

# IMPLEMENTASI METODE DETEKSI TEPI CANNY UNTUK MENGHITUNG JUMLAH UANG KOIN DALAM GAMBAR MENGGUNAKAN OPENCV

Julia Ulfah<sup>1\*</sup>, Nurdin<sup>2</sup>

<sup>1,2</sup>Program Studi Magister Teknologi Informasi, Fakultas Teknik, Universitas Malikussaleh; Jl. Batam, Kampus Bukit Indah - Lhokseumawe, Aceh, Telepon 0645-44450

*Riwayat artikel:*

*Received: 10 Juni 2023*

*Accepted: 10 Juli 2023*

*Published: 1 Agustus 2023*

## Keywords:

Pengolahan Citra;  
Deteksi tepi Canny;  
Gaussian filtering;  
OpenCV;

## Correspondent Email:

julia.227110201008@mhs.unimal.ac.id

© 2023 JITET (Jurnal Informatika dan Teknik Elektro Terapan). This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY NC)

**Abstrak.** Deteksi tepi sering digunakan pada pengolahan citra untuk menemukan batas-batas tepi dari objek gambar dan hal tersebut penting dilakukan untuk menyeleksi tepi bagian citra. Deteksi tepi citra yang digunakan dalam penelitian ini menghitung koin dalam sebuah gambar, sehingga bisa dengan mudah melalui mata manusia tetapi tidak dengan computer. Pada sistem komputer perlu diberikan program dan data uji agar komputer bisa membaca dan menghitung koin layaknya seperti manusia. Tujuan dari penelitian ini adalah untuk menghitung uang koin dalam sebuah gambar menggunakan metode deteksi tepi Canny dan Gaussian filtering. Adapun gambar yang digunakan dalam penelitian ini dalam bentuk \*.jpg dan ujicoba memakai aplikasi Visual studio Code menggunakan OpenCV dan Python. Dari hasil penelitian ini menunjukkan bahwa hasil jumlah koin yang terbaca dari gambar yang diinputkan ke sistem dapat dibaca dengan baik, dengan keakurasian data 90%. Metode deteksi tepi Canny dapat digunakan untuk menghitung jumlah uang koin dalam gambar.

**Abstract.** Edge detection is often used in image processing to find the edge boundaries of image objects and this is important for selecting the edges of image parts. The image edge detection used in this study counts coins in an image, so that it can easily pass through the human eye but not by a computer. In a computer system, it is necessary to provide programs and test data so that the computer can read and count coins like a human. The purpose of this study is to count coins in an image using Canny edge detection and Gaussian filtering methods. The images used in this study are in the form of \*.jpg and tested using the Visual Studio Code application using OpenCV and Python. The results of this study indicate that the results of the number of coins read from the images inputted to the system can be read properly, with a data accuracy of 90%. Canny's edge detection method can be used to count the number of coins in an image.

## 1. PENDAHULUAN

Komputer membutuhkan sebuah citra digital sebagai inputan untuk menampilkan informasi yang dimilikinya, dikarenakan komputer tidak sama seperti mata manusia yang langsung dapat mengenali atau menghitung objek yang dilihat, maka berbeda dengan

komputer harus mengolah gambar / citra yang untuk mengetahui output tersebut apa atau berapa hasil hitungannya, maka diperlukan pengolahan citra pada komputer tersebut. Pengolahan citra adalah suatu bidang ilmu yang berisi bagaimana cara mengubah sebuah citra menjadi bentuk citra yang lain dengan

mempergunakan teknik / suatu metode yang bertujuan mendapatkan informasi dari citra yang diubah tersebut guna memudahkan manusia.

Ber macam-macam aplikasi dan cabang keilmuan yang dibuat berdasarkan ilmu pengolahan citra contohnya seperti aplikasi penginderaan jarak jauh menggunakan satelit maupun pesawat, aplikasi pengenalan pola, dan juga *machine vision*. Pengolahan citra membuat suatu citra diolah menjadi citra yang mempunyai kualitas yang lebih bagus sehingga dapat dikenali dengan mudah oleh manusia maupun mesin komputer. Untuk mengolah citra yang masih asli dibutuhkan deteksi tepi untuk menghilangkan *noise-noise* atau hal-hal yang mengganggu citra itu agar dapat terbaca oleh komputer. Ada beberapa Metode dalam deteksi tepi pada citra ada diantaranya metode *Sobel*, *Canny*, *Laplace*. Penelitian ini menggunakan Metode Canny dalam menghitung jumlah objek koin dalam sebuah gambar.

Penelitian sebelumnya yang mirip ini telah dilakukan oleh Resianta Perangin-angin yang berjudul *Comparison Detection Edge Lines Algoritma Canny dan Sobel* dalam penelitiannya melakukan perbandingan deteksi tepi pada objek berupa Tas Wanita, didapatkan hasil deteksi tepi yang didapat canny memiliki hasil lebih spesifik dan juga halus hasil garis tepi (*edge*) dari sebuah gambar (tas) [1]. Penelitian lainnya oleh Damar Zanuvar Eka Prastya, dkk yaitu “Implementasi Metode Gaussian Filter Dan Median Filter Untuk Penghalusan Gambar” [2] menerangkan bahwa dalam penelitiannya metode gaussian filter sangat baik untuk mengurangi derau (*noise*) dalam gambar. Lalu penelitian yang dilakukan oleh Safriadi, dkk tentang “Analisis Deteksi Tepi Canny Pada Citra Dengan Gaussian Filtering Dan Bilateral Filtering” [3] menjelaskan bahwa untuk gambar yang lebih halus performa bilateral filtering lebih bagus dari gaussian filtering.

Maka dari itu penulis ingin membuat penelitian tentang implementasi metode Canny dan menghitung jumlah objek yang ada dalam sebuah gambar atau citra menggunakan aplikasi visual studio code dan bahasa pemrograman python yang mana aplikasi ini free dan lebih mudah diaplikasikan.

Tujuan dari penelitian ini yaitu penulis ingin menghitung objek yang terdapat dalam

gambar dalam hal ini objeknya adalah berupa koin yang ada dalam sebuah gambar, sehingga kita bisa mengetahui ada berapa koin yang ada dalam gambar tersebut serta melihat bagaimana hasil deteksi tepi Canny diterapkan dalam mendeteksi tepi koin dalam gambar.

## 2. TINJAUAN PUSTAKA

### a. Citra

Citra ialah suatu hal yang menggambarkan kemiripan atau imitasi dari suatu objek atau gambar [4]. Citra adalah hasil output dari suatu sistem pengambilan data. Cara mengambil atau merekam itu bisa berbentuk digital yang file digital yang telah direkam itu langsung bisa disimpan ke dalam media penyimpanan tertentu atau bisa berbentuk analog contohnya : gambar dalam monitor televisi, sinyal-sinyal video, serta bisa juga berbentuk optik contohnya : foto.

### b. Citra Digital

Pengolahan citra digital merujuk kepada cara memproses citra dua dimensi mempergunakan media computer [5]. Citra digital harus direpresentasikan (digambarkan) dalam bentuk angka numerik dengan nilai-nilai secara diskrit atau disebut dengan nilai intensitas cahaya, tujuannya agar komputer dapat mengolah citra digital.

Nilai-nilai intensitas cahaya yang dimaksud digambarkan sebagai nilai-nilai kanal pada sebuah citra digital. Citra yang berukuran 8 bit mempunyai 1 kanal berisi sekelompok nilai dalam rentang nilai 0 – 255, pada citra 16 bit mempunyai 2 kanal, dan pada citra yang berukuran 24 bit mempunyai 3 kanal yang disebut dengan istilah kanal RGB dimana R (*red*), G (*green*), dan B (*blue*).

### c. Pengenalan Pola

Pola merupakan bentuk/wujud yang dapat diidentifikasi dan dapat diartikan dari ciri-ciri yang ada padanya (*features*) [6]. Ciri-ciri yang didapatkan bertujuan sebagai pembeda antara sebuah pola dengan pola yang lain. Suatu ciri-ciri yang dikatakan baik ialah ciri yang memiliki hal pembeda yang beragam dan juga unik, sehingga pola yang terbentuk dari ciri tersebut mempunyai ketepatan yang tinggi.

Suatu pola mendapatkan ciri-cirinya dari hasil penilaian terhadap objek yang diuji.

Pengenalan pola mempunyai tujuan untuk menetapkan atau mengetahui kategori pola yang mana merujuk ciri-ciri yang dimiliki oleh pola tersebut. Pengenalan pola berfungsi melihat perbedaan yang ada pada objek yang satu dengan objek lainnya.

d. *Resizing dan Cropping Citra*

Proses resize dan Crop ini tujuannya untuk mengubah ukuran atau resolusi gambar baik secara vertikal maupun secara horizontal pada suatu citra (gambar) [7]. Citra misalkan diubah ke ukuran 50x50 dalam satuan piksel agar nilai vektor yang diambil menjadi satu macam ukuran yang sama sehingga nantinya memudahkan proses klasifikasi.

e. *Grayscale*

*Grayscale* ialah piksel warna yang berada pada nilai antara gradasi warna hitam dan warna putih [5]. Citra *Grayscale* mempunyai intensitas nilai dalam rentang nilai angka 0 - 255, dimana intensitas nilai pada citra *grayscale* juga merupakan gambaran dari nilai derajat keabuan. nilai 0 menjelaskan kepada warna hitam yang sempurna sedangkan nilai 255 menjelaskan kepada warna putih yang sempurna. Intensitas nilai antara nilai 0 sampai dengan nilai 255 merupakan warna abu-abu.

f. *Gaussian Filtering*

Gaussian filter ialah salah satu metode yang sering digunakan untuk menghilangkan *noise* dengan sangat baik yang sifatnya sebaran normal, sering kali kita jumpai pada citra atau gambar dari hasil proses pengambilan citra digital yang memakai kamera digital, karena hal tersebut adalah kondisi alamiah berakibat dari kepekaan sensor cahaya pada kamera yang dipakai dan cahaya yang dipantulkannya [8].

Nilai-nilai elemen pada filter gaussian dapat dihitung menggunakan persamaan 1.

$$G(i, j) = c \cdot e^{-\frac{(i-u)^2 + (j-v)^2}{2\sigma^2}} \dots\dots\dots (1)$$

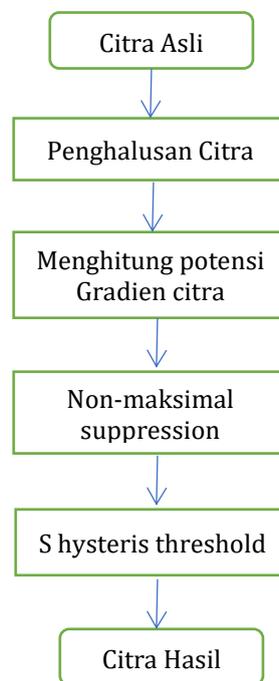
Keterangan :

- c dan  $\sigma$  = konstanta
- G (i,j) = Posisi (i,j)  
                  elemen matriks kernel
- gauss
- (u,v) = Indeks tengah dari matriks  
                  Kernel gauss

g. *Deteksi Tepi Canny*

Salah satu algoritma yang sering dipakai untuk mendeteksi tepi pada gambar salah satunya ialah deteksi tepi canny. Deteksi tepi ini dapat melakukan deteksi tepian gambar dari citra asli (awal) dengan hasil error yang sangat sedikit serta didesain untuk mengeluarkan output berupa tepi citra yang sangat optimal dan bagus [7]. Operator canny dapat menghasilkan hasil deteksi tepi yang halus dan dapat menyaring kegaduhan dari citra awal menggunakan *Gaussian Derivative Kernel*.

Berikut adalah bagan alur metode deteksi tepi canny ditunjukkan oleh gambar 1.



Gambar 1. Bagan Alur deteksi tepi Canny

h. *Uang Koin*

Bentuk koin yang dipakai dalam penelitian ini yaitu berupa uang koin dalam bentuk mata uang Rupiah, yang nilai mata uangnya berbeda-beda, diantaranya yaitu uang koin rupiah Rp. 100, Rp. 200, Rp. 500, dan Rp. 1000. Uang koin ini ada yang berbahan aluminium dan juga ada yang berbahan logam dan perak.

i. *Visual Studio Code*

Visual Studio Code adalah salah satu aplikasi untuk menulis kode program (editor) yang bersifat *code open source* yang dibuat oleh perusahaan Microsoft, aplikasi ini dapat beroperasi pada sistem operasi Windows, Mac OS dan sistem operasi Linux [9].

Visual studio Code mempermudah kita untuk menulis kode program dan bisa digunakan untuk bahasa pemrograman, contohnya bahasa pemrograman PHP, Python, Java, C#, C++, GO.

Visual studio Code juga berkemampuan untuk mengenal bahasa pemrograman apa yang sedang dipakai dan fungsi-fungsi yang dipakai dalam susunan kode program itu nantinya diberikan warna yang bervariasi sebagai pembeda.

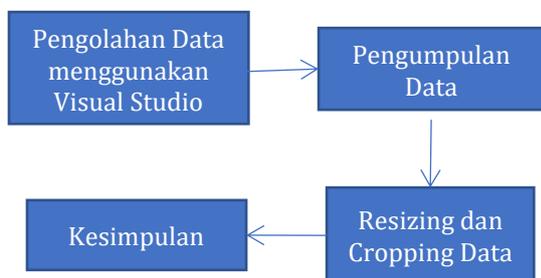
j. OpenCV dan Pemrograman Python

OpenCV (*Open Computer Vision*) ialah *library open source* yang dapat melakukan fungsi pengolahan citra, yang bertujuan untuk mampu mengolah data secara visual yang biasanya dilakukan oleh manusia dapat dilakukan juga oleh komputer.

Pemrograman Python merupakan salah satu bahasa pemrograman level tingkat tinggi yang mempunyai sifat berorientasi objek, interaktif, *interpreter*, dan bisa kita gunakan atau bisa beroperasi di hampir di semua platform seperti : sistem operasi Windows, sistem operasi Linux, dan juga Mac OS [10]. Pemrograman python juga merupakan jenis bahasa pemrograman yang jelas sintaksnya, singkat, efisien dan bisa dikombinasikan dengan modul lain yang siap pakai sehingga memudahkan untuk kita pelajari.

3. METODE PENELITIAN

Tahapan penelitian yang telah dilakukan ditunjukkan oleh gambar 2 berikut :



Gambar 2. Tahapan Penelitian

3.1 Pengumpulan Data

Data yang dipergunakan didapatkan dari data gambar atau citra yang diambil menggunakan kamera handphone, citra tersebut menggunakan format file .JPG berjumlah 10 buah data gambar uang koin dengan jumlah koin yang berbeda-beda dalam setiap gambarnya. Berikut adalah data gambar awal yang didapatkan ditunjukkan dalam tabel 1.

Tabel 1. Data Gambar Uang Koin

No	Nama File	Ukuran	Gambar Awal
1	koinA	7,86 kb	
2	koinB	6,40 kb	
3	koinC	7,55 kb	
4	koinD	8,34 kb	
5	koinE	6,60 kb	
6	koinF	7,47 kb	
7	koinG	5,83 kb	
8	koinH	6,53 kb	
9	koinI	10,8 kb	

10	koinJ	8,12 kb	
----	-------	---------	---

### 3.2 Resizing dan Cropping Data

Pada tahap resizing data ini data berupa gambar yang sudah dikumpulkan diubah ukurannya menjadi ukuran 5 - 11 kb.

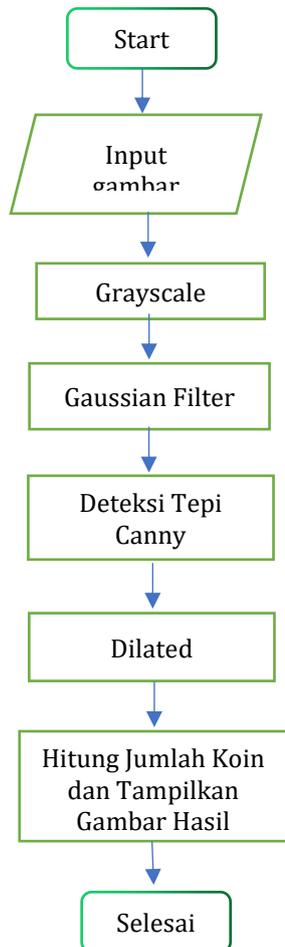
### 3.3 Pengolahan Data dengan VS Code

Tahap selanjutnya mengolah data gambar yang telah di ubah ukurannya menggunakan metode deteksi Canny pada visual studio code untuk menghitung jumlah koin yang ada dalam gambar.

### 3.4 Kesimpulan

Tahap ini adalah mengambil kesimpulan terhadap penelitian yang telah dilakukan.

Berikut adalah flowchart dalam penelitian ini, ditunjukkan oleh gambar 3.



Gambar 3. Flowchart penelitian

## 4. HASIL DAN PEMBAHASAN

Hasil dari pengolahan citra untuk mendapatkan jumlah koin yang terhitung dalam gambar (citra), memakai aplikasi visual studio code, menggunakan OpenCV dan bahasa pemrograman python.

### 4.1 Gambar Koin yang diuji

Gambar yang dipergunakan berjumlah 10 buah gambar uang koin Rupiah. Koin ini berbahan aluminium, logam dan perak. Nominalnya bervariasi yaitu : Rp. 100,. Rp. 200,. Rp. 500,. dan Rp. 1000. Gambar tersebut diambil menggunakan kamera HP.

### 4.2 Grayscale

Pada tahap ini gambar (citra) yang telah diambil diubah ke dalam citra warna *grayscale* (keabuan) menggunakan *threshold*.



Gambar 4. Citra Awal



Gambar 5. Citra grayscale

### 4.3 Gaussian Filter

Pada tahap ini setelah citra di-*threshold* menjadi *grayscale*, lalu dilakukan fungsi filter gaussian agar *noise-noise* pada gambar dari uang koin dapat hilang, berikut sintaks nya dalam python.

```

koin.py
koin.py
1 import cv2
2 import numpy as np
3
4 img = cv2.imread('koinF.jpg', 0)
5 blur = cv2.GaussianBlur(img, (11,11), 0)
6
7
8 cv2.waitKey(0)
9 cv2.destroyAllWindows()
  
```

Gambar 6. List Program Gaussian Filtering

### 4.4 Deteksi Tepi Canny

Setelah citra diberikan filter gaussian lalu citra tersebut dilakukan deteksi tepi menggunakan metode Canny untuk menghitung dan menyeleksi tepi - tepi dari uang koin yang dimasukkan dalam tes pengujian, mainfolnya 40 dan maxfolnya 150. berikut penambahan sintaks dalam pyton.

```

koinF.py
1 import cv2
2 import numpy as np
3
4 img = cv2.imread('koinF.jpg', 0)
5 blur = cv2.GaussianBlur(img, (11,11), 0)
6 edges = cv2.Canny(blur, 40, 150)
7
    
```

Gambar 7. List Program Canny

4.5 Dilated

Proses *dilated* ini dilakukan dalam penelitian ini untuk mempertegas lagi tepi-tepi yang telah berhasil diseleksi menggunakan metode Canny agar gambar hasilnya lebih jelas dan tegas.

4.6 Gambar Hasil dan Perhitungan Koin

Pada tahap ini setelah dilakukan proses dilated pada gambar (citra) uang koin tersebut dilakukan *counter* (perhitungan) jumlah koin menggunakan sintaks berikut :

```

koinF.py
1 import cv2
2 import numpy as np
3
4 img = cv2.imread('koinF.jpg', 0)
5 blur = cv2.GaussianBlur(img, (11,11), 0)
6 edges = cv2.Canny(blur, 40, 150)
7 dilated = cv2.dilate(edges, (1,1), iterations = 2)
8
9 (cnt, _) = cv2.findContours(dilated.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
10 print("Terdapat {} koin dalam gambar".format(len(cnt)))
11
12 #taks = "Terdapat {} koin".format(len(cnt))
13 #font = cv2.FONT_HERSHEY_PLAIN
14 #cv2.putText(dilated, taks, (35, 30), font, 1, (255, 0, 0), 1, cv2.LINE_AA)
15
16 cv2.imshow('Koin', dilated)
17
18 cv2.waitKey(0)
19 cv2.destroyAllWindows()
    
```

Gambar 8. Menghitung Koin

Hasil uji yang dilakukan terhadap gambar koinA.jpg yang berisi 1 koin didalamnya menunjukkan hasil bahwa dari gambar tersebut tepi yang terseleksi dapat terlihat dan terpolo dengan bagus dan koin dapat dihitung dengan benar yaitu 1 koin. Hasil uji coba ditunjukkan oleh gambar 9.



Gambar 9. koinA berisi 1 koin

Hasil uji yang dilakukan terhadap gambar koinF.jpg yang berisi 5 koin didalamnya menunjukkan hasil bahwa dari gambar tersebut tepi yang terseleksi dapat terlihat dan terpolo dengan bagus dan koin yang dapat dihitung dengan benar yaitu 5 koin. Hasil uji coba tertera pada gambar 10.



Gambar 10. koinF berisi 5 koin

Hasil uji yang dilakukan terhadap gambar koinH.jpg yang berisi 5 koin didalamnya menunjukkan hasil bahwa ada 3 koin yang terlihat jelas pola deteksi tepinya, sedangkan 2 koin lagi tidak dapat dideteksi tepi koin-koin tersebut, hal tersebut diakibatkan ada sebagian area gambar agak gelap, sehingga sistem tidak dapat terbaca dengan baik, Hasil uji coba terdapat pada gambar 11.

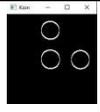
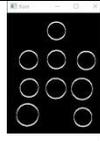


Gambar 11. koinH berisi 3 koin

Berikut adalah hasil menghitung koin dan hasil tepi gambar dari 10 gambar uji yang telah diuji, terdapat pada tabel 2.

Tabel 2. Hasil deteksi tepi dan Perhitungan koin

N o	Nama File	Ukuran	Gambar Awal	Gambar Hasil	Jumlah Koin terhitung
1	koinA	7,86 kb			Terhitung : 1 Koin
2	koinB	6,40 kb			Terhitung : 1 Koin
3	koinC	7,55 kb			Terhitung : 5 Koin
4	koinD	8,34 kb			Terhitung : 3 Koin
5	koinE	6,60 kb			Terhitung : 4 Koin
6	koinF	7,47 kb			Terhitung : 5 Koin
7	koinG	5,83 kb			Terhitung : 4 Koin

8	koinH	6,53 kb			Terhitung : 3 Koin
9	koinI	10,8 kb			Terhitung : 9 Koin
10	koinJ	8,12 kb			Terhitung : 2 Koin

## 5. KESIMPULAN

- a. Metode Canny yang sudah diterapkan untuk menghitung jumlah uang koin yang ada dalam sebuah gambar sudah dapat berjalan dengan baik dan koin terhitung dengan jumlah yang benar, dalam penelitian ini sistem sudah bisa membandingkan koin yang berwarna (bahan tembaga) dan koin yang berwarna putih (bahan aluminium) dengan keakurasian hasil data mencapai 91%.
- b. Dalam penelitian ini kualitas gambar (citra), area gambar yang gelap dan intensitas cahaya dari gambar untuk data ujicoba sangat mempengaruhi hasil dari deteksi tepi dan hasil perhitungan koin.

## UCAPAN TERIMA KASIH

Alhamdulillah, puji syukur ke hadirat Allah SWT, Sholawat beserta Salam dihaturkan kepada Nabi Muhammad SAW, serta ucapan terimakasih untuk keluarga, dosen pembimbing, sahabat, rekan-rekan yang telah memberi semangat, dan motivasi serta bantuan sehingga penulis bisa menyelesaikan artikel ini.

## DAFTAR PUSTAKA

- [1] R. Perangin-angin and E. J. G. Harianja, "Comparison detection edge lines algoritma canny dan sobel," *J. TIMES*, vol. VIII, no. 2, pp. 35–42, 2019.
- [2] D. Z. E. Prastya, D. P. Pamungkas, and R. K. Niswatin, "Implementasi Metode Gaussian Filter Dan Median Filter Untuk Penghalusan Gambar," *Semin. Nas. Inov. Teknol.*, pp. 178–187, 2022.

- [3] Safriadi, A. Essra, and Rahmadani, "Analisis Deteksi Tepi Canny Pada Citra Dengan Gaussian Filtering Dan Bilateral Filtering," *J. ISD*, vol. 2, no. 1, pp. 34–39, 2017.
- [4] K. P. Nurdin, "Klasifikasi Kecantikan Wanita Aceh Pada Citra Menggunakan Metode Adaptive Resonance Theory (ART1)," *TECHSI - J. Penelit. Tek. Inform.*, vol. 8, no. 1, pp. 139–147, 2019, [Online]. Available: <https://ojs.unimal.ac.id/index.php/techsi/article/view/121>
- [5] N. Nurdin, D. Hamdhana, and M. J. Setiawan, "Sistem Pendeteksi Pola Lafadz Allah Dan Muhammad Pada Citra Al-Qur'an Menggunakan Metode Peirce," *TECHSI - J. Tek. Inform.*, vol. 9, no. 2, p. 78, 2017, doi: 10.29103/techsi.v9i2.215.
- [6] S. Nesi, "Pengenalan Pola Untuk Deteksi Uang Koin," *Semin. Nas. Teknol. Inf. Komun. dan Ind. III*, pp. 18–24, 2011.
- [7] M. Musrini, M. Premitasari, R. Rizqika, P. S. Informatika, J. T. Informatika, and F. T. Industri, "Implementasi Metode Edge Detection Dan Learning Vector Quantization Studi Kasus Uang Kertas Rupiah," pp. 128–138, 2017.
- [8] B. Yuwono, "Image Smoothing Menggunakan Mean Filtering, Median Filtering, Modus Filtering Dan Gaussian Filtering," *Telematika*, vol. 7, no. 1, 2010, doi: 10.31315/telematika.v7i1.416.
- [9] N. A. Ramdhan and D. A. Nufriana, "Rancang Bangun Dan Implementasi Sistem Informasi Skripsi Oline Berbasis WEB," *J. Ilm. Intech Inf. Technol. J. UMUS*, vol. 1, no. 02, pp. 1–12, 2019, doi: 10.46772/intech.v1i02.75.
- [10] T. C. A.-S. Zulkhaidi, E. Maria, and Y. Yulianto, "Pengenalan Pola Bentuk Wajah dengan OpenCV," *J. Rekayasa Teknol. Inf.*, vol. 3, no. 2, p. 181, 2019, doi: 10.30872/jurti.v3i2.4033.