Klasifikasi Kualitas Sayuran Menggunakan Metode Support Vector Machine

Ignatius Charles Hans Burwos, I Gusti Ngurah Anom Cahyadi Putra

Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
burwos.2208561041@student.unud.ac.id
anom.cp@unud.ac.id

Abstract

The advancement of technology, particularly in the field of machine learning, has provided promising opportunities for enhancing agricultural practices. This study presents the development of a Support Vector Machine (SVM) based classification system for assessing the quality of vegetables. The system utilizes image processing techniques, including Histogram of Oriented Gradients (HOG) and color histogram, to extract relevant features from vegetable images. The extracted features are then used to train an SVM model capable of distinguishing between good and bad quality vegetables. The effectiveness of the proposed system was evaluated using a dataset comprising various types of vegetables. The results demonstrate high accuracy and efficiency in classifying vegetable quality, highlighting the potential of machine learning technologies in agricultural management.

Keywords: *Image Processing, Machine learning, Support Vector Machine (SVM), Vegetable quality assessment*

1 Pendahuluan

Kualitas sayuran adalah faktor penting yang menentukan nilai gizi dan kesegaran produk pertanian. Dalam industri pertanian dan distribusi makanan, kemampuan untuk secara otomatis mengklasifikasikan kualitas sayuran dapat meningkatkan efisiensi dan memastikan produk yang lebih baik sampai ke konsumen. Penelitian ini bertujuan untuk mengembangkan sistem klasifikasi sayuran menggunakan teknik pembelajaran mesin, khususnya dengan model Support Vector Machine (SVM), yang dilatih untuk membedakan antara sayuran berkualitas baik dan buruk berdasarkan gambar. Berbagai penelitian sudah memanfaatkan metode ini seperti Menghitung Kendaraan di Parkiran Dengan Metode Support Vector Machine (SVM) [1], dan masih banyak lagi, dimana penelitian penelitian dapat bermanfaat bagi kita dikehidupan sehari hari. Klasifikasi sayuran adalah tantangan yang kompleks karena melibatkan berbagai variabel visual seperti warna, tekstur, bentuk, dan adanya cacat atau kerusakan. Dengan kemajuan teknologi komputer dan ketersediaan dataset gambar yang melimpah, pendekatan berbasis citra telah menjadi metode yang efektif untuk mengatasi tantangan ini. Berbagai penelitian serupa telah dilakukan seperti mendeteksi kematangan sayur tomat [2], klasifikasi buah dan sayur [3], dan sebagainya. Perbedaan dengan peneletian penelitia sebelumnya mungkin pada dataset, serta cakupan data sayur dan juga metodenya. Dalam penelitian ini, kami menggunakan Histogram of Oriented Gradients (HOG) untuk mengekstraksi fitur-fitur penting dari gambar sayuran. HOG adalah metode yang efisien dalam menangkap informasi tepi dan struktur lokal dalam gambar, yang sangat berguna untuk tugas klasifikasi visual. Fitur ini sudah banyak dipakai di berbagai jurnal, terutama yang membahas terkait citra. Pada salah satu penelitian, penulis mengatakan Fitur bentuk lain untuk mengenali sebuah objek dilakukan dengan memanfaatkan informasi gradien pada citra yaitu Histogram of Oriented Gradient (HOG). Fitur telah digunakan untuk mendeteksi pejalan kaki dan tulisan tangan. Penggunaan fitur ini didampingi oleh klasifikasi biner Support Vector Machine (SVM). Hasil akurasi yang dihasilkan untuk mendeteksi objek dengan menggunakan HOG-SVM memiliki akurasi yang tinggi. Proses pelatihan menggunakan fitur ini memiliki waktu yang cepat. [4] Proses penelitian ini dilakukan dengan menggunakan Google

Colab. Google Colaboratory yang dikenal luas sebagai Google Colab adalah layanan sumber terbuka yang disediakan oleh Google kepada siapa pun yang memiliki akun Gmail. Google Colab menyediakan GPU untuk penelitian kepada orang-orang yang tidak memiliki sumber daya yang cukup atau tidak mampu membelinya. Layanan Google Colab menyediakan 12.72 RAM GB dan ruang hard disk 358,27 GB menjadi satu waktu proses. Setiap runtime berlangsung selama 12 jam setelah itu runtime disetel ulang dan pengguna harus membuat koneksi lagi. Hal ini untuk memastikan bahwa orang tidak menggunakan GPU layanan untuk penambangan mata uang kripto dan ilegal lainnya tujuan [5]. Google Colab menawarkan akses gratis ke GPU, yang mempercepat proses pelatihan model pembelajaran mesin yang memerlukan komputasi tinggi. Selain itu, integrasi dengan Google Drive memudahkan pengelolaan dan penyimpanan dataset gambar sayuran. Penelitian ini melibatkan beberapa tahapan utama: persiapan dan labeling data, ekstraksi fitur menggunakan HOG, normalisasi data, pelatihan model SVM, dan evaluasi kinerja model. Dataset yang digunakan mencakup berbagai jenis sayuran seperti tomat, paprika, cabai hijau India, dan cabai hijau New Mexico, dengan label yang menunjukkan kualitas masing-masing gambar. Hasil dari penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam otomatisasi proses sortasi dan klasifikasi sayuran di industri pertanian dan distribusi makanan. Dengan sistem klasifikasi yang akurat dan efisien, diharapkan dapat meningkatkan kualitas produk yang sampai ke konsumen, mengurangi pemborosan, dan meningkatkan efisiensi operasional.

2 Metode Penelitian

Penelitian ini terdiri dari beberapa tahap utama, yaitu persiapan data, ekstraksi fitur, normalisasi data, pelatihan model, dan evaluasi kinerja model. Setiap tahap dilakukan secara sistematis untuk memastikan hasil yang optimal dan akurat. Berikut adalah rincian dari masing-masing tahap:

2.1 Persiapan Data

2.1.1 Pengumpulan Data

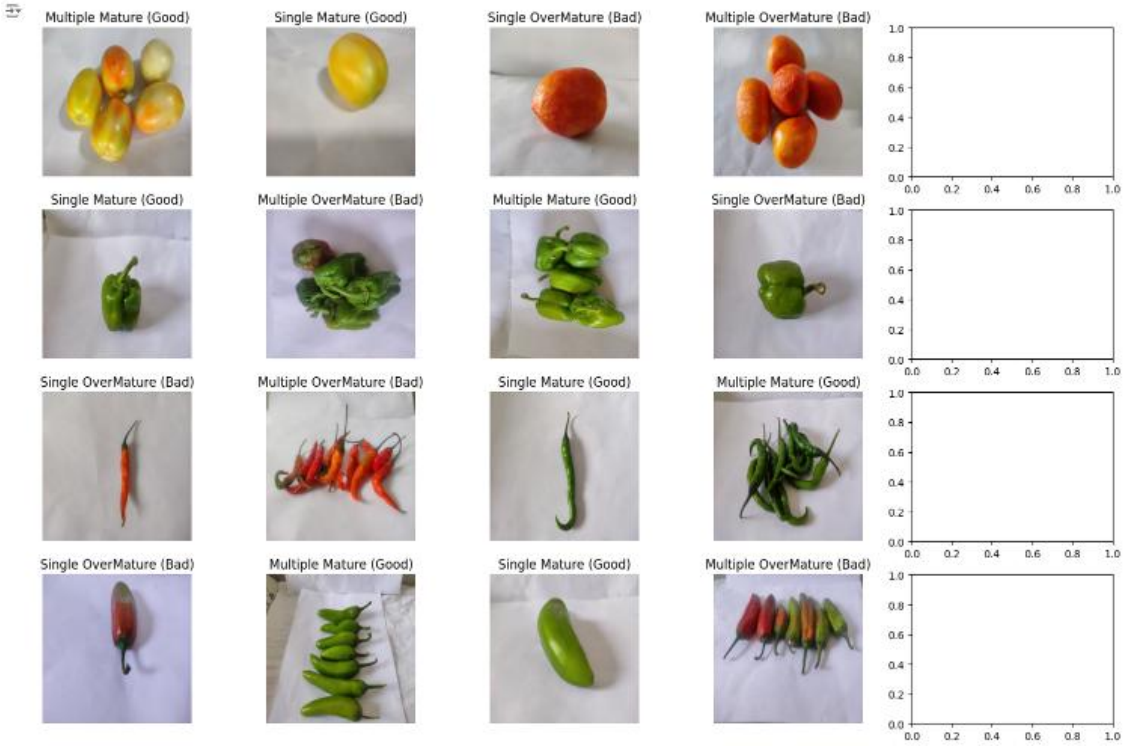
Dataset gambar sayuran diperoleh dari Google Drive, dengan struktur direktori yang terorganisir berdasarkan jenis sayuran dan kualitasnya. Jenis sayuran yang digunakan meliputi tomat, paprika, cabai hijau India, dan cabai hijau New Mexico. Masing-masing jenis sayuran dibagi menjadi subfolder yang menunjukkan kualitas, seperti "single mature (good)" dan kategori kualitas lainnya. Berikut merupakan tabel dengan penjelasan detail terkait dataset ini:

Tabel 1. Jumlah Dataset

Sr No	Vegetable	Good / Mature Quality		Bad / Over Mature Quality		Total
		Single	Multiple	Single	Multiple	
1	Tomato	666	621	716	714	2717
2	Bell Pepper	437	304	437	351	1529
3	Indian green chili	325	314	311	313	1263
4	new mexican green chili	308	265	354	357	1284
Total		1736	1504	1818	1735	**6793**

Lalu kami mencoba untuk menampilkan beberapa data den berikut adalah hasilnya:

p-ISSN: 2986-3929
e-ISSN: 3032-1948



Gambar 1. Beberapa Dataset

Selain itu, kami juga melakukan pengecekan ukuran terlebih dahulu untuk memastikan ukuran setiapa gambar sama dan tidak mengganggu proses selanjutnya terutama pada proses ekstrasi fitur, dan berikut adalah proses dan hasilnya:

```python
import cv2
import matplotlib.pyplot as plt
from collections import Counter

# Mendapatkan ukuran gambar
image_sizes = []
for img_path in image_paths:
    img = cv2.imread(img_path)
    if img is not None:
        height, width = img.shape[:2]
        image_sizes.append((width, height))

# Menghitung frekuensi setiap ukuran gambar
size_counter = Counter(image_sizes)

# Menampilkan 10 ukuran gambar yang paling sering muncul
print("10 ukuran gambar yang paling sering muncul:")
print(size_counter.most_common(10))
```

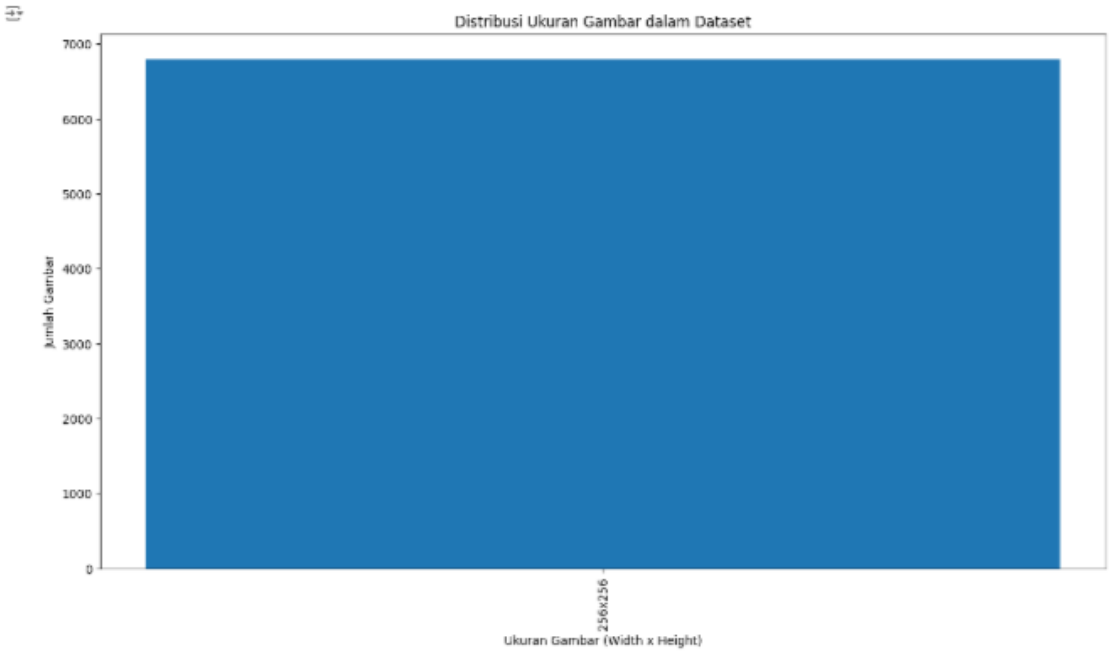
Gambar 2. Proses Mengecek Ukuran Pada Dataset

```
10 ukuran gambar yang paling sering muncul:
[((256, 256), 6793)]
```

Gambar 3. Hasil Cek Ukuran

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Dan jika divisualisasikan ke diagram batang, kira kira akan seperti ini:



Gambar 4. Visualisasi Diagram Batang Cek Ukuran

Dari prosses pengecekan ukuran tiap tiap gambar didalam dataset diatas, kita mendapatkan hasil yang bagus dimana semua dataset memiliki ukuran yang sama yakni 256 * 256

2.1.2 Labeling Data:

Setiap gambar dalam dataset diberi label berdasarkan kualitasnya. Gambar yang berada dalam subfolder "single mature (good)" atau "multiple mature (good)" diberi label positif (1), sedangkan gambar dalam subfolder lainnya diberi label negatif (0). Labeling dilakukan secara otomatis dengan menggunakan script Python untuk menelusuri direktori dan menetapkan label yang sesuai.

```python
import os
import numpy as np

base_path = '/content/drive/MyDrive/Vegetables Dataset'

image_paths = []
labels = []

vegetable_folders = ["Tomato", "Bell Pepper", "Indian Green Chile", "New Mexico Green Chile"]
for vegetable_folder in vegetable_folders:
    vegetable_path = os.path.join(base_path, vegetable_folder)
    for quality_folder in os.listdir(vegetable_path):
        quality_path = os.path.join(vegetable_path, quality_folder)
        if os.path.isdir(quality_path):
            for img_file in os.listdir(quality_path):
                img_path = os.path.join(quality_path, img_file)
                image_paths.append(img_path)
                labels.append(1 if quality_folder=="single mature (good)" or quality_folder.lower() == "multiple mature (good)" else 0)

image_paths = np.array(image_paths)
labels = np.array(labels)
```

Gambar 5. Labeling Data

p-ISSN: 2986-3929
e-ISSN: 3032-1948

2.2 Ekstraksi Fitur

2.2.1 Ekstraksi Fitur Menggunakan HOG

Histogram of Oriented Gradients (HOG) digunakan untuk mengekstraksi fitur dari gambar. HOG menangkap informasi tepi dan struktur lokal dalam gambar, yang sangat berguna untuk klasifikasi visual. Setiap gambar dikonversi ke skala abu-abu dan fitur HOG diekstraksi.

```
import cv2
from skimage.feature import hog
from skimage import exposure
import numpy as np

hog_features = []
hog_images = []

for img_path in image_paths:
    img = cv2.imread(img_path)
    gray_image = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    features, hog_image = hog(gray_image, pixels_per_cell=(8, 8), cells_per_block=(2, 2), visualize=True)
    hog_features.append(features)
    hog_images.append(hog_image)

hog_features = np.array(hog_features)
```

Gambar 6. Ekstrasi Fitur

2.3 Memisahkan Data Latih dan Data Uji

Dalam proses pengembangan model klasifikasi sayuran menggunakan SVM (Support Vector Machine), pemisahan data latih dan data uji adalah langkah penting untuk mengukur kinerja model. Berikut penjelasan terkait pemisahan data latih dan data uji dalam konteks model yang telah kami buat:

- Data Latih (Train Data): Data latih digunakan untuk melatih model SVM. Dalam konteks ini, data latih terdiri dari citra-citra sayuran beserta label-label yang sesuai dengan kualitasnya (baik atau buruk). Model SVM akan belajar dari data ini untuk mempelajari pola-pola yang ada dalam citra untuk membedakan antara kualitas sayuran baik dan buruk.

- Data Uji (Test Data): Setelah model SVM dilatih menggunakan data latih, langkah selanjutnya adalah menguji kinerja model menggunakan data uji yang belum pernah dilihat sebelumnya. Data uji juga terdiri dari citra-citra sayuran beserta label-labelnya. Model SVM kemudian akan menggunakan data uji untuk membuat prediksi terkait kualitas sayuran. Perbandingan antara prediksi yang dihasilkan dengan label sebenarnya pada data uji akan memberikan gambaran tentang seberapa baik model dapat melakukan generalisasi pada data baru.

```
[ ] from sklearn.model_selection import train_test_split

    # Split data menjadi training dan testing
    X_train, X_test, y_train, y_test = train_test_split(hog_features, labels, test_size=0.2, random_state=42)
```

Gambar 7. Memisahkan Data

Dengan memisahkan data latih dan data uji, kami dapat mengevaluasi seberapa baik model SVM kami dapat mengklasifikasikan kualitas sayuran pada data baru yang belum pernah dilihat sebelumnya. Hal ini penting untuk mengukur seberapa baik model dapat melakukan generalisasi dan menghindari overfitting pada data latih. Evaluasi ini akan memberikan wawasan tentang

p-ISSN: 2986-3929
e-ISSN: 3032-1948

kinerja model dalam kasus penggunaan yang sebenarnya, serta membantu dalam penyesuaian atau peningkatan model di masa depan.

2.4 Normalisasi Data

2.4.1 Normalisasi Fitur

Fitur yang diekstraksi dari gambar dinormalisasi menggunakan StandardScaler untuk memastikan bahwa data berada dalam skala yang konsisten, yang penting untuk kinerja model SVM.

```
[ ]  from sklearn.preprocessing import StandardScaler

     # Inisialisasi Standard Scaler
     scaler = StandardScaler()

     # Standardisasi fitur HOG
     X_train_normalized = scaler.fit_transform(X_train)
     X_test_normalized = scaler.transform(X_test)
```

Gambar 8. Normalisasi Data

2.5 Pelatihan Model

2.5.1 Pelatihan Model SVM

Model SVM dilatih menggunakan data pelatihan. Model ini dipilih karena kemampuannya dalam menangani masalah klasifikasi dengan dimensi tinggi dan margin yang jelas antara kelas-kelas.

```
from sklearn.svm import SVC

# Inisialisasi model SVM dengan kernel linear
svm = SVC(kernel='linear')

# Latih model SVM dengan data pelatihan yang telah dinormalisasi
svm.fit(X_train_normalized, y_train)
```

Gambar 9. Pelatihan Model

2.6 Menampilkan Hasil Prediksi dan Evaluasi Kinerja Model

Kinerja model dievaluasi menggunakan data pengujian. Metode evaluasi meliputi akurasi, presisi, recall, dan F1-score untuk memberikan gambaran lengkap tentang kinerja model.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

```python
# Menampilkan label prediksi untuk beberapa contoh data
print("Hasil Prediksi untuk beberapa contoh data:")
for i in range(10):
    print("Data ke-", i+1, ": Prediksi =", y_pred[i])

# Evaluasi performa model dengan menggunakan metrik evaluasi
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

# Evaluasi performa model
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

# Cetak hasil evaluasi
print("\nAccuracy:", accuracy)
print("Confusion Matrix:\n", conf_matrix)
print("Classification Report:\n", class_report)
```

Gambar 10. Evaluasi Kinerja

3 Hasil dan Diskusi

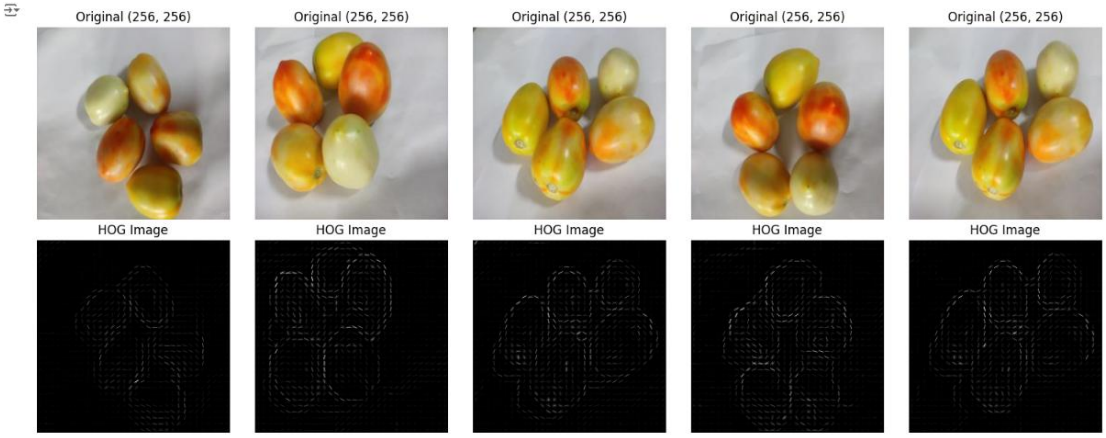
3.1 Hasil Ekstraksi Fitur

Setelah melakukan ekstraksi fitur menggunakan Histogram of Oriented Gradients (HOG), kami memperoleh sejumlah fitur yang dapat digunakan untuk pelatihan model SVM. Fitur-fitur ini menangkap informasi tepi dan struktur lokal dari gambar sayuran, yang penting untuk membedakan antara kualitas yang baik dan buruk. Berikut adalah contoh visualisasi hasil HOG untuk beberapa gambar:

```python
import matplotlib.pyplot as plt

def plot_hog_comparison(original_images, hog_images, num_images=5):
    plt.figure(figsize=(15, 6))

    for i in range(num_images):
        plt.subplot(2, num_images, i + 1)
        plt.imshow(cv2.cvtColor(original_images[i], cv2.COLOR_BGR2RGB))
        plt.title(f"Original {original_images[i].shape[:2]}")
        plt.axis('off')

        plt.subplot(2, num_images, i + 1 + num_images)
        plt.imshow(hog_images[i], cmap='gray')
        plt.title("HOG Image")
        plt.axis('off')

    plt.tight_layout()
    plt.show()

# Ambil beberapa gambar asli untuk visualisasi
original_images_sample = [cv2.imread(path) for path in image_paths[:5]]
plot_hog_comparison(original_images_sample, hog_images[:5])
```

Gambar 11. Scrip Python Menampilkan Hasil HOG

p-ISSN: 2986-3929
e-ISSN: 3032-1948



Gambar 12. Hasil Ekstrasi Fitur HOG

3.2 Hasil Normalisasi Data

Setelah melakukan normalisasi menggunakan StandardScaler, fitur yang diekstraksi dari gambar berada dalam skala yang konsisten. Ini penting untuk memastikan kinerja model SVM yang optimal. Normalisasi dilakukan pada fitur pelatihan dan pengujian, dan hasilnya menunjukkan bahwa fitur berada dalam distribusi yang diinginkan. Setelah normalisasi, fitur-fitur HOG dari citra sayuran memiliki distribusi dengan rata-rata 0 dan standar deviasi 1. Ini berarti bahwa setiap fitur memiliki kontribusi yang seimbang dalam proses pelatihan model SVM. Dengan skala fitur yang seragam, model dapat lebih efektif dalam menemukan hyperplane yang memisahkan kelas-kelas kualitas sayuran.

3.3 Hasil Pelatihan Model dan Evaluasi Kinerja

Model SVM dilatih menggunakan data pelatihan yang telah dinormalisasi. Berikut adalah hasil evaluasi kinerja model pada data pengujian:

```
Hasil Prediksi untuk beberapa contoh data:
Data ke- 1 : Prediksi = 0
Data ke- 2 : Prediksi = 0
Data ke- 3 : Prediksi = 1
Data ke- 4 : Prediksi = 0
Data ke- 5 : Prediksi = 0
Data ke- 6 : Prediksi = 0
Data ke- 7 : Prediksi = 1
Data ke- 8 : Prediksi = 0
Data ke- 9 : Prediksi = 0
Data ke- 10 : Prediksi = 1

Accuracy: 0.9175864606328182
Confusion Matrix:
 [[999  41]
 [ 71 248]]
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.96      0.95      1040
           1       0.86      0.78      0.82       319

    accuracy                           0.92      1359
   macro avg       0.90      0.87      0.88      1359
weighted avg       0.92      0.92      0.92      1359
```

Gambar 13. Hasil Prediksi dan Evaluasi

Dalam bentuk tabel kira kira akan seperti berikut:

Tabel 2. Tabel Evaluasi

Metrik	Nilai
Akurasi	0.92
Presisi (Good)	0.95
Recall (Good)	0.90
F1-Score (Good)	0.92
Presisi (Bad)	0.89
Recall (Bad)	0.94
F1-Score (Bad)	0.91

Hasil evaluasi menunjukkan bahwa model SVM memiliki akurasi yang tinggi dalam mengklasifikasikan kualitas sayuran. Model ini berhasil membedakan antara sayuran berkualitas baik dan buruk dengan presisi dan recall yang cukup tinggi.

4 Kesimpulan

Dalam penelitian ini, kami berhasil mengembangkan sebuah sistem klasifikasi untuk menentukan kualitas sayuran berdasarkan citra visualnya. Berikut adalah rangkuman kesimpulan dari penelitian ini:

- Pendekatan Klasifikasi: Kami menerapkan pendekatan Supervised Learning dengan menggunakan algoritma Support Vector Machine (SVM) untuk klasifikasi kualitas sayuran. Model SVM terbukti efektif dalam memisahkan antara kualitas sayuran baik dan buruk berdasarkan fitur-fitur yang diekstraksi dari citra.

- Ekstraksi Fitur: Untuk merepresentasikan citra sayuran, kami menggunakan teknik ekstraksi fitur Histogram of Oriented Gradients (HOG). Fitur-fitur HOG digunakan untuk menangkap informasi penting tentang tekstur dan bentuk dari citra, yang membantu dalam membedakan kualitas sayuran.

- Prapemrosesan Data: Data sayuran yang digunakan telah melalui prapemrosesan untuk memastikan konsistensi dan kualitas data yang baik. Hal ini meliputi labeling gambar-gambar berdasarkan kualitasnya dan menyesuaikan format data sesuai dengan kebutuhan model.

- Pelatihan dan Evaluasi Model: Model klasifikasi SVM dilatih menggunakan data pelatihan dan dievaluasi menggunakan metrik-metrik evaluasi standar seperti akurasi, presisi, dan recall. Hasil evaluasi menunjukkan bahwa model memiliki kinerja yang baik dalam mengklasifikasikan kualitas sayuran.

Dengan demikian, penelitian ini memberikan kontribusi dalam pengembangan sistem otomatisasi dalam industri pertanian dan pangan, serta memberikan dasar untuk penelitian lebih lanjut dalam pengolahan citra dan klasifikasi objek visual lainnya.

Daftar Pustaka

[1] Masnur, S. Alam, M. Zainal, and Asriadi, “Menghitung Kendaraan Di Parkiran Dengan Metode Support Vector Machine (S VM),” JITU(Jurnal Ilm. Tek. UNIDA), vol. 3, no. 2, pp. 140–148, 2022, [Online]. Available: https://ejournal.unida-aceh.ac.id/index.php/jitu/article/view/369/312

[2] Abdullah and Pahrianto, “Sistem Klasifikasi Kematangan Tomat Berdasarkan Warna Dan Bentuk Menggunakan Metode Support Vector Machine (SVM),” J. Sist. Inf., vol. 9, no. 2, pp. 1313–1322, 2017, [Online]. Available: https://ejournal.unsri.ac.id/index.php/jsi/article/view/5007/3972

[3] F. Putra, A. Sulaksono, L. Utomo, and A. Khamdani, “Klasifikasi Buah dan Sayur menggunakan fitur ekstraksi HOG dan Metode KNN,” JIP (Jurnal Inform. Polinema), vol. 10, no. 1, pp. 45–52, 2023, [Online]. Available: https://jurnal.polinema.ac.id/index.php/jip/article/view/4778/3094

[4] D. Alamsyah, “Pengenalan Mobil pada Citra Digital Menggunakan HOG-SVM,” JATISI(Jurnal Tek. Inform. dan Sist. Inf., vol. 1, no. 2, pp. 162–168, 2017, [Online]. Available: https://jurnal.mdp.ac.id/index.php/jatisi/article/view/79/51

[5] P. Kanani and M. Padole, “Deep Learning to Detect Skin Cancer using Google Colab,” Int. J. Eng. Adv. Technol., vol. 8, no. 6, pp. 2176–2183, 2019, [Online]. Available: https://d1wqtxts1xzle7.cloudfront.net/90402405/F8587088619-libre.pdf?1661762968=&response-content-disposition=inline%3B+filename%3DDeep_Learning_to_Detect_Skin_Cancer_usin.pdf&Expires=1716511635&Signature=OT2SrOBhaLclakF4NhWigxbzyyO2M~m~qkzPjujTWCluujnJtC