



## Klasifikasi Hama Dan Penyakit Daun Kapas Berdasarkan Fitur Tekstur Dan Warna Menggunakan Multilayer Perceptron

Nadhif Mahardika Awandi✉, Budi Nugroho, dan Fawwaz Ali Akbar

Jurusan Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Indonesia

### Info Artikel

#### Sejarah Artikel:

Diterima: 27 Juni 2023

Direvisi: 24 Oktober 2023

Disetujui: 27 Desember 2023

#### Keywords:

Penyakit Tanaman Kapas, GLCM, Color Moment, MLP

### Abstrak

Kapas merupakan bahan baku dari Industri Tekstil dan Produk Tekstil (TPT) di Indonesia. Namun, produksi kapas di Indonesia tidak dapat memenuhi kebutuhan nasional. Untuk meningkatkan produksi kapas, perlu dilakukan identifikasi penyakit dan hama pada tanaman kapas sehingga mendapatkan penanganan yang tepat dan mengatasi terjadinya kegagalan panen. Beberapa penelitian telah dilakukan mengenai identifikasi penyakit pada daun kapas dengan menggunakan kecerdasan buatan. Tetapi, penelitian-penelitian yang telah dilakukan sebelumnya lebih berfokus pada klasifikasi penyakit yang terjadi pada daun kapas. Oleh karena itu, penelitian ini memiliki tujuan untuk membuat sistem yang tidak hanya dapat mengklasifikasi penyakit daun kapas, tetapi dapat juga mengklasifikasi hama yang terdapat pada daun kapas. Berdasarkan pengujian yang telah dilakukan, Tingkat akurasi dalam klasifikasi pada hama dan penyakit tanaman kapas pada data uji menghasilkan nilai akurasi tertinggi sebesar 90%. Nilai tersebut didapatkan pengujian dengan kombinasi fitur Color Moment dan fitur GLCM dengan sudut  $0^\circ$  sebagai input serta arsitektur MLP menggunakan fungsi aktivasi tanh dan 1024 *perceptron* di *hidden layer*.

### Abstract

Cotton is a raw material for the Textile and Textile Products Industry in Indonesia. However, cotton production in Indonesia cannot meet national needs. To increase cotton production, it is necessary to identify diseases and pests in cotton plants so that they receive appropriate treatment and overcome crop failures. Several studies have been carried out regarding the identification of diseases on cotton leaves using artificial intelligence. However, previous research has focused more on the classification of diseases that occur on cotton leaves. Therefore, this research aims to create a system that can not only classify cotton leaf diseases but can also classify pests found on cotton leaves. Based on the tests that have been carried out, the level of accuracy in the classification of pests and diseases of cotton plants in the test data produces the highest accuracy value of 90%. This value was obtained by testing using a combination of the Color Moment feature and the GLCM feature with an angle of  $0^\circ$  as input and the MLP architecture using the tanh activation function and 1024 *perceptrons* in the *hidden layer*.

## PENDAHULUAN

Kapas merupakan bahan baku dari Industri Tekstil dan Produk Tekstil (TPT) di Indonesia. Pada tahun 2020 Indonesia melakukan impor kapas sebanyak 493.451 Ton, sedangkan produksi kapas di dalam negeri hanya sebesar 145 Ton (Kementerian Perindustrian, 2021). Hal tersebut menunjukkan bahwa produksi kapas di Indonesia tidak dapat memenuhi kebutuhan nasional, sehingga dipenuhi dengan cara impor. Peningkatan produksi erat kaitannya dengan teknik budidaya yang diterapkan oleh petani selain faktor iklim dan keadaan lahan atau tanah (Dewi, 2014). Untuk mengatasi terjadinya kegagalan panen, perlu dilakukan identifikasi penyakit dan hama pada tanaman kapas sehingga mendapatkan penanganan yang tepat.

Teknologi klasifikasi berbasis kecerdasan buatan dapat membantu petani mendapatkan hasil identifikasi yang lebih cepat dan tepat. Beberapa penelitian telah dilakukan mengenai identifikasi penyakit pada daun kapas, seperti penelitian yang dilakukan oleh (Amin dkk., 2022) mengenai klasifikasi penyakit daun kapas dengan *neural network*. Penelitian tersebut menggunakan *pre-trained* model VGG-16 dan menghasilkan akurasi sebesar 99,99%.

Penelitian lainnya dilakukan oleh (Kant dkk., 2022) tentang prediksi penyakit daun kapas menggunakan algoritma *ResNet*. Penelitian tersebut menerapkan metode *deep learning* dengan arsitektur *ResNet* dan menghasilkan akurasi sebesar 97%. Selanjutnya penelitian yang dilakukan oleh (Susa dkk., 2022) mengenai deteksi dan klasifikasi daun kapas menggunakan algoritma YOLO. Penelitian tersebut membuat sistem yang dapat mendeteksi penyakit daun kapas dari gambar dan deteksi langsung melalui video, akurasi yang didapatkan dari sistem tersebut sebesar 95,09%.

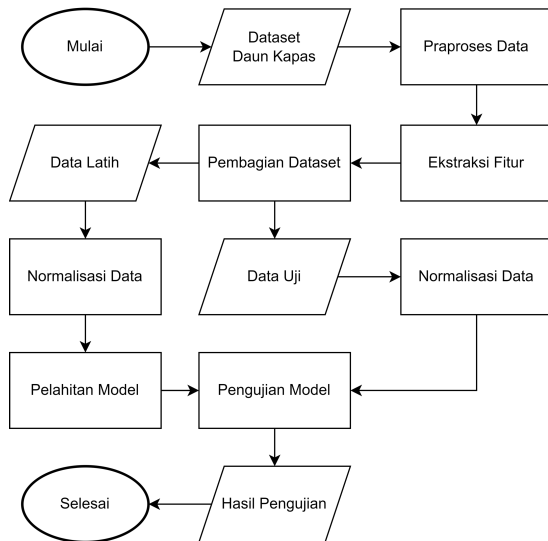
Kemudian penelitian yang dilakukan oleh (Bhimte & Thool, 2018) tentang klasifikasi penyakit daun kapas menggunakan *image processing* dan SVM. Penelitian tersebut menggunakan *Gray Level Co-occurrence Matrix* (GLCM) sebagai ekstraksi fitur serta SVM untuk klasifikasinya dan mendapatkan akurasi sebesar 98.46%. Penelitian lainnya yang dilakukan oleh (Patil & Burkpalli, 2022) mengenai klasifikasi penyakit daun kapas dengan menggunakan kombinasi fitur warna dan tekstur. Penelitian tersebut menggunakan fitur warna, *Gray Level Co-occurrence Matrix* (GLCM) serta *Local Binary Pattern* (LBP) dan melakukan perbandingan terhadap beberapa metode *machine learning*. Hasil penelitian tersebut menunjukkan bahwa metode

*Multilayer Perceptron* (MLP) mendapatkan akurasi paling tinggi sebesar 85%.

Namun, penelitian-penelitian yang telah dilakukan sebelumnya lebih berfokus pada klasifikasi penyakit yang terjadi pada daun kapas. Oleh karena itu, penelitian ini memiliki tujuan untuk membuat sistem yang tidak hanya dapat mengklasifikasi penyakit daun kapas, tetapi dapat juga mengklasifikasi hama yang terdapat pada daun kapas. Sistem yang akan dikembangkan akan menggunakan fitur tekstur dan warna dari daun kapas serta metode MLP untuk klasifikasinya. Metode *feature-based* dipilih karena interpretasi hasil yang mudah, efisiensi komputasi, dan kemampuan beroperasi dengan sumber daya terbatas dibandingkan dengan metode *deep learning* (Abdu dkk., 2020; Wagle & R, 2021). Ekstraksi fitur warna akan menggunakan metode Color Moment dan ekstraksi fitur tekstur menggunakan metode GLCM. Pemilihan metode GLCM didasarkan pada penelitian yang dilakukan oleh (Öztürk & Akdemir, 2018) yang menyatakan bahwa metode GLCM memperoleh akurasi kedua terbaik setelah SFTA dengan akurasi sebesar 92.8%. Menurut penelitian yang dilakukan oleh (Prabowo dkk., 2022) metode GLCM memiliki waktu komputasi yang lebih cepat dibandingkan dengan SFTA. Hasil penelitian ini diharapkan dapat membantu petani para petani untuk mendeteksi jenis hama dan penyakit yang menyerang tanaman kapas, sehingga dapat melakukan penanganan yang sesuai dan dapat meningkatkan produktivitas serta kualitas hasil panen.

## METODE PENELITIAN

Pada saat melakukan penelitian diperlukan langkah-langkah kerja guna mempermudah menyelesaikan penelitian dengan baik. Langkah-langkah kerja yang telah disusun ini bertujuan untuk untuk membantu dalam penyelesaian penelitian dengan baik dan mencapai tujuan penelitian yang telah ditetapkan. Langkah-langkah kerja serta tahapan penelitian ini, ditunjukkan pada Gambar 1.



Gambar 1. Langkah-langkah penelitian

#### A. Alat dan Bahan

Alat dan bahan yang digunakan pada penelitian ini adalah sebagai berikut.

1. Prosesor Intel Core i5-8265U
2. RAM 4 GB DDR4
3. Sistem Operasi Windows 11 Home
4. Python 3.10.9
5. Jupyter Notebook 6.5.2

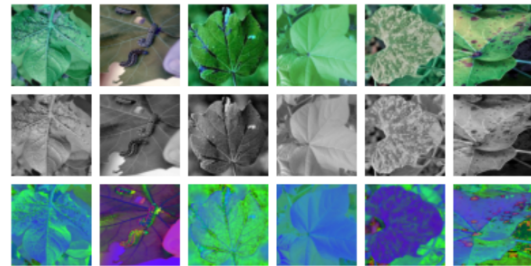
#### B. Pengumpulan Data

Citra hama dan penyakit tanaman kapas yang digunakan pada penelitian ini didapatkan dari website open source Kaggle (<https://www.kaggle.com/datasets/dhamur/cotton-plant-disease>) dengan nama "Cotton plant disease" yang diunggah oleh Dhamodharan. Dataset tersebut memiliki beberapa direktori dan jenis dataset. Dataset dari citra hama dan penyakit tanaman kapas yang digunakan terdapat di direktori "Cotton leaves". Total citra yang terdapat pada direktori tersebut berjumlah 4788 yang terdiri atas 800 citra hama *aphids*, 800 citra hama *army worm*, 800 citra penyakit *bacterial blight*, 800 citra penyakit *powdery mildew*, 788 citra penyakit *target spot* dan 800 citra daun yang sehat.

#### C. Praproses Data

Sebelum masuk pada proses ekstraksi fitur, seluruh citra akan diproses terlebih dahulu dalam tahap praproses data. Pada dataset "Cotton plant disease" terdapat berbagai macam ukuran citra sehingga diperlukan praproses pada dataset tersebut. Pada tahap praproses data adalah dengan memuat dataset citra kemudian mengubah semua ukuran data atau *resizing citra* menjadi ukuran yang sama yaitu dengan ukuran 256 × 256. Setelah semua ukuran citra disamakan

ukurannya, langkah selanjutnya yaitu mengubah warna citra dari RGB menjadi warna *grayscale*. Citra *grayscale* akan digunakan pada metode GLCM. Langkah selanjutnya yaitu mengubah warna citra dari RGB menjadi warna HSV, citra HSV akan digunakan pada metode *Color Moment*.



Gambar 3. Perubahan citra hasil praproses

#### D. Ekstraksi Fitur Tekstur

Metode *Gray Level Co-occurrence Matrix* (GLCM) dimanfaatkan untuk mengekstraksi fitur tekstur citra. GLCM suatu matriks yang menyajikan distribusi spasial hubungan antara dua piksel dalam citra yang memiliki jarak dan orientasi tertentu. GLCM menghitung jumlah kemunculan hubungan ketetanggaan antara piksel-piksel pada citra dalam bentuk matriks berdasarkan jarak serta arah tertentu (Kadir & Susanto, 2013). Berikut merupakan langkah-langkah dari ekstraksi fitur tekstur dengan metode GLCM.

1. Menyiapkan citra *grayscale*
2. Membuat *matrix framework* berdasarkan parameter jarak dan sudut yang digunakan.
3. Selanjutnya *matrix framework* akan dijumlahkan dengan hasil transposnya, sehingga mendapatkan matriks simetris.
4. Setiap nilai elemen pada matriks simetris akan dibagi dengan jumlah total elemen yang terdapat di dalam matriks, sehingga menghasilkan matriks yang telah dinormalisasi.
5. Selanjutnya dilakukan ekstraksi fitur GLCM, seperti kontras, entropi, *inverse difference moment* (IDM), korelasi dan *angular second moment* (ASM).

ASM adalah jumlah dari kuadrat semua elemen dalam matriks GLCM dan menunjukkan seberapa seragam intensitas piksel dalam citra. Dalam GLCM, ASM mengukur kehalusan atau kekasaran permukaan citra. Nilai ASM yang tinggi menunjukkan bahwa pasangan piksel yang memiliki intensitas yang sama lebih terkonsentrasi pada beberapa koordinat, sehingga citra memiliki tekstur yang halus. Sebaliknya, nilai ASM yang rendah menunjukkan bahwa

pasangan piksel dengan intensitas yang sama tersebar secara acak di seluruh citra, sehingga citra memiliki tekstur yang kasar. (Lusiana dkk., 2019).

$$ASM = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P_{i,j}^2 \quad (1)$$

Kontras adalah sebuah ukuran yang menunjukkan perbedaan intensitas antara pasangan piksel pada citra. Kontras dapat dihitung dengan cara mengukur nilai intensitas antara setiap piksel dengan lokasi relatif. Jika semua elemen piksel pada citra mempunyai nilai yang sama secara keseluruhan, maka nilai kontrasnya adalah 0. (Kusanti & Haris, 2018).

$$Kontras = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P_{i,j} (i - j)^2 \quad (2)$$

*Inverse difference moment* merupakan ukuran yang menunjukkan homogen atau seragam distribusi intensitas piksel dari citra. Nilai homogenitas akan tinggi apabila variasi intensitas pada citra rendah, sehingga memperlihatkan citra yang lebih halus dan tidak memiliki perubahan yang drastis antara satu area dengan area lainnya. Sebaliknya, jika variasi intensitas pada citra semakin tinggi, maka nilai homogenitasnya akan semakin kecil, mengindikasikan perbedaan intensitas yang signifikan antara piksel-piksel pada citra. (Priyanka & Kumar, 2020).

$$IDM = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{P_{i,j}}{1+(i-j)^2} \quad (3)$$

Entropi (Entropy) mengukur tingkat ketidakteraturan atau kompleksitas tekstur pada citra, dan dihitung sebagai jumlah dari seluruh elemen GLCM dikalikan dengan logaritma natural dari elemen tersebut. Jika nilai elemen GLCM relatif sama, artinya citra memiliki struktur atau pola yang sederhana dan berulang-ulang, sehingga nilai entropi-nya rendah. Sedangkan jika nilai elemen GLCM lebih variatif, artinya citra memiliki struktur atau pola yang kompleks dan tidak berulang-ulang, sehingga nilai entropi-nya tinggi (Widodo dkk., 2018).

$$Entropi = - \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P_{i,j} \log P_{i,j} \quad (4)$$

Korelasi pada metode GLCM mengukur seberapa kuat hubungan linier antara keberadaan pasangan piksel dengan keabuan tertentu pada citra. Semakin tinggi nilai korelasi, semakin kuat keterkaitan linier antara pasangan piksel dengan keabuan tertentu pada citra. (Priyanka & Kumar, 2020).

$$Korelasi = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P_{i,j} (i-\mu)(j-\mu)}{\sigma^2} \quad (5)$$

$$\mu = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P_{i,j} i \quad (6)$$

$$\sigma^2 = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P_{i,j} (i - \mu)^2 \quad (7)$$

#### E. Ekstraksi Fitur Warna

Metode *Color Moment* pada penelitian ini digunakan untuk mengekstraksi fitur warna citra. Dalam menggunakan *Color Moments*, terdapat tiga jenis moment yang digunakan, yaitu rata-rata, standar deviasi, dan *skewness* (Nisa dkk., 2022). Rata-rata merepresentasikan rata-rata warna pada citra, standar deviasi merepresentasikan variasi warna pada citra, dan *skewness* merepresentasikan kemiringan distribusi warna pada citra. Berikut merupakan langkah-langkah dari ekstraksi fitur tekstur dengan metode GLCM.

1. Menyiapkan citra dengan ruang warna HSV. Setelah menyiapkan citra HSV.
2. Selanjutnya memisahkan setiap elemen pada citra HSV.
3. Kemudian pada setiap nilai *hue*, *saturation*, *value* akan dilakukan perhitungan rata-rata, standar deviasi dan *skewness*.

#### F. Pembagian Dataset

Setelah diperoleh fitur tekstur dan fitur warna, akan dilakukan pembagian pada dataset dengan data latih sebanyak 3830 data (80% dari dataset) dan data uji sebanyak 958 data (20% dari dataset). Data latih akan digunakan sebagai sampel data masukan (*input*) dalam proses pelatihan model klasifikasi, sementara data uji akan digunakan untuk mengevaluasi kinerja model yang telah dilatih.

#### G. Normalisasi Data

Setelah dilakukan ekstraksi fitur dan pembagian data menjadi data latih dan data uji, dilakukanlah proses normalisasi data untuk memastikan bahwa data memiliki skala yang sama dan tidak ada variabel data yang mendominasi variabel data lainnya. Metode normalisasi yang digunakan pada penelitian ini adalah *Z-Score*. Normalisasi *Z-Score* mengubah semua data pada skala standar dengan rata-rata nol dan standar deviasi satu. Hal ini dilakukan dengan mengurangi setiap nilai data dengan rata-rata seluruh data, kemudian membaginya dengan standar deviasi dari seluruh data.

#### H. Klasifikasi

Tahapan klasifikasi pada penelitian ini, akan memanfaatkan metode *Multilayer Perceptron* (MLP). MLP merupakan jenis jaringan syaraf tiruan yang sudah sering digunakan (Sinaga, 2020). MLP merupakan salah satu

topologi jaringan saraf tiruan yang terdiri dari *perceptron-perceptron* yang saling terhubung dan membentuk beberapa lapisan. MLP terdiri dari tiga lapisan yakni lapisan masukan, lapisan tersembunyi, dan lapisan keluaran. (Irfan dkk., 2017). Terdapat dua proses klasifikasi yang harus dilakukan, yakni pelatihan model dan pengujian model.

Pelatihan dilakukan untuk mencari nilai bobot yang menghubungkan antar lapisan sehingga diperoleh hasil uji dengan error yang minimum. Data latih dari hasil ekstraksi fitur akan digunakan pada pelatihan model. Setelah dilakukan proses pelatihan maka diperoleh konfigurasi nilai bobot yang memberikan nilai akurasi tertinggi atau nilai error pembelajaran terendah. Berikut merupakan langkah-langkah dari pelatihan model dengan metode MLP.

1. Menentukan arsitektur MLP, seperti jumlah *perceptron* pada *hidden layer*, fungsi aktivasi, fungsi *loss* dan *optimizer*, jumlah *epoch* serta *learning rate*.
2. Selanjutnya melakukan inisialisasi bobot secara acak untuk setiap penghubung antara *perceptron* pada *input layer* dan *perceptron* pada *hidden layer* serta setiap penghubung antara *perceptron* pada *hidden layer* dan *perceptron* pada *output layer*. Setiap *perceptron* pada *hidden layer* dan *output layer* harus dilakukan juga inisialisasi bias.
3. Setelah itu dilakukan perhitungan nilai setiap *perceptron* pada *hidden layer* menggunakan Persamaan 8 dan dilanjutkan dengan perhitungan menggunakan fungsi aktivasi pada setiap nilai *perceptron* di *hidden layer* seperti pada Persamaan 9.
4. Selanjutnya dilakukan perhitungan nilai setiap *perceptron* pada *output layer* menggunakan Persamaan 10 dan dilanjutkan dengan perhitungan menggunakan fungsi aktivasi pada setiap nilai *perceptron* di *output hidden layer* seperti pada Persamaan 11.

$$net_j^h = \sum_{i=0} x_i w_{ji}^h + \theta_j^h \quad (8)$$

$$f_j^h = f_j^h(net_j^h) \quad (9)$$

$$net_k^o = \sum_{j=0} f_j^h w_{kj}^o + \theta_k^o \quad (10)$$

$$f_k^o = f_k^o(net_k^o) \quad (11)$$

5. Setelah itu dilakukan perhitungan error menggunakan fungsi *loss* dengan menggunakan persamaan 12.

$$E = -\sum y_i \ln \hat{y}_i \quad (12)$$

6. Kemudian dilanjutkan dengan membarui nilai bobot pada *output layer* seperti pada

Persamaan 13 dan membarui nilai bias pada *output layer* menggunakan Persamaan 14.

$$w_{j,l}^h(t+1) = w_{j,l}^h(t) - \eta \frac{\delta E}{\delta w_{j,l}^h} \quad (13)$$

$$w_{k,j}^o(t+1) = w_{k,j}^o(t) - \eta \frac{\delta E}{\delta w_{k,j}^o} \quad (14)$$

7. Selanjutnya dilanjutkan dengan membarui nilai bobot pada *hidden layer* menggunakan Persamaan 15 dan membarui nilai bias pada *hidden layer* menggunakan Persamaan 16.

$$\theta_j^h(t+1) = \theta_j^h(t) - \eta \frac{\delta E}{\delta \theta_j^h} \quad (15)$$

$$\theta_k^o(t+1) = \theta_k^o(t) - \eta \frac{\delta E}{\delta \theta_k^o} \quad (16)$$

8. Apabila nilai *epoch* belum mencapai maksimum, maka proses pelatihan akan diulangi. Tetapi jika nilai *epoch* sudah mencapai maksimum, maka proses pelatihan selesai.

Proses mengirimkan informasi dari *input layer* sampai *output layer* dinamakan *feed forward*. Sedangkan Untuk melatih *Multilayer Perceptron* (MLP), algoritma yang umumnya digunakan adalah *backpropagation*. Pada *backpropagation* dilakukan pembaharuan bobot secara bertahap (dari *output* ke *input layer*) berdasarkan error atau *loss* (Olivya dkk., 2018). Pembaharuan bobot pada suatu layer dipengaruhi oleh perubahan bobot pada layer sebelumnya. Untuk meminimalkan *loss*, algoritma *backpropagation* menggunakan metode *gradient descent* (Putra, 2020).

Setelah pelatihan model selesai dilakukan, tahap selanjutnya menguji model yang sudah dilatih menggunakan data uji. Pengujian model dilakukan pada data uji dan telah melewati tahapan *praproses*. Setelah proses pengujian nilai prediksi akan dibandingkan dengan nilai label asli yang terdapat pada data uji, dihitung berapa jumlah prediksi yang tepat. Langkah-langkah pengujian model pada metode MLP hampir sama dengan pada pelatihan model, tetapi tahapan yang digunakan hanya sampai *feed forward* saja tanpa melakukan pembaharuan variabel atau *backpropagation*.

## HASIL DAN PEMBAHASAN

Pada penelitian ini akan dilakukan dua skenario pengujian. Pengujian yang pertama dilakukan untuk mengetahui jenis fitur yang tepat untuk klasifikasi hama dan penyakit tanaman kapas. Sedangkan pengujian kedua dilakukan untuk mengetahui arsitektur *Multilayer Perceptron* (MLP) yang sesuai dalam klasifikasi

hama dan penyakit tanaman kapas. Kedua pengujian yang dilakukan akan menggunakan *learning rate* sebesar 0,01 dan *epoch* sebanyak 100.

Pengujian untuk mengetahui jenis fitur yang tepat akan dilakukan sebanyak 8 kali. Pengujian ini akan mengkombinasikan fitur tekstur dari *Gray Level Co-occurrence Matrix* (GLCM) dan fitur warna dari *Color Moment*. Dari beberapa fitur yang diusulkan oleh Haralick, hanya lima fitur yang digunakan oleh Newsam dan Kammath untuk GLCM, yaitu kontras, entropi, *inverse difference moment* (IDM), korelasi dan *angular second moment* (ASM). (Kadir &

Susanto, 2013). Oleh karena itu pada penelitian ini menggunakan lima fitur yang diusulkan oleh Newsam dan Kammath. Fitur-fitur ini diekstrak berdasarkan jarak 1 serta empat sudut 0°, 45°, 90°, dan 135°. Fitur warna dari metode Color Moment yang akan digunakan, yaitu rata-rata *hue*, rata-rata *saturation*, rata-rata *value*, standar deviasi *hue*, standar deviasi *saturation*, standar deviasi *value*, skewness *hue*, skewness *saturation* dan skewness *value*.

Tabel 1. Hasil Pengujian Fitur

No	Fitur	Presisi (%)	Recall (%)	F1-Score (%)	Akurasi (%)
1	GLCM dengan sudut 0°	41	39	36	39
2	GLCM dengan sudut 0° dan 45°	45	43	40	43
3	GLCM dengan sudut 0°, 45° dan 90°	47	46	43	45
4	GLCM dengan sudut 0°, 45°, 90° dan 135°	48	47	44	47
5	<i>Color Moment</i> dan GLCM dengan sudut 0°	<b>85</b>	<b>85</b>	<b>84</b>	<b>84</b>
6	<i>Color Moment</i> dan GLCM dengan sudut 0° dan 45°	84	83	83	83
7	<i>Color Moment</i> dan GLCM dengan sudut 0°, 45° dan 90°	83	83	83	83
8	<i>Color Moment</i> dan GLCM dengan sudut 0°, 45°, 90° dan 135°	83	82	82	82

Berdasarkan hasil pengujian dari Tabel 1 pada 8 jenis fitur, akurasi terendah didapatkan ketika menggunakan fitur GLCM dengan sudut 0° dengan nilai akurasi sebesar 38%. Sedangkan nilai akurasi tertinggi didapatkan pada pengujian dengan kombinasi fitur *Color Moment* dan fitur GLCM dengan sudut 0° yang mendapatkan nilai akurasi sebesar 85%. Dari hasil tersebut, fitur yang akan digunakan pada pelatihan dan pengujian untuk mencari arsitektur MLP yang paling sesuai adalah kombinasi fitur Color Moment dan fitur GLCM dengan sudut 0°.

Pengujian untuk menentukan arsitektur MLP yang paling sesuai akan dilakukan sebanyak 12 kali. Pengujian ini dilakukan dengan menggunakan tiga fungsi aktivasi pada *hidden layer*, yaitu *sigmoid*, *tanh* dan *relu*. Pada setiap fungsi aktivasi akan dilakukan empat pengujian, yaitu dengan jumlah *perceptron* 128, 256, 512, 1024 pada *hidden layer*.

Tabel 2. Hasil Pengujian Aktivasi Sigmoid

No	Jumlah Perceptron	Presisi (%)	Recall (%)	F1-Score (%)	Akurasi (%)
1	128	84	84	84	84
2	256	85	85	84	84
3	512	85	84	84	84
4	1024	<b>85</b>	<b>85</b>	<b>85</b>	<b>85</b>

Berdasarkan hasil pengujian fungsi aktivasi *sigmoid* dari Tabel 2, jumlah *perceptron*

128, 256, 512 pada *hidden layer* mendapatkan nilai akurasi yang sama yaitu sebesar 84%. Sedangkan nilai akurasi tertinggi didapatkan pada pengujian dengan jumlah *perceptron* 1024 pada *hidden layer* yang mendapatkan nilai akurasi sebesar 85%.

Tabel 3. Hasil Pengujian Aktivasi Tanh

No	Jumlah Perceptron	Presisi (%)	Recall (%)	F1-Score (%)	Akurasi (%)
1	128	86	86	86	86
2	256	86	86	86	86
3	512	88	88	88	88
4	1024	<b>90</b>	<b>90</b>	<b>90</b>	<b>90</b>

Berdasarkan hasil pengujian fungsi aktivasi *tanh* dari Tabel 3, jumlah *perceptron* 128, 256 pada *hidden layer* mendapatkan nilai akurasi yang sama yaitu sebesar 86%. Sedangkan nilai akurasi tertinggi didapatkan pada pengujian dengan jumlah *perceptron* 1024 pada *hidden layer* yang mendapatkan nilai akurasi sebesar 90%.

Tabel 4. Hasil Pengujian Aktivasi Relu

No	Jumlah Perceptron	Presisi (%)	Recall (%)	F1-Score (%)	Akurasi (%)
1	128	51	35	34	36
2	256	43	34	32	35
3	512	48	34	33	32
4	1024	<b>53</b>	<b>37</b>	<b>36</b>	<b>38</b>

Berdasarkan hasil pengujian fungsi aktivasi *relu* dari Tabel 4, akurasi terendah didapatkan ketika menggunakan jumlah *perceptron* 512 pada *hidden layer* dengan nilai akurasi sebesar 32%. Sedangkan nilai akurasi tertinggi didapatkan pada pengujian dengan jumlah *perceptron* 1024 pada *hidden layer* yang mendapatkan nilai akurasi sebesar 38%. Dari hasil pengujian terhadap jumlah *perceptron* pada *hidden layer*, dapat dikatakan bahwa peningkatan jumlah *perceptron* pada *hidden layer* tidak menjamin akan selalu meningkatkan nilai akurasi dan mungkin dapat menghasilkan nilai akurasi yang lebih rendah. Dari pelatihan dan pengujian yang dilakukan terhadap fungsi aktivasi *relu*, fungsi aktivasi *relu* tidak cocok digunakan pada penelitian ini. Hal tersebut dikarenakan fitur-fitur yang digunakan sebagai *input* pada proses pelatihan dan pengujian banyak yang bernilai negatif. Ketika nilai *input* negatif, fungsi *relu* akan mengubah output menjadi nol,

sehingga gradien untuk *perceptron* tersebut menjadi nol. Sebagai hasilnya, *perceptron* tersebut tidak lagi berkontribusi dalam pembelajaran dan dapat dianggap mati.

Dari perbandingan nilai akurasi tertinggi pada masing-masing fungsi aktivasi, nilai akurasi terendah didapatkan ketika menggunakan fungsi aktivasi *relu* dan 1024 *perceptron* di *hidden layer* dengan nilai akurasi sebesar 38%. Sedangkan nilai akurasi tertinggi didapatkan pada pengujian dengan fungsi aktivasi *tanh* dan 1024 *perceptron* di *hidden layer* yang mendapatkan nilai akurasi sebesar 90%. Dari hasil pengujian terhadap jumlah *perceptron* pada *hidden layer*, dapat dikatakan bahwa peningkatan jumlah *perceptron* pada *hidden layer* tidak menjamin akan selalu meningkatkan nilai akurasi dan mungkin hanya akan menghasilkan nilai akurasi yang sama ataupun mengalami penurunan.

Tabel 5. Hasil Pengujian Fitur

No	Penelitian	Metode	Kelas	Akurasi (%)
1	(Kant dkk., 2022)	ResNet	<i>Blue Disease, Leaf Curl, Mosaic Disease, Crown Gall, Fusarium Vilt, Bacterial Blight, Verticillium wilt, Grey mildew, Healthy</i>	97
2	(Amin dkk., 2022)	VGG16	<i>Bacterial Blight, Curl Virus, Verticillium Wilt, Healthy</i>	99,97
3	(Susa dkk., 2022)	YOLO	<i>Diseased Leaf, Healthy Leaf, Diseased Plant, Healthy Plant</i>	95,09
4	(Bhimte & Thool, 2018)	GLCM SVM	<i>Bacterial Blight, Alternaria Leaf Spot, Gray Mildew, Magnesium Deficiency, Healthy</i>	98,46
5	(Patil & Burkpalli, 2022)	GLCM, LBP, Color, SVM, MLP, Random Forest, Logistic Model Tree	<i>Alternating Blight, Bacterial Bight, Grey Mildew, Healhty</i>	91
6	Penelitian Ini	GLCM, Color Moment, MLP	<i>Aphids, Army Worm, Bacterial Blight, Powdery Mildew, Target Spot, Healthy</i>	90

Perbandingan hasil penelitian ini dengan penelitian terdahulu dapat dilihat pada Tabel 5. Penelitian terdahulu yang menggunakan *deep learning* mendapatkan akurasi antara 95.09%, 97% dan 99.97%. Sedangkan penelitian yang menggunakan *feature-based* mendapatkan akurasi 91% dan 98,46%. Penelitian ini mendapatkan akurasi yang paling kecil sebesar 90%. Jika dibandingkan dengan penelitian lain, penelitian ini memiliki kelebihan pada kelas yang dapat diklasifikasi. Penelitian ini tidak hanya mengklasifikasi penyakit pada daun kapas, tetapi mengklasifikasi hama yang terdapat pada daun kapas.

## SIMPULAN

Metode *Gray Level Co-occurrence Matrix* (GLCM) sebagai ekstraksi tekstur dan *Color Moment* sebagai ekstraksi fitur warna serta metode *Multilayer Perceptron* (MLP) sebagai klasifikasi bisa diterapkan dengan baik untuk melakukan klasifikasi hama dan penyakit tanaman kapas. Tingkat akurasi dalam klasifikasi pada hama dan penyakit tanaman kapas pada data uji menghasilkan nilai akurasi tertinggi sebesar 90%. Nilai tersebut didapatkan pengujian dengan kombinasi fitur *Color Moment* dan fitur GLCM dengan sudut 0° sebagai input serta arsitektur MLP menggunakan fungsi aktivasi *tanh* dan 1024 *perceptron* di *hidden layer*.



Pada pengembangan penelitian selanjutnya, dapat menambahkan kelas atau dataset yang digunakan, sehingga hama dan penyakit yang bisa diklasifikasi lebih beragam.

Selain itu, dapat juga menggunakan serta menggabungkan metode lain dalam proses ekstraksi fitur ataupun klasifikasi, sehingga dapat mendapatkan hasil akurasi yang lebih baik.

## DAFTAR PUSTAKA

- Abdu, A. M., Mokji, M. M. M., & Sheikh, U. U. (2020). Machine learning for plant disease detection: an investigative comparison between support vector machine and deep learning. *IAES International Journal of Artificial Intelligence (IJ-AI)*, 9(4), 670. <https://doi.org/10.11591/ijai.v9.i4.pp670-683>
- Amin, J., Anjum, M. A., Sharif, M., Kadry, S., & Kim, J. (2022). Explainable Neural Network for Classification of Cotton Leaf Diseases. *Agriculture*, 12(12), 2029. <https://doi.org/10.3390/agriculture12122029>
- Bhimte, N. R., & Thool, V. R. (2018). Diseases Detection of Cotton Leaf Spot Using Image Processing and SVM Classifier. *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, 340–344. <https://doi.org/10.1109/ICCONS.2018.8662906>
- Dewi, E. S. (2014). *Aspek Agronomi Tanaman Kapas: Budidaya Dan Pengembangan*. Dapur Buku.
- Irfan, M., Alldino, B., Sumbodo, A., & Candradewi, I. (2017). Sistem Klasifikasi Kendaraan Berbasis Pengolahan Citra Digital dengan Metode Multilayer Perceptron. *IJEIS*, 7(2), 139–148.
- Kadir, A., & Susanto, A. (2013). *Teori dan Aplikasi Pengolahan Citra* (1 ed.). CV. ANDI OFFSET.
- Kant, S. A., Pavani, M., & Kumar, K. K. (2022). Cotton Leaf Diseases Detection And Prediction Using Resnet Algorithm. *Journal Of Harbin Institute Of Technology*, 54(11), 49–57.
- Kementerian Perindustrian. (2021). *Analisis Pembangunan Industri 2021 Mendorong Kinerja Industri Tekstil dan Produk Tekstil* (III). Kementrian Perindustrian.
- Kusanti, J., & Haris, N. A. (2018). Klasifikasi Penyakit Daun Padi Berdasarkan Hasil Ekstraksi Fitur GLCM Interval 4 Sudut. *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, 03(01), 1–6.
- Lusiana, V., Amin, I. Al, Hartono, B., & Kristianto, T. (2019). Ekstraksi Fitur Tekstur Menggunakan Matriks Glcm Pada Citra Dengan Variasi Arah Obyek. *Proceeding SENDI\_U*, 0(0). <https://unisbank.ac.id/ojs/index.php/sendu/article/view/7398>
- Nisa, I. Z., Endah, S. N., Sasongko, P. S., Kusumaningrum, R., Khadijah, & Rismiyati. (2022). Klasifikasi Citra Sampah Menggunakan Support Vector Machine dengan Ekstraksi Fitur Gray Level Co-Occurrence Matrix dan Color Moments. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 9(5), 921–930. <https://doi.org/10.25126/jtiik.2022954868>
- Olivya, M., Tungadi, E., Bua, N., & Rante. (2018). Klasifikasi Kualitas Biji Kopi Ekspor Menggunakan Jaringan Saraf Tiruan Backpropagation. *Jurnal INSTEK (Informatika Sains dan Teknologi)*, 3(2), 299–308.
- Öztürk, Ş., & Akdemir, B. (2018). Application of Feature Extraction and Classification Methods for Histopathological Image using GLCM, LBP, LBGLCM, GLRLM and SFTA. *Procedia Computer Science*, 132, 40–46. <https://doi.org/10.1016/j.procs.2018.05.057>
- Patil, B. M., & Burkpalli, V. (2022). Cotton Leaf Disease Classification by Combining Color and Texture Feature-based Approach. *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 1783–1789. <https://doi.org/10.1109/ICICCS53718.2022.9788405>
- Prabowo, D. P., Sulistiyawati, P., & Pramunendar, R. A. (2022). Pengenalan Citra Batik Menggunakan Fitur Fraktal Berdasarkan Metode Support Vector Machine (SVM). *Jurnal Informatika Upgris*, 8(2), 61–65.
- Priyanka, & Kumar, D. (2020). Feature Extraction and Selection of kidney Ultrasound Images Using GLCM and PCA. *Procedia Computer Science*, 167, 1722–1731. <https://doi.org/10.1016/j.procs.2020.03.382>
- Putra, J. W. G. (2020). *Pengenalan Pembelajaran Mesin dan Deep Learning*.
- Sinaga, D. (2020). Jaringan Saraf Tiruan Infeksi Mata Dengan Menggunakan Metode Berarsitektur Multi Layer Perceptron.



- Jurnal Majalah Ilmiah Informasi dan Teknologi Ilmiah (INTI)*, 7(2), 189–192.
- Susa, J. A. B., Nombrefia, W. C., Abustan, A. S., Macalisang, J., & Maaliw, R. R. (2022). Deep Learning Technique Detection for Cotton and Leaf Classification Using the YOLO Algorithm. *2022 International Conference on Smart Information Systems and Technologies (SIST)*, 1–6. <https://doi.org/10.1109/SIST54437.2022.9945757>
- Wagle, S. A., & R, H. (2021). Comparison of Plant Leaf Classification Using Modified AlexNet and Support Vector Machine. *Traitement du Signal*, 38(1), 79–87. <https://doi.org/10.18280/ts.380108>
- Widodo, R., Widodo, A. W., & Supriyanto, A. (2018). Pemanfaatan Ciri Gray Level Co-Occurrence Matrix (GLCM) Citra Buah Jeruk Keprok (*Citrus reticulata* Blanco) untuk Klasifikasi Mutu. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(11), 5769–5776. <http://j-ptiik.ub.ac.id>