

## PENGENALAN MOTIF BATIK INDONESIA MENGGUNAKAN DETEKSI TEPI CANNY DAN *TEMPLATE MATCHING*

<sup>[1]</sup>Fera Flaurensia,<sup>[2]</sup>Tedy Rismawan,<sup>[3]</sup>Rahmi Hidayati

<sup>[1]</sup> <sup>[2]</sup> <sup>[3]</sup>Jurusan Sistem Komputer, Fakultas MIPA Universitas Tanjungpura  
Jl. Prof. Dr. H. Hadari Nawawi, Pontianak  
Telp./Fax.: (0561) 577963

e-mail:

<sup>[1]</sup>fera.fr99@student.untan.ac.id, <sup>[2]</sup>tedyrismawan@siskom.untan.ac.id,  
<sup>[3]</sup>rahmihidayati@siskom.untan.ac.id

### Abstrak

*Batik menjadi salah satu identitas bangsa Indonesia yang telah ditetapkan UNESCO sebagai Warisan Budaya Tak Benda Warisan Manusia. Hal ini membuktikan bahwa batik telah menjadi salah satu bentuk kekayaan kebudayaan Indonesia yang patut dilestarikan. Upaya pelestarian batik tentunya tak terlepas dari upaya pemahaman dan pengenalan corak dan motif batik Indonesia yang sangat beragam. Seiring dengan berkembangnya teknologi pengolahan citra sekarang ini, proses pengenalan motif batik dapat dibantu dengan menggunakan metode-metode dalam pengolahan citra. Penelitian ini bertujuan untuk membuat suatu aplikasi pengenalan motif batik Indonesia yang memanfaatkan pengolahan citra menggunakan deteksi tepi canny dan template matching. Hasil dari penelitian ini adalah semua motif batik yang diuji dapat dikenali oleh aplikasi yang dibangun. Persentase kemiripan dalam pengujian mengenali motif batik sebesar 89,44%, berdasarkan hasil tersebut, maka dapat disimpulkan bahwa pengenalan pola dengan metode deteksi tepi canny dan template matching dapat digunakan untuk pengenalan motif batik Indonesia.*

**Kata Kunci** : deteksi tepi, canny, *template matching*, pengenalan batik, pengolahan citra

## 1. PENDAHULUAN

Upaya pelestarian batik tentunya tak terlepas dari upaya pemahaman dan pengenalan corak dan motif batik Indonesia yang sangat beragam. Keberagaman ini membuat masyarakat sulit membedakan motif batik yang ada dan daerah asal batik tersebut. Oleh sebab itu diperlukan bantuan teknologi pengolahan citra yang dapat mengenali motif batik Indonesia

## 2. LANDASAN TEORI

### 2.1. Batik

Dalam Kamus Besar Bahasa Indonesia, batik dijelaskan sebagai kain bergambar yang dibuat secara khusus dengan menuliskan atau menerakan malam (lilin) pada kain, kemudian pengolahannya diproses dengan cara tertentu. Setiap daerah di Indonesia memiliki ciri khas tertentu pada motif batik dan pakem (cara motif diorganisasi)[1]

Motif batik dibagi menjadi dua kelompok besar, yaitu motif geometri dan non-geometri. Motif geometri misalnya motif parang, motif ceplok, dan motif lereng atau liris. Motif non-geometri misalnya motif semen, motif lung-lungan, dan motif buketan[2]

### 2.2. Citra Digital

Citra digital adalah citra kontinu yang diubah dalam bentuk diskrit, baik koordinat ruang maupun intensitas cahayanya. Citra digital dapat dinyatakan dalam matrik dua dimensi  $f(x,y)$  dimana 'x' dan 'y' merupakan koordinat *pixel* dalam matrik dan 'f' merupakan derajat intensitas *pixel* tersebut. Citra digital berbentuk matrik dengan ukuran M x N akan tersusun sebagai berikut [3]

$$f(x,y) = \begin{pmatrix} f(0,0) & f(0,1) & f(0,2) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) & \dots & f(1,N-1) \\ f(2,0) & f(2,1) & f(2,2) & \dots & f(2,N-1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & f(M-1,2) & \dots & f(M-1,N-1) \end{pmatrix}$$

Dimana :

M = banyaknya baris pada array citra

N = banyaknya kolom pada array citra

### 2.3. Pengolahan Citra

Pengolahan citra adalah istilah umum untuk berbagai teknik yang keberadaannya untuk memanipulasi dan memodifikasi citra dengan berbagai cara.

Aplikasi-aplikasi dalam pengolahan citra sesungguhnya menggunakan prinsip dasar dalam pengolahan citra yang dapat diklasifikasikan sebagai berikut[4]

1. Peningkatan Kecerahan dan Kontras
2. Penghilangan Derau
3. Pencarian Bentuk Objek

#### 2.3.1. Penapisan Citra (*Filtering*)

Pada pengolahan citra digital, filter digunakan untuk menekan frekuensi tinggi pada citra. Tujuan dari proses penapisan adalah membuat citra menjadi tampak lebih baik atau lebih jelas untuk analisis [5]

Proses penapisan tidak dapat dilepaskan dari teori kernel (*mask*) dan konvolusi. Kernel digunakan pada proses konvolusi. *Edge detection* merupakan salah satu proses pengolahan citra yang menggunakan filter, yang dilakukan melalui metode konvolusi. Operasi konvolusi bekerja dengan menggeser kernel piksel per piksel, yang hasilnya disimpan dalam matrik baru[6]

#### 2.3.2. Segmentasi Citra

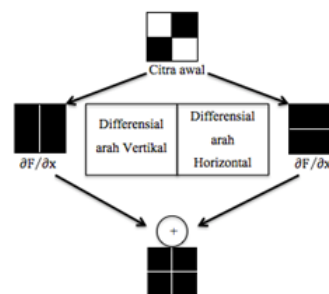
Segmentasi citra merupakan proses yang ditujukan untuk mendapatkan objek-objek yang terkandung dalam citra atau membagi citra ke dalam beberapa daerah dengan setiap objek atau daerah memiliki kemiripan atribut. Pada citra yang mengandung sejumlah objek, proses untuk memilah semua objek tentu akan lebih kompleks. Secara prinsip, segmentasi dilakukan untuk mendapatkan objek yang menjadi perhatian [4]

### 2.4. Deteksi Tepi

Yang dimaksud dengan tepi (*edge*) adalah perubahan nilai intensitas derajat keabuan yang mendadak (besar) dalam jarak yang singkat. Perbedaan intensitas inilah yang menampakkan rincian pada gambar. Tepi biasanya terdapat pada batas antara dua

daerah berbeda pada suatu citra

Tepi mencirikan batas-batas objek dan karena itu tepi berguna untuk proses segmentasi dan identifikasi objek di dalam citra. Tujuan operasi pendeteksian tepi adalah untuk meningkatkan penampakan garis batas suatu daerah atau objek di dalam citra. Karena tepi termasuk ke dalam komponen berfrekuensi tinggi, maka pendeteksian tepi dapat dilakukan dengan *high-pass filter*[7]



Gambar 1. Proses Deteksi Tepi  
(Sumber : Yodha, 2014)

#### 2.4.1. Deteksi Tepi Canny

Algoritma ini memberikan tingkat kesalahan rendah, melokalisasi titik-titik tepi (jarak piksel-piksel tepi yang ditemukan deteksi dan tepi sesungguhnya sangat pendek), dan hanya memberikan satu tanggapan untuk satu tepi. Terdapat enam langkah yang dilakukan untuk mengimplementasikan deteksi tepi *canny*. Keenam langkah tersebut dijabarkan berikut:[3]

Langkah 1:

Pertama-tama dilakukan penapisan terhadap citra dengan tujuan untuk menghilangkan derau. Hal ini dapat dilakukan dengan menggunakan filter *Gaussian*.

Langkah 2:

Setelah penghalusan gambar terhadap derau dilakukan, dilakukan proses untuk mendapatkan kekuatan tepi (*edge strength*). Hal ini dilakukan dengan menggunakan operator sobel. Selanjutnya, gradien citra dapat dihitung melalui persamaan:

$$|G| = |G_x| + |G_y| \quad (1)$$

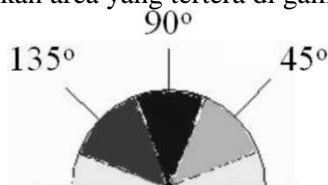
Langkah 3:

Langkah ketiga berupa perhitungan arah tepi. Persamaan yang digunakan untuk keperluan ini:

$$\theta = \tan^{-1} (G_y, G_x) \quad (2)$$

Langkah 4:

Setelah arah tepi diperoleh, hubungkan antara arah tepi dengan sebuah arah yang dapat dilacak dari citra. Selanjutnya, arah tepi yang diperoleh dimasukkan ke dalam salah satu kategori dari keempat arah tadi berdasarkan area yang tertera di gambar 2.



Gambar 2. Area untuk mengkonversi arah tepi ke dalam kategori salah satu arah  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , dan  $135^\circ$

(Sumber : Kadir & Susanto, 2013)

Berikut adalah aturan konversi yang berlaku: arah =

$$\begin{cases} 0, & 0 \leq x < 22,5 \text{ atau } x \geq 157,5 \\ 45, & 22,5 \leq x < 67,5 \\ 90, & 67,5 \leq x < 112,5 \\ 135, & 112,5 \leq x \leq 157,5 \end{cases} \quad (3)$$

Semua arah tepi yang berkisar antara  $0^\circ$  dan  $22,5^\circ$  serta  $157,5^\circ$  dan  $180^\circ$  diubah menjadi  $0^\circ$ . Semua arah tepi yang berkisar antara  $22,5^\circ$  dan  $67,5^\circ$  diubah menjadi  $45^\circ$ . Semua arah tepi yang berkisar antara  $67,5^\circ$  dan  $112,5^\circ$  diubah menjadi  $90^\circ$ . Semua arah tepi yang berkisar antara  $112,5^\circ$  dan  $157,5^\circ$  diubah menjadi  $135^\circ$ .

Langkah 5:

Setelah arah tepi diperoleh, penghilangan nonmaksimum dilaksanakan. Penghilangan nonmaksimum dilakukan di sepanjang tepi pada arah tepi dan menghilangkan piksel-piksel (piksel diatur menjadi 0) yang tidak dianggap sebagai tepi. Dengan cara seperti itu, diperoleh tepi tipis.

Langkah 6:

Langkah keenam berupa proses yang disebut *hysteresis*. Proses ini menghilangkan garis-garis yang seperti terputus-putus pada tepi objek. Peng-ambangan hysteresis dilakukan dengan melibatkan dua ambang  $T_1$  (ambang bawah) dan  $T_2$  (ambang atas). Nilai yang kurang dari  $T_1$  akan diubah menjadi hitam (nilai 0) dan yang lebih dari  $T_2$  akan menjadi putih (nilai 255). Nilai yang lebih dari atau sama dengan  $T_1$  tetapi kurang dari  $T_2$  untuk sementara akan diberi nilai 128, dan yang menyatakan nilai abu-abu atau belum jelas akan dijadikan 0 atau 255.

## 2.5. Pengenalan Pola

Secara umum pengenalan pola (*pattern recognition*) adalah suatu ilmu untuk mengklasifikasi atau menggambarkan sesuatu berdasarkan pengukuran kuantitatif fitur (ciri) atau sifat utama dari suatu objek. Pola sendiri adalah suatu entitas yang terdefinisi dan dapat diidentifikasi serta diberi nama[5]

Pengenalan pola bertujuan menentukan kelompok atau kategori pola berdasarkan ciri-ciri yang dimiliki oleh pola tersebut. Dengan kata lain, pengenalan pola membedakan suatu obyek dengan obyek lain[7]

### 2.5.1. *Template Matching*

*Template matching* adalah salah satu teknik dalam pengolahan citra digital yang berfungsi untuk mencocokkan tiap-tiap bagian dari suatu citra dengan citra yang menjadi *template* (acuan). Suatu citra masukan yang mengandung *template* tertentu dibandingkan dengan *template* acuan. *Template* ditempatkan pada pusat bagian citra yang akan dibandingkan dan dihitung seberapa banyak titik yang paling sesuai dengan *template*. Langkah ini diulangi terhadap keseluruhan citra masukan yang akan dibandingkan. Nilai kesesuaian titik yang paling besar antara citra masukan dan citra *template* menandakan *template* tersebut merupakan citra *template* yang paling sesuai dengan citra masukan [8]

Metode pencocokan pola adalah salah satu metode terapan dari teknik konvolusi. Teknik konvolusi dilakukan dengan meng-kombinasikan citra masukan dengan citra sumber acuan, hingga akan didapatkan nilai koefisien korelasi yang besarnya antara -1 dan +1. Rumus yang digunakan adalah:[9]

$$r = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (x_{ij} - \bar{x})(y_{ij} - \bar{y})}{\sqrt{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (x_{ij} - \bar{x})^2 \cdot \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (y_{ij} - \bar{y})^2}} \quad (4)$$

Keterangan :

- $r$  : Koefisien korelasi
- $x$  : Citra acuan (*template*)
- $\bar{x}$  : Nilai rata-rata citra acuan
- $y$  : Nilai rata-rata citra masukan
- $y$  : Citra masukan
- $M, N$  : Jumlah *pixel* citra

### 3. METODOLOGI PENELITIAN

Penelitian ini menggunakan metodologi penelitian yang mencakup studi literatur, kemudian dilakukan analisa kebutuhan akan komponen-komponen yang dibutuhkan dalam pembuatan aplikasi, selanjutnya perancangan sistem, kemudian dilakukan tahap implementasi dan pengujian, serta analisis.

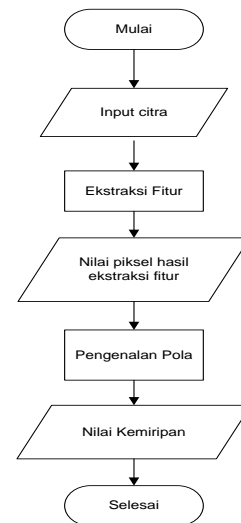
### 4. PERANCANGAN SISTEM

#### 4.1. Rancangan Sistem Secara Umum

Sistem pengenalan motif batik ini dilakukan berdasarkan nilai hasil ekstraksi fitur motif batik dengan menggunakan metode deteksi tepi *canny* dan pencocokan pola dengan menggunakan metode *template matching*. Motif-motif batik yang digunakan dalam penelitian ini meliputi motif batik yang termasuk dalam motif geometri (motif parang, ceplok blingon, liris cemeng, dan liris panjang madura) dan motif non-geometri (motif semen rante, semen sinom, lung-lungan bledakan, babon angrem, dan buketan). *Flowchart* rancangan sistem dapat dilihat pada gambar 3.

Penjelasan rincian dari masing-masing tahapan akan dijelaskan sebagai berikut :

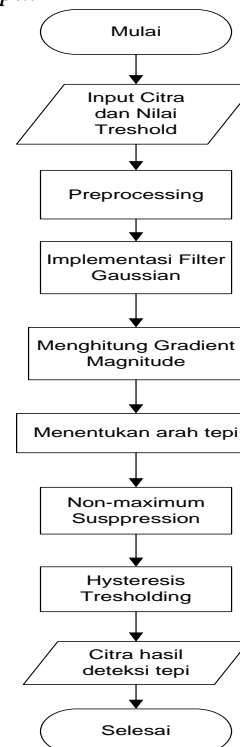
1. Sistem dimulai dari *input* citra
2. Dilanjutkan proses ekstraksi fitur. Pada tahap ekstraksi fitur ini, metode yang digunakan adalah deteksi tepi *canny*. Proses ini menghasilkan citra dengan nilai piksel berupa 0 dan 255. *Flowchart* ekstraksi fitur bentuk dapat dilihat pada gambar 4.
3. Setelah diperoleh nilai piksel hasil ekstraksi ciri, tahap selanjutnya adalah mencocokkan nilai citra *input* tersebut dengan nilai citra acuan. Tahap ini disebut tahap pengenalan pola. Pada tahap ini digunakan metode *template matching*. *Flowchart* pengenalan pola dapat dilihat pada gambar 5.
4. Hasil dari sistem adalah nilai kemiripan citra *input* dengan masing-masing citra acuan.



Gambar 3. *Flowchart* Rancangan Sistem

#### 4.1.1. Rancangan Ekstraksi Fitur

Metode ekstraksi yang digunakan pada penelitian ini adalah deteksi tepi *canny*. Ekstraksi fitur dilakukan pada citra acuan dan citra *input*



Gambar 4. *Flowchart* Ekstraksi Fitur

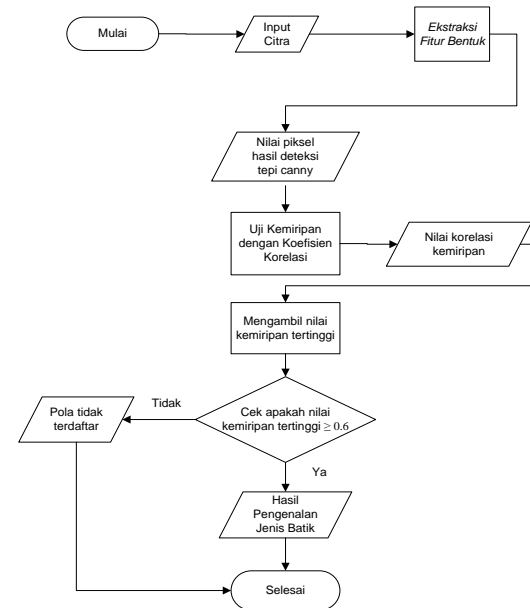
Penjelasan rincian dari masing-masing tahapan akan dijelaskan sebagai berikut :

1. Proses dimulai.
2. *Input* citra dan nilai treshold. Pada penelitian ini, nilai *low threshold* yang digunakan adalah 0,05 dan nilai *high*

*threshold* yang digunakan adalah 0,15.

3. Pada tahap *preprocessing*, citra yang berada pada ruang warna RGB dikonversi ke citra *grayscale*.
4. Menghilangkan derau yang ada pada citra dengan mengimplementasikan *filter Gaussian*.
5. Menghitung *gradient magnitude* dengan menggunakan operator *sobel* dan menentukan arah tepi. Setelah proses penentuan *gradient magnitude*, gambar terlihat telah menunjukkan tepinya dengan cukup jelas, namun demikian, tepi-tepinya terlalu lebar sehingga tidak menunjukkan dimana tepatnya tepi-tepinya. Untuk mengatasi hal tersebut arah tepi harus ditentukan.
6. Langkah selanjutnya adalah proses *Non-Maximum Supression*. Tujuan dari langkah ini adalah untuk mengubah tepi yang kabur pada citra gradien menjadi tepi yang tajam. Selanjutnya menentukan arah gradien terdekat sesuai dengan arah 8 ketetanggaan. Selanjutnya besar gradien piksel (x,y) dengan besar piksel pada titik dari dua arah yg ditentukan pada langkah sebelumnya. Jika besar gradien piksel (x,y) lebih besar dari kedua titik tadi, maka piksel tersebut dipertahankan. Tetapi jika gradiennya lebih kecil dari salah satu saja dari kedua titik tadi, maka piksel tersebut diubah menjadi 0.
7. Melakukan proses *hysteresis thresholding* dengan menggunakan dua nilai ambang yaitu *low threshold* dan *high threshold*. Sederhananya *hysteresis thresholding* bertujuan untuk klasifikasi dua buah nilai *high threshold* dan *low threshold* suatu piksel untuk dijadikan sebagai piksel tepi atau tidak jika nilainya lebih besar atau sama dengan ambang batas tersebut. Selanjutnya penentuan tepi dilakukan dengan cara mengganti pixel angka 0 atau 255 jika telah melewati syarat ambang batas yang ditentukan.
8. Menampilkan citra hasil deteksi tepi menggunakan metode *canny*
9. Deteksi tepi menggunakan metode *canny* selesai.

#### 4.1.2. Rancangan Pengenalan Pola



Gambar 5. Flowchart Pengenalan Pola

Penjelasan rincian dari masing-masing tahapan akan dijelaskan sebagai berikut :

1. Proses dimulai dengan melakukan *input* citra yang akan diuji ke sistem
2. Dilanjutkan proses ekstraksi fitur bentuk yang menghasilkan nilai piksel citra hasil deteksi tepi *canny*, yang mana *flowchart* ekstraksi fitur bentuk ditampilkan pada gambar 4.
3. Data nilai hasil ekstraksi fitur kemudian diuji kemiripan menggunakan metode *template matching* yang mana data masukan akan dibandingkan dengan masing-masing data. Proses uji kemiripan ini akan menghasilkan data berupa nilai korelasi kemiripan antara citra acuan dengan citra *input*.
4. Cek antara nilai kemiripan di poin 3. Nilai kemiripan tertinggi akan dibandingkan dengan nilai korelasi kemiripan yang telah ditetapkan yaitu 0,6.
5. Apabila nilai korelasi kemiripan  $\geq 0,6$ , maka program akan menghasilkan keluaran berupa informasi hasil pengenalan motif batik. Apabila nilai korelasi kemiripan  $< 0,6$  maka program akan mengeluarkan keluaran berupa informasi "Motif Tidak Dikenali".
6. Proses selesai.

## 5. IMPLEMENTASI DAN PENGUJIAN

Tahap implementasi sistem ini merupakan tahap awal untuk pengenalan motif batik. Terdapat 2 proses dalam melakukan pengenalan, yaitu proses ekstraksi fitur dan proses pencocokan pola. Citra yang digunakan dalam proses ekstraksi fitur dan proses pencocokan pola ini berupa citra warna (*true color*) dengan file berekstensi \*.jpg dan berukuran 200x200 piksel. Citra yang digunakan sebagai citra acuan di dalam penelitian ada 9 jenis motif batik, yaitu motif parang, ceplok blingon, liris cemeng, liris panjang madura, semen rante, semen sinom, lung-lungan bledakan, babon angram, dan buketan. Hasil ekstraksi fitur dan pengenalan motif batik tersebut ditampilkan pada aplikasi.

### 5.1.1. Pengujian Sistem untuk Motif Dikenali

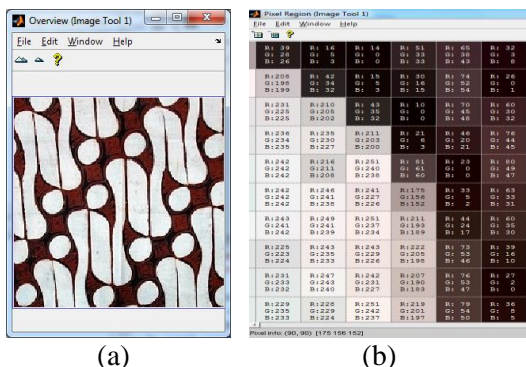
Citra yang akan digunakan untuk pengujian jenis motif dikenali adalah 9 buah citra motif batik. Proses dimulai dengan mengklik tombol 'Cari', kemudian pilih file citra yang akan diuji, Citra *input* kemudian melalui proses *grayscale* dan proses deteksi tepi *canny*. Proses ekstraksi fitur dijelaskan sebagai berikut:

#### 1. Input Citra

Citra yang dimasukkan berupa citra berwarna. Citra yang dijadikan citra uji, yaitu citra *parang\_uji.jpg* beserta matrik pikselnya dapat dilihat pada gambar 6(a) dan gambar 6(b)

Dari citra masukan motif parang diperoleh piksel dengan nilai sebagai berikut:

$$\begin{aligned} \text{parang\_uji}_{(89, 92)} &= [243 \ 229 \ 226] \\ \text{parang\_uji}_{(90, 90)} &= [175 \ 159 \ 152] \\ \text{parang\_uji}_{(92, 89)} &= [80 \ 49 \ 47] \end{aligned}$$

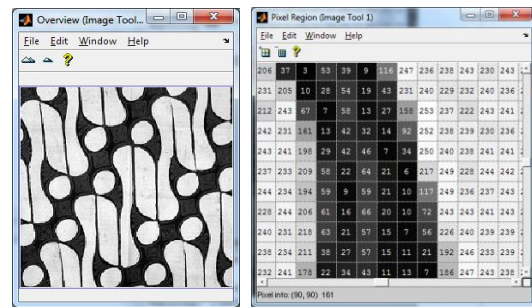


Gambar 6. citra *parang\_uji.jpg* beserta matrik pikselnya

#### 2. Konversi Grayscale

Citra motif batik yang berupa citra berwarna akan diubah menjadi citra berskala keabuan. Untuk mengkonversi citra berwarna menjadi citra berskala keabuan digunakan perhitungan manual sebagai berikut:

$$\begin{aligned} \text{parang\_uji}_{(90, 90)} &= [175 \ 159 \ 152] \\ &= (175 \times 0,29) + (159 \times 0,587) + \\ &\quad (152 \times 0,114) \\ &= 50,75 + 93,33 + 17,328 \\ &= 161,40 \end{aligned}$$



Gambar 7. citra berskala keabuan beserta matrik pikselnya

Hasil piksel yang diperoleh setelah proses konversi adalah

$$\begin{aligned} \text{parang\_uji\_gray}_{(89, 92)} &= 233 \\ \text{parang\_uji\_gray}_{(90, 90)} &= 161 \\ \text{parang\_uji\_gray}_{(92, 89)} &= 53 \end{aligned}$$

#### 3. Implementasi Filter Gaussian

Untuk mencegah kesalahan deteksi tepi karena *noise*, maka *noise* tersebut harus dikurangi. Hal ini dapat dilakukan dengan proses *smoothing* atau *filtering*. Pada deteksi tepi *canny*, proses ini menggunakan *filter gaussian*. Kernel gaussian yang digunakan adalah sebagai berikut:

$$\frac{1}{115} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Untuk mengimplementasikan *gaussian filtering* pada citra berskala keabuan dapat menggunakan ketetanggaan dari piksel yang akan diuji sehingga membentuk matriks 5x5. Matriks 5x5 dari pusat koordinat (90,90) pada citra berskala keabuan adalah:

$$\begin{bmatrix} 231 & 205 & 10 & 28 & 54 \\ 212 & 243 & 67 & 7 & 58 \\ 242 & 231 & 161 & 13 & 42 \\ 243 & 241 & 198 & 29 & 42 \\ 237 & 233 & 209 & 58 & 22 \end{bmatrix}$$

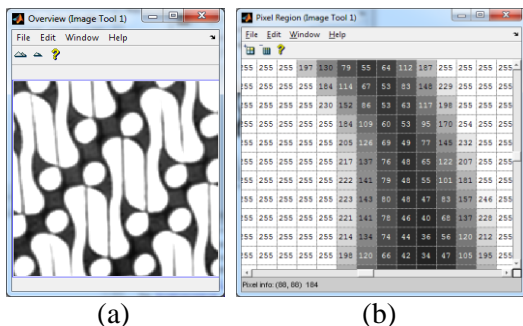


Perhitungan *gaussian filtering* adalah menerapkan konvolusi pada matriks citra dengan kernel gaussian. Tidak semua piksel dikenai konvolusi, yaitu baris dan kolom yang terletak di tepi citra. Hal ini disebabkan piksel yang berada pada tepi tidak memiliki tetangga yang lengkap sehingga rumus konvolusi tidak berlaku pada piksel seperti itu. Bagian yang tidak diproses tidak diikutkan dalam citra hasil. Akibatnya, ukuran citra hasil mengecil. Hasil konvolusi parang\_uji\_gray (90, 90) berada pada parang\_uji\_gaussian (88, 88). Nilai piksel hasil konvolusi berupa: parang\_uji\_gaussian (88, 88)  

$$= (237 \times 2 + 233 \times 4 + 209 \times 5 + 58 \times 4 + 22 \times 2 + 243 \times 4 + 241 \times 9 + 198 \times 12 + 29 \times 9 + 42 \times 4 + 242 \times 5 + 231 \times 12 + 161 \times 15 + 13 \times 12 + 42 \times 5 + 212 \times 4 + 243 \times 9 + 67 \times 12 + 7 \times 9 + 58 \times 4 + 231 \times 2 + 205 \times 4 + 10 \times 5 + 28 \times 4 + 54 \times 2) / 115$$
  

$$= 21122 / 115 = 183.669$$

Setelah citra diproses dengan konvolusi, perlu dilakukan pengaturan nilai piksel agar berada pada jangkauan [0 255]. Nilai yang kurang dari 0 diubah menjadi 0 dan yang melebihi 255 diubah menjadi 255. Gambar 7(a) dan gambar 7(b) menunjukkan hasil *gaussian filtering* dan matriks pikselnya



(a) (b)  
Gambar 7. Citra hasil *gaussian filtering* beserta matrik pikselnya

4. Menghitung Gradien Magnitude

Untuk menghitung *gradien magnitude* pada penelitian ini digunakan operator sobel. Kernel matriks sobel yang digunakan adalah:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Untuk menghitung *gradien magnitude* dari citra hasil *gaussian filtering* dapat menggunakan ketetanggaan dari piksel yang akan diuji sehingga membentuk matriks 3x3. Matriks 3x3 dari pusat koordinat (88,88) pada citra hasil *gaussian filtering* adalah:

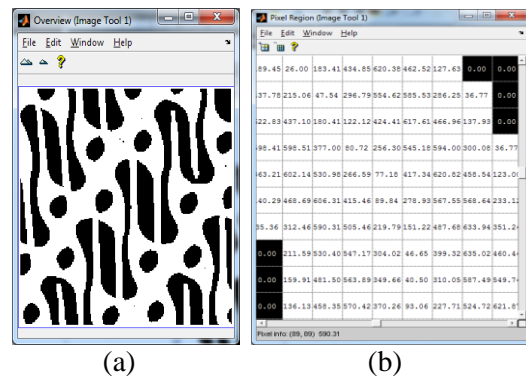
$$\begin{bmatrix} 230 & 152 & 86 \\ 255 & 184 & 109 \\ 255 & 205 & 126 \end{bmatrix}$$

Perhitungan manual *gradien magnitude* adalah dengan menerapkan konvolusi pada matriks citra hasil *gaussian filtering* dengan kernel matriks sobel. Penerapan konvolusi menyebabkan penambahan masing-masing satu piksel pada tepi matriks. Sehingga hasil konvolusi parang\_uji\_gaussian(88, 88) berada pada G (89, 89). Perhitungan nilai piksel hasil konvolusi berupa:

$$|G| = \sqrt{G_x^2 + G_y^2}$$
  

$$= \sqrt{(-565)^2 + (-171)^2}$$
  

$$= \sqrt{319.225 + 29.241} = 590,31$$



(a) (b)  
Gambar 7. Citra perhitungan *gradien magnitude* beserta matrik pikselnya

5. Menghitung Arah Tepi

Tahap selanjutnya adalah menghitung arah tepian dengan menggunakan  $G_x$  dan  $G_y$  pada proses perhitungan *gradien magnitude*. Arah tepi ditentukan dengan menggunakan persamaan:

$$\theta = \tan^{-1} \frac{S_y}{S_x}$$
  
 Perhitungan arah tepi dari G(89, 89) adalah:  

$$\theta = \tan^{-1} \frac{-171}{-565}$$
  

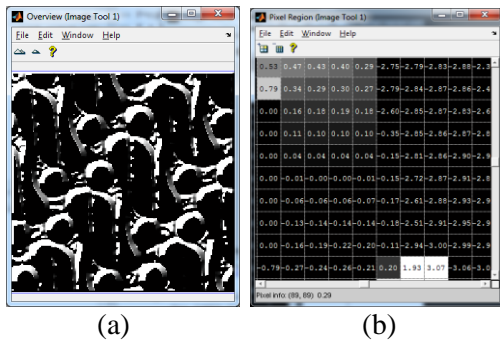
$$= \tan^{-1} (0,302)$$
  

$$= 0,29$$

Setelah arah tepi diperoleh, perlu menghubungkan antara arah tepi dengan sebuah arah yang dapat dilacak dari citra. Selanjutnya arah tepi yang diperoleh akan dimasukkan ke dalam salah satu kategori berikut:

- Jika  $\theta > 0$  dan  $\theta < 22,5$  atau  $\theta > 157,5$  dan  $\theta < -157,5$  maka diubah menjadi  $0^\circ$
- Jika  $\theta > 22,5$  dan  $\theta < 67,5$  atau  $\theta > -157,5$  dan  $\theta < -112,5$  maka diubah menjadi  $45^\circ$
- Jika  $\theta > 67,5$  dan  $\theta < 112,5$  atau  $\theta > -112,5$  dan  $\theta < -67,5$  maka diubah menjadi  $90^\circ$
- Jika  $\theta > 112,5$  dan  $\theta < 157,5$  atau  $\theta > -67,5$  dan  $\theta < -22,5$  maka diubah menjadi  $135^\circ$

Nilai  $\theta$  pada koordinat (89,89) adalah 0,29. Maka nilai  $\theta$  ini akan diubah menjadi  $0^\circ$ . Gambar 8(a). dan gambar 8(b) menunjukkan citra hasil perhitungan arah tepi dan matriks pikselnya

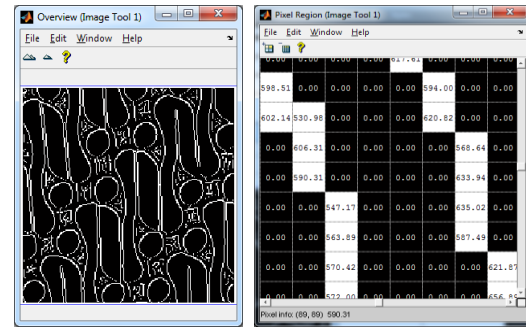


Gambar 8. Citra perhitungan arah tepi dan matrik pikselnya

#### 6. Non-maximum Suppression

Penghilangan nonmaksimum dilakukan di sepanjang tepi pada arah tepi dan menghilangkan piksel-piksel (piksel diatur menjadi 0) yang tidak dianggap sebagai tepi. Tujuan dari langkah ini adalah untuk mengubah tepi yang ‘kabur’ menjadi tepi yang ‘tajam’, lalu menentukan arah gradien terdekat sesuai dengan arah 8 ketetanggaan.

Misalnya arah tepi adalah 0. Apabila titik yang menjadi perhatian mempunyai koordinat (r,c), dua titik tetangga berupa (r, c-1) dan (r,c+1). Apabila gradien titik perhatian lebih besar daripada gradien kedua tetangga, nilainya akan dipertahankan. Sebaliknya, jika nilai titik perhatian lebih kecil daripada nilai salah satu atau kedua tetangga, nilainya akan diabaikan (diubah menjadi nol).



Gambar 9. Citra hasil perhitungan non-maximum suppression dan matrik pikselnya

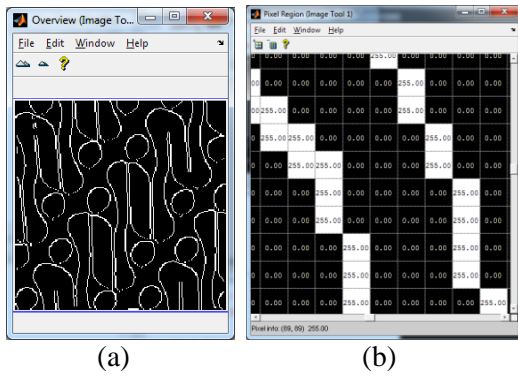
#### 7. Hysteresis Thresholding

Langkah terakhir adalah *hysteresis thresholding* atau klasifikasi tiap piksel apakah termasuk dalam kategori piksel tepi atau bukan. *Hysteresis thresholding* dilakukan dengan melibatkan dua ambang batas.

*Hysteresis thresholding* dilakukan dengan melibatkan ambang bawah dan ambang atas. Nilai ambang yang digunakan pada penelitian ini mengikuti nilai ambang yang umum digunakan untuk deteksi tepi canny. Ambang bawah yang digunakan adalah 0,05 dan ambang atas adalah 0,15. Pada *hysteresis thresholding*, nilai piksel yang kurang dari ambang bawah akan diubah menjadi hitam (nilai 0) dan nilai yang lebih dari ambang atas diubah menjadi putih (nilai 255). Untuk sementara nilai yang lebih dari atau sama ambang bawah tapi kurang dari ambang atas akan diberi nilai 128.

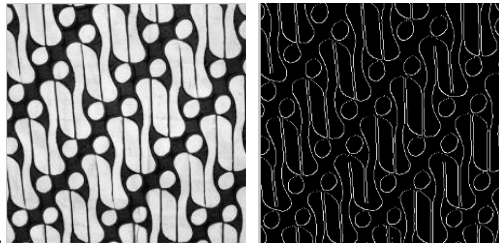
Selanjutnya, dilakukan pengujian kondisi 8 ketetanggaan. Apabila mendapatkan 8 piksel tetangga bernilai 225 maka piksel yang menjadi pusat perhatian akan diubah menjadi bernilai 225. Proses pengujian seperti itu dilakukan sampai tidak ada lagi perubahan dari nilai 128 menjadi 225. Tahap selanjutnya, semua piksel yang bernilai 128 dan memiliki kurang dari 8 piksel tetangga yang bernilai 225 akan diubah menjadi 0. Gambar 10(a) dan gambar 10(b) menunjukkan citra hasil perhitungan *hysteresis thresholding*, yakni hasil deteksi tepi canny dan matriks pikselnya.





Gambar 10. Citra hasil *hysteresis thresholding* dan matrik pikselnya

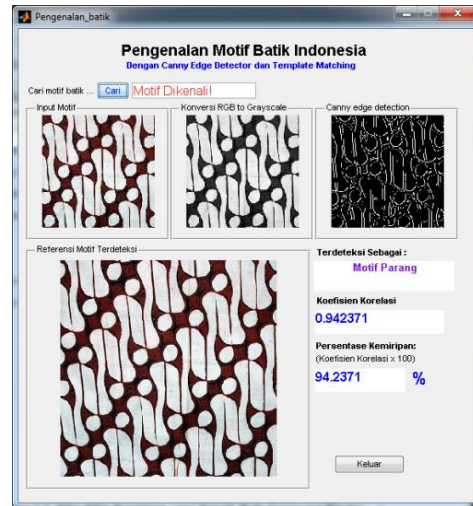
Hasil proses *grayscale* dan proses deteksi tepi *canny* untuk citra motif parang ditampilkan pada gambar 11. berikut.



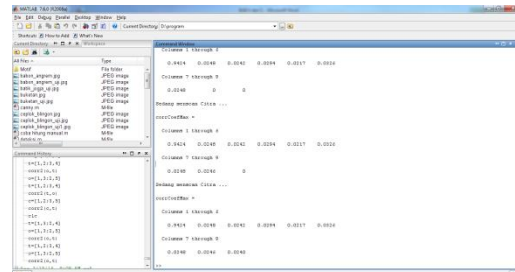
Gambar 11. (a) Hasil *Grayscale* (b) Hasil Deteksi Tepi Citra Batik Parang

Kemudian akan dihitung korelasi antara citra *input* dengan citra acuan. Hasil perhitungan korelasi akan ditampilkan pada ‘Koefisien Korelasi’ dan nama jenis batik yang dikenali akan ditampilkan pada ‘Terdeteksi Sebagai’. Sistem juga akan menampilkan citra motif batik acuan yang dikenali sebagai motif yang sama dengan citra *input* seperti pada gambar 12.

Pada gambar 13, dapat dilihat bahwa nilai kemiripan tertinggi adalah 0,942 yang merupakan hasil perhitungan koefisien korelasi antara citra acuan motif parang dengan citra uji. Nilai korelasi antara citra uji dengan citra-citra acuan yang lainnya lebih kecil dari nilai *threshold* korelasi yang telah ditetapkan. Citra *input* dapat dikenali sebagai citra yang sama dengan citra acuan yang telah ditetapkan.



Gambar 12. Tampilan Hasil Pengujian Motif Dikenali



Gambar 13. Nilai-Nilai Kemiripan yang Tampil Pada *Command Window*

Pada Gambar 13 dapat dilihat bahwa nilai kemiripan tertinggi adalah 0,942 yang merupakan hasil perhitungan koefisien korelasi antara citra acuan motif parang dengan citra uji. Nilai korelasi antara citra uji dengan citra-citra acuan yang lainnya lebih kecil dari nilai *threshold* korelasi yang telah ditetapkan. Citra *input* dapat dikenali sebagai citra yang sama dengan citra acuan yang telah ditetapkan.

### 5.1.2. Pengujian Sistem untuk Motif Tidak Dikenali

Citra yang akan digunakan untuk pengujian jenis motif dikenali adalah 3 buah citra selain citra yang sama dengan citra acuan.

Proses pengujian untuk ‘Motif Tidak Dikenali’ melewati tahap-tahap seperti yang telah dijelaskan pada pengujian sebelumnya. Setelah didapatkan hasil dengan presentase kemiripan tertinggi, yang mana tidak memenuhi syarat koefisien korelasi  $\geq 0,6$  maka

selanjutnya sistem akan menampilkan ‘Motif Tidak Dikenali’ serta menampilkan ‘Korelasi < Threshold’.



Gambar 14. Tampilan Hasil Pengujian Motif Tidak Dikenali

Hasil rekap pengujian citra dapat dilihat pada tabel 1.

Tabel 1. Rekap Hasil Pengujian Sistem Pengenalan Motif Batik

No	Citra Input	Hasil Pengujian	Persentase Kemiripan	Keberhasilan Pengenalan
1	Motif Parang	'Motif Parang'	92,4	Ya
2	Motif Ceplok Blingon	'Motif Ceplok Blingon'	93,1	Ya
3	Motif Liris Cemeng	'Motif Liris Cemeng'	88,8	Ya
4	Motif Liris Panjang Madura	'Motif Liris Panjang Madura'	89,2	Ya
5	Motif Semen Rante	'Motif Semen Rante'	88,5	Ya
6	Motif Semen Sinom	'Motif Semen Sinom'	92,1	Ya
7	Motif Lung-Lungan Bledakan	'Motif Lung-Lungan Bledakan'	92,8	Ya
8	Motif Babon Angrem	'Motif Babon Angrem'	88,1	Ya
9	Motif Buketan	'Motif Buketan'	80,2	Ya
10	Citra Ukiran Bebas	'Motif Tidak dikenali'	-	Ya
11	Citra Tenun Ikat	'Motif Tidak dikenali'	-	Ya
12	Citra Kain Songket	'Motif Tidak dikenali'	-	Ya

Persentase keberhasilan pengujian secara umum:

Dari 12 kali pengujian, terdapat 12 kali pengujian valid, dan tidak terdapat pengujian tidak valid. Namun hasil pengujian valid memiliki persentase kemiripan. Adapun rata-rata persentase kemiripan dari pengujian valid adalah:

$$\begin{aligned} & \% \text{ kemiripan pengujian} \\ & = \frac{92,4 + 93,1 + 88,8 + 89,2 + 88,5 + 92,1 + 92,8 + 88,1 + 80,2}{9} \times 100\% \\ & = \frac{805,2}{9} \times 100\% = 89,44\% \end{aligned}$$

## 6. KESIMPULAN DAN SARAN

### 6.1. Kesimpulan

Berdasarkan pengujian dan analisis terhadap sistem Pengenalan Jenis Batik Indonesia dengan Deteksi Tepi Canny dan *Template Matching*, dapat ditarik kesimpulan sebagai berikut :

1. Dari 9 motif batik yaitu motif parang, ceplok blingon, liris cemeng, liris panjang madura, semen rante, semen sinom, lung-lungan bledakan, babon angrem, dan buketan. ditambah 3 motif pengujian (selain 9 jenis objek batik Indonesia). Persentase ketepatan pengenalan adalah 89,44%.
2. Koefisien korelasi pada penelitian ini digunakan untuk menghitung nilai kemiripan citra uji dengan citra acuan. Apabila nilai koefisien korelasi mendekati 1, maka citra uji semakin sama (mirip) dengan citra acuan, dan apabila nilai koefisien korelasi mendekati -1, maka citra uji semakin berbeda (tidak mirip) dengan citra acuan.

Sesuai dengan persentase keberhasilan, deteksi tepi canny dan *template matching* dapat digunakan dalam mengenali motif batik.

### 6.2. Saran

Adapun saran untuk penelitian kedepannya agar lebih baik adalah sebagai berikut :

1. Penelitian pengenalan ini menggunakan metode canny dan *template matching*, pada penelitian selanjutnya dapat dicoba dengan menggunakan metode ekstraksi dan metode pengenalan yang berbeda.

2. Citra acuan dan citra uji yang digunakan dapat ditambah jumlahnya sehingga citra acuan dan citra uji benar-benar bisa mencerminkan ciri khas dari motif batik yang ada.

#### DAFTAR PUSTAKA

- [1] Wulandari, A. (2011). *Batik Nusantara (Makna Filosofi, Cara Pembuatan & Industri Batik)*. Yogyakarta: Andi.
- [2] Yodha, Y. W., & Kurniawan, A. W. (2013). *Perbandingan Penggunaan Deteksi Tepi Dengan Metode Laplace, Sobel, Prewit dan Canny Pada Pengenalan Pola*. Semarang: Program Studi Teknik Informatika Universitas Dian Nuswantoro.
- [3] Fahmi. (2007). Peancangan Algoritma Pengolahan Citra Mata Menjadi Citra Polar Iris Sebagai Bentuk Antara Sistem Biometrik. *Karya Ilmiah Teknik Elektro, Universitas Sumatra Utara Medan*.
- [4] Kadir, A., & Susanto, A. (2013). *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta: Andi.
- [5] Putra, D. (2010). *Pengolahan Citra Digital*. Yogyakarta: Andi Offset.
- [6] Arifin, & Budiman. (2009). *Edge Detection Menggunakan Metode Roberts' Cross*. Medan: STMIK Mikroskil.
- [7] Munir, R. (2004). *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: Informatika.
- [8] Bahri, R. S., & Maliki, I. (2012). *Perbandingan Algoritma Template Matching dan Feature Extraction Pada Optical Character Recognition*. Bandung: Teknik Informatika Universitas Komputer Indonesia.
- [9] Bowo, S. A., Hidayatno, A., & Isnanto, R. (2011). Analisis Deteksi Tepi Untuk Mengidentifikasi Pola Daun. Tesis Teknik UNDIP