

# Deteksi Relief Candi Borobudur Menggunakan Metode Template Matching

Puspadevi Anggotra<sup>a1</sup>, Agus Muliantara<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Bali, Indonesia

<sup>1</sup>devianggotra@gmail.com

<sup>2</sup> muliantara@unud.ac.id

## Abstract

*Borobudur Temple is a famous Buddhist temple that has an area of about 2500 square meters. This temple has been a famous tourist destination since a long time ago. Additionally, this building's walls are adorned with 504 Buddha sculptures and 2,672 relief panels.. The large number of reliefs makes people often do not know information about these reliefs. Even though each relief itself actually has many stories of its own. Due to this issue, a relief detector is required so that visitors can learn more details about the reliefs. In this case, the authors use template matching with the Minimum Square Difference and Normalized Cosine Coefficient methods to identify relief. The stage for relief detection starts from the preprocessing stage and then performs template matching using 2 different methods. The test results from 23 images obtained the percentage of success of the system using the Normalized Cosine Coefficient method 23,52% and Minimum Square Difference method is 84,31%.*

**Keywords:** Relief, Temple, Template Matching, Image Processing

## 1. Pendahuluan

Candi Borobudur merupakan destinasi wisata berupa sebuah candi yang terletak di Kabupaten Magelang, Jawa Tengah. Bangunan megah ini diakui oleh wisatawan domestik dan mancanegara sebagai candi Budha terbesar di dunia. Luas dari candi Borobudur sekitar 2500 meter persegi, dengan panjang 121,66 meter, lebar 121,38 meter, dan tinggi 35,40 meter. Dengan jutaan blok batu yang memebentuk Candi Borobudur membuat bagunan candi ini sangatlah luas. Bangunan dari Candi Borobudur terdiri dari tiga susunan yang disamakan dengan bidang alam semesta menurut ajaran Buddha. Selain itu, bangunan ini juga dihiasi oleh 2.672 panel relief dan 504 arca Buddha [1].

Jumlah relief yang banyak membuat masyarakat sering kali tidak tahu informasi mengenai relief tersebut. Padahal tiap relief sendiri sebenarnya memiliki banyak cerita tersendiri. Belum lagi dengan jumlahnya yang banyak kerap kali relief itu susah untuk diidentifikasi. Sumber Daya Manusia juga tidak bisa selamanya bisa menyebarkan informasi mengenai relief kepada wisatawan mengingat banyaknya wisatawan yang tidak menggunakan tour guide dan manusia juga tidak selamanya bisa mengingat sejarah 2.672 relief. Dikarenakan permasalahan tersebut maka dibutuhkan adanya pendeteksi relief yang dapat membantu wisatawan untuk mengetahui informasi lanjut mengenai relief-relief tersebut sekaligus mengetahui cerita yang melatarbelakangi relief tersebut.

Dalam penelitian ini, metode template matching digunakan oleh penulis untuk mendeteksi relief. Template matching adalah sebuah teknik dalam pemrosesan citra digital yang digunakan untuk membandingkan kecocokan citra referensi (template) dengan setiap bagian dari suatu citra masukan [2]. Bagaimana cara otak manusia mengenali kembali bentuk-bentuk atau pola-pola dijelaskan dalam metode ini dengan implementasinya pada komputer. Pada penelitian sebelumnya, yang berhubungan dengan penelitian ini yaitu deteksi logo universitas yang diteliti oleh Sayuti Rahman, dkk, 2020 mendapatkan akurasi 84%. Didasarkan penelitian yang dilakukan sebelumnya, maka pada penelitian ini penulis melakukan menggunakan metode template matching sebagai metode untuk melakukan deteksi relief candi borobudur [3].

## 2. Metodologi Penelitian

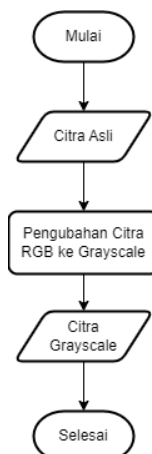
Tahapan pembuatan aplikasi ini adalah mengumpulkan data gambar untuk template (referensi) dan data uji. Gambar yang dimasukkan ke dalam sistem akan melalui tahap pra-pemrosesan (*pre-processing*) setelahnya gambar akan template matching dengan library OpenCV pada python.

### 2.1 Dataset

Data yang dipakai dalam penelitian ini merupakan data sekunder berupa data gambar yang didapatkan dari <https://www.kaggle.com/> [4]. Penulis akan menggunakan 51 gambar relief sebagai data uji dan 11 gambar relief sebagai template. Gambar relief ini terdiri dari beberapa jenis dengan tingkat kecerahan yang berbeda, saturasi yang berbeda dan ukuran yang berbeda.

### 2.2 Preprocessing Image

Sebelum melakukan template matching, data gambar yang masuk akan melalui tahap preprocessing terlebih dahulu. Tahap preprocessing merupakan tahap yang paling penting dalam template matching [5]. Hal ini dilakukan dilakukan untuk meningkatkan kualitas gambar (citra) sehingga dapat dianalisa secara lebih baik. Tanpa tahap preprocessing data tidak akan bisa diproses secara baik. Pada tahap ini citra RGB akan diubah menjadi grayscale. Tahapan preprocessing dapat dilihat pada Gambar 1.



**Gambar 1.** Flowchart Preprocessing

### 2.3 Template Matching

Proses mencari suatu objek yang dijadikan referensi (template) pada keseluruhan objek yang terkandung dalam suatu citra dikenal dengan template matching. Metode ini sering dipakai buat mendeteksi huruf, nomor, sidik jari (fingerprint) & pelaksanaan-pelaksanaan pencocokan gambaran lainnya. Template Mathing merupakan salah satu penerapan teknik konvolusi yang secara generik didefinisikan menjadi suatu cara buat mengkombinasikan 2 butir deret nomor yg membentuk deret nomor ke tiga [6]. Cara bekerja template matching adalah dengan membandingkan keseluruhan objek tersebut [7]. Gambar input ditempatkan di bawah gambar template. Setelah citra template dibandingkan dengan citra input, maka hasil akhirnya akan berupa citra grayscale. Gambar skala abu-abu ini akan menunjukkan seberapa banyak lingkungan piksel akan cocok dengan gambar template [8].

Terdapat beberapa metode dalam proses template matching. Yang akan digunakan dalam penelitian ini adalah Minimum Square Difference (TM\_SQDIFF) dan Normalized Cosine Coefficient (TM\_CCoeff\_NORMED). Minimum Square Difference digunakan ketika ingin mencari perbedaan minimum antara gambar template dan gambar sumber. Sedangkan Normalized Cosine Coefficient karena kita perlu mendapatkan nilai maksimum, bukan nilai minimum. Yang berarti kita perlu mencari beberapa objek, bukan satu. Kedua Metode ini sendiri merupakan turunan dari metoder correlation coefficient.

Untuk setiap lokasi T di atas I, metrik hasil yang dihitung disimpan dalam matriks hasil R. Setiap koordinat (x, y) dalam gambar sumber (yang juga memiliki lebar dan tinggi yang valid untuk gambar template) berisi entri di matriks hasil R. Matriks R tidak berukuran sama dengan template aslinya. Ini dikarenakan seluruh template harus pas di dalam gambar sumber untuk korelasi yang akan dihitung [9].

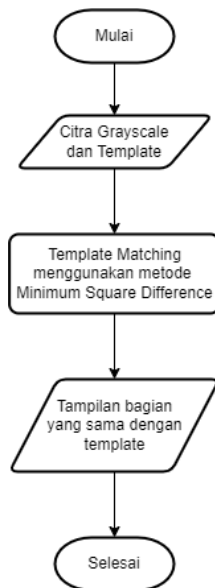
$$R(x, y) = \sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2$$

**Gambar 2.** Rumus Metode Minimum Square Difference [10]

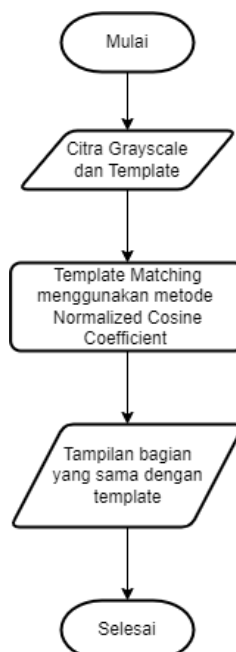
$$R(x, y) = \frac{\sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))}{\sqrt{\sum_{x', y'} T'(x', y')^2 \cdot \sum_{x', y'} I'(x + x', y + y')^2}}$$

**Gambar 3.** Rumus Metode Normalized Cosine Coefficient [10]

Tahapan template matching dengan metode Minimum Square Difference dan dengan metode Normalized Cosine Coefficient dapat dilihat pada Gambar 4 dan 5.



**Gambar 4.** Flowchart Template Matching dengan Metode Minimum Square Difference



**Gambar 5.** Flowchart Template Matching dengan Metode Normalized Cosine Coefficient

## 2.4 Desain Sistem

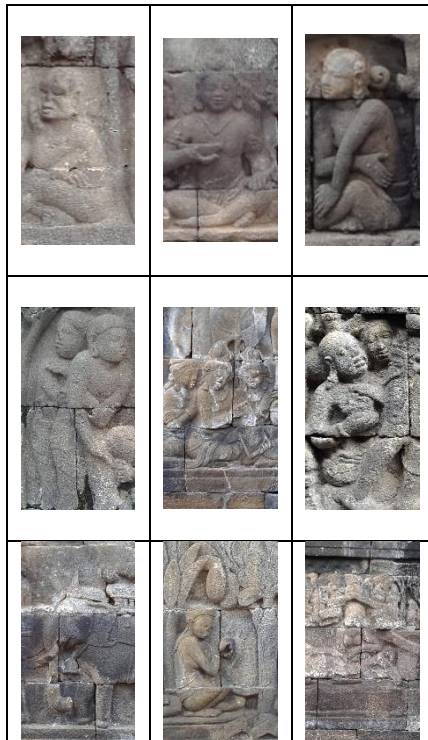
Gambaran umum dari sistem yang ini dimulai dengan memasukan citra asli. Citra asli kemudian akan melewati tahap preprocessing yang akan menghasilkan keluaran berupa citra grayscale. Setelahnya citra akan melewati proses perbandingan dengan template. Apabila objek yang dijadikan template cocok atau mirip dengan suatu objek pada citra maka objek tersebut akan ditandai dengan persegi panjang berwarna biru.

### 3. Hasil dan Pembahasan

#### a. Template

Template atau referensi yang digunakan sendiri terdiri dari 3 gambar citra

**Table 1.** Citra Template



b. Citra Uji Menggunakan Metode Minimum Square Difference

Berikut merupakan beberapa hasil citra yang terdeteksi ketika diuji menggunakan metode minimum square difference.

**Table 2.** Hasil Citra Yang Berhasil Dideteksi Menggunakan Metode Minimum Square Difference



Berikut merupakan beberapa hasil citra yang tidak berhasil terdeteksi dengan benar ketika diuji menggunakan metode minimum square difference.

**Table 3.** Hasil Citra Yang Tidak Terdeteksi Menggunakan Metode Minimum Square Difference



- c. Citra Uji Menggunakan Metode Normalized Cosine Coefficient  
Berikut merupakan beberapa hasil citra yang terdeteksi ketika diuji menggunakan metode normalized cosine coefficient.

**Table 2.** Hasil Citra Uji Yang Berhasil Dideteksi Menggunakan Metode Normalized Cosine Coefficient



Berikut merupakan beberapa hasil citra yang tidak berhasil terdeteksi dengan benar ketika diuji menggunakan metode normalized cosine coefficient.

**Table 2.** Hasil Citra Yang Tidak Terdeteksi Menggunakan Metode Normalized Cosine Coefficient



Adapun beberapa simpulan berdasarkan pengujian dengan 23 data uji dengan dua metode yang berbeda :

1. Hasil deteksi dengan metode template matching sangat dipengaruhi oleh pemilihan template. Hal ini dikarenakan perbedaan sudut pengambilan gambar, pencahayaan dan kualitas citra sangat mempengaruhi hasil.
2. Dari 51 citra uji 43 citra uji berhasil dideteksi, 8 citra uji tidak terdeteksi dengan benar menggunakan metode Minimum Square Difference. Sedangkan dengan menggunakan metode Normalized Cosine Coefficient, dari 51 citra uji hanya 12 citra uji yang berhasil dideteksi.
3. Hasil pengujian dari 51 citra didapatkan persentase keberhasilan sistem dengan menggunakan metode Minimum Square Difference adalah 84,31%. Sedangkan dengan menggunakan metode Normalized Cosine Coefficient persentase keberhasilan sistem hanya 23,52%.

#### **4. Kesimpulan**

Berdasarkan dari hasil penelitian dan pengujian yang telah dilakukan, didapatkan kesimpulan bahwa metode template matching dapat digunakan untuk mendeteksi relief pada Borobudur. Hasil pengujian relief yang dilakukan dengan menggunakan metode Minimum Square Difference mencapai akurasi sebesar 84,31% sehingga dapat dikatakan cukup akurat untuk sistem deteksi relief Candi Borobudur. Sedangkan dengan menggunakan metode Normalized Cosine Coefficient persentase keberhasilan sistem hanya 23,52%. Hal ini menyebabkan metode Normalized Cosine Coefficient tidak dianjurkan untuk digunakan dalam sistem ini.



## References

- [1] "Candi Borobudur, Pusaka Kolosal Magelang, Jawa Tengah", Wonderful Indonesia, <https://www.indonesia.travel/id/id/destinasi/java/magelang-regency/borobudur> [Diakses 02 Oktober 2022].
- [2] G. J. Pradana Putra and I. G. N. A. Cahyadi Putra, "University Logo Recognition Application Design Using Template Matching Method," *JELIKU (Jurnal Elektron. Ilmu Komput. Udayana)*, vol. 9, no. 2, p. 221, 2020, doi: 10.24843/jlk.2020.v09.i02.p08.
- [3] S. Rahman and U. Sahira, "Pengenalan Iris Mata Menggunakan Metode Template Matching Correlation," *J. Teknol. Inf.*, vol. 2, no. 2, p. 105, 2019, doi: 10.36294/jurti.v2i2.424.
- [4] R. A. Yunmar, "Dataset Images of Borobudur Temple Relief", Kaggle, <https://www.kaggle.com/datasets/rajiva/dataset-of-borobudur-temple-relief> [Diakses 26 September 2022].
- [5] W. D. Sidi and I. G. Arta Wibawa, "Sum of Squared Difference (SSD) Template Matching Testing on Writing Learning Application," *JELIKU (Jurnal Elektron. Ilmu Komput. Udayana)*, vol. 8, no. 4, p. 453, 2020, doi: 10.24843/jlk.2020.v08.i04.p11.
- [6] S. M. Chairi, G. Garuda, J. Matias, and Sirati Fika, "PENERAPAN METODE TEMPLATE MATCHING PADA CITRA BERWARNA," vol. 7, pp. 310–312, 2019.
- [7] M. Sonsank, Y. Huda, and K. Budayawan, "Penerapan Metode Template Matching Dalam Menganalisa Cacat Pada Keping Pcb," *Voteteknika (Vocational Tek. Elektron. dan Inform.)*, vol. 3, no. 1, 2015, doi: 10.24036/voteteknika.v3i1.5183.
- [8] M. Naharul, H. Najihul, and S. Adinugroho, "Implementasi Metode Template Matching untuk Mengenali Nilai Angka pada Citra Uang Kertas yang Dipindai," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 2, pp. 1550–1556, 2019.
- [9] A. Rosebrock, "OpenCV Template Matching ( cv2.matchTemplate )", Pyimagesearch, <https://pyimagesearch.com/2021/03/22/opencv-template-matching-cv2-matchtemplate/> [Diakses 29 Oktober 2022]
- [10] "Template Matching", OpenCV, [https://docs.opencv.org/3.4/de/da9/tutorial\\_template\\_matching.html](https://docs.opencv.org/3.4/de/da9/tutorial_template_matching.html) [Diakses 29 Oktober 2022]

Halaman ini sengaja dibiarkan kosong