



Identifikasi Kesehatan Daun Tanaman Padi Menggunakan Klasifikasi Biner Sehat dan Tidak Sehat dengan Algoritma *Convolutional Neural Network* (CNN) Di Kabupaten Klaten

Shelvi Azizah^{1*}, Afu Ichsana Pradana², Dwi Hartanti³

^{1),2),3)}Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Duta Bangsa Surakarta
Jl. Bhayangkara No.55, Tipes, Kec. Serengan, Kota Surakarta, Jawa Tengah 57154
**email: 202020883@mhs.udb.ac.id*

(Naskah masuk: 04 Mei 2024; direvisi : 30 Juli 2024;diterima untuk diterbitkan: 05 Agustus 2024)

ABSTRAK – Tanaman padi merupakan tanaman pangan yang vital di Indonesia, di mana Klaten telah menjadi salah satu pemasok beras utama dengan capaian produksi sebesar 101 ribu ton pada tahun 2020. Namun, tantangan yang dihadapi adalah serangan penyakit seperti blast, hawar daun, dan layu bakteri yang dapat mengakibatkan kerugian besar dalam hasil panen jika tidak ditangani dengan efektif. Untuk mengatasi masalah ini, penelitian dilakukan dengan menggunakan *Convolutional Neural Network* (CNN), sebuah algoritma yang umumnya digunakan untuk pemrosesan gambar. Dalam penelitian ini, proses melibatkan dua tahap utama yaitu Fitur Ekstraksi dan Fully Connected Layer, dengan memanfaatkan dataset berupa 2400 gambar yang dikategorikan ke dalam kelas sehat dan tidak sehat. Hasil penelitian menunjukkan tingkat akurasi yang sangat tinggi, dengan akurasi tertinggi mencapai 0.9653 dan akurasi validasi mencapai 0.8125, serta loss yang rendah dengan jumlah epochs sebanyak 20. Melalui teknologi CNN, penelitian ini memberikan kontribusi penting dalam pemantauan kesehatan tanaman padi di Kabupaten Klaten, Indonesia, yang diharapkan dapat membantu meningkatkan produktivitas dan mengurangi kerugian hasil panen.

Kata Kunci – Tanaman padi; *Convolutional Neural Network*; Pemrosesan gambar; Deep Learning; Klaten.

Identification of Rice Plant Leaf Health Using Binary Classification of Healthy and Unhealthy with *Convolutional Neural Network* (CNN) Algorithm in Klaten District

ABSTRACT – Rice is a vital food crop in Indonesia, where Klaten has become one of the main rice suppliers with a production achievement of 101 thousand tons in 2020. However, the challenge faced is the attack of diseases such as blast, leaf blight, and bacterial wilt which can result in huge losses in yield if not handled effectively. To address this issue, research was conducted using *Convolutional Neural Network* (CNN), an algorithm commonly used for image processing. In this study, the process involved two main stages namely Feature Extraction and Fully Connected Layer, utilizing a dataset of 2400 images categorized into healthy and unhealthy classes. The results show a very high level of accuracy, with the highest accuracy reaching 0.9653 and validation accuracy reaching 0.8125, as well as low loss with a total of 20 epochs. Through CNN technology, this research makes an important contribution to monitoring the health of rice plants in Klaten Regency, Indonesia, which is expected to help increase productivity and reduce crop losses.

Keywords – Rice plant; *Convolutional Neural Network*; Image processing; Deep Learning; Klaten.

1. PENDAHULUAN

Tanaman padi merupakan salah satu tanaman pangan yang ada di Indonesia. Sebagai salah satu tanaman pangan utama, padi menjadi penghasil karbohidrat utama bagi sebagian besar penduduk di Indonesia. Selain itu, pertanian padi juga menciptakan lapangan pekerjaan bagi jutaan petani di seluruh negeri. Karena pentingnya peran padi dalam memenuhi kebutuhan pangan dan ekonomi, upaya untuk meningkatkan produktivitas dan kualitas panen padi selalu menjadi fokus utama dalam pembangunan pertanian di Indonesia.

Klaten telah menjadi salah satu pemasok beras utama di Indonesia, terbukti dengan kontribusi yang signifikan dalam sektor pertanian negara. Pada tahun 2020, capaian panen di Klaten mencapai surplus beras sebesar 101 ribu ton [1]. Kelebihan produksi ini tidak hanya mencerminkan kemampuan yang kuat dalam pertanian, tetapi juga berdampak positif pada ketahanan pangan dan ekonomi nasional. Klaten telah membuktikan dirinya sebagai salah satu daerah yang memainkan peran penting dalam memenuhi kebutuhan pangan bagi masyarakat Indonesia, serta memberikan kontribusi yang berarti dalam upaya mencapai kemandirian pangan negara.

Mengetahui penyakit yang menyerang tanaman padi memiliki peran yang sangat penting dalam menjaga produktivitas dan kesejahteraan petani. Tanaman padi rentan terhadap berbagai penyakit seperti blas, hawar daun, dan penyakit layu bakteri, yang dapat menyebabkan kerugian besar dalam hasil panen jika tidak diatasi dengan tepat. Dengan memahami gejala-gejala penyakit yang muncul, petani dapat mengambil langkah-langkah pencegahan dan pengendalian yang tepat, seperti penggunaan varietas tahan penyakit, pengaturan jadwal penyemprotan pestisida, atau tindakan budidaya lainnya [2]. Upaya ini penting untuk mencegah penyebaran penyakit dan meminimalkan kerugian yang timbul, sehingga memastikan kelangsungan produksi padi yang stabil dan hasil panen yang ideal bagi petani serta ketersediaan pangan bagi masyarakat.

Kecerdasan buatan merupakan ilmu yang ditujukan untuk melakukan pembuatan *software* dan *hardware* yang memiliki fungsi untuk melakukan pemikiran seperti manusia [3]. Salah satu contoh dari kecerdasan buatan adalah *computer vision*. *Computer vision* merupakan salah satu teknologi yang memungkinkan komputer mampu melihat dan mengenali objek yang ada di sekitarnya layaknya manusia [4]. Salah satu algoritma yang digunakan dalam *computer vision* yaitu *Convolutional Neural Network*. *Convolutional Neural Network* (CNN) adalah jenis neural network yang dirancang khusus untuk menangani masalah pengolahan citra [5]. Kelebihan

metode *Convolutional Neural Network* (CNN) yaitu dapat secara otomatis mengekstraksi karakteristik penting dari setiap gambar yang lebih berfungsi daripada seluruh gambar, sehingga proses deteksi menjadi lebih akurat [6]. Selain itu, CNN lebih efisien daripada metode *neural network* lainnya dalam hal memori dan kompleksitas. Salah satu aplikasi dalam *computer vision* adalah *image classification*. *Image classification* bertujuan untuk mengelompokkan citra ke dalam fitur-fitur yang terdapat pada citra tersebut [7].

Convolutional Neural Network (CNN) digunakan dalam *image classification* untuk mengklasifikasikan tanaman sehat dan tidak sehat berdasarkan gambar. CNN dapat ekstraksi fitur, pembelajaran representasi, klasifikasi, dan deteksi anomali, memungkinkan pengembangan sistem pemantauan otomatis untuk meningkatkan produksi pertanian dan mengurangi kerugian akibat penyakit tanaman.

Pada penelitian lain, dilakukan pembuatan model CNN untuk mengidentifikasi tanaman padi yang diklasifikasikan menjadi tiga jenis, yaitu blas, *brown spot*, dan hispa. Total data sampel yang digunakan adalah 900 citra, dengan masing-masing 300 citra pada setiap kelasnya. Hasil pengujian menunjukkan rata-rata akurasi sebesar 92% pada data latih dan 77% pada data validasi [8].

Selain itu, ada sebuah penelitian mengenai penggunaan CNN untuk mendeteksi penyakit pada citra daun tomat. Penelitian ini menggunakan 10 kelas dengan total 18.160 citra. Proses pembagian data yang diterapkan adalah 85% untuk data latih, 10% untuk data validasi, dan 5% untuk data uji. Tingkat akurasi yang dicapai adalah 94% dengan kesalahan sebesar 6%. Implementasi dilakukan dalam aplikasi mobile atau *Android* [9].

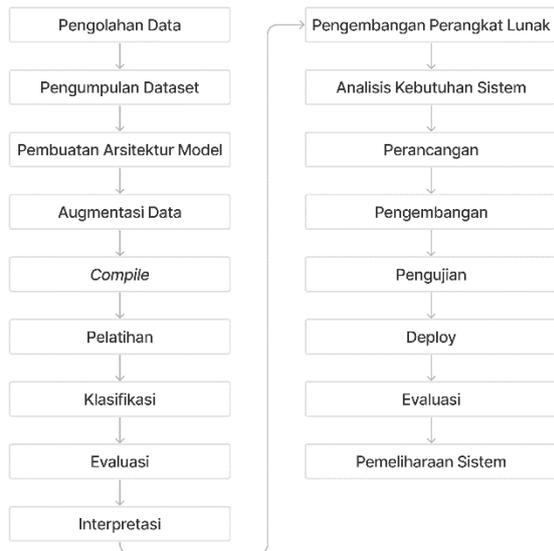
Sebuah penelitian juga dilakukan dengan menggunakan CNN dengan model arsitektur Inception V3. Jumlah dataset yang digunakan adalah 240 citra yang terbagi menjadi tiga kelas berdasarkan jenis penyakit pada daun tanaman padi, yaitu blas, hawar daun, dan tungro. Hasil pengujian menunjukkan akurasi sebesar 93,75% dengan nilai loss 0,3076 [10].

Pada penelitian ini, penulis menggunakan 2400 gambar daun tanaman padi dan optimasi Adam untuk membuat model dengan empat lapisan konvolusi dan tiga kernel RGB. Hasilnya, daun tanaman padi diklasifikasikan ke dalam dua kategori kelas, yaitu sehat dan tidak sehat, beserta tingkat akurasinya. Implementasi dilakukan di sebuah *website*.

2. METODE DAN BAHAN

Bab ini akan menjelaskan tentang tahapan penelitian mulai dari pengolahan data sampai dengan pengembangan perangkat lunak.

Pengolahan data terdiri dari pengumpulan dataset, pembuatan arsitektur model, augmentasi data, kompilasi, pelatihan (*training*), klasifikasi, evaluasi, dan interpretasi. Sedangkan untuk pengembangan perangkat lunak menggunakan metode agile yang terdiri dari analisis kebutuhan sistem, perancangan, pengembangan (*development*), pengujian (*testing*), *deploy*, evaluasi, dan pemeliharaan sistem [11]. Gambar 1 menunjukkan gambaran alur proses penelitian.



Gambar 1 Tahapan Penelitian

Pengumpulan Dataset

Kumpulan data yang biasa disebut dengan dataset sangat penting untuk melakukan klasifikasi, terutama dalam pengolahan gambar. Dataset merupakan kumpulan data yang didapatkan di *warehouse* yang mana di dalam dataset terdapat kumpulan informasi dari data pada masa lampau [12]. Tahapan ini biasanya disebut dengan *data preparation*. *Data preparation* adalah suatu proses atau langkah yang dilakukan untuk membuat data mentah menjadi data yang berkualitas [13]. Dalam kasus ini, dataset ini diperoleh dari *website* Kaggle dan terdiri dari 2.000 gambar dan 400 data citra dari daun tanaman padi yang diperoleh dari persawahan di area Kabupaten Klaten. Selanjutnya, dataset dibagi menjadi 2.000 gambar dari *website* Kaggle untuk data pelatihan (*training*), 200 gambar yang diperoleh dari persawahan di area Kabupaten Klaten untuk data pengujian (*testing*), dan 200 gambar yang diperoleh dari persawahan di area Kabupaten Klaten untuk data validasi (*validation*). Data pelatihan digunakan untuk mengajarkan model melakukan klasifikasi, data pengujian digunakan untuk menguji kinerja model secara objektif, dan data validasi digunakan untuk mengetahui seberapa baik model dapat diterapkan pada data baru. Oleh karena itu, pengelolaan dataset yang baik merupakan langkah

awal yang sangat penting untuk menjamin keberhasilan proses klasifikasi. Tabel 1 berisi data pembagian kelas daun tanaman padi. Gambar 2 menunjukkan contoh daun tanaman padi yang termasuk dalam kelas sehat (*healthy*) dan tidak sehat (*not healthy*).

Tabel 1. Pembagian Kelas Daun Tanaman Padi

No	Kelas	Jumlah
1	Healthy	1200
2	Not Healthy	1200

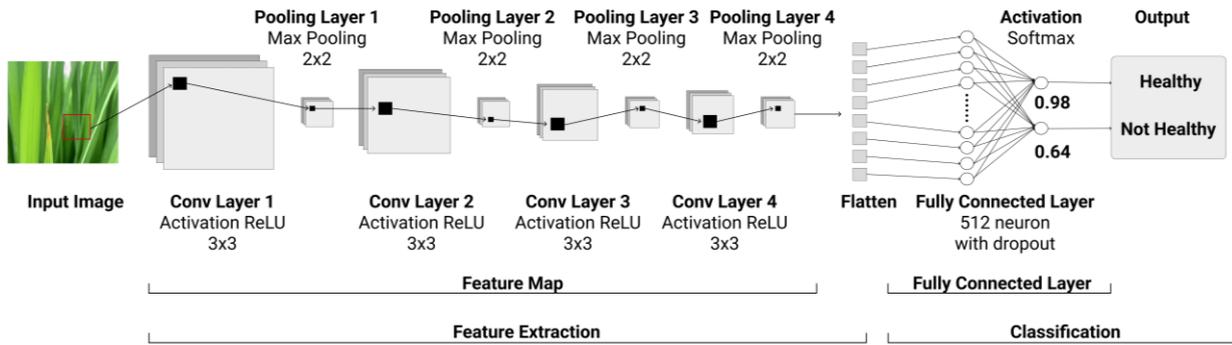


Gambar 2. Contoh Daun Tanaman Padi

Pembuatan Arsitektur Model

Pada tahapan ini dilakukan perancangan arsitektur model CNN. Model CNN paling sesuai jika digunakan untuk menyelesaikan permasalahan yang berkaitan dengan pengolahan data gambar. CNN Terdiri dari 2 lapisan arsitektur yaitu *feature learning/feature extraction* dan *classification* [14]. Dalam arsitektur yang dirancang, terdapat input dengan 3 channel RGB berukuran 150 x 150 pixel, diikuti oleh 4 lapisan konvolusi, *Max pooling*, *Flatten*, *Dense*, *Dropout*, dan *Dense (output)*. Gambar 3 menunjukkan rancangan arsitektur CNN yang akan diimplementasikan.

Penggunaan *TensorFlow* untuk membuat model CNN untuk melakukan klasifikasi gambar menggunakan Keras, ini adalah contoh pembuatan model menggunakan jenis model *Sequential*, yang merupakan tumpukan layer linear. *Tensorflow* adalah *library open source* untuk komputasi numerik dan *machine learning* skala besar [15]. Pada lapisan *Dense*, lapisan terhubung penuh dengan 512 neuron dan fungsi aktivasi ReLU. Model ini juga menggunakan *Dense* dengan lapisan terhubung akhir yang memiliki 2 unit neuron untuk *output* dengan fungsi aktivasi *softmax* untuk melakukan klasifikasi. Hasil yang dicapai oleh model sangat dipengaruhi oleh jumlah parameter yang ditentukan dalam arsitektur model. Setiap layer dan parameter yang ditentukan



Gambar 3 Arsitektur CNN

dalam arsitektur model dapat mempengaruhi kemampuan model untuk mengekstrak fitur dari data, mempelajari pola yang kompleks, dan dengan mudah menggeneralisasi data baru. Tabel 2 berisi *hyper parameter* yang diterapkan dalam model CNN.

Tabel 2. *Hyper Parameter* CNN

No	<i>Hyper Parameter</i>	Value
1	<i>Convolutional layer</i>	4
2	<i>Max pooling layer</i>	4
3	<i>Drop out rate</i>	0.5
4	<i>Activation function</i>	<i>Relu</i>
5	<i>Optimizer</i>	<i>Adam</i>
6	<i>Loss function</i>	<i>Categorical cross entropy</i>
7	<i>Learning rate</i>	0.001
8	<i>Epocs</i>	50
9	<i>Batch size</i>	32
10	<i>Crop size</i>	64
11	<i>Step per epocs</i>	9

Augmentasi Data

Salah satu teknik penting dalam proses pelatihan, validasi, dan pengujian model klasifikasi menggunakan *deep learning* adalah augmentasi data. Augmentasi data merupakan teknik yang dapat digunakan untuk mengurangi *overfitting* dengan meningkatkan ukuran dataset dalam upaya yang seminimum mungkin [16]. Singkatnya dengan data yang sedikit proses augmentasi akan menghasilkan lebih banyak variasi gambar. Pada penelitian ini terdapat beberapa augmentasi data yang diterapkan seperti merotasi acak gambar antara -10 hingga +10 derajat, menggeser gambar secara horizontal dan vertikal, melakukan pembesaran gambar antara 80-100%, membalikkan gambar secara acak, serta mengatur kecerahan gambar dalam rentang 50% hingga 100%. Tujuan dari augmentasi data adalah untuk meningkatkan variasi data yang digunakan dalam pelatihan model, sehingga model dapat belajar dari berbagai variasi gambar yang mungkin.

Compile

Sebelum melakukan pelatihan, akan dilakukan proses kompilasi terlebih dahulu. Pada tahap ini, model akan disiapkan untuk melakukan pelatihan dengan beberapa konfigurasi seperti *optimizer* dan fungsi kerugian. *Optimizer* yang digunakan untuk model ini adalah Adam dengan *learning rate* 0.001, serta *loss function* yang digunakan adalah *categorical cross entropy*. Optimizer Adam adalah algoritma yang merupakan perkembangan dari algoritma *Stochastic Gradient Descent (SGD)* klasik dimana bobot *network* telah diperbarui [17].

Pelatihan (Training)

Pelatihan digunakan untuk mengoptimalkan parameter model agar mereka dapat mempelajari pola dalam data pelatihan dan membuat prediksi yang akurat. Model akan belajar untuk menyesuaikan diri dengan pola-pola dalam data dengan melakukan iterasi (*epochs*) pada dataset berulang kali. Jumlah *epochs* dalam pelatihan dan *steps* juga mempengaruhi hasil dari akurasi dari pelatihan. Semakin banyak *epochs* yang dilakukan maka akan semakin banyak waktu yang dibutuhkan namun model akan belajar lebih baik. Selain itu, jumlah *batch size* juga mempengaruhi proses pelatihan. Semakin banyak *batch size* maka akan semakin lama proses pelatihan dilakukan.

Pada proses pelatihan ini, digunakan *batch size* sebesar 32 dan ukuran crop gambar sebesar 64 piksel. Selain itu, jumlah *epoch* yang digunakan adalah 50 dengan 9 langkah untuk pelatihan dan 1 langkah untuk validasi. Pada proses pelatihan akan tersedia riwayat (*history*) berupa akurasi, kerugian (*loss*), akurasi validasi, dan kerugian validasi.

Klasifikasi

Setelah proses pelatihan dilakukan, langkah selanjutnya adalah menyimpan model sehingga dapat diintegrasikan dengan *backend website*. Setelah itu, dilakukan proses klasifikasi untuk memprediksi kelas gambar. Klasifikasi adalah metode untuk menentukan suatu anggota ke dalam kelas tertentu yang telah ditetapkan sebelumnya [18]. Klasifikasi ini

dilakukan dengan memilih gambar yang akan diklasifikasikan dan mengubahnya menjadi *array* yang sesuai dengan format yang diharapkan oleh model. Selanjutnya, model akan melakukan prediksi kelas gambar dengan menentukan probabilitas setiap gambar terhadap kelas yang sudah dikategorikan. Kelas sebelumnya sudah dikategorikan menjadi 2, yaitu "Healthy" dan "Not healthy". Selanjutnya, nilai probabilitas tertinggi akan dicari untuk menafsirkan prediksi model terhadap gambar. Hasilnya kemudian dicetak untuk menampilkan kelas dan probabilitas dari label kelas tersebut.

Evaluasi

Evaluasi digunakan untuk memastikan bahwa model yang dibangun berfungsi dengan baik dan memenuhi kebutuhan. Evaluasi model klasifikasi dilakukan dengan menggunakan nilai *confusion matrix*, *precision*, *recall*, dan *F1 Score*. *Confusion matrix* adalah alat ukur berbentuk matrik yang digunakan untuk mendapatkan jumlah ketepatan klasifikasi terhadap kelas dengan algoritma yang dipakai [19]. *Confusion matrix* akan memberikan penilaian performa klasifikasi berdasarkan objek dengan benar atau salah dan menghasilkan nilai *accuracy*, *recall*, *precision* serta *F1 Score* [20]. Penyesuaian pada model akan terus dilakukan hingga ditemukan model yang memenuhi persyaratan dan kinerja terbaik.

Interpretasi

Setelah melakukan sejumlah evaluasi dan penyesuaian terhadap model, langkah selanjutnya adalah memilih model dengan performa dan kinerja terbaik untuk digunakan dalam pengembangan perangkat lunak selanjutnya. Model terbaik dipilih berdasarkan evaluasi nilai *confusion matrix*, *precision*, *recall*, dan *F1 Score*.

Analisis Kebutuhan Sistem

Analisis kebutuhan sistem pada tahapan pengembangan perangkat lunak sangatlah penting untuk memastikan bahwa kebutuhan pengguna dipenuhi dan untuk merinci secara detail perangkat lunak seperti apa yang dibutuhkan. Dengan melakukan analisis kebutuhan sistem, tim pengembangan dapat mengidentifikasi fungsionalitas, kinerja, keamanan, dan elemen lainnya yang diperlukan oleh perangkat lunak untuk mencapai tujuan bisnis dan memuaskan pengguna akhir.

Perancangan

Pada tahap perancangan maka akan dilakukan pembuatan desain sistem, yang mencakup arsitektur, antarmuka pengguna, dan aliran data, desain tampilan antarmuka dapat dibuat dengan menggunakan *tools* seperti Figma. Gambar 4 menunjukkan rancangan alur program yang dibuat.



Gambar 4. Rancangan Alur Program

Pengembangan (Development)

Pada tahap pengembangan, sistem akan dibangun sesuai dengan desain yang telah disiapkan menggunakan HTML, CSS, dan framework CSS. Selain itu, integrasi dengan model yang telah ada juga akan dilakukan untuk membuat *website* siap digunakan.

Pengujian (Testing)

Pengujian dilakukan untuk memastikan bahwa sistem berfungsi dengan baik dan sesuai dengan kebutuhan pengguna (*user requirement*). Pengujian perangkat lunak sangat penting untuk melihat kualitas dari *software* yang telah dibuat apakah ada kendala atau tidak dalam pengoperasiannya. Kegiatan pengujian ini akan dilakukan menggunakan *black box testing*. *Black box testing* merupakan metode pengujian untuk mengetahui kualitas dari fungsionalitas perangkat lunak [21]. Tabel 3 berisi skenario pengujian yang digunakan pada pengujian sistem dengan metode *black box*.

Tabel 3. Rencana Pengujian Website

ID	Pengujian	Hasil yang diinginkan
N1	Ketika mengklik tombol "Choose File" maka akan muncul tampilan untuk memilih file gambar yang akan digunakan	Muncul tampilan untuk memilih file gambar
N2	Ketika tombol "Prediksi" di klik maka akan menampilkan gambar yang dimasukkan	Menampilkan gambar yang akan dimasukkan

N3 Ketika tombol “Prediksi” di klik maka tampil hasil dari klasifikasi berupa label kelas dan akurasi. Menampilkan hasil dari klasifikasi berupa label kelas dan akurasi.

dengan bentuk *sequential*.

```
In [8]: model.summary()
```

Layer (type)	Output Shape	Param #
conv2d_4 (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d_4 (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_5 (Conv2D)	(None, 72, 72, 64)	18496
max_pooling2d_5 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_6 (Conv2D)	(None, 34, 34, 128)	73856
max_pooling2d_6 (MaxPooling2D)	(None, 17, 17, 128)	0
conv2d_7 (Conv2D)	(None, 15, 15, 256)	295168
max_pooling2d_7 (MaxPooling2D)	(None, 7, 7, 256)	0
Flatten_1 (Flatten)	(None, 12544)	0
dense_2 (Dense)	(None, 512)	6423040
dropout_1 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 2)	1026

 Total params: 6812482 (25.99 MB)
 Trainable params: 6812482 (25.99 MB)
 Non-trainable params: 0 (0.00 Byte)

Gambar 6. Rangkuman model CNN

Deploy

Setelah melakukan pengujian dan memastikan bahwa sistem bekerja dengan baik, tahap selanjutnya adalah *deployment*. Ada banyak cara untuk melakukan *deployment*, seperti menggunakan kontrol versi Git atau *web hosting*. Selain itu juga dapat menggunakan Python Anywhere ataupun vercel.

Evaluasi

Pengguna akan memberikan evaluasi dalam bentuk umpan balik, yang memberikan umpan balik tentang sistem yang telah dibuat. Jika ada umpan balik yang perlu diperbaiki, perbaikan akan dilakukan selama proses pemeliharaan perangkat.

Pemeliharaan Sistem

Siklus pemeliharaan sistem terdiri dari proses evaluasi yang menemukan kesalahan atau kekurangan dalam sistem. Setelah kesalahan ditemukan, langkah selanjutnya adalah melakukan pemeliharaan untuk memperbaikinya. Tujuan utama pemeliharaan sistem adalah untuk membuat sistem yang baik dan dapat digunakan oleh pengguna dengan kinerja, keandalan, dan fungsionalitas yang sesuai dengan kebutuhan mereka.

3. HASIL DAN PEMBAHASAN

Pada bagian ini akan dijelaskan hasil dan pembahasan dari penelitian yang sudah dilakukan. Pembahasan akan dibagi menjadi 2 yaitu pengolahan data dan pengembangan perangkat lunak. Hasil ini diperoleh dari penelitian yang sudah dilakukan.

1. Pengolahan data

```
Jumlah Data Train tiap kelas
Healthy: 1000 , Not healthy: 1000
Jumlah Data Test tiap kelas
Healthy: 100 , Not healthy: 100
Jumlah Data validasi tiap kelas
Healthy: 100 , Not healthy: 100
```

Gambar 5. Pembagian Dataset

Dataset telah berhasil dimuat dan dibagi menjadi data pelatihan (*training*), data pengujian (*testing*), dan data validasi (*validation*). Gambar 5 menunjukkan pembagian data citra. Selanjutnya, proses pembuatan model untuk melakukan pembelajaran terhadap data citra daun tanaman padi. Gambar 6 menunjukkan rangkuman dari model yang dibuat. Model dibuat menggunakan algoritma *deep learning* yaitu CNN

Setelah pembuatan arsitektur model, langkah selanjutnya adalah augmentasi data gambar. *Crop size* dengan ukuran 148 diterapkan, dan beberapa fungsi augmentasi data digunakan untuk membuat gambar lebih bervariasi, sehingga model dapat mempelajari data gambar dengan lebih baik. *Generator* juga diterapkan untuk memproses data gambar, dan kemudian data dibagi ke dalam kelas kategori menggunakan *categorical cross entropy*. Gambar 7 menunjukkan jumlah data citra dari setiap kelas.

```
Found 2000 images belonging to 2 classes.
Found 200 images belonging to 2 classes.
Found 200 images belonging to 2 classes.
```

Gambar 7. Pembagian ke dalam kelas kategori

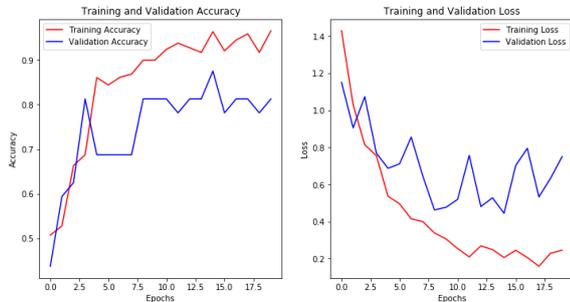
Sebelum proses *training* dimulai, model harus di *compile* terlebih dahulu. Optimizer Adam dengan learning rate 0.001 dan fungsi *loss* menggunakan *categorical cross entropy* serta *matrix accuracy* diterapkan. Optimizer Adam menghasilkan nilai akurasi yang lebih baik dibandingkan dengan SGD karena adaptif mempelajari bobot [22]. Setelah itu, proses *training* dilakukan dengan *batch size* sebesar 32 untuk 20 *epochs*, dengan 9 langkah per *epoch* untuk *training* dan 1 langkah untuk validasi. Hasil tertinggi dari akurasi *training* terjadi pada *epoch* ke-20, dengan nilai *loss* sebesar 0.2458 dan akurasi sebesar 0.9653. Sedangkan pada data validasi, terdapat nilai *loss* sebesar 0.7503 dan akurasi sebesar 0.8125. Gambar 8 menunjukkan informasi lengkap mengenai hasil

pelatihan.

```
Epoch 11/20
9/9 [=====] - 76s 8s/step - loss: 0.2543 - accuracy: 0.9236 - val_loss: 0.5199 - val_accuracy: 0.8125
Epoch 12/20
9/9 [=====] - 77s 9s/step - loss: 0.2096 - accuracy: 0.9375 - val_loss: 0.7566 - val_accuracy: 0.7812
Epoch 13/20
9/9 [=====] - 78s 9s/step - loss: 0.2687 - accuracy: 0.9271 - val_loss: 0.4811 - val_accuracy: 0.8125
Epoch 14/20
9/9 [=====] - 76s 8s/step - loss: 0.2491 - accuracy: 0.9167 - val_loss: 0.5281 - val_accuracy: 0.8125
Epoch 15/20
9/9 [=====] - 76s 8s/step - loss: 0.2033 - accuracy: 0.9632 - val_loss: 0.4446 - val_accuracy: 0.8750
Epoch 16/20
9/9 [=====] - 78s 9s/step - loss: 0.2453 - accuracy: 0.9281 - val_loss: 0.7019 - val_accuracy: 0.7812
Epoch 17/20
9/9 [=====] - 76s 8s/step - loss: 0.2064 - accuracy: 0.9444 - val_loss: 0.7948 - val_accuracy: 0.8125
Epoch 18/20
9/9 [=====] - 77s 9s/step - loss: 0.1592 - accuracy: 0.9583 - val_loss: 0.5327 - val_accuracy: 0.8125
Epoch 19/20
9/9 [=====] - 77s 9s/step - loss: 0.2292 - accuracy: 0.9167 - val_loss: 0.6335 - val_accuracy: 0.7812
Epoch 20/20
9/9 [=====] - 77s 9s/step - loss: 0.2458 - accuracy: 0.9653 - val_loss: 0.7583 - val_accuracy: 0.8125
```

Gambar 8. Hasil Proses Pelatihan

Model kemudian disimpan untuk proses integrasi dengan perangkat lunak. Setelah itu, hasil dari proses *training* ditampilkan dalam bentuk grafik untuk mengetahui korelasi antara seberapa baik model yang telah dibuat. Gambar 9 menampilkan grafik hasil dari *training* yang dilakukan. Dari grafik tersebut, terlihat bahwa model sudah cukup baik. Gambar 10 memperlihatkan hasil prediksi dan label yang benar beserta dengan tingkat akurasi.



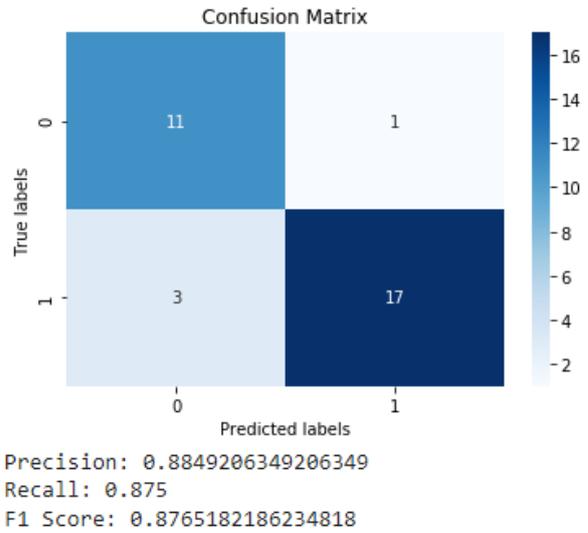
Gambar 9. Grafik Loss dan Accuracy



Gambar 10. Hasil Prediksi

Langkah terakhir adalah melakukan evaluasi menggunakan *confusion matrix*. Dari *confusion matrix*, kita dapat melihat hubungan antara *True Label* (label sebenarnya) dan *Predicted Label* (label yang diprediksi oleh model). Selain itu, nilai *Precision* sebesar 0.885, *Recall* sebesar 0.875, dan *F1 Score* mencapai nilai 0.876, yang menunjukkan bahwa model memiliki kinerja yang sempurna dalam mengklasifikasikan data. Ini menandakan bahwa model mampu mengidentifikasi semua *instance positif* dengan benar (*precision*), mendeteksi semua *instance positif* yang seharusnya ditemukan (*recall*), dan memiliki keseimbangan yang baik antara *precision* dan *recall* (*F1 Score*). Dengan nilai-nilai ini, kita dapat percaya pada kemampuan model untuk melakukan prediksi

dengan akurat. Gambar 11 menunjukkan *confusion matrix*.



Gambar 11. Confusion Matrix

2. Pengembangan perangkat lunak

Pada pembahasan sebelumnya telah dijelaskan mengenai pemrosesan data sampai dengan evaluasi. Pada pembahasan ini maka akan dijelaskan mengenai pengembangan perangkat lunak. Model yang sebelumnya sudah disimpan maka akan diintegrasikan dengan tampilan *website* dengan menggunakan *html* dan *tailwind css*. Gambar 12 menunjukkan halaman awal tampilan *website*.



Gambar 12. Halaman awal website

Selanjutnya pilih file dengan klik tombol "Choose file". Gambar 13 menunjukkan tampilan untuk memilih file.



Gambar 13. Memilih file gambar

Gambar 14 memperlihatkan tampilan setelah file dipilih. Terdapat nama file tertera pada tampilan *website*.



Gambar 14. Setelah memilih file gambar

File telah dipilih dan tombol "Prediksi" telah diklik untuk melakukan prediksi. Gambar yang dimasukkan akan ditampilkan bersama dengan prediksi dan tingkat akurasi. Gambar 15 menunjukkan hasil jika prediksi sehat (*healthy*). Gambar 16 menunjukkan hasil jika prediksi tidak sehat (*not healthy*). Proses ini melibatkan analisis data dari gambar yang dipilih untuk menghasilkan prediksi yang tepat dan menentukan tingkat akurasi dari prediksi tersebut.



Gambar 15. Hasil prediksi *healthy*



Gambar 16. Hasil prediksi *not healthy*

Pengujian pada implementasi website ini menggunakan metode *black box*. Tabel 4 menjelaskan hasil pengujian dengan menampilkan ID, hasil yang didapatkan, dan keterangan pengujian (sesuai atau tidak sesuai). Berdasarkan hasil tersebut, dapat disimpulkan bahwa semua fungsi sudah sesuai dan dapat digunakan.

Tabel 4. Hasil Pengujian Website

ID	Hasil yang didapatkan	Keterangan
H1	Berhasil memunculkan tampilan untuk memilih file gambar	Sesuai
H2	Berhasil menampilkan gambar yang dimasukkan ketika	Sesuai

tombol "Prediksi" di klik
 H3 Berhasil menampilkan Sesuai hasil dari klasifikasi berupa label kelas dan akurasi.

4. KESIMPULAN

Dari penelitian yang dilakukan, dapat disimpulkan bahwa penggunaan metode *Convolutional Neural Network* (CNN) dapat digunakan untuk melakukan prediksi terhadap penyakit pada daun tanaman padi. Dengan menggunakan 2400 data citra beserta augmentasi, performa model yang dibuat dapat ditingkatkan. Pemilihan parameter-parameter tertentu sangat mempengaruhi hasil prediksi. Penggunaan aktivasi *Softmax* pada layer terakhir dengan *loss categorical cross entropy* terbukti cukup efektif. Hasil dari model menunjukkan akurasi tinggi dengan nilai *loss* sebesar 0.2458 dan akurasi sebesar 0.9653. Pada data validasi, terdapat nilai *loss* sebesar 0.7503 dan akurasi sebesar 0.8125. Evaluasi *Precision* sebesar 0.885, *Recall* sebesar 0.875, dan *F1 Score* mencapai nilai 0.876, menandakan bahwa model mampu melakukan prediksi dengan sangat baik dan dapat diimplementasikan dengan baik pada *website*.

DAFTAR PUSTAKA

- [1] y. p. jateng, "Klaten Surplus Beras 101 Ribu Ton," 30 Juli 2020. [Online]. Available: <https://jatengprov.go.id/beritadaerah/klaten-surplus-beras-101-ribu-ton/>.
- [2] M. A. V. P. Syaikhul Anam Alidrus, "Deteksi Penyakit Pada Daun Tanaman Padi Menggunakan Metode Convolutional Neural Network," *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)*, no. 9, pp. 103-109, 2021.
- [3] A. I. P. L. Dwi Hartanti, "Komprasi Algoritma Decision Tree, SVM dan ANN untuk Reservasi Hotel," *Duta.com (Jurnal Ilmiah Teknologi Informasi dan Komunikasi)*, vol. 16, pp. 21-27, 2023.
- [4] S. T. R. A. H. Joko Subur, "Pemanfaatan Teknologi Computer Vision untuk Deteksi Ukuran Ikan Bandeng dalam Membantu Proses Sortir Ikan," *CYCLOTRON : Jurnal Teknik Elektro*, pp. 52-60, 2024.
- [5] R. A. P. ANHAR, "Perancangan dan Implementasi Self-Checkout System pada Toko Ritel menggunakan Convolutional Neural Network (CNN)," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, pp. 466 - 478, 2023.

- [6] R. W. A. Afu Ichsan Pradana, "Deteksi Ketepatan Penggunaan Masker Wajah Dengan Algoritma CNN Dan Haar Cascade," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 9, pp. 2305-2316, 2022.
- [7] R. A. P. Anhar, "Perancangan dan Implementasi Self-Checkout System pada Toko Ritel menggunakan Convolutional Neural Network (CNN)," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 11, no. 2, pp. 466-478, 2023.
- [8] M. A. V. P. Syaikhul Anam Alidrus, "Deteksi Penyakit Pada Daun Tanaman Padi Menggunakan Metode Convolutional Neural Network," *SENAMIKA*, pp. 103-109, 2021.
- [9] D. P. S. Chrisno R. Kotta, "Implementasi Metode Convolutional Neural Network untuk Mendeteksi Penyakit pada Citra Daun Tomat," *Jurnal Pekommas*, vol. 7, no. 2, pp. 123-132, 2022.
- [10] M. K. A. B. S. F. R. P. P. S. Syenira Sheila, "Deteksi Penyakit Pada Daun Padi Berbasis Pengolahan Citra Menggunakan Metode Convolutional Neural Network (CNN)," *JURNAL MULTINETICS*, vol. 9, no. 1, pp. 27-34, 2023.
- [11] A. F. M. B. Suhari1, "Sistem Informasi Kepegawaian Menggunakan Metode Agile Development di CV. Angkasa Raya," *Jurnal Teknologi dan Informasi (JATI)*, vol. 12, pp. 30-44, 2022.
- [12] I. P. F. I. Siti Emalia Saqila, "Analisis Perbandingan Kinerja Clustering Data Mining Untuk Normalisasi Dataset," *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 5, no. 2, pp. 356-365, 2023.
- [13] A. I. P. Dwi Hartanti, "Komparasi Algoritma Machine Learning dalam Identifikasi Kualitas Air," *SMARTICS Journal*, vol. 9, pp. 1-6, 2023.
- [14] S. D. Khairul Azmi, "Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat," *Jurnal Unitek*, vol. 16, no. 1, pp. 28-40, 2023.
- [15] A. S. Risnaldy Fatwa Muharram, "Implementasi artificial intelligence untuk deteksi masker secara realtime dengan tensorflow dan ssdmobilenet Berbasis python," *Jurnal Widya*, vol. 3, pp. 281-290, 2022.
- [16] A. I. S. Riestiya Zain Fadillah, "Data Augmentasi Untuk Mengatasi Keterbatasan Data Pada Model Penerjemah Bahasa Isyarat Indonesia (BISINDO)," *JURNAL INFORMATIKA*, vol. 8, pp. 208-214, 2021.
- [17] H. B. H. Y. P. A. T. Y. P. Naim Rochmawati, "Analisa Learning rate dan Batch size Pada Klasifikasi Covid Menggunakan Deep learning dengan Optimizer Adam," *JIEET (Journal Information Engineering and Educational Technology)*, vol. 5, pp. 44-48, 2021.
- [18] A. N. F. S. E. N. Nabilla Yolanda Paramitha, "Klasifikasi Penyakit Stroke Menggunakan Metode Naïve Bayes," *Jurnal Siger Matematika*, vol. 4, pp. 11-16, 2023.
- [19] A. S. A. Laila Qadrini, "Decision Tree Dan Adaboost Pada Klasifikasi Penerima Program Bantuan Sosial," *Jurnal Inovasi Penelitian*, vol. 7, 2021.
- [20] A. I. P. Wijiyanto, "Perbandingan Data Untuk Memprediksi Ketepatan Studi Berdasarkan Atribut Keluarga Menggunakan machine Learning," *JIKA (Jurnal Informatika) Universitas Muhammadiyah Tangerang*, vol. 8, pp. 221-228, 2024.
- [21] M. W. A. Yahya Dwi Wijaya, "Pengujian Blackbox Sistem Informasi Penilaian Kinerja Karyawan Pt Inka (Persero) Berbasis Equivalence Partitions," *Jurnal Digital Teknologi Informasi*, vol. 4, no. 1, pp. 22-26, 2021.
- [22] W. Afu Ichsan Pradana, "Identifikasi Jenis Kelamin Otomatis Berdasarkan Mata Manusia Menggunakan Convolutional Neural Network (CNN) dan Haar Cascade Classifier," *G-Tech : Jurnal Teknologi Terapan*, vol. 8, pp. 502-511, 2024.
- [23] Andika, "Studi Literatur: Pengujian Perangkat Lunak," *Jurnal Minfo Polgan*, vol. 9, pp. 16-20, 2020.