

Implementasi CNN dan SVM untuk Identifikasi Penyakit Tomat via Daun

Felix¹, Said Faisal², Theresia F M Butarbutar³, Pahala Sirait⁴

Program Studi Teknik Informatika, STMIK Mikroskil

¹felix.pandi@mikroskil.ac.id, ²151112696@students.mikroskil.ac.id,

³151111488@students.mikroskil.ac.id, ⁴pahala@students.mikroskil.ac.id

Abstrak

Tanaman tomat sangat rentan terhadap serangan penyakit, penyakit pada tanaman tomat dapat diketahui dari bercak yang terdapat pada daun. Penyakit dapat dikenali secara visual karena memiliki ciri warna dan tekstur yang unik. Tetapi pengenalan secara visual memiliki kekurangan yaitu sulit dalam mengenali kemiripan antara satu jenis penyakit dengan penyakit lain sehingga berdampak pada kurang akuratnya penyakit yang diidentifikasi. Pada penelitian ini, dibangun suatu sistem yang dapat menentukan penyakit serta memberikan informasi berupa solusi penanganan dalam mencegah atau menangani penyakit yang menyerang daun tomat melalui identifikasi citra digital menggunakan supervised classification. Citra yang akan diidentifikasi sebelumnya melewati proses transformasi warna RGB (Red Green Blue) ke HSV (Hue Saturation Value), HSV (Hue Saturation Value) ke Grayscale, dan proses ekstraksi fitur tekstur GLCM (Gray Level Co-occurrence Matrix). Hasil ekstraksi fitur tekstur diklasifikasi dengan SVM (Support Vector Machine) dan CNN (Convolutional Neural Network) untuk menentukan penyakit yang diderita oleh daun tomat. Pengujian dilakukan dengan 200 sampel citra daun tomat, 160 citra sebagai data latih dan 40 citra sebagai data uji. Hasil pengujian menunjukkan metode CNN memiliki persentase rata-rata accuracy 97.5%, precision 95.45%, recall 95% dan error 5%. Sedangkan SVM menghasilkan rata-rata accuracy 95%, precision 90.83%, recall 90% dan error 10%. Dari hasil pengujian dapat dinyatakan bahwa pada penelitian ini CNN adalah classifier yang lebih baik dibandingkan SVM.

Kata Kunci : Penyakit tomat, GLCM, CNN, SVM

Abstract

Tomato plants are very susceptible to diseases, tomato plant diseases can be known from the spots on the leaves. This disease can be recognized because it has a unique color and texture. But visual recognition has a disadvantage that is difficult to recognize the similarity between one type of disease with another disease so that the impact on the inaccuracy of the disease identified. In this study, a system was built to determine disease and provide information in the form of treatment solutions to prevent or treat diseases that attack tomato leaves through digital image identification using supervised classification. The image that will be identified earlier goes through the process of RGB (Red Green Blue) color transformation to HSV (Hue Saturation Value), HSV (Hue Saturation Value) to Grayscale, and the GLCM (Gray Level Co-occurrence Matrix) texture feature extraction process. Texture feature extraction results are classified with SVM (Support Vector Machine) and CNN (Convolutional Neural Network) to determine the disease suffered by tomato leaves. The test was carried out with 200 sample images of tomato leaves, 160 images as training data and 40 images as test data. The test results show the CNN method has an average percentage accuracy of 97.5%, precision 95.45%, recall

95% and error 5%. Whereas SVM produces an average accuracy of 95%, precision 90.83%, recall 90% and error 10%. From the test results it can be stated that in this study CNN is a better classifier than SVM.

Keyword : *Tomato Disease, GLCM, CNN, SVM*

1. PENDAHULUAN

Tanaman tomat merupakan sayuran yang terkenal di Indonesia dan merupakan komoditas sayuran yang harus mengalami peningkatan dari tahun ke tahun [1]. Sampai saat ini tanaman tomat dibudidayakan oleh petani, baik dataran rendah maupun dataran tinggi dan tumbuh hampir di seluruh wilayah di Indonesia [2].

Tanaman tomat sangat rentan terhadap serangan penyakit, penyakit merupakan salah satu faktor utama menurunnya kualitas dan kuantitas produksi pertanian secara global [3]. Penyakit pada tanaman tomat dapat diketahui dari bercak yang terdapat pada daun [4]. Penyakit ini dapat dikenali secara visual karena memiliki ciri warna dan tekstur yang unik. Tetapi pengenalan secara visual memiliki kekurangan yaitu sulit dalam mengenali kemiripan antara satu jenis penyakit dengan penyakit lain sehingga berdampak pada subjektivitas dan kurang akuratnya penyakit yang diidentifikasi [5].

Beberapa penelitian terkait pendeteksian penyakit pada tanaman sebelumnya telah dilakukan dengan judul “Identifikasi Penyakit Pada Tanaman Tomat Berdasarkan Warna Dan Bentuk Daun Dengan Metode *Naive Bayes Classifier* Berbasis Web”. Penelitian ini membahas tentang identifikasi pada daun tomat dengan proses ekstraksi fitur warna serta bentuk menggunakan HSV (*Hue, Saturation, Value*) dan operator *Sobel* yang memberikan persentase keberhasilan sebesar 82,98% [6]. Penelitian lainnya berjudul “SVM and ANN Based Classification of Plant Diseases Using Feature Reduction Technique”. Pada penelitian ini digunakan ekstraksi fitur warna berdasarkan RGB dan HSI serta tekstur oleh GLCM pada gambar, kemudian fitur dilatih dan diklasifikasi dengan SVM dan ANN untuk menentukan penyakit tanaman. Hasil penelitian menunjukkan bahwa klasifikasi dengan SVM mencapai accuracy 92.17% lebih tepat untuk identifikasi dan klasifikasi penyakit tanaman dibandingkan ANN yang mencapai accuracy 87.48% [7]. Penelitian lainnya berjudul “Detection of Affected Part of Plant Leaves and Classification of Diseases Using CNN Technique” yang membahas tentang pendeteksian penyakit daun dengan menggunakan CNN (*Convolutional Neural Network*). Pada mulanya penelitian tersebut menggunakan *Bilateral Filter* untuk menghilangkan noise yang terdapat pada citra. Kemudian, dilakukan segmentasi menggunakan metode FCM (*Fuzzy C-Means*). Dilanjutkan dengan ekstraksi fitur tekstur oleh GLCM (*Gray Level Co-Occurance Matrix*) dan RLM (*Run Length Matrix*). Dan pada proses akhir dilakukan klasifikasi menggunakan CNN (*Convolutional Neural Network*). Hasil yang didapat berupa nilai accuracy rate 98%, precision rate 96%, error rate 2% dan recall rate 97% [8].

Berdasarkan uraian diatas, pada penelitian ini akan diimplementasikan beberapa metode- metode penyelesaian. Pertama, citra yang awalnya berwarna RGB (*Red, Green, Blue*) ditransformasikan ke HSV (*Hue, Saturation, Value*). Kemudian, dilakukan ekstraksi fitur tekstur menggunakan GLCM (*Gray Level Co-Occurrence Matrix*). Citra hasil akan diklasifikasikan dengan CNN (*Convolutional Neural Network*) dan SVM (*Support Vector Machine*), kemudian dibandingkan untuk menentukan *classifier* yang lebih baik dalam mendeteksi penyakit pada daun tomat secara otomatis. Citra daun tomat yang telah terdeteksi penyakit akan diberikan informasi berupa solusi penanganan dalam mengatasi atau mencegah penyakit yang menyerang daun tomat tersebut. Oleh karena itu, penelitian tugas akhir ini

berjudul, “Deteksi dan Pengenalan Penyakit Tumbuhan Pada Citra Daun Menggunakan Metode *Convolutional Neural Network* dan *Support Vector Machine*”.

2. KAJIAN PUSTAKA DAN METODE PENELITIAN

2.1 Kajian Pustaka

2.1.1 Citra *Grayscale*

Citra *grayscale* yaitu citra yang nilai pikselnya merepresentasikan derajat keabuan atas intensitas warna putih. Setiap nilai piksel di dalam citra *grayscale* sesuai dengan kecerahannya. Nilai piksel citra *grayscale* akan direpresentasikan oleh *byte* atau *word* dengan nilai 8-bit, intensitas kecerahan yang bervariasi dimulai dari 0 sampai 255, “0” direpresentasikan sebagai hitam dan “255” direpresentasikan sebagai putih [9].

$$\text{Grayscale} = ((R * 0.2989) + (G * 0.5870) + (B * 0.1140)) \quad (1)$$

Keterangan:

R = *Red*

G = *Green*

B = *Blue*

2.1.2 HSV (*Hue Saturation Value*)

Menurut beberapa penelitian, HSV (*Hue Saturation Value*) adalah model warna yang lebih baik digunakan untuk keperluan pengolahan citra dan *computer vision*. Konversi dari RGB (*Red Green Blue*) ke HSV (*Hue Saturation Value*) dapat dilakukan dengan persamaan-persamaan berikut [10].

$$r = \frac{R}{255} \quad (2)$$

$$g = \frac{G}{255} \quad (3)$$

$$b = \frac{B}{255} \quad (4)$$

$$V = \max(r, g, b) \quad (5)$$

$$Vm = V - \min(r, g, b) \quad (6)$$

$$S = \begin{cases} 0 & \text{jika } V = 0 \\ \frac{Vm}{V} & \text{jika } V > 0 \end{cases} \quad (7)$$

$$H = \begin{cases} 0^\circ & \text{jika } S = 0 \\ 60^\circ \times \left[\frac{(g-b)}{Vm} \right] \text{ mod } 6 & \text{jika } V = r \\ 60^\circ \times \left[2 + \frac{(b-r)}{Vm} \right] & \text{jika } V = g \\ 60^\circ \times \left[4 + \frac{(r-g)}{Vm} \right] & \text{jika } V = b \end{cases} \quad (8)$$

Keterangan :

R = *Red*

G = *Green*

B = *Blue*

r = Nilai R yang dinormalisasi

g = Nilai G yang dinormalisasi

b = Nilai B yang dinormalisasi

H = *Hue*

S = *Saturation*

V = *Value*

Vm = Hasil pengurangan nilai tinggi dengan nilai rendah pada *value*

2.1.3 GLCM (*Gray Level Co-occurrence Matrix*)

Pertama kali diusulkan oleh Haralick pada tahun 1973 dengan 28 fitur untuk menjelaskan pola spasial. GLCM (*Gray Level Co-occurrence Matrix*) menggunakan perhitungan tekstur pada orde kedua. Pengukuran tekstur pada orde pertama menggunakan perhitungan statistik didasarkan pada nilai piksel citra asli semata, seperti varians dan tidak memperhatikan hubungan ketetanggaan piksel. Pada orde kedua, hubungan antar pasangan dua piksel citra asli diperhitungkan [11]. Berikut rumus parameter tekstur yang digunakan pada penelitian ini [12]:

$$Energi = \sum_{ij} p(i,j)^2 \quad (9)$$

$$Entropi = \sum_{i,j} p(i,j) \log_2(p(i,j)) \quad (10)$$

Keterangan:

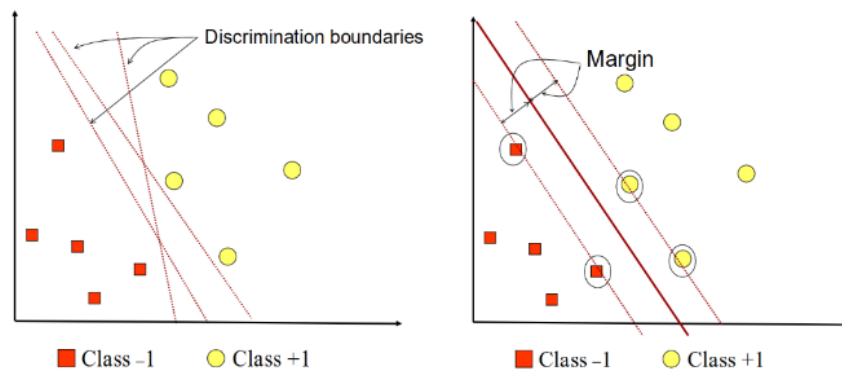
i = nilai baris matriks

j = nilai kolom matriks

$p(i,j)$ = nilai elemen *Co-occurrence Matrix* baris (i) dan kolom (j)

2.1.4 SVM (*Support Vector Machine*)

SVM (*Support Vector Machine*) adalah sistem pembelajaran yang menggunakan ruang hipotesis berupa fungsi-fungsi *linier* dalam sebuah ruang fitur (*feature space*) berdimensi tinggi, dilatih dengan metode pembelajaran yang didasarkan pada teori optimasi dengan mengimplementasikan *learning bias* yang berasal dari teori pembelajaran statistik [13].



Gambar 1 SVM Berusaha Menemukan *Hyperplane* Terbaik

Prinsip dasar SVM (*Support Vector Machine*) adalah *linear classifier*, dan selanjutnya dikembangkan agar dapat bekerja pada *non-linear problem*. *Non-linear problem* adalah masalah tidak tersebarnya data pada 2 dimensi. Sehingga data dapat ditransformasikan ke dimensi yang lain. Sebagai contoh, dapat dilakukan *polynomial mapping* dengan menggunakan fungsi $\phi: \mathbb{R}^2 \rightarrow \mathbb{R}^3$ yang didefinisikan sebagai berikut :

$$\phi: (x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \quad (11)$$

Tentunya tidak semua data harus diubah ke 3 dimensi, bisa saja 4, 5 atau lebih, tergantung data dan kebutuhan dalam mengklasifikasikan data tersebut [14].

Cara lain untuk menghitung nilai tanpa harus mentransformasikan vektor, dapat digunakan sebuah *kernel*. *Kernel* adalah sebuah fungsi yang dapat mengembalikan hasil dari *dot product* di tempat yang berbeda. *Kernel* sendiri memiliki beberapa tipe. Tipe- tipe *kernel* yang sering digunakan pada penelitian adalah sebagai berikut [14]:

a. *Linear Kernel*

Linear kernel adalah *kernel* yang sangat sederhana. *Linear kernel* bekerja dengan baik untuk mengklasifikasikan teks.

b. *Polynomial Kernel*

Polynomial kernel memiliki 2 parameter tambahan yaitu C , yang merepresentasikan nilai *constant*, dan d yang merepresentasikan derajat dari *kernel*. dengan rumus sebagai berikut:

c. RBF(*Radial Basis Function*) atau *Gaussian Kernel*

Gaussian / RBF (Radial Basis Function) adalah nilai yang tergantung hanya pada *distance* dari sumber atau dari beberapa titik.

$$K(x, x') = \exp(-\gamma \|x - x'\|^2) \quad (12)$$

SVM (*Support Vector Machine*) dapat menghasilkan *binary classifier* yang berguna untuk mengklasifikasikan dua *class*. Tetapi, *dataset* dalam kasus dunia nyata sangat jarang yang hanya menggunakan 2 *class*. Ada beberapa cara untuk menjadikan SVM (*Support Vector Machine*) dapat mengklasifikasikan lebih daripada 2 *class* atau yang umum disebut *multi-class* yaitu sebagai berikut [14]:

a. *One-Against-All*

Pada pendekatan ini, dalam mengklasifikasikan *class* dibentuk *binary classifier* yang berbeda pada tiap *class*.

b. *One-Against-One*

Pada pendekatan ini, dilakukan pencarian perbedaan antara satu *class* dengan satu *class* yang lainnya.

c. DAGSVM (*Directed Acyclic Graph SVM*)

Pelatihan data digunakan pendekatan yang sama dengan *One-Against-One*, untuk mempercepat pengujian digunakan *Directed Acyclic Graph (DAG)* untuk memilih *classifier* yang digunakan.

Metode *Decomposition* adalah metode yang dirancang untuk mengatasi masalah *quadratic programming* yang terjadi pada proses SVM. Metode ini mengatasi masalah *dual problem* dengan memecahkan urutan *subproblem quadratic programming* yang lebih kecil. Pada tahun 1999 Platt mengusulkan untuk selalu menggunakan ukuran *set* kerja sekecil mungkin, yaitu dua elemen. Pilihan ini secara dramatis menyederhanakan metode dekomposisi. Sehingga didapati metode SMO (*Sequential Minimal Optimization*) dengan *Maximum Violating Pair* yang ditunjukkan sebagai berikut [15]:

$$\text{Inisialisasi coefficient} \quad : \forall k \in \{1 \dots n\} \alpha_k \leftarrow 0 \quad (13)$$

$$\text{Inisialisasi gradient} \quad : \forall k \in \{1 \dots n\} g_k \leftarrow 1$$

: **loop**

$$\text{Maximal violating pair} \quad : i \leftarrow \max_i y_i g_i \text{ subject to } y_i \alpha_i < B_i$$

$$: j \leftarrow \min_j y_j g_j \text{ subject to } A_j < y_j \alpha_j$$

$$\text{Criterion optimal} \quad : \text{if } (y_i g_i - y_j g_j) < \text{tolerance OR iteration} \geq \text{max iteration stop}$$

$$\text{Mencari direction} \quad : \lambda \leftarrow \min\left(0, \min\left(\min(B_i - y_i \alpha_i, y_j \alpha_j + A_j), \frac{y_i g_i - y_j g_j}{K_{ii} + K_{jj} - 2K_{ij}}\right)\right)$$

$$\text{Perbaharui gradient} \quad : \forall k \in \{1 \dots n\} g_k \leftarrow g_k - \lambda y_k K_{ik} + \lambda y_k K_{jk}$$

$$\text{Perbaharui coefficient} \quad : \alpha_i \leftarrow \alpha_i + y_i \lambda \quad \alpha_j \leftarrow \alpha_j + y_j \lambda$$

: **end loop**

Metode SMO (*Sequential Minimal Optimization*) dengan *Maximum Violating Pair* tersebut akan digunakan untuk melakukan pelatihan pada *dataset*. Untuk pengujian atau klasifikasi pada data baru, dilakukan dengan rumus berikut [15]:

$$\text{sign}(\sum_i \alpha_i^* y_i K(x_i, x) + b) \quad (14)$$

i = indeks *support vector*

α^* = *lagrange multiplier*

y = kelas yang tersedia

K = *kernel*

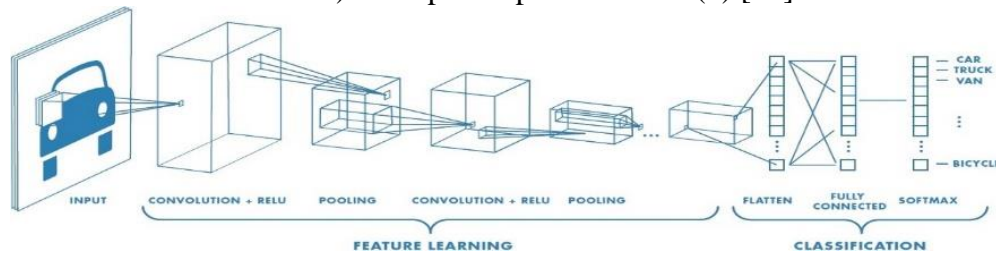
x = data fitur

b = *bias*

2.1.5 Convolutional Neural Network (CNN)

Penemuan CNN (*Convolutional Neural Network*) pertama kali dilakukan oleh Hubel dan Wiesel mengenai *virtual cortex* pada indera penglihatan kucing. Secara teknis, CNN (*Convolutional Neural Network*) adalah sebuah arsitektur yang dapat dilatih dan terdiri dari beberapa tahap, masukan (*input*) dan keluaran (*output*) [16].

Secara teknis, CNN (*Convolutional Neural Network*) melatih dan menguji setiap gambar masukan melalui serangkaian proses, yaitu *convolutional layer* diikuti oleh *pooling* untuk mengekstraksi fitur dari gambar masukan berturut-turut. Setelah operasi *pooling*, citra di-*flatten* dan kemudian dimasukkan ke proses *fully connected-layer* untuk melaksanakan tugas pengklasifikasian. Arsitektur yang menggambarkan proses- proses yang terjadi pada metode CNN (*Convolutional Neural Network*) ditampilkan pada Gambar (2) [17].



Gambar 2 Arsitektur dari CNN

a. *Convulusi Layer*

CNN (*Convolutional Neural Network*) menggunakan *filter* yang juga disebut sebagai *kernel* untuk mendeteksi fitur- fitur, seperti sudut yang terdapat di seluruh gambar. *Filter* hanyalah sebuah matriks nilai yang disebut bobot, dilatih untuk mendeteksi fitur tertentu. *Filter* bergerak di setiap bagian gambar untuk memeriksa apakah fitur yang dimaksudkan untuk dideteksi tersedia. Untuk memberikan nilai yang menunjukkan seberapa tepat fitur yang tersedia tersebut, *filter* melakukan operasi konvolusi [11].

b. *Downsampling*

Downsampling berguna untuk mempercepat proses pelatihan dan mengurangi jumlah memori yang dikonsumsi oleh *network*. *Downsampling* yang umum digunakan adalah *max pooling* [11].

c. *Fully-Connected Layer*

Fully-Connected Layer dari *neural network* adalah proses dimana matriks masukan di-*flatten* menjadi fitur vektor dan melewati jaringan *neuron* untuk memprediksi probabilitas

keluaran [11]. Proses ini menggunakan ANN (*Artificial Neural Network*) yang terdiri atas 2 tahapan yaitu *feedforward* [18]:

$$z = \sum_{i,j} x_{ij}w_{ij} + b_i \quad (15)$$

$$\sigma(z) \begin{cases} \text{sigmoid activation} = \frac{1}{(1+e^{-z})} \\ \text{softmax activation} = \frac{e^{z_{ij}-\max(z)_j}}{\sum e^{z_{ij}-\max(z)_j}} \end{cases} \quad (16)$$

Keterangan:

- i = nilai baris matriks
- j = nilai kolom matriks
- z = jumlah bobot masukan
- w = weight
- b = bias
- σ = fungsi aktivasi

Untuk tahapan yang kedua yaitu *backpropagation*, persamaan- persamaan yang digunakan adalah sebagai berikut [19]:

$$z\delta_{ij} = \frac{1}{n}(a_{ij} - y_{ij}) w_{ij}^T \quad (17)$$

$$a\delta_{ij} = z\delta_{ij} \times \sigma'(z) \quad (18)$$

$$\sigma'(z) = (1 - \sigma(z)) \quad (19)$$

$$w_{ij} = w_{ij} - \eta x_{ij}^T a\delta_{ij} \quad (20)$$

$$b_{ij} = b_{ij} - \eta \sum a\delta_{ij} \quad (21)$$

Keterangan:

- $z\delta$ = pencarian *gradient*
- $a\delta$ = hasil perkalian *gradient* dengan aktivasi *derivative*
- η = *learning rate*
- w = *weight*
- b = *bias*
- σ' = fungsi aktivasi *derivative*

2.1.6 Evaluasi

Confusion matrix adalah matriks untuk mengukur kinerja suatu klasifikasi yang memiliki 4 variabel yaitu TP (*True Positive*) berarti ketika sistem memprediksi positif dan hasilnya benar, TN (*True Negative*) berarti ketika sistem memprediksi negatif dan hasilnya benar, FP (*False Positive*) berarti ketika sistem memprediksi positif dan hasilnya salah, FN (*False Negative*) berarti ketika sistem memprediksi negatif dan hasilnya salah [20]. Ada beberapa perhitungan yang dapat digunakan sebagai pengujian pada sistem diantaranya [8]:

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)} \quad (22)$$

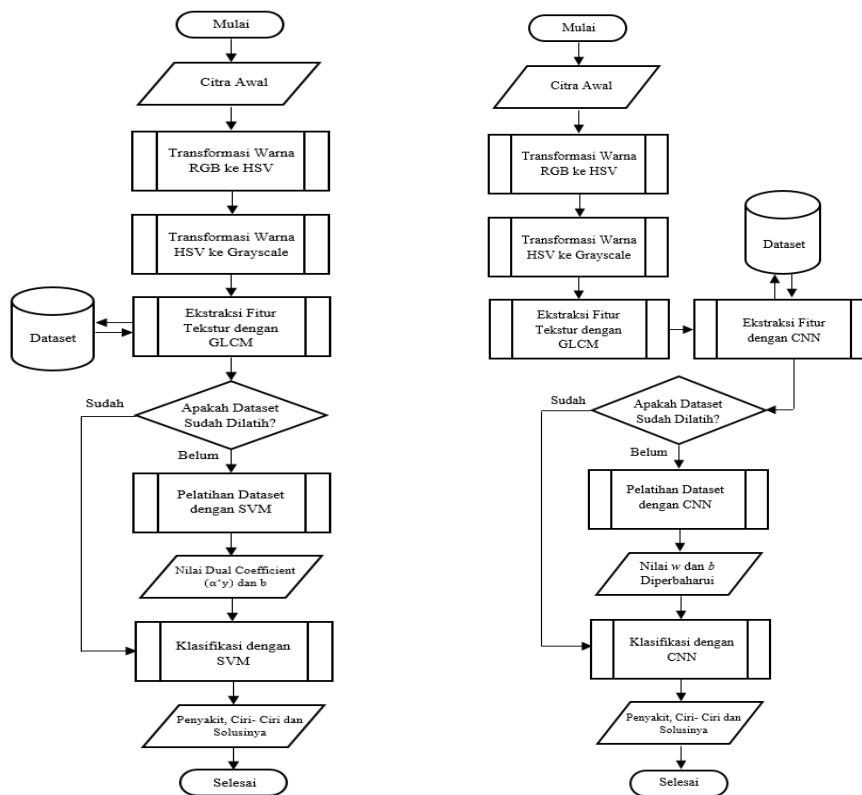
$$\text{Error Rate} = \frac{\text{jumlah gambar yang salah diidentifikasi}}{\text{jumlah gambar}} \quad (23)$$

$$\text{Precision} = \frac{TP}{(TP + FP)} \quad (24)$$

$$\text{Recall} = \frac{TP}{(TP + FN)} \quad (25)$$

2.2 Analisis Proses

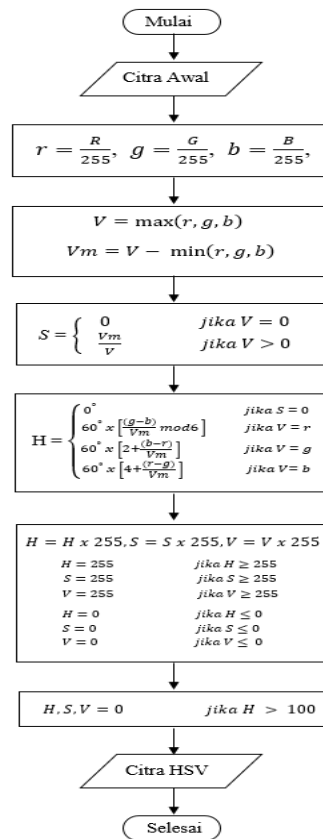
Sistem deteksi dan pengenalan penyakit pada citra daun tomat yang dikembangkan pada penelitian ini dimodelkan dengan *flowchart* yang terdapat pada Gambar (3).



Gambar 3 *Flowchart* Sistem yang Dikembangkan dengan (a) SVM dan (b) CNN

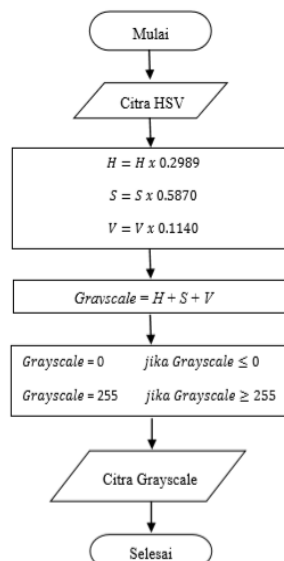
2.2.1 Transformasi Warna RGB ke HSV

Pada tahapan ini dilakukan proses transformasi warna dari citra awal RGB (*Red Green Blue*) ke HSV (*Hue Saturation Value*). Keterangan rumus yang digunakan pada proses terdapat pada subbab 2.2.2. Tahapan- tahapan dalam melakukan transformasi warna dari citra RGB (*Red Green Blue*) ke HSV (*Hue Saturation Value*) dimodelkan oleh *flowchart* pada Gambar (4).

Gambar 4. *Flowchart* Proses Transformasi Warna RGB ke HSV

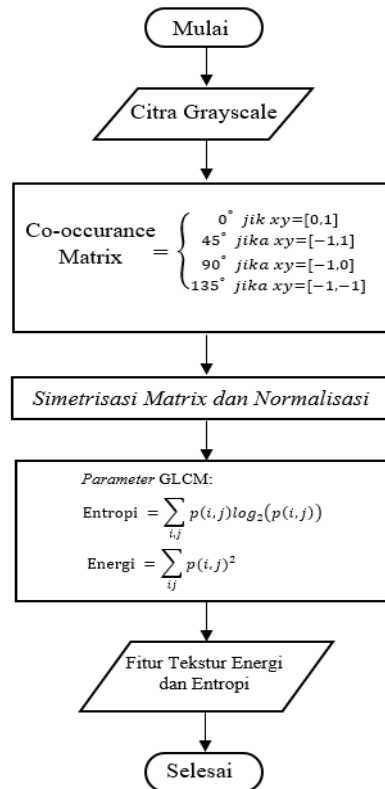
2.2.2 Transformasi Warna HSV ke *Grayscale*

Pada tahapan ini, dilakukan proses transformasi warna citra HSV (*Hue Saturation Value*) ke *grayscale*. Transformasi ini dilakukan agar proses terfokus pada satu *layer* dan perhitungan akan menjadi lebih cepat. Proses perhitungan sama seperti perhitungan pada konversi citra RGB (*Red Green Blue*) ke *grayscale* yang keterangan rumusnya dapat dilihat pada subbab 2.2.1. Adapun proses- prosesnya dimodelkan oleh *flowchart* pada Gambar (5).

Gambar 5. *Flowchart* Proses Transformasi Warna HSV ke *Grayscale*

2.2.3 Ekstraksi Fitur GLCM

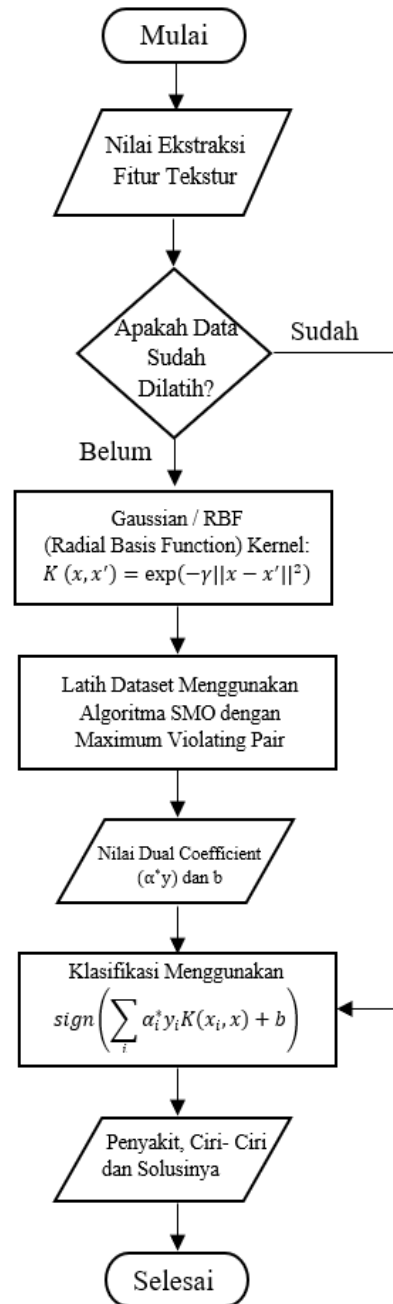
Pada tahap ini dilakukan proses ekstraksi fitur tekstur dari citra *grayscale* dengan GLCM (*Gray Level Co-occurrence Matrix*). Ekstraksi fitur tekstur dilakukan untuk mengambil ciri khas pada gambar, fitur ini sangat berpengaruh dalam pengklasifikasian data untuk menentukan penyakit yang diderita daun. Langkah-langkah dalam melakukan ekstraksi fitur GLCM (*Gray Level Co-occurrence Matrix*) dimodelkan dalam *flowchart* pada Gambar (6).



Gambar 6. *Flowchart* Ekstraksi Fitur GLCM

2.2.4 Pelatihan dan Pengujian dengan SVM

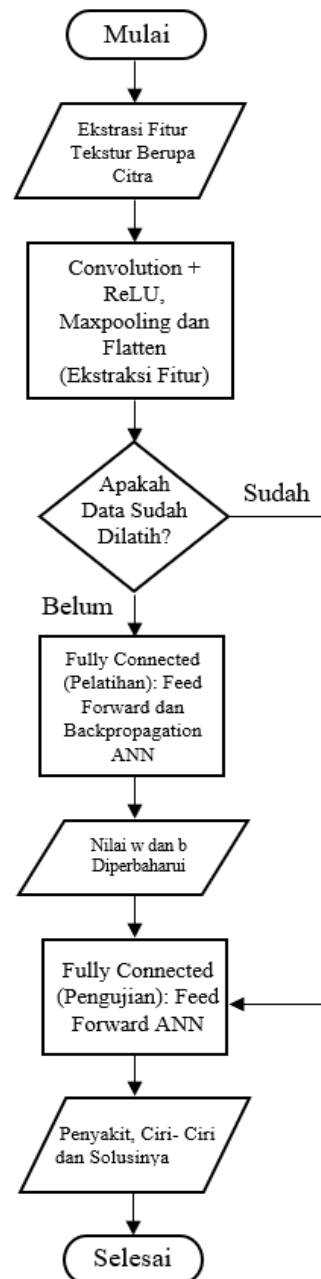
Proses GLCM (*Gray Level Co-occurrence Matrix*) menghasilkan 2 data fitur tekstur yang berupa nilai bertipe *float*. Kedua nilai ekstraksi fitur tersebut disimpan ke dalam *dataset*. *Dataset* akan digunakan sebagai data pelatihan pada SVM (*Support Vector Machine*). Tahapan-tahapan untuk pelatihan dan pengujian terdapat di dalam *flowchart* pada Gambar (7).



Gambar 7. Flowchart Pelatihan dan Pengujian (Klasifikasi) SVM

2.2.5 Pelatihan dan Pengujian dengan CNN

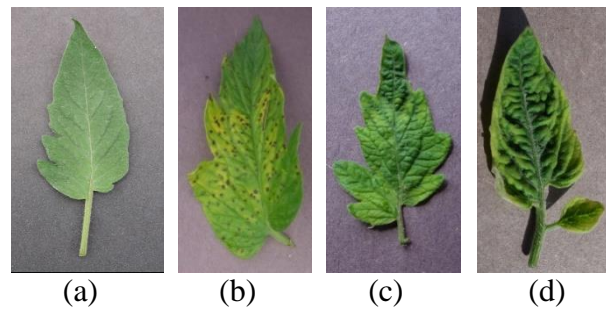
Ekstraksi fitur tekstur dari proses GLCM (*Gray Level Co-occurrence Matrix*) sebelumnya menghasilkan 2 (dua) data fitur tekstur (energi dan entropi) berupa matriks. Matriks-matriks tersebut kemudian di-convert ke citra. Citra digunakan sebagai masukan pada proses *convolution* dan ReLU (*Rectified Linear Unit*), *maxpooling (downsampling)* dan *flatten*. Proses-proses tersebut adalah proses ekstraksi fitur pada metode CNN (*Convolutional Neural Network*). Proses ekstraksi fitur dan fitur yang dihasilkan oleh CNN (*Convolutional Neural Network*) berbeda dengan proses ekstraksi fitur dan fitur tekstur yang dihasilkan oleh GLCM (*Gray Level Co-occurrence Matrix*). Tahapan-tahapan penggunaan metode CNN (*Convolutional Neural Network*) dapat dimodelkan dengan *flowchart* yang terdapat pada Gambar (8).



Gambar 8. Flowchart Pelatihan dan Pengujian (Klasifikasi) CNN

3. HASIL DAN PEMBAHASAN






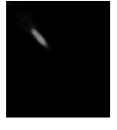



Pada penelitian ini digunakan 200 citra daun tomat yang terbagi dari empat kelas data penyakit (daun sehat, daun terserang penyakit *septorial leaf spot*, *tomato mosaic* dan *tomato yellow leaf curl*), kemudian dibagi menjadi 160 citra sebagai data latih dan 40 citra sebagai data uji. Berikut merupakan sampel data dari masing-masing kelas penyakit pada penelitian ini yang ditunjukkan pada Gambar (9).












Gambar 9. Sampel data (a) daun sehat, (b) daun terserang penyakit *septorial leaf spot*, (d) *tomato mosaic* dan (c) *tomato yellow leaf curl*

Semua data citra melewati proses transformasi warna RGB (*Red Green Blue*) ke HSV (*Hue Saturation Value*), HSV (*Hue Saturation Value*) ke *grayscale* dan ekstraksi fitur tekstur menggunakan GLCM (*Gray Level Co-occurrence Matrix*). GLCM (*Gray Level Co-occurrence Matrix*) menghasilkan dua parameter fitur tekstur yaitu energi dan entropi. Terdapat dua hasil tipe variabel ekstraksi fitur tekstur oleh GLCM (*Gray Level Co-occurrence Matrix*). Yang pertama berupa nilai *float* dan yang kedua berupa citra. Fitur tekstur yang bertipe nilai *float* digunakan untuk nilai masukan pada klasifikasi SVM (*Support Vector Machine*), sedangkan fitur tekstur yang berupa citra digunakan untuk nilai masukan pada klasifikasi menggunakan CNN (*Convolutional Neural Network*). Tabel (1) dan Tabel (2) menunjukkan hasil dari proses klasifikasi SVM (*Support Vector Machine*) dan CNN (*Convolutional Neural Network*) pada ekstraksi fitur tekstur GLCM (*Gray Level Co-occurrence Matrix*).

Tabel 1 Daftar Sampel Hasil Proses Pengujian Daun Sehat

NO	Citra Masukan (Citra RGB)	Ekstraksi Fitur (GLCM)				Hasil Klasifikasi Penyakit	
		SVM		CNN		SVM	CNN
		Energi	Entropi	Energi	Entropi		
1		12.595687171060508	10.424199572657875			Sehat	Sehat
2		14.945856043352434	10.178255513115607			Sehat	Sehat
3		11.623859761209868	10.485482226931657			Sehat	Sehat

Tabel 2 Daftar Sampel Hasil Proses Pengujian Daun Berpenyakit (*Tomato Yellow Leaf Curl*)

NO	Citra Masukan (Citra RGB)	Ekstraksi Fitur (GLCM)				Hasil Klasifikasi Penyakit	
		SVM		CNN		SVM	CNN
		Energi	Entropi	Energi	Entropi		
1		2.426313450004592	12.495121774891063			Yellow Leaf Curl	Yellow Leaf Curl
2		2.2436281270747176	12.58303462382472			Yellow Leaf Curl	Yellow Leaf Curl
3		2.699707349403331	12.383003821242504			Yellow Leaf Curl	Yellow Leaf Curl

Pada Tabel (1) hasil proses pengujian yang dilakukan pada daun sehat, rata-rata nilai energi yang dihasilkan oleh ekstraksi fitur tekstur untuk klasifikasi SVM (*Support Vector Machine*) adalah ± 11 sampai 14 dan rata-rata nilai entropi yang muncul adalah ± 10 . Sedangkan pada Tabel (2) hasil proses pengujian yang dilakukan pada daun terserang penyakit *tomato yellow leaf curl*, rata-rata nilai energi yang dihasilkan adalah ± 2 dan rata-rata nilai entropi adalah ± 12 . Sehingga dari Tabel (1) dan Tabel (2) dapat diketahui bahwa hasil dari proses klasifikasi daun sehat dan daun berpenyakit dipengaruhi oleh ekstraksi fitur tekstur (energi dan entropi). Ekstraksi fitur tekstur menghasilkan nilai-nilai yang unik. Pada penyakit yang sama, nilai dan citra hasil proses ekstraksi fitur tekstur saling menyerupai (jarak diantara nilai *float* fitur tidak begitu jauh dan citra fitur yang dihasilkan memiliki kemiripan).

Adapun pengujian pada penelitian ini dilakukan terhadap 40 citra data uji. Hasil dari proses pengujian terdapat pada Tabel (3).

Tabel 3 Daftar Hasil Proses Pengujian SVM dan CNN

No	Citra Masukan	Penyakit Yang Dikenali Oleh Sistem							
		Sehat		Septorial Leaf Spot		Tomato Mosaic		Tomato Yellow Leaf Curl	
		SVM	CNN	SVM	CNN	SVM	CNN	SVM	CNN
1	Daun Sehat_1.jpg	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						
2	Daun Sehat_2.jpg	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						
3	Daun Sehat_3.jpg	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						
4	Daun Sehat_4.jpg	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						
5	Daun Sehat_5.jpg	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						
6	Daun Sehat_6.jpg	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						
7	Daun Sehat_7.jpg	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						
8	Daun Sehat_8.jpg	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						
9	Daun Sehat_9.jpg	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						
10	Daun Sehat_10.jpg	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						
11	Septorial Leaf Spot_1.jpg			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
12	Septorial Leaf Spot_2.jpg			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
13	Septorial Leaf Spot_3.jpg			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
14	Septorial Leaf Spot_4.jpg			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
15	Septorial Leaf Spot_5.jpg			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
16	Septorial Leaf Spot_6.jpg			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
17	Septorial Leaf Spot_7.jpg			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
18	Septorial Leaf Spot_8.jpg			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
19	Septorial Leaf Spot_9.jpg			×	<input checked="" type="checkbox"/>				
				(terdeteksi: tomato mosaic)					
20	Septorial Leaf Spot_10.jpg			×	<input checked="" type="checkbox"/>				
				(terdeteksi: tomato mosaic)					
21	Tomato Mosaic_1.jpg					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
22	Tomato Mosaic_2.jpg					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
23	Tomato Mosaic_3.jpg					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
24	Tomato Mosaic_4.jpg					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
25	Tomato Mosaic_5.jpg					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
26	Tomato Mosaic_6.jpg					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
27	Tomato Mosaic_7.jpg					<input checked="" type="checkbox"/>	×		
							(terdeteksi: tomato yellow leaf curl)		
28	Tomato Mosaic_8.jpg					×	<input checked="" type="checkbox"/>		
						(terdeteksi: tomato yellow leaf curl)			
29	Tomato Mosaic_9.jpg					×	×		
						(terdeteksi: tomato yellow leaf curl)	(terdeteksi: septorial leaf spot)		
30	Tomato Mosaic_10.jpg					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
31	Tomato Yellow Leaf Curl_1.jpg							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
32	Tomato Yellow Leaf Curl_2.jpg							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
33	Tomato Yellow Leaf Curl_3.jpg							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
34	Tomato Yellow Leaf Curl_4.jpg							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
35	Tomato Yellow Leaf Curl_5.jpg							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
36	Tomato Yellow Leaf Curl_6.jpg							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
37	Tomato Yellow Leaf Curl_7.jpg							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
38	Tomato Yellow Leaf Curl_8.jpg							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
39	Tomato Yellow Leaf Curl_9.jpg							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
40	Tomato Yellow Leaf Curl_10.jpg							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Dari hasil pengujian yang dilakukan pada 40 citra data uji yang terdapat pada tabel (3) maka dapat dilakukan perhitungan untuk menentukan *accuracy*, *precision*, *recall* dan *error* pada tiap masing- masing kelas data penyakit. Setelah selesai dilakukan perhitungan *accuracy*, *precision*, *recall* dan *error* dari keempat kelas data penyakit tersebut, maka dilakukan pencarian rata- rata dari keempat pengujian tersebut. Hasil perhitungan ditampilkan pada Tabel (4) dan Tabel (5).

Tabel 4 Nilai Persentase Klasifikasi

Deteksi Penyakit	TP		FP		TN		FN		Accuracy		Precision		Recall		Error	
	SVM	CNN	SVM	CNN	SVM	CNN	SVM	CNN	SVM	CNN	SVM	CNN	SVM	CNN	SVM	CNN
Sehat	10	10	0	0	30	30	0	0	100%	100%	100%	100%	100%	100%	0%	0%
Septorial Leaf Spot	8	10	0	1	28	29	2	0	95%	97.5%	100%	90.9%	80%	100%	20%	0%
Tomato Mosaic	8	8	2	0	28	30	2	2	90%	95%	80%	100%	80%	80%	20%	20%
Tomato Yellow Leaf Curl	10	10	0	0	30	30	0	0	95%	97.5%	83.33%	90.9%	100%	100%	0%	0%
Rata-rata									95%	97.5%	90.83%	95.45%	90%	95%	10%	5%

Tabel 5 Nilai Rata- Rata Persentase Klasifikasi

Jenis Pengujian	Persentase Hasil Pengujian	
	SVM	CNN
Rata- Rata <i>Accuracy</i>	95%	97.5%
Rata- Rata <i>Precision</i>	90.83%	95.45%
Rata- Rata <i>Recall</i>	90%	95%
Rata- Rata <i>Error</i>	10%	5%

Hasil yang ditunjukkan oleh Tabel (5) menyatakan bahwa persentase pengujian CNN (*Convolutional Neural Network*) yaitu rata- rata *accuracy*, *precision*, *recall* lebih tinggi dan juga persentase nilai *error* lebih rendah dibandingkan hasil dari pengujian SVM (*Support Vector Machine*). Semakin tinggi nilai dari *accuracy*, *precision* dan *recall* maka semakin baik klasifikasi yang dilakukan oleh sistem, sedangkan semakin kecil nilai dari *error* maka semakin kecil kemungkinan sistem mengalami kesalahan di dalam mengidentifikasi penyakit. Oleh karena itu, terlihat dengan jelas pada penelitian ini didapati CNN (*Convolutional Neural Network*) merupakan *classifier* yang lebih baik dibandingkan SVM (*Support Vector Machine*).

4. KESIMPULAN

Berdasarkan hasil dari penelitian yang telah dilakukan terhadap deteksi dan pengenalan penyakit tumbuhan pada citra daun tomat dapat disimpulkan sebagai berikut:

1. Hasil pengujian sangat dipengaruhi oleh ekstraksi fitur tekstur (energi dan entropi), karena pelatihan yang dilakukan hanya berdasarkan fitur tekstur saja. Oleh karena itu, perlu dilakukan *preprocessing* yang tepat sehingga dapat memperjelas tekstur dari citra. Tekstur yang semakin jelas dapat memberikan ciri khas dari penyakit pada suatu citra tersebut, sehingga pengujian pada sistem akan menjadi lebih baik.
2. Dari 200 citra daun tomat, 160 citra sebagai data latih dan 40 citra sebagai data uji. Pengujian dengan algoritma SVM (*Support Vector Machine*) untuk mengidentifikasi penyakit yang

diderita oleh daun tomat menghasilkan rata-rata *accuracy* adalah 95%, rata-rata *precision* adalah 90.83%, rata-rata *recall* adalah 90% dan rata-rata *error* adalah 10%. Sedangkan pengujian dengan algoritma CNN (*Convolutional Neural Network*) menghasilkan nilai yang lebih baik yaitu rata-rata *accuracy* adalah 97.5%, *precision* adalah 95.45%, *recall* adalah 95% dan *error* adalah 5%.

3. Perbandingan hasil antara algoritma SVM (*Support Vector Machine*) dengan CNN (*Convolutional Neural Network*) didapati bahwa CNN (*Convolutional Neural Network*) adalah *classifier* yang lebih baik digunakan untuk mendeteksi penyakit daun pada citra daun tomat di dalam penelitian ini.

5. SARAN

Adapun beberapa saran yang dapat disampaikan untuk penelitian yang lebih lanjut adalah sebagai berikut:

1. Penelitian ini hanya dapat mengenali penyakit pada satu objek daun di dalam citra, diharapkan penelitian selanjutnya dapat mengenali berbagai objek daun sekaligus di dalam suatu citra.
2. Untuk ekstraksi fitur hanya menggunakan fitur tekstur GLCM (*Gray Level Co-occurrence Matrix*). Ekstraksi fitur tersebut sangat baik digunakan sebagai acuan untuk data pelatihan dan pengujian dengan daun pada tanaman yang sama. Tetapi, agar dapat mendeteksi lebih dari 1 (satu) tipe tanaman, karena pada tanaman yang berbeda baik warna dan bentuk daun juga berbeda-beda. Maka dapat digunakan fitur warna dan bentuk agar mendapatkan ciri khas dari daun pada tanaman tersebut.
3. Pada penelitian selanjutnya dapat dilakukan operasi morfologi terlebih dahulu sebelum ekstraksi fitur. Dengan operasi morfologi sistem dapat menyeleksi penyakit yang terdapat pada daun. Sehingga penelitian lebih spesifik ke arah penyakit yang timbul pada daun dan diharapkan hasil pengujian yang didapat lebih maksimal.
4. Aplikasi dapat dibuat berbasis android supaya aplikasi identifikasi dapat digunakan secara *mobile*.

DAFTAR PUSTAKA

- [1] Kiswondo, S., 2011. PENGGUNAAN ABU SEKAM DAN PUPUK ZA TERHADAP PERTUMBUHAN DAN HASIL TANAMAN TOMAT (*Lycopersicum esculentum* Mill). *EMBRYO*, VIII(1), pp. 9-17.
- [2] Gunaeni, N. & Purwati, E., 2013. Uji Ketahanan terhadap Tomato Yellow Leaf Curl Virus pada Beberapa Galur Tomat (Resistance Test of Tomato Lines to Tomato Yellow Leaf Curl Virus). *J.Hort*, XXIII(1), pp. 65-71.
- [3] Strange, R. N. & Scott, P. R., 2005. PLANT DISEASE: A Threat to Global Food Security. *Annual Review of Phytopathology*, 2 February, pp. 83-116.
- [4] Mentari, M., Ginardi, R. V. H. & Fatichah, C., 2015. SEGMENTASI PENYAKIT PADA CITRA DAUN TEBU MENGGUNAKAN FUZZY C MEANS – SUPPORT VECTOR MACHINE DENGAN FITUR WARNA a*. *Jurnal Ilmiah Teknologi Informasi*, XIII(1), pp. 45-52.
- [5] Rakhmawati, P. U., Pranoto, Y. M. & Setyati, E., 2018. KLASIFIKASI PENYAKIT DAUN KENTANG BERDASARKAN FITUR TEKSTUR DAN FITUR WARNA MENGGUNAKAN SUPPORT VECTOR MACHINE. *Seminar Nasional Teknologi dan Rekayasa (SENTRA)*, pp. 1-8.
- [6] Alviansyah, F., Ruslianto, I. & Diponegoro, M., 2017. IDENTIFIKASI PENYAKIT PADA TANAMAN TOMAT BERDASARKAN WARNA DAN BENTUK DAUN DENGAN

- METODE NAIVE BAYES CLASSIFIER BERBASIS WEB. *Jurnal Coding Sistem Komputer Untan*, V(1), pp. 23-32.
- [7] Pujari, J. D., Yakkundimath, R. & Byadgi, A. S., 2016. SVM and ANN Based Classification of Plant Diseases Using Feature Reduction Technique. *International Journal of Interactive Multimedia and Artificial Intelligence*, 3(7), pp. 6-14.
- [8] Blessy, A. & Wise, D. D. C. J. W., 2018. Detection of Affected Part of Plant Leaves and Classification of Diseases Using CNN Technique. *International Journal of Engineering and Techniques*, 4(2), pp. 823-829.
- [9] Jayaraman, S., Esakkirajan, S. & Veerakumar, T., 2009. *Digital Image Processing*. 1st ed. New Delhi: Tata McGraw Hill.
- [10] Hidayatullah, P., 2017. *Pengolahan Citra Digital Teori dan Aplikasi Nyata*. 1st ed. Bandung: Informatika Bandung.
- [11] Kadir, A. & Susanto, A., 2013. *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta: Andi Offset.
- [12] Wibowo, F. & Harjoko, A., 2017. Klasifikasi Mutu Pepaya Berdasarkan Ciri Tekstur GLCM Menggunakan Jaringan Saraf Tiruan. *Jurnal Nasional Pendidikan Teknik Informatika*, III(2), pp. 100-104.
- [13] Nello, C. & Taylor, J. S., 2000. An Introduction to Support Vector Machines. *Cambridge University Press*.
- [14] Kowalczyk, A., 2017. Support Vector Machines Succinctly. In: C. Wright, ed. *Support Vector Machines Succinctly*. Morrisville: Syncfusion, pp. 13-114.
- [15] Bottou, L. & Lin, C.-J., 2006. *Support Vector Machine Solvers*. [Online] Available at: <http://leon.bottou.org/publications/pdf/lin-2006.pdf>, diakses tgl 5 Mei 2019.
- [16] Escontrela, A., 2018. *Towards Data Science*. [Online] Available at: <https://towardsdatascience.com/convolutional-neural-networks-from-the-ground-up-c67bb41454e1>, diakses tgl 6 Desember 2018.
- [17] Prabhu, 2018. *Understanding of Convolutional Neural Network (CNN)—Deep Learning*. [Online] Available at: <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>, diakses tgl 4 Mei 2019.
- [18] Nielsen, M., 2018. *CHAPTER 3 Improving The Way Neural Networks Learn*. [Online] Available at: http://neuralnetworksanddeeplearning.com/chap3.html#introducing_the_cross-entropy_cost_function, diakses tgl 5 Mei 2019.
- [19] Sivasurya, 2018. *Neural Networks from Scratch. Easy vs hard*. [Online] Available at: <https://towardsdatascience.com/neural-networks-from-scratch-easy-vs-hard-b26ddc2e89c7>, diakses tgl 5 Mei 2019.
- [20] Narkhede, S., 2018. *Understanding Confusion Matrix*. [Online] Available at: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>, diakses tgl 23 Juli 2019.