

Penerapan Algoritma *Support Vector Machine* untuk Klasifikasi Motif Citra Batik Solo Berdasarkan Fitur *Multi-Autoencoders*

Rezky Arisanti Putri¹, Naim Rochmawati²

^{1,2}Jurusan Teknik Informatika, Teknik Informatika, Universitas Negeri Surabaya

¹rezkyarisanti99@gmail.com

²naimrochmawati@unesa.ac.id

Abstrak—Batik merupakan kain tradisional warisan kebudayaan masyarakat Indonesia. Penggunaan kain batik sebagai bahan pakaian telah populer sejak kerajaan Majapahit. Sebelum sepopuler sekarang, batik telah menjadi pilihan masyarakat Indonesia sebagai bahan pakaian tradisional karena memiliki ciri khas di setiap motifnya. Motif batik memiliki keberagaman dengan ciri yang menonjol pada setiap lilinnya. Sebagai wujud upaya pelestarian batik, penelitian mengenai klasifikasi batik dilakukan untuk mendeteksi motif citra batik. Berdasarkan hal tersebut, penelitian ini dilakukan menggunakan metode pembelajaran berdasarkan data citra batik dengan harapan dapat memberikan hasil klasifikasi motif citra batik yang lebih akurat dan efisien serta tingkat akurasi terbaik. Adapun metode penelitian yang digunakan yaitu Algoritma Support Vector Machine sebagai Algoritma klasifikasi dan Multi-Autoencoder sebagai proses ekstraksi fitur. Jaringan Multi-Autoencoder yang dirancang merupakan penggabungan dari dua atau tiga model Autoencoder. Model Autoencoder yang dibuat menggunakan ekstraksi dari citra Edge, citra Grayscale, dan citra SIFT. Selanjutnya, akan diklasifikasi menggunakan Algoritma Support Vector Machine. Penelitian ini menggunakan data citra batik sejumlah 2.256 gambar yang terbagi menjadi 7 class yaitu motif Parang, Buketan, Ceplok, Kawung, Truntum, Semen Rante, dan Sidomukti. Data gambar batik dibagi menjadi data learning dan data testing dengan presentase sebesar 80% untuk learning dan 20% untuk testing. Percobaan dilakukan berulang untuk mendapatkan model Multi-Autoencoder terbaik untuk proses klasifikasi. Penelitian ini menghasilkan tingkat akurasi sebesar 29,581% untuk kombinasi citra Edge-SIFT; 18,407% untuk citra Edge-Grayscale; 29,157% untuk citra Grayscale-SIFT; serta 30,159% untuk kombinasi citra Edge-SIFT-Grayscale.

Kata Kunci—Klasifikasi Motif Batik, Batik Solo, Support Vector Machine, Autoencoder, Multi-Autoencoder.

I. PENDAHULUAN

Batik merupakan salah satu warisan kebudayaan Indonesia yang menjadi ciri khas masyarakat Indonesia. Kain batik banyak digunakan baik untuk pakaian resmi maupun sehari-hari. Sebagai ciri khas masyarakat Indonesia, batik memiliki ragam corak dan motif dengan karakteristik yang berbeda tiap wilayah asalnya. Batik dianggap sebagai pakaian tradisional yang banyak digunakan oleh masyarakat Indonesia. Batik adalah kain dekoratif yang memiliki peran yang sangat penting dalam acara tradisional seperti upacara tradisional, baik budaya jawa maupun budaya lain di wilayah Indonesia [1]. Pada umumnya, seni batik merupakan seni melukis lilin atau cara menghias kain dengan menutupi bagian kain dengan lapisan

lilin kemudian mencelupkan kain. Teknik pencelupan ini digunakan untuk menutupi bagian kain yang tidak dilapisi dengan lilin. Bagian kain yang dilapisi lilin akan tetap berwarna aslinya dan ketika lilinnya dihilangkan, maka akan terlihat perbedaan antara daerah yang dicelup dan tidak dicelup membentuk sebuah pola [2].

Meskipun kain batik telah populer di hampir seluruh dunia, namun beberapa jenis kain batik menunjukkan motif yang sangat khas suatu daerah. Motif menunjukkan fitur atau karakteristik untuk mengidentifikasi asal usul kain batik [3]. Untuk melestarikan warisan budaya Indonesia ini, diperlukan penelitian yang berkesinambungan. Salah satu topik penelitian yang banyak dikembangkan dalam bidang komputer sains yaitu klasifikasi batik. Topik penelitian ini juga relevan dengan topik yang masih aktual yaitu ekstraksi fitur. Sebuah model mesin learning membutuhkan hasil ekstraksi fitur dari gambar batik untuk mencapai akurasi klasifikasi yang tinggi [4]. Fitur digunakan untuk mewakili karakteristik gambar sehingga dapat mengidentifikasi gambar secara akurat. Namun, motif batik yang bervariasi akan menyebabkan kesulitan dalam mengidentifikasi gambar batik. Sehingga diperlukan sebuah metode yang dapat digunakan untuk mengekstraksi fitur batik berdasarkan motifnya.

Penelitian mengenai klasifikasi motif batik telah banyak dikembangkan oleh para peneliti. Salah satunya dilakukan oleh Azhar pada tahun 2015 dengan judul Batik Image Classification Using SIFT Feature Extraction, Bag of Features and Support Vector Machine [5]. Hasil akhir penelitian menunjukkan bahwa pengembangan sebuah metode ekstraksi fitur menggunakan SIFT dan SVM sebagai classifier dapat digunakan untuk mengenali motif batik dengan tingkat akurasi sebesar 97.67%, 95.47%, dan 79% dalam kondisi gambar normal, gambar dirotasi dan diubah skalanya. Adi & Soelistio juga melakukan penelitian dengan tema yang sama yaitu klasifikasi motif batik pada jurnalnya yang berjudul “Classification Batik Bird Motif with Gabor Filter, PCA, SVM Using Batik Tanjung Bumi Bangkalan And Other Batik As Subject” [6]. Hasil akhir penelitian menunjukkan bahwa hasil akurasi penelitian dengan menggunakan Gabor Filter, PCA, dan SVM adalah sebesar 100% pada proses training dan 66.68% pada proses testing. Dari beberapa penelitian tersebut, metode ekstraksi fiturnya masih menggunakan handcrafted extraction features, sehingga penelitian yang akan dilakukan menggunakan metode pengembangan Deep Learning. Dengan menggunakan arsitektur jaringan Deep Learning, penelitian ini

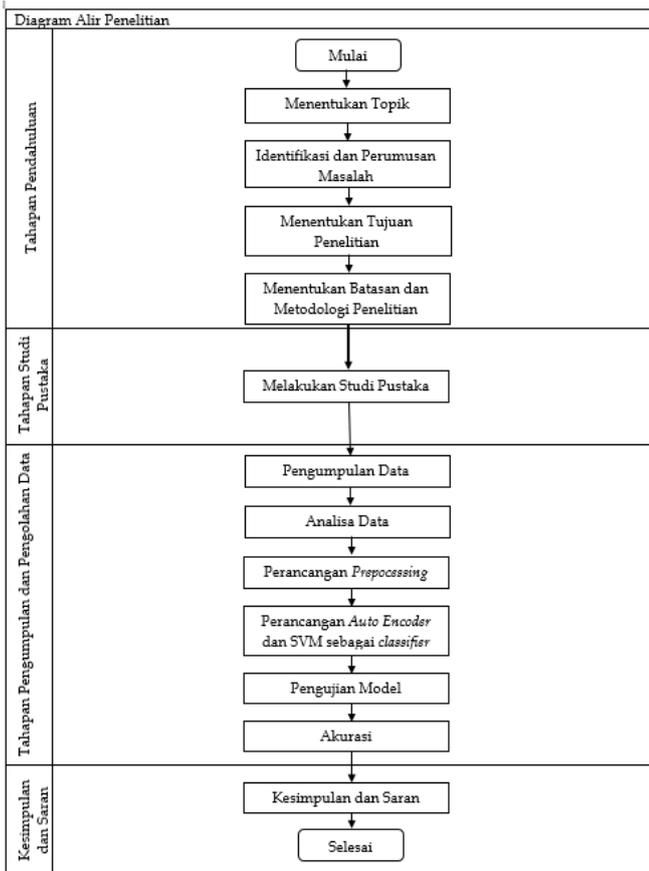
akan menghasilkan sistem klasifikasi motif citra batik yang baik dengan hasil akurasi yang maksimal tentunya. Metode pengembangan Deep Learning yang akan digunakan dalam penelitian ini. Model Deep Learning mempunyai banyak metode yang digunakan dalam klasifikasi gambar maupun pengenalan objek. Model pengembangan Deep Learning yang digunakan yaitu Multi-Autoencoders.

Berdasarkan permasalahan yang telah dijelaskan, penelitian ini mencoba merancang sebuah sistem untuk mengklasifikasikan motif batik berdasarkan fitur yang ada pada motifnya. Sistem yang dirancang akan menggunakan model jaringan Multi-Autoencoders untuk mendapatkan fitur dari citra batik, yang nantinya akan digunakan sebagai input dari Algoritma Support Vector Machine untuk klasifikasi motif batik. Penggunaan Algoritma Support Vector Machine ini dikarenakan dalam beberapa masalah linear maupun non-linear algoritma ini dapat bekerja secara efektif.

II. METODE PENELITIAN

Cara paling mudah untuk memenuhi persyaratan format penulisan adalah dengan menggunakan dokumen ini sebagai template. Kemudian ketikkan teks Anda ke dalamnya.

Alur penelitian sistem “Klasifikasi Motif Batik Menggunakan Algoritma Multi-Autoencoders dan Support Vector Machine sebagai classifier”, digambarkan dalam bagan berikut.



Gbr. 1 Alur Penelitian

Penelitian yang dilakukan merupakan penelitian mengenai penerapan sistem klasifikasi motif batik menggunakan salah satu metode Deep Learning yaitu Multi-Autoencoders dan Support Vector Machine. Pada penelitian ini, model jaringan Multi-Autoencoders digunakan sebagai metode ekstraksi fitur pada citra batik untuk menghasilkan fitur yang akan digunakan sebagai klasifikasi motif batik berdasarkan fiturnya. Sedangkan Algoritma Support Vector Machine (SVM) digunakan sebagai classifier-nya.

Penelitian ini terbagi dalam dua tahap penelitian yaitu tahap pembelajaran (training) dan tahap pengujian (testing). Pada tahap training, penelitian dilakukan dengan menginputkan data training berupa citra batik ke dalam model jaringan Multi-Autoencoders sebagai ekstraksi fitur dan SVM sebagai classifier. Sesuai dengan namanya, tahap ini merupakan tahap sistem melakukan pembelajaran mengenai jaringan yang diusulkan dengan tujuan untuk memperbarui bobot pada tiap lapisan. Sedangkan, untuk proses klasifikasi dengan algoritma SVM, proses training menghasilkan support vector yang akan digunakan kembali untuk proses pengujian. Sedangkan pada tahap pengujian, data testing berupa citra batik digunakan sebagai citra input ke dalam sistem. Fitur citra batik pada data testing akan dibandingkan dengan fitur hasil training, maka akan didapatkan data citra yang sesuai dengan kelasnya.

Sebelumnya, data citra yang telah dikumpulkan dibagi menjadi dua bagian, yaitu data training dan data testing. Prosentase pembagian data training dan testing masing-masing yaitu sebesar 80% dan 20%. Total keseluruhan data yang digunakan dalam penelitian ini yaitu 2.256 gambar. Dari total 2.256 gambar tersebut, sebanyak 1.876 gambar digunakan untuk data training dan 380 gambar digunakan untuk data testing.

Alur jalannya sistem klasifikasi motif citra batik menggunakan Algoritma Autoencoder adalah sebagai berikut:

1. Cropping citra
 Data gambar yang didapatkan dari proses pengumpulan data memiliki ukuran panjang dan lebar yang tidak sama. Oleh karena itu, sebelum masuk ke proses selanjutnya, gambar harus dilakukan proses cropping agar proses dapat berjalan dengan baik.
2. Resize ukuran citra
 Setelah dilakukan proses cropping, gambar dilakukan proses resize agar ukuran panjang dan lebarnya sama.
3. Mengubah vector menjadi citra
 Proses mengubah vector citra merupakan proses dimana data yang berbentuk gambar diubah menjadi vector matriks yang berisi angka agar bisa diolah pada proses selanjutnya. Proses pengubahan vector dibagi menjadi 3 bagian, dikarenakan vector citra yang digunakan dalam penelitian ini mencakup citra grayscale, citra edge, dan citra SIFT. Pada pengolahan vector fitur SIFT, hasil vector yang didapatkan memiliki jumlah yang tidak sama antara satu gambar dengan gambar lainnya. Sehingga, penelitian ini menggunakan Arsitektur Bag Of Features (BOF) yang diadopsi dari arsitektur Bag Of Words (BOW). Pada proses ini juga dilakukan proses normalisasi agar besaran angka vector fitur yang

dihasilkan tidak terlalu besar. Proses normalisasi menggunakan dua cara, yaitu proses MinMaxscaller untuk citra SIFT dan proses normalisasi pembagian untuk citra Edge dan Grayscale. Proses normalisasi MinMaxscaller menggunakan rumus persamaan berikut:

$$\text{Normalisasi} = \frac{\text{Nilai awal} - \text{nilai min}}{\text{Nilai max} - \text{nilai min}} \quad (1)$$

Nilai ekstraksi fitur akan dinormalisasi ke dalam rentang 0 sampai 1. Adapun tahapan normalisasi yaitu dengan mencari nilai maksimum dan minimum dari nilai ekstraksi fitur yang telah diperoleh pada tahap ekstraksi fitur menggunakan SIFT. Setelah itu dimasukkan ke dalam persamaan dan menghasilkan nilai yang siap digunakan untuk proses selanjutnya.

Sedangkan normalisasi citra Gray dan Edge menggunakan persamaan (2).

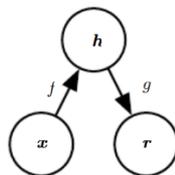
$$\text{Normalisasi} = \frac{\text{pixel citra}}{\text{pixel terbesar}} \quad (2)$$

Menurut Persamaan (2), pixel terbesar merupakan range angka terbesar pada citra. Karena citra berada dalam range antara 0 sampai 255, maka pixel yang terbesar digunakan nilai 255. Setelah dilakukan proses normalisasi, data siap digunakan untuk proses selanjutnya.

4. Proses Autoencoders

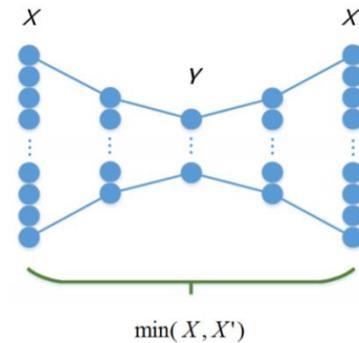
Autoencoders adalah sebuah jaringan Neural Network yang dilatih untuk menyalin input maupun outputnya. Autoencoders dikatakan sebagai algoritma unsupervised learning, dimana tujuan optimasinya adalah nilai output yang dihasilkan sama dengan nilai input yang dimasukkan atau bisa dituliskan: $y(i) = x(i)$. Jaringan ini memiliki tiga buah layer yaitu input layer, output layer dan hidden layer. Autoencoders mempunyai sebuah hidden layer yang mendeskripsikan sebuah code yang digunakan untuk merepresentasikan inputnya.

Konsep jaringan ini yaitu mempelajari input dan berusaha untuk melakukan rekonstruksi kembali pada data inputnya namun dengan informasi yang lebih sedikit. Jaringan ini terdiri atas dua bagian yaitu sebuah fungsi encoder $h = f(x)$ dan sebuah fungsi decoder yang menghasilkan rekonstruksi $r = g(h)$.



Gbr. 2 Struktur Umum Multi-Autoencoders

Fungsi encoder berfungsi untuk merubah input ke dalam bentuk dimensi yang lebih kecil atau bisa juga disebut juga dengan kompresi. Sedangkan fungsi decoder berfungsi untuk merekonstruksi code layer menjadi bentuk aslinya. Dengan mempertimbangkan sejumlah data latih x dengan jumlah data latih sebanyak n dan fitur sebanyak m . Hasil output dari encoder y mewakili representasi data x yang telah direduksi. Decoder digunakan untuk merekonstruksi data asli x dari hasil representasi encoder y dengan meminimalkan perbedaan antara x dan x' yang ditunjukkan pada Gambar berikut.



Gbr. 3 Autoencoders menghasilkan fitur baru y dengan meminimalkan rekonstruksi yang hilang antara x dan x'

Proses Autoencoders merupakan proses learning data menggunakan vector citra yang telah disimpan dalam format file .jlb. Proses Autoencoder ini juga berperan sebagai proses untuk mengurangi dimensi data agar proses klasifikasi yang dilakukan setelahnya menjadi lebih cepat.

Encoder adalah sebuah fungsi f yang merepresentasikan sebuah input x untuk menyembunyikan representasi y , yang dituliskan dalam formula berikut.

$$y = f(x) = s_f(wx + b_x) \quad (3)$$

dimana s_f merupakan sebuah fungsi aktivasi non-linier dan jika itu adalah fungsi identitas, Autoencoders akan melakukan proyeksi linier. Encoder ini memiliki parameter yaitu bobot matriks w dan sebuah vector bias $b \in R_n$.

Fungsi decoder direpresentasikan oleh fungsi g , untuk mewakili y dari rekonstruksi x .

$$x' = g(y) = s_g(w'y + b_y) \quad (4)$$

dimana s_g adalah fungsi aktivasi decoder, baik menggunakan identitas maupun sigmoid, yang biasanya menghasilkan rekonstruksi linear. Parameter yang digunakan yaitu vector bias b dan matriks w' .

Proses training untuk jaringan Autoencoders digunakan untuk menemukan parameter $\theta = (w, b_x, b_y)$ yang meminimalkan rekonstruksi yang hilang pada data input x . Formula untuk proses training dituliskan sebagai berikut.

$$\theta = \min_{\theta} L(x, x') = \min_{\theta} L(x, g(f(x))) \quad (5)$$

Untuk rekonstruksi linier, rekonstruksi yang hilang (L1) yang umumnya menggunakan square error, dituliskan dalam persamaan berikut.

$$L_1(\theta) = \sum_{i=1}^n \|x_i - x'\|^2 = \sum_{i=1}^n \|x_i - g(f(x_i))\| \quad (6)$$

Sedangkan untuk rekonstruksi nonlinier, rekonstruksi yang hilang (L2) yang umumnya menggunakan cross entropy, dituliskan dalam persamaan berikut.

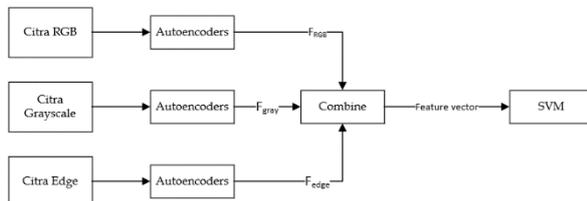
$$L_2(\theta) = - \sum_{i=1}^n [x_i \log(y_i) + (1 - x_i) \log(1 - y_i)] \quad (7)$$

dimana $x_i \in x$, $x'_i \in x'$ dan $y_i \in y$.

5. Proses Multi-Autoencoders

Multi-Autoencoders merupakan model jaringan Deep Neural Network dengan masalah utamanya yaitu bagaimana cara melatih parameter optimasi pada jaringannya.

Proses Multi-Autoencoders merupakan proses penggabungan vector fitur dari jaringan Autoencoder. Ketentuan penggabungan merujuk pada penelitian yang dilakukan oleh Mabu [7].



Gbr. 4 Jaringan Multi-Autoencoders

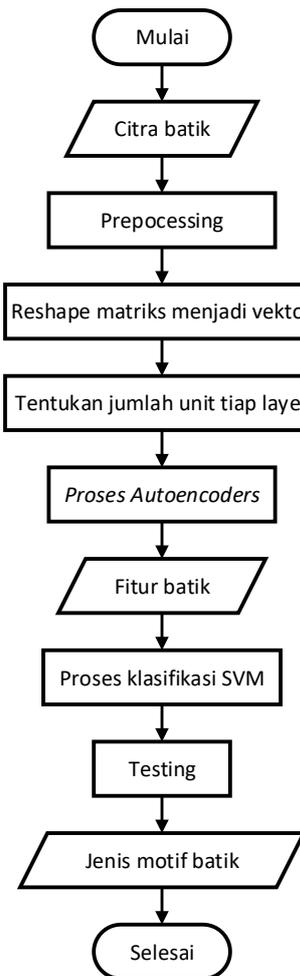
6. Proses Support Vector Machine (SVM)

Setelah dilakukan penggabungan vector fitur, model yang didapatkan dari jaringan Multi-Autoencoders akan dilearning menggunakan Algoritma SVM. Kernel yang digunakan dalam penelitian ini adalah kernel linear. Hasil dari proses learning ini adalah model SVM yang akan digunakan pada proses testing dan perhitungan akurasi.

7. Proses testing dan perhitungan akurasi

Selanjutnya, model SVM akan digunakan untuk proses testing. Proses testing dilakukan untuk menentukan model SVM terbaik yang akan digunakan untuk menentukan klasifikasi motif batik.

Alur jalannya sistem klasifikasi motif citra batik menggunakan fitur *Multi-Autoencoders* dapat dilihat pada bagan berikut:



Gbr. 5 Alur Penelitian

III. HASIL DAN PEMBAHASAN

Tahapan sistem klasifikasi motif batik adalah sebagai berikut.

A. Pengolahan Data Batik

Data gambar yang telah dikumpulkan selanjutnya diolah agar dapat digunakan untuk proses selanjutnya. Pengumpulan Data pada penelitian ini menggunakan metode Library Research. Pada metode Library Research, data diperoleh dari literatur, membaca jurnal-jurnal ilmiah dalam negeri maupun luar negeri, dan penelitian-penelitian sebelumnya serta dokumen lain yang berhubungan dengan penelitian. Dalam penelitian ini, data yang digunakan merupakan data gambar berupa gambar batik.

Data yang telah dikumpulkan dibagi menjadi dua bagian yaitu data learning dan data testing. Presentase data learning dan data testing masing-masing yaitu sebesar 80% dan 20%. Setelah dibagi, data diolah dengan menjadikannya dalam format file .csv.

B. Preprocessing Data

Preprocessing data merupakan pra proses sebelum data diolah kedalam proses inti yaitu proses ekstraksi fitur dan klasifikasi data. Tujuan dari proses ini adalah agar data yang diproses menjadi data yang berkualitas sehingga menghasilkan tingkat akurasi klasifikasi yang optimal.

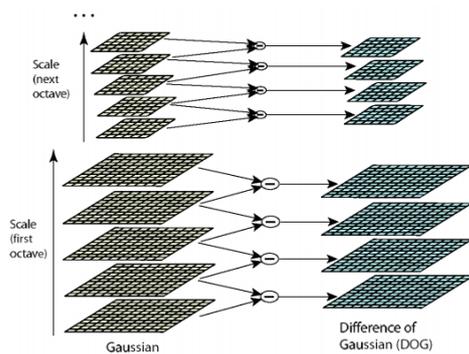
Tahap awal dalam preprocessing data adalah melakukan cropping citra. Tahap cropping citra akan menghasilkan gambar batik dengan bentuk sisi yang sama panjang. Setelah dilakukan proses cropping, citra batik kemudian di resize menjadi ukuran 250x250 pixel. Proses resize dilakukan agar data citra yang akan diolah memiliki ukuran dan perlakuan yang sama. Selanjutnya, citra akan diproses menjadi bentuk citra grayscale, citra edge dan citra SIFT. Vector fitur yang telah didapat kemudian dilakukan proses normalisasi.

Tahapan preprocessing citra SIFT merupakan tahap ekstraksi fitur menggunakan metode ekstraksi fitur Scale Invariant Features Transform. Adapun tahap-tahap ekstraksi fitur tekstur menggunakan SIFT yaitu:

1. Scale-space Extreme Detection

Inisialisasi keypoint yaitu mengidentifikasi lokasi dan skala dengan menggunakan operasi konvolusi. Sebuah gambar yang telah dikonversi menjadi citra grayscale di konvolusikan dengan menggunakan *Gaussian Convolution (Gaussian Blur)*, dengan meningkatkan nilai standar deviasi untuk setiap konvolusinya.

Selanjutnya, membuat skala ruang (*scale space*) dengan tingkatan octave, dimana penurunan satu tingkatan octave citra konvolusi yang berukuran setengah dari citra pada tingkatan octave yang sebelumnya. Kemudian dilakukan proses konvolusi (blur) pada octave yang baru. Langkah ini diulangi hingga gambar terlalu kecil untuk diproses, biasanya berakhir sampai 4 octave atau 4 tahap. Setelah didapatkan scale-space, dihitung perbedaan antar 2 scale-space berturut-turut. Perbedaan kedua scale-space dihitung menggunakan *Different of Gaussian (DoG)* untuk mengeliminasi [8].



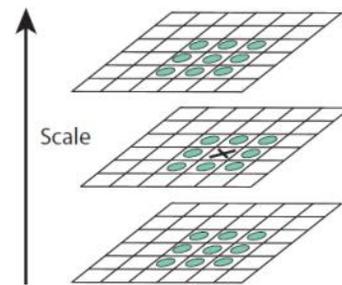
Gbr. 6 Eliminasi dengan DoG

Gambar 2.10 menunjukkan gambar DoG yang dihasilkan memiliki ukuran yang berbeda tiap octave. Dapat dilihat pada Gambar tersebut, pendekatan yang dilakukan dengan mengambil pasangan gambar

pertama dan kedua lalu dikurangkan, kedua dan ketiga, selanjutnya berturut-turut. Pada Gambar tersebut hanya diterapkan DoG pada satu octave, perhitungan akan dilakukan pada semua octave.

2. Keypoint Localization

Lokalisasi keypoint adalah mencari nilai extrema (maximum dan minimum) untuk mengetahui titik-titik yang akan menjadi kandidat keypoint. Setelah menghitung *Different of Gaussian (DoG)* dilanjutkan dengan menghitung Laplacian of Gaussian (LoG) yang akan menghasilkan titik-titik atau point. Selanjutnya, mencari maksima atau minima dari DoG space. Pencarian dilakukan dengan menemukan titik maxima/minima kasarnya, dengan cara melewati setiap piksel dan memeriksa tetangga terdekatnya [8].



Gbr. 7 Tetangga extrema DoG space

Pada Gambar 2.11, x menandakan piksel, sedangkan lingkaran biru menandakan tetangganya. Total keseluruhan tetangga yang didapat yaitu sebanyak 26 tetangga. Perhitungan 26 tetangga ini didapatkan dari 8 piksel pada gambar yang sama, 2 piksel yang bersesuaian pada gambar yang sama di octave yang sama pula, serta tetangga dari 2 piksel yang bersesuaian (16 piksel). 'x' bertindak sebagai keypoint jika x adalah yang terbesar dari ke-26 tetangga sekitar titik tersebut. Tetangga ekstrema dari DoG space akan menjadi kandidat keypoint. Sedangkan untuk kotak yang putih atau kosong hanya dilewati saja dan tidak diperhitungkan.

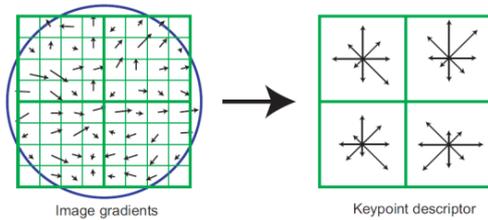
3. Orientation Assignment

Penandaan orientasi adalah menetapkan orientasi pada setiap keypoint yang telah diseleksi dengan mengumpulkan arah gradient dan nilai besaran sekitar pada setiap keypoint, sehingga didapatkan orientasi yang menonjol.

4. Deskripsi keypoint

Deskripsi keypoint adalah pembuatan descriptor untuk setiap keypoint yang berfungsi untuk mengidentifikasi setiap keypoint. Caranya, dengan mengambil daerah kecil di sekitar keypoint tersebut, lalu membagi daerah tersebut menjadi ukuran 4 x 4, sehingga menghasilkan 16 area. 16 area yang dihasilkan ini kemudian dibagi lagi menjadi enam belas jendela 4 x 4, sehingga didapatkan area sel sebanyak 16 x 16.

Selanjutnya, membuat histogram orientasi gradien pada tiap sel. Histogram ini dihitung dari magnitudo dan nilai orientasi dari sample wilayah 16×16 tersebut. Magnitudo dihitung dengan menggunakan fungsi Gaussian dengan nilai tetha (θ) adalah satu setengah lebar descriptor, sehingga didapatkan semua nilai histogram dari vector descriptor. Jika dihitung ada $4 \times 4 = 16$ histogram yang masing-masing memiliki delapan bins, sehingga untuk tiap descriptor memiliki 128 elemen[8].

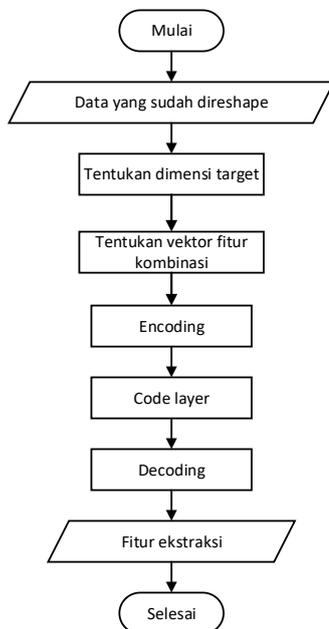


Gbr. 8 Keypoint Descriptor

C. Proses Autoencoder

Setelah dilakukan proses normalisasi citra, Vector fitur yang dihasilkan dari preprocessing kemudian diproses menggunakan Algoritma Autoencoder. Masing-masing vector fitur dirancang menggunakan arsitektur jaringan Autoencoder yang terdiri atas encode layer, code layer dan decode layer. Encode layer merupakan layer input jaringan yang merupakan jumlah vector fitur yang dihasilkan. Code layer merupakan layer yang merepresentasikan layer encode (input) namun memiliki dimensi yang lebih rendah dengan jumlah node yang lebih rendah. Sedangkan decode layer merupakan layer output yang memiliki dimensi data yang sama dengan layer input. Namun, pada penelitian ini jaringan yang digunakan hanya pada code layer.

Adapun proses Algoritma Autoencoder adalah sebagai berikut.



Gbr. 9 Algoritma Autoencoder

Pada jaringan ini menghasilkan model yang nantinya akan digunakan untuk proses learning Algoritma SVM. Model yang dihasilkan terdiri dari model Autoencoder, model encode, dan history jaringan. Range pola jaringan ini menggunakan jumlah node antara 200 sampai 1000 node.

Jaringan dilatih menggunakan Autoencoder untuk mendapatkan model Autoencoder yang tepat untuk citra Grayscale, Edge dan SIFT. Proses learning menggunakan jaringan Autoencoder dilakukan dua kali, yaitu dengan mengurangi range jaringan. Sebelum itu, model Autoencoder yang dihasilkan pada tiap citra fitur dipilih yang memiliki tingkat akurasi yang terbaik dari masing-masing range.

D. Proses Multi-Autoencoder

Setelah didapatkan model vector fitur terbaik dari beberapa vector citra, dilakukan proses combine vector. Aturan untuk mengkombinasikan vector ini mengacu pada jurnal [4] dengan ketentuan sebagai berikut.

- 1) $\{ F_{Edge}(d), F_{SIFT}(d), F_{edge}(d) \}$
- 2) $\{ F_{SIFT}(d), F_{Gray}(d) \}$
- 3) $\{ F_{Edge}(d), F_{SIFT}(d) \}$
- 4) $\{ F_{Gray}(d), F_{edge}(d) \}$

Vector fitur kombinasi yang telah dibuat kemudian disimpan dalam format file .jlb untuk digunakan pada proses learning Algoritma SVM.

E. Proses Support Vector Machine

Pada proses learning SVM menggunakan fungsi aktivasi linear. Langkah-langkah perhitungan metode *Support Vector Machine* adalah sebagai berikut.

- 1) Hasil ekstraksi fitur yang didapat dari proses *Multi-Autoencoders* merupakan pengolahan data awal dari Algoritma *Support Vector Machine*.
- 2) Menentukan Fungsi Kernel, fungsi kernel yang digunakan adalah fungsi kernel linear.
- 3) Menghitung matriks kernel, dengan menggunakan persamaan (3).

$$K(x, xi) = \Phi(x) \cdot \Phi(xi) \quad (8)$$

- 4) Menghitung nilai Ld terbesar. Setelah didapatkan nilai tiap elemen matriks kernel $K(x, xi)$, nilai tersebut digunakan untuk menggantikan dot-product $x_i \cdot x_j$ dalam persamaan dualitas Lagrange Multiplier yang dituliskan pada persamaan (4).

$$Ld = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (9)$$

- 5) Menghitung nilai weight, dengan menggunakan persamaan (5).

$$w = \sum_{i=1}^N \alpha_i y_i x_i \quad (10)$$

- 6) Menghitung nilai bias dengan menggunakan persamaan berikut.

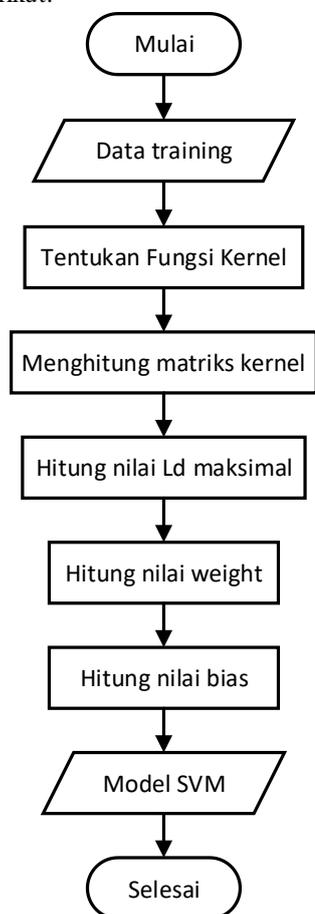
$$b = y_i - (w \cdot x_i) \quad (11)$$

Kombinasi	Hasil Learning	Hasil Testing
SIFT + Edge	58,129%	29,581%
SIFT + Gray	50,013%	29,157%
Gray + Edge	99,251%	18,407%
SIFT + Edge + Gray	60,934%	30,159%
SIFT	66,934%	36,159%

7) Hasil output dari perhitungan tersebut adalah mendapatkan model SVM. Model SVM inilah yang akan digunakan sebagai proses klasifikasi. Bentuk persamaan model SVM dituliskan pada persamaan berikut.

$$f(\phi(x)) = \text{sign}(w \cdot \phi(x) + b) \quad (12)$$

Adapun flowchart Algoritma Support Vector Machine adalah sebagai berikut.

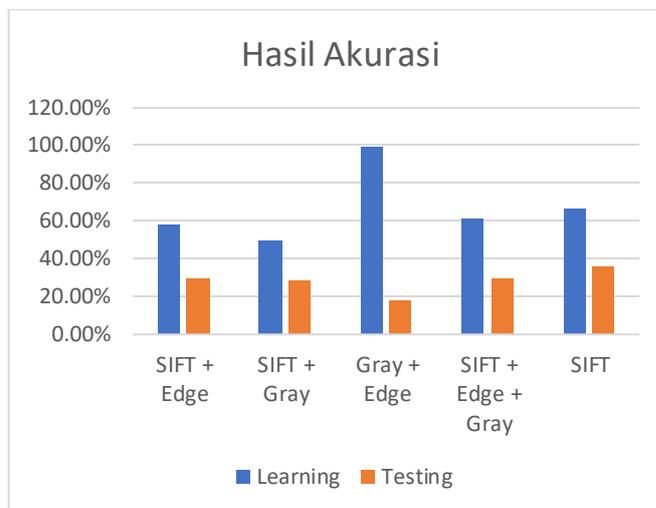


Gbr. 10 Algoritma Support Vector Machine

Proses klasifikasi menggunakan Algoritma SVM menghasilkan model SVM yang digunakan untuk proses learning dan testing. Setelah dilakukan proses learning dan testing, didapatkan hasil akurasi klasifikasi motif citra batik Solo, Adapun hasil akurasi data learning dan data testing adalah sebagai berikut.

TABEL I
HASIL AKURASI

Adapun perbandingan hasil akurasi dari kelima model disajikan pada diagram berikut.



Gbr. 11 Hasil Akurasi

KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, kesimpulan yang didapat dari keseluruhan proses dan hasil pembahasan menghasilkan sistem klasifikasi motif citra batik berdasarkan fitur Multi-Autoencoder dengan memiliki tampilan antarmuka sistem. Dengan memperhitungkan tingkat akurasi klasifikasi pada tiap model jaringan Autoencoder maka model Autoencoder yang digunakan untuk proses testing adalah arsitektur jaringan SIFT A9, dikarenakan akurasi pada jaringan ini lebih baik dari arsitektur jaringan lain maupun kombinasi jaringan Autoencoder.

Dalam beberapa percobaan yang telah dilakukan maka didapatkan tingkat akurasi yang didapatkan adalah sebesar 29,581% untuk kombinasi model Autoencoder citra Edge dan SIFT; 18,407% untuk citra Edge dan Grayscale; 29,157% untuk citra Grayscale dan SIFT; serta 30,159% untuk kombinasi citra Edge, SIFT, dan Grayscale.

UCAPAN TERIMA KASIH

Puji syukur Alhamdulillah senantiasa penulis ucapkan kepada Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan proses pembuatan jurnal ini hingga selesai. Tak lupa juga ucapan terimakasih kepada pihak-pihak yang telah membantu proses penyusunan jurnal ini, sehingga penulis dapat menyelesaikan jurnal tepat pada waktunya.

REFERENSI

- [1] Maynard, M., 2004. *Dress and Globalisation*. First ed. Manchester ; New York: Manchester University Press.

- [2] Lestari, S. . D., 2012. *Mengenal Aneka Batik*. First ed. Jalan Pulokambang Kav. J. 15 Kawasan Industri Pulogadung, Jakarta Timur: PT Balai Pustaka (Persero).
- [3] Kasim, A. A., Wardoyo, R. & Harjoko, A., 2017. Batik Classification with Artificial Neural Network Based on Texture-Shape Feature of Main Ornament. *I.J. Intelligent Systems and Applications*, 2017, 6, pp. 55-65.
- [4] Gultom, Y., Masikome, R. J. & Arymurthy, A. M., 2018. BATIK CLASSIFICATION USING DEEP CONVOLUTIONAL NETWORK TRANSFER LEARNING. *Jurnal Ilmu Komputer dan Infomasi (Journal of a Science and Information)*. 11/2 (2018), pp. 59-66.
- [5] Azhar, R. et al., 2015. Batik Image Classification Using SIFT Feature Extraction, Bag of Features and Support Vector Machine. *Procedia Computer Science* 72, pp. 24-30.
- [6] Adi, R. V. & Soelistio, Y. E., n.d. CLASSIFICATION BATIK BIRD MOTIF WITH GABOR FILTER,PCA,SVM USING BATIK TANJUNG BUMI BANGKALAN AND OTHER BATIK AS SUBJECT. *Jurnal Teknologi Informasi*, pp. 1-6.
- [7] Mabu, S., Kobayashi, K., Obayashi, M. & Kuremoto, T., 2018. Unsupervised Image Classification Using Multi-Autoencoder and K-Means++. *Journal of Robotics, Networking and Artificial Life*, Volume 5, pp. 75-78.
- [8] Lowe, D., 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer*, 60(2), pp. 91-110