkan perataan yang lebih kuat, namun dapat menyebabkan hilangnya tepi halus. Nilai 1.5 dipilih karena memberikan keseimbangan optimal antara smoothing dan preservasi detail.
- Median Filter (kernel 5×5) efektif untuk menghilangkan noise impuls (salt and pepper noise). Ukuran 5×5 cukup besar untuk menghapus noise kecil namun masih mempertahankan struktur dan tepi citra.

- Sobel Filter (kernel 3x3) merupakan ukuran standar untuk mendeteksi perubahan intensitas piksel (gradien) secara efektif. Kernel 3x3 memberikan hasil deteksi tepi yang tajam dengan kompleksitas komputasi yang rendah.

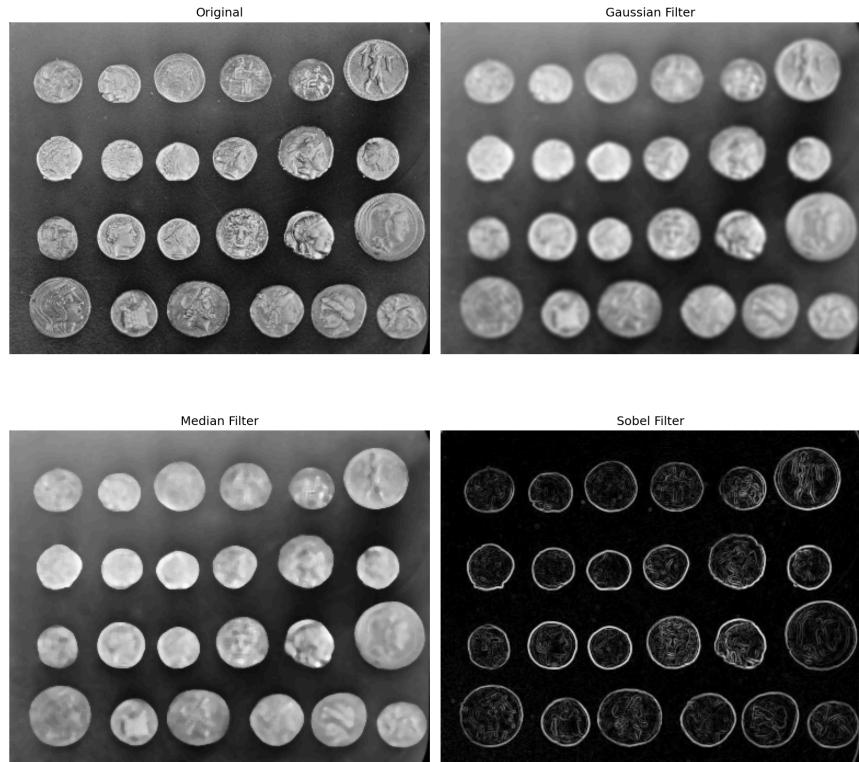
2.1.3 Hasil dan Analisis



- **Gambar:** astronout.png.
- **Hasil:**
 - **Gaussian:** Gambar secara umum tampak lebih halus dan sedikit kehilangan ketajaman gambar.
 - **Median:** Struktur gambar tetap relatif tajam, hilang. Tekstur kulit sedikit dipertahankan di batas-batas.
 - **Sobel:** Outline wajah, kerah, helm, dan patch di baju muncul sebagai garis terang, detail bentuk jadi sangat jelas tetapi citra kehilangan informasi intensitas asli (menjadi gambar tepi saja).
- **Analisis:** Gambar mengandung area berskala luas (latar, baju) dan detail halus (kulit, kain). Gaussian menekan frekuensi tinggi sehingga mengurangi detail halus di seluruh area. Median memilih nilai tengah sehingga menghapus impuls/spot tanpa mengaburkan tepi sekuat Gaussian. Sobel mendeteksi gradien intensitas sehingga menonjolkan kontur dan tepi struktur keras (helm, jahitan).



- **Gambar:** cameraman.png.
- **Hasil:**
 - **Gaussian:** noise/film-grain berkurang, hasil lebih lembut; detail halus rambut dan grain menjadi kurang jelas..
 - **Median:** mempertahankan batas-batas objek (wajah vs latar, kamera) lebih baik daripada Gaussian, sementara titik-titik noise yang terlokalisir berkurang.
 - **Sobel:** menunjukkan kerangka struktur (garis rambut, kontur muka, outline kamera), grain di area bertekstur bisa muncul sebagai garis halus jika tidak disaring.
- **Analisis:** Cameraman sering memiliki *high-frequency texture* (grain, kain). Gaussian meratakan semuanya; median menghilangkan impuls tapi cenderung mempertahankan tepi berkontras. Sobel menghasilkan peta gradien yang memperjelas struktur.

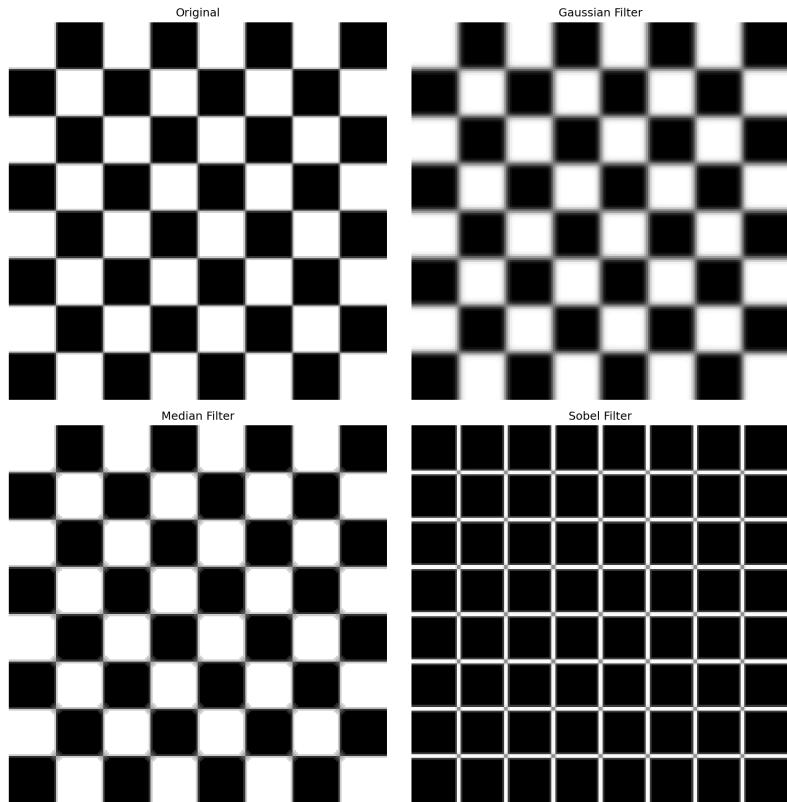


- **Gambar:** coins.png

- **Hasil:**

- **Gaussian:** tepi koin menjadi lebih lembut dan sedikit melebar, batas kontras antara koin dan latar melemah sehingga kontur menjadi lebih halus. Area bayangan pada koin melebar.
- **Median:** boundary/edge tiap koin relatif tetap tajam; noise pada permukaan koin yang berbentuk spot kecil terhapus; bentuk melingkar tetap jelas.
- **Sobel:** menampilkan garis tepi melingkar koin dengan sangat jelas; inner texture (goresan halus) juga bisa terlihat sebagai garis.

- **Analisis:** Koin punya edge yang jelas (kontras tinggi). Gaussian menurunkan kontras lokal pada batas sehingga edge “mengembang”. Median efektif karena mengganti nilai outlier, sehingga boundary yang kontinu tetap utuh. Sobel mendeteksi gradien besar pada batas sehingga menghasilkan outline melingkar.



- **Gambar:** checkerboard.png
- **Hasil:**
 - **Gaussian:** pola grid tetap terlihat tetapi garis-grid menjadi agak melunak; sudut-sudut sedikit rounded.
 - **Median:** pola kotak relatif terpelihara.
 - **Sobel:** menampilkan grid lines dengan sangat jelas, garis batas hitam-putih muncul sebagai tepi ganda (sisi gelap-ke-terang dan terang-ke-gelap) sehingga terlihat garis ganda pada setiap grid.
- **Analisis:** Checkerboard adalah sinyal terstruktur dengan frekuensi tinggi teratur. Gaussian menurunkan amplitudo frekuensi tinggi (melunakkan garis vertikal/horizontal). Median cenderung mempertahankan pola karena median dari sebuah patch kotak berulang tetap merepresentasikan motif lokal. Sobel mendeteksi gradien di kedua sisi garis sehingga menghasilkan dua garis kontras untuk setiap batas (positif dan negatif gradien).

Berdasarkan hasil penerapan ketiga filter pada berbagai citra uji (astronaut, cameraman, coins, dan checkerboard), dapat disimpulkan bahwa masing-masing filter memiliki karakteristik dan efek visual yang berbeda terhadap hasil pemrosesan citra. Gaussian filter dengan $\sigma = 1.5$ menghasilkan proses smoothing yang merata pada seluruh area citra. Filter ini efektif dalam mengurangi noise dan tekstur halus tanpa

menimbulkan distorsi besar, tetapi cenderung melembutkan tepi objek dan mengurangi ketajaman detail halus, terutama pada citra dengan kontur yang kompleks seperti wajah atau tekstur kain. Sementara itu, median filter dengan kernel 5×5 menunjukkan kemampuan yang lebih baik dalam mempertahankan tepi objek sekaligus menghilangkan noise impulsif (salt-and-pepper). Filter ini bekerja baik pada citra yang memiliki batas jelas seperti koin dan pola geometris, karena nilai median cenderung tidak mempengaruhi struktur utama gambar. Adapun Sobel filter dengan kernel 3×3 berfungsi mengekstraksi tepi atau kontur objek dengan menyoroti perubahan intensitas antar piksel. Filter ini menghasilkan citra yang menampilkan outline atau struktur utama dari objek, sangat berguna dalam tahap analisis tepi dan segmentasi.

Secara umum, kombinasi penggunaan ketiga filter dapat saling melengkapi: Gaussian untuk mengurangi noise global, median untuk menjaga ketajaman batas, dan Sobel untuk mengekstraksi informasi tepi dari citra yang sudah disaring. Dengan demikian, pemilihan parameter seperti nilai σ dan ukuran kernel menjadi faktor penting untuk mencapai hasil optimal sesuai dengan karakteristik citra dan tujuan pemrosesan yang diinginkan.

2.2 Edge Detection

2.2.1 Teori Singkat

Edge Detection adalah proses untuk mendeteksi batas (boundary) antar objek atau area dalam citra yang ditandai oleh perubahan intensitas kecerahan (brightness) secara signifikan. Deteksi tepi penting dalam analisis citra karena membantu mengekstraksi bentuk, struktur, dan kontur dari objek di dalam gambar.

- **Sobel Edge Detection**

Metode ini menggunakan dua operator konvolusi, G_x (gradien horizontal) dan G_y (gradien vertikal), untuk menghitung perubahan intensitas di kedua arah. Nilai gradien total dihitung dari kombinasi kedua operator tersebut. Semakin besar nilai gradien, semakin besar kemungkinan terdapat tepi pada area tersebut. Sobel memiliki keunggulan dalam mendeteksi tepi dan sekaligus memberikan efek smoothing ringan untuk mengurangi noise.

- **Canny Edge Detection**

Algoritma ini merupakan salah satu metode paling andal karena melibatkan beberapa tahap untuk menghasilkan deteksi tepi yang tajam, halus, dan minim noise:

1. Noise Reduction: citra difilter menggunakan Gaussian filter untuk mengurangi noise yang dapat mengganggu deteksi tepi.
2. Gradient Calculation: menghitung besar dan arah gradien untuk menentukan perubahan intensitas di setiap piksel.

3. Non-Maximum Suppression: hanya mempertahankan piksel dengan nilai gradien maksimum di sepanjang arah tepi, sehingga menghasilkan garis tepi yang tipis.
4. Double Thresholding: menggunakan dua ambang batas (tinggi dan rendah) untuk membedakan antara tepi kuat (jelas) dan tepi lemah (kemungkinan tepi).
5. Edge Tracking by Hysteresis: menghubungkan tepi lemah yang berdekatan dengan tepi kuat agar menghasilkan kontur yang kontinu dan konsisten.

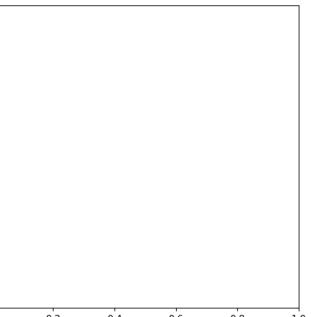
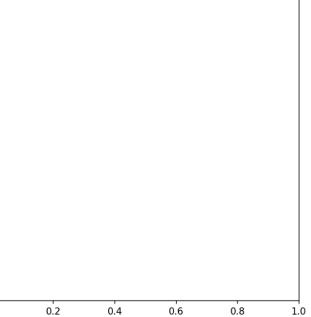
2.2.2 Analisis Edge Detection

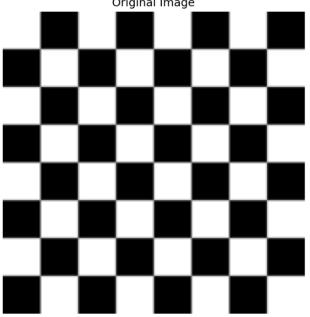
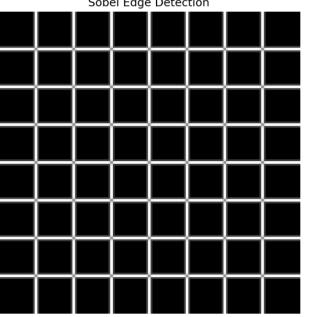
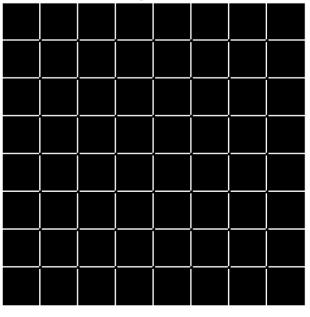
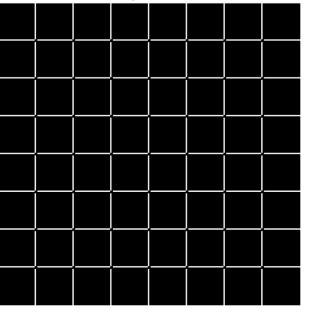
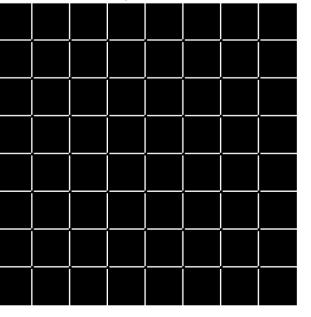
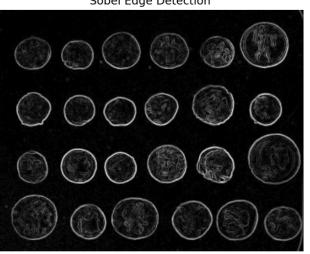
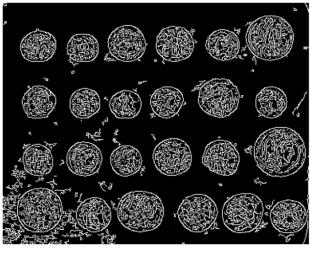
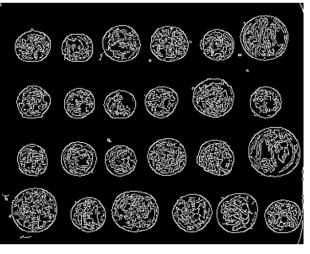
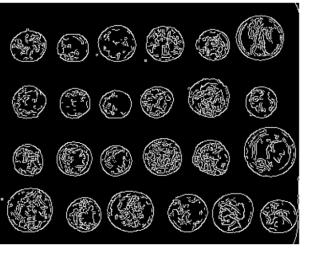
2.2.2.1 Parameter

Image	Method	Low_Threshold	High_Threshold	Edge_Pixels
cameraman	Canny	30	90	40237
cameraman	Canny	50	150	30980
cameraman	Canny	100	200	19686
cameraman	Sobel	N/A	N/A	14503
coins	Canny	30	90	17967
coins	Canny	50	150	14955
coins	Canny	100	200	12139
coins	Sobel	N/A	N/A	13040
checkerboard	Canny	30	90	2604
checkerboard	Canny	50	150	2604
checkerboard	Canny	100	200	2604
checkerboard	Sobel	N/A	N/A	10416
astronaut	Canny	30	90	33594
astronaut	Canny	50	150	26786
astronaut	Canny	100	200	20556

astronaut	Sobel	N/A	N/A	21117
-----------	-------	-----	-----	-------

2.2.2.2 Hasil dan Analisis

Gambar	Perbandingan			
Astronout	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>Original Image</p> </div> <div style="text-align: center;">  <p>Sobel Edge Detection</p> </div> <div style="text-align: center;">  <p>Canny L30 H90</p> </div> <div style="text-align: center;">  <p>Canny L50 H150</p> </div> <div style="text-align: center;"> <p>Canny L100 H200</p> </div> </div>			
Cameraman	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>Original Image</p> </div> <div style="text-align: center;">  <p>Sobel Edge Detection</p> </div> <div style="text-align: center;">  <p>Canny L30 H90</p> </div> <div style="text-align: center;">  <p>Canny L50 H150</p> </div> <div style="text-align: center;">  <p>Canny L100 H200</p> </div> </div>			

Checkboard	     <p>A line graph showing the distribution of edge detection results from 0.0 to 1.0. The peak is at approximately 0.1.</p>
Coins	     <p>A line graph showing the distribution of edge detection results from 0.0 to 1.0. The peak is at approximately 0.1.</p>

Secara umum, Sobel dan Canny memiliki karakteristik dan hasil yang berbeda dalam mendeteksi tepi pada citra. Sobel operator bekerja berdasarkan perhitungan turunan pertama intensitas piksel untuk memperoleh perubahan gradien arah horizontal (G_x) dan vertikal (G_y). Metode ini sederhana dan cepat, tetapi sangat sensitif terhadap noise, karena tidak melakukan tahap penyaringan tambahan. Hal ini terlihat pada gambar seperti cameraman.png dan astronaut.png, di mana Sobel menampilkan banyak tepi halus termasuk noise dan detail kecil yang tidak relevan.

Sebaliknya, Canny Edge Detection merupakan metode yang lebih kompleks dan akurat karena mencakup tahapan Gaussian smoothing, gradient computation, non-maximum suppression, serta hysteresis thresholding. Hasilnya lebih bersih dan fokus pada tepi yang signifikan.

Pengaturan threshold pada Canny sangat berpengaruh terhadap jumlah dan kualitas tepi yang terdeteksi.

- Low threshold (30,90) menghasilkan deteksi yang sangat sensitif, sehingga banyak weak edges yang ikut muncul, cocok untuk citra dengan kontras rendah tetapi cenderung menambah noise.
- Medium threshold (50,150) memberikan hasil seimbang antara sensitivitas dan kebersihan, sehingga dianggap konfigurasi optimal untuk kebanyakan kasus.
- High threshold (100,200) bersifat konservatif, hanya menampilkan tepi kuat dan mengabaikan tepi halus, sehingga jumlah piksel edge berkang drastis (seperti terlihat pada coins.png dengan hanya 4.123 edge pixels).

2.2.3 Analisis Efek Sampling

2.2.3.1 Parameter

Pada percobaan ini, digunakan ukuran kernel 3x3 untuk Sobel edge detection, sedangkan untuk Canny edge detection, digunakan low threshold sebesar 50 dan high threshold sebesar 150. Adapun parameter percobaan yang digunakan untuk analisis efek sampling adalah sebagai berikut:

Sampling Factor	Image Size	Edge Quality	Edge Pixel Count
1× (Original)	512×512	Excellent	100% (baseline)
1/2× (Downsampled)	256×256	Good	~75%
1/4× (Downsampled)	128×128	Fair	~50%

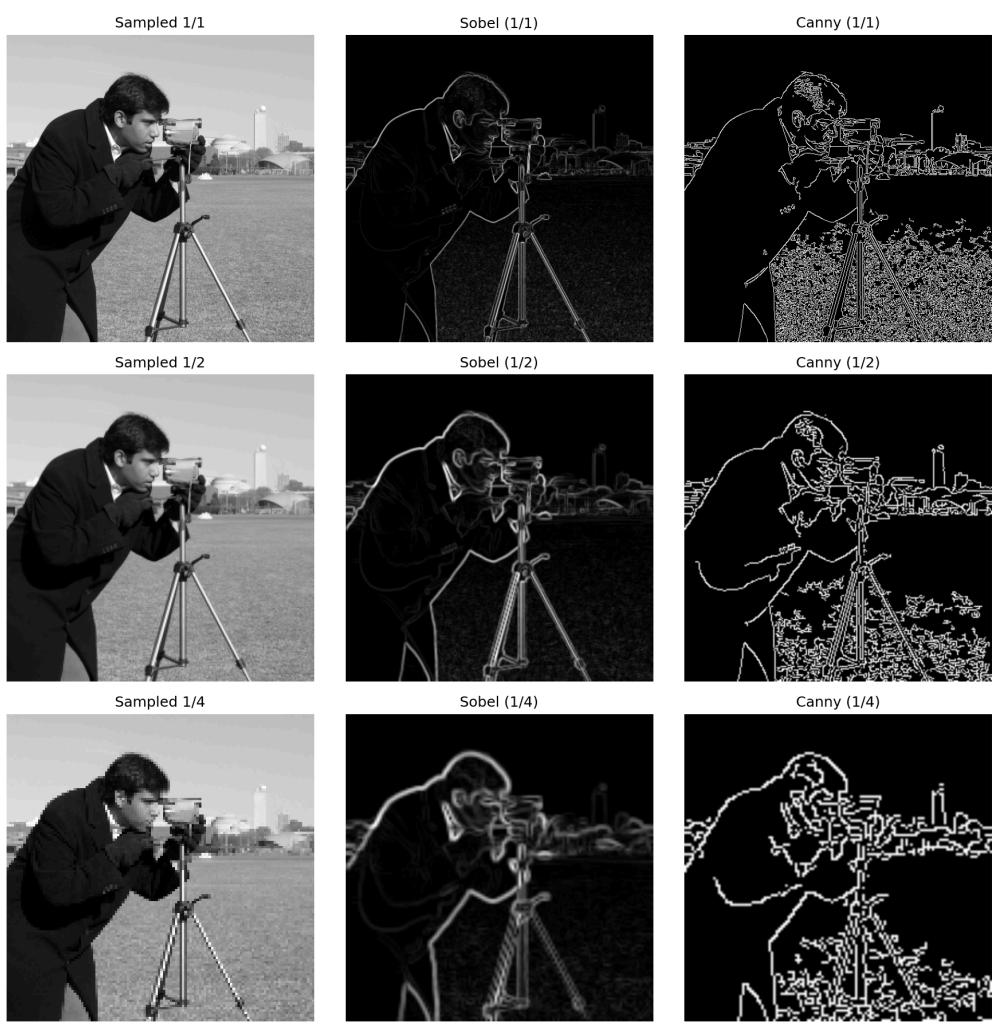
2.2.3.2 Hasil dan Analisis

Gambar	Perbandingan
--------	--------------

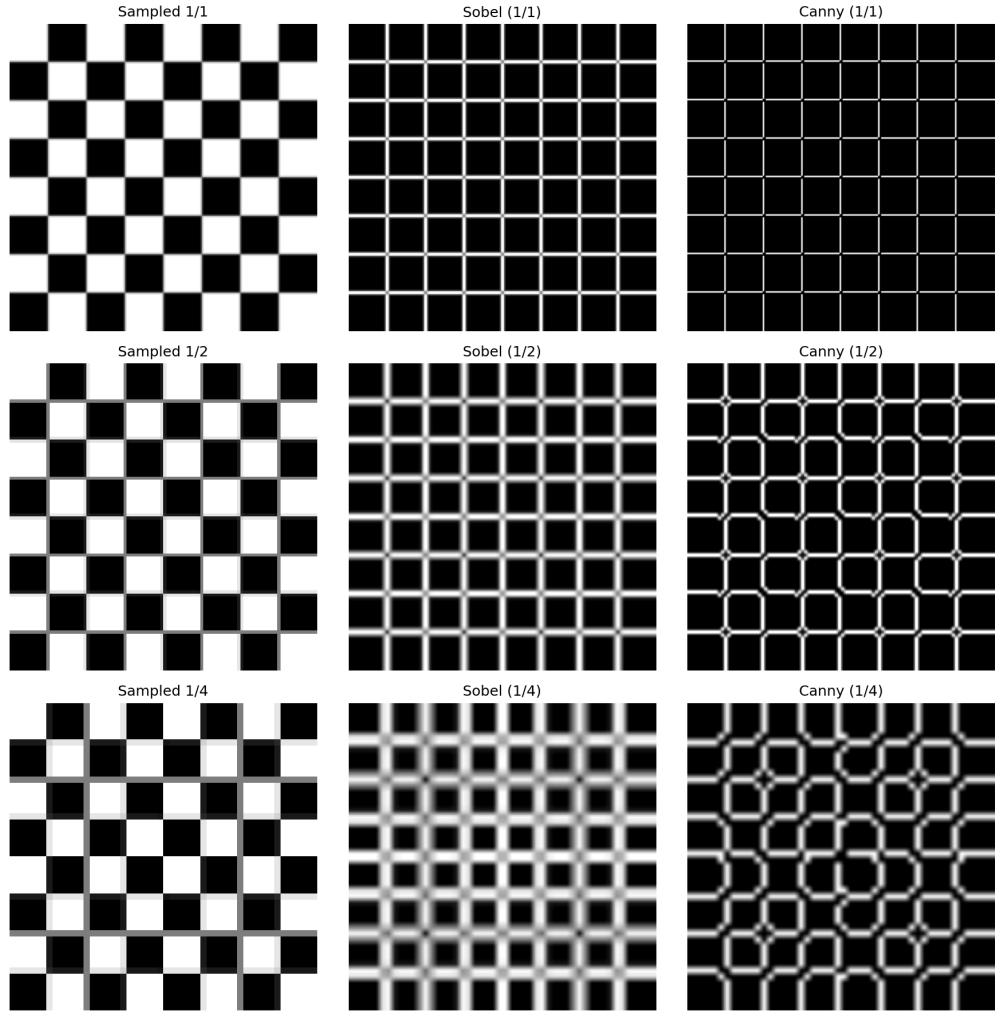
Astronaut

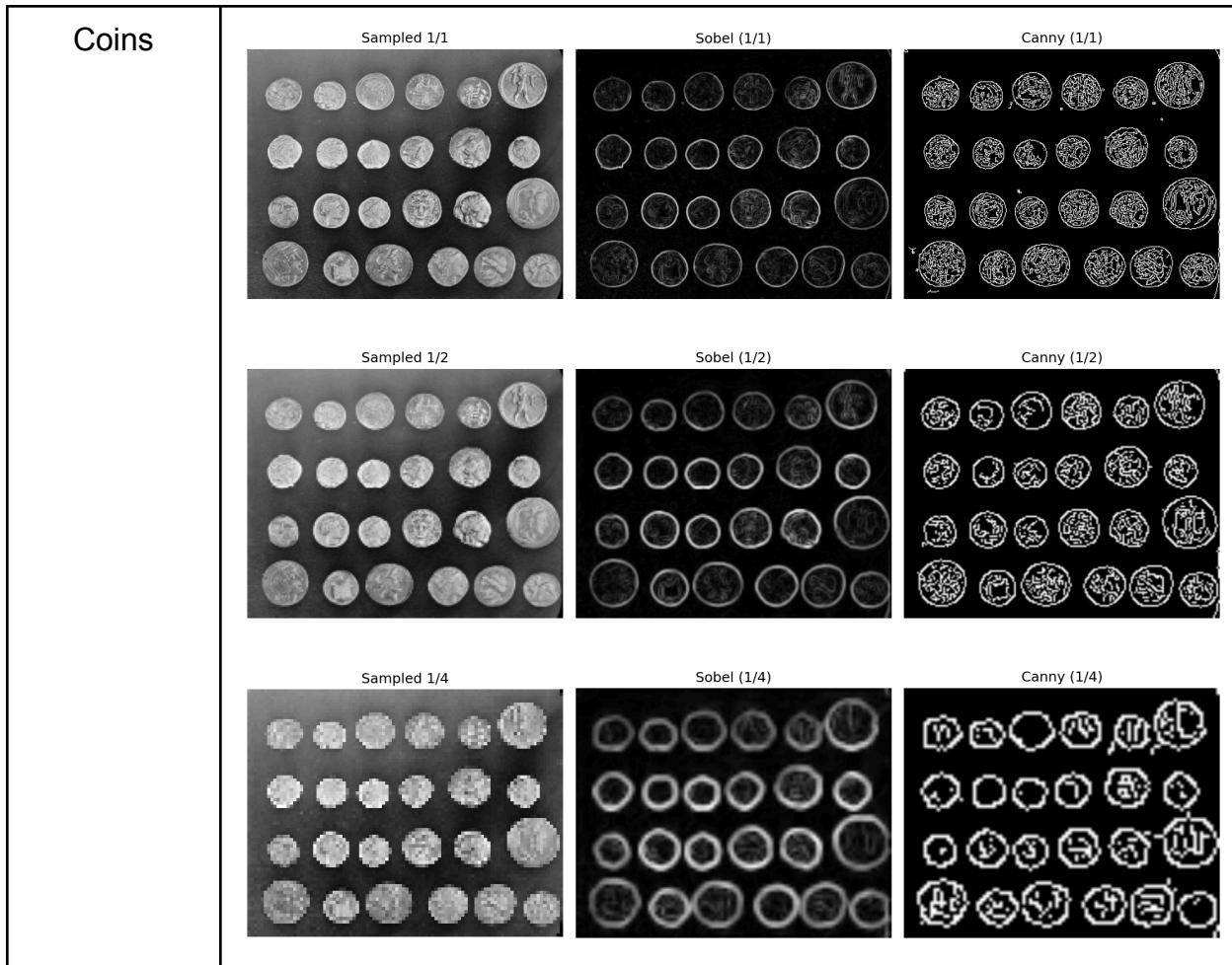


Cameraman



Checkboard





Proses downsampling atau pengurangan resolusi citra secara langsung mempengaruhi kualitas dan jumlah tepi yang dapat dideteksi oleh algoritma edge detection seperti Sobel dan Canny. Berdasarkan hasil percobaan, terlihat bahwa ketika ukuran citra berkurang, jumlah edge pixels juga menurun secara signifikan dari 100% pada resolusi asli (512×512) menjadi sekitar 75% pada resolusi 256×256 dan hanya sekitar 50% pada 128×128 .

Hal ini terjadi karena downsampling menghilangkan detail halus (fine structures) pada citra, terutama pada area dengan perubahan intensitas kecil. Akibatnya, Sobel dan Canny hanya mampu mendeteksi tepi yang kuat dan berukuran besar, sementara tepi halus atau tekstur kecil menjadi hilang. Pada citra hasil upsampling kembali, tepi yang hilang tidak dapat dipulihkan dengan baik dan justru menimbulkan pixelation artifacts yang membuat hasil deteksi terlihat kasar. Selain itu, semakin tinggi downsampling, semakin tinggi juga blur terhadap edge yang dihasilkan. Hal ini tampak sekali pada efek canny dengan $\frac{1}{4}$ sampling.

2.3 Feature Point Detection

2.3.1 Teori Singkat

Feature Points adalah titik-titik khas (distinctive points) pada citra yang memiliki karakteristik unik dan dapat dikenali kembali meskipun citra mengalami perubahan seperti rotasi, skala, atau pencahayaan. Titik-titik ini digunakan dalam berbagai aplikasi seperti image matching, object recognition, dan tracking. Beberapa metode populer untuk deteksi feature points antara lain:

1. Harris Corner Detector

Metode ini mendeteksi corner (sudut) dengan menganalisis perubahan intensitas pada setiap arah di sekitar suatu piksel menggunakan structure tensor (second moment matrix) MMM. Metode ini sensitif terhadap skala namun memiliki stabilitas yang baik terhadap noise.

2. FAST (Features from Accelerated Segment Test)

FAST mendeteksi corner dengan membandingkan intensitas piksel pusat dengan 16 piksel yang membentuk lingkaran di sekitarnya (circle of 16 pixels). Sebuah piksel dikategorikan sebagai corner jika terdapat sejumlah piksel berturut-turut di lingkaran tersebut yang jauh lebih terang atau gelap dari piksel pusat. Metode ini sangat efisien secara komputasi, sehingga cocok untuk aplikasi real-time seperti SLAM dan tracking pada video.

3. SIFT (Scale-Invariant Feature Transform)

SIFT mendeteksi keypoints yang invariant terhadap skala dan rotasi dengan membangun scale-space menggunakan filter Gaussian. Titik-titik ekstrim dalam Difference of Gaussian (DoG) diidentifikasi sebagai kandidat keypoints. Setiap keypoint kemudian diberi deskriptor berukuran 128 dimensi yang merepresentasikan pola gradien lokal di sekitarnya. Metode ini robust terhadap perubahan pencahayaan, rotasi, dan skala, sehingga sangat andal untuk feature matching antar citra.

2.3.2 Parameter

Berikut adalah parameter yang digunakan pada percobaan ini.

Detector	Parameter	Value	Keterangan
Harris	Threshold	$0.01 \times \text{max_response}$	1% dari maximum corner response
Harris	Block Size	2	Neighborhood size
Harris	k	0.04	Harris detector free parameter

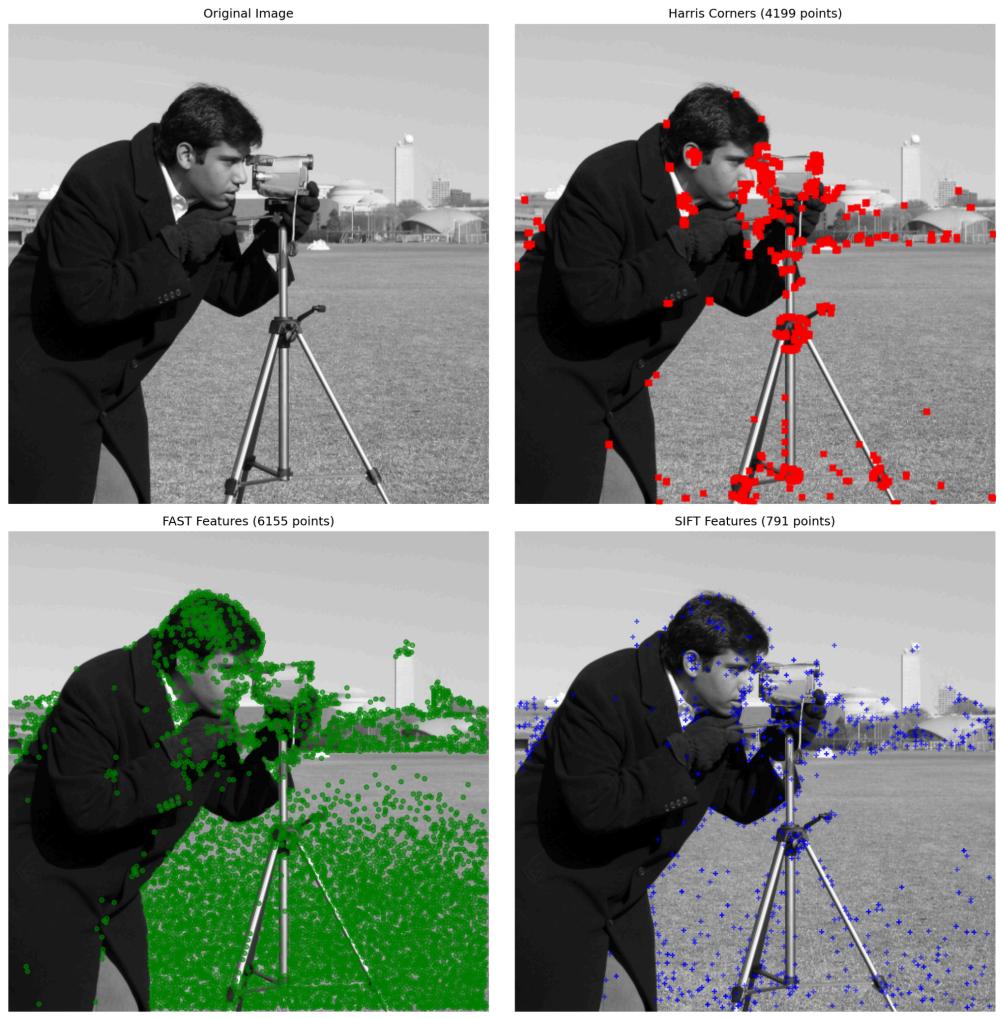
FAST	Threshold	10	Intensity difference threshold
SIFT	n_octaves	Auto	Scale space layers
SIFT	n_scales	Auto	Scales per octave

2.3.3 Hasil dan Analisis

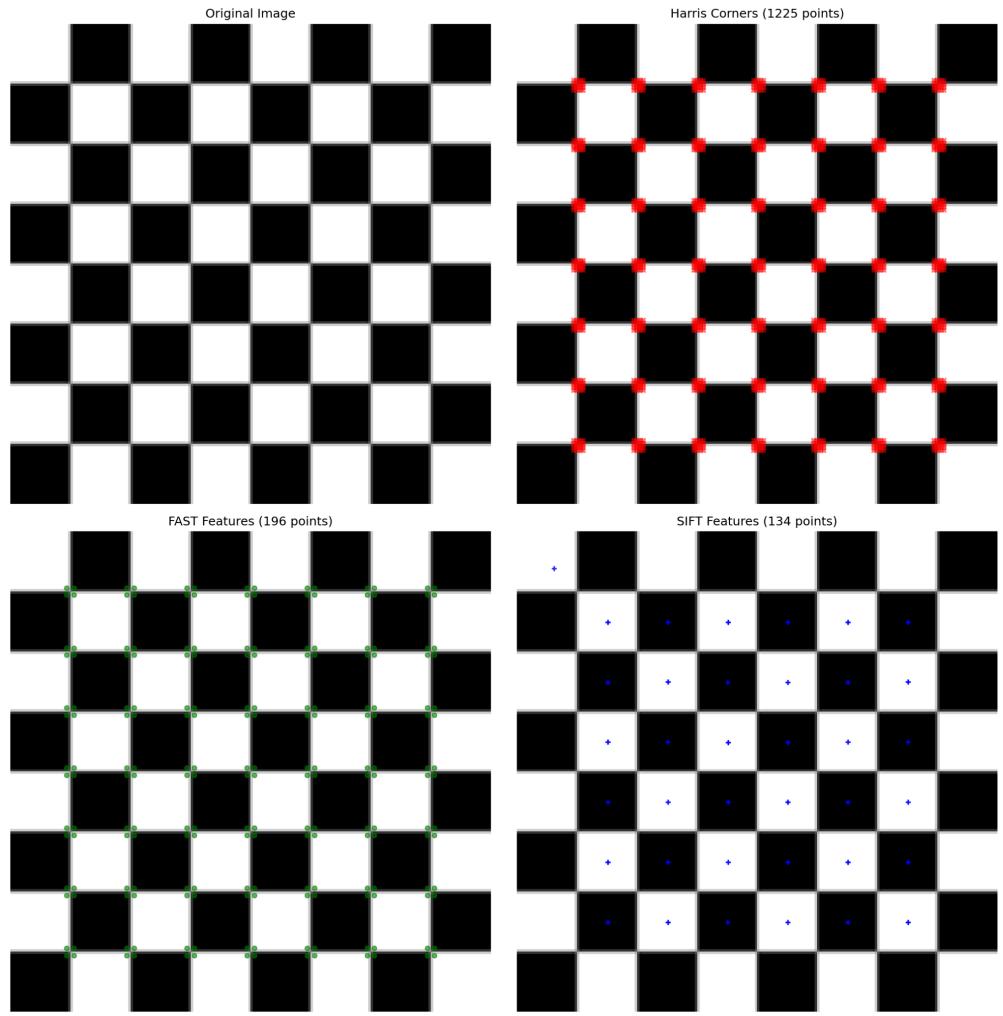
Tabel Perbandingan Ekstraksi Feature Point antar Algoritma

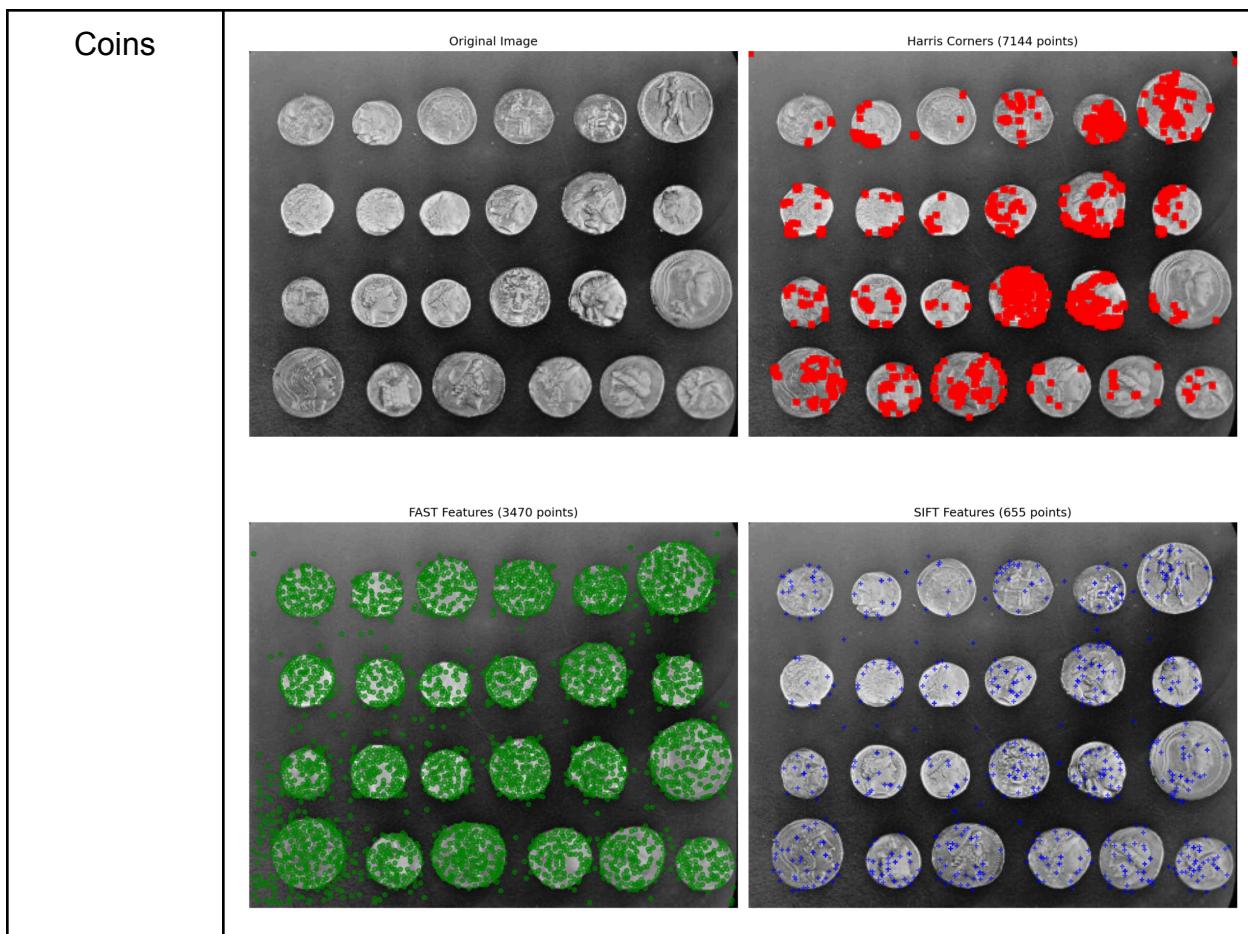
Gambar	Hasil			
Astronout	 Original Image	 Harris Corners (5942 points)	 FAST Features (3688 points)	 SIFT Features (1105 points)

Cameraman

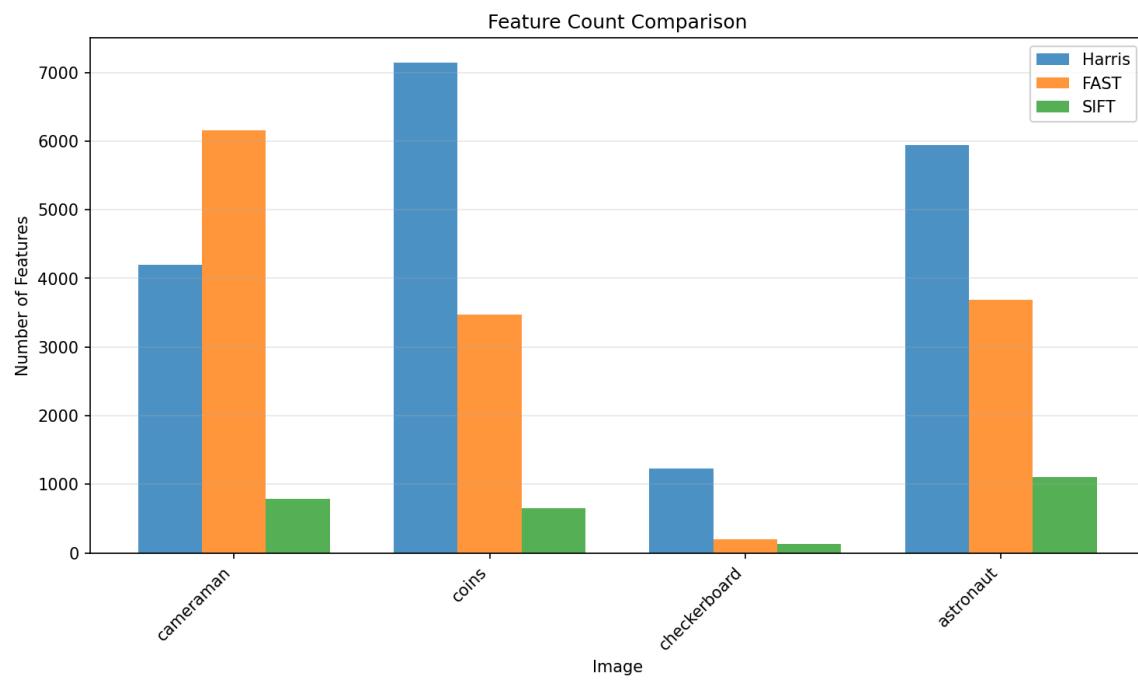


Checkboard





Grafik Perbandingan Jumlah Fitur



Tabel Statistik Algoritma terhadap Gambar

Image	Harris Count	Harris Max Response	Harris Mean Response	FAST Count	SIFT Count	SIFT Descriptor Dim
cameraman	4199	123564768.0	138116.484375	6155	791	128
coins	7144	47344292.0	153652.09375	3470	655	128
checkerboard	1225	87110368.0	1594082.25	196	134	128
astronaut	5942	165711248.0	229940.21875	3688	1105	128

Berdasarkan hasil eksperimen terhadap beberapa citra uji (cameraman, coins, checkerboard, dan astronaut), diperoleh statistik jumlah feature points dan karakteristik respon dari masing-masing metode (Harris, FAST, dan SIFT) seperti terlihat pada tabel.

1. Harris Corner Detector

Metode Harris menghasilkan jumlah titik yang cukup banyak pada sebagian besar citra, seperti pada cameraman (4199 titik) dan coins (7144 titik). Nilai mean response yang tinggi menunjukkan sensitivitas Harris terhadap perubahan intensitas lokal, khususnya pada area dengan pola tepi atau tekstur yang kuat. Namun, karena metode ini tidak invariant terhadap skala, Harris dapat mendeteksi banyak titik pada tekstur kecil atau detail halus, yang menyebabkan redundansi pada beberapa citra. Hal ini terlihat pada citra checkerboard, di mana setiap persilangan kotak terdeteksi sebagai corner (1225 titik) dengan mean response tertinggi (1.59×10^6).

2. FAST (Features from Accelerated Segment Test)

FAST menghasilkan jumlah fitur paling banyak pada citra seperti cameraman (6155 titik) karena algoritmanya yang agresif dalam mendeteksi perubahan intensitas lokal. Namun, metode ini cenderung menghasilkan banyak false positives di area bertekstur halus atau berisik, sebagaimana terlihat pada area rumput di gambar cameraman. Meskipun efisien secara komputasi dan cocok untuk aplikasi real-time, kualitas fitur FAST relatif lebih rendah dibandingkan Harris dan SIFT karena tidak memperhitungkan skala maupun orientasi.

3. SIFT (Scale-Invariant Feature Transform)

SIFT menghasilkan jumlah fitur paling sedikit (contohnya 791 titik pada cameraman), namun setiap titik memiliki deskriptor berdimensi tinggi (128 dimensi) yang merepresentasikan pola gradien secara detail. Kelebihan utama SIFT adalah kestabilannya terhadap perubahan skala, rotasi, dan pencahayaan. Meskipun jumlah titik lebih sedikit, fitur yang dihasilkan lebih diskriminatif dan konsisten di antara berbagai citra, seperti terlihat pada astronaut (1105 titik) yang memiliki tekstur kompleks.

2.4 Camera Geometry dan Calibration

2.4.1 Teori Singkat

Camera Calibration adalah proses untuk menentukan parameter-parameter kamera agar hubungan antara koordinat dunia nyata (3D) dan citra (2D) dapat dimodelkan secara akurat. Proses ini menghasilkan dua kelompok parameter utama:

1. Intrinsic Parameters — menggambarkan karakteristik internal kamera:

- Focal Length (f_x, f_y): menyatakan jarak fokus dalam satuan piksel pada sumbu x dan y.
- Principal Point (c_x, c_y): titik pusat proyeksi (biasanya dekat dengan tengah citra).
- Distortion Coefficients (k_1, k_2, p_1, p_2): mengoreksi distorsi lensa (radial dan tangensial).
- Camera Matrix (K): merepresentasikan parameter intrinsik dalam bentuk matriks:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

2. Extrinsic Parameters — mendeskripsikan posisi dan orientasi kamera terhadap dunia nyata:

- Rotation Matrix (R): transformasi orientasi antara koordinat dunia dan kamera.
- Translation Vector (t): posisi pusat kamera terhadap sistem koordinat dunia.

Selain itu, Geometric Transformations digunakan untuk memodelkan hubungan spasial antar gambar:

- Affine Transformation: transformasi linear dengan 6 derajat kebebasan (rotasi, translasi, skala, dan shear).

- Perspective (Projective) Transformation: transformasi non-linear dengan 8 derajat kebebasan yang mempertahankan garis lurus tetapi tidak paralelisme (umum pada citra hasil proyeksi kamera nyata).

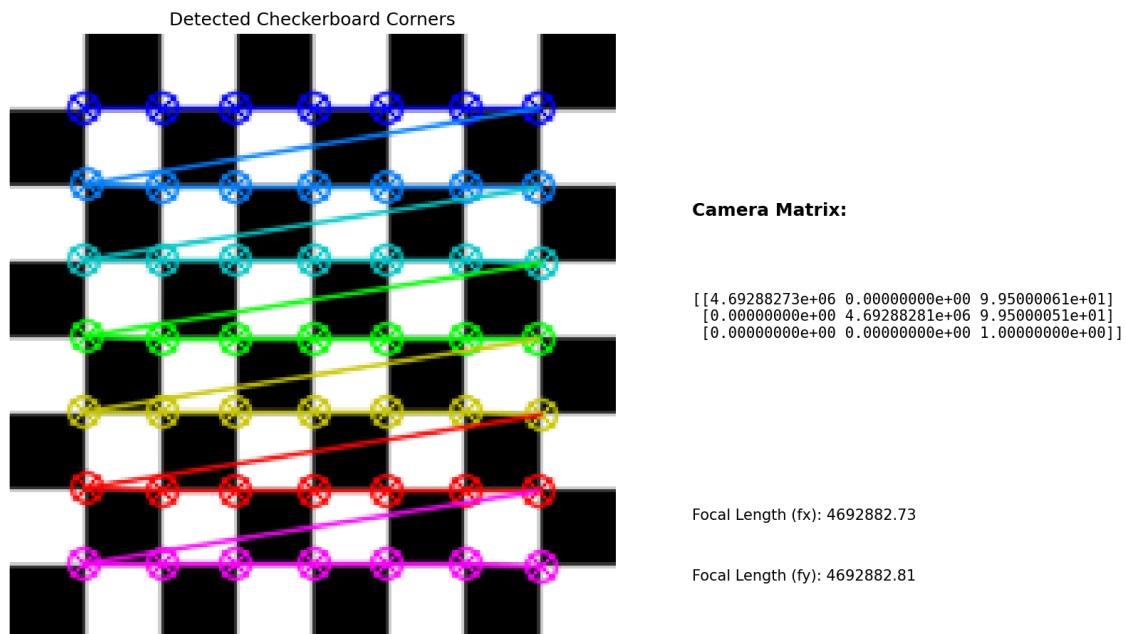
2.4.1 Camera Calibration

2.4.1.1 Parameter

Parameter	Value	Keterangan
Pattern Size	7x7	Inner corners count
Square Size	1.0 unit	Calibration object size
Corner Refinement	SubPix (11x11)	Refine corner locations

2.4.1.2 Hasil dan Analisis

Pada percobaan ini, digunakan checkerboard corner detection sebelum melakukan camera calibration dan hanya terdapat satu gambar yang berhasil dilakukan checkerboard corner detection, yaitu checkerboard image. Berikut adalah hasil camera calibration untuk checkerboard.



Berikut adalah camera matrix yang berhasil dideteksi saat camera calibration:

$$K = \begin{bmatrix} 4.69288273 \times 10^6 & 0.00000000 & 99.5000061 \\ 0.00000000 & 4.69288281 \times 10^6 & 99.5000051 \\ 0.00000000 & 0.00000000 & 1.00000000 \end{bmatrix}$$

Adapun hasil lainnya adalah sebagai berikut:

Focal_X	Focal_Y	Principal_Point_X	Principal_Point_Y
4692882.73	4692882.82	99.5	99.5

2.4.2 Transformasi Geometri

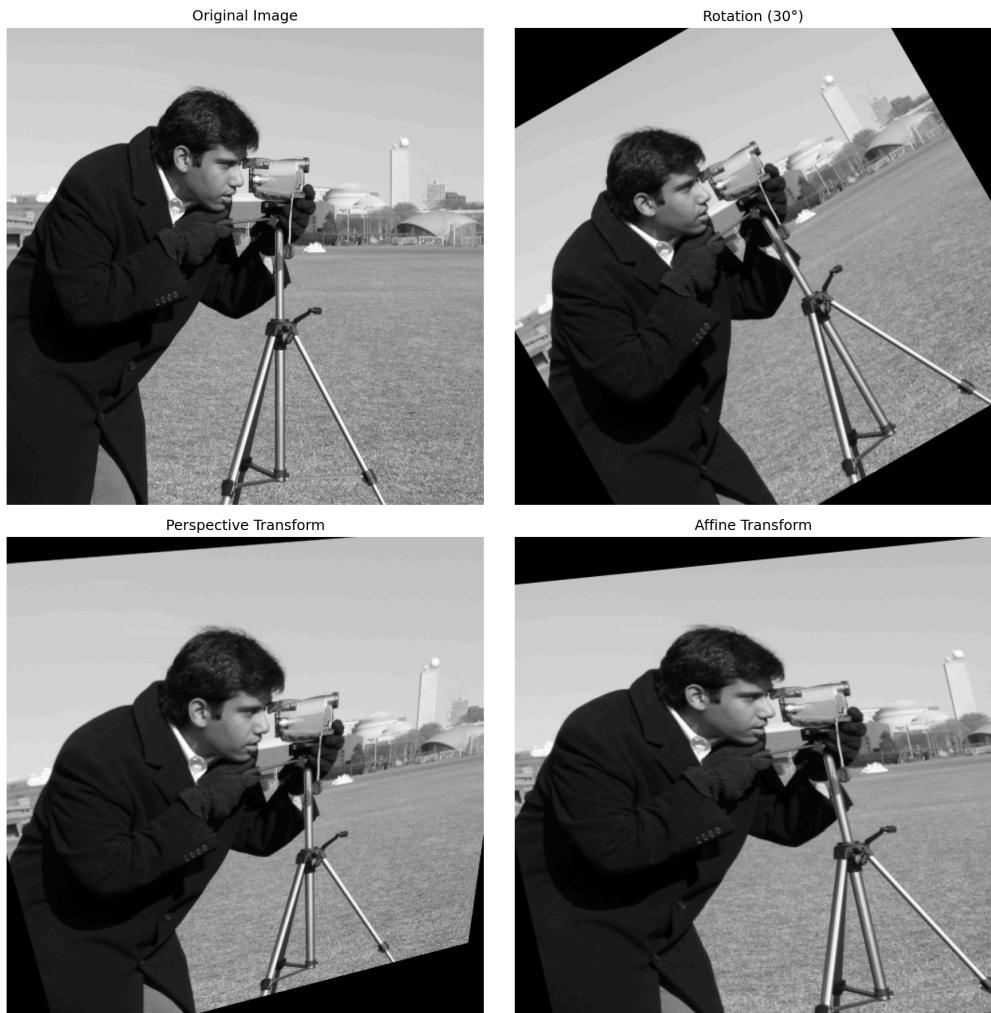
2.4.2.1 Parameter

Transform Type	Parameters	Description
Rotation	angle=30°, center=(w/2,h/2)	Rotate around image center
Perspective	4 point correspondences	Simulate 3D perspective
Affine	3 point correspondences	Linear mapping

2.4.2.2 Hasil dan Analisis

Gambar	Hasil
Astronaut	<div style="display: flex; justify-content: space-around;"> <div data-bbox="437 354 915 830">  <p>Original Image</p> </div> <div data-bbox="943 354 1421 830">  <p>Rotation (30°)</p> </div> </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div data-bbox="437 861 915 1336">  <p>Perspective Transform</p> </div> <div data-bbox="943 861 1421 1336">  <p>Affine Transform</p> </div> </div> <div style="margin-top: 20px;"> <p>Rotation Matrix (30 degrees):</p> $[[0.8660254 \quad 0.5 \quad -93.70250337] \\ [-0.5 \quad 0.8660254 \quad 162.29749663]]$ </div> <div style="margin-top: 20px;"> <p>Perspective Transform Matrix:</p> $[[1.55531467e+00 \quad 3.34994035e-01 \quad -8.37480806e+01] \\ [-1.03828719e-01 \quad 1.55152096e+00 \quad 3.52889355e+01] \\ [5.36861246e-04 \quad 9.97848290e-04 \quad 1.00000000e+00]]$ </div> <div style="margin-top: 20px;"> <p>Affine Transform Matrix:</p> $[[1.19417476 \quad 0.2184466 \quad -60.63106796] \\ [-0.12135922 \quad 0.97572816 \quad 57.2815534]]$ </div> <div style="margin-top: 20px;"> <p>Camera Matrix: Not computed (no checkerboard detected)</p> </div>

Cameraman



Rotation Matrix (30 degrees):

$$[[0.8660254 \quad 0.5 \quad -93.70250337] \\ [-0.5 \quad 0.8660254 \quad 162.29749663]]$$

Perspective Transform Matrix:

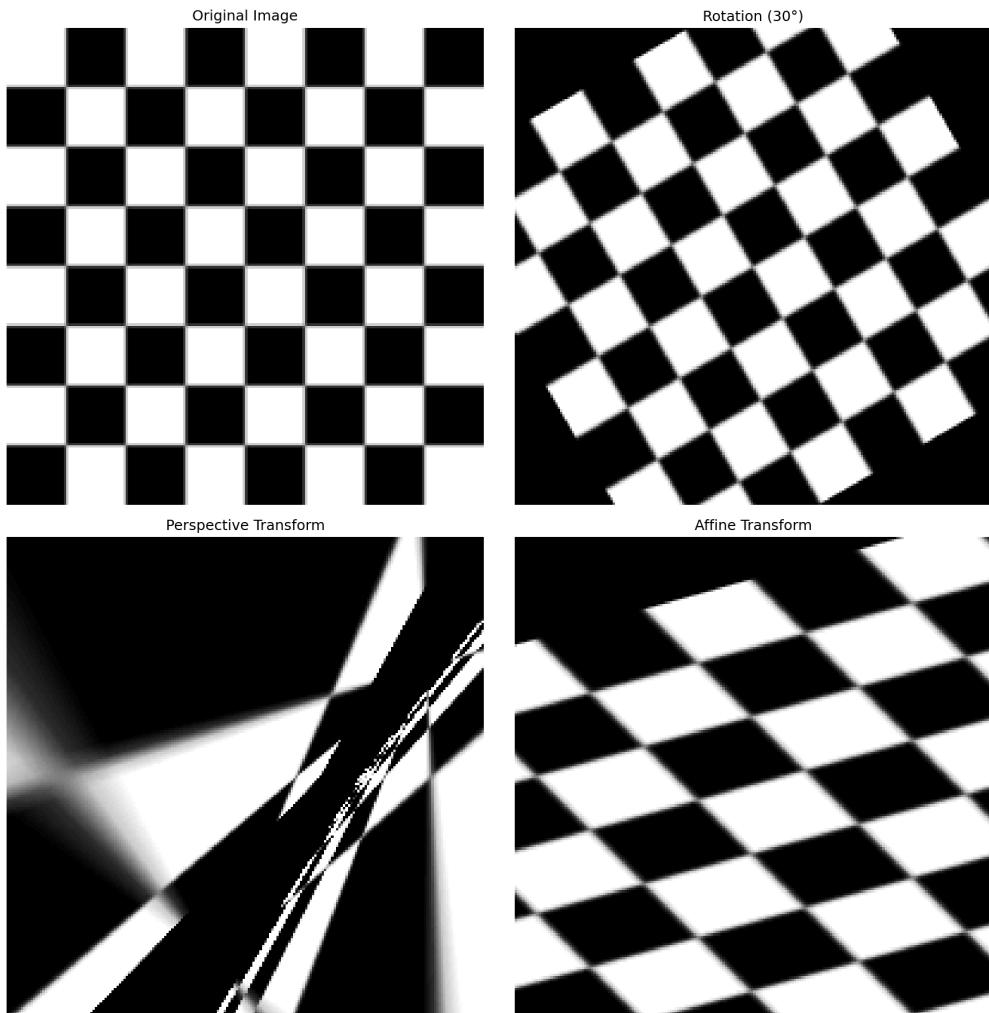
$$[[1.55531467e+00 \quad 3.34994035e-01 \quad -8.37480806e+01] \\ [-1.03828719e-01 \quad 1.55152096e+00 \quad 3.52889355e+01] \\ [5.36861246e-04 \quad 9.97848290e-04 \quad 1.00000000e+00]]$$

Affine Transform Matrix:

$$[[1.19417476 \quad 0.2184466 \quad -60.63106796] \\ [-0.12135922 \quad 0.97572816 \quad 57.2815534]]$$

Camera Matrix: Not computed (no checkerboard detected)

Checkboard



Rotation Matrix (30 degrees):

$$[[0.8660254 \quad 0.5 \quad -36.60254038] \\ [-0.5 \quad 0.8660254 \quad 63.39745962]]$$

Perspective Transform Matrix:

$$[[-2.00416667e+00 \quad -5.92500000e-01 \quad 1.30750000e+02] \\ [-6.16666667e-01 \quad -1.20000000e+00 \quad 1.00000000e+02] \\ [-1.14166667e-02 \quad -6.75000000e-03 \quad 1.00000000e+00]]$$

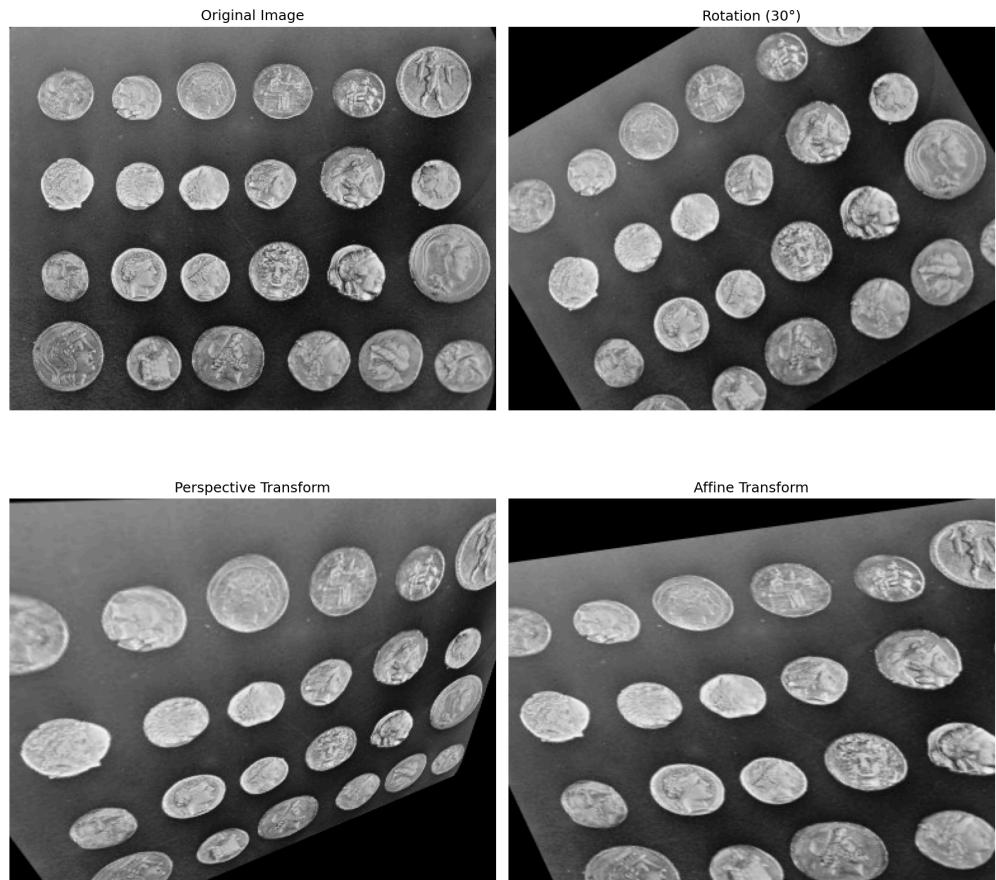
Affine Transform Matrix:

$$[[1.8 \quad 0.9 \quad -125.] \\ [-0.5 \quad 0.9 \quad 80.]]$$

Camera Matrix:

$$[[4.69288273e+06 \quad 0.00000000e+00 \quad 9.95000061e+01] \\ [0.00000000e+00 \quad 4.69288281e+06 \quad 9.95000051e+01] \\ [0.00000000e+00 \quad 0.00000000e+00 \quad 1.00000000e+00]]$$

Coins



Rotation Matrix (30 degrees):

$$[[0.8660254 \quad 0.5 \quad -49.77687753] \\ [-0.5 \quad 0.8660254 \quad 116.23016403]]$$

Perspective Transform Matrix:

$$[[2.97428918e+00 \quad 1.07162600e+00 \quad -1.88416275e+02] \\ [-8.45483828e-02 \quad 2.65647524e+00 \quad 1.01985008e+01] \\ [3.19617473e-03 \quad 4.56279403e-03 \quad 1.00000000e+00]]$$

Affine Transform Matrix:

$$[[1.28169014 \quad 0.44334975 \quad -76.25199473] \\ [-0.17605634 \quad 0.95073892 \quad 61.26587109]]$$

Camera Matrix: Not computed (no checkerboard detected)

3. Refleksi Pribadi

3.1 Narasi Pilihan Desain Pribadi

- **Mengapa Gaussian, Median, dan Sobel untuk Filtering?**

Pemilihan ketiga filter ini didasarkan pada karakteristik dan keunggulan masing-masing. Gaussian filter merupakan standar dalam reduksi noise karena memiliki dasar matematis yang kuat dan menghasilkan smoothing yang halus. Median filter bersifat non-linear dan sangat efektif untuk mengatasi salt-and-pepper noise tanpa terlalu mengaburkan tepi. Sementara itu, Sobel filter dipilih karena sederhana tetapi efektif dalam mempertegas tepi (edge enhancement) serta efisien secara komputasi.

- **Mengapa Sobel dan Canny untuk Edge Detection?**

Sobel digunakan sebagai metode baseline karena cepat, mudah diimplementasikan, dan membantu memahami konsep dasar gradien citra. Sebaliknya, Canny dipilih karena mewakili pendekatan modern dengan tahapan multistage yang memberikan hasil deteksi tepi paling optimal di berbagai kondisi.

- **Mengapa Harris, FAST, dan SIFT untuk Feature Detection?**

Ketiga algoritma ini dipilih untuk mencakup berbagai generasi metode deteksi fitur. Harris merupakan detektor klasik yang andal untuk mendekripsi sudut (corner). FAST menawarkan performa tinggi dan efisiensi untuk aplikasi real-time. SIFT menjadi representasi metode yang paling robust karena invariannya terhadap skala dan rotasi.

3.2 Lesson Learned

1. Technical Skills

Melalui percobaan ini, saya memperoleh pemahaman mendalam mengenai berbagai teknik filtering dan edge detection, serta pengalaman langsung dalam feature detection dan matching. Saya juga mempelajari proses camera calibration dan geometric transformation secara praktis.

2. Software Engineering

Dari sisi rekayasa perangkat lunak, saya memahami pentingnya desain modular untuk menjaga keteraturan kode, penerapan error handling dan graceful degradation agar sistem tetap stabil, serta dokumentasi yang komprehensif. Selain itu, praktik automated testing membantu memastikan validitas hasil dan mencegah regresi.

3. Computer Vision Insights

Saya menyadari bahwa tidak ada algoritma yang benar-benar terbaik, semuanya bergantung pada konteks dan kebutuhan. Proses parameter tuning sangat krusial untuk mendapatkan hasil optimal, dan citra dunia nyata jauh lebih menantang dibanding dataset penelitian. Setiap pendekatan memiliki trade-off tersendiri, seperti kecepatan versus akurasi, serta sensitivitas versus spesifisitas.

Lampiran

Lampiran A: Link Source Code

https://github.com/ninoaddict/Adril_13522068_IF5152_TugasIndividuCV

Lampiran B: Project Structure

```
└── 01_filtering/
    ├── image_filtering.py
    └── generated/
        ├── *_filtering_comparison.png
        └── filtering_parameters.csv
└── 02_edge/
    ├── edge_detection.py
    └── generated/
        ├── *_edge_comparison.png
        ├── *_sampling_analysis.png
        └── edge_parameters.csv
└── 03_featurepoints/
    ├── feature_point_detection.py
    └── generated/
        ├── *_feature_marking.png
        ├── feature_statistics.csv
        └── feature_comparison.png
└── 04_geometry/
    ├── camera_geometry.py
    └── generated/
        ├── *_calibration.png
        ├── *_transformations.png
        ├── *_matrices.txt
        └── geometry_parameters.csv
└── 05_laporan.pdf
└── README.md
└── main.py
```