Handwriting Recognition in Hanacaraka Javanese Character using Support Vector Machine and Scale Invariant Feature Transform Algorithm

Penerapan Algoritma Support Vector Machine dan Scale Invariant Feature Transform untuk Pengenalan Tulisan Tangan pada Karakter Hanacaraka Aksara Jawa

Rama Tri Agung1, Dessyanto Boedi Prasetyo2, Mangaras Yanu Florestiyanto3

1,2,3 Informatika, Universitas Pembangunan Nasional Veteran Yogyakarta, Indonesia

1\*123180053@student.upnyk.ac.id, 2dess@upnyk.ac.id, 3mangaras.yanu@upnyk.ac.id

|  |  |
| --- | --- |
| **Informasi Artikel** | **Abstract** |
| Received:  Revised:  Accepted:  Published: | Purpose:  Hanacaraka Javanese script is one of the ancestral heritage of the Indonesian nation inherent in Javanese culture. This Javanese script is threatened with extinction due to the decreasing number of users and also the interest of the community. The Javanese handwritten character recognition system is one way to help the preservation of this culture.  Design/methodology/approach:  As in previous research, the system's application can be developed using a variety of methods, including CNN, KNN, and SVM. Such methods have their own drawbacks. As a result, the SVM method and SIFT as feature extraction to improve classification performance were used in this research. This research used 2940 data with 20 classes that were augmented with seven variations and several preprocessing steps, including resize, grayscaling, and histogram equalization.  Findings/result:  Parameter optimization is accomplished by experimenting with 20 different combinations of several parameters, including image size, K value, C value, and Gamma value. The best parameters were obtained by the test, which were the image size of 192x192 pixels, the value of K = 750, the value of C = 3, and the value of Gamma = 0.10235. The accuracy performance values of the SVM-SIFT model classification using these parameters can reach 92.11% in the training data and 94.55% in the test data. The performance of such models demonstrates that SIFT feature extraction can significantly improve accuracy on SVM models.  Originality/value/state of the art:  This research conducted handwriting recognition on Javanese hancaraka characters using the SVM and SIFT as feature extraction methods. Then 2940 Javanese script hanacaraka data have been augmented with seven variations and preprocessed resizing, grayscaling, and histogram equalization before being trained in the model. |
|  | **Abstrak** |
| *Keywords:hanacaraka javanese character; support vector machine; scale invariant feature transform*  Kata kunci: hanacaraka aksara jawa; *support vector machine*; *scale invariant feature transform* | Tujuan:  Hanacaraka aksara jawa merupakan salah satu warisan leluhur bangsa Indonesia yang melekat dalam budaya Jawa. Bahasa aksara jawa ini terancam punah karena semakin berkurang jumlah penggunanya dan juga pada minat masyarakat. Sistem pengenalan karakter tulisan tangan aksara jawa menjadi salah satu cara untuk menunjang kelestarian budaya tersebut.  Perancangan/metode/pendekatan:  Penerapan sistem dapat dikembangkan dengan beberapa metode seperti yang telah dilakukan pada penelitian sebelumnya yaitu CNN, KNN, dan SVM. Metode tersebut memiliki kelemahannya masing-masing. Sehingga pada penelitian ini menggunakan metode SVM dan SIFT sebagai ekstraksi fitur untuk meningkatkan performa klasifikasi. Penilitian ini menggunakan 2940 data dengan 20 kelas yang telah dilakukan augmentasi tujuh variasi dan beberapa *preprocessing* yaitu *resize*, *grayscaling*, dan *equalization histogram*.  Hasil:  Optimasi parameter dilakukan dengan menguji pada 20 kombinasi terhadap beberapa parameter yaitu *size* gambar, nilai K, nilai C, dan nilai Gamma. Pengujian tersebut mendapatkan parameter terbaik yaitu *size* gambar 192x192 piksel, nilai K = 750, nilai C = 3, dan nilai Gamma = 0.10235. Hasil klasifikasi model SVM-SIFT menggunakan parameter tersebut dapat menghasilkan nilai performa akurasi terbaik yaitu mencapai 92.11% pada data latih dan 94.55% pada data uji. Performa model tersebut membuktikan bahwa ekstraksi fitur SIFT dapat meningkatkan akurasi yang signifikan pada model SVM.  Keaslian/ *state of the art*:  Penelitian ini melakukan pengenalan tulisan tangan pada karakter hancaraka aksara jawa dengan menggunakan metode SVM dan SIFT sebagai ekstraksi fitur. Kemudian 2940 data hanacaraka aksara jawa telah dilakukan diaugmentasi dengan tujuh variasi dan dilakukan *preprocessing* *resize, grayscaling,* dan *equalization histogram* sebelum dilatih dalam model. |

# Pendahuluan

Aksara Jawa "ha-na-ca-ra-ka" merupakan salah satu warisan leluhur bangsa Indonesia (Sari et al., 2018). Aksara Jawa juga bagian dari bahasa Jawa yang melekat dalam budaya Jawa. Ratusan bahasa daerah di Indonesia terancam punah. Salah satu bahasa daerah yang terancam adalah bahasa Jawa (Lorentius et al., 2019). Pengguna bahasa Jawa ini semakin berkurang jumlahnya dan hanya sedikit remaja yang mengenal aksara Jawa dengan jelas (Setiawan et al., 2019), karena minat masyarakat terhadap aksara jawa juga sangat memprihatinkan (Lorentius et al., 2019). Melihat kondisi tersebut, maka perlu sebuah sistem yang dapat mengenali huruf-huruf Hanacaraka Aksara Jawa (Lorentius et al., 2019). Pengenalan tulisan tangan ini berguna untuk menunjang kelestarian Aksara Jawa sebagai alat atau perangkat lunak yang memiliki kemampuan untuk mengenali tulisan tangan karakter Aksara Jawa secara otomatis (Dewa et al., 2018).

Pengenalan tulisan tangan pada karakter hanacaraka aksara jawa telah diusulkan oleh beberapa penelitian sebelumnya dengan menggunakan metode yaitu *Convolutional Neural Network* (CNN) (Dewa et al., 2018) (Rismiyati et al., 2017) (Wibowo et al., 2018), *K-Nearest Neighbor* (KNN) (Sari et al., 2018), dan *Support Vector Machine* (SVM) (Rismiyati et al., 2018). Pada metode CNN memiliki performa klasifikasi yang sangat baik dalam bidang ini dengan tingkat akurasi yang dapat mencapai 94.57% (Wibowo et al., 2018), CNN juga mahir dalam menangani inputan yang bersifat *noisy* (Rajesh et al., 2016), namun akurasi yang tinggi pada CNN membutuhkan jumlah data latih yang banyak (Wibowo et al., 2018) dan dengan metode yang kompleks ini maka arsitekturnya akan cukup sulit dibangun serta dapat terjadinya *overfitting* (Rajesh et al., 2016). KNN adalah metode yang sederhana, efektif, mudah diterapkan, tidak parametrik dan memberikan tingkat kesalahan yang rendah dalam proses pelatihan (Thamilselvana & Sathiaseelan, 2015), metode ini tidak membutuhkan jumlah *dataset* yang banyak (Sari et al., 2018), tapi kekurangannya pada metode KNN relatif memiliki hasil performa yang kurang baik daripada metode lainnya (Naufal et al., 2021) dan sulit menemukan nilai parameter optimal (Thamilselvana & Sathiaseelan, 2015). Kemudian metode SVM merupakan metode paling efektif dalam klasifikasi, terutama populer dalam klasifikasi teks, memiliki akurasi yang cukup tinggi (Thamilselvana & Sathiaseelan, 2015), tidak memilki masalah dalam *overfitting* (Rajesh et al., 2016), dan tidak membutuhkan jumlah *dataset* yang sangat besar (Rismiyati et al., 2018). Namun sayangnya, metode ini cukup sulit untuk mencari model parameter yang cocok maupun optimal dalam klasifikasi (Thamilselvana & Sathiaseelan, 2015).

Beberapa penelitian diatas, metode SVM dapat digunakan secara fleksibel tanpa membutuhkan *dataset* yang besar dan memiliki performa yang cukup baik, namun penelitian sebelumnya yang menggunakan metode tersebut belum dapat menyaingi akurasi dari metode CNN yang diatas 90% (Rismiyati et al., 2018) (Sari et al., 2018). Untuk dapat meningkatkan akurasi dibutuhkan bantuan fitur ekstraksi, pada penilitian pengenalan tulisan tangan karakter lainnya (Thailand, Bangla dan Latin) telah mengusulkan metode yaitu menggunakan *Scale Invariant Feature Transform Descriptor* (SIFT) yang berpengaruh dalam peningkatan akurasi klasifikasi secara signifikan menjadi diatas 95% dan mengungguli performa fitur ekstraksi *Histograms of Oriented Gradients* (HOG) (Surinta et al., 2015).

Sehingga pada penelitian ini pengenalan tulisan tangan pada karakter hanacaraka aksara jawa akan menerapkan metode SVM dengan bantuan SIFT dalam meningkatkan performa akurasi. Dari metode tersebut akan dibandingkan dan menemukan bagaimana pengaruh terhadap akurasi jika metode SIFT diterapkan dalam ekstraksi fitur.

# Metode/Perancangan

Pada penelitian ini terdapat metodologi penelitian dengan beberapa tahapan secara sistematis dengan menerapkan metode pengembangan *machine learning* merujuk pada penelitian oleh Rasyidi, et al. (2021), Dewa, et al. (2018), dan Schröera, et al. (2021) yang konten dari tahapan itu sendiri telah disesuaikan dengan kebutuhan dari penelitian ini. Adapun tahapan tersebut dipecah menjadi beberapa tahapan yaitu pengumpulan data, persiapan data, pemodelan data, pengujian data.

## Pengumpulan Data

Pada penelitian ini dataset karakter hanacaraka aksara jawa diambil dari *website* Kaggle oleh Phiard yang berjumlah 2632 karakter dengan 20 jenis karakter didalamnya. Namun pada dataset tersebut telah dilakukan augmentasi sejumlah 6 varian setiap gambarnya termasuk varian normal. Sayangnya augmentasi yang dilakukan oleh Phiard tidak signifikan terlihat berbeda dengan gambar asli nya sehingga pada penelitian ini hanya mengambil gambar yang normal sebanyak 420 karakter dengan 20 jenis karakter tanpa augmentasi oleh Phiard. Data gambar yang telah dikumpulkan tersebut merupakan gambar yang berwarna hitam putih dengan ukuran 224x224 piksel.

## Persiapan Data

Setelah mengetahui kebutuhan data yang digunakan, pada tahap persiapan data yaitu data akan diolah terlebih dahulu sebelum masuk pada tahapan pemodelan data, sehingga tahap ini akan menghasilkan keluaran sebuah fitur yang sudah rapi dan siap dilatih. Tahapan ini akan dibagi menjadi tiga tahapan penting yaitu augmentasi data & *preprocessing* gambar, ekstraksi fitur, dan *preprocessing* fitur.

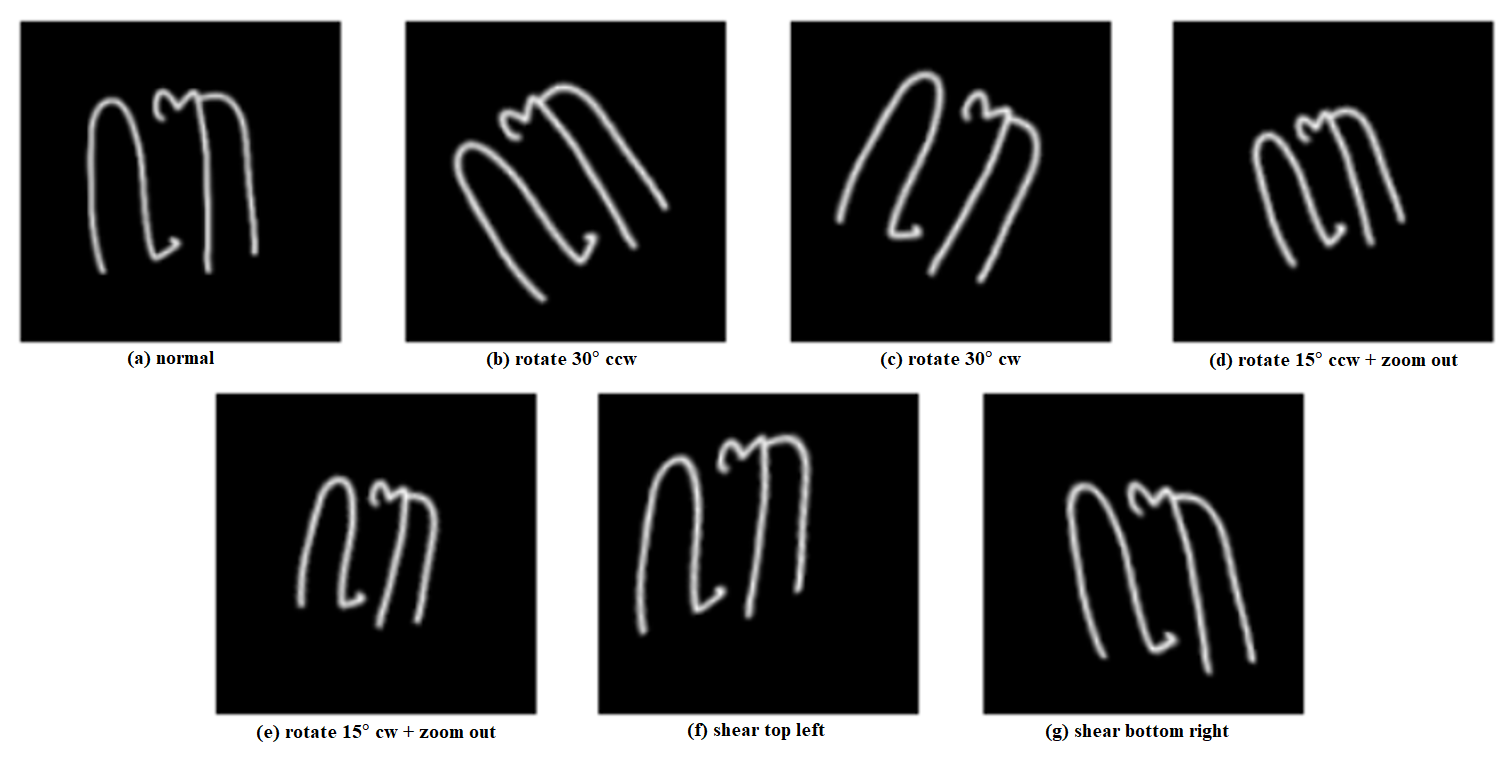
### Augmentasi Data & *Preprocessing* Gambar

Augmentasi data adalah strategi yang memungkinkan praktisi untuk secara signifikan meningkatkan keragaman data yang tersedia untuk model pelatihan, tanpa benar-benar mengumpulkan data baru (Sanjaya & Ayub, 2020). Terdapat banyak metode augmentasi yang dapat dilakukan, namun pada penelitian ini menggunakan tiga metode augmentasi gambar yaitu rotasi gambar, pengecilan gambar, dan *shear* gambar. Penggunaan ketiga metode tersebut masih tetap menjaga kualitas gambar yang ada dan tidak membuat gambar menjadi rusak ataupun sulit dilihat sehingga variasi *dataset* masih layak untuk digunakan. Setelah augmentasi dilakukan maka jumlah *dataset* menjadi 2940 dengan tujuh variasi gambar. Beberapa varian hasil augmentasi yang dilakukan dapat dilihat pada **Tabel 1.**

Selain itu *preprocessing* gambar dilakukan untuk memperbaiki gambar dan menyediakan data gambar yang terbaik. Beberapa hal yang dilakukan pada tahapan ini yaitu mengganti ukuran gambar, mengganti warna gambar menjadi hitam putih dan melakukan perataan histogram pada gambar. Ukuran gambar diganti menjadi 192x192 piksel, kemudian gambar diubah menjadi hitam putih dengan menggunakan metode weighted grayscaling, setelah itu proses terakhir dilakukan dengan meratakan histogram pada sebuah matriks gambar agar membuat gambar menjadi lebih tajam. Berikut **Gambar 1** merupakan hasil dari semua proses pengolahan gambar beserta hasil augmentasinya.

**Tabel 1**. Variasi Hasil Augmentasi Gambar

|  |  |
| --- | --- |
| No. | Varian Gambar |
| 1. | Normal |
| 2. | Rotasi 30 derajat berlawanan jarum jam |
| 3. | Rotasi 30 derajat searah arah jarum jam |
| 4. | Rotasi 15 derajat berlawanan jarum jam dan pengecilan gambar |
| 5. | Rotasi 15 derajat searah arah jarum jam dan pengecilan gambar |
| 6. | *Shear* gambar kearah atas kiri |
| 7. | *Shear* gambar kearah bawah kanan |



**Gambar 1.** Hasil Augmentasi dan Preprocessing Gambar pada Karakter ‘ba’

### Ekstraksi Fitur

Dalam penelitian ini ekstraksi fitur menggunakan *Scale Invariant Feature Transform* (SIFT). Deskriptor SIFT menciptakan lingkungan 16x16 yang dipartisi menjadi 16 subwilayah masing-masing 4x4 piksel. Untuk setiap piksel dalam subwilayah, SIFT menambahkan vektor gradien piksel ke histogram arah gradien dengan mengkuantisasi setiap orientasi ke salah satu dari 8 arah dan memberi bobot kontribusi setiap vektor berdasarkan besarnya. Setiap arah gradien selanjutnya dibobot dengan skala Gaussian = n/2 di mana n adalah ukuran lingkungan dan nilai-nilai didistribusikan ke bin tetangga menggunakan interpolasi trilinear untuk mengurangi efek batas saat sampel bergerak di antara posisi dan orientasi (Mortensen et al. 2005). Namun dalam penerapannya, hasil fitur yang diekstrak oleh SIFT pada gambar tidak dapat digunakan langsung kedalam model *machine learning*, karena *input* yang diharapkan oleh model berbeda dengan *output* yang dikeluarkan oleh SIFT, hasil ekstraksi SIFT akan berukuran dinamis sesuai dengan banyaknya fitur *keypoint* terhadap gambar sedangkan model *machine learning* membutuhkan *input* fitur yang statis. Sehingga dalam penerapannya hasil fitur dari SIFT akan diolah kembali ke dalam bentuk *Bag of Feature* (BoF) atau *Bag of Words* (BoW) dengan bantuan metode K-Means dalam klasterisasi seluruh fitur dari semua data gambar yang telah diekstrak. Kemudian fitur akhir akan dihasilkan dengan cara mencari nilai klaster menggunakan K-Means terhadap BoW yang telah dibuat dengan jumlah fitur sebanyak jumlah klaster yang ditentukan. Fitur akhir itulah yang akan menjadi input model yang dibangun. Maka dari itu proses ekstraksi fitur ini akan dibagi menjadi tiga proses utama, yaitu ekstraksi deskriptor dari SIFT, membuat BoW dari deskriptor menggunakan K-Means, dan membuat fitur akhir dari BoW.

1. Ekstraksi deskriptor dari SIFT

Ada empat langkah utama yang terlibat dalam algoritma ekstraksi fitur SIFT yaitu *scale-space peak selection*, *keypoint localization*, *orientation assignment*, dan *keypoint descriptor* (Wang et al., 2013) (Hassan et al., 2018). Pertama *scale-space peak selection* dilakukan dengan mendeteksi *points of interest*, atau yang disebut juga *keypoint* pada SIFT. Dambar asli diambil dan menghasilkan urutan gambar blur terus-menerus, kemudian mengubah ukuran gambar asli menjadi 50% dari ukurannya dan menghasilkan gambar blur berulang kali. Citra dikonvolusikan dengan filter *Gaussian* pada skala yang berbeda, kemudian diambil perbedaan dari citra *Gaussian* yang berurutan. *Keypoint* kemudian diambil sebagai maxima/minima dari *Difference of Gaussians* (DoG) yang terjadi pada beberapa skala. Berikut persamaan DoG.

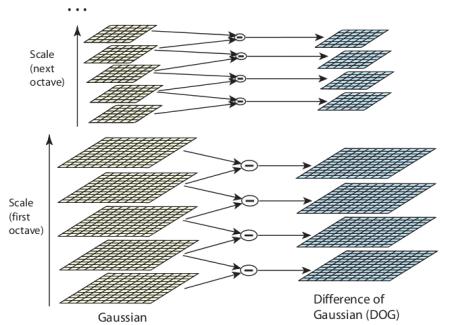
(1)

Dimana adalah konvolusi dari citra asli dengan *Gaussian Blur* dalam skala yaitu.

(2)

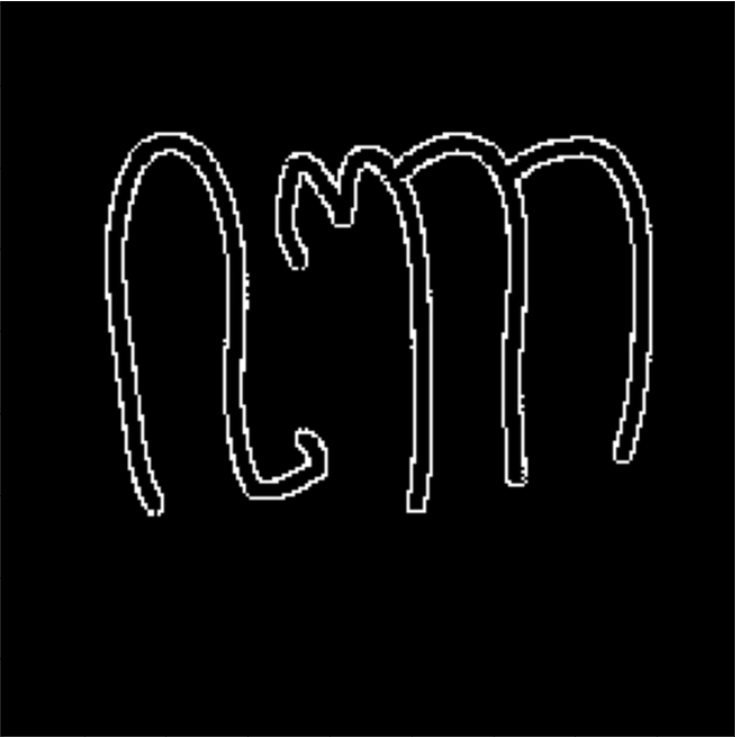
(3)

Nilai merupakan nilai kontinu dari beberapa nilai skala yang berbeda-beda untuk membandingkan beberapa hasil dari pengurangan konvolusi *Gaussian Blur*. Pada penelitian ini nilai sigma yang digunakan yaitu dan nilai .



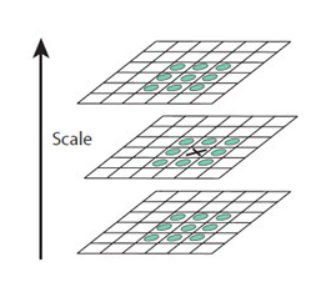
**Gambar 2**. *Scale Space* dan DoG

Salah satu contoh hasil gambar perhitungan DoG yang dilakukan dapat dilihat pada **Gambar 3**.



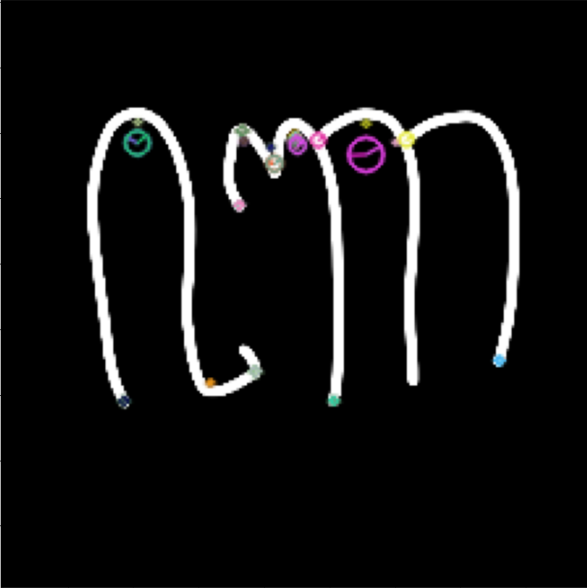
**Gambar 3.** Hasil Perhitungan DoG

Setelah gambar DoG diperoleh, *keypoints* diidentifikasi sebagai *local minima/maxima* dari gambar DoG di seluruh skala. Hal ini dilakukan dengan membandingkan setiap piksel dalam gambar DoG dengan delapan tetangganya pada skala yang sama dan sembilan piksel tetangga yang sesuai di setiap skala tetangga. Jika nilai piksel adalah maksimum atau minimum di antara semua piksel yang dibandingkan, maka dipilih sebagai *candidate* *keypoint*.



**Gambar 4.** Proses pencarian kandidat *keypoint* pada tiap skala

Setelah itu, *keypoint* tersebut akan tersebar disetiap adanya extrema, sehingga banyak *keypoint* yang terbentuk tidak akurat atau tidak tepat karena tidak adanya batasan ataupun *threshold* dalam pencarian sebelumnya. Maka selanjutnya *keypoint localization* dilakukan untuk pencocokan detail fitur ke data terdekat untuk lokasi, skala, dan rasio kelengkungan utama yang akurat. Pertama, untuk setiap titik kunci kandidat, interpolasi data terdekat digunakan untuk menentukan posisinya secara akurat. Interpolasi ini dilakukan menggunakan *Quadratic Taylor Expansion* dari fungsi skala DoG, dengan kandidat *keypoint* sebagai titik asal. Kedua, untuk membuang *keypoint* yang memiliki kontras rendah, maka nilai *Taylor Expansion* orde kedua dihitung pada suatu *offset* tertentu. Terakhir, perlu menghilangkan *keypoint* yang kurang tepat terhadap sejumlah *noise* yang kecil. Pada setiap tepi objek terdapat nilai kelengkungan utama untuk menjadi *keypoint* yang stabil, untuk mengatasinya dapat dilakukan dengan memecahkan nilai *eigenvalues* pada matriks *Hessian* orde kedua. Hasil tahapan ini dilihat pada **Gambar 5**.



**Gambar 5.** Hasil identifikasi *interest keypoints*

Selanjutnya setiap *keypoint* diberikan satu atau lebih orientasi berdasarkan arah gradien gambar lokal. Ini adalah langkah kunci dalam mencapai invarian terhadap rotasi karena *deskriptor keypoint* dapat direpresentasikan relatif terhadap orientasi ini dan karenanya mencapai invarian terhadap rotasi gambar. Besaran gradien dan orientasi diperoleh dengan menggunakan persamaan. Besaran dan orientasi dihitung untuk semua piksel yang mengelilingi titik-titik kunci. Setelah itu, histogram dapat dibuat.

(4)

(5)

Citra yang dihaluskan *Gaussian* pada skala *keypoint* diambil sehingga semua komputasi dilakukan dengan cara *scale-invariant*. Untuk sampel gambar pada skala , besaran gradien , dan orientasi , dihitung terlebih dahulu menggunakan rumus diatas.

Terakhir, menghitung vektor deskriptor untuk setiap *keypoints* sedemikian rupa sehingga deskriptor sangat unik/berbeda dan sebagian tidak berubah untuk variasi yang tersisa seperti iluminasi, sudut pandang 3D, dll. Dengan representasi ini, dimungkinkan dapat dengan mudah memperoleh fitur yang diperlukan. Untuk melakukanya, matriks 16x16 yang disekitar *keypoints* diatur dan matriks ini dibagi menjadi 16 matriks ukuran 4x4. Di dalam setiap matriks 4x4, besaran gradien dan orientasi dapat diperoleh. Histogram ini dibagi menjadi delapan bin dan jumlah orientasi yang ditambahkan ke bin tergantung pada besaran gradien. Sehingga, setiap titik kunci dideskripsikan oleh 4\*4\*8=128 dimensi vektor.

1. Membuat BoW dari deskriptor menggunakan K-Means

Pada tahapan ini seluruh fitur deskriptor dari setiap gambar dikumpulkan dalam satu data yang digunakan untuk membuat *Bag of Words* (BOW) dengan menerapkan metode *K-Means Clustering*. BoW menjadi kamus dari seluruh fitur penting yang telah di ekstrak sebelumnya. Data berukuran dimana N adalah jumlah gabungan fitur seluruh citra dan 128 adalah ukuran hasil deskriptor pada setiap fitur. Penggunaan metode K-Means dilakukan untuk mencari satu kelompok/klaster yang dapat mewakili fitur-fitur yang berdekatan menjadi fitur penting. Seluruh nilai *centroids* dari hasil perhitungan K-Means menjadi fitur yang disimpan dalam BoW. Berikut beberapa persamaan yang dibutuhkan pada metode K-Means menggunakan *Euclidean Distance*.

(6)

Selanjutnya, kelompokkan data-data yang menjadi anggota pada setiap *cluster*. Nilai pusat *cluster* yang baru dapat dihitung dengan cara mencari nilai rata-rata dari data-data yang menjadi anggota pada *cluster* tersebut menggunakan rumus berikut.

(7)

Dimana 𝑥𝑖 ∈ *cluster* ke-k dan p adalah banyaknya anggota *cluster* ke-k. Perhitungan dilakukan berulang hingga sudah tidak ada lagi data yang berpindah ke *cluster* yang lain.

1. Membuat fitur akhir dari BoW

Tahapan ekstraksi fitur terakhir membentuk fitur akhir pada setiap citra dari BoW. Pembentukan fitur ini dilakukan mirip seperti proses sebelumnya yaitu melakukan ekstraksi pada sebuah citra menggunakan SIFT untuk mendapatkan beberapa deskriptor unik citra tersebut, kemudian dari fitur deskriptor tersebut dilakukan perhitungan jarak *Euclidean Distance* pada **Persamaan 6** terhadap BoW untuk mendapatkan fitur akhir dari citra tersebut. Fitur dari citra tersebut sama dengan ukuran klaster yang sebelumnya dilakukan yaitu berukuran 750 fitur, isinya merupakan jumlah deskriptor pada klaster-klaster terdekat diantara 750 *centroids* (BoW).

### *Preprocessing* Fitur

Pada tahap ini *feature scaling* dan *split* *dataset* akan dilakukan untuk menyediakan data dengan distribusi yang baik, sehingga perlakuan pada *training* model dapat diterapkan dengan lebih baik. *Feature scaling* dilakukan dengan metode normalisasi *MinMaxScaling* yang mengubah nilai menjadi rentang 0-1. Selanjutnya *splitting* *dataset* dilakukan dengan rasio perbandingan 85% : 15%, Sehingga data terpecah menjadi 2499 data latih dan 441 data uji*.*

## Pemodelan Data

Penelitian ini menggunakan SVM dalam pembuatan model *machine learning*. *Support Vector Machine* (SVM), adalah salah satu algoritma pembelajaran mesin terbaik, yang diusulkan pada 1990-an dan sebagian besar digunakan untuk pengenalan pola (Pradhan, 2012). Pada penelitian ini kernel yang digunakan adalah *Radial Basis Function* (RBF). Kernel RBF cocok digunakan dalam kelas yang banyak dan fitur yang banyak. Model yang optimal dari himpunan *hyperplanes* di *data training* dihitung dengan algoritma optimasi SVM (Surinta et al., 2015). Berikut rumus kernel RBF yang digunakan.

(8)

(9)

Dalam kasus klasifikasi *multiclass*, perhitungan dilakukan sebanyak *K* kelas sehingga nilai prediksi akhir berbentuk vektor dengan panjang *K*. Dan nilai bobot akan sebanyak *K\*n* fitur.

Algoritma pelatihan SVM ini menggunakan metode *sequential learning* yang sederhana dan tidak banyak memakan waktu (Harahap et al., 2018). Berikut langkah-langkah pelatihan yang dilakukan (Wijayanti et al., 2018).

1. Inisialisasi pada σ1 lalu parameter selanjutnya, seperti .
2. Menemukan nilai matrik Hessian dengan menggunakan persamaan.

(10)

1. Dimulai dari nilai data *i* sampai dengan data *j*, dijelaskan dalam persamaan dibawah ini:

(11)

(12)

(13)

1. Proses ke 3 diulang terus sampai mendapatkan nilai iterasi maksimal ( ≤ ).
2. Selanjutnya mendapatkan nilai *support vector*, misalkan *threshold* .

Setelah *training* SVM dilakukan maka hasil parameter alfa akan dihasilkan untuk digunakan dalam proses klasifikasi, proses klasifikasi yang dilakukan membutuhkan nilai alfa dan nilai bias, berikut langkah-langkah yang dilakukan.

1. Langkah awal yang dilakukan yaitu mencari nilai bias.

(14)

1. Kemudian dengan nilai alfa dan bias, maka klasifikasi dapat dilakukan dengan langkah berikut.

(15)

Dalam penelitian ini, SVM *multiclass* diterapkan dengan metode *one-against-rest (one-vs-all)*. Metode ini melakukan perlakuan terhadap masing-masing kelas dengan membandingkan seluruh sisa kelasnya, apabila terdapat *k* kelas proses klasifikasi dilakukan sebanyak *k* kelas tersebut dengan membandingkan kelas pertama dengan seluruh kelas lainnya begitu pula perlakuan yang sama dilakukan pada kelas lainnya. Ketika salah satu kelas akan dilatih maka kelas lainnya akan digabung menjadi nilai negatif dari kelas yang dituju.

## Pengujian Data

Pada penelitian ini pengujian berfokus pada performa akurasi terhadap model klasifikasi. Pengujian ini dilakukan dengan melakukan optimasi *hyperparameter tuning* untuk membantu model menemukan parameter terbaik agar mendapatkan kinerja yang maksimal. *Hyperparameter tuning* adalah proses mencari nilai parameter optimal dengan menentukan daftar parameter dan rentang pencarian untuk setiap parameter (Andini et al., 2022). Ukuran gambar, nilai K pada K-Means, nilai C pada SVM, dan nilai Gamma pada RBF menjadi skenario pengujian optimasi parameter. Pada parameter ukuran gambar terdapat 4 nilai yang akan dikombinasikan dalam satuan piksel yaitu 128x128, 160x160, 192x192, dan 224x224. Kemudian pada parameter nilai K terdapat 5 nilai yang akan dikombinasikan yaitu 180, 250, 500, 750, dan 1000. Selanjutnya pada parameter nilai C dan G memiliki nilai yang dinamis menyesuaikan ukuran dan distribusi masing-masing data.

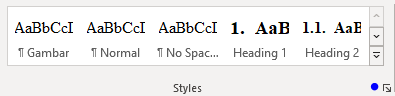
# Hasil dan Pembahasan

Hasil dan pembahasan memuat hasil penelitian dan pembahasana terkait hasil penelitian tersebut. Setiap gambar tabel yang ditampilkan harus disertai penjelasan agar pembaca bisa memahami isi dari gambar maupun tabel tersebut. Penjelasan terkait data yang disajikan harus disampaikan pada bagian ini dengan tujuan untuk memperjelas kegunaan data pada penelitian.

## Cara menampilkan style

Untuk menampilkan style apa saja yang digunakan pada template ini (bisa dilakukan pada MS Office Word 2010 dan versi setelahnya), bisa perhatikan langkah-langkah berikut ini:

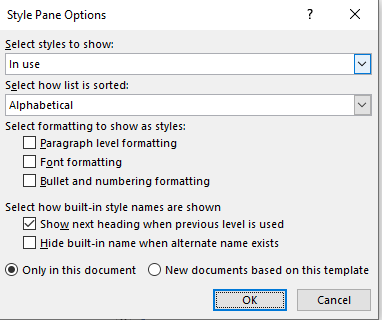
1. Pilih menu **Home** pada toolbar
2. Pada ribbon **Styles**, klik icon arah panah ke pojok kanan bawah seperti diperlihatkan pada **Gambar 2.**



**Gambar 2.** *Ribbon Styles*

Kemudian akan muncul jendela styles pada bagian kanan layar kerja.

1. Pada jendela styles, pilih menu **options**. Kemudian akan muncul jendela menu options.
2. Pada jendela menu options, buatlah pengaturan seperti diperlihatkan pada **Gambar 3**.



**Gambar 3.** *Style Pane*

1. Maka pada jendela styles akan muncul semua jenis style yang digunakan pada template ini.
2. Petunjuk ini berlaku bagi pengguna MS Office Word 2010 – Word 365.
3. Petunjuk nomor 1-7 dibuat menggunakan style **List Paragraph**

## Sitasi dan Referensi

Sitasi dan refrensi yang digunakan mengacu pada **IEEE Style.** Untuk mempermudahpenulisan, disarankan untuk menggunakan *software* pendukung untuk sitasi dan referensi seperti Mendeley, Zotero, EndNote, dll.

Contoh penggunaan sitasi dengan menggunakan IEEE style adalah sebagai berikut.

Contoh sitasi terhadap satu artikel: The feature extraction method applied in the pre-processed image is the Gray Level Co-occurrence Matrix (GLCM) with six parameters (Entropy, Angular Second Moment, Contrast, Inverse Different Moment, Correlation, and Variance) [1].

Contoh sitasi terhadap dua artikel: Pengumpulan data dilakukan dengan menggunakan API untuk kemudian diterapkan pada sebuah perusahaan [2], [3].

Contoh sitasi terhadap lebih dari dua artikel berurutan: Dalam proses pembersihan data dilakukan penghapusan data yang tidak lengkap atau tidak sesuai dengan atribut yang akan digunakan dalam penelitian. Sementara pada tahap transformasi, data diubah menjadi adjacency matrix yang kemudian dianalisis menggunakan SNA [3]–[5].

Contoh sitasi terhadap lebih dari dua artikel yang tidak semuanya berurutan: Sistem informasi skripsi yang saat ini digunakan oleh program studi Teknik Informatika UPN “Veteran” Yogyakarta masih menggunakan cara konvensional, sehingga ditemui kesulitan dalam proses pemeriksaan kemiripan dokumen yang ada dalam proposal tugas akhir mahasiswa. Pemeriksaan kemiripan dokumen dapat dilakukan dengan menggunakan algoritme Winnowing [1], [3]–[5].

## Syarat referensi

Pustaka atau referensi utama harus berasal dari artikel yang diterbitkan pada Jurnal Ilmiah dengan jumlah **minimal 15 referensi utama**. Referensi utama **harus** terbit dalam rentang waktu **10 tahun terakhir** dari pengajuan artikel.

# Kesimpulan dan Saran

Kesimpulan yang ditulis harus bisa menjawab tujuan penelitian. Pada bagian ini juga harus disertai saran untuk penelitian selanjutnya. Kesimpulan dan saran ditulis dalam bentuk paragraf dan bukan dalam bentuk penomoran.

# Daftar Pustaka

[1] S. Saifullah and V. A. Permadi, “Comparison of Egg Fertility Identification based on GLCM Feature Extraction using Backpropagation and K-means Clustering Algorithms,” in *Proceeding - 2019 5th International Conference on Science in Information Technology: Embracing Industry 4.0: Towards Innovation in Cyber Physical System, ICSITech 2019*, 2019, pp. 140–145.

[2] S. P. Tahalea and A. SN, “Central Actor Identification of Crime Group using Semantic Social Network Analysis,” *Indones. J. Inf. Syst.*, vol. 2, no. 1, p. 24, 2019.

[3] Hermantoro, A. P. Suryotomo, A. I. Uktoro, and R. A. Renjani, “Unmanned Aerial Vehicle Application for Plantation Mapping and Automatic Oil Palm Trees Counting on Oil Palm Plantation Management,” in *International Conference on the Role of Agricultural Engineering for Sustainable Agriculture Production*, 2016, no. December, pp. 47–50.

[4] Awang Hendrianto Pratomo, W. Kaswidjanti, and S. Mu’arifah, “Implementasi Algoritma Region of Interest ( ROI ) Untuk Meningkatkan Performa Algoritma Deteksi Dan Klasifikasi Kendaraan,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 1, pp. 155–162, 2020.

[5] S. P. Tahalea, “Identifikasi Peran Hero DOTA2 Menggunakan Social Network Analysis,” *TEKNOMATIKA*, vol. 12, no. 2, pp. 81–86, 2020.