

IMPLEMENTASI *CONTENT BASED IMAGE RETRIEVAL* (CBIR) PADA CITRA BATIK BESUREK YANG TIDAK UTUH MENGGUNAKAN METODE *SPEEDED UP ROBUST FEATURES* (SURF) DAN *FAST LIBRARY APPROXIMATED NEAREST NEIGHBOR* (FLANN)

Rezki Oksaputri¹, Ernawati², Desi Andreswari³

^{1,2,3}Program Studi Informatika, Fakultas Teknik, Universitas Bengkulu
Jl. WR. Supratman Kandang Limun Bengkulu 38371A INDONESIA
(telp: 0736-341022; fax: 0736-341022)

¹rezkioksaputri24@gmail.com

²ernawati@unib.ac.id

³desi.andreswari@unib.ac.id

Abstrak: Pengenalan citra dengan menggunakan citra batik besurek memiliki kelemahan yaitu kadang kala citra batik besurek yang ingin diproses merupakan citra yang tidak utuh atau tidak sempurna citra asli yang berada didalam database. Berdasarkan permasalahan tersebut, maka dilakukan penelitian untuk memproses temu kembali citra batik besurek yang tidak utuh menggunakan metode *speeded up robust features* (SURF) untuk proses ekstraksi *keypoint* pada citra dan *fast library approximated nearest neighbor* (FLANN) untuk proses *matching*-nya. Adapaun tujuan dari penelitian yang akan dilakukan adalah untuk mengetahui persentase citra yang dapat dikenali dengan menggunakan metode SURF dan FLANN. Pada penelitian ini digunakan 67 citra batik besurek dengan motif kaligrafi, raflesia, kaligrafi dan raflesia, burung kuau dan raflesia, burung kuau, dan rembulan. Kondisi citra yang digunakan pada pengujian adalah citra yang ditutupi dari 10% hingga 90% dan citra yang diambil dengan sudut pengambilan yang tidak baik. Pada penelitian ini digunakan nilai *recall*, *precision*, dan akurasi untuk mengetahui keberhasilan metode SURF dan FLANN dalam mengenali citra batik besurek yang tidak utuh. Penelitian ini mendapatkan hasil bahwa metode SURF dan FLANN dapat mengenali citra batik besurek yang ditutupi hingga 90% serta didapatkan nilai *recall* tertinggi adalah 100% pada motif rembulan, *precision* tertinggi adalah 78,82% pada motif kaligrafi dan raflesia, dan akurasi tertinggi adalah 86,1% pada motif rembulan.

Kata Kunci: batik besurek, *content based image retrieval* (CBIR), *speeded up robust fetures* (SURF), *fast library approximated nearest neighbor* (FLANN)

Abstract: Retrieval image by using an image of batik besurek have the weakness that sometimes batik besurek image that wants to be processed image that is not complete or not as perfect as original image residing in the database. The purpose of this research is to know the percentage of the image that can be recognized by using SURF and FLANN method. In this research used 67 images of batik besurek with motifs of calligraphy, raflesia, calligraphy and raflesia, burung kuau and raflesia, burung kuau,

and rembulan. Query image condition used in the test is the image covered from 10% to 90% as many as 54 images, the image taken with the angle of shooting is not good as much as 3 images, the image comes from a patchwork of 5 pieces, and the image taken with the camera application of 3 pieces. In this research used the value of recall, precision, and accuracy to determine the success of SURF and FLANN methods in recognizing the image of batik besurek that is complete. This research got the

result that SURF and FLANN method can recognize batik besurek image covered up to 90% and get the highest recall value is 100% at moon motif, the highest precision is 78,82% on motif of calligraphy and raflesia, and highest accuracy is 86,1% on the rembulan motif.

Keywords : *batik besurek, content based image retrieval (CBIR), speeded up robust fetures (SURF), fast library approximated nearest neighbor (FLANN)*

I. PENDAHULUAN

Batik besurek merupakan salah satu warisan budaya dari Kota Bengkulu yang dikenal sebagai Kota Gading Cempaka. Batik besurek memiliki karakteristik yang unik. Karakteristik batik besurek dilihat dari motif batik besurek itu sendiri, misalnya motif batik besurek motif burung kuau memiliki karakteristik yang berbeda dengan batik besurek motif kaligrafi. Dengan karakteristik ini dapat dibedakan jenis batik besurek. Pengenalan citra batik besurek dapat dilakukan dengan pengolahan citra digital dengan menggunakan karakteristik pada masing-masing batik besurek.

Namun, pengenalan citra batik besurek dengan melihat karakteristik citra ini memiliki kelemahan yaitu kadang kala citra batik besurek yang ingin diproses merupakan citra yang tidak utuh (bukan citra yang mengandung *noise*, gelap, atau silau), hal ini dapat disebabkan karena objek lain yang menutupi, pengambilan gambar yang tidak sempurna, atau keadaan objek itu sendiri yang mengalami perubahan (objek diperbesar, diperkecil, atau diputar). Hal ini menyebabkan karakteristik batik besurek yang ada pada citra batik besurek tersebut akan mengalami perubahan karena terdapat karakteristik yang hilang, sehingga dapat menyebabkan hasil pengenalan citra menggunakan citra batik besurek tersebut tidak sesuai dengan hasil yang diinginkan.

Untuk menghindari kelemahan ini, maka dilakukan penelitian yang berdasarkan pada

content based image retrieval (CBIR) dimana CBIR ini merupakan proses untuk melakukan pencarian citra digital pada suatu *database* citra. Yang dimaksud dengan "*Content Based*" di sini adalah: objek yang dianalisa dalam proses pencarian itu adalah *actual contents* (kandungan aktual) sebuah citra. Istilah konten pada konteks ini merujuk pada warna, bentuk, tekstur, atau informasi lain yang didapatkan dari citra tersebut [1].

II. LANDASAN TEORI

A. Batik Besurek

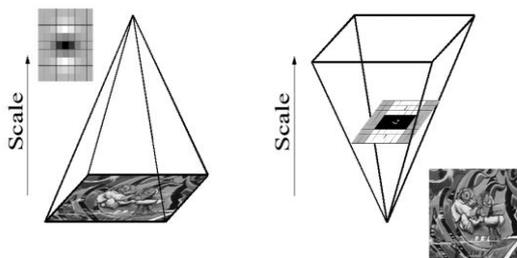
Provinsi Bengkulu merupakan salah satu kawasan yang berada di wilayah Sumatera. Nama Bengkulu sebagai wilayah penghasil batik belum terkenal secara luas. Namun, provinsi dengan ibu kota Bengkulu ini memiliki ciri khas batik yang tidak kalah mutunya dari batik hasil daerah lainnya. Batik Bengkulu tersebut bernama batik besurek [2].

Kain Batik Besurek adalah batik tulis tradisional khas Bengkulu yang termasuk batik pesisir dengan motif dominan kaligrafi Arab dihiasi perpaduan *flora* dan *fauna* yang sarat akan makna simbolis, melambangkan hubungan manusia dan alam dengan sang pencipta. Besurek (surat) berarti menulis atau melukis kaligrafi dan relief alam pada bidang kain, yang digunakan untuk kebutuhan sandang dalam tradisi masyarakat Bengkulu. Warna dasar yang dominan kain besurek adalah merah, biru, coklat dan kuning sesuai dengan kebutuhan dan penggunaannya [3]. Batik Besurek memiliki motif yang beragam. Mulai dari motif kaligrafi, bunga raflesia, burung kuau, relung paku, dan rembulan [4].

B. Speeded Up Robust Features (SURF)

Metode *Speeded Up Robust Features (SURF)* merupakan sebuah metode deteksi fitur yang menggunakan *keypoint* dari sebuah citra/gambar. *Keypoint* itu sendiri adalah bagian-bagian dari sebuah citra/gambar yang nilainya kuat/tetap ketika mengalami perubahan skala, rotasi, blurring, transformasi 3 dimensi, pencahayaan dan juga perubahan bentuk. Perubahan bentuk itu bisa terjadi karena bentuk citra data awal yang tidak utuh atau tidak sempurna gambar yang ada di dalam citra sampel tersebut. Citra uji yang tidak utuh mungkin karena ada objek lain yang menutupi, atau pengambilan gambar yang tidak sempurna, atau keadaan objek itu sendiri yang telah mengalami perubahan [5].

Ruang Skala (Scale Space)



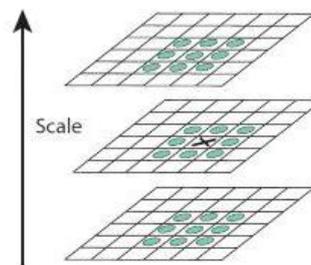
Gambar 1. Representasi Ruang Skala (Scale Space)

Dengan ukuran citra yang berbeda-beda, akan sangat sulit bagi kita untuk membandingkan fitur-fitur yang terdapat pada citra tersebut. Oleh sebab itu, diperlukan suatu proses yang menangani perbedaan ukuran dengan menggunakan metode perbandingan skala. Dalam metode ini, kita menggunakan ruang skala di mana citra diimplementasikan dalam bentuk sebuah *image pyramid* (Gambar 1.). Citra secara berulang akan diperhalus (*smoothing*) dengan fungsi Gaussian dan secara beruntun dengan cara subsampling untuk mencapai tingkat tertinggi pada piramida. Dengan menggunakan *integral image*, perhitungan

ini tidak perlu dilakukan secara iteratif dengan menggunakan filter yang sama, tetapi dapat filter dengan ukuran sembarang ke dalam beberapa skala citra yang berbeda [6].

Lokalisasi Keypoint

Untuk mencari lokasi *keypoint* yang invariant terhadap rotasi maka dilakukan proses yang terdiri dari tiga langkah, yaitu [7]: Menentukan nilai ambang (*threshold*) penentuan lokasi. Ketika nilai ambang dinaikkan, jumlah *keypoint* yang terdeteksi lebih kecil dan sebaliknya. Oleh karena itu, nilai ambang bisa disesuaikan untuk setiap aplikasi. Langkah selanjutnya adalah proses *non-maxima suppression* yaitu proses yang bertujuan untuk mencari sekumpulan calon *keypoint*. Proses ini dilakukan dengan membandingkan setiap pixel citra pada *scale space* dengan *26- Neighbors*, yang terdiri atas 8 titik di *scale* asli dan 9 di setiap *scale* atas dan bawahnya (total keseluruhan $9+8+9=26$ -Neighbors). Hasil dari *non-maxima suppression* adalah *keypoint* yang dihasilkan citra.



Gambar 2. Non Maxima Suppression

Proses terakhir adalah proses pencarian *keypoint* dengan menggunakan interpolasi dengan data yang dekat dengan *keypoint*. Ini dilakukan dengan mencocokkan *quadratic* 3 dimensi. untuk mendapatkan interpolasi data maksimal, maka digunakan pendekatan *Taylor Expansion* $H(x, y, \sigma)$ seperti pada persamaan 2.1. [8]:

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \quad (1)$$

Dengan x adalah vektor dan D memetakan x ke skalar. Lokasi ekstrim yang diinterpolasi, $\hat{x} = (x, y, s)$, ditemukan dengan mencari turunan dari fungsi diatas dan diberi nilai nol, sehingga [8]:

$$\hat{x} = -\frac{\partial^2 D^{-1} \partial D}{\partial x^2 \partial x} \quad (2)$$

Jika \hat{x} lebih besar dari 0.5 pada x, y atau arah σ , lokasi perlu disesuaikan lagi dengan melakukan interpolasi kembali. Prosedur ini diulang sampai nilai \hat{x} kurang dari 0.5. Bila interpolasi telah dilakukan beberapa kali tetapi hasilnya tidak memenuhi nilai dibawah 0.5, maka *keypoint* tersebut dihapus. *Keypoint* yang tersisa adalah yang stabil [8].

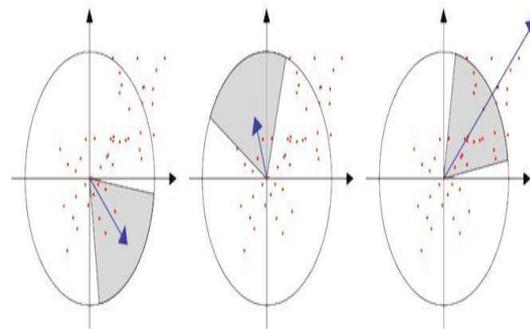
Deskriptor *Keypoint* dan Proses *Matching*

Deskriptor ditentukan berdasarkan daerah *pixel* sekitar *keypoint*. Deskriptor ini menggambarkan distribusi intensitas *pixel* tetangga disekitar *keypoint*, informasi gradien yang di ekstrak Pada SURF dihitung distribusi respon wavelet Haar order pertama dalam arah x dan arah y . Selain itu, SURF menggunakan integral gambar untuk meningkatkan kecepatan proses. Hal tersebut bertujuan untuk mengurangi waktu komputasi pada ekstraksi fitur dan pada proses *matching* fitur, yang juga telah terbukti meningkatkan *robust* [6].

Proses awal yang dilakukan adalah mencocokkan orientasi yang dihasilkan berdasarkan informasi dari daerah yang berbentuk lingkaran disekitar piksel yang menjadi *keypoint*. Kemudian membuat daerah berbentuk kotak pada orientasi yang terpilih dan mengekstrak deskriptor SURF dari daerah tersebut. Pada proses selanjutnya, proses *matching* fitur antara dua gambar dilakukan. Berikut ini dua langkah perhitungan deskriptor *keypoint*:

C. Pembuatan Orientasi

Pembuatan orientasi dilakukan dengan menghitung respon Haar wavelet dalam arah x dan arah y pada daerah *pixel* tetangga disekitar *keypoint* yang berbentuk lingkaran dengan jarak $6s$, dimana s adalah *scale* dari *keypoint* yang dideteksi. Ukuran wavelet ditentukan menjadi $4s$. Filtering dipercepat prosesnya menggunakan integral gambar, dan keluaran yang dihasilkan direpresentasikan sebagai titik-titik pada ruang dengan respon horizontal sepanjang sumbu axis dan juga respon vertikal sepanjang sumbu koordinat. Orientasi dominan dihitung menggunakan jumlah semua respon dalam pergeseran orientasi dengan ukuran *window* $\pi/3$ seperti pada Gambar 3, kemudian respon horizontal dan vertikal dalam *window* tersebut dijumlahkan. Dari kedua respon yang dijumlahkan sehingga menghasilkan sebuah vektor orientasi lokal. Vektor terpanjang inilah yang merupakan orientasi *keypoint* [9].



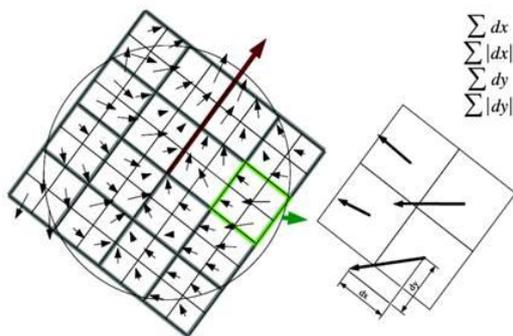
Gambar 3. Jendela Pergeseran Orientasi

D. Deskriptor Berdasarkan Jumlah Respn Haar Wavelet

Untuk proses ekstraksi deskriptor, langkah pertama yang dilakukan adalah membuat daerah kotak sekitar *keypoint*, dimana *keypoint* sebagai pusat dari daerah kotak tersebut, dan orientasinya di sekitar orientasi yang ditentukan. Berikut ini

merupakan gambar jendela orientasi perhitungan deskriptor pada skala yang berbeda [7] :

Pada tahap selanjutnya, respon wavelet dx dan dy dijumlahkan masing- masing sub-region. Selain memberikan informasi tentang polar dari perubahan intensitas, juga akan dihasilkan jumlah nilai absolut dari respon $|dx|$ dan $|dy|$. Masing-masing sub-region mempunyai empat dimensi deskriptor vektor v , yaitu dx , dy , $|dx|$ dan $|dy|$. Untuk 4×4 sub-region, maka panjang vektor deskriptornya adalah 64 sebagaimana di ilustrasikan pada Gambar 4 berikut ini [7]:



Gambar 4. Perhitungan Deskriptor

Sementara untuk mendapatkan hasil vektor v dapat dilihat pada persamaan berikut ini [7]:

$$v = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|, \right) \quad (3)$$

E. Fast Library Approximated Nearest Neighbor (FLANN)

Metode *Fast Library Approximated Nearest Neighbor* (FLANN) adalah sebuah *library* untuk melakukan pencarian cepat, perkiraan piksel tetangga, yang terdapat pada *space* dimensi yang tinggi. *Library* ini merupakan kumpulan algoritma yang bekerja dengan baik untuk menemukan nilai tetangga terdekat, sementara untuk hasil parameter yang optimal, tergantung pada kumpulan data yang digunakan [7].

Metode FLANN ini digunakan untuk *matching* fitur SURF. Fitur SURF ini terdiri dari *keypoint*

dan deskriptor yang berupa vektor. Untuk satu gambar di basis data terdapat banyak *cluster* untuk masing-masing fitur SURF. *Cluster* ini dibuat otomatis dengan *K- Nearest Neighbor* (KNN) menggunakan tipe indeks kd tree, dimana *K- Nearest Neighbor* (KNN) ini akan mencari jarak terkecil antara vektor fitur dengan vektor pada *cluster*. Proses pencocokan fitur pada gambar *query* dan fitur pada gambar dalam basis data, vektor *keypoint* dan deskriptor pada gambar *query* akan dicocokkan nilainya menggunakan KNN *search*.

KNN *search* akan mencari cluster pada basis data yang nilai vektor deskriptornya paling dekat jaraknya dengan vektor deskriptor pada gambar *query*. Setelah *cluster* didapat, kemudian akan dicari nilai vektor deskriptor pada *cluster* tersebut yang sama atau paling dekat dengan vektor deskriptor pada gambar *query*. Jika terdapat yang sama, maka ada satu *keypoint* yang cocok antara kedua citra tersebut. Selanjutnya, untuk masing-masing gambar akan ditarik suatu garis dari satu titik ke titik lainnya yang memiliki nilai *keypoint* yang cocok. [9].

Rasio Jarak

Banyak fitur dari gambar tidak memiliki kecocokan yang benar dengan citra batik besurek yang ada pada *database* hal ini terjadi karena gambar yang tidak baik atau karena tidak terdeteksi pada dengan citra batik besurek yang ada pada *database*. Oleh karena itu, sangat berguna untuk memiliki cara untuk membuang fitur yang tidak sesuai dengan citra *database*. Perkiraan yang lebih baik adalah dengan membandingkan jarak tetangga terdekat dengan tetangga terdekat kedua. Perkiraan ini memiliki performa yang baik karena kecocokan benar yang cocok harus memiliki tetangga yang secara

signifikan lebih dekat daripada kecocokan salah terdekat untuk mencapai pencocokan yang lebih baik. Pada penelitian David G. Lowe, yang mendapatkan hasil bahwa untuk kecocokan yang benar dan salah ditunjukkan dalam rasio yang terdekat dengan tetangga terdekat kedua dari masing-masing keypoint, penelitiannya menolak semua kecocokan di mana rasio jarak lebih besar dari 0,8 yang menghilangkan 90% kecocokan salah saat membuang kurang dari 5% dari kecocokan yang benar [8].

Jadi untuk deskriptor X , dicari deskriptor terdekat X_1 dan deksriptor terdekat X_2 rasio jarak dari $d(X, X_1)$ dan $0.8 * d(X, X_2)$ maka dapat diterima bahwa deskriptor tersebut memiliki kecocokan yang benar.

III. METODOLOGI

Pengujian pada penelitian ini menggunakan 67 citra batik besurek di dalam database dan 65 citra uji yang terdiri dari 56 citra batik besurek yang berasal dari 6 motif batik besurek yaitu motif kaligrafi, raflesia, kaligrafi dan raflesia, burung kuau, burung kuau dan raflesia, dan rembulan dimana setiap citra batik besurek ditutupi dari 10% hingga 90%, 5 citra yang berasal dari kain perca, 3 citra yang diambil dengan miring (pengambilan yang tidak sempurna) dan 3 citra yang diambil dari kamera aplikasi.

Digunakan nilai *recall*, *precision* dan akurasi untuk mengetahui tingkat keberhasilan aplikasi, *recall* digunakan untuk mengukur perbandingan jumlah citra uji relevan yang ditemukembalikan terhadap seluruh citra database yang relevan. Persamaan untuk mencari nilai *recall* dapat dilihat pada persamaan (4).

$$Recall = \frac{|Ra|}{|R|} \quad (4)$$

Precision digunakan untuk mengukur perbandingan citra uji relevan yang ditemukembalikan dengan banyak citra yang ditemukembalikan. Persamaan untuk mencari nilai *precision* dapat dilihat pada persamaan (5).

$$Precision = \frac{|Ra|}{|A|} \quad (5)$$

Akurasi merupakan tingkat kedekatan antara nilai prediksi dengan nilai aktual, yang dapat didefinisikan pada matriks *confusion* pada Tabel 1.

Tabel 1. Matriks *Confusion*

		Nilai Sebenarnya	
		TRUE	FALSE
Nilai Prediksi	TRUE	TP (True Positive) Hasil temu yang benar	FP (False Positive) Hasil Temu Yang salah
	FALSE	FN (False Negative) Hasil berkaitan yang tidak ditemukan	TN (True Negative) Hasil yang tidak berkaitan yang tidak ditemukan

Persamaan untuk mencari nilai *precision* dapat dilihat pada persamaan (6).

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

Keterangan :

Ra = citra yang relevan yang ditemukan

R = citra yang relevan dalam *database*

A = Jumlah citra yang ditemukembalikan

TP = True Positive

TN = True Negative

FP = False Positive

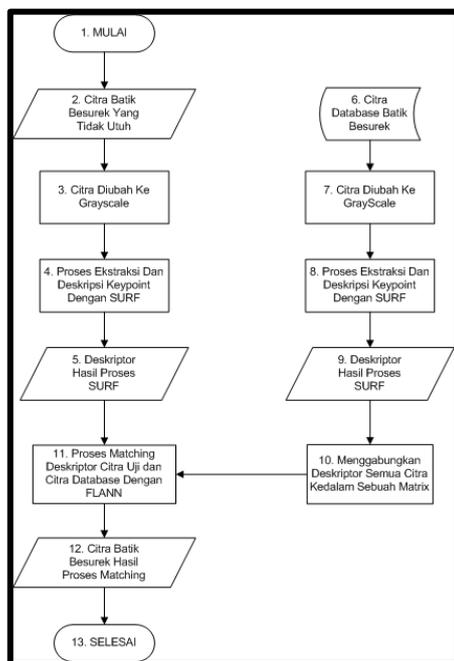
FN = False Negative

IV. ANALISIS DAN PERANCANGAN

Alur sistem merupakan analisis tahapan kerja sistem yang akan dibangun. Alur ini dimulai dari pengguna memasukkan *input*-an berupa citra batik besurek yang tidak utuh sampai dengan menghasilkan keluaran berupa citra batik besurek

hasil proses pengenalan. Secara garis besar alur sistem yang akan dibangun dapat dilihat pada Gambar 5.

Proses dimulai ketika menginputkan citra batik besurek, kemudian citra batik besurek tersebut akan diubah menjadi citra *grayscale*. Citra *grayscale* batik besurek tersebut akan diekstraksi *keypoint*-nya dengan metode SURF, hasil proses ini berupa deskriptor, yakni *keypoint* yang telah dideskripsikan. Citra batik besurek yang ada di dalam *database* akan diubah menjadi citra *grayscale* kemudian akan diekstraksi *keypoint*-nya dengan metode SURF. Hasil proses ekstraksi ini berupa deskriptor yang kemudian deskriptor pada setiap citra akan digabungkan kedalam sebuah matriks. Setelah itu akan dilakukan proses *matching* pada deskriptor citra uji dan deskriptor citra di dalam *database* dengan metode FLANN.



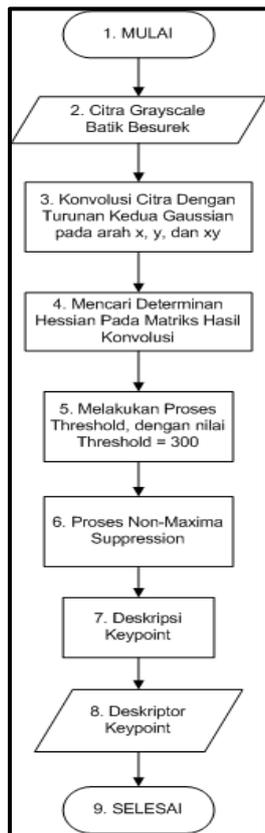
Gambar 5. Diagram Alir Sistem Temu Kembali Citra Batik Besurek Yang Tidak Utuh Menggunakan SURF

Berikut adalah diagram alir proses ekstraksi *keypoint* pada citra menggunakan metode *Speeded Up Robust Features* (SURF) yang dapat dilihat

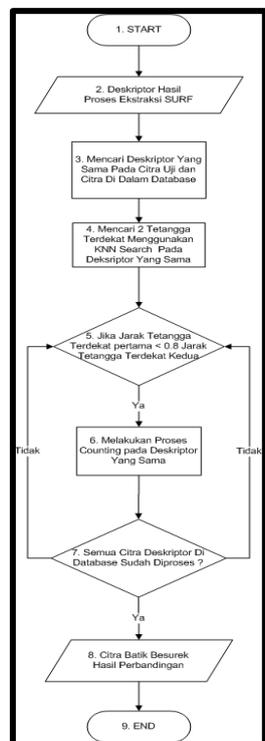
pada Gambar 6. Citra *grayscale* batik besurek akan dikonvolusi dengan turunan kedua Gaussian pada arah x , y , dan xy . Setelah itu akan dicari determinan hessian pada matriks hasil proses konvolusi sebelumnya, kemudian akan dilakukan proses *threshold* matriks hasil proses ini dengan nilai *threshold* yang digunakan adalah 300.

Setelah itu akan dilakukan proses *non-maxima suppression* yang bertujuan untuk mendapatkan calon *keypoint*, dimana pada calon *keypoint* ini akan dilakukan proses interpolasi untuk mendapatkan *keypoint*. *Keypoint* yang didapat selanjutnya akan dideskripsikan menjadi deskriptor berdasarkan jumlah dari *Haar Wavelet Responses*.

Gambar 7 merupakan diagram alir proses *matching* dengan menggunakan metode *Fast Library Approximated Nearest Neighbor* (FLANN). Proses awal dari proses *matching* dengan FLANN ini adalah dengan menginputkan deskriptor hasil proses ekstraksi sebelumnya dimana akan dicari deskriptor yang sama pada citra uji dan citra didalam database, kemudian akan dicari 2 tetangga terdekat pada deskriptor tersebut. Selanjutnya akan dilakukan proses *ratio distance* dimana jika jarak tetangga terdekat pertama 0,8 kali jarak tetangga terdekat kedua, maka deskriptor tersebut sama, dan akan dilakukan proses penambahan jumlah deskriptor yang sama untuk citra tersebut. Proses ini dilakukan untuk semua citra di dalam database. Setelah proses dilakukan untuk semua citra di dalam database, kemudian akan dilakukan perbandingan terhadap semua hasil deskriptor pada semua citra, dan akan ditampilkan citra yang memiliki nilai deskriptor yang sama terhadap citra uji yang paling besar.



Gambar 6. Diagram Alir Proses Ekstraksi Keypoint dengan SURF



Gambar 7. Diagram Alir Proses FLANN

V. PEMBAHASAN

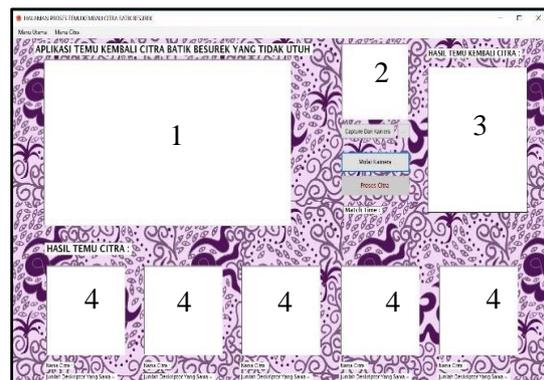
Adapun antarmuka untuk halaman utama aplikasi temu kembali citra batik besurek yang tidak utuh ini dapat dilihat pada Gambar 8.



Gambar 8. Halaman Utama Aplikasi

Aplikasi ini memiliki 2 halaman proses yaitu:

1. Halaman Proses Temu Kembali Citra:



Gambar 9. Halaman Proses Temu Kembali

Halaman proses temu kembali ini merupakan halaman dimana dilakukan proses temu kembali citra batik besurek yang tidak utuh. Adapun keterangan yang terdapat pada antarmuka pada Gambar 9. adalah:

1 = *picturebox* kamera.

2 = *picturebox* citra yang diinput (diproses menjadi citra *grayscale*).

3= Tabel menampilkan nama citra dan nilai deskriptor yang sama dari hasil temu kembali citra yang diinput.

4 = 5 hasil teratas dari perankingan hasil proses temu kembali.

2. Halaman Lihat Proses *Matching*

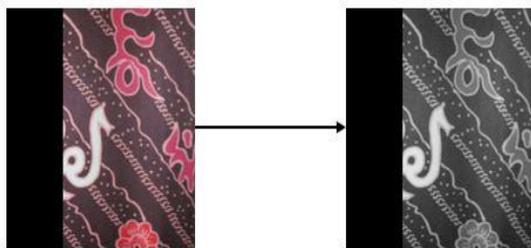


Gambar 10. Halaman Lihat Proses *Matching*

Halaman ini merupakan halaman untuk melihat hasil *matching* deskriptor antara citra uji dengan citra yang dipilih pada salah satu dari 5 citra teratas hasil proses temu kembali pada Gambar 10. Halaman ini akan menampilkan citra uji dan citra yang dipilih secara berdampingan dan akan ditarik garis untuk deskriptor yang sama pada kedua citra.

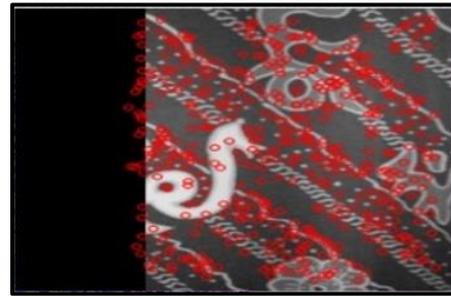
VI. HASIL PENGUJIAN

Pengujian ini diawali dengan menginputkan citra batik besurek yang tidak utuh dan merubah citra tersebut menjadi citra *grayscale* seperti pada Gambar 11.



Gambar 11. Citra diubah menjadi citra *grayscale*

Setelah diubah menjadi citra *grayscale*, akan dilakukan proses ekstraksi dan deskripsi *keypoint* pada citra dengan menggunakan metode *Speeded Up Robust Features* (SURF). Hasil proses ekstraksi *keypoint* pada citra dengan menggunakan metode SURF ini dapat dilihat pada Gambar 12.



Gambar 12. Hasil Ekstraksi *Keypoint* Citra

Kemudian, proses ekstraksi dan deskripsi *keypoint* juga akan dilakukan pada citra batik besurek di dalam *database*. Setelah itu akan dilakukan proses *matching* dengan menggunakan metode *Fast Library Approximated Nearest Neighbor* (FLANN), dimana parameter yang digunakan pada proses *matching* ini adalah *keypoint* yang telah dideskripsikan (deskriptor). Proses ini menggunakan *KNN-search* untuk mencari deskriptor yang sama, setelah itu dilakukan pencarian jarak terhadap 2 tetangga terdekat kemudian dilakukan proses rasio jarak terhadap jarak 2 tetangga terdekat tersebut. Digunakan nilai rasio = 0,8 berdasarkan penelitian yang dilakukan oleh David G. Lowe dimana penelitiannya menolak semua kecocokan di mana rasio jarak lebih besar dari 0,8 yang menghilangkan 90% kecocokan salah saat membuang kurang dari 5% dari kecocokan yang benar, jika nilai rasio ini terpenuhi, maka akan dilakukan proses penambahan pada nilai jumlah deskriptor yang sama pada citra tersebut.

Hasil dari proses *matching* ini adalah nilai deskriptor yang sama pada setiap citra batik besurek di dalam *database* terhadap citra yang diinputkan, kemudian akan dilakukan proses perankingan terhadap nilai deskriptor yang sama. Hasil proses temu kembali ini dapat dilihat pada Gambar 13.



Gambar 13. Hasil Proses Temu Kembali

Setelah itu, dapat dilihat deskriptor yang sama pada citra uji dan citra yang dipilih dari 5 citra teratas yang ditampilkan. Pada Gambar 14, merupakan contoh lihat proses *matching* dengan citra yang dipilih merupakan citra ranking pertama pada Gambar 14. Gambar 14 menunjukkan deskriptor yang sama pada citra uji dan citra yang terpilih. Setelah itu, akan dihitung nilai *recall*, *precision*, dan akurasi menggunakan persamaan 4, 5, dan 6.



Gambar 14. Lihat Proses *Matching* Untuk Citra Ranking Pertama

Dengan menggunakan contoh pengujian pada Gambar 14, maka diketahui:

$$R = 53$$

$$Ra = 16$$

$$A = 21$$

Sehingga nilai *recall* dan *precision*-nya :

$$Recall = \frac{16}{53} = 0,302$$

$$Precision = \frac{16}{21} = 0,762$$

Sedangkan untuk nilai akurasi, diketahui bahwa nilai :

$$TP = 16 ; FP = 5 ; FN = 37 ; TN = 9$$

Sehingga nilai akurasi adalah :

$$Akurasi = \frac{16 + 9}{16 + 5 + 37 + 9} = 0,37$$

$$= 0,37 \times 100 = 37\%$$

Nilai *matchtime* pada proses pengujian ini adalah 8930 ms = 8,93 detik.

Pengujian dilakukan dengan menggunakan 65 citra, dengan 54 citra merupakan citra batik besurek motif kaligrafi, raflesia, kaligrafi dan raflesia, burung kuau dan raflesia, burung kuau, dan rembulan yang ditutupi dari 10% hingga 90% dengan menggunakan *background* berwarna hitam. Kemudian 5 citra yang berasal dari kain perca dengan motif kaligrafi dan raflesia, 3 citra dengan motif kaligrafi dan raflesia yang diambil dengan sudut pengambilan yang tidak baik, sehingga menyebabkan motif menjadi miring kedalam. Kemudian 3 citra dengan motif kaligrafi dan raflesia yang diambil dengan menggunakan kamera aplikasi. Kondisi citra batik besurek berada di dalam *database* adalah utuh dan diambil dengan baik sehingga memperlihatkan keseluruhan motif yang ada pada citra batik besurek tersebut. Nilai *recall*, *precision*, dan akurasi masing-masing motif berdasarkan pengujian yang telah dilakukan dapat dilihat pada Tabel 2.

Tabel 2. Nilai *Recal*, *Precision*, dan Akurasi Masing-Masing Motif

No	Nama Motif	Recall	Precision	Akurasi
1	Kaligrafi	44,44%	17,74%	75,0%
2	Raflesia	77,78%	13,44%	76,8%
3	Kaligrafi dan Raflesia	44,25%	78,82%	47,0%
4	Burung Kuau	77,78%	13,36%	84,7%
5	Burung Kuau dan Raflesia	88,89%	16,67%	80,6%
6	Rembulan	100%	10,70%	86,1%

Pada pengujian ini diketahui bahwa metode *Speeded Up Robust Features* dan *Fast Library Approximated Nearest Neighbor* (FLANN) dapat mengenali citra batik besurek dimana citra batik besurek yang ditutupi dari 10% hingga 10%.

VII. KESIMPULAN

Berdasarkan pengujian yang telah dilakukan, maka dapat disimpulkan bahwa:

1. Metode *Speeded Up Robust Features* (SURF) dan *Fast Library Approximated Nearest Neighbor* (FLANN) dapat mengenali citra yang tidak utuh yang ditutupi 10% hingga 90%.
2. Persentase rata-rata nilai *recall* dan *precision* adalah *recall* = 55,5% dan *precision* = 66,48%. Waktu pengenalan citra batik besurek yang tidak utuh pada aplikasi ini adalah 7,767 detik.
3. Nilai *recall* terbesar dimiliki motif burung kua dan raflesia yaitu sebesar 88,89% dan nilai *precision* terbesar dimiliki oleh motif kaligrafi dan raflesia yaitu 78.82%.

VIII. SARAN

Berdasarkan analisis yang telah dilakukan pada penelitian ini, terdapat beberapa saran yang dapat diberikan untuk pengembangan penelitian ini. Berikut saran yang dapat diberikan:

1. Pengembangan sistem yang selanjutnya diharapkan mampu untuk menyeleksi deskriptor yang cocok pada citra uji dan citra batik besurek yang ada di dalam *database* seperti dengan menggunakan metode RANSAC (*Random Sample Consensus*)

sehingga mendapatkan hasil pengenalan yang lebih baik.

2. Sebaiknya deskriptor pada citra batik besurek disimpan ke dalam *database* sehingga dapat mengurangi waktu proses pencocokan.

REFERENSI

- [1] A. Sumarna, "CBIR Berdasarkan Ekstraksi Fitur Warna Menggunakan Java," 2010.
- [2] Y. Brasilka, Ernawati and D. Andreswari, "Klasifikasi Citra Batik Besurek Berdasarkan Ekstraksi Fitur Tekstur Menggunakan Jaringan Syaraf Tiruan Self Organizing Map (SOM)," *Jurnal Rekursif*, vol. 3, pp. 133-145, 2015
- [3] D. B. Magfira, Ernawati and D. Andreswari, "Aplikasi Peningkatan Resolusi Citra Motif Batik Menggunakan Metode Interpolasi Spline Kuadratik (Studi Kasus: Citra Motif Batik Besurek Kota Bengkulu)," *Jurnal Rekursif*, vol. 3, pp. 123-131, 2015.
- [4] Ernawati, K. Anggriani, F. F. Coastera and P. Primayana, "Deteksi Tepi Citra Batik Besurek Motif Gabungan Menggunakan Fuzzy Inferences System (Fis)," in *Seminar Nasional Teknologi Informasi 2017*, 2017.
- [5] F. A. Lubis, H. Sunandar, G. L. Ginting and L. T. Sianturi, "Implementasi Metode Speed Up Features Dalam Mendeteksi Wajah," vol. 3, pp. 22-27, 2016.
- [6] H. Kusumo, "Aplikasi Pengenalan Pola Citra Tanda Tangan Dengan Metode Speeded-Up Robust Features (SURF) Pada Single Board Computer Beaglebone Black," Skripsi, 2015.
- [7] F. Yansyah, "Deteksi Wajah Menggunakan Metode Speed-Up Robust Features (SURF)," Skripsi, 2014.
- [8] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, pp. 1-28, 2004.
- [9] K. D. Tania, "Pengenalan Gambar Menggunakan Sebagian Data Gambar," *Jurnal Generic*, vol. 5, pp. 12-14, 2010.