

DETEKSI PENYAKIT HAWAR DAUN JAGUNG MENGUNAKAN EKSTRASI FITUR GLCM DAN METODE ARTIFICIAL NEURAL NETWORK BACKPROPAGATION

(CORN LEAF BLIGHT DETECTION USING GLCM FEATURE EXTRACTION AND ARTIFICIAL NEURAL NETWORK BACKPROPAGATION METHOD)

Permadi Hidayat*, Fitri Bimantoro*, I Gede Pasek Suta Wijaya
Jurusan Teknik Informatika, Fakultas Informatika, Universitas Mataram
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA
Email: madihdyt17@gmail.com, bimo@unram.ac.id, gpsutawijaya@unram.ac.id

**Penulis korespondensi*

Jagung merupakan salah satu jenis makanan pokok yang terkenal di dunia, karena merupakan salah satu tanaman pangan penting, selain gandum dan padi. Hal ini harus ditingkatkan seiring dengan kebutuhan jagung dengan memperhatikan penyakit yang dapat menyebabkan gagal panen. Penelitian ini bertujuan menentukan model pembelajaran yang optimal untuk mendeteksi penyakit hawar daun jagung yakni dengan menggunakan metode GLCM untuk ekstraksi ciri dan klasifikasi menggunakan metode Artificial Neural Network Backpropagation. Pengujian dilakukan dengan menyiapkan terlebih dahulu dataset berupa foto penyakit hawar daun jagung dan kondisi normal daun jagung, setelah itu dilakukan preprocessing berupa resize, ekstraksi fitur dengan menggunakan GLCM dengan ciri energy, contrast, correlation, dan homogeneity. Melakukan pelatihan ANN-BP pada hasil ekstraksi fitur dilanjutkan dengan pengujian ANN-BP yang belum digunakan pada proses pelatihan, analisa hasil pengujian model ANN-BP dengan hasil dari Confusion Matrix seperti akurasi, presisi, dan recall. Harapan dari penelitian ini adalah diharapkan dapat tercipta metode deteksi penyakit daun hawar yang optimal, efektif dan tepat, yang menunjukkan bahwa metode tersebut dapat memprediksi dengan tepat apakah sebuah citra daun jagung normal atau terkena penyakit. Eksperimen dilakukan dengan menggunakan 2 jenis neuron input untuk daun sehat dan sakit, batas error sebesar 0.001, terdapat 1 hidden layer, learning rate sebesar 0.3, epoch 400, dan pembagian dataset sebesar 80:20. Dalam model ini, terjadi peningkatan akurasi tertinggi dengan penambahan 3720 citra, mencapai 99.5% akurasi, 100% untuk precision, dan 100% untuk recall.

Key Words: Deteksi penyakit hawar daun jagung, GLCM, ANN-BP.

I. PENDAHULUAN

Jagung merupakan salah satu jenis makanan pokok yang terkenal di dunia, karena merupakan salah satu tanaman pangan penting, selain gandum dan padi. Di Amerika Tengah dan Selatan, jagung menjadi sumber karbohidrat utama, serta menjadi alternatif pangan di Amerika Serikat. Di beberapa daerah di Indonesia, seperti Bali dan Nusa Tenggara, jagung juga digunakan sebagai pangan pokok, terutama beberapa varietas lokal yang masih dipertahankan. Selain sebagai sumber karbohidrat, jagung juga ditanam sebagai pakan ternak, serta diambil minyak dan dibuat tepung meizena dari bulirnya[1].

Di Indonesia, jagung menjadi salah satu bahan makanan pokok yang penting. Menurut data Kementerian Pertanian, produksi jagung pada tahun 2017 seberat 28,9 juta ton, dengan luas lahan panen sebesar 5,5 juta hektar. Hal ini menunjukkan bahwa produktivitas jagung nasional pada tahun tersebut sebesar 52,27 kuintal/hektar dan akan meningkat setiap tahunnya seiring kebutuhan dan pertumbuhan penduduk[2]. Walaupun banyak petani yang memperoleh hasil panen yang memuaskan, namun ada juga petani yang mengalami kendala, terutama dalam menangani penyakit tanaman jagung. Hal ini sering terjadi karena petani tidak memperhatikan penanganan yang benar, sehingga mengakibatkan gagal panen[3].

Berdasarkan Jurnal "Epidemiologi Penyakit Daun Disebabkan Jamur pada Tanaman Jagung di Kecamatan Indralaya Utara Kabupaten Ogan Ilir" tahun 2020 Dampak dari penyakit tanaman jagung yang merupakan kendala utama dalam produksi dapat menyebabkan peningkatan atau konsumsi jagung. Penyakit bulai atau daun merupakan salah satu penyakit utama dari tanaman jagung (*Peronosclerospora sp.*), hawar upih yakni (*Rhizoctonia sp.*), penyakit karat (*Puccinia sp.*), bercak daun (*Drechslera/Helminthosporium sp.*), busuk tongkol/batang (*Fusarium sp.*), pembusukan pada biji jagung (*Aspergillus sp.*). Contoh seperti penyakit hawar daun (*Bipolaris maydis*) yang menyerang pada bagian

daun dan dapat mempengaruhi hasil produksi, jamur *Exserohilum turcicum* yang menyebabkan kehilangan hasil mencapai 70%, serta penyakit bercak daun yang di sebabkan oleh patogen (*Curvularia sp*) yang merupakan penyakit penting terhadap tanaman jagung yang dapat menurunkan hasil panen. Untuk mengendalikan hama dan penyakit jagung tersebut, direkomendasikan menggunakan komponen pengendalian yang meliputi: varietas tahan, kultur teknis, musuh alami dan pertisida[4]. Oleh karena itu, diperlukan metode yang dapat mendeteksi adanya penyakit hawar daun jagung dengan tingkat akurasi yang tinggi. Salah satu metode yang dapat digunakan adalah dengan menggunakan ekstraksi fitur *Gray Level Co-occurrence Matrix* (GLCM) dan metode *Artificial Neural Network* (ANN) *backpropagation*. Ekstraksi fitur GLCM dapat digunakan untuk mengekstrak informasi tentang tekstur daun jagung yang terkena penyakit, sedangkan ANN *backpropagation* dapat digunakan untuk memprediksi adanya penyakit pada daun jagung tersebut[5].

Gray Level Co-occurrence Matrix (GLCM) merupakan salah satu metode *feature extraction* yang digunakan untuk mengukur *texture* dari sebuah citra. GLCM mengukur kemiripan antara dua piksel yang berdekatan dengan menghitung kejadian-kejadian tertentu dari piksel-piksel tersebut. GLCM menghitung kejadian-kejadian ini dengan menghitung jumlah kemunculan setiap pasangan piksel yang memiliki tingkat keabuan yang sama di dalam citra. Setelah kejadian-kejadian tersebut dihitung, GLCM menggunakannya untuk menghitung berbagai statistik yang menggambarkan *texture* citra, seperti homogenitas, kontras, dan entropi. Statistik ini kemudian dapat digunakan untuk mengklasifikasikan citra menurut ciri-ciri yang dimilikinya. *Artificial Neural Network* (ANN) adalah salah satu model jaringan saraf tiruan yang sering digunakan dalam pembelajaran mesin. ANN terdiri dari sejumlah node atau unit yang disebut "*neuron*" yang terhubung satu sama lain dengan menggunakan bobot. ANN menggunakan teknik pembelajaran yang disebut "*propagasi balik*" (*backpropagation*) untuk mengoptimalkan bobot dan biasanya agar dapat menghasilkan output yang diinginkan. ANN sering digunakan untuk menyelesaikan masalah yang berkaitan dengan identifikasi, prediksi, pengenalan pola, dan sebagainya. ANN memiliki dasar matematis yang kuat dan obyektif, serta memiliki bentuk persamaan dan nilai koefisien yang dapat digunakan untuk meminimalisasi jumlah kuadrat *error* melalui data latihan (*training set*)[6].

II. TINJAUAN PUSTAKA

2.1. Hawar daun jagung

Pada penelitian yang menggunakan jaringan syaraf tiruan dilakukan untuk melakukan identifikasi kerusakan daun tanaman apel menggunakan fitur GLCM dan JST . Pada penelitian tersebut menggunakan algoritma *Gray Level Co-occurrence Matrix* (GLCM) untuk nilai input. *Artificial Neural Network*/ Jaringan syaraf tiruan (JST) digunakan sebagai algoritma klasifikasi. Dari hasil penelitian tersebut diperoleh hasil terbaik yaitu untuk

akurasi, 74,75%[7]. Pada penelitian lain yang dilakukan oleh untuk melakukan identifikasi kematangan buah mentimun menggunakan GLCM dan JST. Hasil yang diperoleh dari penelitian tersebut sebesar 89,6% akurasi[8]. Penelitian metode serupa dengan ekstraksi ciri untuk tekstur yaitu pada penelitian tersebut *Gray Level Co-occurrence Matrix* (GLCM) digunakan untuk memperoleh ekstraksi ciri dari klasifikasi mutu pepaya lalu digunakan pada ANN. Pada penelitian tersebut hasil akurasi tertinggi dari daun tanaman buah belimbing sebesar 86.11%[9]. Pada referensi tersebut diperoleh hasil yang baik pada ekstraksi ciri dan klasifikasinya.

Salah satu penyakit tanaman yang menjadi penyakit utama daun jagung yakni penyakit hawar daun merupakan penyakit mengakibatkan hingga 70% kehilangan hasil panen jagung. Penyebab dari penyakit hawar daun adalah jamur *Exserohilum turcicum*. Penyakit ini dapat menyerang tanaman jagung selama periode pertumbuhan, terutama saat masa pertumbuhan vegetatif. Gejala dari penyakit ini dapat dilihat dari munculnya bercak kuning pada daun, yang kemudian akan menjadi coklat dan rontok. Untuk mengendalikan penyakit hawar daun, ada beberapa hal yang dapat dilakukan, seperti menanam varietas jagung yang tahan terhadap jamur *Exserohilum turcicum*, mengendalikan populasi hama yang dapat menyebarkan jamur, dan menggunakan pupuk dan pestisida yang sesuai[10].

2.2. Kecerdasan Buatan

Awalnya, komputer hanya mampu mengolah data, namun seiring perkembangan teknologi, saat ini komputer dapat mengolah pengetahuan melalui teknik kecerdasan buatan (*artificial intelligence technique*). Dengan teknik ini, proses pengambilan keputusan menjadi lebih cepat dan akurat. Manusia menggunakan pendekatan ini untuk membuat komputer dapat berpikir seperti cara manusia dalam memecahkan masalah[11].

Untuk membuat mesin yang cerdas, perlu memberikan bekal pengetahuan sehingga memiliki kemampuan menalar. Membuat aplikasi kecerdasan buatan membutuhkan dua bagian utama yakni :

1. Basis pengetahuan (*knowledge base*) merupakan bagian utama dari aplikasi kecerdasan buatan. Basis pengetahuan ini berisi fakta-fakta, teori, pemikiran, serta hubungan antar satu dengan yang lainnya. Dengan demikian, mesin dapat memiliki pengetahuan yang diperlukan untuk melakukan inferensi atau menarik kesimpulan.
2. Motor inferensi (*inference engine*) merupakan bagian utama lain dari aplikasi kecerdasan buatan. Motor ini memiliki kemampuan untuk menarik kesimpulan berdasarkan pengetahuan dan pengalaman yang tersimpan dalam basis pengetahuan. Dengan demikian, mesin dapat membuat keputusan yang akurat dan cepat.

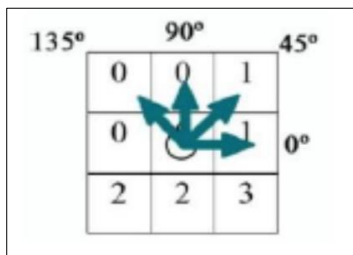
2.3. Pengenalan Pola

Pengenalan pola adalah suatu proses yang digunakan untuk mengenali pola atau sifat-sifat dari suatu

objek atau sistem. Ini dapat dilakukan dengan menggunakan serangkaian proses yang dilakukan secara serial, mulai dari proses deteksi atau segmentasi hingga proses pengenalan. Proses deteksi merupakan tahap pertama dalam pengenalan pola, di mana objek atau sistem yang akan dikenali diidentifikasi dan diklasifikasikan. Setelah itu, objek tersebut akan dianalisis melalui proses ekstraksi fitur, yaitu proses untuk mengekstrak ciri-ciri atau fitur penting dari objek tersebut. Kemudian, fitur tersebut akan digunakan untuk mengukur kemiripan dengan fitur yang telah disimpan dalam database, sehingga objek tersebut dapat dikenali atau tidak. Contoh aplikasi dari pengenalan pola adalah pengenalan wajah menggunakan data wajah ORL (*Olivetti Research Laboratory*)[12].

2.4. Gray Level Co-Occurrence Matrix (GLCM)

Gray Level Co-occurrence Matrix (GLCM) adalah metode ekstraksi fitur dalam pemrosesan citra. Metode ini mencirikan tekstur pada citra dengan cara menghitung frekuensi munculnya pasangan dua piksel dengan intensitas derajat keabuan tertentu dalam jarak (d) yang dinyatakan dalam piksel dan orientasi arah dengan sudut (θ) yang dinyatakan dalam sudut 0° , 45° , 90° , dan 135° [13]. Ilustrasi GLCM dapat digambarkan seperti pada Gambar 2.2.



Gambar 2.2 Ilustrasi arah GLCM

Perhitungan jarak dan orientasi arah akan menghasilkan sebuah matriks kookurensi. Matriks kookurensi yang dihasilkan kemudian ditambahkan dengan matriks transposenya untuk menjadikannya simetris terhadap sumbu diagonal. Setiap elemen matriks lalu dinormalisasi dengan membagi tiap elemen dengan jumlah total pasangan piksel. Selanjutnya dilakukan pengukuran ciri tekstur dari citra. Di dalam penelitian ini, digunakan beberapa ekstraksi ciri pada metode GLCM yang diantaranya sebagai berikut:

1. Energy

Fitur tekstur mewakili ukuran konsentrasi pasangan intensitas dalam matriks. Semakin tinggi nilai energi, maka nilai keseragaman akan semakin tinggi yang mengindikasikan banyaknya nilai piksel yang mirip dengan piksel yang lain. Sebaliknya, ketika nilai energi kecil menandakan nilai GLCM yang *heterogeny*[13]. Untuk menghitung nilai energy, dapat menggunakan persamaan (2-2):

$$\sum_{i,j=0}^{levels-1} = P_{i,j}^2$$

2. Contrast

Contrast pada fitur GLCM menunjukkan ukuran penyebaran (momen inersia) elemen-elemen matriks citra[14]. Jika nilai contrast semakin tinggi, maka

perbedaan intensitas pada nilai piksel yang bertetangga juga tinggi[13]. Nilai contrast dapat dihitung dengan menggunakan persamaan (2-3):

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

3. Correlation

Correlation menyatakan ukuran ketergantungan linier derajat keabuan citra sehingga dapat memberikan petunjuk adanya struktur linier dalam citra[20]. Perhitungan untuk mendapatkan nilai correlation dapat menggunakan persamaan (2-4):

$$\sum_{i,j=0}^{levels-1} = P_{i,j} \left[\frac{(i-\mu_i)(j-\mu_j)}{(\sigma_i^2)(\sigma_j^2)} \right] \quad (2-3)$$

4. Homogeneity

Homogeneity mengembalikan nilai yang mengukur kedekatan distribusi elemen pada GLCM terhadap GLCM diagonal. Perhitungan homogeneity dapat menggunakan persamaan (2-5):

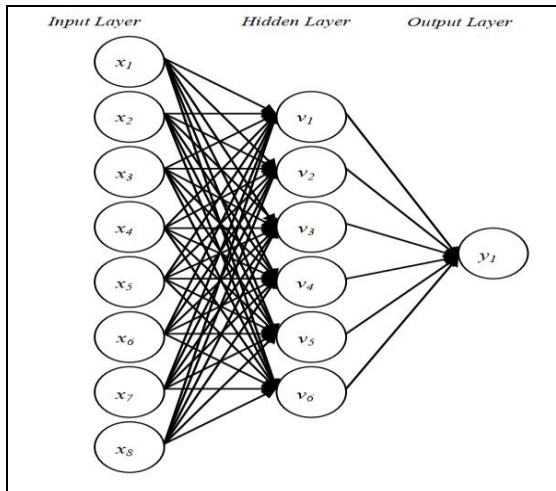
$$\sum_{i,j=0}^{levels-1} = \frac{P_{ij}}{1+(i-j)^2}$$

Ekstraksi 4 fitur GLCM di atas didesain untuk mendapatkan hasil dari deteksi hawar daun jagung yang lebih cepat serta penggunaan daya komputasi yang lebih rendah. Motivasi utama adalah pembuatan model yang ringan dengan parameter yang kecil agar waktu komputasi yang dibutuhkan menjadi lebih cepat. Penggunaan 4 fitur ini juga terbukti dapat menghasilkan akurasi yang tinggi sebesar 92.00% pada penelitian Klasifikasi Daun Empon-Empon menggunakan metode *Gray Level Co-Occurrence Matrix* dan algoritma K-NN[15].

2.5. Jaringan Saraf Tiruan *Backpropagation*

Jaringan saraf tiruan tercipta sebagai suatu generalisasi model matematis dari pemahaman manusia (*human cognition*) yang didasarkan atas asumsi sebagai berikut[16]:

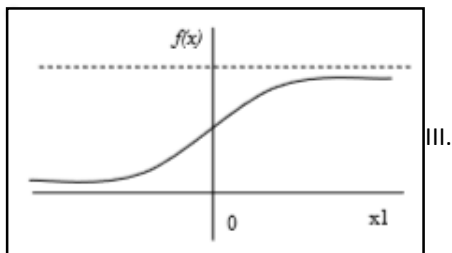
1. Pemrosesan informasi terjadi pada elemen sederhana yang disebut neuron.
2. Isyarat mengalir di antara sel saraf/neuron melalui suatu sambungan penghubung.
3. Setiap sambungan penghubung memiliki bobot yang bersesuaian.
4. Setiap sel saraf akan merupakan fungsi aktivasi terhadap isyarat hasil penjumlahan berbobot yang masuk kepadanya untuk menentukan isyarat keluarannya.



Gambar 2. 3 Arsitektur Jaringan Saraf Tiruan Backpropagation

Jaringan Saraf Tiruan *Backpropagation* memiliki beberapa *neuron* yang ada dalam satu atau lebih lapis tersembunyi. Gambar 2.3 [9] diatas adalah arsitektur *backpropagation* dengan n buah masukan ($X_1, X_2, X_3, \dots, X_n$) ditambah sebuah bias, sebuah lapis tersembunyi yang terdiri dari j unit ditambah sebuah bias, serta k buah unit keluaran[17].

Pada Jaringan Saraf Tiruan *backpropagation*, fungsi aktivasi yang dipakai harus memenuhi beberapa syarat yaitu : kontinu, terdiferensial dengan mudah dan merupakan fungsi yang tidak turun. Salah satu fungsi yang memenuhi ketiga syarat tersebut sehingga sering dipakai adalah fungsi *sigmoid* biner yang memiliki range (0,1). Diberikan $F(x) = \frac{1}{1+e^{-x}}$ dengan turunan $f'(x) = f(x)(1 - f(x))$. Grafik fungsinya tampak pada Gambar 2.4.



Gambar 2. 4 Fungsi Aktivasi Sigmoid Biner[18]

Fungsi lain yang sering dipakai adalah fungsi *sigmoid bipolar* yang bentuk fungsinya mirip dengan fungsi *sigmoid* biner, tapi dengan *range* (-1,1). Diberikan $F(x) = \frac{2}{1+e^{-x}} - 1$ dengan turunan $f'(x) = \frac{(1+f(x))(1-f(x))}{2}$. Grafik fungsinya tampak pada Gambar 2.5.

2.6. Confusion Matrix

Confusion Matrix digunakan dalam pengelompokan data klasifikasi ke dalam empat bagian dan akan digunakan untuk menghitung besar akurasi pengujian[19][19]. Tabel *confusion Matrix* dapat dilihat dalam Tabel 2.1

Tabel 2. 1 Confusion Matrix

		Nilai Sebenarnya	
		Benar	Salah
Prediksi	Benar	<i>True Positive</i>	<i>False Positive</i>
	Salah	<i>False Negative</i>	<i>True Negative</i>

True Positives (TP) menunjukkan prediksi bernilai benar sesuai dengan kondisi yang sesungguhnya bernilai benar. *False Positives* (FP) menunjukkan prediksi bernilai benar di mana pada kondisi yang sesungguhnya bernilai salah. *True Negatives* (TN) menunjukkan prediksi bernilai salah sesuai dengan kondisi yang sesungguhnya bernilai salah. *False Negatives* (FN) menunjukkan prediksi bernilai salah di mana pada kondisi yang sesungguhnya bernilai benar. Dari hasil *Confusion Matrix* maka didapatkan nilai akurasi, presisi, dan *recall*[20].

Akurasi merupakan rasio prediksi benar (positif dan negatif) dengan keseluruhan data. Adapun perhitungan akurasi dapat dilihat dalam persamaan (2-17).

$$akurasi = \frac{TP+TN}{TP+FN+FP+TN} \quad (2-16)$$

Presisi merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Adapun perhitungan presisi dapat dilihat dalam persamaan (2-18).

$$presisi = \frac{TP}{TP+FP} \quad (2-17)$$

Recall merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Adapun perhitungan *recall* dapat dilihat dalam persamaan (2-19)[18][21].

$$recall = \frac{TP}{TP+FN} \quad (2-18)$$

III. METODE PENELITIAN

3.1. Alat dan Bahan Penelitian

Alat dan bahan yang diperlukan dalam penelitian berupa *hardware*, *software*, dan dataset berupa foto daun jagung normal dan terkena hawar daun berformat .jpg.

1. Alat Penelitian

Alat-alat yang digunakan dalam penelitian ini sebagai berikut:

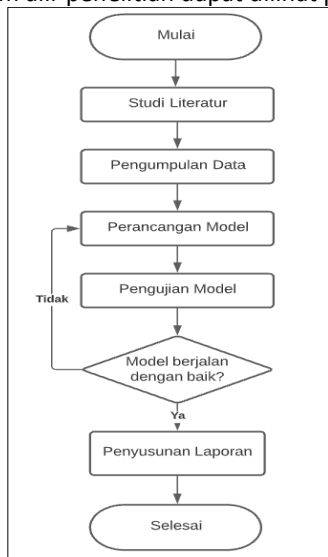
 - a. Laptop Asus X550I Quadcore AMD Bristol Bridge FX 9830P Gen7 CPU - 3GHz (4 Core), dengan AMD Radeon R7 900 Mhz, RAM 8GB.
 - b. Sistem Operasi Windows 10 Home, Oracle VM VirtualBox 7.0.8, Ubuntu 22.04 LTS
 - c. Jupyter Lab dan Google Colab
2. Bahan Penelitian

Bahan penelitian yang digunakan dalam penelitian ini yaitu dataset hawar daun jagung berformat file .jpg yakni yang bersumber dari website *kaggle.com*, jumlah kelas yang digunakan ada 2 yakni sehat dan berpenyakit hawar pada daun jagung, dengan ukuran citra asli yaitu 256x256

pixel di *resize* menjadi 128x128 pixel yang akan dilakukan penghapusan *background* dan rotasi pada sejumlah 3000 citra.

3.2. Diagram Alir Penelitian

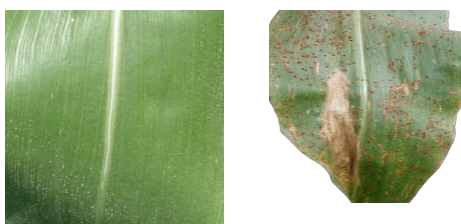
Dalam pelaksanaan penelitian ini terdapat beberapa tahapan. Tahapan pertama melakukan studi literatur. Studi literature yang dilakukan yaitu membaca jurnal, buku, dan penelitian pengenalan pola terkait tulisan tangan aksara. Selanjutnya dilakukan pengumpulan dataset. Dataset tersebut akan digunakan sebagai data latih dan data uji. Setelah itu dilakukan pembuatan model *Backpropagation* untuk hawar daun jagung. Setelah model selesai dibuat dilanjutkan dengan melakukan pengujian skenario menggunakan GLCM, setelah berhasil dilakukan pengujian skenario maka akan didapatkan nilai akurasi dari hawar daun jagung menggunakan GLCM dan model *Backpropagation*. Diagram alir penelitian dapat dilihat pada Gambar 3.1 .



Gambar 3. 1 Diagram Alir Penelitian

3.3 Pengumpulan Dataset

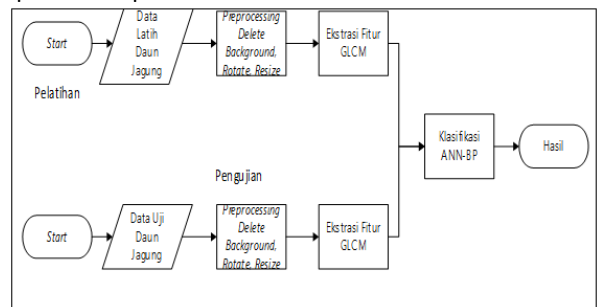
Bahan penelitian yang digunakan dalam penelitian ini merupakan dataset hawar daun jagung yang bersumber dari website *kaggle.com* yakni "*PlantifyDr Dataset*" dan "*Corn Leaf Disease Detection with ResNet - PyTorch*" dengan link download <https://www.kaggle.com/datasets/vipooooool/new-plant-diseases-dataset> dan <https://www.kaggle.com/code/emrearslan123/corn-leaf-disease-detection-with-resnet-pytorch/data>, terdapat *folder* didalam dataset tersebut bernama *Blight* yang artinya hawar, sehingga diperoleh data sebesar 3000 citra. Berikut ini merupakan dataset daun sehat dan berpenyakit pada Gambar 3.2.



Gambar 3.2 Daun sehat (kiri), Daun berpenyakit (kanan)

3.4. Pembuatan Model

Terdapat beberapa tahapan dalam pembangunan model menggunakan ekstraksi fitur GLCM dan klasifikasi *Backpropagation*. Diagram alir pembangunan model dapat dilihat pada Gambar 3.2.



Gambar 3. 3 Diagram Alir Pembangunan Model

3.5. Data Pre-Processing

Pada data *pre-processing* terdapat Penghapusan *background*, rotasi dan *resize* pada dataset proses ini berfungsi untuk melakukan kesetaraan atau perlakuan yang sama terhadap semua dataset yang telah dikumpulkan, dengan urutan yakni menghapus *background* hitam belakang pada daun hawar yang berpenyakit dikarena yang sehat sudah tidak ada *background* belakang. Rotasi pada daun jagung ke arah yang sama, dan selanjutnya *resize* dari 256x256 pixel menjadi 128x128 pixel agar dataset menjadi tampak zoom dan mengurangi *background* yang terlihat dan mendekati dataset saat diambil secara langsung.

3.6. Ekstraksi Fitur GLCM

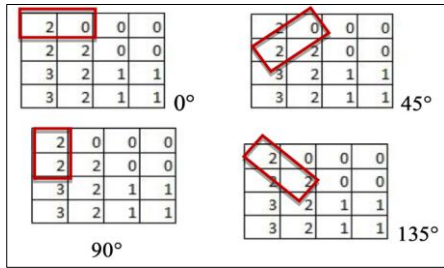
Proses ekstraksi fitur yang diimplementasikan pada penelitian ini adalah dengan memanfaatkan metode GLCM. Citra yang telah melalui tahap *preprocessing* pada proses sebelumnya akan menjadi *input* pada proses ekstraksi fitur. Terdapat 2 tahapan dalam proses ini yaitu pembentukan matriks GLCM dan perhitungan nilai fitur GLCM.

1. Pembentukan matriks kookurensi, dimana matriks ini merupakan matriks yang menunjukkan jumlah pasangan piksel yang bertetangga pada baris ke-*i* dan kolom ke-*j* dengan orientasi sudut dan jarak tertentu. Misalkan terdapat sebuah matriks dari citra *grayscale* yang direpresentasikan oleh matriks A pada Gambar 3.3.

$$A = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 3 & 2 & 1 & 1 \\ 3 & 2 & 1 & 1 \end{bmatrix}$$

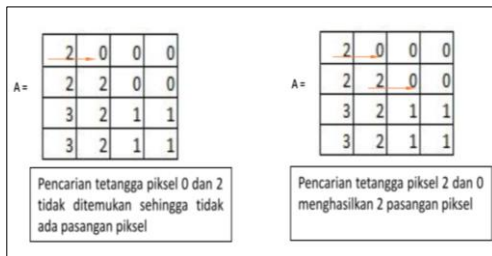
Gambar 3.4 Contoh Matriks A

Dari matriks A kemudian dibentuk matriks GLCM dengan orientasi sudut 0° , 45° , 90° , dan 135° pada jarak (*d*) sama dengan 1 seperti pada Gambar 3.4.



Gambar 3.5 orientasi sudut pada matriks A

Berikut adalah contoh proses pembuatan matriks kookurensi dengan derajat 0 dan jarak 1 yang dapat dilihat pada Gambar 3.15.



Gambar 3.6 Proses pembentukan matriks kookurensi

Hasil dari perhitungan jumlah ketetanggaan tersebut dimasukkan ke dalam sebuah tabel kookurensi seperti pada Gambar 3.6.

	0	1	2	3
0	3	0	0	0
1	0	2	0	0
2	2	2	1	0
3	0	0	2	0

Keterangan:
 = Baris
 = Kolom

Gambar 3.7 Matriks kookurensi

Setelah didapatkan matriks kookurensi, dilakukan normalisasi sehingga matriks kookurensi berisi probabilitas atau kemungkinan munculnya kookurensi antar piksel. Normalisasi dilakukan dengan membagi nilai jumlah kookurensi tiap hubungan ketetanggaan terhadap total jumlah kookurensi seperti pada Gambar 3.7.

	0	1	2	3
0	0.250	0	0	0
1	0	0.167	0	0
2	0.167	0.167	0.083	0
3	0	0	0.167	0

Keterangan:
 = Baris
 = Kolom

Gambar 3.8 Matriks kookurensi setelah normalisasi

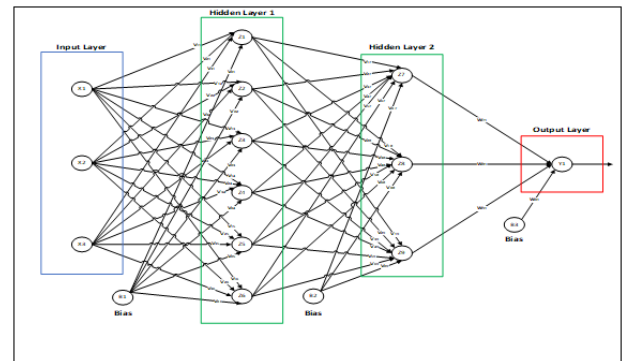
3.7. Klasifikasi *Backpropagation*

Proses klasifikasi yang digunakan pada penelitian ini yaitu menggunakan metode *backpropagation*. *Output* dari pengklasifikasian dari metode ini yakni model pembelajaran pada kasus berupa hawar daun jagung. Sedangkan *input* dari proses pengklasifikasian menggunakan nilai *output* yang didapatkan dari proses ekstraksi fitur menggunakan metode GLCM. Berikut contoh perhitungan metode *backpropagation* menggunakan data pada tabel 3.1

Tabel 3. 1 Contoh Data Eksraksi Fitur

No	X1	X2	X3	Target
1	0	1	0	1
2	1	1	0	0
3	1	0	0	0
4	0	0	1	1

Model arsitektur yang digunakan pada data di atas menggunakan model arsitektur pada Gambar 3.5



Gambar 3. 8 Arsitektur Backpropagation

Backpropagation di atas terdiri dari *input layer*, 2 *hidden layer*, dan *output layer*. Di dalam *input layer* terdapat 3 *neuron* sebagai *input*, 6 *neuron* pada *hidden layer* pertama, 3 *neuron* 1 *neuron* pada *hidden layer* kedua, dan 1 *neuron* pada *output layer*. Contoh pelatihan yang digunakan memiliki beberapa beberapa ketentuan:

1. Batas *error* = 0.0001
2. Batas *epoch* = 1000
3. *Learning rate* = 0.1
4. Fungsi aktivasi menggunakan *sigmoid biner*

Terdapat 3 langkah utama dalam menyelesaikan model *backpropagation* yaitu langkah propagasi maju, propagasi mundur, dan perubahan bobot. Sebelum melakukan tahapan propagasi mundur, propagasi maju, dan perubahan bobot, perlu dilakukan inialisasi nilai bobot pada masing-masing *layer*.

Tabel 3. 2 Nilai Bias dan bobot dari input layer ke hidden layer pertama

DARI/KE	BIAS (B1)	x1	x2	x3
Z1	0,22	0,023	0,045	0,067
Z2	0,112	0,241	0,208	0,211
Z3	0,27	0,034	0,023	0,136
Z4	0,212	0,111	0,12	0,149
Z5	0,102	0,101	0,121	0,512
Z6	0,144	0,125	0,064	0,056

Tabel 3.2 di atas merupakan nilai bias dan bobot dari *input layer* ke *hidden layer* pertama menggunakan nilai *random*. Nilai ini nantinya akan digunakan untuk mencari nilai fungsi aktivasi dan sigma pada *hidden layer* pertama.

Tabel 3. 3 Nilai Bias dan bobot dari hidden layer pertama ke hidden layer kedua

DARI/KE	Bias (B2)	Z1	Z2	Z3	Z4
Z7	0,032	0,162	0,165	0,075	0,054
Z8	0,005	0,016	0,147	0,064	0,081
Z9	0,072	0,026	0,042	0,098	0,069

Tabel 3.3 di atas merupakan nilai bias dan bobot dari *hidden layer* pertama ke *hidden layer* kedua menggunakan nilai *random*. Nilai ini nantinya akan digunakan untuk mencari nilai fungsi aktivasi dan sigma pada *hidden layer* kedua.

Tabel 3. 4 Nilai Bias dan bobot dari hidden layer kedua ke output layer

DARI/KE	Bias (B3)	Z7	Z8	Z9
Y1	0,15	0,026	0,076	0,122

Tabel 3.3 di atas merupakan nilai bias dan bobot dari *hidden layer* kedua ke *output layer* menggunakan nilai *random*. Nilai ini nantinya akan digunakan untuk mencari nilai fungsi aktivasi dan sigma pada *output layer* kedua yang nantinya digunakan sebagai acuan untuk melakukan propagasi mundur.

3.8. Teknik dan Skenario Pengujian

Teknik pengujian sistem yang dilakukan adalah dengan menghitung nilai akurasi, presisi, dan recall. Nilai tersebut didapatkan dari *confusion matrix*. Perhitungan nilai akurasi, presisi, dan recall dilakukan untuk menguji kelayakan dari sistem yang dibuat karena menghitung prediksi positif dari keseluruhan data, prediksi positif dari keseluruhan data yang diprediksi benar, dan prediksi positif dari keseluruhan data yang benar positif. Persamaan yang digunakan dalam menghitung akurasi, presisi, dan recall terdapat pada persamaan (2-17), (2-18), (2-19). Berikut contoh perhitungannya:

Tabel 3. 5 confusion matrix 3x3

		Prediksi kelas		
		A	B	C
Kelas aktual	A	35 (AA)	20 (AB)	10 (AC)
	B	40 (BA)	55 (BB)	15 (BC)
	C	10 (CA)	10 (CB)	70 (CC)

Sehingga nilai akurasi dapat dihitung dengan membandingkan nilai prediksi yang bernilai benar pada seluruh kelas dengan keseluruhan jumlah data menggunakan persamaan:

$$akurasi = \frac{TP+TN}{TP+FN+FP+TN} \quad (2-17)$$

$$akurasi = \frac{AA + BB + CC}{AA + AB + AC + BA + BB + BC + CA + CB + CC}$$

$$akurasi = \frac{35 + 55 + 70}{35 + 20 + 10 + 40 + 55 + 15 + 10 + 10 + 70}$$

$$akurasi = \frac{160}{265}$$

$$akurasi = 0.6$$

Untuk nilai presisi pada kelas A dapat dihitung dengan membandingkan nilai prediksi yang bernilai benar terhadap kelas A dengan keseluruhan yang diprediksi sebagai kelas A, dan begitu pula untuk kelas yang lainnya menggunakan persamaan:

$$presisi = \frac{TP}{TP+FP} \quad (2-18)$$

$$presisi A = \frac{AA}{AA + BA + CA}$$

$$presisi A = \frac{35}{35 + 40 + 10}$$

$$presisi A = \frac{35}{85}$$

$$presisi A = 0.4$$

Untuk nilai recall pada kelas A dihitung dengan membandingkan nilai prediksi yang bernilai benar terhadap kelas A dengan keseluruhan yang secara actual merupakan kelas A, dan begitu pula untuk kelas yang lainnya menggunakan persamaan:

$$recall = \frac{TP}{TP+FN} \quad (2-19)$$

$$recall A = \frac{AA}{AA + AB + AC}$$

$$recall A = \frac{35}{35 + 20 + 10}$$

$$recall A = \frac{35}{65}$$

$$recall A = 0.5$$

Pada sistem deteksi hawar daun jagung digunakan metode *backpropagation* untuk tahap pengujianya terdapat beberapa parameter yang nantinya akan digunakan dalam ekstraksi fitur dan klasifikasi yaitu:

- Jumlah sudut yang digunakan adalah 4 sudut (0°, 45°, 90°, 135°)
- Jumlah kelas : 2 kelas, berpenyakit dan sehat
- Jumlah *hidden layer* : 1 *Hidden layer*
- *Learning rate* sebagai parameter uji : 0.1, 0.2, 0.3
- Batas epoch sebagai parameter uji : 100, 200, 300, 400, 500
- Batas *error* sebagai parameter uji : 0.001
- Neuron *hidden layer*: 16
- Fungsi aktivasi *sigmoid biner* untuk *output layer*.

- Pembagian dataset : 70:30, 80:20, 90:10
Jadwal kegiatan deteksi penyakit hawar pada daun jagung seperti pada Tabel 3.6 berikut.

Tabel 3.6 Jadwal kegiatan perancangan

No.	Kegiatan	Waktu (Bulan)				Keterangan
		I	II	III	IV	
1	Analisa					Analisa kebutuhan
2	Perancangan					Perancangan model
3	Coding					Pengkodean model
4	Testing					Pengujian model
5	Dokumentasi					Dokumentasi model

IV. HASIL DAN PEMBAHASAN

4.1. Pengaruh Variasi Pembagian Dataset

Dalam penelitian ini, dilakukan pembagian dataset dengan perbandingan 70:30, 80:20, dan 90:10 untuk melihat berapa nilai akurasi terbaik yang dapat dicapai pada 2 jenis input dan 1 *hidden layer* dengan jumlah input-an tertentu. Tabel 4.1 menunjukkan hasil terbaik yang dicapai dari pembagian dataset tersebut. Pembagian dataset ini mempengaruhi hasil akurasi yang didapatkan dalam penelitian ini.

Tabel 4.1 Akurasi terbaik variasi pembagian dataset

		Dataset		
		70:30	80:20	90:10
Test	Akurasi	0.992	0.994	0.993
	Precision	1.0	1.0	1.0
	Recall	1.0	1.0	1.0
Train	Akurasi	0.992	0.994	0.993
	Precision	1.0	1.0	1.0
	Recall	1.0	1.0	1.0
	Error	0.00096	0.00061	0.00065

Dari Tabel 4.1 tersebut disimpulkan hasil tertinggi dari percobaan dengan variasi pembagian dataset, pembagian 80:20 menghasilkan performa terbaik dengan akurasi 99.4%, presisi 100%, dan *recall* 100%. Meskipun dilakukan variasi pembagian dataset, perubahan ini tidak memberikan dampak yang signifikan pada hasil akurasi.

4.2. Pengaruh Batas Epoch

Dalam penelitian ini, dilakukan pengujian pengaruh variasi batas *epoch* terhadap akurasi yang didapatkan dengan menggunakan variasi *epoch* sebesar 100, 200, 300, 400, dan 500 pada 2 jenis input jumlah input-an dan pembagian dataset yakni sebesar 80:20. Hasil akurasi dari pengujian tersebut disajikan pada Tabel 4.2.

Tabel 4.2 Akurasi terbaik variasi batas epoch

		Epoch				
		100	200	300	400	500
Test	Akurasi	0.993	0.9937	0.991	0.994	0.993
	Precision	1.0	1.0	1.0	1.0	1.0
	Recall	1.0	1.0	1.0	1.0	1.0
Train	Akurasi	0.993	0.993	0.990	0.994	0.993
	Precision	1.0	1.0	1.0	1.0	1.0
	Recall	1.0	1.0	1.0	1.0	1.0
	Error	0.00069	0.00062	0.00085	0.00061	0.00073

Pada tabel 4.2, dapat disimpulkan bahwa hasil terbaik diperoleh pada *epoch* 400, dengan tingkat akurasi sebesar 99.4%, presisi 100%, dan *recall* 100%. Peningkatan batas *epoch* yang telah ditentukan pada penelitian memberikan pengaruh yang cukup signifikan terhadap peningkatan tingkat akurasi yang diperoleh. Meskipun peningkatan batas *epoch* dapat meningkatkan akurasi, titik maksimal peningkatan akurasi tercapai pada *epoch* 400. Oleh karena itu, batas *epoch* 400 akan digunakan pada tahap selanjutnya sebagai parameter yang optimal untuk pelatihan model. Dengan menggunakan batas *epoch* 400, diharapkan model dapat mencapai tingkat akurasi yang tinggi dan stabil pada tahap selanjutnya.

4.3. Pengaruh Learning Rate

Pengujian terhadap variasi learning rate dan pengaruhnya pada penelitian ini dilakukan terhadap tingkat akurasi yang didapatkan. Variasi learning rate yang digunakan adalah 0.1, 0.2, dan 0.3 dengan batas epoch 400 dan pembagian dataset 80:20 Hasil terbaik dari pengujian tersebut disajikan pada Tabel 4.3. Pada tabel 4.3 menunjukkan hasil akurasi terbaik yang didapatkan dari variasi learning rate yang berbeda.

Tabel 4.3 Akurasi terbaik variasi learning rate

		Learning Rate			
		0.1	0.2	0.3	0.4
Test	Akurasi	0.991	0.992	0.994	0.991
	Precision	1.0	1.0	1.0	1.0
	Recall	1.0	1.0	1.0	1.0
Train	Akurasi	0.991	0.992	0.994	0.991
	Precision	1.0	1.0	1.0	1.0
	Recall	1.0	1.0	1.0	1.0
	Error	0.00094	0.00078	0.00061	0.00057

Dari Tabel 4.3, dapat disimpulkan bahwa hasil terbaik diperoleh pada *learning rate* 0.3, dengan tingkat akurasi sebesar 99.4%, presisi 100%, dan *recall* 100%. Namun, pada *learning rate* 0.4, terjadi penurunan akurasi. Oleh karena itu, penelitian hanya mencakup pembahasan hingga *learning rate* 0.3. Selanjutnya, dapat dinyatakan bahwa besar atau kecilnya nilai *learning rate* tidak memberikan pengaruh yang signifikan terhadap tingkat akurasi yang diperoleh dalam penelitian ini. Hal ini bisa mengindikasikan bahwa pada rentang nilai *learning rate* yang diteliti, perbedaan besar atau kecilnya nilai

learning rate tidak memiliki dampak yang signifikan terhadap hasil akurasi.

4.4. Validasi

Pada validasi dilakukan pengujian ulang dengan menambah sejumlah citra yang pada pemrosesan ini dengan menambahkan lebih banyak citra atau data dalam pelatihan model, dapat meningkatkan representasi atau diwakili dalam bentuk yang dapat diproses oleh model terhadap variasi data yang lebih luas, mengurangi risiko *overfitting* yakni terjadi ketika model terlalu terfokus pada data pelatihan dan tidak dapat melakukan generalisasi dengan baik pada data baru, dan secara potensial meningkatkan kinerja model dengan memberikan lebih banyak contoh untuk belajar. Ini membantu model untuk lebih baik dalam menggeneralisasi atau melakukan prediksi yang akurat. Pengujian dilakukan dengan arsitektur jaringan yang sudah ditentukan sebelumnya yakni 80:20 pembagian dataset, *epoch* 400, batas *error* 0.001, terdapat 1 *hidden layer*, 0.3 *learning rate*.

Tabel 4. 4 Hasil pengujian arsitektur jaringan

		3000 Citra	3720 Citra
Test	Akurasi	0.994	0.995
	Precision	1.0	1.0
	Recall	1.0	1.0
Train	Akurasi	0.994	0.995
	Precision	1.0	1.0
	Recall	1.0	1.0
Error		0.00061	0.00070
Waktu (s)		0.90	1.48

Dari hasil penelitian yang dilakukan pada label 4.4, ditemukan bahwa penggunaan dataset daun jagung sebanyak 3720 citra menghasilkan performa terbaik. Proses validasi menunjukkan tingkat akurasi yang cukup baik, mencapai 99.5% dari total 3720 dataset yang diuji. Berdasarkan hasil ini, dapat disimpulkan bahwa model arsitektur *Artificial Neural Network* (ANN) dengan metode *Backpropagation* mampu dengan baik mendeteksi penyakit pada daun jagung. Hasil yang baik ini dapat dicapai karena penambahan jumlah data yang signifikan dari 3000 menjadi 3720 citra. Keberagaman dan jumlah data yang lebih banyak memberikan dampak positif pada akurasi model dibandingkan dengan jumlah data yang lebih sedikit.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan analisis dan pengujian model arsitektur ANN yang telah dilakukan, berikut adalah kesimpulan yang dapat diambil:

1. Pada model arsitektur jaringan terbaik untuk deteksi penyakit pada daun jagung, dalam hal ini adalah menggunakan 1 *hidden layer*, *learning rate* sebesar 0.3, *epoch* 400, dan pembagian dataset sebesar 80:20. Model ini mencapai tingkat akurasi training sebesar 99.5%.
2. Peningkatan akurasi model arsitektur ANN *backpropagation* setelah diuji ulang dengan

penambahan jumlah dataset dari 3000 menjadi 3720 citra menghasilkan akurasi tertinggi sebesar 99.5%. Hal ini menunjukkan bahwa jumlah dataset juga memiliki pengaruh signifikan terhadap tingkat keakuratan model.

B. Saran

Berikut ini saran dari laporan penelitian ini:

1. Melakukan peningkatan kualitas citra agar diperlakukan sama pada tiap-tiap data seperti saat pengambilan dataset, pencahayaan, letak dan kualitas dataset yang seharusnya konsisten.
2. Dapat mencoba variasi model arsitektur pada pengujian *neural network*, seperti menambahkan *hidden layer* dan mengatur jumlah *neuron* dengan lebih beragam.
3. Untuk mencegah adanya *overfitting* dengan cara memperbaiki *background* putih pada dataset yang ada pada sistem, agar tidak adanya kecenderungan model yang melakukan pembelajaran mesin terlalu baik dalam mempelajari data pelatihan tertentu, tetapi tidak mampu secara umum melakukan generalisasi dengan baik untuk data baru.

REFERENCES

- [1] S. S. Manek and U. Joka, "Sistem Pendukung Keputusan Kelayakan Panen pada Tanaman Jagung," *J. Saintek Lahan Kering*, vol. 3, no. 2, pp. 38–41, 2020, [Online]. Available: <http://savana-cendana.faperta-unimor.id/index.php/SLK/article/view/1220>
- [2] R. Kurniawan Budhi, A. Prayitno, and S. Elvina, "Pengenalan Pola Daun untuk Pendeteksi Dini Penyakit Tanaman Jagung Menggunakan Deteksi Tepi Sobel," *Semin. Nas. APTIKOM*, pp. 340–346, 2019.
- [3] A. D. Nurcahyati, R. M. Akbar, and S. Zahara, "Klasifikasi Citra Penyakit pada Daun Jagung Menggunakan Deep Learning dengan Metode Convolution Neural Network (CNN)," *SUBMIT J. Ilm. Teknol. Infomasi dan Sains*, vol. 2, no. 2, pp. 43–51, 2022, doi: 10.36815/submit.v2i2.1877.
- [4] H. Hamidson, S. Suwandi, and N. Nurhayati, "Epidemiologi Penyakit Daun Disebabkan Jamur pada Tanaman Jagung di Kecamatan Indralaya Utara Kabupaten Ogan Ilir," *Pros. Semin. Nas. Lahan Suboptimal ke-8*, pp. 427–437, 2020.
- [5] I. Fathurrahman and I. Gunawan, "Pengenalan Citra Logo Kendaraan Menggunakan Metode Gray Level Co-Occurrence Matrix (GLcm) dan Jst-Backpropagation," *Infotek J. Inform. dan Teknol.*, vol. 1, no. 1, pp. 47–55, 2018, doi: 10.29408/jit.v1i1.894.
- [6] I. Fitriaty and Y. A. Mustofa, "Deteksi Penyakit Tanaman Daun Bayam Menggunakan Metode GLCM dan Artificial Neural Network (ANN)," *Cosphijournal.Unisan.Ac.Id*, vol. 3, no. 1, pp. 21–25, 2019, [Online]. Available:

- <https://www.cosphijournal.unisan.ac.id/index.php/cosphihome/article/view/83>
- [7] E. P. Suwanto, M. Ezar, and A. Rivian, "Identifikasi Kerusakan Daun Tanaman Apel Menggunakan Fitur GLCM Dan JST," *J. Algoritm.*, vol. 2, no. 1, pp. 73–81, 2021.
- [8] I. Fathurrahman, A. Muliawan Nur, and F. Fathurrahman, "Identifikasi Kematangan Buah Mentimun Berbasis Citra Digital Menggunakan Jaringan Syaraf Tiruan Backpropagation," *Infotek J. Inform. dan Teknol.*, vol. 2, no. 1, pp. 27–33, 2019, doi: 10.29408/jit.v2i1.976.
- [9] F. Wibowo and A. Harjoko, "khazanah informatika Klasifikasi Mutu Pepaya Berdasarkan Ciri Tekstur GLCM Menggunakan Jaringan Saraf Tiruan," *J. Ilmu Komput. dan Inform.*, vol. 3, no. 2, pp. 100–104, 2017.
- [10] E. H. Rachmawanto and H. P. Hadi, "Optimasi Ekstraksi Fitur Pada Knn Dalam Klasifikasi Penyakit Daun Jagung," *Dinamik*, vol. 26, no. 2, pp. 58–67, 2021, doi: 10.35315/dinamik.v26i2.8673.
- [11] M. I. Rosadi, M. Lutfi, and S. Artikel, "Identifikasi Jenis Penyakit Daun Jagung Menggunakan Deep Learning Pre-Trained Model," *J. Keilmuan Dan Apl. Tek. Inform.*, vol. 5, no. 36, pp. 35–42, 2019, [Online]. Available: <https://doi.org/10.35891/explorit>
- [12] Z. Abidin, Y. Fredyatama, P. Teknik Informasi, S. K. Tinggi Teknik Pati Jl Raya Pati-Trangkil, and P. Jawa Tengah, "Klasifikasi Daun Empon-Empo Menggunakan Metode Gray Level Co-Occurrence Matrix Dan Algoritma K-Nn," *J. Sains, Teknol. dan Ind.*, vol. 18, no. 02, pp. 261–267, 2021.
- [13] Z. A. A. Feri Agustina, "Identifikasi Citra Daging Ayam Kampung dan Broiler Menggunakan Metode GLCM dan Klasifikasi-NN," *J. Infokam*, vol. XVI, no. 1, pp. 25–36, 2020.
- [14] N. Djaenuddin, S. Suriani, and A. H. Talanca, "Kombinasi Biopestisida Formulasi Bacillus subtilis BNt8 dan Pestisida Nabati untuk Pengendalian Penyakit Hawar Daun Bipolaris maydis pada Jagung," *J. Penelit. Pertan. Tanam. Pangan*, vol. 2, no. 1, p. 43, 2018, doi: 10.21082/jpptp.v2n1.2018.p43-49.
- [15] A. Rusae, B. Metboki, and B. Atini, "Identifikasi Cendawan Patogen pada Tanaman Sorgum di Timor Tengah Utara," *Savana Cendana*, vol. 3, no. 04, pp. 69–71, 2018, doi: 10.32938/sc.v3i04.463.
- [16] V. Sutojo, T; Mulyanto, Edi; Suhartono, *Kecerdasan Buatan*. Jakarta Barat: Halaman Moeka Publishing, 2011.
- [17] Ahmad Izzuddin and M. Rizal Wahyudi, "Pengenalan Pola Daun untuk Membedakan Tanaman Padi dan Gulma Menggunakan Metode Principal Components Analysis (PCA) dan Extreme Learning Machine (ELM)," *ALINIER J. Artif. Intell. Appl.*, vol. 1, no. 1, 2020, doi: 10.36040/aliner.v1i1.2521.
- [18] A. A. Paturrahman and I. G. P. S. Wijaya, "Analisis Pengenalan Pola Daun Berdasarkan Fitur Canny Edge Detection dan Fitur GLCM Menggunakan Metode Klasifikasi k-Nearest Neighbor (kNN)," *J. Comput. Sci. Informatics Eng.*, vol. 5, no. 1, 2021, doi: 10.29303/jcosine.v5i1.388.
- [19] M. H. Saenudin Fauzan; Riza Ibnu Adam, "Classification of Covid-19 Using Feature Extraction GLCM and SVM Algorithm," *J. Mantik*, vol. 5, no. Vol. 5 No. 1 (2021): May: Manajemen, Teknologi Informatika dan Komunikasi (Mantik), pp. 179–183, 2021, [Online]. Available: <https://iocsience.org/ejournal/index.php/mantik/article/view/1284/907>
- [20] M. Ramadhani, "Klasifikasi Jenis Jerawat Berdasarkan Tekstur dengan Menggunakan Metode GLCM," *e-Prociding of Engineering*, vol. 5, no. 1, 2018.
- [21] D. Jumantoro, A., hartanto, R., prastiyanto, "Aplikasi Jaringan Saraf Tiruan Backpropagation Untuk Memprediksi Penyakit THT Di Rumah Sakit Mardi Rahayu Kudus," *J. Tek. Elektro*, vol. 1, no. 1, pp. 11–21, 2009.