



# Studi Perbandingan Teknik *Thresholding* Untuk Binarisasi Tanda Tangan Menggunakan OpenCV2: Metode Sederhana, Adaptif, Dan Otsu

Mohammad Ramdani<sup>1\*</sup>, Muhammad Fairuz Dzulfikar<sup>2</sup>, Zidan Absar Abdallah<sup>3</sup>, Zulfikar Reza Pahlevi<sup>4</sup>, Perani Rosyani<sup>5</sup>

<sup>1,2,3,4,5</sup>Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia

Email: <sup>1\*</sup>[m.ramdanisupriyadi@gmail.com](mailto:m.ramdanisupriyadi@gmail.com), <sup>2</sup>[fairuzdzulfikar@gmail.com](mailto:fairuzdzulfikar@gmail.com), <sup>3</sup>[zidanabsar03@gmail.com](mailto:zidanabsar03@gmail.com), <sup>4</sup>[fikareza183@gmail.com](mailto:fikareza183@gmail.com), <sup>5</sup>[dosen00837@unpam.ac.id](mailto:dosen00837@unpam.ac.id)

(\* : coresponding author)

**Abstrak** – Proses binarisasi tanda tangan adalah langkah penting dalam pengenalan dan verifikasi tanda tangan otomatis. Penelitian ini mengembangkan, mengimplementasikan, dan mengevaluasi tiga teknik *thresholding*—sederhana, adaptif, dan Otsu untuk binarisasi tanda tangan menggunakan pustaka OpenCV2. *Thresholding* sederhana menggunakan satu ambang batas untuk seluruh gambar, *thresholding* adaptif menyesuaikan ambang batas berdasarkan bagian gambar, dan metode Otsu menentukan ambang batas otomatis menggunakan analisis histogram. Penelitian ini membandingkan efektivitas dan keandalan ketiga teknik tersebut dalam berbagai kondisi pencahayaan. Hasil menunjukkan bahwa metode *thresholding* sederhana kurang efektif pada kondisi pencahayaan gelap, sementara *thresholding* adaptif menunjukkan peningkatan performa dalam menangani variasi pencahayaan. Metode Otsu memberikan hasil terbaik dengan kemampuan menentukan ambang batas optimal secara otomatis dan menghasilkan gambar biner yang akurat dan jelas.

**Kata Kunci:** Binarisasi Tanda Tangan, *Thresholding*, OpenCV2, Pengenalan Tanda Tangan.

**Abstract** – The process of signature binarization is a crucial step in automatic signature recognition and verification. This research develops, implements, and evaluates three *thresholding* techniques—simple, adaptive, and Otsu—for signature binarization using the OpenCV2 library. Simple *thresholding* uses a single threshold for the entire image; adaptive *thresholding* adjusts the threshold based on different parts of the image; and the Otsu method determines the threshold automatically through histogram analysis. This study compares the effectiveness and reliability of these three techniques under various lighting conditions. The results indicate that simple *thresholding* is less effective in dark lighting conditions, while adaptive *thresholding* shows improved performance in handling lighting variations. The Otsu method provides the best results with its ability to automatically determine the optimal threshold, producing accurate and clear binary images.

**Keywords:** Signature Binarization, *Thresholding*, OpenCV2, Signature Recognition.

## 1. PENDAHULUAN

Pengenalan tanda tangan penting dalam identitas dan validasi dokumen. Di era digital, otomatisasi dan verifikasi tanda tangan sangat dibutuhkan. Langkah penting dalam proses ini adalah binarisasi tanda tangan, yaitu mengubah gambar tanda tangan dari grayscale menjadi biner untuk membedakan tanda tangan dari latar belakang

*Thresholding* adalah metode umum untuk binarisasi, yang memisahkan piksel gambar menjadi hitam (tanda tangan) dan putih (latar belakang) berdasarkan ambang batas. Ada tiga teknik *thresholding* yang sering digunakan: sederhana, adaptif, dan Otsu. Metode sederhana menggunakan satu ambang batas untuk seluruh gambar, metode adaptif menyesuaikan ambang berdasarkan bagian gambar, dan metode Otsu menentukan ambang otomatis menggunakan analisis histogram.

Penelitian ini membandingkan ketiga teknik tersebut menggunakan pustaka OpenCV2. Tujuannya adalah menilai keefektifan dan keandalan tiap teknik dalam binarisasi tanda tangan, serta membantu memilih teknik terbaik untuk aplikasi praktis. Hasil penelitian diharapkan dapat mendukung pengembangan sistem pengenalan tanda tangan yang lebih canggih dan andal.

## 2. METODOLOGI

Metode yang digunakan untuk mengimplementasikan ketiga teknik thresholding menggunakan pustaka OpenCV2, yaitu:

- Thresholding Sederhana: Penggunaan fungsi `cv2.threshold` untuk menerapkan thresholding sederhana pada gambar tanda tangan.
- Thresholding Adaptif: Penerapan thresholding adaptif dengan menggunakan fungsi `cv2.adaptiveThreshold` dengan berbagai parameter yang disesuaikan.
- Binarisasi Otsu: Penggunaan metode Otsu untuk menentukan ambang batas secara otomatis dari histogram gambar menggunakan `cv2.threshold` dengan opsi `cv2.THRESH_OTSU`.

### 2.1 Persiapan Data



**Gambar 1.** Gambar Tanda Tangan Dalam Intensitas Pencahayaan Yang Minim



**Gambar 2.** Gambar Tanda Tangan Dalam Intensitas Pencahayaan Yang Memadai

### 2.2 Implementasi Teknik Thresholding

Setiap teknik thresholding diimplementasikan menggunakan pustaka OpenCV2 (versi 4.9.0) dalam bahasa pemrograman Python. Berikut adalah detail implementasi masing-masing teknik:

#### a. Thresholding Sederhana

Metode sederhana menggunakan nilai ambang global untuk memisahkan piksel objek dan latar belakang dalam citra tanda tangan. Prosesnya melibatkan penggunaan fungsi `cv.threshold` dengan parameter yang ditentukan secara manual.

#### b. Thresholding Adaptif

Teknik ini mengadaptasi ambang batas berdasarkan lingkungan lokal gambar, memungkinkan penyesuaian yang lebih baik terhadap variasi pencahayaan dalam gambar tanda tangan. Implementasinya menggunakan fungsi `cv.adaptiveThreshold` dengan metode yang dipilih (misalnya, `cv.ADAPTIVE_THRESH_MEAN_C`) dan parameter yang disesuaikan seperti ukuran blok dan konstanta C.

### c. Binarisasi Otsu

Metode Otsu menggunakan analisis histogram untuk menentukan ambang batas secara otomatis, maksimalkan pemisahan antara piksel objek dan latar belakang. Penerapannya menggunakan fungsi `cv.threshold` dengan opsi `cv2.THRESH_OTSU`.

## 2.3 Proses Eksperimen

Setiap teknik thresholding diuji pada dataset yang disiapkan. Proses eksperimen melibatkan langkah-langkah berikut:

- Memuat gambar tanda tangan dari dataset.
- Menerapkan masing-masing teknik thresholding (sederhana, adaptif, dan Otsu) pada gambar.
- Membandingkan hasil biner yang dihasilkan dari setiap teknik, baik secara visual maupun dengan menggunakan metrik evaluasi yang sesuai.

## 3. IMPLEMENTASI DAN HASIL

```
1. Thresholding Sederhana

import cv2 as cv2
import numpy as np
from matplotlib import pyplot as plt

# Memuat gambar dalam grayscale
img = cv2.imread('path_to_image.jpg', cv2.IMREAD_GRAYSCALE)
assert img is not None, "file could not be read, check with os.path.exists()"

# aplikasi simple thresholding
ret, thresh1 = cv2.threshold(img, 127, 255, cv2.THRESH_BINARY)
ret, thresh2 = cv2.threshold(img, 127, 255, cv2.THRESH_BINARY_INV)
ret, thresh3 = cv2.threshold(img, 127, 255, cv2.THRESH_TRUNC)
ret, thresh4 = cv2.threshold(img, 127, 255, cv2.THRESH_TOZERO)
ret, thresh5 = cv2.threshold(img, 127, 255, cv2.THRESH_TOZERO_INV)

# Menampilkan hasil
titles = ['GAMBAR ASLI', 'BINARY', 'BINARY_INV', 'TRUNC', 'TOZERO', 'TOZERO_INV']
images = [img, thresh1, thresh2, thresh3, thresh4, thresh5]

for i in range(6):
    plt.subplot(2, 3, i + 1), plt.imshow(images[i], 'gray', vmin=0, vmax=255)
    plt.title(titles[i])
    plt.xticks([], plt.yticks([]))
plt.show()
```

**Gambar 3.** Thresholding Sederhana

```
2. Thresholding Adaptif

import cv2 as cv2
import numpy as np
from matplotlib import pyplot as plt

# Memuat gambar dalam grayscale
img = cv2.imread('path_to_image.jpg', cv2.IMREAD_GRAYSCALE)
assert img is not None, "file could not be read, check with os.path.exists()"

# aplikasi median blur pada gambar
img = cv2.medianBlur(img, 5)

# aplikasi global thresholding
ret, th1 = cv2.threshold(img, 127, 255, cv2.THRESH_BINARY)

# aplikasi adaptive thresholding
th2 = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY, 11, 2)
th3 = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 11, 2)

# menampilkan hasil
titles = ['Gambar Asli', 'Global Thresholding (v = 127)', 'Adaptive Mean Thresholding', 'Adaptive Gaussian Thresholding']
images = [img, th1, th2, th3]

for i in range(4):
    plt.subplot(2, 2, i + 1), plt.imshow(images[i], 'gray')
    plt.title(titles[i])
    plt.xticks([], plt.yticks([]))
plt.show()
```

**Gambar 4.** Thresholding Adaptif

```
3. Binarisasi Otsu

import cv2 as cv2
import numpy as np
from matplotlib import pyplot as plt

# Memuat gambar dalam grayscale
img = cv2.imread('path_to_image.jpg', cv2.IMREAD_GRAYSCALE)
assert img is not None, "file could not be read, check with
os.path.exists()"

# Global thresholding
ret1, th1 = cv2.threshold(img, 127, 255, cv2.THRESH_BINARY)

# Otsu thresholding
ret2, th2 = cv2.threshold(img, 0, 255, cv2.THRESH_BINARY +
cv2.THRESH_OTSU)

# Otsu thresholding dengan filter Gaussian
blur = cv2.GaussianBlur(img, (5, 5), 0)
ret3, th3 = cv2.threshold(blur, 0, 255, cv2.THRESH_BINARY +
cv2.THRESH_OTSU)

# menampilkan hasil
images = [img, 0, th1,
          img, 0, th2,
          blur, 0, th3]
titles = ['Gambar Asli Dengan Nois', 'Histogram', 'Global Thresholding
(v=127)',
          'Gambar Asli Dengan Nois', 'Histogram', "Otsu's Thresholding",
          'Gaussian filtered Image', 'Histogram', "Otsu's Thresholding"]

for i in range(3):
    plt.subplot(3, 3, i * 3 + 1), plt.imshow(images[i * 3], 'gray')
    plt.title(titles[i * 3]), plt.xticks([], plt.yticks([]))
    plt.subplot(3, 3, i * 3 + 2), plt.hist(images[i * 3].ravel(), 256)
    plt.title(titles[i * 3 + 1]), plt.xticks([], plt.yticks([]))
    plt.subplot(3, 3, i * 3 + 3), plt.imshow(images[i * 3 + 2], 'gray')
    plt.title(titles[i * 3 + 2]), plt.xticks([], plt.yticks([]))
plt.show()
```

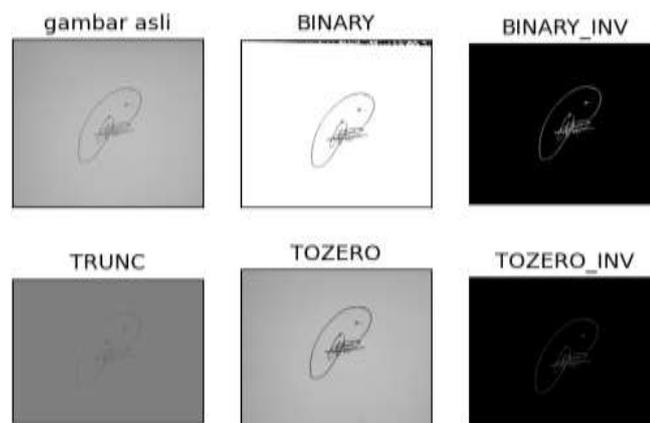
**Gambar 5.** Thresholding Otsu

Thresholding Sederhana: Menggunakan nilai ambang batas global (128 dalam contoh ini) untuk menghasilkan citra biner di mana piksel dengan intensitas di atas ambang batas dianggap sebagai objek (tanda tangan).

Thresholding Adaptif: Menggunakan metode thresholding adaptif dengan rata-rata berdasarkan lingkungan lokal citra untuk menyesuaikan ambang batas tergantung pada nilai piksel di sekitarnya.

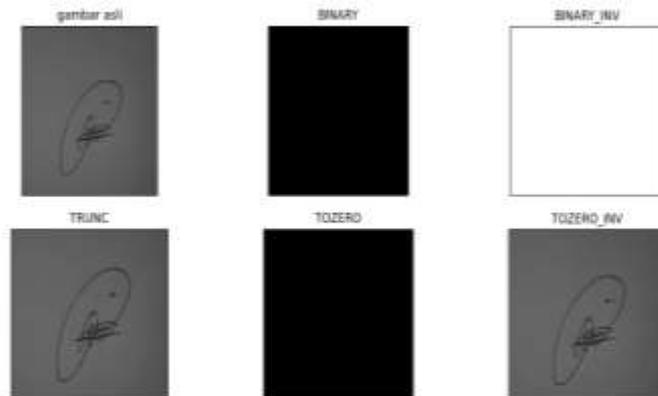
Binarisasi Otsu: Menggunakan analisis histogram untuk menentukan ambang batas secara otomatis yang memaksimalkan pemisahan antara objek (tanda tangan) dan latar belakang.

#### 4. ANALISIS DAN DISKUSI

**Gambar 6.** Thresholding Sederhana Terang

Deskripsi: Gambar hasil binarisasi menggunakan metode thresholding sederhana pada gambar dengan kondisi pencahayaan terang.

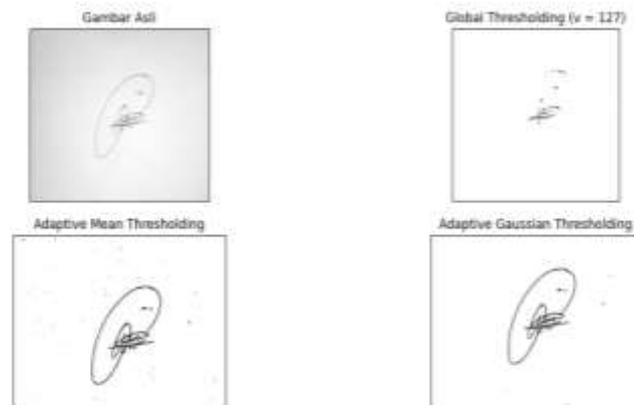
Analisis: Pada kondisi ini, metode thresholding sederhana berhasil memisahkan tanda tangan dari latar belakang dengan cukup baik, meskipun terdapat sedikit noise di sekitar tanda tangan.



**Gambar 7.** Thresholding Sederhana Gelap

Deskripsi: Gambar hasil binarisasi menggunakan metode thresholding sederhana pada gambar dengan kondisi pencahayaan gelap.

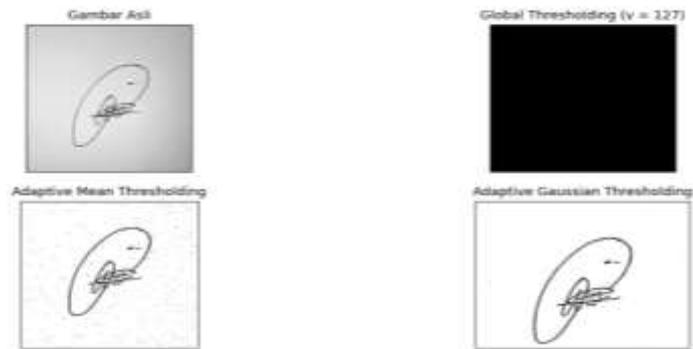
Analisis: Teknik ini kurang efektif pada kondisi pencahayaan gelap, di mana beberapa bagian tanda tangan tidak teridentifikasi dengan jelas, menghasilkan beberapa bagian yang tidak terpisah dengan baik dari latar belakang.



**Gambar 8.** Thresholding Adaptif Terang

Deskripsi: Gambar hasil binarisasi menggunakan metode thresholding adaptif pada gambar dengan kondisi pencahayaan terang.

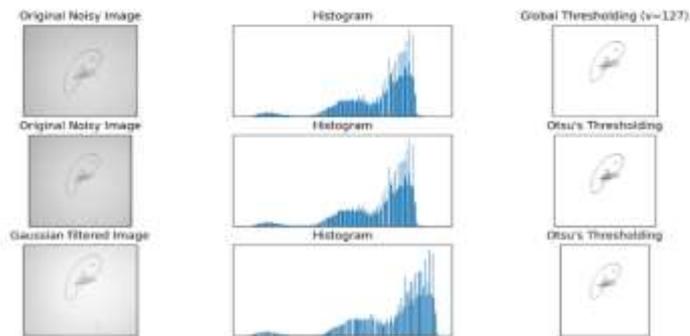
Analisis: Thresholding adaptif memberikan hasil yang lebih baik dibandingkan metode sederhana pada kondisi terang, dengan kemampuan yang lebih baik untuk menyesuaikan ambang batas berdasarkan variasi lokal pada gambar.



**Gambar 9.** Thresholding Adaptif Gelap

Deskripsi: Gambar hasil binarisasi menggunakan metode thresholding adaptif pada gambar dengan kondisi pencahayaan gelap.

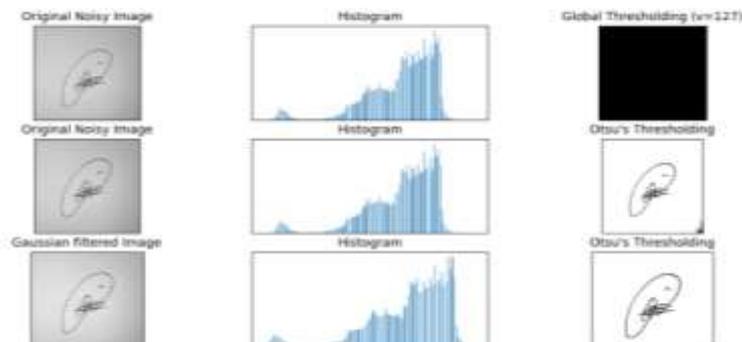
Analisis: Teknik ini juga menunjukkan peningkatan performa pada kondisigelap dibandingkan metode sederhana, namun masih terdapat beberapa bagian yang tidak terpisah sempurna.



**Gambar 10.** Binarisasi Otsu Terang

Deskripsi: Gambar hasil binarisasi menggunakan metode Otsu pada gambar dengan kondisi pencahayaan terang.

Analisis: Metode Otsu berhasil menentukan ambang batas optimal secara otomatis, memberikan hasil pemisahan yang jelas dan akurat pada kondisi pencahayaan terang.



**Gambar 11.** Binarisasi Otsu Gelap



Deskripsi: Gambar hasil binarisasi menggunakan metode Otsu pada gambar dengan kondisi pencahayaan gelap.

Analisis: Binarisasi Otsu juga menunjukkan performa yang baik pada kondisi pencahayaan gelap, dengan hasil pemisahan yang lebih baik dibandingkan metode sederhana dan adaptif, meskipun beberapa bagian kecil masih mengandung noise.

## 5. KESIMPULAN

Berdasarkan hasil yang diperoleh, dapat disimpulkan bahwa:

Thresholding Sederhana Efektif pada gambar dengan kondisi pencahayaan terang, namun kurang efektif pada kondisi pencahayaan gelap. Teknik ini menghasilkan noise yang signifikan dan kesulitan dalam memisahkan tanda tangan dari latar belakang dalam kondisi pencahayaan yang tidak merata.

Thresholding Adaptif Memberikan hasil yang lebih baik dibandingkan metode sederhana, terutama dalam kondisi pencahayaan yang tidak merata. Teknik ini mampu menyesuaikan ambang batas secara lokal, sehingga lebih efektif dalam menangani variasi pencahayaan dan noise, meskipun masih terdapat beberapa bagian yang tidak terpisah sempurna pada kondisi gelap.

Binarisasi Otsu Menunjukkan performa terbaik di antara ketiga teknik, baik pada kondisi pencahayaan terang maupun gelap. Metode Otsu berhasil menentukan ambang batas optimal secara otomatis, menghasilkan pemisahan yang lebih akurat dan jelas, meskipun terdapat sedikit noise pada beberapa gambar.

Secara keseluruhan, metode binarisasi Otsu direkomendasikan untuk binarisasi tanda tangan dalam berbagai kondisi pencahayaan karena kemampuannya dalam menentukan ambang batas secara optimal dan menghasilkan gambar biner yang jelas dan akurat. Penelitian ini memberikan kontribusi dalam memilih teknik thresholding yang tepat untuk aplikasi pengenalan tanda tangan digital, sehingga dapat meningkatkan keandalan dan akurasi sistem verifikasi tanda tangan.

## REFERENCES

- Anggraeni, D., & Wibawa, C. (2023). Perbaikan Citra Tanda Tangan Digital Menggunakan Metode Otsu Thresholding dan Sauvola. *Jurnal Ilmiah Matrik*, 25(1), 28–34. <https://doi.org/10.33557/jurnalmatrik.v25i1.2324>
- Razabni, D., Medinah, E., & Sinurat, S. (2020). Analisa dan Perbandingan Algoritma Otsu Thresholding dengan Algoritma Region Growing Pada Segmentasi Citra Digital. *Journal of Computer System and Informatics (JoSYC)*, 2(1), 9–16.
- Irfani, M. H., & Gasim, G. (2024). Segmentasi teks pada citra tulisan tangan kalimat menggunakan metode Median Filtering dan Otsu. *Teknosains: Media Informasi dan Teknologi*, 18(1), 88-97. <https://doi.org/10.24252/teknosains.v18i1.44307>