

SISTEM TEMU KEMBALI CITRA BATIK BERDASARKAN KARAKTERISTIK TEPI DAN WARNA

Diana Ayu Windari, Fajar Astuti Hermawati*

Teknik Informatika, Fakultas Teknik, Universitas 17 Agustus 1945 Surabaya

Email: *fajarastuti@untag-sby.ac.id

ABSTRAK

Batik merupakan warisan kebudayaan Indonesia yang perlu dijaga, salah satunya dapat memanfaatkan teknologi dengan menggunakan citra batik. Content Based Image Retrieval (CBIR) merupakan teknik pencarian citra dengan memasukkan informasi berupa citra query yang diinginkan oleh pengguna terhadap beberapa data citra dengan memanfaatkan database. Teknik CBIR didasarkan pada perbandingan warna, bentuk, serta tekstur citra. Dalam sistem temu kembali citra diperlukan pengekstraksian karakteristik dari citra. Karakteristik yang akan di ekstrak dalam citra batik adalah tepi dan warna yang merupakan karakteristik utama batik. Tidak hanya itu saja, pada sistem ini akan memanfaatkan teknik CBIR menggunakan metode LoG (Laplacian of Gaussian) dan Law Texture. Sistem temu kembali citra batik berdasarkan karakteristik tepi dan warna ini bertujuan untuk memudahkan pengguna dalam menemukan kembali citra batik di dalam komputer dengan karakteristik yang pengguna inginkan. Sistem akan melakukan perbandingan antara citra acuan yang pengguna pilih dengan berbagai macam citra batik yang pengguna miliki

Kata Kunci: Citra Batik, Content Based Image Retrieval (CBIR), Sistem Temu Kembali Citra.

1. Pendahuluan

Tak sedikit orang yang memanfaatkan teknologi untuk mengembangkan batik. Salah satunya dengan memanfaatkan citra batik. Pemanfaatan citra batik dilakukan karena lebih efisien serta dapat mengatasi beberapa kendala yang muncul, salah satunya adalah banyaknya batik di Indonesia. Oleh karena itu peran teknologi sangat dibutuhkan dengan memanfaatkan citra batik yang telah di ambil dengan perekam digital. Citra batik tersebut dapat disimpan kedalam komputer, sehingga dapat dipergunakan sewaktu-waktu. Tetapi kadang kala user mengalami kesulitan dalam mencari gambar batik yang telah disimpan dalam komputer. Melihat kondisi ini

diperlukan aplikasi yang dapat mempermudah pengguna dalam pencarian citra batik yang terdapat di komputer.

Content Based Image Retrieval (CBIR) merupakan teknik pencarian citra dengan memasukkan informasi berupa citra acuan (*query*) yang diinginkan oleh pengguna terhadap beberapa data citra dengan memanfaatkan basisdata [1]–[3]. Teknik CBIR didasarkan pada perbandingan antara citra acuan dengan citra yang terdapat pada basisdata. Teknik ini akan menyeleksi antara citra acuan dan citra lain dengan memanfaatkan informasi dari setiap citra, karena setiap citra memiliki karakteristik yang berbeda-beda. Walaupun sebuah citra terlihat

mirip namun nilai yang terkandung pada citra tersebut bisa sangat berbeda. Perbandingan antar citra bisa memanfaatkan warna, bentuk, serta tekstur yang dimiliki citra tersebut.

Pada sistem ini akan digunakan metode LoG (*Laplacian of Gaussian*) untuk mendeteksi tepi citra dan Law Texture untuk mendeteksi tekstur citra. Hasil akhir sistem akan menampilkan delapan citra dimana delapan citra tersebut memiliki nilai kesalahan terkecil.







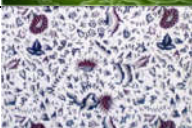

2. Tinjauan Pustaka

Penelitian tentang sistem temu kembali citra berbasis isi atau disebut dengan Content Based Image Retrieval (CBIR) sudah banyak dilakukan. Penelitian Hermawati [4] menerapkan teknik color-edge extraction yang merupakan kombinasi antara fitur warna dan tepi untuk sistem temu kembali citra berbasis isi. Pada kasus yang lain, Hermawati dkk mengajukan kombinasi fitur warna dan tektur berbasis transformasi wavelet pada ruang warna tertentu untuk mengambil kembali citra berbasis region [5]–[7].

Penelitian CBIR yang diimplementasikan untuk temu kembali citra batik juga banyak dilakukan. Dalam penelitiannya, Arwanda dan Agani [2] menggunakan ekstraksi ciri berbasis tekstur dengan menerapkan filter Gabor wavelet 2D. Sedangkan Wicaksono dkk [3] menerapkan transformasi Curvelet pada ruang warna HSV untuk mengekstraksi fitur *color-texture* pada citra batik. Sama dengan penelitian sebelumnya, Yumarlin dkk [8] juga menerapkan transformasi Curvelet pada ruang warna HSV untuk mendapatkan fitur pada citra batik dengan kekhususan citra batik pesisir.

Fitur *color-texture* berbeda digunakan oleh Nasir dkk [9] untuk mengekstraksi fitur pada citra kain tradisional yaitu kombinasi antara *local binary pattern* dengan ruang warna HSV. Fitur berbeda diterapkan oleh [1] yang menerapkan metode *speeded up robust features* (SURF) untuk proses ekstraksi titik pada citra batik besurek yang tidak utuh.

Tabel 1. Citra Yang Digunakan Dalam Penelitian

Citra Acuan	Jumlah Citra Dalam Database
	10
	10
	10
	10
	10
	10
	10
	10

3. Metode

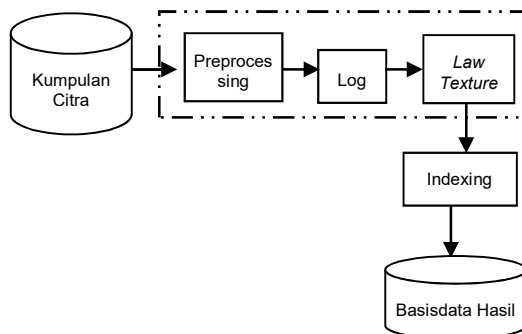
3.1 Data

Data yang digunakan dalam penelitian ini adalah potongan citra

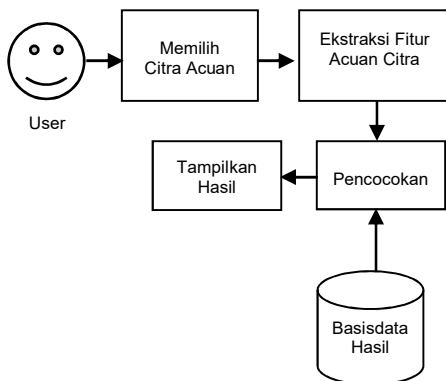
batik dengan berbagai corak dan warna, seperti yang tersaji pada Tabel 1.

3.2 Perancangan Sistem

Gambar 1 merupakan gambaran sistem secara garis besar. Proses dimulai dari tahap preprocessing kemudian diikuti dengan ekstraksi fitur tekstur dalam ruang warna yang digunakan. Fitur yang dihasilkan kemudian diindeks dan disimpan dalam basis data fitur.



Gambar 1. Blok Diagram Sistem



Gambar 2. Blok Diagram Tahap Retrieval

Pada tahap pengambilan kembali (*retrieval*), seperti terlihat pada Gambar 2, user memilih citra acuan yang diinginkan, yang kemudian diekstraksi fiturnya dan dicocokkan dengan fitur yang ada dalam basisdata. Hasil pencocokan ditampilkan mulai

dari yang mempunyai nilai kesamaan paling tinggi.

3.2.1. Preprocessing

Semakin besar ukuran citra maka semakin lama pula waktu yang dibutuhkan untuk memproses citra tersebut. Oleh karena itu dalam tahap ini akan dilakukan pengecilan ukuran yang diharapkan dapat menekan waktu yang dibutuhkan untuk memproses citra tersebut. Proses pengecilan ini memanfaatkan toolbox yang ada pada matlab yaitu fungsi *imresize*.

3.2.2. Ekstraksi Fitur Tepi dan Warna

Proses ini dilakukan untuk mendapatkan karakteristik tepi dan warna citra melalui beberapa proses. Ekstraksi fitur tepi dan warna dilakukan pada citra acuan dan pada kumpulan citra. Sedangkan untuk hasil dari ekstraksi pada kumpulan citra akan disimpan pada basisdata hasil yang kemudian akan digunakan kembali untuk proses pencocokan dengan hasil ekstraksi pada citra acuan.

Adapun tahapan algoritma ekstraksi fitur tepi dan warna adalah sebagai berikut:

1. Masukkan berupa I sebagai citra dan n untuk jumlah citra.
2. Mengubah citra menjadi bernilai double dengan menggunakan toolbox matlab, $p7=double(p6)$;
3. Mengubah citra rgb menjadi hsv. $p8=rgb2hsv(p7)$;
4. Memisahkan nilai h , s dan v citra. $h=p8(:,:,1)$; $s=p8(:,:,2)$; $v=p8(:,:,3)$;
5. Mengekstrak tepi citra dengan teknik *Laplacian of Gaussian* (LoG) menggunakan nilai v . $vv=edge(v,'log')$;
6. Selesai

3.2.3. Segmentasi Gambar dengan *Law Texture*

Law Texture [4][5] merupakan metode untuk memperoleh fitur tekstur pada citra. Pada *Law Texture* ini menggunakan mask local untuk mendeteksi berbagai jenis tekstur. Mask kovolusi 5x5 digunakan untuk menghitung nilai dari tekstur. Berikut adalah dimensi konvolusi filter :

$$\begin{aligned} L5 \text{ (LEVEL)} &= [1 \ 4 \ 6 \ 4 \ 1] \\ E5 \text{ (EDGE)} &= [-1 \ -2 \ 0 \ 2 \ 1] \\ S5 \text{ (SPOT)} &= [-1 \ 0 \ 2 \ 0 \ -1] \\ R5 \text{ (RIPPLE)} &= [1 \ -4 \ 6 \ -4 \ 1] \\ W5 \text{ (WAVE)} &= [-1 \ 2 \ 0 \ -2 \ 1] \end{aligned}$$

Mask diatas dapat dikombinasikan dengan *mask* yang lain, hasil dari kombinasi antar *mask* akan digunakan untuk proses pemfilteran. Dibawah ini adalah contoh kombinasi antara E5 dan L5, dimana E5 ditransposisikan terlebih dahulu baru kemudian dikalikan dengan L5 :

$$\begin{bmatrix} -1 \\ -2 \\ 0 \\ 2 \\ 1 \end{bmatrix} \times [1 \ 4 \ 6 \ 4 \ 1] = \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Hasil dari perkalian mask akan difilterkan pada citra. Dalam melakukan proses pemfilteran digunakan toolbox yang ada pada matlab yaitu fungsi *imfilter*.

3.2.4. Pengindeksan

Pengindeksan ini akan menggunakan informasi pelabelan pada proses sebelumnya. Masing-masing citra batik akan menghasilkan nilai data atau piksel yang berbeda. Untuk setiap batik, sistem akan memberikan suatu indeks dimana indeks tersebut diperoleh dari piksel yang membentuk citra batik tersebut.

Hasil dari pengindeksan akan disimpan dalam basisdata hasil.

3.2.5. Tahap Retrieval

Dengan memanfaatkan GUI (*Graphical User Interface*) [8] [9] yang terdapat pada matlab, pengguna akan melakukan pemilihan terhadap citra batik sebagai acuan yang diinginkan. Selanjutnya, citra batik acuan yang telah terpilih tersebut diekstraksi dengan menerapkan metode Log (*Laplacian of Gaussian*) dan *Law Texture*.

Hasil ekstraksi yang diperoleh nantinya akan digunakan sebagai kunci acuan secara keseluruhan dari proses yang terdapat pada sistem ini. Hasil yang didapatkan kemudian akan dihitung kecocokannya dengan fitur indeks dalam basisdata hasil.

Pencocokan akan menghasilkan suatu nilai, nilai tersebut adalah nilai kesalahan. Nilai kesalahan ini nantinya akan dicari yang paling kecil. Karena nilai kesalahan yang didapatkan adalah acak maka nilai kesalahan hasil dari perbandingan tersebut akan diurutkan dari yang paling kecil.

4. Hasil dan Pembahasan

Pengujian dilakukan untuk mendapatkan informasi tentang performa dari sistem yang telah dibuat, serta menemukan besarnya nilai keefektifan sistem tersebut. Semakin efektif suatu sistem maka nilai keefektifan juga semakin besar. Dalam menemukan nilai keefektifan sistem temu kembali citra batik ini akan dipergunakan rumus *precision*.

Namun sebelum itu, sistem akan diuji terlebih dahulu yang dikaitkan dengan beberapa pertanyaan tentang keefektifan ekstraksi pada sistem agar pengujian lebih terarah. Pertanyaan-pertanyaan tersebut antara lain :

1. Berapa keefektifan ekstraksi citra batik berdasarkan warna ?
2. Berapa keefektifan ekstraksi citra batik berdasarkan tepi ?
3. Berapa keefektifan ekstraksi citra batik berdasarkan tekstur dengan metode *Law Texture* ?

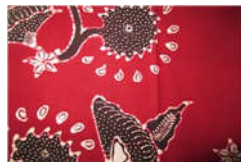
Untuk menjawab pertanyaan diatas pengujian akan dibagi menjadi tiga bagian. Pengujian dilakukan untuk mengukur kemampuan sistem untuk mendapatkan kembali (*retrieve*) dokumen-dokumen yang relevan.

Pengukuran keefektifan atau akurasi hasil dari proses akan menggunakan *precision*, yang dirumuskan sebagai berikut [7]:

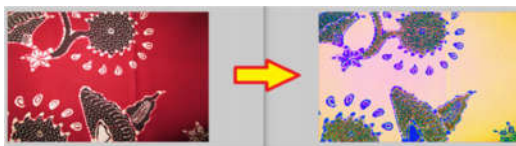
$$Precision = \frac{\text{Jumlah dokumen relevan yang diretrieve}}{\text{Total dokumen yang diretrieve}} \quad (1)$$

4.1. Pengujian Keefektifan Ekstraksi Warna Citra Batik

Gambar 3 merupakan citra batik yang dipilih untuk pengamatan terhadap hasil dari proses dalam sistem. Citra batik tersebut dipilih karena memiliki warna, bentuk dan tekstur yang mudah untuk diamati. Pemilihan citra yang tepat diharapkan dapat menunjang atau memberikan hasil yang lebih optimal pada penelitian ini. Gambar 4 adalah contoh citra batik RGB yang dirubah menjadi HSV.

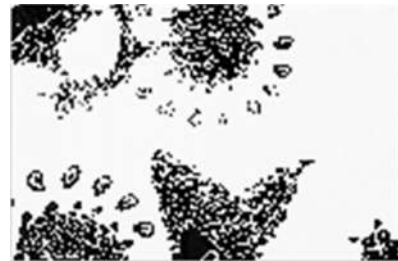


Gambar 3. Batik Merah

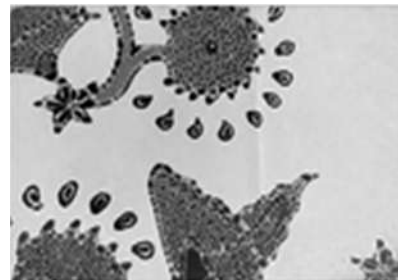


Gambar 4. Perubahan Citra RGB menjadi HSV

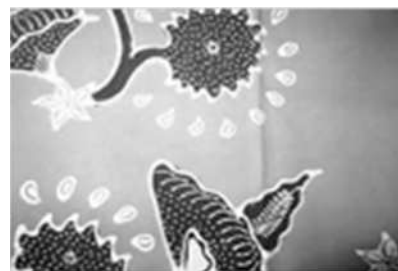
HSV cukup ideal bila dibandingkan RGB untuk digunakan didalam sistem ini. H (*Hue*) menyatakan warna, S (*Saturation*) menyatakan kekuatan warna tersebut, dan V (*Value*) menyatakan kecerahan. Bila ditunjukkan dengan gambar maka akan terlihat seperti Gambar 5 untuk hasil nilai H, Gambar 6 untuk hasil nilai S, dan Gambar 7 untuk hasil nilai V.



Gambar 5. Hasil Hue Batik Merah



Gambar 6. Hasil Saturation Batik Merah



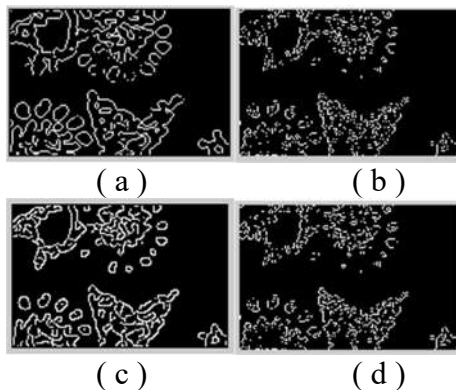
Gambar 7. Hasil Value Batik Merah

4.2. Pengujian Keefektifan Ekstraksi Tepi Citra Batik

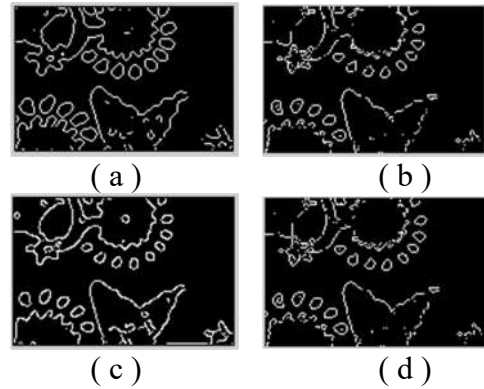
Pengujian kali ini mengenai keefektifan ekstraksi tepi citra batik. Dari penjelasan tersebut dapat disimpulkan bahwa metode pendeteksi tepi sangat berpengaruh terhadap keefektifan dari sistem ini, oleh karena

itu penentuan metode yang akan digunakan pada sistem merupakan hal yang penting untuk memperoleh hasil yang maksimal. Pengujian ini dilakukan dengan menggunakan metode Log (*Laplacian of Gaussian*), *prewitt*, *canny*, dan *sobel*. Metode-metode tersebut akan digunakan pada pengujian ini dan dicari metode yang memiliki nilai *precision* terbaik. Pengujian akan memanfaatkan nilai H (*Hue*), S (*Saturation*), dan V (*Value*) dari citra batik. Gambar 8, 9 dan 10 berturut-turut menunjukkan hasil deteksi tepi pada ruang warna H, S dan V dengan menggunakan metode LoG, *prewitt*, *canny* dan *sobel*.

Secara visual metode *Log* lebih baik dari pada metode lain. Pada Gambar 8(c) yang menggunakan metode *canny*, tepi batasnya terlihat jelas dan terlihat sedikit lebih tebal namun terdapat tepi yang hilang, sedangkan hasil dari metode LoG pada Gambar 8(a) semua tepi berhasil terdeteksi namun garis tepinya tidak setebal *canny*.

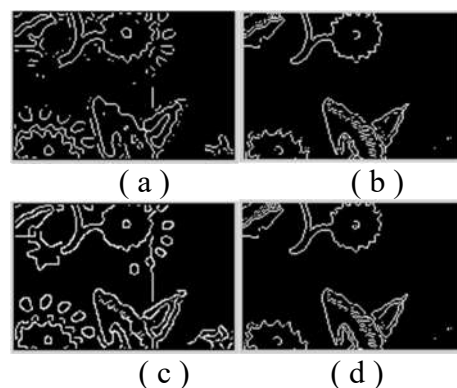


Gambar 8. Hasil deteksi tepi pada ruang warna H (*Hue*) dengan metode (a) Log (b) *Prewitt* (c) *Canny* (d) *Sobel*



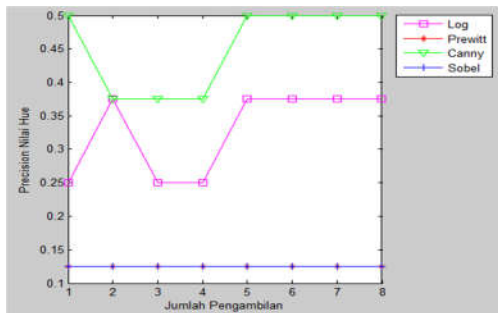
Gambar 9. Hasil deteksi tepi pada ruang warna S (*Saturation*) dengan metode (a) Log (b) *Prewitt* (c) *Canny* (d) *Sobel*

Pada hasil visual untuk ruang warna S semua gambar menunjukkan hasil yang cukup baik dan cukup jelas. Ini membuktikan bahwa penggunaan S (*Saturation*) lebih baik daripada H (*Hue*). Dari empat gambar tersebut yang memiliki garis tepi paling jelas dan bagus adalah Gambar 9(c) yang menggunakan metode *canny*. Sedangkan untuk metode LoG terdapat garis tepi yang hilang yang terdapat ditengah bunga. Lalu gambar 9(b) yang menggunakan metode *prewitt* dan Gambar 9(d) yang menggunakan metode *sobel* menunjukkan hasil yang serupa.



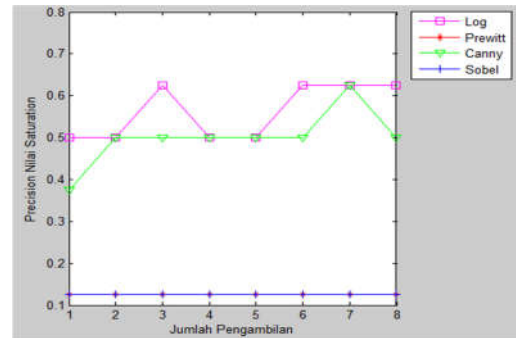
Gambar 10. Hasil deteksi tepi pada ruang warna V (*Value*) dengan metode (a) Log (b) *Prewitt* (c) *Canny* (d) *Sobel*

Pada hasil visual hasil deteksi tepi pada ruang warna V, semua gambar menunjukkan hasil yang cukup berbeda satu dengan yang lain. Metode *canny* tetap menghasilkan garis tepi paling tebal dan jelas serta terlihat lebih baik dari yang lainnya. Terdapat beberapa garis tepi yang hilang pada Gambar 10(a) yang menggunakan metode LoG, sehingga terlihat tidak utuh. Sedangkan pada Gambar 10(b) yang menggunakan metode *prewitt* dan Gambar 10(d) yang menggunakan metode *sobel* menunjukkan hasil yang serupa dan tidak begitu detail.



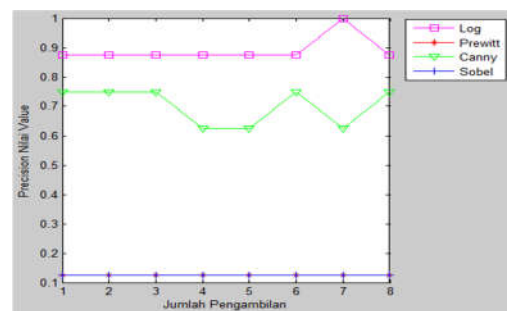
Gambar 11. Grafik *Precision Nilai H (Hue)*

Dari grafik pada Gambar 11 dapat disimpulkan bahwa metode terbaik untuk mendeteksi tepi dari nilai H (*Hue*) adalah dengan menggunakan metode *canny*. Metode *canny* memberikan nilai yang terbaik sebesar 0.5, dimana dari nilai tersebut dapat diartikan bahwa hanya setengah (empat) citra yang ditampilkan serupa dengan citra acuan. Nilai tersebut dianggap kurang memuaskan. Sedangkan metode *sobel* dan *prewitt* menghasilkan nilai yang besarnya sama.



Gambar 12. Grafik *Precision Nilai S (Saturation)*

Gambar 12 menunjukkan hasil yang berbeda dari gambar 9. Gambar 12 menunjukkan bahwa metode LoG memberikan hasil yang lebih baik dari metode *canny*, *prewitt* dan *sobel*. Bahkan nilai terbaik yang didapatkannya sebesar 0.6, dimana nilai tersebut lebih tinggi 0.1 dari nilai terbaik pada Gambar 11 yang didapatkan dari metode *canny*. Hasil yang didapatkan dari dua pengujian tersebut membuktikan bahwa penggunaan nilai S (*Saturation*) lebih baik dari nilai H (*Hue*).



Gambar 13. Grafik *Precision Nilai V (Value)*

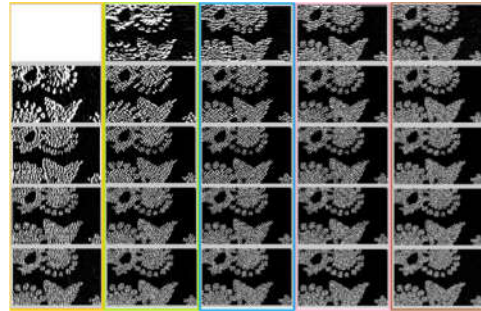
Pada Gambar 13 yang menunjukkan presisi pada ruang warna V, metode Log (*Laplacian of Gaussian*) mendapatkan hasil yang lebih baik dari metode yang lainnya. Bahkan hasil yang didapatkan kali ini lebih bagus dari pengujian ekstraksi

tepi sebelumnya. Nilai tertinggi pada metode Log (*Laplacian of Gaussian*) adalah satu yang artinya citra yang ditampilkan seluruhnya serupa dengan citra acuan. Sedangkan untuk nilai terendah adalah diatas 0.8, hal ini berarti hanya ada satu citra yang tidak serupa. Diurutkan kedua adalah metode *canny* dimana rata-rata dari metode ini masih lebih baik dari pengujian sebelumnya. Sedangkan metode *prewitt* dan *sobel* mendapatkan hasil yang sama. Ini membuktikan bahwa pemanfaatan nilai V (*Value*) lebih baik dibandingkan dengan H (*Hue*) dan S (*Saturation*). Untuk penggunaan metode, metode Log (*Laplacian of Gaussian*) dengan memanfaatkan nilai V (*Value*) lebih memuaskan dari yang lain.

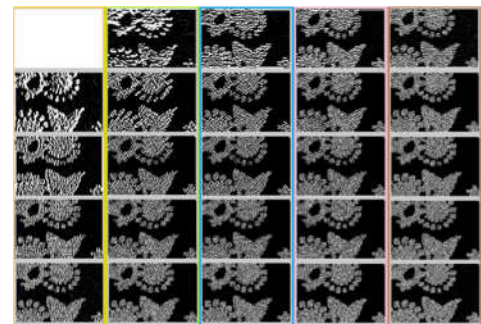
4.3. Pengujian Keefektifan Ekstraksi Tekstur Citra Batik

Ekstraksi tekstur dalam citra diharapkan mampu memberikan kontribusi terhadap ketepatan sistem dalam menemukan citra yang serupa. Karena tekstur juga merupakan bagian penting dalam citra batik. Pada sistem metode *Law Texture* memiliki dua vektor masukan berupa v1 dan v2 yang nantinya akan digunakan untuk mengekstraksi tekstur dari citra. v1 dan v2 terdiri dari lima jenis vektor yang masing-masing berupa *level*, *edge*, *spot*, dan *ripple*.

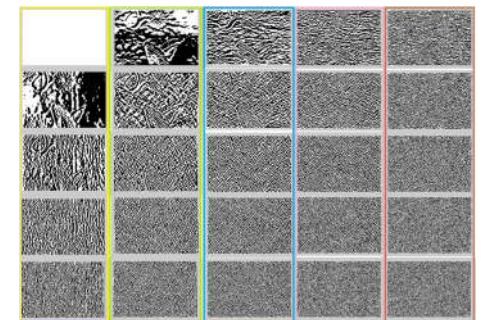
Dalam Gambar 14, 15, dan 16 terdapat lima kotak yang dibedakan warnanya. Pada kotak berwarna kuning v1 diset *level*, kotak berwarna hijau v1 diset *edge*, kotak berwarna biru v1 diset *spot*, kotak berwarna merah muda v1 diset *wave*, dan terakhir kotak berwarna coklat v1 diset *ripple*. Jika diurutkan dari atas ke bawah v2 diset *level*, *edge*, *spot*, *wave*, dan *ripple*.



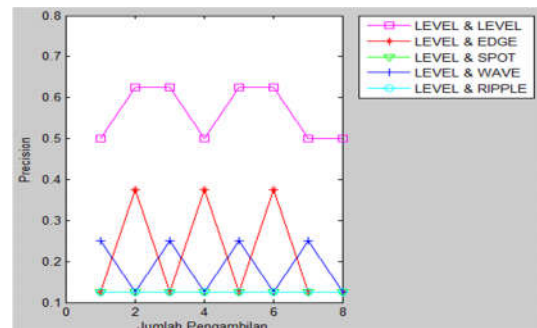
Gambar 14. Hasil Ekstraksi Tekstur dengan Nilai H (*Hue*)



Gambar 15. Hasil Ekstraksi Tekstur dengan Nilai S (*Saturation*)

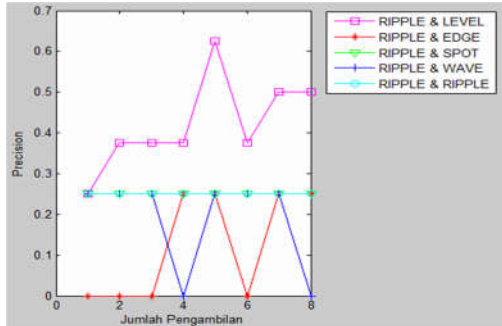


Gambar 16. Hasil Ekstraksi Tekstur dengan Nilai V (*Value*)



Gambar 17. Grafik Precision Ekstraksi Tekstur Nilai H dengan v1 Level

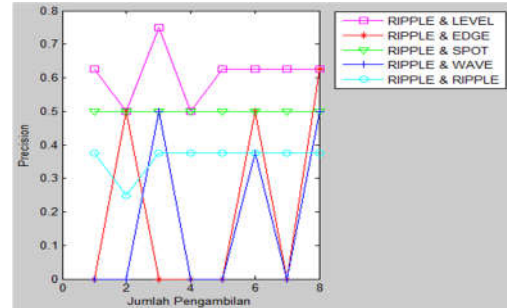
Dari Gambar 17 dapat dilihat bahwa kombinasi *level* dan *level* dengan memanfaatkan nilai H menjadi yang terbaik



Gambar 18. Grafik *Precision* Ekstraksi Tekstur Nilai S dengan $v1$ *Ripple*

Jika diamati secara visual hasil ekstraksi tekstur pada ruang warna S yang didapatkan pada Gambar 15 serupa dengan yang didapatkan pada pengujian pada ruang warna H yang dapat dilihat pada Gambar 14. Sedangkan untuk hasil pengujian secara data *ripple* dan *level* adalah yang terbaik dan dapat dilihat pada Gambar 18.

Sedangkan untuk ruang warna V, jika diamati hasil pengujian pada Gambar 16 terlihat begitu berbeda dari hasil dua pengujian sebelumnya. Tetapi ada beberapa yang memiliki ciri sama yaitu kombinasi antara *level* dengan *level* terlihat putih semua. Kombinasi antara *edge* dengan *level* atau *level* dengan *edge* yang tekstur bunganya terlihat timbul. Sedangkan untuk hasil pengujian secara data dapat dilihat pada Gambar 19.



Gambar 19. Grafik *Precision* Ekstraksi Tekstur Nilai V dengan $v1$ *Ripple*

Dari seluruh percobaan dengan *Law Texture* didapatkan hasil bahwa nilai H (*Hue*) lebih bagus menggunakan kombinasi antara *level* dengan *level*, sedangkan nilai S (*Saturation*) lebih bagus menggunakan kombinasi antara *ripple* dengan *level*, dan yang terakhir nilai V (*Value*) lebih bagus menggunakan kombinasi *ripple* dengan *level*.

5. Penutup

Dari beberapa hasil percobaan yang telah dilakukan pada bab sebelumnya terdapat kesimpulan yang dapat diambil antara lain :

1. Kombinasi fitur tepi Laplacian of Gaussian (LoG) dengan komponen Value pada ruang warna HSV memberikan nilai *precision* mencapai 0.9.
2. Kombinasi tekstur pada komponen Value pada ruang warna HSV juga memberikan nilai presisi paling tinggi dibandingkan dengan komponen lain dalam ruang warna tersebut.
3. Penggunaan warna, tepi dan texture menambah presisi hasil pencarian dari pada hanya menggunakan salah satunya.

Untuk penelitian selanjutnya perlu dilakukan percobaan dengan

menggunakan metode ekstraksi fitur yang lain seperti Gabor.

6. Daftar Pustaka

- [1] R. OKSAPUTRI, E. Ernawati, and A. Desi, "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)," *Pseudocode*, vol. 5, no. 2, pp. 18–28, 2018.
- [2] A. S. Arwanda, "Content Based Image Retrieval Batik Tradisional Yogyakarta Dengan Ekstrasi Ciri Berdasarkan Tekstur Filter Gabor Wavelets 2D Skripsi Content Based Image Retrieval Dengan Ekstrasi Ciri Berdasarkan Tekstur Filter Gabor Wavelets 2D," *Ticom*, vol. 1, no. 3, pp. 12–18, 2013.
- [3] A. Y. Wicaksono, N. Suciati, and D. Purwitasari, "Implementasi Transformasi Curvelet dan Ruang Warna HSV untuk Temu Kembali Citra Batik Berbasis Isi pada Situs Batik," *J. Tek. POMITS*, vol. 2, no. 1, pp. 1–4, 2013.
- [4] F. A. Hermawati, "Sistem Temu Kembali Citra Berdasarkan Karakteristik Bentuk dengan Metode Color-Edge Extraction," in *Seminar Nasional Teknik 2009*, 2009, pp. 253–257.
- [5] F. A. Hermawati, "Segmentasi Citra Berdasarkan Karakteristik Color-Texture," *KONVERGENSI*, vol. 1, no. 1, pp. 1–8, 2005.
- [6] F. A. Hermawati, H. H. Tjandrasa, and N. Suciati, "Evaluasi Representasi Warna Untuk Retrieval Citra Berbasis Region," *J. Saintek*, vol. 9, no. 2, pp. 101–107, 2005.
- [7] F. A. Hermawati, H. Tjandrasa, and N. Suciati, "Sistem Retrieval Citra Berbasis Region Dengan Transformasi Wavelet Berdasarkan Karakteristik Color-Texture," *KONVERGENSI*, vol. 2, no. 1, pp. 1–9, 2006.
- [8] Y. MZ, E. Utami, and A. Amborowati, "Temu Kembali Citra Batik Pesisir," *J. Inf. Interaktif*, vol. 2, no. 1, pp. 1–9, 2017.
- [9] M. Nasir, N. Suciati, and A. Y. Wijaya, "Kombinasi Fitur Tekstur Local Binary Pattern yang Invariant terhadap Rotasi dengan Fitur Warna Berbasis Ruang Warna HSV untuk Temu Kembali Citra Kain Tradisional," *J. Inspirat.*, vol. 7, no. 1, pp. 42–51, 2017.