

PERBANDINGAN METODE OTSU THRESHOLDING DAN K-MEANS CLUSTERING UNTUK SEGMENTASI OBJEK PADA CITRA DIGITAL BERBASIS PYTHON

Muhammad Ilman Nafia Nursetya¹, Muhammad Naufal Rozaan², Ainurrasyid Rahman³, Ahmad Ilham⁴

¹Program Studi Informatika, Fakultas Ilmu Komputer dan Teknologi Digital, Universitas Muhammadiyah Semarang, Semarang, Indonesia

²Program Studi Informatika, Fakultas Ilmu Komputer dan Teknologi Digital, Universitas Muhammadiyah Semarang, Semarang, Indonesia

³Program Studi Informatika, Fakultas Ilmu Komputer dan Teknologi Digital, Universitas Muhammadiyah Semarang, Semarang, Indonesia

⁴Program Studi Informatika, Fakultas Ilmu Komputer dan Teknologi Digital, Universitas Muhammadiyah Semarang, Semarang, Indonesia

Email: ¹Ilmannafia20026@gmail.com, ²naufalrozaan49@gmail.com, ³ainurrasyidrahman6@gmail.com, ⁴dapurachmadilham@gmail.com

Info Artikel

Riwayat Artikel:

Diterima 9, Juli, 2026
Perbaikan 14, Juli, 2026
Disetujui 17, Juli, 2026

Keywords:

image segmentation, otsu thresholding, k-means clustering, digital image processing, python

ABSTRAK

Segmentasi citra merupakan tahap penting dalam pengolahan citra digital untuk memisahkan objek dari latar belakang sebelum dilakukan analisis lebih lanjut, seperti penghitungan jumlah objek, pengenalan pola, maupun ekstraksi fitur. Penelitian ini membandingkan dua metode segmentasi citra grayscale, yaitu Otsu Thresholding dan K-Means Clustering, yang diimplementasikan menggunakan bahasa pemrograman Python dengan pustaka scikit-image, scikit-learn, dan NumPy. Citra uji yang digunakan adalah citra koin logam berukuran 303x384 piksel dengan jumlah objek sebenarnya (ground truth) sebanyak 24 koin. Kedua metode diuji melalui tahap pra-pemrosesan, thresholding/clustering, operasi morfologi (penghapusan objek kecil, closing, dan penutupan lubang), serta pelabelan komponen terhubung untuk menghitung jumlah objek hasil segmentasi. Hasil pengujian menunjukkan bahwa kedua metode berhasil mendeteksi 23 dari 24 objek dengan akurasi yang sama, yaitu 95,83%, serta menghasilkan mask segmentasi yang identik secara piksel (Intersection over Union/IoU = 1,0000) karena histogram citra bersifat bimodal dengan kontras tinggi. Meskipun demikian, dari sisi efisiensi komputasi, Otsu Thresholding jauh lebih cepat (32,82 milidetik) dibandingkan K-Means Clustering (137,98 milidetik) karena Otsu menyelesaikan permasalahan secara analitis dalam satu kali pemindaian histogram, sedangkan K-Means memerlukan proses iteratif. Penelitian ini menyimpulkan bahwa Otsu Thresholding lebih direkomendasikan untuk kasus segmentasi citra dengan distribusi intensitas bimodal, sedangkan K-Means Clustering lebih fleksibel digunakan pada kasus segmentasi yang lebih kompleks atau multimodal.

ABSTRACT

Image segmentation is a crucial step in digital image processing used to separate objects from the background prior to further analysis, such as object counting, pattern recognition, and feature extraction. This study compares two grayscale image segmentation methods, Otsu Thresholding and K-Means Clustering, implemented

in the Python programming language using the scikit-image, scikit-learn, and NumPy libraries. The test image used is a 303x384-pixel coin image with a known ground truth of 24 coin objects. Both methods were evaluated through preprocessing, thresholding/clustering, morphological operations (small-object removal, closing, and hole filling), and connected-component labeling to count the resulting objects. The results show that both methods successfully detected 23 of 24 objects with the same accuracy of 95.83%, and produced pixel-identical segmentation masks (Intersection over Union/IoU = 1.0000) because the image histogram is bimodal with high contrast. However, in terms of computational efficiency, Otsu Thresholding was considerably faster (32.82 ms) than K-Means Clustering (137.98 ms), since Otsu solves the problem analytically in a single histogram scan, whereas K-Means requires an iterative process. This study concludes that Otsu Thresholding is more suitable for image segmentation with a bimodal intensity distribution, while K-Means Clustering is more flexible for more complex or multimodal segmentation cases.

Ini adalah artikel akses terbuka di bawah lisensi CC BY-SA.



Penulis Korespondensi:

Muhammad Ilman Nafia Nursetya

Program Studi Informatika, Fakultas Ilmu Komputer Dan Teknologi Digital, Universitas Muhammadiyah Semarang

Alamat: Gedung GKB 2Lt. 7, Ruang 707, Jl. Kedungmundu Raya No.18, Semarang 50273, Indonesia

Email: penulispertama@unimus.ac.id

1. PENDAHULUAN

Pengolahan citra digital (*digital image processing*) merupakan salah satu bidang ilmu komputer yang berkembang pesat dan banyak diterapkan pada berbagai bidang, seperti kedokteran, pertanian, keamanan, hingga industri manufaktur. Sebagai langkah krusial dalam pemrosesan citra digital, segmentasi berfungsi untuk mengisolasi entitas target dari latar belakangnya. Proses ini menjadi landasan bagi tahapan analisis berikutnya, seperti identifikasi pola, pengukuran jumlah objek, maupun ekstraksi atribut morfologis dari citra yang bersangkutan.

Terdapat berbagai pendekatan yang dapat digunakan untuk melakukan segmentasi citra, di antaranya adalah pendekatan berbasis *thresholding* dan pendekatan berbasis *clustering*. Metode Otsu Thresholding banyak digunakan karena efisiensinya dalam menentukan ambang batas secara otonom. Teknik ini bekerja dengan cara mencari titik potong

optimal yang meminimalkan variansi dalam kelas atau memaksimalkan variansi antar kelas antara piksel objek dan latar belakang.. Di sisi lain, Algoritma K-Means Clustering, yang termasuk dalam kategori pembelajaran unsupervised, digunakan untuk mengelompokkan piksel ke dalam k kelompok berdasarkan kedekatan nilai intensitas antar piksel tersebut. Kedua metode ini memiliki karakteristik komputasi yang berbeda: Otsu bersifat deterministik dan diselesaikan secara analitis, sedangkan K-Means bersifat iteratif dan bergantung pada inisialisasi.

Bahasa pemrograman Python banyak digunakan dalam penelitian pengolahan citra digital karena ketersediaan pustaka (*library*) yang lengkap dan mudah digunakan, seperti OpenCV, scikit-image, scikit-learn, dan NumPy. Pustaka-pustaka tersebut menyediakan implementasi siap pakai untuk berbagai algoritma pengolahan citra, termasuk Otsu Thresholding dan K-Means Clustering, sehingga memudahkan peneliti untuk

melakukan eksperimen dan analisis performa metode secara efisien.

Berdasarkan uraian tersebut, penelitian ini bertujuan untuk mengimplementasikan dan membandingkan metode Otsu Thresholding dan K-Means Clustering untuk segmentasi objek pada citra digital menggunakan Python. Perbandingan dilakukan berdasarkan tiga aspek, yaitu jumlah objek yang berhasil terdeteksi, akurasi terhadap jumlah objek sebenarnya (*ground truth*), dan waktu komputasi yang dibutuhkan oleh masing-masing metode.

2. METODE PENELITIAN

2.1. Data dan Perangkat Penelitian

Citra uji yang digunakan pada penelitian ini adalah citra *grayscale* koin logam berukuran 303x384 piksel yang bersumber dari kumpulan citra bawaan pustaka *scikit-image*. Citra ini dipilih karena memiliki jumlah objek yang telah diketahui secara pasti (*ground truth*) yaitu 24 koin, sehingga hasil segmentasi kedua metode dapat dievaluasi secara kuantitatif. Penelitian ini menggunakan bahasa pemrograman Python 3.12 dengan pustaka *scikit-image* 0.26 untuk operasi pengolahan citra dan morfologi, *scikit-learn* 1.8 untuk algoritma K-Means Clustering, *NumPy* untuk komputasi numerik, serta *Matplotlib* untuk visualisasi hasil.

2.2. Tahapan Penelitian

Tahapan penelitian yang dilakukan meliputi lima langkah utama, yaitu: (1) memuat citra uji; (2) melakukan segmentasi menggunakan metode Otsu Thresholding; (3) melakukan segmentasi menggunakan metode K-Means Clustering; (4) melakukan pasca-pemrosesan morfologi pada kedua hasil segmentasi; dan (5) melakukan evaluasi serta perbandingan hasil kedua metode.

Metode Otsu Thresholding bekerja dengan mencari nilai ambang t yang memaksimalkan variansi antar kelas (*between-class variance*) antara piksel latar belakang dan piksel objek, sebagaimana dirumuskan pada Persamaan (1).

$$\sigma^2_B(t) = \omega_0(t) \cdot \omega_1(t) \cdot [\mu_0(t) - \mu_1(t)]^2 \quad (1)$$

pada Persamaan (1), $\omega_0(t)$ dan $\omega_1(t)$ berturut-turut merupakan proporsi jumlah piksel pada kelas latar belakang dan kelas objek, sedangkan $\mu_0(t)$ dan $\mu_1(t)$ merupakan nilai rata-rata intensitas piksel pada masing-masing kelas. Nilai t yang memaksimalkan $\sigma^2_B(t)$ dipilih sebagai nilai ambang optimal secara otomatis.

Metode K-Means Clustering dengan jumlah kelompok $k=2$ digunakan untuk mengelompokkan piksel berdasarkan kemiripan nilai intensitasnya. Algoritma ini bekerja dengan meminimalkan fungsi objektif pada Persamaan (2).

$$J = \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2 \quad (2)$$

pada Persamaan (2), k adalah jumlah kelompok (dalam penelitian ini $k=2$, yaitu kelompok latar belakang dan kelompok objek), C_i adalah kelompok ke- i , x adalah nilai intensitas piksel, dan μ_i adalah titik pusat (*centroid*) kelompok ke- i . Kelompok dengan nilai pusat intensitas tertinggi dipilih sebagai kelompok objek (*foreground*).

Setelah proses *thresholding* maupun *clustering*, kedua mask biner hasil segmentasi diberikan pasca-pemrosesan morfologi yang sama, yaitu penghapusan objek kecil (*remove small objects*), operasi penutupan (*closing*) dengan elemen struktur berbentuk lingkaran berjari-jari 3 piksel, dan penutupan lubang kecil (*remove small holes*). Jumlah objek pada masing-masing hasil segmentasi kemudian dihitung menggunakan pelabelan komponen terhubung (*connected-component labeling*).

2.3. Metrik Evaluasi

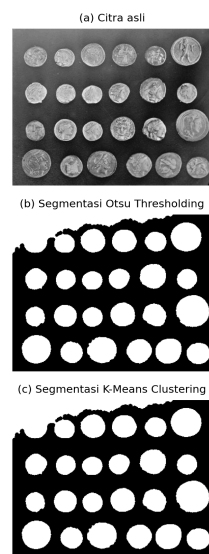
Perbandingan performa kedua metode dilakukan berdasarkan tiga metrik, yaitu: (1) jumlah objek yang berhasil terdeteksi dibandingkan dengan *ground truth* sebanyak 24 objek; (2) akurasi jumlah objek, dihitung dengan Persamaan (3); dan (3) waktu komputasi (dalam milidetik) yang dibutuhkan oleh masing-masing metode untuk menyelesaikan proses segmentasi hingga pelabelan objek.

$$\text{Akurasi} = (1 - |\text{Nterdeteksi} - \text{Nsebenarnya}| / \text{Nsebenarnya}) \times 100\% \quad (3)$$

Selain ketiga metrik tersebut, kesamaan piksel antara mask hasil Otsu dan mask hasil K-Means juga dihitung menggunakan *Intersection over Union* (IoU) untuk mengetahui seberapa identik kedua hasil segmentasi secara spasial.

3. HASIL DAN PEMBAHASAN

Implementasi kedua metode dilakukan pada citra uji koin logam. Nilai ambang optimal yang diperoleh dari metode Otsu Thresholding adalah $T=107$, sedangkan metode K-Means Clustering menghasilkan dua titik pusat kelompok yang batasnya secara efektif berada pada intensitas yang hampir sama dengan nilai ambang Otsu. Hasil visual segmentasi kedua metode ditunjukkan pada Gambar 1.



Gambar 1. Hasil Segmentasi Citra Koin: (a) Citra Asli, (b) Otsu Thresholding, (c) K-Means Clustering

Secara visual, mask hasil segmentasi Otsu Thresholding dan K-Means Clustering pada Gambar 1 tidak dapat dibedakan satu sama lain. Hal ini dikonfirmasi secara kuantitatif melalui perhitungan IoU antara kedua mask yang bernilai 1,0000, yang berarti kedua metode menghasilkan hasil segmentasi yang identik secara piksel. Kondisi ini terjadi karena histogram intensitas citra koin bersifat bimodal dengan kontras yang tinggi antara latar belakang dan objek, sehingga baik pendekatan analitis (Otsu) maupun pendekatan iteratif berbasis jarak (K-Means dengan $k=2$ pada data satu dimensi) menempatkan batas pemisah pada titik yang sama.

Perbandingan lengkap hasil kedua metode disajikan pada Tabel 1.

Tabel 1. Perbandingan Hasil Segmentasi Otsu Thresholding dan K-Means Clustering

Metode	Parameter	Objek Terdeteksi	Akurasi (%)	Waktu (ms)
Ground Truth	—	24	—	—
Otsu Thresholding	$T = 107$	23	95,83	32,82
K-Means Clustering	$k = 2$	23	95,83	137,98

Berdasarkan Tabel 1, kedua metode berhasil mendeteksi 23 dari 24 objek dengan akurasi yang sama, yaitu 95,83%. Selisih satu objek yang tidak terdeteksi kemungkinan besar disebabkan oleh dua koin yang posisinya saling bersinggungan pada citra asli, sehingga setelah operasi morfologi *closing* kedua koin tersebut menyatu menjadi satu komponen terhubung dan terhitung sebagai satu objek. Meskipun akurasi jumlah objek dan hasil segmentasi kedua metode identik, terdapat perbedaan signifikan pada aspek waktu komputasi. Otsu Thresholding hanya membutuhkan 32,82

milidetik, sedangkan K-Means Clustering membutuhkan 137,98 milidetik, atau sekitar 4,2 kali lebih lambat. Perbedaan ini terjadi karena Otsu Thresholding menyelesaikan permasalahan secara analitis dengan satu kali pemindaian histogram intensitas piksel, sedangkan K-Means Clustering harus melakukan proses iteratif berupa pembaruan titik pusat kelompok secara berulang hingga konvergen, dan pada penelitian ini bahkan dijalankan sebanyak empat kali inisialisasi berbeda ($n_init=4$) untuk menghindari hasil optimum lokal.

Temuan ini menunjukkan bahwa untuk kasus segmentasi citra dengan distribusi intensitas bimodal dan kontras tinggi seperti pada citra koin, metode Otsu Thresholding lebih unggul dari sisi efisiensi komputasi tanpa mengorbankan akurasi. Namun demikian, metode K-Means Clustering tetap memiliki keunggulan berupa fleksibilitas, karena dapat diperluas untuk melakukan segmentasi multi-kelas ($k>2$) maupun digunakan pada citra berwarna dengan memanfaatkan lebih dari satu kanal fitur, sesuatu yang tidak dapat dilakukan secara langsung oleh Otsu Thresholding yang hanya bekerja pada citra grayscale dengan dua kelas.

4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan dan membandingkan metode Otsu Thresholding dan K-Means Clustering untuk segmentasi objek pada citra digital menggunakan Python. Hasil pengujian pada citra koin logam dengan 24 objek *ground truth* menunjukkan bahwa kedua metode mencapai akurasi jumlah objek yang sama, yaitu 95,83% (23 dari 24 objek terdeteksi), serta menghasilkan mask segmentasi yang identik secara piksel ($IoU=1,0000$) karena histogram citra bersifat bimodal dengan kontras tinggi. Meskipun demikian, Otsu Thresholding terbukti sekitar 4,2 kali lebih cepat (32,82 ms) dibandingkan K-Means Clustering (137,98 ms) karena sifatnya yang analitis dan tidak iteratif. Berdasarkan temuan tersebut, Otsu Thresholding lebih direkomendasikan untuk kasus segmentasi citra dengan distribusi intensitas bimodal, sedangkan K-Means Clustering lebih sesuai digunakan pada kasus segmentasi yang lebih kompleks, multimodal, atau melibatkan citra berwarna. Penelitian selanjutnya dapat dikembangkan dengan menambah jumlah kelompok K-Means ($k>2$), menguji kedua metode pada dataset citra yang lebih beragam, serta membandingkannya dengan metode segmentasi berbasis pembelajaran mendalam (*deep learning*) seperti U-Net.

REFERENCES

- BRADSKI, G., 2000. The OpenCV Library. Dr. Dobb's Journal of Software Tools, 25(11), pp.120–125.

- GONZALEZ, R.C. and WOODS, R.E., 2018. Digital Image Processing. 4th ed. New York: Pearson.
- HARRIS, C.R., MILLMAN, K.J., VAN DER WALT, S.J., GOMMERS, R., VIRTANEN, P., COURNAPEAU, D., et al., 2020. Array Programming with NumPy. *Nature*, 585, pp.357–362.
- MACQUEEN, J., 1967. Some Methods for Classification and Analysis of Multivariate Observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1, pp.281–297.
- OTSU, N., 1979. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), pp.62–66.
- PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., et al., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, pp.2825–2830.
- PUTRA, D., 2010. *Pengolahan Citra Digital*. Yogyakarta: Andi Offset.
- VAN DER WALT, S., SCHÖNBERGER, J.L., NUNEZ-IGLESIAS, J., BOULOGNE, F., WARNER, J.D., YAGER, N., GOUILLART, E. and YU, T., 2014. Scikit-image: Image Processing in Python. *PeerJ*, 2, e453.