

## DEEP PRE-TRAINED MODEL MENGGUNAKAN ARSITEKTUR DENSENET UNTUK IDENTIFIKASI PENYAKIT DAUN PADI

Arif Faizin, Arief Tri Arsanto, Moch. Lutfi, Abdur Rochim Musa

Prodi Teknik Informatika, Universitas Yudharta Pasuruan

arifusan@yudharta.ac.id, arief\_inf@yudharta.ac.id,

moch.lutfi@yudharta.ac.id, abdurrochimmusa@gmail.com

### ABSTRAK

Salah satu penyebab menurunnya produksi padi ialah serangan hama. hama padi meliputi hama busuk daun, hawar daun, dan hawar batang. Jenis penyakit yang berbeda memerlukan perlakuan yang berbeda, akan tetapi tidak semua petani paham akan sifat hama yang dapat menyebabkan kesalahan penanganan. Untuk mempermudah menyelesaikan masalah yang ada pada identifikasi hama penyakit daun padi perlu adanya inovasi sistem klasifikasi hama daun padi. Sistem ini didasarkan pada metode Convolutional Neural Network (CNN), metode atau model deep learning yang dikembangkan untuk meng-cover kelemahan metode machine learning. Namun, deep learning juga memiliki kelemahan yaitu waktu komputasi pada proses training yang sangat lama dan ukuran data besar. Oleh sebab itu model pra-pelatihan dibutuhkan untuk memperbaiki tingkat akurasi dan performance deep-learning serta mempermudah dalam pengembangan struktur model tanpa membangunnya dari awal. Dan pada ini sudah diimplementasikan model yang digunakan menghasilkan tingkat akurasi terbaik 98% dan validasi 93% serta error yang dihasilkan training dan validasi yaitu 0.07 dan 0.25 sedangkan waktu komputasi yang dibutuhkan untuk training dan validasi adalah 31 detik.

**Kata kunci:** *Deep Learning, Densenet, CNN, Pre-trained Model.*

### 1. PENDAHULUAN

Daun padi merupakan bagian terpenting pada tubuh tanaman dan juga yang paling mudah untuk diidentifikasi gejala penyakit yang terjadi pada tanaman padi. dalam hal ini dikarenakan penampakan daun lebih lebar dibandingkan dengan bagian tubuh lainnya, sehingga perubahan warna, bentuk dan bercak daun terjadi akibat serangan penyakit dapat terlihat lebih jelas, dan penyakit daun pada padi telah terdeteksi terbagi menjadi tiga kategori, seperti Smut, Bacterial Blight, dan Brown Spot [1]. Oleh sebab itu, daun padi bisa dipakai sebagai langkah pertama untuk mendeteksi penyakit pada daun padi [2]. Tujuan penelitian yang dilakukan untuk identifikasi morfologi gejala penyakit yang muncul pada daun padi dengan memakai teknik pengolah citra sebagai langkah awal untuk mendeteksi penyakit pada tanaman padi. Manfaat dari identifikasi diharapkan dapat membantu petani mengendalikan penyebaran penyakit padi.

Dengan perkembangan ilmu pengetahuan teknologi dan komputer dapat memungkinkan untuk identifikasi penyakit padi menggunakan teknik kecerdasan buatan, salah satunya adalah penggunaan metode deep learning [3]. Deep learning dianggap lebih baik dari metode tradisional seperti pada umumnya machine-learning [4]. Metode dan algoritma yang paling populer pada deep-learning ialah cnn (convolution neural network). Seperti yang kita bahas keuntungan menggunakan algoritma CNN pada metode dee-learning untuk identifikasi dan deteksi penyakit daun padi [5], hasil eksperimen studi [6] menunjukkan bahwa model tersebut digunakan untuk mengklasifikasikan penyakit padi hasil akurasinya adalah 93,3% dan untuk validasinya adalah 92,68%.

Peneliti lainnya telah membuktikan efektivitas performance cnn dalam mengidentifikasi penyakit melalui objek gambar, dan eksperimen hasil menunjukkan 91,83% tingkat akurasi untuk deteksi penyakit daun tanaman pada dataset publik [7]. Penelitian [8] mendeskripsikan deteksi dan klasifikasi penyakit padi menggunakan tiga jenis kategori penyakit daun padi yaitu Bacterial Leaf Blight, Leaf Smut, dan Brown Spot, kemudian untuk diekstraksi daun padi adalah bentuk, warna serta tekstur. Sedangkan algoritma yang diimplementasikan pada proses klasifikasi adalah svm (support vector machine). Hasil akurasi pada data latih mencapai 93,33% dan pada data validasi adalah 73,33%.

Pada penelitian ini Kontribusi dan tujuan peneliti adalah untuk mengembangkan framework deep pre-trained model arsitektur CNN-Densenet [9], sebuah desain yang diusulkan untuk mengklasifikasikan penyakit daun berdasarkan citra daun padi, sehingga proses input yang dilakukan dengan manual pada klasifikasi penyakit daun padi tidak diperlukan. Hal ini untuk mempersingkat waktu proses klasifikasi dan akurasi yang di hasilkan lebih baik dari penelitian sebelumnya, dan diharapkan bisa meningkatkan efisiensi proses klasifikasi deteksi penyakit daun pada padi.

### 2. TINJAUAN PUSTAKA

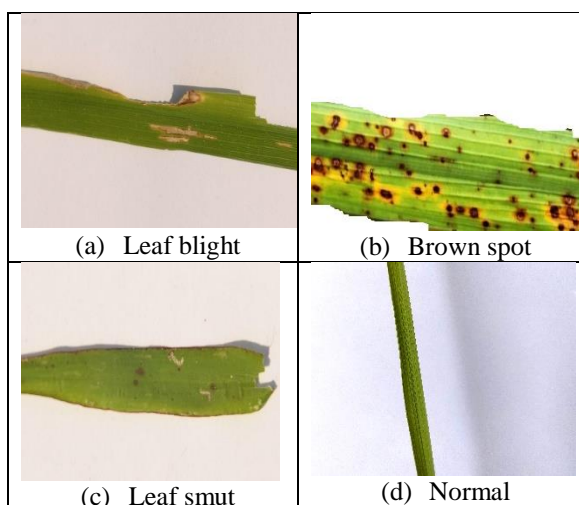
#### 2.1. Penelitian Terkait

Sebagai alternatif dari metode yang ada, dilakukan oleh peneliti [10] melakukan proses segmentasi pada daun padi berdasarkan statistik piksel citra daun. Statistik segmentasi nilai piksel yang dipakai adalah 0,20 dari maksimum hue, saturation, (HSV) dan nilai setiap komponen warna (0,20 \*

maksimum). Metode yang diusulkan diuji dengan komponen warna hsv dan grayscale menunjukkan hasil pengujian bahwa komponen warna “hue” berhasil dipecah, tetapi komponen warna “saturation”, “lightness” dan “grayscale” tidak terbelah. [11] membandingkan teknik algoritme yang berbeda seperti DC (Decision Tree), RF (Random Forest), NB (Naive Bayes), SVM dan JST. Dataset ini memiliki 3 kelas/penyakit: hawar daun, coklat bercak, dan penyakit busuk daun, tiap kategori berisi 40 data gambar dengan format tipe JPG. Tiga jenis model dapat dihasilkan dari perbandingan lima teknik algoritmik: model overfitting (hutan acak, pohon keputusan, naive bayes), model underfitting (SVM), dan model yang baik (ANN). Dari beberapa algoritma model terbaik adalah jst dengan nilai akurasi 87%. Model jst sangat konsisten menggungguli di peringkat 2 dan algoritma KNN juga konsisten kinerja baik pada data latih dan data uji akan tetapi masalah overfitting belum terbukti. Pada penelitian selanjutnya [12], penyakit padi diklasifikasikan menjadi tiga jenis berdasarkan foto penyakit daun padi. Tiga jenis penyakit termasuk: hawar daun, coklat bercak, dan penyakit busuk daun. Model yang dikembangkan menggunakan metode transfer learning menggunakan model pre-trained dari Resnet101, berupa lapisan padat, lapisan putus sekolah, dan lapisan normalisasi batch di atas lapisan yang terhubung penuh, ditambah beberapa lapisan arsitektur. Metode yang diusulkan pada data validasi klasifikasi adalah 100% dan nilai loss function adalah 5,61%.

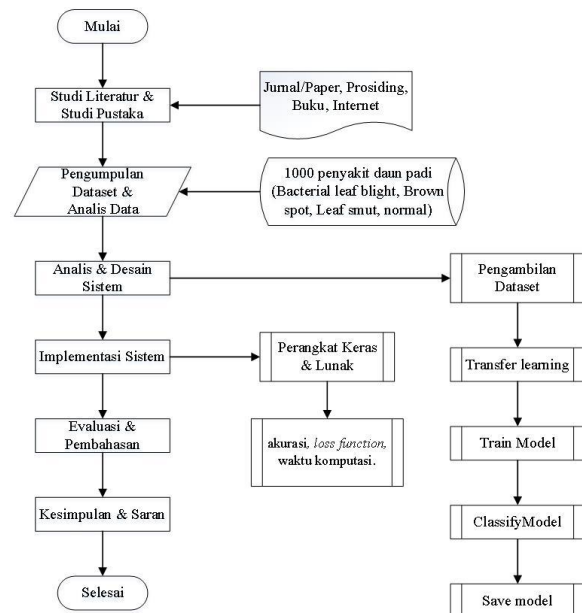
## 2.2. Jenis Penyakit Padi dan Daun Padi

Beras merupakan makanan pokok dan produksinya perlu dijaga. Budidaya padi tidak akan terlepas dari ancaman penyakit maupun hama yang sering menyerang. Penggundulan daun merupakan salah satu penyakit padi. Penyakit daun padi diklasifikasikan menjadi hawar daun, bercak coklat, penyakit busuk api dan penyakit normal.



Gambar 1. Jenis penyakit pada duan padi

Argumentasi untuk metode pada penelitian yang dilakukan dengan cara mengumpulkan data dan informasi yang menguatkan kebenaran materi yang dituangkan dalam argumen. Ada beberapa langkah yang perlu dilakukan selain memperbaiki masalah-masalah yang ada pada perancangan sistem. Bab ini menjelaskan bahan dan alat yang digunakan untuk pengembangan sistem dan metode penelitian.



Gambar 2. Alur penelitian

### 1. Studi literatur

Proses tahapan penelitian yang harus lakukan meliputi metode penelitian, landasan teori, dan referensi atau acuan penelitian sejenisnya. Pengumpulan data merupakan bagian dari metode penelitian namun pada penelitian yang dilakukan menggunakan studi literatur dan pustaka.

### 2. Pengumpulan data

Dataset daun padi diperoleh dari seorang petani padi di Kabupaten Pasuruan. Dataset yang digunakan sebanyak 1000 dataset yang terdiri dari 250 Bacterial leaf blight, 250 daun normal, 250 Brown spot, dan 250 Leaf smut. Ukuran piksel daun yang digunakan 1024x1024 piksel.

### 3. Data Analisis

Proses pengolahan dataset awal dilakukan untuk memotong gambar dan memisahkan dari daun Brown spot, Leaf smut, Bacterial leaf blight, dan normal. Pengolahan data dapat dilakukan dengan para petani yang mengerti penyakit daun tanaman padi.

### 4. Desain sistem

Desain sistem merupakan tahap rencana, gambaran maupun proses membangun sistem akan dibuat dengan metode penelitian yang diusulkan.

### 5. Implementasi sistem

Proses implementasi penelitian adalah proses coding di pemrog python pada google colab.

### 6. Uji coba dan evaluasi

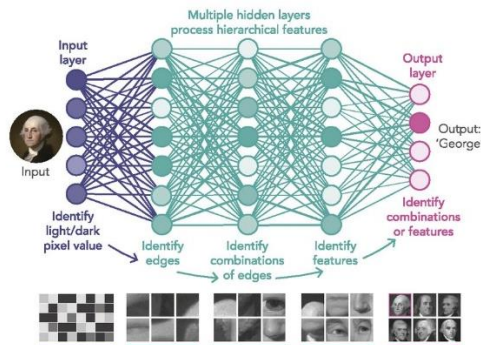
Ujicoba serta evaluasi yang dilakukan untuk melihat hasil performance dari metode yang diusulkan terdiri dari waktu komputasi, loss function serta akurasi hasil.

## 3. METODE PENELITIAN

### 3.1. Deep Learning

Dalam beberapa tahun terakhir, kecerdasan buatan telah berkembang pesat. Masalah yang sangat sulit diselesaikan manusia di masa lalu akan menjadi lebih mudah diselesaikan dengan hadirnya kecerdasan buatan. Kecerdasan buatan mengalami kesulitan menerapkan intuisi pada pengetahuan, sehingga konsep pembelajaran yang mendalam digunakan untuk memecahkan masalah ini.

Pembelajaran mendalam [13], bagian dari kecerdasan buatan dan pembelajaran mesin, adalah pengembangan jaringan saraf multi-layer yang menyediakan tugas-tugas presisi seperti pengenalan objek, pengenalan ucapan, dan terjemahan ucapan. Representasi sederhana yang digunakan model deep learning memungkinkan komputerisasi dapat membuat konsep ataupun pola yang lebih kompleks seperti halnya ditunjukkan pada gambar 2 berbeda dari teknik pembelajaran mesin tradisional karena secara konsep otomatis melakukan representasi data sebagai contoh video, gambar dan teks tanpa memperkenalkan aturan kode atau pengetahuan domain manusia.



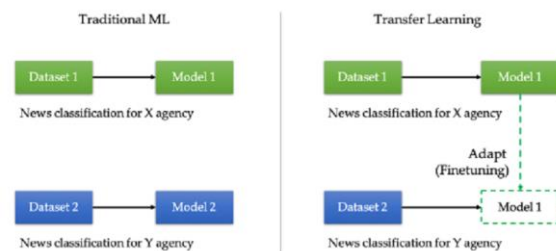
Gambar 3. Layer deep-learning

### 3.2. Pre-trained (Transfer learning)

Transfer learning merupakan proses pemakaian kembali model dan pengetahuan untuk keterkaitan dengan tugas-tugas lain [14][15][16]. Pembelajaran transfer juga dianggap sebagai pendalaman dari algoritma machine learning yang ada. Sudah banyak peneliti yang mendalami penelitian terkait dengan transfer learning dan bagaimana memahami pengetahuan yang ditransfer. Dimulai dari situ, istilah “learning to learn”, “integrasi pengetahuan”, dan “transfer induktif” sudah dipakai secara sinonim transfer-learning. Berbagai penelitian serta publikasi ilmiah memberikan definisi dari konteks yang berbeda [13].

TL ingin menggunakan pengetahuan dari tugas T1 untuk menyelesaikan masalah di tugas T2. Kami

berasumsi bahwa T1 terkait dengan T2, jadi jika Anda fasih di T1 (atau lebih fasih daripada tidak di T1), Anda dianggap fasih di T2. Dapat dilihat gambar 4, yang menunjukkan perbedaan antara menggunakan machine learning dan menggunakan TL. ciri khas machine learning adalah melatih model untuk setiap tugas. TL menggunakan model yang ada, yang disebut model pra-terlatih, untuk tugas-tugas baru. TL dimotivasi tidak hanya oleh kesamaan dua tugas, tetapi juga oleh ketersediaan data. Misalnya, tugas T1 memiliki kumpulan data besar dan tugas T2 memiliki kumpulan data kecil. Karena T1 dan T2 serupa, model yang dipasang ke T2 lebih cepat menyatu di T1 daripada melatih model di T2 dari awal.



Gambar 4. Mesin tradisional vs transfer learning.

Proses menyesuaikan model yang telah dilatih sebelumnya disebut penyetelan dapat dilihat pada gambar 5. Pertama, ganti lapisan terakhir (lapisan prediksi) dari model yang telah dilatih sebelumnya dengan lapisan baru secara acak yang diinisialisasi. Kemudian latih kembali model yang ada, dengan menerapkan data dari T2. Secara garis besar ada tiga kategori metode fine tuning.

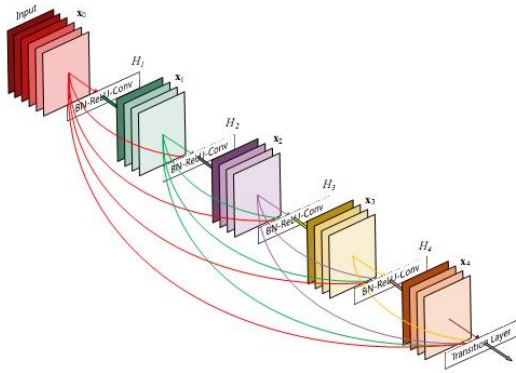
1. Bekukan beberapa lapisan. Bekukan beberapa lapisan (parameter tidak diperbarui selama penyempurnaan) dan latih yang lain.
2. Latih hanya layer terakhir yang baru. Bekukan semua kecuali lapisan terakhir tugas T2.
3. Latih semua level. Setelah memodifikasi layer terakhir, latih semua layer untuk tugas T2.



Gambar 5. Proses transfer learning

### 3.3. Densenet

Jaringan konvolusi padat (DenseNet) merupakan penghubung pada lapisan ke setiap lapisan lainnya dengan cara feedforward. Dalam jaringan konvolusi lapisan-L tradisional, ada koneksi L-1 pada setiap lapisan layer dan lapisan selanjutnya, akan tetapi pada jaringan tersebut mempunyai koneksi langsung  $L(L+1)/2$ . DenseNet memiliki beberapa keunggulan yang menarik. Ini mengurangi masalah gradien, meningkatkan ekspansi fungsi, mempromosikan penggunaan kembali fungsi, dan sangat mengurangi jumlah parameter [17].



Gambar 6. Arsitektur densenet

Pada Gambar 6, tiap kombinasi lapisan layer memakai batch normalisasi, aktif ReLU serta konvolusi menggunakan filter 3x3. disetiap blok terdapat input berupa bentuk matriks piksel dari citra, selanjutnya masuk ke tahap normalisasi batch untuk dapat mengurangi overfitting pada proses pembelajaran, aktivasi ReLU dipakai untuk merubah nilai variabel  $x$  menjadi nilai 0 jika variabel  $x$  nilainya negatif, begitu juga sebaliknya jika nilai variabel  $x$  tetap digunakan jika nilainya sama dengan 0 atau tidak kurang dari 0, konvolusi filter 3x3 untuk proses citra raster yang telah digunakan sebagai operasi aktivasi ReLU akan dikalikan dengan nilai matriks konvolusi filter 3x3 dan outputnya adalah nilai matriks yang telah diproses sebelumnya.

### 3.4. Preprocessing

Tahapan preprocessing dilakukan untuk resize untuk mengubah ukuran citra serta memperkecil ukuran citra pada arah horizontal dan vertikal. Hal ini bertujuan untuk menyeragamkan ukuran dari masing-masing citra yang digunakan selama proses training dan testing dikarenakan terjadinya perbedaan ukuran akan mengakibatkan proses training yang lama dan tingkat akurasi akan mempengaruhi. Normalisasi data merupakan proses menormalisasikan piksel mulai 0 sampai dengan 1.

$$rescale = \frac{img}{255}$$

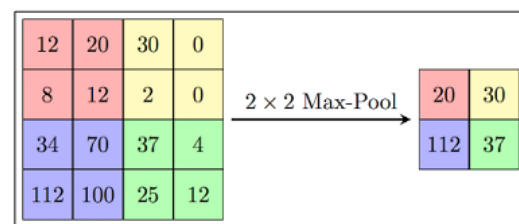
### 3.5. Convolution

Lapisan konvolusi adalah blok utama CNN dan terdiri dari berbagai filter yang diinisialisasi secara acak untuk melakukan operasi konvolusi yang bertindak sebagai ekstraktor fitur untuk memeriksa representasi fitur dari gambar input. Pada lapisan convolutional, neuron diatur dalam peta fitur. Setiap neuron dalam peta fitur terhubung sebagai bidang reseptif ke neuron di lapisan konvolusi sebelumnya melalui serangkaian bobot terlatih, juga dikenal sebagai bank filter[3].

$$h(x, y) = f(x, y) \times g(x, y)$$

### 3.6. Max Pooling

Max pooling bertujuan untuk mencapai invariansi spasial dengan mengurangi resolusi peta fitur, peta fitur yang dikumpulkan sesuai dengan lapisan layer sebelumnya. Metode pengambilan sampel yang dikurangi yang digunakan di sebagian besar CNN adalah agregasi maksimal. Agregasi maksimum membagi nilai output dari convolutional lapisan layer membentuk grid kecil yang banyak serta mendapatkan nilai terbesar pada tiap mesh untuk membangun nilai matriks gambar reduksi. Di sisi lain, rata-rata pooling dibagi nilai luaran menjadi banyak level yang lebih kecil kemudian diambil nilai rata-rata pada setiap level dalam gambar untuk membangun matriks reduksi [18]. Gambar 7. mengilustrasikan proses maxpooling.



Gambar 7. Ilustrasi max pooling

### 3.7. Normalisasi Batch

Operasi normalisasi batch digunakan untuk mempercepat proses pelatihan dan meningkatkan kecepatan pembelajaran selama pembuatan model. Normalisasi batch bekerja dengan menghaluskan distribusi setiap nilai input. Distribusi ini terus berubah selama proses pelatihan karena perubahan parameter lapisan sebelumnya[19].

Persamaan rata-rata batch mini:

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i$$

Persamaan MBV (Mini Batch Variance) :

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2$$

Persamaan normalisasi:

$$x_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

### 3.8. Aktivasi ReLU

Diperkenalkan oleh Geoffrey Hinton dan Vinod Nair dan digunakan dalam jaringan syaraf tiruan, fungsi aktivasi Rectified Linear Units (ReLU) digunakan untuk mengubah nilai  $x$  menjadi 0 jika nilai  $x$  negatif. Jika nilainya tidak kurang dari 0, berlaku sebaliknya[20].

$$f(x_i) = \max(0, x_i)$$



### 3.9. Aktivasi Softmax

Aktivasi softmax merupakan penerapan pada lapisan layer terakhir dari jaringan saraf tiruan. Nama softmax lebih umum digunakan dari pada Sigmoid, ReLU, ataupun Tanh. Distribusi probabilitas dipakai untuk menjumlah dari vektor bilangan real. Sedangkan fungsi softmax mengembalikan output berupa rentang nilai 0 sampai 1 dengan probabilitas jumlah sama dengan 1[21].

$$f(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

### 3.10. Cross Entropi

Cross entropy adalah fungsi loss dan gradient yang digunakan dalam klasifikasi ganda. pengukuran dua distribusi probabilitas dan entropi relatif pada dataset yang sama. Fungsi cross entropy loss digunakan untuk learning metode NN. Prinsip kerjanya adalah dengan mereduksi logaritma negatif dari dataset[22].

$$L_{\text{cross-entropy}} = - \sum_j y_i \log(y)$$

### 3.11. Confusion Matriks

Confusion matrix adalah matriks perbandingan antara hasil prediksi dan nilai aktual, serta dapat dengan jelas menampilkan prediksi dan detail prediksi untuk setiap kategori klasifikasi[16][23]. Pengukuran data dievaluasi dengan akurasi, presisi, dan recall. Hasil pengukuran ditampilkan dalam tabel klasifikasi yang mudah dibaca. Berikut perhitungannya:

Tabel 1. Confusion Matrix

Confusion matrix		Prediksi	
		Positive	Negative
Actual	Positive	A	B
	Negative	C	D
		sensitivity	specificity
		$A/(A+C)$	$D/(B+D)$
Accuracy		$(A+D)/(A+B+C+D)$	

## 4. HASIL DAN PEMBAHASAN

Sebelum dilakukan penelitian perlu adanya kebutuhan perangkat keras (hardware) dan perangkat lunak (software) yang dapat digunakan dalam implementasi penelitian di bawah ini tabel kebutuhan software dan hardware adalah sebagai berikut.

Tabel 1. Kebutuhan software dan hardware

No	Hardware	Spesifikasi
1	Laptop	Dell E7270
2	Procesor	Core i7-6600U@2.60GHz
3	Ram	16GB
4	SSD	NVMe 512GB
No	Software	
1	Windows 10	
2	Chrome + Google Colab	

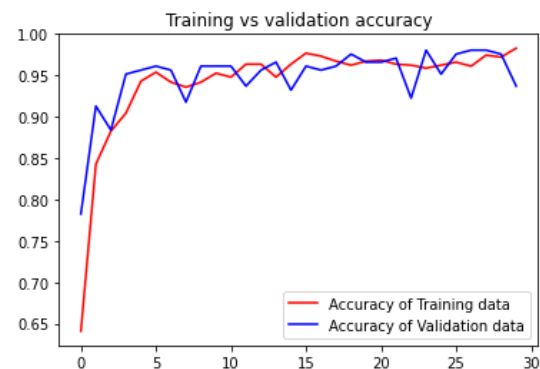
Pada tahap implementasi source code di program python dengan memanfaatkan Google Colab pada

implementasi metode yang diusulkan paramater transfer learning yang di gunakan adalah optimation menggunakan Adam, Batch Size 32, Learning rate 0.0001, Activation Softmax, Epoch 30 dan Shape input menggunakan 80,80,3. Sedangkan hasil akurasi yang didapatkan dengan menggunakan parameter yang di jabarkan menghasilkan tingkat akurasi yang cukup baik, berikut tabel hasil akurasi dengan arsitektur Densenet.

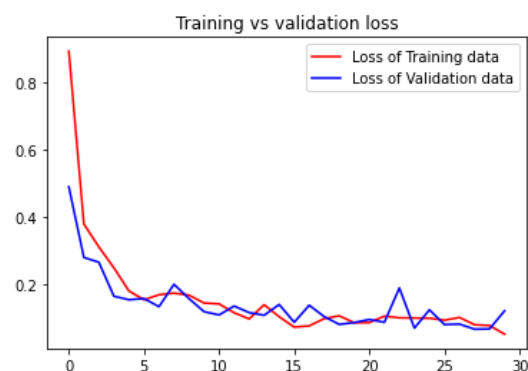
Tabel 2. Hasil Akurasi Densenet

Learning_rate	0.0001		
Data	Train	Valid	Waktu komputasi
Accuracy	98%	93%	31 detik
Error_rate	0.05	0.12	

Setelah dilakukan proses traning dan validasi terhadap dataset yang dipakai tahap selanjutnya dilakukan proses visualisasi grafik, di bawah ini adalah grafik hasil akurasi dan loss funtion.



Gambar 8. Grafik akurasi



Gambar 9. Grafik error rate

Berdasarkan hasil gambar 8 dan 9 serta tabel 2 dapat disimpulkan bahwa menggunakan deep pre-trained model densenet dengan epoch 30, batch\_size 32, learning rate 0.0001 dan dataset training berjumlah 1000 data citra daun padi dengan menggunakan target size 80x80x30 menghasilkan tingkat akurasi training 98% sedangkan validasi menggunakan 400 data dan akurasi validasi terbaik adalah 93% dan waktu komputasi yang di butuhkan adalah 31 detik. Dari perbandingan tersebut dapat disimpulkan bahwa

jumlah epoch berpengaruh terhadap tingkat akurasi, selain itu juga semakin banyak epoch yang digunakan maka semakin tinggi tingkat akurasi yang didapatkan.

## 5. KESIMPULAN DAN SARAN

Pada hasil penelitian yang sudah dilakukan dengan menggunakan pre-trained model densenet dengan memanfaatkan tools bantu google colab menghasilkan tingkat akurasi terbaik 98% dan validasi 93% serta error dari masing-masing proses training dan validasi yaitu 0.07 dan 0.25 sedangkan waktu komputasi yang dibutuhkan untuk training dan validasi adalah 31 detik maka kesimpulan dari hasil penelitian bahwa metode ini sangat baik jika diimplementasikan terhadap data penelitian yang digunakan.

## DAFTAR PUSTAKA

- [1] W. F. H. Purnamawati, Annida, Wawan Nugroho, Destiana Putri, "Deteksi Penyakit Daun pada Tanaman Padi Menggunakan Algoritma Decision Tree, Random Forest, Naïve Bayes, SVM dan KNN," *InfoTekJar*, vol. 5, no. 1, pp. 212–215, 2020, [Online]. Available: <https://doi.org/10.30743/infotekjar.v5i1.2934>.
- [2] S. Saidah, M. B. Adinegara, R. Magdalena, and N. K. Caecar, "Identifikasi Gejala Penyakit Padi Menggunakan Operasi Morfologi Citra," *TELKA - Telekomun. Elektron. Komputasi dan Kontrol*, vol. 5, no. 2, pp. 114–121, 2019, doi: 10.15575/telka.v5n2.114-121.
- [3] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, doi: 10.1038/nature14539.
- [4] M. Z. Alom *et al.*, "A state of the art survey on deep learning theory and architectures," *Electronics*, vol. 8, no. 3, 2019, doi: 10.3390/electronics8030292.
- [5] C. R. Rahman *et al.*, "Identification and recognition of rice diseases and pests using convolutional neural networks," *BiosystemsEngineering*, vol. 194, pp. 112–120, 2020, doi: 10.1016/j.biosystemseng.2020.03.020.
- [6] A. Islam, N. Rahman Shuvo, M. Shamsojjaman, S. Hasan, S. Hossain, and T. Khatun, "An Automated Convolutional Neural Network Based Approach for Paddy Leaf Disease Detection," (*IJACSA*) *International J. Adv. Comput. Sci. Appl.*, vol. 12, no. 1, pp. 280–288, 2021, doi: 10.14569/IJACSA.2021.0120134.
- [7] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkar, "Using deep transfer learning for image based plant disease identification," *Comput. Electron. Agric.*, vol. 173, no. March, p. 105393, 2020, doi: 10.1016/j.compag.2020.105393.
- [8] H. B. Prajapati, J. P. Shah, and V. K. Dabhi, "Detection and classification of rice plant diseases," *Intell. Decis. Technol. (IOS Press.*, vol. 11, no. 3, pp. 357–373, 2017, doi: 10.3233/IDT-170301.
- [9] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. K. Behera, "Deep feature based rice leaf disease identification using support vector machine," *Comput. Electron. Agric.*, vol. 175, no. December 2019, p. 105527, 2020, doi: 10.1016/j.compag.2020.105527.
- [10] K. Anwar and S. Setyowibowo, "Segmentasi Kerusakan Daun Padi pada Citra Digital," *J. Edukasi dan Penelit. Inform.*, vol. 7, no. 1, p. 39, 2021, doi: 10.26418/jp.v7i1.42331.
- [11] A. Purnamawati, W. Nugroho, D. Putri, and ..., "Deteksi Penyakit Daun pada Tanaman Padi Menggunakan Algoritma Decision Tree, Random Forest, Naïve Bayes, SVM dan KNN," *InfoTekJar*, 2020, [Online]. Available: <https://jurnal.uisu.ac.id/index.php/infotekjar/article/view/2934>.
- [12] Ulfah Nur Oktaviana, Ricky Hendrawan, Alfian Dwi Khoirul Annas, and Galih Wasis Wicaksono, "Klasifikasi Penyakit Padi berdasarkan Citra Daun Menggunakan Model Terlatih Resnet101," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 6, pp. 1216–1222, 2021, doi: 10.29207/resti.v5i6.3607.
- [13] I. G. and Y. B. and A. Courville, *Deep learning*, vol. 29, no. 7553, 2016.
- [14] B. Pardamean, T. W. Cenggoro, R. Rahutomo, A. Budiarto, and E. K. Karuppiyah, "Transfer Learning from Chest X-Ray Pre-trained Convolutional Neural Network for Learning Mammogram Data," *Procedia Comput. Sci.*, vol. 135, pp. 400–407, 2018, doi: 10.1016/j.procs.2018.08.190.
- [15] D. Sarkar, R. Bali, and T. Ghosh, *Hands-On Transfer Learning with Python Implement Advanced Deep Learning and Neural Network Models Using TensorFlow and Keras*. 2018.
- [16] Z. Jiang, Z. Dong, W. Jiang, and Y. Yang, "Recognition of rice leaf diseases and wheat leaf diseases based on multi-task deep transfer learning," *Comput. Electron. Agric.*, vol. 186, no. April, p. 106184, 2021, doi: 10.1016/j.compag.2021.106184.
- [17] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2261–2269, 2017, doi: 10.1109/CVPR.2017.243.
- [18] V. Suárez-Paniagua and I. Segura-Bedmar, "Evaluation of pooling operations in convolutional architectures for drug-drug interaction extraction," *BMC Bioinformatics*, vol. 19, pp. 92–93, 2018, doi: 10.1186/s12859-018-2195-1.
- [19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *32nd Int. Conf. Mach.*

- Learn. ICML 2015*, vol. 1, pp. 448–456, 2015.
- [20] A. F. Agarap, “Deep Learning using Rectified Linear Units (ReLU),” no. 1, pp. 2–8, 2018, [Online]. Available: <http://arxiv.org/abs/1803.08375>.
- [21] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation Functions: Comparison of trends in Practice and Research for Deep Learning,” pp. 1–20, 2018, [Online]. Available: <http://arxiv.org/abs/1811.03378>.
- [22] K. P. Murphy, *Machine learning : A Probabilistic Perspective*, vol. 5, no. 2. The MIT Press, 2012.
- [23] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009, doi: 10.1016/j.ipm.2009.03.002.