

Sistem Informasi Deteksi Penyakit Pada Tanaman Padi (Brown Spot, Hispa, Leaf Blast) Menggunakan Metode Convolutional Neural Network (CNN)

Yusuf Fadlila Rachman¹, Pratiwi Susanti^{2*}, Affriza Brilyan Relo Pambudi Agus Putra³, Nuriya Imroatu Rahmawati¹

¹Program Studi Teknik Informatika, Universitas Sebelas Maret, Indonesia.

²Program Studi Teknik Informatika, Universitas PGRI Madiun, Indonesia.

³Program Studi Bisnis Digital, Universitas Telkom Purwokerto, Indonesia.

Artikel Info

Kata Kunci:

Artificial Intelligence;
Brown Spot;
Convolutional Neural Network;
Deteksi;
Hispa;
Leaf Blast.

Keywords:

Artificial Intelligence;
Brown Spot;
Convolutional Neural Network;
Deteksi;
Hispa;
Leaf Blast.

Riwayat Artikel:

Submitted: 14 November 2024

Accepted: 30 November 2024

Published: 30 November 2024

Abstrak: Sebagai salah satu produsen padi terbesar di dunia, sering menghadapi penurunan produksi akibat serangan penyakit padi. Pendeteksian penyakit secara manual kurang efektif karena keterbatasan pengetahuan petani. Solusi yang ditawarkan untuk mengatasi masalah ini adalah untuk mengembangkan sistem informasi berbasis kecerdasan buatan yang mampu mendeteksi secara otomatis penyakit pada tanaman padi, termasuk Brown Spot, Hispa, dan Leaf Blast, dengan menggunakan metode Convolutional Neural Network (CNN). Sistem ini diharapkan dapat membantu petani dalam melakukan deteksi dini terhadap penyakit padi, sehingga meningkatkan efisiensi dan kualitas produksi pertanian. Sistem mengolah data gambar padi dan mendeteksi kondisi kesehatannya, termasuk mendeteksi padi sehat serta penyakit Brown Spot, Hispa, dan Leaf Blast. Penelitian ini menggunakan 3.355 dataset yang dibagi menjadi 335 untuk proses training, 335 untuk testing, dan 2.685 untuk validasi. Metode yang digunakan pada pengembangan sistem menerapkan pendekatan pengembangan perangkat lunak Waterfall, yang mencakup analisis kebutuhan, desain sistem, implementasi, pengujian, dan pemeliharaan sistem. Sistem "Paddy-AI" yang dikembangkan mampu mencapai akurasi 85% dalam mendeteksi gambar.

Abstract: As one of the largest rice producers in the world, it often faces a decline in production due to rice disease attacks. Manual detection of diseases is less effective due to the limited knowledge of farmers. The solution offered to overcome this problem is to develop an artificial intelligence-based information system capable of automatically detecting diseases in rice plants, including Brown Spot, Hispa, and Leaf Blast, using the Convolutional Neural Network (CNN) method. The system is expected to assist farmers in conducting early detection of rice diseases, thereby improving the efficiency and quality of agricultural production. The system processes rice image data and detects its health condition, including detecting healthy rice as well as Brown Spot, Hispa, and Leaf Blast diseases. This research used 3,355 datasets divided into 335 for training, 335 for testing, and 2,685 for validation. The method used in system development applies the Waterfall software development approach, which includes requirements analysis, system design, implementation, testing, and system maintenance. The developed 'Paddy-AI' system is able to achieve 85% accuracy in detecting images.

Corresponding Author:

Pratiwi Susanti

Email: pratiwi.susanti@unipma.ac.id

PENDAHULUAN

Penurunan produksi pertanian akibat terjangkitnya penyakit masih kerap dialami oleh petani di Indonesia. Beberapa faktor yang menjadi penyebab turunnya produksi padi, antara lain miskonsepsi pengetahuan tentang jenis dan macam penyakit, kurangnya pelatihan dan penyuluhan mengenai perawatan pertanian, sehingga terbatasnya informasi dikalangan petani (Sitompul, 2022). Penyakit tanaman padi disebabkan oleh bakteri, jamur, cuaca, kelembapan tanah, virus, dan hama. Kondisi ini menyebabkan sekitar 200.000 - 300.000 ton padi terserang hama setiap tahun di Indonesia (Felix & all, 2020). Penyakit seperti blast, hawar daun, hawar batang, kresek, brown spot, dan hispa harus mendapatkan penanganan yang tepat untuk mencegah penurunan kualitas dan kuantitas produksi.

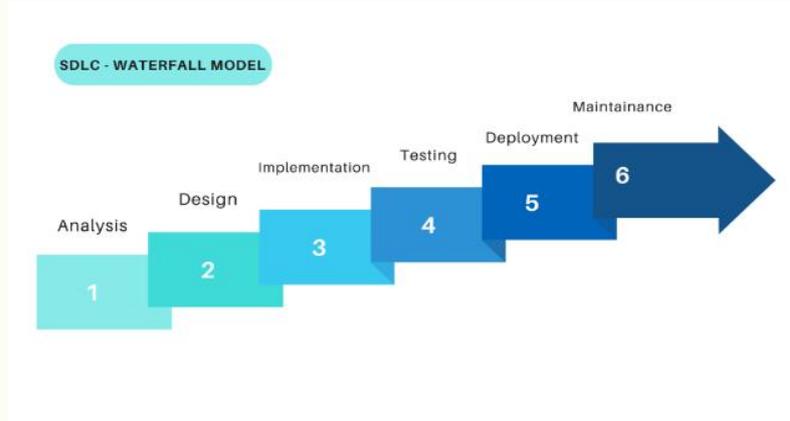
Pestisida kimia sering digunakan oleh petani untuk mengatasi penyakit pada tanaman padi, namun jika tidak digunakan dengan benar atau terlambat, dapat menyebabkan dampak negatif. Oleh karena itu, solusi deteksi dini dan pengendalian penyakit tanaman padi bisa dilakukan dengan menggunakan teknologi kecerdasan buatan (Masykur, 2022). Penelitian terdahulu oleh Jinan dan Hayadi, 2022 menunjukkan bahwa penggunaan *Convolutional Neural Network (CNN)* dalam deteksi penyakit padi, yaitu *Bacterial Leaf Blight*, *Brown Spot*, dan *Leaf Spot* mampu menghasilkan akurasi sebesar 91% (Jinan, 2022). Algoritma CNN juga telah digunakan oleh perusahaan besar seperti Google dan Facebook dalam berbagai aplikasi, menunjukkan keandalan dan akurasi algoritma tersebut dalam pengolahan citra (Raswir, 2022).

Pada penelitian ini, dilakukan proses diagnose penyakit pada tanaman padi berdasarkan gambar daun. Proses diagnose dilakukan secara otomatis dengan tujuan untuk mengetahui jenis penyakit padi, yaitu *brown spot*, atau *hispa* (Sitompul, 2022). Proses pengenalan penyakit padi dilakukan menggunakan algoritma *Convolutional Neural Network* dan *Package Keras* (Azhar, Santoso, & Soetrisno, 2021). Penggunaan package keras dapat memudahkan pembuatan model jaringan syaraf tiruan dan implementasi *Deep Learning* (Irawan, 2022). Sistem deteksi penyakit padi dijalankan menggunakan web browser, dimana pengguna cukup mengunggah foto sample padi kemudian secara otomatis sistem akan mendeteksi penyakit berdasarkan foto tersebut, serta memberikan deskripsi dan langkah-langkah penanganan yang tepat sesuai dengan penyakit yang terdeteksi.

Penelitian ini bertujuan untuk meningkatkan efisiensi dan kualitas produksi padi. Sistem identifikasi penyakit pada tanaman padi secara otomatis dapat membantu dalam perawatan tanaman, mempercepat proses deteksi penyakit, dan memberikan langkah penanganan yang tepat. Sistem yang dibangun belum pernah terealisasi sebelumnya dan diharapkan dapat menjadi solusi yang berguna bagi sektor pertanian.

METODE

Penelitian yang sebelumnya pernah dilakukan oleh Cycle (Rahmawati, 2023) dengan pendekatan best practice dari *Waterfall Software Development Life*. Diagram *Waterfall* dapat dilihat pada Gambar 1.



Gambar 1. Metode Penelitian (Findawati, 2018)

Penjelasan metode penelitian menggunakan waterfall sebagai berikut:

1. Analisis Kebutuhan Sistem.

Pada tahap ini, analisis dilakukan terhadap kebutuhan fungsional dan non-fungsional yang mendukung sistem. Kebutuhan fungsional mencakup fitur, data, pengguna, dan hak akses. Kebutuhan non-fungsional meliputi ketahanan, kecepatan, dan keamanan sistem.

2. Desain dan Perancangan Sistem

Pada tahap ini, dilakukan desain mockup sistem dan perancangan database. Perancangan mockup sistem dilakukan menggunakan figma sedangkan database yang digunakan adalah MySQL.

3. Implementasi.

Tahap ini melibatkan pembuatan sistem informasi dan deteksi dini penyakit pada tanaman padi menggunakan metode CNN dengan framework Flask. Sistem ini dibangun menggunakan bahasa pemrograman Python sebagai backend, HTML sebagai frontend, serta library Keras dan TensorFlow (Benbrahim, Hachim, & Amine, 2020). Dataset citra daun padi dibagi menjadi empat kelas dan diimplementasikan pada Google Collab untuk membuat data training, data test, dan data validasi (Guntara, 2023). Database MySQL digunakan untuk menyimpan data yang telah dideteksi dan informasi yang ditampilkan oleh sistem (Novendri, Saputra, & Firman, 2019).

4. Pengujian

Tahap ini dilakukan untuk mengidentifikasi bug atau error pada sistem. Pengujian performa untuk klasifikasi dilakukan menggunakan (Xu, 2020). Pengujian dilakukan untuk memastikan program bebas dari kesalahan sebelum diluncurkan ke pengguna lain. Kesalahan program diklasifikasikan menjadi kesalahan sintaks, run-time error, dan kesalahan logika.

5. Deployment

Pada penelitian ini bertujuan menghasilkan sistem informasi dan deteksi dini penyakit tanaman padi yang dapat digunakan oleh petani dan sektor pertanian untuk pencegahan awal kasus penyakit tanaman padi yang dapat menurunkan produksi. Sistem ini diharapkan akurat dalam mendeteksi penyakit pada tanaman padi dan membantu petani dengan informasi yang terdeteksi.

6. Maintenance dan Alih Pengetahuan

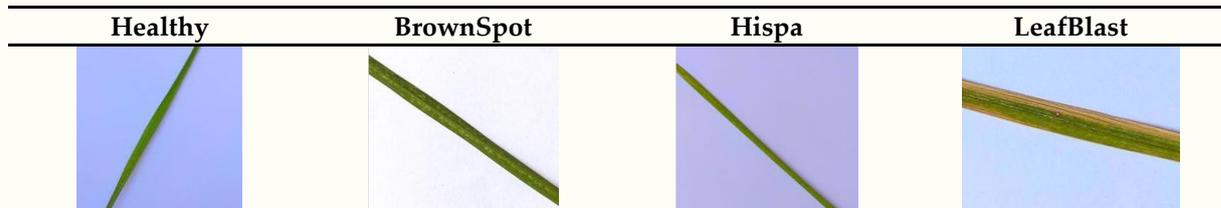
Setelah sistem dikembangkan dan diuji, pemeliharaan dilakukan untuk memastikan sistem berjalan dengan baik dan maksimal. Pemeliharaan bertujuan memperpanjang daya guna sistem. Alih pengetahuan penting untuk memastikan sistem dapat digunakan dengan mudah oleh pengguna sesuai dengan analisis dan perancangan yang telah ditetapkan.

Dataset Penelitian

Pada riset ini, dataset diperoleh dari Kaggle dengan judul "rice_leaf_image" yang telah diupload pada tahun 2022 oleh Nizor Ogbezuode (Sitompul, 2022). Dataset pada penyakit brown spot yaitu sebanyak 523 dataset, 1.488 dataset pada kondisi healthy, 565 dataset pada penyakit hispa, dan 779

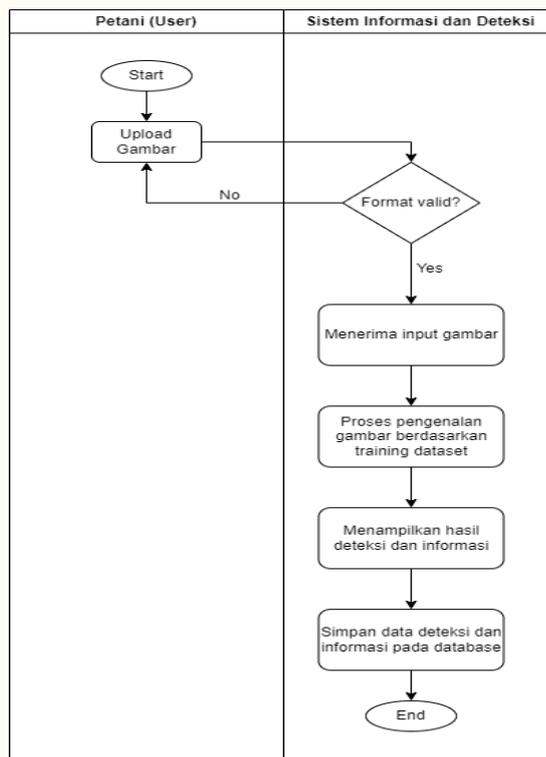
dataset pada penyakit Leaf Blast. Dataset citra daun padi yang telah dibagi menjadi empat kelas yaitu tanaman padi dengan kondisi citra daun sehat atau dengan penyakit brown spot, hispa, dan leaf blast. Selanjutnya akan diimplementasikan menggunakan google collab untuk membuat data training, data test, dan data validasi (Guntara, 2023). Pada saat training, sistem akan melakukan proses pengenalan gambar yang telah tersedia pada dataset menggunakan algoritma CNN sesuai dengan pembagian kelas yang ada (Nurani, 2022). Dataset dapat dilihat pada Tabel 1.

Tabel 1. Foto Padi



Alur Sistem

Pada penelitian ini, alur sistem yang dibuat melibatkan beberapa langkah utama yang di ilustrasikan melalui diagram swimlane. Diagram swimlane ini memberikan gambaran yang jelas tentang alur proses yang terjadi dalam sistem yang dikembangkan (Nurlita & Anggraini, 2023). Diagram swimlane dapat dilihat pada Gambar 3, yang memberikan ilustrasi visual dari alur kerja sistem dari awal hingga akhir.



Gambar 2. Alur Sistem

Berikut adalah penjelasan tahapan penelitian secara detail:

1. Setelah sistem terbuka, user atau petani dapat mengunggah file gambar dengan format jpg. Sistem akan memeriksa format file gambar yang diinputkan, jika tidak sesuai maka sistem akan menampilkan pemberitahuan kepada user agar melakukan upload ulang sesuai dengan format.
2. Setelah user mengupload gambar, sistem akan memproses pengenalan gambar menggunakan algoritma Convolutional Neural Network (CNN) yang telah dilatih sebelumnya. Jika sistem telah mengenali gambar yang diinputkan oleh user, maka sistem akan menampilkan hasil deteksi

penyakit serta menampilkan informasi berupa deskripsi penyakit serta langkah penanganan yang harus dilakukan.

3. Setelah sistem berhasil menampilkan hasil deteksi dan informasi penyakit tanaman padi, data tersebut akan tersimpan pada database, termasuk gambar yang diupload oleh user melalui sistem.

HASIL DAN PEMBAHASAN

Analisis Kebutuhan Sistem

Tahap pertama penelitian dimulai dengan mengidentifikasi kebutuhan fungsional dan non-fungsional aplikasi. Kebutuhan fungsional dan non-fungsional didapatkan dari hasil analisis permasalahan dan literatur review. Tabel kebutuhan fungsional dan non fungsional dapat dilihat pada Tabel 2 dan 3.

Tabel 2. Kebutuhan Fungsional Aplikasi

<i>Requirement ID</i>	<i>Description</i>
RF-01	Petani melakukan upload file pada sistem RF-01.1 Petani dapat melakukan upload file ke dalam sistem dengan format gambar berupa jpg atau jpeg.
RF-02	Sistem melakukan training pada gambar yang diinputkan RF-02.1 Sistem melakukan menangkap inputan gambar dan melakukan training sesuai dengan yang telah disimpan pada save model .pth
RF-03	Sistem melakukan deteksi gambar RF-03.1 Sistem melakukan proses deteksi gambar sesuai dengan hasil training yang telah didapatkan sebelumnya. RF-03.2 Sistem menampilkan hasil informasi berdasarkan proses deteksi yang telah dilakukan.
RF-04	Sistem melakukan simpan data pada <i>database</i> . RF-04.1 Sistem menyimpan data ke dalam <i>database</i> setelah proses deteksi selesai dilakukan Sistem menampilkan kembali hasil deteksi dan informasi.
RF-05	RF-05.1 Sistem menyimpan <i>history</i> deteksi dan informasi pada database dan ditampilkan melalui website. RF-05.2 Sistem menampilkan kembali hasil deteksi dan informasi pada website dengan memilih detail. RF-05.3 Sistem melakukan hapus data history dengan memilih <i>button</i> hapus.

Berdasarkan pada tabel tersebut, kebutuhan fungsional pada sistem ini memiliki dua requirement ID dengan masing-masing deskripsi di dalamnya. Kebutuhan fungsional ini telah menggambarkan proses serta layanan yang akan tersedia pada sistem.

Tabel 3. Kebutuhan Non-Fungsional

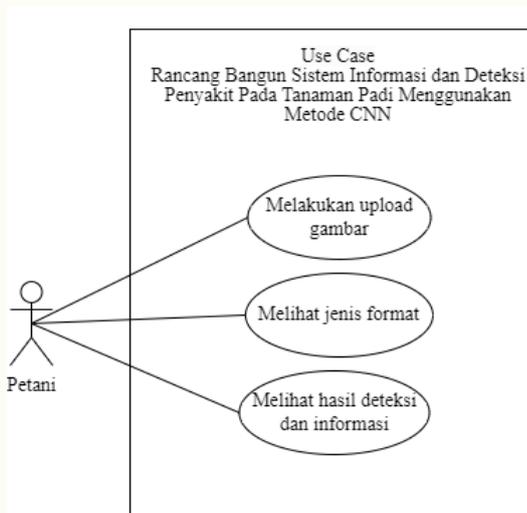
<i>Requirement ID</i>	<i>Attribute Software</i>	<i>Description</i>
NF-01	<i>Performance</i>	NF-01.1 Sistem dapat melakukan deteksi gambar dengan cepat dan akurat.

		NF-01.2	Sistem dapat melakukan proses deteksi secara berulang dengan objek yang sama ataupun berbeda tanpa adanya kelambatan atau <i>bug</i> .
		NF-01.3	Sistem menampilkan informasi secara tepat dan akurat sesuai dengan deteksi objek yang dihasilkan.
NF-02	<i>Usability</i>	NF-02.1	Sistem memberikan tampilan web yang mudah sehingga tidak menyulitkan pengguna dalam pemakaiannya.
NF-03	<i>Portability</i>	NF-03.1	Sistem dibangun menggunakan framework flask yang lebih ringan sehingga pengguna lebih mudah dalam mengaksesnya.

Berdasarkan pada tabel di atas, kebutuhan non fungsional pada sistem ini memiliki tiga requirement ID dengan attribute software berupa performance untuk menjelaskan kecepatan sistem, usability untuk menjelaskan kemudahan penggunaan sistem, dan portability untuk menjelaskan kemudahan dalam mengakses sistem.

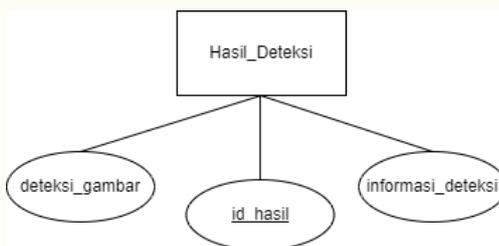
Desain dan Perancangan Sistem

Pada sistem ini, terdapat satu aktor utama yaitu user atau petani. Berikut merupakan diagram use case dari Sistem informasi dan deteksi dini penyakit pada tanaman padi menggunakan algoritma CNN dan framework flask, yang mendeskripsikan interaksi antar aktor. Use case diagram dapat dilihat pada gambar 4.



Gambar 3. Use Case Diagram (Findawati, 2018)

Pada sistem ini, petani dapat melakukan proses upload file berupa gambar dengan format jpg. Ketika gambar telah diupload, maka sistem akan menerima input gambar tersebut dan melakukan proses deteksi serta pengenalan objek berdasarkan data training pada dataset yang telah dilakukan melalui algoritma CNN. Perancangan database pada penelitian ini menggunakan Entity Relationship Diagram (ERD). Rancangan ERD dari Sistem informasi dan deteksi penyakit pada tanaman padi menggunakan metode convolutional neural network (CNN) dapat dilihat pada Gambar 4.



Gambar 4. Entity Relationship Diagram (Findawati, 2018)

Berdasarkan gambar ERD di atas, terdapat entitas hasil_deteksi. Entitas hasil_deteksi memiliki atribut id_hasil, deteksi_gambar, dan informasi_deteksi. Sehingga penyimpanan pada database dilakukan secara langsung melalui sistem ketika telah terdapat gambar yang diupload oleh user dan telah melalui tahap deteksi.

Implementasi Convolutional Neural Network

Dataset ini dibagi menjadi tiga bagian: training, testing, dan validation. Masing-masing bagian terdiri dari 335 data untuk testing dan training, serta 2685 data untuk validation. Pembagian ini penting untuk mempengaruhi proses implementasi dan hasil akurasi dari model CNN yang dihasilkan. Model arsitektur CNN yang digunakan adalah ResNet50, yang dijalankan menggunakan library PyTorch (Benbrahim, Hachim, & Amine, 2020). Model ini menghasilkan hasil simpan dengan format .pth.

Berikut adalah tahapan implementasi algoritma CNN:

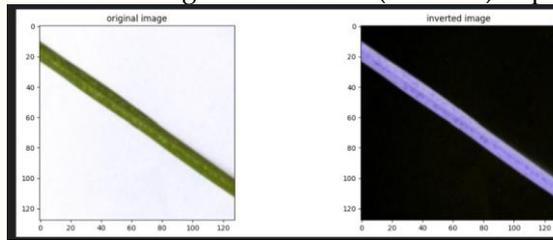
1. Membuat folder ipynb dan mengimport library yang diperlukan seperti numpy, pandas, os, matplotlib, torch. Implementasi pembuatan model dapat dilakukan menggunakan jupyter notebook, Kaggle, maupun google collab.
2. Memanggil dataset yang digunakan untuk training dan validation model. Saat memasukkan dataset, pastikan dataset benar, gambar jelas, dan jumlah dataset jangan terlalu sedikit agar dapat menghasilkan akurasi yang baik.
3. Memasukkan number of unique. Proses ini dilakukan untuk mencatat dan menentukan jumlah jenis dan kondisi padi yang dimasukkan dalam dataset. Sehingga terdapat 4 jenis tanaman padi dengan kondisi brown spot, hispa, dan leaf blast.
4. Melakukan training dan test dataset dengan tensor dimana menggunakan library pytorch dengan menyertakan transform.compose untuk melakukan penggabungan beberapa transformasi data menjadi satu transformasi yang dapat diterapkan pada data. Resize dibuat dengan ukuran 128 x 128 pixel. Sehingga, tahap ini digunakan untuk mempersiapkan gambar dan format yang digunakan dalam pelatihan dan pengujian model.
5. Melakukan visualisasi single image untuk memeriksa dan memahami informasi mengenai contoh awal dalam dataset gambar seperti bentuk gambar, label numerik, label teks, dan representasi pixel pada gambar dataset yang telah dimasukkan. Hasil representasi dapat dilihat pada Gambar 5.

```

... Image shape: torch.Size([3, 128, 128])
Image Label: 0
Image Label: BrownSpot
tensor([[[[0.9922, 0.9922, 0.9922, ..., 0.9647, 0.9647, 0.9647],
         [0.9922, 0.9922, 0.9922, ..., 0.9647, 0.9647, 0.9647],
         [0.9922, 0.9922, 0.9922, ..., 0.9569, 0.9569, 0.9569],
         ...,
         [0.9961, 0.9922, 0.9922, ..., 0.9569, 0.9569, 0.9569],
         [0.9922, 0.9922, 0.9922, ..., 0.9688, 0.9688, 0.9688],
         [0.9882, 0.9922, 0.9922, ..., 0.9688, 0.9688, 0.9688]],
        ...,
        [[0.9922, 0.9922, 0.9922, ..., 0.9647, 0.9647, 0.9647],
         [0.9922, 0.9922, 0.9922, ..., 0.9647, 0.9647, 0.9647],
         [0.9922, 0.9922, 0.9922, ..., 0.9569, 0.9569, 0.9569],
         ...,
         [0.9961, 0.9922, 0.9922, ..., 0.9569, 0.9569, 0.9569],
         [0.9922, 0.9922, 0.9922, ..., 0.9688, 0.9688, 0.9688],
         [0.9882, 0.9922, 0.9922, ..., 0.9688, 0.9688, 0.9688]],
        ...,
        [[1.0000, 1.0000, 1.0000, ..., 1.0000, 1.0000, 1.0000],
         [1.0000, 1.0000, 1.0000, ..., 1.0000, 1.0000, 1.0000],
         [1.0000, 1.0000, 1.0000, ..., 0.9961, 0.9961, 0.9961],
         ...,
         [1.0000, 1.0000, 1.0000, ..., 1.0000, 1.0000, 1.0000],
         [1.0000, 1.0000, 1.0000, ..., 1.0000, 1.0000, 1.0000],
         [0.9961, 1.0000, 1.0000, ..., 1.0000, 1.0000, 1.0000]]]])
...
    
```

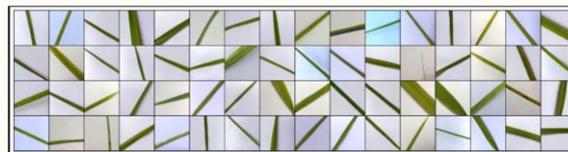
Gambar 5. Visualisasi Single Image

- Menampilkan contoh gambar pada dataset. Pada proses ini akan menampilkan dua gambar secara berdampingan yaitu gambar asli dan gambar terbalik (inverted) dapat dilihat pada Gambar 8.



Gambar 6. Contoh Gambar pada Dataset

- Validasi dataset dan data loader. Proses ini akan dilakukan validasi dataset dengan membagi menjadi dua subset yaitu dataset train dan dataset val. Lalu melakukan data loader untuk mengelola data dalam batch dalam pelatihan dan pengujian model machine learning atau jaringan syaraf. Data loader dapat memuat dan mengirimkan data secara efisien dalam jumlah batch yang ditentukan ke model dalam proses pelatihan dan pengujian.
- Visualisasi batch of image. Pada proses ini dilakukan untuk memeriksa dan memahami data yang digunakan dalam pelatihan model dengan menampilkan gambar-gambar dalam satu batch dataset pelatihan yang telah diatur dalam bentuk grid. Hasil visualisasi batch of image dapat dilihat pada Gambar 9.



Gambar 7. Visualisasi Batch of Image

- Membentuk klasifikasi model. Dalam membuat model klasifikasi gambar, telah diimplementasikan menggunakan class "ImageClassificasionBase" yang merupakan sub kelas dari "nn.Module".
- Memasukkan model ResNet50. Model arsitektur ResNet50 dimasukkan dalam model pelatihan sebagai basis.
- Training dan Evaluation. Model telah dilakukan training menggunakan fit model klasifikasi gambar dengan training loop. Fungsi fit ini digunakan untuk melatih model dengan melakukan iterasi pada dataset pelatihan dalam jumlah epoch yang ditentukan. Pada setiap epoch, fungsi ini melakukan Langkah-langkah pelatihan seperti menghitung loss, gradient, dan memperbarui parameter model menggunakan optimizer. Setelah itu, fungsi ini melakukan evaluasi pada dataset validasi.
- Memasukkan GPU yang digunakan. Dengan menggunakan fungsi get_default_device(), to_device(), dan kelas DeviceDataLoader, data dapat dengan mudah dipindahkan ke perangkat yang sesuai dan mengiterasi melalui DataLoader yang sudah dipindahkan ke perangkat tersebut. Hal ini bermanfaat ketika ingin menggunakan GPU untuk mempercepat pelatihan model jika tersedia, atau menggunakan CPU jika tidak tersedia GPU.
- Training model. Proses ini telah dilakukan selama 10 epoch dengan learning rate 0.001, menggunakan model yang telah didefinisikan sebelumnya, dan loader data pelatihan (train_loader) serta loader data validasi (val_loader). Selain itu, digunakan optimizer Adam (torch.optim.Adam) sebagai optimizer yang digunakan dalam pelatihan. Hasil nilai epoch yang ditampilkan setelah pelatihan dapat dilihat pada Tabel 3.

Tabel 3. Perbandingan Nilai Epoch

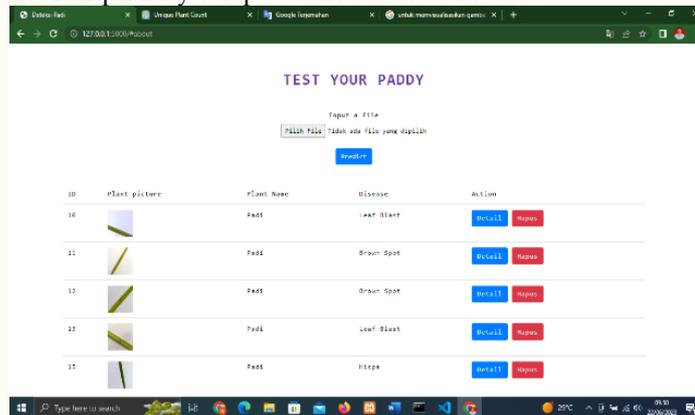
Epoch	Train loss	Acc	Vall loss	Acc
0/9	1.1060	0.5505	0.8491	0.6866
1/9	0.7179	0.7251	0.6419	0.7612
2/9	0.5341	0.7933	0.7038	0.7284

3/9	0.3566	0.8723	0.5805	0.7940
4/9	0.2313	0.9214	0.6160	0.7821
5/9	0.2328	0.9199	0.5977	0.7851
6/9	0.1660	0.9430	0.7946	0.7791
7/9	0.1157	0.9639	0.6377	0.8090
8/9	0.0990	0.9706	0.6901	0.8281
9/9	0.0876	0.9798	0.6096	0.8507

Berdasarkan hasil tersebut, nilai valuations akurasi maksimal yang dihasilkan yaitu sebesar 85,07% dengan epoch sebanyak 10 parameter.

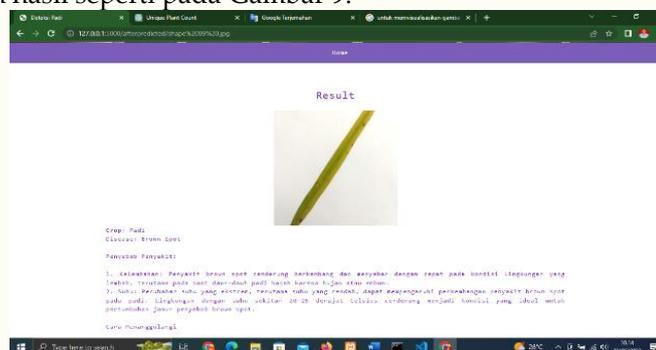
Implementasi Website

Aplikasi pendeteksi penyakit padi akan bekerja ketika menerima inputan berupa gambar. Ketika gambar dimasukkan ke dalam halaman website, maka sistem akan melakukan klasifikasi penyakit sesuai dengan indikasi gambar dan menampilkan penyebab serta cara pencegahan sesuai dengan jenis penyakit yang terdeteksi. Hasil prediksi berupa gambar, jenis penyakit, dan deskripsi disimpan dalam sistem dan dapat ditampilkan kembali oleh pengguna. Pengguna juga dapat menghapus riwayat prediksi jika diperlukan. Tampilannya dapat dilihat ada Gambar 8.



Gambar 8. Hasil Upload Gambar

Gambar 8 menunjukkan halaman upload gambar padi dan histori hasil diagnose penyakit padi. Hasil dari deteksi penyakit yang akan dimasukkan ke dalam database dan ditampilkan pada halaman upload gambar padi. Ketika sistem berhasil mendiagnosa penyakit padi, maka sistem akan menampilkan halaman hasil seperti pada Gambar 9.



Gambar 9. Halaman Result

Halaman result merupakan halaman hasil berdasarkan deteksi yang telah ditampilkan oleh sistem. Pada halaman ini, sistem akan mendeteksi serta menampilkan jenis penyakit sesuai dengan gambar yang telah diinput, penyebab serta solusi atau cara pencegahan.

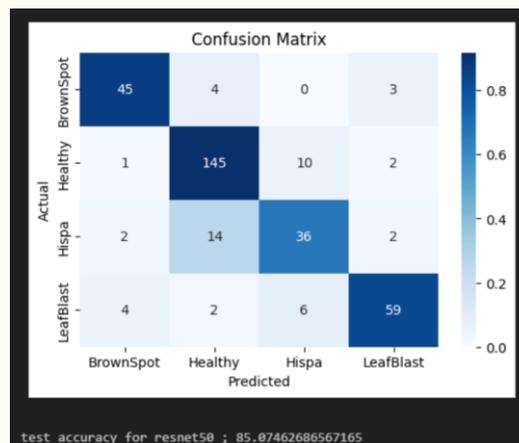
Pengujian Sistem

Pengujian dengan metode white box dilakukan untuk mengukur akurasi model berdasarkan nilai epoch yang dievaluasi melalui confusion matrix setelah tahap running dan save model (Suryadewiansyah & Eng Tju, 2023). Confusion matrix ini memungkinkan untuk melihat peluang dan rata-rata akurasi model, serta jumlah dataset yang terbaca dengan baik dan yang tidak terbaca (Xu, 2020). Hasil confusion matrix dapat dilihat pada Gambar 10.

	precision	recall	f1-score	support
0.0	0.87	0.87	0.87	52
1.0	0.88	0.92	0.90	158
2.0	0.69	0.67	0.68	54
3.0	0.89	0.83	0.86	71
accuracy			0.85	335
macro avg	0.83	0.82	0.83	335
weighted avg	0.85	0.85	0.85	335

Gambar 10. Hasil Accuracy

Hasil pengujian menunjukkan akurasi total sebesar 85%, dengan empat kelas yang dideteksi: Brown Spot (0.0), Healthy (1.0), Hispa (2.0), dan Leaf Blast (3.0). Data ini juga menampilkan nilai precision, recall, f1-score, dan support untuk masing-masing kelas. Hasil pengujian confusion matrix dapat dilihat pada Gambar 11.



Gambar 11. Hasil Confusion Matrix

Confusion matrix menunjukkan nilai actual dan predicted dengan rincian sebagai berikut: 45 dataset Brown Spot terdeteksi benar, 145 dataset Healthy terdeteksi benar, 36 dataset Hispa terdeteksi benar, dan 59 dataset Leaf Blast terdeteksi benar dari total 335 data yang digunakan untuk proses training dan validation.

KESIMPULAN

Berdasarkan pembahasan dan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa penyakit pada tanaman padi dapat diklasifikasikan menggunakan metode Convolutional Neural Network (CNN). Sistem yang dikembangkan menggunakan algoritma CNN dengan model ResNet50, dilengkapi dengan library PyTorch, dan Framework Flask. Sistem ini mampu mencapai akurasi 85% dalam mendeteksi jenis penyakit padi, yaitu brown spot, hispa, dan leaf blast. Sistem dilengkapi dengan history yang berguna sebagai tempat penyimpanan hasil deteksi. Fitur ini berguna agar user dapat mengakses hasil deteksi sebelumnya tanpa perlu mengunggah ulang gambar. Hasil deteksi juga dilengkapi dengan penjelasan tentang deskripsi dan atasan-langkah penanganan yang tepat sesuai dengan penyakit yang terdeteksi. Pengembangan selanjutnya dapat mencakup integrasi beberapa model CNN yang disesuaikan dengan dataset untuk meningkatkan akurasi, serta penambahan

masukannya jenis penyakit tanaman padi agar dapat mendeteksi penyakit tanpa atasan jenis dan klasifikasi, sehingga menjadi sistem yang lebih komprehensif dan berguna bagi petani.

DAFTAR PUSTAKA

- Azhar, K. M., Santoso, I., & Soetrisno, Y. A. A. (2021). Implementasi Deep Learning Menggunakan Metode Convolutional Neural Network Dan Algoritma Yolo Dalam Sistem Pendeteksi Uang Kertas Rupiah Bagi Penyandang Low Vision. *Transient J. Ilm. Tek. Elektro*, 10(3), 502-509.
- Benbrahim, H., Hachimi, H., & Amine, A. (2020). Deep convolutional neural network with tensorflow and keras to classify skin cancer images. *Scalable Computing: Practice and Experience*, 21(3), 379-390.
- Chatterjee, A., Roy, B., & all, e. (2022). Image Classification Using Keras. *IJRTI*.
- Damayanti, E., Vitianingsih, A. V., Kacung, S., Suhartoyo, H., & Maukar, A. L. (2024). Sentiment Analysis of Alfabeta Application User Reviews Using Long Short-Term Memory (LSTM) and Support Vector Machine (SVM) Methods. *Decode: Jurnal Pendidikan Teknologi Informasi*, 4(2), 509-521. <https://doi.org/10.51454/decode.v4i2.478>
- Felix, F., Wijaya, J., Sutra, S. P., Kosasih, P. W., & Sirait, P. (2020). Implementasi Convolutional Neural Network Untuk Identifikasi Jenis Tanaman Melalui Daun. *Jurnal SIFO Mikroskil*, 21(1), 1-10.
- Findawati. (2018). *Buku Ajar Rekayasa Perangkat Lunak*. Sidoarjo: Umsida Press.
- Guntara, R. G. (2023). Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning Yolov7. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 5(1), 55-60.
- Hastari, D., Winanda, S., Pratama, A. R., Nurhaliza, N., & Ginting, E. S. (2024). Application of Convolutional Neural Network ResNet-50 V2 on Image Classification of Rice Plant Disease. *Public Research Journal of Engineering, Data Technology and Computer Science*, 1(2).
- Husen, D., Kusriani, K., & Kusnawi, K. (2022). Deteksi Hama Pada Daun Apel Menggunakan Algoritma Convolutional Neural Network. *Jurnal Media Informatika Budidarma*, 6(4), 2103-2110.
- Irawan, F. A., Sudarma, M., & Khriane, D. C. (2021). Rancang Bangun Aplikasi Identifikasi Penyakit Tanaman Pepaya California Berbasis Android Menggunakan Metode Cnn Model Arsitektur Squeezenet. *Jurnal SPEKTRUM Vol*, 8(2).
- Jinan, A., & Hayadi, B. H. (2022). Klasifikasi Penyakit Tanaman Padi Menggunakan Metode Convolutional Neural Network Melalui Citra Daun (Multilayer Perceptron). *Journal of Computer and Engineering Science*, 37-44.
- Jonathan, A., & Wasito, I. (2023). Perancangan Aplikasi Pengenalan Aksara Jawa Digital Menggunakan Convolutional Neural Network dan Computer Vision. *Decode: Jurnal Pendidikan Teknologi Informasi*, 3(2), 364-377. <https://doi.org/10.51454/decode.v3i2.209>
- Kholilurrahman, M., Syaefi, W. A., & Nurhayati, O. D. (2023). Image Processing Classification of Rice Leaf Color Images Using the Convolutional Neural Network Method. *Jurnal Ilmiah Sains*, 175-186. <https://doi.org/10.35799/jis.v23i2.50415>
- Masykur, F., Setyawan, M. B., & Winangun, K. (2022). Optimalisasi Epoch Pada Klasifikasi Citra Daun Tanaman Padi Menggunakan Convolutional Neural Network (CNN) MobileNet t Epoch Optimization on Rice Leaf Image Classification Using Convolutional Neural Network (CNN) MobileNet. *no. July*, 581-590.

- Ngantung, R. K., & Pakereng, M. I. (2021). Model Pengembangan Sistem Informasi Akademik Berbasis User Centered Design Menerapkan Framework Flask Python. *Jurnal Media Informatika Budidarma*, 5(3), 1052-1062.
- Novendri, M. S., Saputra, A., & Firman, C. E. (2019). Aplikasi inventaris barang pada mts nurul islam dumai menggunakan php dan mysql. *lentera dumai*, 10(2).
- Nurani, D., Yanuar, I. L., & Putra, A. D. (2022). Klasifikasi Jenis Penyakit Pada Citra Daun Padi Menggunakan Algoritma Convolution Neural Network. *Jurnal Teknologi Informasi dan Komputer*, 8(3).
- Nurlita, I., & Anggraini, R. (2023). Analysis and Design of Incoming and Outgoing Cash Accounting Information Systems at Kilometer 28 Laundry using the Pieces and Waterfall Methods with Unified Modeling Language (Uml) Tools. *Formosa Journal of Applied Sciences*, 2(6), 1065-1090.
- Rahmawati, I., Abdussalaam, F., & Sari, I. (2023). Tata Kelola Rekam Medis Berbasis Elektronik Dalam Pengelolaan Pelaporan Instalasi Rawat Jalan Dengan Metode Waterfall. *Decode: Jurnal Pendidikan Teknologi Informasi*, 3(2), 310-321. <https://doi.org/10.51454/decode.v3i2.201>
- Rasywir, E., Sinaga, R., & Pratama, Y. (2020). Analisis dan Implementasi Diagnosis Penyakit Sawit dengan Metode Convolutional Neural Network (CNN). *Jurnal Khatulistiwa Informatika*, 22(2), 117-123.
- Sethy, P. K., Barpanda, N. K., Rath, A. K., & Behera, S. K. (2020). Image processing techniques for diagnosing rice plant disease: a survey. *Procedia Computer Science*, 167, 516-530.
- Sitompul, P., Okprana, H., & Prasetio, A. (2022). Identifikasi Penyakit Tanaman Padi Melalui Citra Daun Menggunakan DenseNet 201. *JOMLAI: Journal of Machine Learning and Artificial Intelligence*, 1(2), 143-150.
- Suryadewiansyah, M. K., & Tju, T. E. E. (2022). Naïve bayes dan confusion matrix untuk efisiensi analisa intrusion detection system alert. *Jurnal Nasional Teknologi dan Sistem Informasi*, 8(2), 81-88.
- Xu, J., Zhang, Y., & Miao, D. (2020). Three-way confusion matrix for classification: A measure driven view. *Information sciences*, 507, 772-794.