

Klasifikasi Ngengat dan Kupu-Kupu Menggunakan Metode GLCM dan Support Vector Machine

I Dewa Made Mardana^{a1}, Luh Gede Astuti^{a2}

^aProgram Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
¹dewamardana33@gmail.com
²lg.astuti@unud.ac.id (Corresponding Author)

Abstract

Butterflies and moths are two types of insects that share similarities in their appearance and physical characteristics. Both insects exhibit a variety of colors, patterns, and body shapes that are often difficult to distinguish. This research aims to classify butterflies and moths using feature extraction from the Gray-Level Co-occurrence Matrix. The feature extraction process involves extracting values such as correlation, homogeneity, contrast, and energy from angles of 0°, 45°, 90°, and 135° in each butterfly and moth image. Furthermore, the Support Vector Machine method is used for classification. The research results indicate that using feature extraction from the Gray-Level Co-occurrence Matrix and the Support Vector Machine method can achieve an accuracy of 68.11%, with precision, recall, and F1-Score values of 70.0%, 68.0%, and 68.0%, respectively.

Keywords: Classification, Gray-Level Co-occurrence Matrix, Feature extraction, Support Vector Machine, Butterflies, Moths

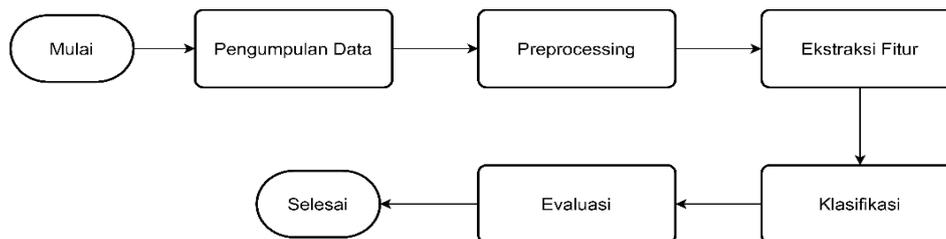
1. Pendahuluan

Akhir akhir ini penelitian di bidang citra mengalami perkembangan yang sangat pesat, Penelitian di bidang citra meliputi pendeteksian objek, analisis emosi wajah dan klasifikasi. Klasifikasi merupakan sebuah penelitian yang dilakukan untuk membedakan antara dua atau lebih objek, klasifikasi bertujuan untuk mempermudah membedakan sebuah objek tertentu seperti membedakan jenis cengkeh berdasarkan tekstur daun [1], klasifikasi penyakit pada padi berdasarkan daun [2] dan klasifikasi jenis kupu – kupu [3]. Ngengat adalah sebuah serangga yang sangat mirip dengan kupu – kupu, cukup sulit membedakan ngengat dan kupu – kupu sehingga ini menjadi landasan penelitian ini untuk melakukan klasifikasi untuk membedakan ngengat dan Kupu – kupu. Untuk melakukan klasifikasi kupu – kupu dan ngengat perlu dilakukan analisis untuk mencari struktur pola supaya bisa dilakukan klasifikasi. *Gray-Level Co-occurrence Matrix (GLCM)* adalah metode yang digunakan untuk mengekstraksi fitur tekstur dari citra, yang berfokus pada hubungan antara intensitas piksel dalam citra [4]. Metode ini bekerja dengan menghitung frekuensi munculnya pasangan nilai *grayscale* pada citra yang berdekatan. Hasil analisis ini kemudian digunakan sebagai fitur untuk *Support vector machine (SVM)*, yang merupakan algoritma klasifikasi yang dapat digunakan untuk mengklasifikasikan objek berdasarkan fitur-fitur yang diperoleh dari citra. *Support vector machine* merupakan metode yang bekerja dengan cara mendefinisikan batas antara dua kelas dengan jarak maksimal dari data yang terdekat, jarak maksimal ini didapatkan dengan menemukan *hyperplane* (garis pemisah) terbaik pada input *space* yang diperoleh dengan mengukur margin *hyperplane* [5]. Metode ini bekerja dengan mencari *hyperplane* yang terbaik untuk memisahkan kelas-kelas yang berbeda. SVM dapat digunakan dengan berbagai jenis kernel, seperti kernel *linear*, *polynomial*, dan *radial basis function (RBF)*, yang memungkinkan penggunaan metode ini dalam berbagai aplikasi. Penggunaan metode *Gray-Level Co-occurrence Matrix* dan *Support Vector Machine* dalam proses klasifikasi sudah pernah dilakukan seperti pada penelitian mengenai klasifikasi daging menggunakan metode *Gray-Level Co-occurrence Matrix* dan *Support Vector Machine* [6], yang dimana menggunakan metode *Gray-Level Co-occurrence Matrix* untuk menganalisis struktur pola pada citra daging sedangkan metode SVM digunakan untuk memprediksi jenis daging

berdasarkan fitur-fitur yang diperoleh dari metode *Gray-Level Co-occurrence Matrix*. Pada penelitian ini dapat menunjukkan tingkat pengenalan yang juga tinggi, dengan menghasilkan nilai akurasi sebesar 87,5%. Oleh karena itu pada penelitian ini akan dilakukan penggunaan metode *Gray-Level Co-occurrence Matrix (GLCM)* dan *Support Vector Machine (SVM)* untuk klasifikasi antara kupu – kupu dan ngengat.

2. Metode Penelitian

Dalam upaya mencapai hasil terkait klasifikasi Ngengat dan Kupu-kupu menggunakan *Gray-Level Co-occurrence Matrix*, penelitian ini mengikuti serangkaian langkah-langkah yang telah ditetapkan. Langkah-langkah penelitian tersebut meliputi tahap-tahap seperti pengumpulan data, *preprocessing* data, ekstraksi fitur menggunakan *Gray Level Co-occurrence Matrix*, pemilihan fitur yang relevan, pelatihan model klasifikasi, dan evaluasi yang digambarkan sebagai berikut:



Gambar 1. Alur Penelitian

Seluruh tahapan ini akan dilakukan dengan menggunakan virtual komputer pada *google Colab* yaitu sebuah *executable document* yang dapat digunakan untuk menyimpan, menulis, serta membagikan program yang telah ditulis melalui *Google Drive* [7].

2.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data sekunder yang diambil dari website *Kaggle* [8], data yang diambil merupakan data citra, data citra yang digunakan dibagi menjadi dua jenis yaitu data citra kupu – kupu yang terdiri dari 5 jenis kupu – kupu sebanyak 602 data dan data citra ngengat yang terdiri 5 jenis ngengat sebanyak 664 data. Sehingga dalam penelitian ini menggunakan 1266 data citra. Contoh dataset yang digunakan ditunjukkan pada gambar 2.



Gambar 2. Contoh Dataset

2.2 Preprocessing

Preprocessing merupakan langkah awal dalam pemrosesan citra, yang bertujuan untuk mempersiapkan data dengan membersihkannya agar siap untuk tahapan selanjutnya. Dalam penelitian ini, terdapat tiga tahapan utama dalam proses *preprocessing*. Pertama, dilakukan *Resize* ke ukuran 260 x 160 piksel untuk menstandarisasi proporsi citra dalam dataset, Ukuran ini dipilih berdasarkan Hasil tertinggi diperoleh pada dataset citra dengan ukuran gambar

256x160 piksel [3], sehingga semua citra memiliki ukuran yang seragam. Untuk melakukan *resize* pada dataset menggunakan *library* dari *OpenCV*, *OpenCV* adalah sebuah *library* open-source yang dikembangkan oleh Intel yang fokus untuk menyederhanakan pemrograman terkait citra digital [9]. Kedua, dilakukan proses *Cropping* untuk menghilangkan background yang tidak relevan atau mengganggu, sehingga fokus analisis lebih terarah pada objek yang diinginkan. Untuk proses *Cropping* menggunakan *rembg*, *rembg* adalah perpustakaan yang digunakan untuk menghapus latar belakang dari gambar [10]. Selanjutnya, dilakukan konversi ke *grayscale* untuk mengubah citra menjadi skala keabuan menggunakan *OpenCV* [9], Hasil dari proses ini memungkinkan representasi yang lebih sederhana namun tetap mempertahankan informasi penting dalam citra. Terakhir akan dilakukan pemberian label pada data kupu – kupu dan Ngentat.

2.3 Ekstraksi Fitur

Dalam penelitian ini, ekstraksi fitur dilakukan menggunakan *Gray-Level Co-occurrence Matrix* (GLCM), metode yang digunakan untuk mengekstraksi fitur tekstur dari citra, yang berfokus pada hubungan antara intensitas piksel dalam citra [4]. GLCM digunakan untuk merepresentasikan hubungan antara dua piksel yang bertetangga, berdasarkan intensitas keabuan, jarak, dan sudut di antara mereka. Dalam penelitian ini Terdapat empat sudut yang akan digunakan, yaitu 0°, 45°, 90°, 135°. Ekstraksi fitur ini menghasilkan beberapa metrik seperti *correlation*, *homogeneity*, *contrast*, dan *energy*, yang kemudian akan digunakan sebagai parameter dalam proses pengklasifikasian. Proses ekstraksi fitur dengan menggunakan *Gray-Level Co-occurrence Matrix* (GLCM) ini menggunakan *library* dari *skimage.feature* yaitu module *greycomatrix* untuk mendapatkan GLCM matrix, dan *greycomprops* untuk menghitung metrik pada GLCM [11].

2.4 Klasifikasi

Klasifikasi merupakan proses pengelompokan objek ke dalam kategori atau kelas yang telah ditentukan sebelumnya. Dalam proses ini, model klasifikasi mempelajari dan mengenali pola yang ada pada setiap data yang tersedia. Pola-pola ini kemudian digunakan sebagai acuan untuk mengklasifikasikan data baru yang masuk. Dalam pengembangan model, algoritma *Support Vector Machine* (SVM) diterapkan sebagai metode klasifikasi. SVM adalah metode klasifikasi yang mampu memisahkan dua kelas dengan mencari *hyperplane* terbaik. Dengan menggunakan SVM, model dapat mempelajari pola-pola yang kompleks dan mampu mengklasifikasikan data baru. Dataset dibagi menjadi dua bagian, di mana 80% digunakan sebagai data latih dan 20% sebagai data uji. Untuk Pemodelan *Support Vector Machine* Menggunakan *library* dari *sklearn.svm* [12].

2.5 Evaluasi

Evaluasi kinerja sebuah model klasifikasi sangat penting untuk memahami sejauh mana model tersebut dapat melakukan prediksi dengan akurat. Salah satu metode evaluasi yang umum digunakan adalah *Confusion Matrix*, yang merupakan tabel visualisasi performa model berdasarkan perbandingan antara nilai aktual dan nilai prediksi. *Confusion Matrix* memberikan informasi tentang akurasi, *precision*, *recall*, dan *F1-score*.

3. Hasil dan Diskusi

Berdasarkan tahapan penelitian yang telah dilakukan, diperoleh hasil dari setiap tahapannya, yang meliputi Pengumpulan Data, *preprocessing*, ekstraksi ciri, identifikasi, dan akurasi. Selanjutnya, tahapan penelitian yang tergambar pada Gambar 1 akan dianalisis secara menyeluruh untuk mengevaluasi keakuratan metode yang digunakan, yaitu *Gray Level Co-occurrence Matrix* (GLCM) dan *Support Vector Machine* (SVM).

3.1 Preprocessing

Dataset yang sudah didapatkan ini kemudian akan di proses pada tahapan *preprocessing*. Tahapan *preprocessing* yang dilakukan melibatkan beberapa langkah. Langkah pertama adalah melakukan *resize* data citra ke ukuran 256 x 160 untuk menstandarisasi ukuran citra. Selanjutnya,

dilakukan proses *cropping*. *Cropping* dilakukan untuk menghilangkan bagian latar belakang yang tidak diperlukan, sehingga fokus analisis lebih terarah pada objek yang diinginkan. Setelah itu, dilakukan konversi data citra ke citra *grayscale* untuk memudahkan representasi visual dalam skala keabuan.

```
resize = cv2.resize(img_np, (256,160))  
img_without_bg = remove(resize)  
gray = cv2.cvtColor(img_without_bg, cv2.COLOR_BGR2GRAY)
```

Gambar 3. kode *Resize*, *Cropping* dan *grayscale* pada data citra

Data Citra yang sudah di konversi ke citra *grayscale* akan diberikan label, data citra kupu – kupu akan diberikan label 1, sedangkan Ngengat akan diberikan label 2. Hasil dari tahapan preprocessing ini dapat dilihat pada gambar 3.



Gambar 4. Citra Ngengat setelah melalui Preprocessing

3.2 Ekstraksi Fitur

Dalam penelitian ini, ekstraksi fitur dilakukan menggunakan metode Gray-Level Co-occurrence Matrix. Citra yang digunakan sebagai inputan telah melalui proses *preprocessing* sebelumnya. Proses ekstraksi fitur dimulai dengan pembentukan matriks *co-occurrence* dari setiap citra. Selanjutnya, fitur ekstraksi GLCM seperti *correlation*, *homogeneity*, *contrast*, *energy* dihitung dari matriks *co-occurrence* dengan menggunakan empat sudut yang berbeda, yaitu 0°, 45°, 90°, dan 135°. Beberapa contoh hasil ekstraksi fitur sudut 0° ditunjukkan pada Tabel 1.

```
from skimage.feature import greycomatrix, greycoprops  
  
# ----- calculate greycomatrix() & greycoprops() for angle 0, 45, 90, 135 -----  
def calc_glcm_all_agls(img, label, props, dists=[1], agls=[0, np.pi/4, np.pi/2, 3*np.pi/4], lvl=256, sym=True, norm=True):  
    glcm = greycomatrix(img, distances=dists, angles=agls, levels=lvl, symmetric=sym, normed=norm)  
    feature = []  
    glcm_props = [property for name in props for property in greycoprops(glcm, name)[0]]  
    for item in glcm_props:  
        feature.append(item)  
    feature.append(label)  
  
    return feature  
  
properties = ['correlation', 'homogeneity', 'contrast', 'energy']  
calc_glcm_all_agls(img, label, props=properties)
```

Gambar 5. Kode Tahapan Ekstraksi Fitur dengan GLCM

Tabel 1. Contoh Hasil Ekstraksi fitur sudut 0°

correlation_0	homogeneity_0	contrast_0	energy_0	label
973.771.099.613	6.253.468.288.021	23.065.051.470.591	5.810.545.413.696	1
9.725.048.637.706	5.048.634.779.628	1.656.605.882.353	4.321.822.166.234	1
9.647.664.472.336	5.207.696.740.206	4.113.839.950.980	4.774.696.706.391	1
9.637.722.697.987	5.256.519.999.442	3.366.884.558.823	4.789.941.160.407	1
9.788.590.105.854	5.417.382.290.576	3.856.483.333.334	4.850.718.826.043	2
9.797.343.101.682	5.357.271.974.141	36.022.159.313.729	43.418.496.872.864	2
9.603.929.785.765	4.757.571.711.068	5.404.516.176.471	42.532.967.176.171	2
9.669.742.827.517	46.902.275.106.077	3.844.170.098.039	4.142.118.872.375	2

3.3 Klasifikasi

Pada tahap klasifikasi, digunakan algoritma *Support Vector Machine* (SVM) dengan kernel linear. Proses klasifikasi dimulai dengan penskalaan fitur menggunakan *StandardScaler* untuk memastikan bahwa semua fitur memiliki skala yang seragam. Setelah itu, data dibagi menjadi data latih (*training data*) dan data uji (*test data*) dengan menggunakan *train_test_split* dari library *scikit-learn*. Model SVM kemudian dilatih menggunakan data latih dan dievaluasi menggunakan data uji. Dengan menggunakan kernel linear, model klasifikasi SVM diharapkan mampu memisahkan dengan baik antara citra-citra kupu-kupu dan ngengat. Proses klasifikasi ini bertujuan untuk menghasilkan model klasifikasi yang akurat dan dapat digunakan untuk mengklasifikasikan data baru dengan tepat.

```
#Penskalaan Fitur Menggunakan StandardScaler
scaler = StandardScaler()
scaler.fit(X)
X = scaler.transform(X)

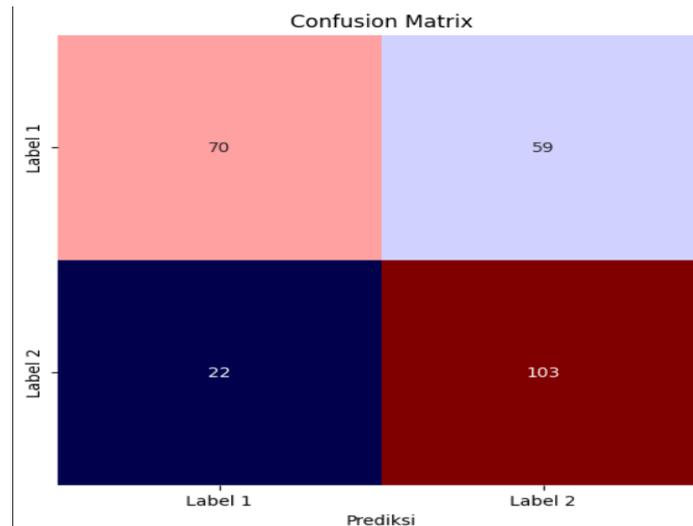
#Pembagian Data Latih dan uji ( test_size=0.2 = 20% untuk Testing )
x_train, x_test, y_train, y_test = train_test_split(X,Y, test_size=0.2, random_state=42)

#Modeling SVM Dengan Menggunakan Kernel Linear,
clf = SVC(kernel='linear')
clf.fit(x_train, y_train)
```

Gambar 6. Penskalaan Fitur dan Pelatihan Model SVM

3.4 Evaluasi

Evaluasi merupakan tahapan terakhir dalam penelitian ini yang bertujuan untuk menilai kinerja dari model klasifikasi yang telah dikembangkan. Pada tahap evaluasi ini, model *Support Vector Machine* (SVM) yang telah dilatih sebelumnya akan digunakan untuk melakukan prediksi terhadap data uji. Setelah melakukan prediksi, dilakukan perhitungan untuk mengetahui jumlah data yang berhasil diprediksi dengan benar dan jumlah data yang gagal diprediksi. Hasilnya bisa dilihat pada Gambar 5.



Gambar 7. Confusion Matrix

Berdasarkan *Confusion Matrix* yang terdapat pada Gambar 5, dapat dilihat bahwa dari total 129 data dengan label 1 yang merupakan kupu-kupu, model berhasil memprediksi dengan benar sebanyak 70 data, sementara 59 data diprediksi sebagai ngengat. Sedangkan dari total 125 data dengan label 2 yang merupakan ngengat, model berhasil memprediksi dengan benar sebanyak 103 data, namun terdapat 22 data yang salah diprediksi sebagai kupu-kupu. Evaluasi model dilakukan dengan menggunakan nilai *precision*, *recall*, dan *F1-score*, nilai akurasi dari model juga dievaluasi untuk mendapatkan gambaran keseluruhan tentang kinerja model klasifikasi. Detail evaluasi ini ditampilkan dalam Gambar 6.

```

Jumlah data berhasil diprediksi: 173
Jumlah data gagal diprediksi: 81
      precision    recall  f1-score   support

   1         0.76     0.54     0.63     129
   2         0.64     0.82     0.72     125

 accuracy          0.68     254
 macro avg         0.70     0.68     0.68     254
 weighted avg      0.70     0.68     0.67     254

Akurasi SVM : 68.11%
    
```

Gambar 8. Hasil Evaluasi

Berdasarkan Hasil Evaluasi yang dapat dilihat pada gambar 6, Model *Support Vector Machine* yang dilatih dengan fitur ekstraksi *Gray-Level Co-occurrence Matrix correlation* seperti *homogeneity*, *contrast*, dan *energy*, dengan menggunakan empat sudut yaitu 0°, 45°, 90°, dan 135° berhasil mendapatkan akurasi sebesar 68.11%, nilai *precision* 70.0%, nilai *recall* 68. % dan *f1-score* sebesar 68.0%

4. Kesimpulan

Berdasarkan penelitian yang dilakukan dari tahapan pengumpulan data, *preprocessing*, ekstraksi fitur, dan klasifikasi, dapat disimpulkan bahwa penggunaan metode Ekstraksi Fitur *Gray-Level Co-occurrence Matrix* dengan mengambil fitur *correlation*, *homogeneity*, *contrast*, dan *energy* dari sudut 0°, 45°, 90°, dan 135° menggunakan Model klasifikasi yang dikembangkan menggunakan

Support Vector Machine mampu menghasilkan akurasi sebesar 68,11%. Metode ini berhasil mengklasifikasikan jenis-jenis serangga, seperti kupu-kupu dan ngengat, dengan nilai *precision*, *recall*, dan *F1-Score* masing-masing sebesar 70,0%, 68,0%, dan 68,0%. Saran untuk penelitian selanjutnya dapat meliputi beberapa tahapan tambahan, seperti seleksi fitur, ekstraksi fitur warna, dan tahapan lainnya. Selain itu, juga disarankan untuk mempertimbangkan penggunaan model SVM multiclass guna meningkatkan akurasi model.

Daftar Pustaka

- [1] S. Talib, S. Sudin, and M. D. Suratin, "Penerapan Metode Support Vector Machine (Svm) Pada Klasifikasi Jenis Cengkeh Berdasarkan Fitur Tekstur Daun," *J. PROSISKO*, vol. 11, 2024.
- [2] A. A. Huda, B. Setiaji, and F. R. Hidayat, "Implementasi Gray Level Co-Occurrence Matrix (GlcM) Untuk Klasifikasi Penyakit Daun Padi," *J. Pseudocode*, vol. 9, 2022, [Online]. Available: www.ejournal.unib.ac.id/index.php/pseudocode
- [3] D. Satria, Y. Kartika, and H. Maulana, "Classification of color features in butterflies using the Support Vector Machine (SVM)," *Netw. Secur. Inf. Syst.*, vol. 2, no. 2, pp. 83–87, 2021.
- [4] A. K. Ulandari, G.K. Ramdhani, Wahyuningsih, M. N. Arwansyuri and F. Bimantoro, "Klasifikasi Jeruk Segar dan Busuk Melalui GLCM dan HSV dengan Menggunakan Metode ANN," *Pros. Semin. Nas. Teknol. DAN SAINS TAHUN 2024*, vol. 3, 2024.
- [5] R. A. Rizal, I. S. Girsang, and S. A. Prasetyo, "Klasifikasi Wajah Menggunakan Support Vector Machine (SVM)," *REMIK (Riset dan E-Jurnal Manaj. Inform. Komputer)*, vol. 3, no. 2, p. 1, 2019, doi: 10.33395/remik.v3i2.10080.
- [6] N. Neneng, K. Adi, and R. Isnanto, "Support Vector Machine Untuk Klasifikasi Citra Jenis Daging Berdasarkan Tekstur Menggunakan Ekstraksi Ciri Gray Level Co-Occurrence Matrices (GLCM)," *J. Sist. Inf. BISNIS*, vol. 6, no. 1, p. 1, Nov. 2016, doi: 10.21456/vol6iss1pp1-10.
- [8] Gerry, (2023, April). Butterfly & Moths Image Classification 100 species, Version 13. Retrieved May 1, 2024 from <https://www.kaggle.com/discussions/general/46091>
- [9] H. A. Sidharta, "Introduction to Open CV" Binus University. <https://binus.ac.id/malang/2017/10/introduction-to-open-cv/> (accessed May.2, 2024).
- [10] M. Das, "Rembg: Menghapus Latar Belakang dengan Mudah dengan Python" Medium. <https://medium.com/@HeCanThink/rembg-effortlessly-remove-backgrounds-in-python-c2248501f992> (accessed May.2, 2024).
- [11] M. Yunus, "Feature Extraction : Gray Level Co-occurrence Matrix (GLCM)" Medium. <https://yunusmuhammad007.medium.com/feature-extraction-gray-level-co-occurrence-matrix-glcM-10c45b6d46a1> (accessed May.2, 2024).
- [12] "Support Vector Machines" scikit-learn.org. <https://scikit-learn.org/stable/modules/svm.html> (accessed May. 1, 2024).

Halaman ini sengaja dibiarkan kosong