

## Implementasi OpenCV dengan Metode *Image Thresholding* pada Gambar

Ade Maulana<sup>1\*</sup>, Firda Auliattunnajah<sup>2</sup>, Nazdan Rosidin<sup>3</sup>, Muhamad Ramadien Rizki Darmawan<sup>4</sup>, Perani Rosyani<sup>5</sup>

<sup>1-5</sup>Fakultas Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Jl. Raya Puspipetek No. 46, Kel. Buaran, Kec. Serpong, Kota Tangerang Selatan. Banten 15310, Indonesia

Email: <sup>1</sup>[ademaaulana0127@email.com](mailto:ademaaulana0127@email.com), <sup>2</sup>[frda.auliattun.n@gmail.com](mailto:frda.auliattun.n@gmail.com), <sup>3</sup>[nzdrosidin28@gmail.com](mailto:nzdrosidin28@gmail.com), <sup>4</sup>[muhammad.ramadien@gmail.com](mailto:muhammad.ramadien@gmail.com), <sup>5</sup>[dosen00837@gmail.com](mailto:dosen00837@gmail.com)

(\* : coressponding author)

**Abstrak**– Abstrak Pengolahan citra adalah aspek krusial dalam analisis visual di berbagai disiplin seperti kedokteran, industri, dan pengenalan pola. Metode Image Thresholding merupakan salah satu teknik efektif untuk segmentasi objek dalam gambar berdasarkan tingkat kecerahan piksel. Studi ini mengimplementasikan metode Image Thresholding menggunakan OpenCV untuk menerapkan berbagai pendekatan thresholding pada gambar digital. Hasil implementasi menunjukkan kemampuan teknik ini dalam memisahkan objek dari latar belakang dengan baik, meskipun memerlukan penyesuaian parameter yang tepat untuk hasil optimal.

**Kata Kunci:** OpenCV, Image Thresholding, Pengolahan Citra, Segmentasi Gambar, Deteksi Objek.

**Abstract**– *Abstract Image processing is a crucial aspect of visual analysis in various fields such as medicine, industry, and pattern recognition. The Image Thresholding method is an effective technique for segmenting objects in an image based on the brightness level of pixels. This study implements the Image Thresholding method using OpenCV to apply various thresholding approaches to digital images.*

**Keywords:** *OpenCV, Image Thresholding, Image Processing, Image Segmentation, Object Detection*

### 1. PENDAHULUAN

Pengolahan citra digital adalah proses krusial dalam analisis gambar yang bertujuan untuk ekstraksi informasi yang bermanfaat dari data visual. Metode Image Thresholding digunakan untuk memisahkan objek dari latar belakang berdasarkan ambang batas tertentu dalam intensitas piksel. OpenCV menyediakan berbagai metode thresholding yang dapat diterapkan dengan mudah untuk berbagai aplikasi pengolahan citra. Artikel ini bertujuan untuk mengeksplorasi dan menganalisis implementasi metode Image Thresholding menggunakan OpenCV pada gambar digital. (Hoffman, n.d.)

OpenCV, singkatan dari Open Source Computer Vision Library, adalah perpustakaan sumber terbuka yang sangat populer untuk pemrosesan gambar dan penglihatan komputer. Dibuat pertama kali oleh Intel pada tahun 1999, OpenCV sangat mudah digunakan dan memiliki banyak fitur, menjadikannya salah satu alat penting dalam pengembangan aplikasi penglihatan komputer.

Beberapa fitur utama OpenCV adalah sebagai berikut:

- Pemrosesan Gambar: Memungkinkan untuk melakukan manipulasi gambar seperti pemotongan, pemosisian, dan transformasi.
- Pengenalan Pola dan Deteksi Objek: Mendukung pengenalan wajah, deteksi objek, dan pengenalan pola.
- Pengolahan Citra: Menyediakan algoritma untuk filterisasi, segmentasi, dan ekstraksi fitur citra.
- Kalibrasi Kamera: Memungkinkan kalibrasi kamera dan geometri stereo.
- Pemrosesan Video: Dapat memancarkan gambar dengan lebih baik.

OpenCV tersedia dalam berbagai bahasa pemrograman, seperti C++, Python, Java, dan MATLAB, yang membuatnya lebih mudah untuk diintegrasikan ke berbagai platform dan digunakan dalam berbagai aplikasi, seperti visi mesin, pengolahan gambar medis, kontrol industri, dll. (Widyadhana, 2021).

Teknik pemrosesan citra yang dikenal sebagai thresholding digunakan untuk membagi gambar menjadi bagian yang lebih mudah diamati. Metode ini mengubah gambar grayscale

bergradasi menjadi gambar biner hitam-putih. Ini memisahkan objek dari latar belakang berdasarkan intensitas piksel.

### Jenis-Jenis Metode Thresholding

#### 1. *Thresholding Global (Thresholding Sederhana):*

- Menggunakan satu nilai threshold untuk seluruh gambar.
- Setiap piksel di gambar dibandingkan dengan nilai threshold:
  - Piksel di atas threshold diberi nilai satu (putih).
  - Piksel di bawah threshold diberi nilai nol (hitam).
- Contoh: `cv2.threshold` di OpenCV.

```
python Copy code  
  
import cv2  
import numpy as np  
  
img = cv2.imread('image.jpg', 0) # Membaca gambar dalam mode grayscale  
ret, binary_img = cv2.threshold(img, 127, 255, cv2.THRESH_BINARY)
```

**Gambar 1.** Contoh *Thresholding Global*

#### 2. *Thresholding Adaptif:*

- Menggunakan nilai threshold yang berbeda untuk wilayah berbeda dari gambar.
- Cocok untuk gambar dengan pencahayaan tidak merata.
- Menggunakan metode seperti Mean atau Gaussian untuk menentukan threshold lokal.
- Contoh: `cv2.adaptiveThreshold` di OpenCV.

```
python Copy code  
  
adaptive_thresh = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,  
                                       cv2.THRESH_BINARY, 11, 2)
```

**Gambar 2.** Contoh *Thresholding Adaptif*

#### 3. *Otsu's Thresholding:*

- Mengotomatisasi pemilihan nilai threshold berdasarkan histogram intensitas gambar.
- Meminimalkan varians dalam dua kelas piksel (latar belakang dan objek).
- Contoh: `cv2.threshold` dengan parameter `cv2.THRESH_BINARY + cv2.THRESH_OTSU`.

```
python Copy code  
  
ret, otsu_thresh = cv2.threshold(img, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
```

**Gambar 3.** Contoh *Otsu's Thresholding*

### Aplikasi Thresholding

- a. Deteksi Tepi: Membantu mendeteksi tepi objek dalam gambar.
- b. Pengenalan Karakter: OCR (Pengenalan Karakter Optical) membedakan teks dari latar belakang.
- c. Segmentasi Objek: Memisahkan objek tertentu dari latar belakang untuk analisis tambahan.

- d. Pengolahan Citra Medis: Memperjelas batas antara jaringan atau struktur dalam gambar medis seperti sinar-X atau MRI.

Thresholding dapat menyederhanakan gambar yang kompleks menjadi format yang lebih mudah dianalisis, yang memungkinkan identifikasi dan pengolahan objek dengan lebih efektif.(Bernadetha, 2022).

## 2. METODE PENELITIAN

### 2.1 Persyaratan dan Persiapan

- a. **Prasyarat:** Instalasi OpenCV
- b. **Persiapan Data:** Data yang digunakan terdiri dari berbagai jenis gambar digital dengan variasi tingkat kecerahan, kontras, dan kompleksitas objek.

### 2.2 Implementasi Metode *Image Thresholding*

Implementasi metode Image Thresholding dilakukan menggunakan Python dan pustaka OpenCV. Langkah-langkah implementasi meliputi:

- a. **Pembacaan Gambar:** Mengimpor gambar digital ke dalam program menggunakan OpenCV.
- b. **Konversi ke Grayscale:** Mengubah gambar ke mode grayscale untuk memfasilitasi proses thresholding.
- c. **Penerapan Metode Thresholding:** Menerapkan beberapa metode thresholding seperti Binary, Adaptive, dan Otsu untuk memisahkan objek dari latar belakang.
- d. **Evaluasi Hasil:** Mengukur kejelasan dan akurasi hasil thresholding, serta mengevaluasi performa masing-masing metode.

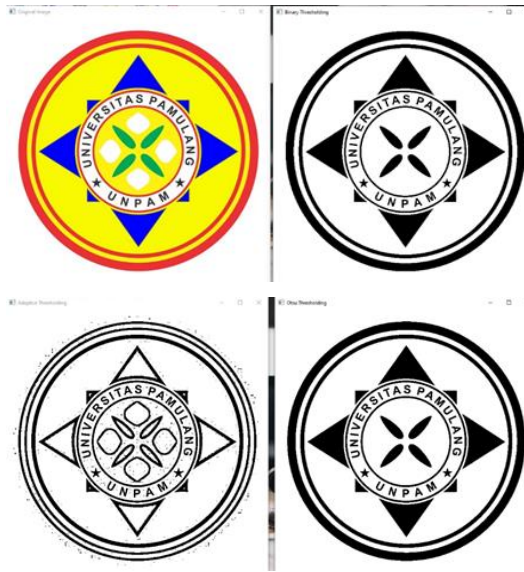
### 2.3 Implementasi dengan OpenCV

Berikut adalah implementasi menggunakan OpenCV dan Python:

```

import cv2.py
C:\Users\acer> cd import cv2.py > ...
1 import cv2
2 import numpy as np
3
4 # Membaca gambar
5 image = cv2.imread('gambar.jpg', cv2.IMREAD_COLOR)
6
7 # Konversi ke Grayscale
8 gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
9
10 # Metode Thresholding Binary
11 ret, binary_threshold = cv2.threshold(gray_image, 127, 255, cv2.THRESH_BINARY)
12
13 # Metode Thresholding Adaptive
14 adaptive_threshold = cv2.adaptiveThreshold(gray_image, 255, cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY, 11, 2)
15
16 # Metode Thresholding Otsu
17 ret2, otsu_threshold = cv2.threshold(gray_image, 0, 255, cv2.THRESH_BINARY+cv2.THRESH_OTSU)
18
19 # Menampilkan hasil
20 cv2.imshow('Original Image', image)
21 cv2.imshow('Binary Thresholding', binary_threshold)
22 cv2.imshow('Adaptive Thresholding', adaptive_threshold)
23 cv2.imshow('Otsu Thresholding', otsu_threshold)
24
25 cv2.waitKey(0)
26 cv2.destroyAllWindows()
    
```

**Gambar 4.** Implementasi



**Gambar 5.** Hasil Output Figure 1

Hasil implementasi menunjukkan bahwa metode Binary Thresholding sederhana dan efektif untuk gambar dengan kontras yang jelas. Adaptive Thresholding cocok untuk gambar dengan pencahayaan tidak merata tetapi cenderung menghasilkan noise, sementara Otsu's Thresholding memberikan segmentasi yang optimal secara otomatis dengan hasil yang bersih dan detail yang baik, menjadikannya pilihan terbaik untuk berbagai jenis gambar.

- **Gambar Asli (Original Image):** Gambar ini merupakan logo berwarna dari Universitas Pamulang yang akan diolah menggunakan berbagai metode thresholding.
- **Binary Thresholding :** Pikel diubah menjadi hitam atau putih berdasarkan ambang batas 127. Objek dipisahkan dengan jelas dari latar belakang, namun beberapa detail kecil mungkin hilang.
- **Adaptive Thresholding:** Ambang batas dihitung untuk area kecil gambar, menyesuaikan secara dinamis berdasarkan variasi pencahayaan lokal. Metode ini lebih sensitif dan dapat menghasilkan noise.
- **Otsu's Thresholding:** Menentukan nilai ambang batas optimal secara otomatis. Efektif dalam memisahkan objek dari latar belakang dengan hasil yang bersih dan detail yang baik.

### **3. ANALISA DAN PEMBAHASAN**

#### **3.1 Kelebihan Metode Thresholding**

- a. Sederhana dan Cepat: Metode ini hanya memerlukan perbandingan nilai intensitas piksel dengan nilai threshold, sehingga mudah digunakan dan cepat dilakukan.
- b. Efisiensi Komputasi: Cocok untuk aplikasi real-time karena menggunakan lebih sedikit sumber daya komputasi daripada metode segmentasi yang lebih kompleks.
- c. Penggunaan Luas: Banyak aplikasi, seperti pemrosesan gambar medis, pengenalan karakter optik (OCR), dan sistem pengawasan, dapat menggunakannya.
- d. Hasil Jelas: Menghasilkan gambar biner yang memudahkan analisis tambahan, seperti penghitungan objek dan deteksi tepi. (Houssein et al., 2023)

#### **3.2 Kekurangan Metode Thresholding**

- a. Kekurangan Metode Thresholding Sensitif terhadap Pencahayaan: Thresholding global sangat sensitif terhadap perubahan pencahayaan dan kontras gambar. Pencahayaan yang tidak merata dapat menyebabkan hasil yang tidak akurat.

- b. Keterbatasan untuk Gambar Kompleks: Ini tidak akan bekerja dengan baik pada gambar dengan latar belakang yang kompleks atau ketika intensitas objek dan latar belakang hampir sama.
- c. Pilihan Threshold: Seringkali, memilih nilai ambang yang tepat memerlukan pengalaman atau pengetahuan sebelumnya tentang gambar, terutama untuk metode ambang global.
- d. Tidak Menangani Noise dengan Baik: Gambar yang berisik atau berkualitas rendah dapat menghasilkan thresholding yang buruk karena suara dapat mengganggu perbedaan antara objek dan latar belakang.
- e. Keterbatasan Segmentasi Detail: Karena hanya bergantung pada nilai intensitas piksel, tidak dapat menangani detail halus dalam gambar. (Heryanto et al., 2020)

### 3.3 Solusi untuk Mengatasi Keterbatasan

- a. Thresholding Adaptif: Tangani variasi pencahayaan lokal dalam gambar dengan metode thresholding adaptif.
- b. Pra-Pemrosesan Gambar: Sebelum menerapkan thresholding, teknik pra-pemrosesan seperti filtering dan smoothing digunakan untuk mengurangi noise.
- c. Metode Gabungan: Untuk mendapatkan hasil yang lebih baik pada gambar yang kompleks, gunakan thresholding bersama dengan teknik segmentasi lain seperti clustering atau deteksi tepi.
- d. Penyesuaian Dinamis: Untuk menentukan nilai penyesuaian yang paling cocok berdasarkan distribusi intensitas gambar, gunakan metode otomatis seperti penyesuaian Otsu. (Rofi'i & Ningtias, 2022)

### 3.4 Kelebihan *Image Processing*

- a. **Otomatisasi dan Kecepatan:** Pemrosesan gambar dapat dilakukan secara otomatis dan cepat, mengurangi waktu dan usaha manusia dalam analisis data visual.
- b. **Peningkatan Kualitas Gambar:** Dapat digunakan untuk meningkatkan kualitas gambar, seperti meningkatkan kontras, mengurangi noise, dan memperjelas detail gambar.
- c. **Analisis Akurat:** Memberikan hasil analisis yang lebih akurat dan konsisten dibandingkan analisis manual, terutama dalam aplikasi medis, pengawasan, dan kontrol kualitas.
- d. **Pemrosesan Real-time:** Banyak aplikasi pemrosesan gambar dapat dilakukan dalam waktu nyata, seperti dalam pengenalan wajah atau sistem pengawasan video.
- e. **Ekstraksi Fitur:** Memungkinkan ekstraksi fitur penting dari gambar untuk aplikasi lebih lanjut seperti pengenalan objek, deteksi tepi, dan segmentasi gambar.
- f. **Fleksibilitas:** Dapat diterapkan dalam berbagai bidang seperti kedokteran, industri, pertanian, pengawasan, dan hiburan. (Irianto, 2016)

### 3.5 Kekurangan *Image Processing*

- a. **Kebutuhan Komputasi Tinggi:** Beberapa teknik pemrosesan gambar, terutama yang kompleks seperti pemrosesan video atau analisis gambar resolusi tinggi, memerlukan sumber daya komputasi yang besar.
- b. **Ketergantungan pada Kualitas Gambar:** Kualitas hasil pemrosesan sangat tergantung pada kualitas gambar input. Gambar yang buram, berisik, atau dengan pencahayaan buruk dapat menghasilkan analisis yang kurang akurat.
- c. **Pemrosesan Gambar Berisik:** Gambar yang memiliki noise tinggi atau artefak lain bisa sulit diproses dengan baik, meskipun ada teknik untuk mengurangi noise.
- d. **Keterbatasan Algoritma:** Tidak semua algoritma pemrosesan gambar dapat bekerja dengan baik di semua kondisi. Misalnya, metode thresholding sederhana mungkin tidak efektif pada gambar dengan pencahayaan tidak merata.
- e. **Biaya Implementasi:** Pengembangan sistem pemrosesan gambar yang kompleks bisa memerlukan biaya yang tinggi, baik dalam hal perangkat keras maupun perangkat lunak.



- f. **Kesalahan Interpretasi:** Algoritma pemrosesan gambar mungkin salah menafsirkan objek atau fitur dalam gambar, terutama dalam situasi yang tidak terduga atau kompleks. (Suhairi, 2018)

#### 4. KESIMPULAN

Studi ini menunjukkan bahwa implementasi metode Image Thresholding menggunakan OpenCV efektif dalam memisahkan objek dari latar belakang dalam gambar digital. Dengan memilih metode thresholding yang tepat dan mengatur parameter dengan baik, teknik ini dapat digunakan untuk berbagai aplikasi pengolahan citra dengan hasil yang memuaskan. Meskipun masih ada beberapa tantangan yang perlu diatasi, penggunaan OpenCV untuk thresholding memberikan kontribusi signifikan dalam analisis dan pemrosesan gambar di berbagai bidang.

#### REFERENCES

- Bernadetha, S. (2022). *Perhitungan Objek Pada Citra Menggunakan Teknik Thresholding*. [https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2022-2023/Makalah/Makalah/Makalah-IF4073-Citra-Sem1-2022\(16\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2022-2023/Makalah/Makalah/Makalah-IF4073-Citra-Sem1-2022(16).pdf)
- Heryanto, I. W. A., Artama, Kurniawan, M. W. S., & Gunadi, G. A. (2020). Segmentasi Warna dengan Metode Thresholding. *Wahana Matematika Dan Sains*, 14(1), 54–64.
- Hoffman, D. W. (n.d.). *Dasar Pengolahan Citra Digital*.
- Houssein, E. H., Mohamed, G. M., Ibrahim, I. A., & Wazery, Y. M. (2023). An efficient multilevel image thresholding method based on improved heap-based optimizer. In *Scientific Reports* (Vol. 13, Issue 1). Nature Publishing Group UK. <https://doi.org/10.1038/s41598-023-36066-8>
- Irianto, S. Y. (2016). Analisa Citra Digital dan Content Based Image Retrieval. *Perpustakaan Nasional RI, December*.
- Rofi'i, M., & Ningtias, D. R. (2022). Local Adaptive Thresholding Menggunakan Metode Sauvola sebagai Tahapan Pra Pengolahan pada Data Citra Isyarat ECG. *Jurnal Teori Dan Aplikasi Fisika*, 10(1), 103. <https://doi.org/10.23960/jtaf.v10i1.2904>
- Suhairi, M. (2018). *Pengurangan Kompleksitas Komputasi Pada Multiview Hvc Berbasis Perangkat Fpga*.
- Widyadhana, I. H. (2021). *Deteksi Pengambilan Barang Dari Rak Dengan Pengolahan Citra*.