

# **SISTEM KLASIFIKASI PENYAKIT TANAMAN PADI MENGGUNAKAN MACHINE LEARNING PADA PLATFORM ANDROID**

**Ilham Widhi Nugroho; Dedi Gunawan**  
**Program Studi Teknik Informatika, Fakultas Komunikasi dan**  
**Informatika, Universitas Muhammadiyah Surakarta**

## **Abstrak**

Padi menjadi tanaman penghasil bahan makanan pokok bagi masyarakat Indonesia, tetapi produksi padi di Indonesia terus mengalami penurunan dari tahun ke tahun. Penurunan produksi disebabkan oleh berbagai factor, salah satunya karena tidak stabilnya cuaca yang menyebabkan peningkatan potensi penyakit tanaman padi. Dengan perkembangan teknologi informasi saat ini, terdapat banyak cara untuk mendeteksi secara dini penyakit – penyakit yang ada pada tanaman padi, salah satunya menggunakan teknologi *machine learning* yang diimplementasikan pada platform android. Algoritma yang digunakan adalah CNN (*Convolutional Neural Network*) dan berhasil mendapatkan tingkat akurasi 83,49%. Pada penelitian ini model SDLC yang digunakan adalah *waterfall*. Pengujian *black box* berjalan dengan baik dibuktikan dengan semua menu, tampilan, dan tombol pada aplikasi telah berjalan dengan sesuai fungsinya. Aplikasi yang dirancang telah dapat membantu petani dalam mengidentifikasi penyakit tanaman padi didasarkan pada survey SUS (*System Usability Scale*) yang mendapatkan skor rata-rata 77,5 dari 30 responden yang artinya sistem ini layak digunakan dan dapat diterima oleh pengguna.

**Kata Kunci:** android, *black box*, *machine learning*, penyakit tanaman padi.

## **Abstract**

*Rice is a staple food crop for Indonesian people, but rice production in Indonesia continues to decline from year to year. The decline in production was caused by various factors, one of which was unstable weather which caused an increase in the potential for rice plant diseases. With the current development of information technology, there are many ways to detect diseases in rice plants early, one of which is using machine learning technology implemented on the Android platform. The algorithm used is CNN (Convolutional Neural Network) and succeeded in getting an accuracy rate of 83.49%. In this research, the SDLC model used is waterfall. The black box test went well as evidenced by all the menus, displays and buttons in the application running according to their function. The application designed can help farmers identify rice plant diseases based on the SUS (System Usability Scale) survey which received an average score of 77.5 from 30 respondents, which means this system is suitable for use and can be accepted by users.*

**Keywords:** android, *black box*, *machine learning*, rice disease.

## 1. PENDAHULUAN

Pertanian merupakan suatu aktivitas yang memanfaatkan sumber daya alam yang dikelola oleh manusia untuk menghasilkan bahan pangan. Masyarakat bergantung pada pertanian untuk memenuhi kebutuhannya. Pertanian telah dianggap sebagai salah satu sektor yang dapat memberikan lapangan pekerjaan dan menghasilkan bahan makanan pokok bagi masyarakat. Salah satu tanaman utama yang ditanam dalam sektor pertanian di Indonesia adalah tanaman padi.

Kebanyakan negara Asia mengkonsumsi beras sebagai sumber karbohidrat utama mereka. Negara-negara lain seperti di Eropa, Australia, dan Amerika mengkonsumsi beras dalam jumlah yang lebih sedikit. Meskipun Indonesia merupakan negara produsen beras besar dunia, tetapi faktanya Indonesia masih mengimpor beras dari negara lain untuk memenuhi kebutuhan stok beras (Ariska & Qurniawan, 2021). Produksi beras dari Januari – April 2024 diperkirakan akan mencapai 10,71 juta ton beras dimana mengalami penurunan sebesar 2,28 juta ton atau 17,52% dibandingkan dengan produksi beras pada Januari – April 2023 yang sebesar 12,98 juta ton beras (Badan Pusat Statistik, 2024). Produksi beras yang kurang bisa disebabkan oleh banyak faktor, salah satunya adalah tidak stabilnya iklim seperti suhu tinggi yang berlangsung terlalu lama (Jaisyurahman et al., 2020). Tidak stabilnya iklim dan curah hujan yang terjadi dapat meningkatkan potensi penyakit pada tanaman padi. Terdapat banyak jenis penyakit pada tanaman padi yang menyebabkan turunnya kuantitas dan kualitas hasil panen tanaman padi. Penyakit – penyakit pada tanaman padi dapat disebabkan oleh hama, bakteri, ataupun jamur yang dapat diidentifikasi melalui daun atau malainya. Identifikasi penyakit pada tanaman padi secara dini diperlukan agar dapat meningkatkan kualitas dan kuantitas hasil panen, namun proses identifikasi secara manual dinilai kurang efisien, memakan waktu dan biaya (Zhou et al., 2019).

Di era sekarang yang serba digital, identifikasi penyakit pada tanaman dapat dilakukan dengan memanfaatkan teknologi *machine learning* yang menggunakan citra sebagai data olahan. Penggunaan teknologi *machine learning* dapat menghasilkan model yang dapat mengklasifikasikan jenis penyakit berdasarkan kumpulan citra dari penyakit tanaman padi pada daun ataupun buah. Petani di Indonesia kebanyakan berasal dari masyarakat yang belum menempuh pendidikan tinggi sehingga mereka tidak memiliki pengetahuan untuk mengidentifikasi suatu penyakit pada tanaman padi. Maka dari itu pada penelitian ini penulis akan mengimplementasikan model *machine learning* untuk mengklasifikasikan penyakit tanaman padi pada *smartphone* berbasis android yang berfokus pada penyakit *bacterial leaf blight*, *bacterial leaf streak*, *bacterial panicle blight*, *blast*, *brown spot*, *dead heart*, *downy*

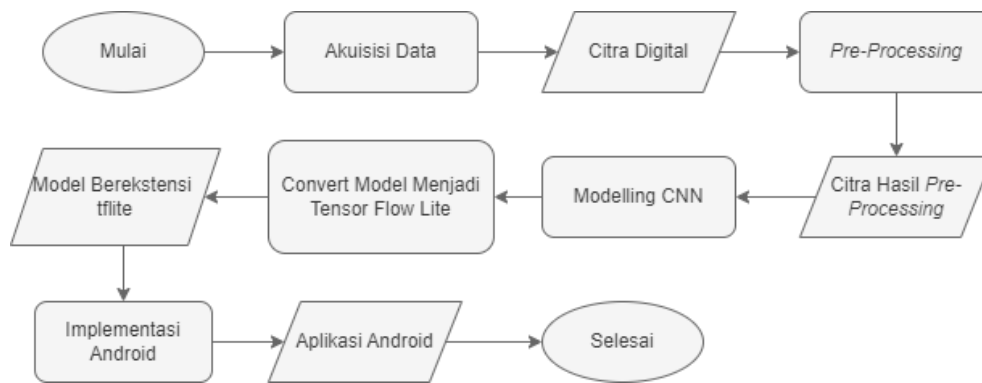
*mildew, hispa, normal, dan tungro.*

Sistem klasifikasi penyakit tanaman padi ini digunakan untuk mengklasifikasikan penyakit tanaman padi dari *input* berupa citra dan akan menentukan inputan citra tersebut termasuk ke dalam jenis penyakit apa. Dalam penelitian ini akan digunakan algoritma CNN (*Convolutional Neural Network*) untuk melatih model menggunakan dataset berupa citra dari penyakit tanaman padi yang bersumber dari kaggle (Paddy Doctor et al., 2022). Berikut beberapa penelitian terdahulu yang menjadi acuan: Penelitian yang dilakukan (Santosa et al., 2023) menunjukkan bahwa algoritma CNN memungkinkan untuk mengklasifikasi tiga jenis penyakit daun padi dengan akurasi akhir sebesar 99,66%. Penelitian lain yang dilakukan (Rino, 2023) menunjukkan bahwa model yang dibuat dari algoritma CNN dapat diimplementasikan pada android dan mendapatkan akurasi sebesar 91%. Penelitian lain yang dilakukan oleh (Naufal, 2021) menunjukkan bahwa algoritma CNN memiliki performa terbaik untuk mengklasifikasikan citra cuaca dengan akurasi 0.942, precision 0.943, recall 0.942, dan F1 Score 0.942 dibandingkan dengan algoritma SVM (Support Vector Machine) dan KNN (K-Nearest Neighbors).

Tujuan sistem klasifikasi penyakit tanaman padi ini adalah untuk membantu para petani mengenali penyakit yang menjangkit tanaman padi sehingga dapat dilakukan penanganan terhadap penyakit dengan cara yang sesuai sehingga tidak membebani petani karena pembelian obat yang tidak sesuai dengan penyakit yang menjangkit tanaman padi. Hasil akhir penelitian ini berupa sebuah aplikasi berbasis android yang dapat mendeteksi penyakit tanaman padi.

## **2. METODE**

Pada penelitian ini akan dilakukan beberapa langkah untuk dapat mengembangkan aplikasi android yang dapat digunakan untuk mendeteksi penyakit pada tanaman padi diantaranya yaitu akuisisi data berupa citra digital, kemudian data tersebut dilakukan pre-processing, lalu akan disusun arsitektur model CNN, kemudian model CNN yang sudah terlatih akan dikonversi menjadi bentuk tflite yang akan diimplementasikan pada android menggunakan Android Studio. Berikut merupakan diagram alur sistem yang akan dibuat :



Gambar 1. Diagram Alur Sistem

## 2.1 Akuisisi Data

Pada tahap akuisisi data, dataset diperoleh dari situs [www.kaggle.net](https://www.kaggle.net) (Paddy Doctor et al., 2022). Dataset yang akan digunakan adalah sebanyak 10453 citra gambar yang terdiri dari 9 kelas penyakit tanaman padi, yaitu penyakit *bacterial leaf blight*, *bacterial leaf streak*, *bacterial panicle blight*, *blast*, *brown spot*, *dead heart*, *downy mildew*, *hispa*, *tungro*, dan satu kelas tanaman padi normal.

## 2.2 Pre-Processing

### A. Data Sampling

Sebelum model CNN dilatih, dataset penyakit tanaman padi akan diambil sampelnya terlebih dahulu dengan membaginya menjadi *training sample data* dan *testing sample data*. Dataset penyakit tanaman padi dibagi menjadi 80% data training dan 20% data testing untuk menentukan pengaruh dari jumlah data training terhadap akurasi yang dihasilkan dengan mengabaikan data testing (Anggoro, 2020).

### B. Normalization Data

Pada tahap *pre-processing* selanjutnya akan dilakukan normalisasi untuk memastikan bahwa setiap parameter input pada setiap piksel memiliki distribusi data yang serupa dengan cara rescale atau mengatur ulang skala gambar dengan membagi setiap piksel dengan 255, sehingga nilai data yang akan diproses akan dinormalisasi dan berada pada kisaran nol hingga satu.

## 2.3 Convolutional Neural Network (CNN)

CNN merupakan bagian dari Artificial Neural Network (ANN) yang memiliki kemampuan mendeteksi informasi dengan akurasi tinggi. CNN merupakan jaringan *multilayer* atau biasa disebut *fully-connected layer*, sehingga *output* dari satu lapisan akan menjadi *input* dari lapisan berikutnya. Umumnya terdiri dari *layer input*, *hidden layer*, dan *layer output* (Listyarini & Anggoro, 2021). *Feature extraction* adalah

tahapan untuk mengekstrak fitur informatif yang penting untuk pengurangan dimensi. *Feature extraction* membantu mengurangi jumlah data yang berlebihan dari dataset; meningkatkan kecepatan pembelajaran; dan mengurangi jumlah data yang dibutuhkan untuk mendeskripsikan data dalam jumlah besar (Veeramreddy et al., 2024). Pada tahap *feature extraction*, terdapat empat lapisan konvolusi, *fully connected layer*, dan satu aktivasi *softmax*. Setiap lapisan konvolusi menggunakan aktivasi ReLU (*Rectified Linear Units*) dan melakukan proses *max pooling*, yang selanjutnya masuk ke tahap klasifikasi.

#### 2.3.1 Lapisan Konvolusi

Lapisan konvolusi adalah tahap pertama pemrosesan data. Lapisan konvolusi berguna untuk melakukan konvolusi berulang pada gambar yang dimasukkan dan menghasilkan filter yang didasarkan pada ukuran piksel (Anggoro et al., 2023).

#### 2.3.2 Aktivasi ReLU

ReLU menjaga hasil citra proses konvolusi pada domain yang ditentukan positif. Ini berarti bahwa setiap nilai negatif yang dihasilkan dari proses konvolusi akan melalui proses ReLU dan menjadikan nilai negatif tersebut sama dengan nol (Santosa et al., 2023).

#### 2.3.3 *Pooling Layer*

*Pooling layer* adalah proses mengurangi ukuran gambar untuk meningkatkan posisi fitur agar bervariasi. *Max pooling* adalah teknik yang paling umum digunakan pada algoritma CNN. *Max pooling* bekerja dengan membagi output lapisan konvolusi menjadi beberapa *grid*, dan setiap *filter* pergeseran menerima nilai maksimum untuk menggantikan matriks yang direduksi (Anggoro et al., 2023).

#### 2.3.4 *Fully Connected Layer*

Tahap akhir dalam algoritma CNN adalah *fully connected layer*, di mana neuron dari lapisan sebelumnya terhubung sepenuhnya dengan lapisan berikutnya. Setelah proses *flatten* yang mengubah data menjadi satu dimensi, kemudian *fully connected layer* akan bertanggung jawab atas pengolahan data untuk klasifikasi (Anggoro et al., 2023).

#### 2.3.5 Aktivasi *Softmax*

*Softmax* adalah fungsi yang mengambil vektor real  $K$  sebagai input dan dinormalisasi menjadi distribusi probabilitas yang terdiri dari  $K$  probabilitas. Sebelum softmax digunakan, beberapa komponen vektor dapat bernilai negatif atau

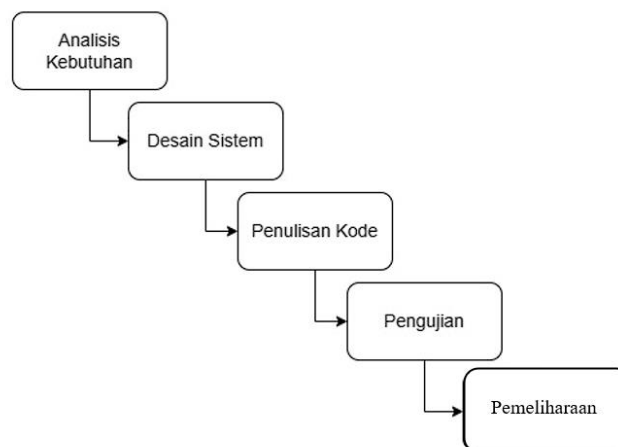
lebih besar dari satu, tetapi setelah softmax digunakan, setiap komponen akan berada dalam interval nol hingga satu, yang membuatnya dapat dianggap sebagai probabilitas (Anggoro et al., 2023).

#### 2.4 Convert Model

Setelah model CNN berhasil dilatih dan mendapatkan tingkat akurasi yang diinginkan maka model dapat dikonversi menjadi model *Tensor Flow Lite* berekstensi tflite yang kemudian digunakan untuk mengembangkan aplikasi android pada Android Studio.

#### 2.5 Implementasi Android

Pada tahap pengembangan aplikasi android akan digunakan metodologi SDLC (*System Development Life Cycle*). SDLC adalah metodologi umum yang digunakan untuk pengembangan sistem atau perangkat lunak. Metode SDLC menggunakan pendekatan sistematis untuk menjaga kualitas perangkat lunak dari waktu ke waktu. Pada penelitian ini model SDLC yang digunakan adalah *waterfall*. Kelebihan model *Waterfall*, yang digunakan dalam penelitian ini, adalah sebagai berikut: kebutuhan sistem dijelaskan dengan jelas sehingga mudah dipahami dan dikelola; penyimpangan dapat diidentifikasi sejak awal; serta proses dan hasilnya didokumentasikan dengan baik (Khan et al., 2020). Alur dari *waterfall* dapat dilihat pada Gambar 2.



Gambar 2. Diagram alur *waterfall*

##### 2.5.1 Analisis Kebutuhan

Analisa kebutuhan merupakan tahap pertama dalam model *waterfall*, pada tahap ini dilakukan perencanaan dan pengumpulan apa saja yang diperlukan dalam pengembangan. Pada penelitian ini metode yang digunakan untuk mengumpulkan data adalah wawancara dengan petani dan melakukan observasi langsung di sawah. Proses pengumpulan kebutuhan dilaksanakan dengan seksama untuk menentukan

spesifikasi yang cocok dengan kebutuhan perangkat lunak. Sementara itu, analisis juga mempertimbangkan perangkat keras yang dimiliki oleh petani agar sistem yang dirancang tidak memberatkan pengguna. (Gunawan et al., 2021). Ada dua jenis kebutuhan, yaitu fungsional dan non-fungsional. Kebutuhan fungsional mencakup semua fungsi yang harus dilakukan oleh sistem untuk mencapai tujuannya. Sedangkan kebutuhan non-fungsional mengacu pada batasan yang diperlukan untuk sistem beroperasi dengan baik (Mawan & Anggoro, 2021).

#### 2.5.1.1 Kebutuhan Fungsional

Berdasarkan data yang dikumpulkan melalui wawancara kepada petani, sistem harus dapat mengidentifikasi jenis penyakit tanaman padi secara akurat, informasi tentang penyakit tanaman padi, serta dapat memberikan informasi tentang cara mengatasi penyakit tanaman padi tersebut.

#### 2.5.1.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional merupakan kebutuhan yang diperlukan pengguna akhir untuk menggunakan sistem ini.

- *Hardware*

*Hardware* adalah peralatan berbentuk fisik yang berfungsi sebagai alat untuk menjalankan *software* selama penelitian ini. Spesifikasi minimum perangkat keras yang diperlukan antara lain: *smartphone* yang memiliki kamera, RAM (*Random Access Memory*) minimum 2GB, 1.4GHz *quad-core processor*, penyimpanan internal 8GB, dan ukuran layar 1080 x 720 pixels.

- *Software*

Perangkat lunak yang digunakan adalah sistem operasi Android dengan versi minimum 7.0 (nougat) yang berguna untuk menjalankan aplikasi hasil akhir dari penelitian ini.

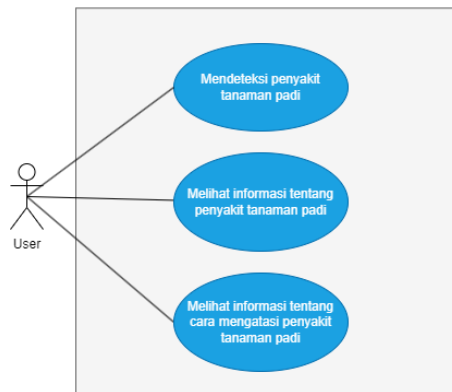
#### 2.5.2 Desain Sistem

Pada tahap ini, setelah data yang diperlukan telah dikumpulkan, perancangan berbagai bagian sistem akan dimulai. Dalam penelitian ini, desain sistem yang digunakan adalah *use case diagram*, *activity diagram*, struktur tabel, dan *mockup*.

##### 2.5.2.1 Use Case Diagram

Use case adalah diagram yang menggambarkan hubungan antara sistem dengan pengguna aplikasi. Pada sistem ini hanya memiliki satu jenis pengguna

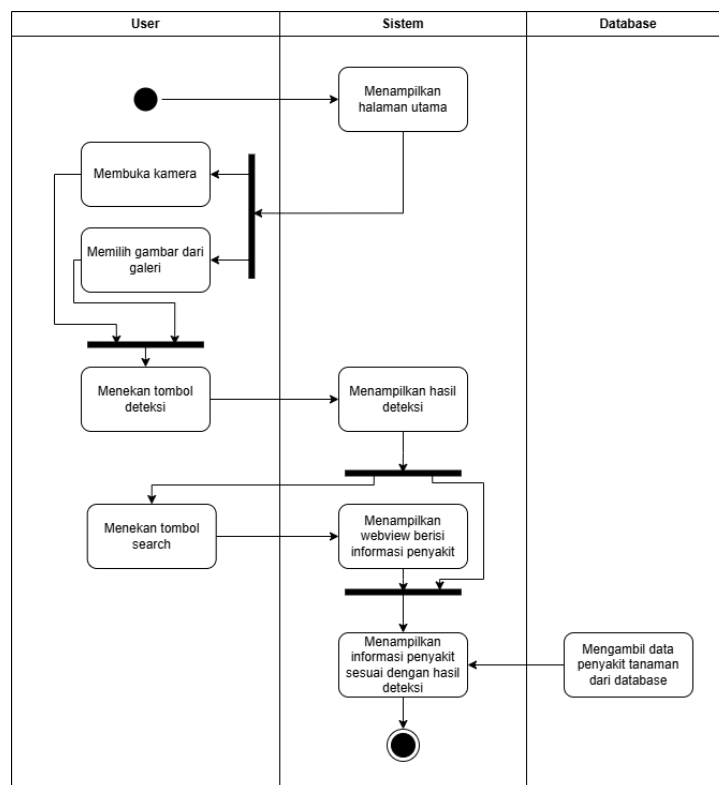
yaitu user. User memiliki akses untuk mendeteksi penyakit tanaman padi, melihat informasi penyakit, dan cara mengatasi penyakit tanaman padi.



Gambar 3. Use Case Diagram

### 2.5.2.2 Activity Diagram

*Activity diagram* menunjukkan alur sistem secara keseluruhan. Diagram ini mirip dengan diagram *use case* tetapi menggunakan simbol seperti tindakan, keputusan, dan garis penghubung. Proses pengembangan akan lebih mudah dengan diagram yang menunjukkan bagaimana sistem akan berjalan.



Gambar 4. Diagram Activity User

Pengguna dapat memilih antara membuka kamera untuk mengambil gambar secara langsung atau dapat memilih gambar yang sudah tersedia pada galeri.

Setelah pengguna memilih gambar, pengguna harus menekan tombol deteksi untuk dapat menampilkan hasil deteksi serta informasi tentang jenis penyakit yang menjangkit tanaman padi.

### 2.5.2.3 Struktur Tabel

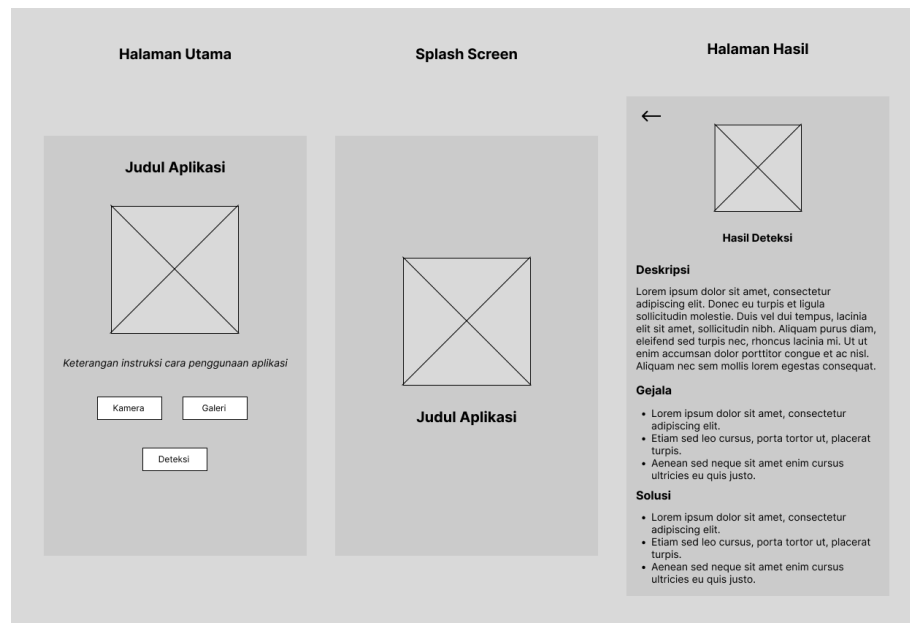
Struktur tabel berguna untuk menampilkan *field* apa saja yang ada dalam satu tabel dari *database*.

| penyakit_tanaman_padi |         |
|-----------------------|---------|
| id                    | integer |
| nama_penyakit         | varchar |
| deskripsi             | varchar |
| gejala                | varchar |
| solusi                | varchar |

Gambar 5. Struktur tabel penyakit\_tanaman\_padi

Pada Gambar 5 menunjukkan bahwa pada tabel penyakit tanaman padi memiliki *field – field* yang berguna untuk menyimpan informasi tentang penyakit tanaman padi.

### 2.5.2.4 Mockup



Gambar 6. Mockup Aplikasi

Pada Gambar 6 menunjukkan gambaran dari halaman – halaman yang ada pada aplikasi. Halaman yang akan ditampilkan saat pertama kali aplikasi dibuka adalah halaman *splash screen*, yang memuat judul dan logo aplikasi, halaman ini akan otomatis beralih ke halaman utama aplikasi yang ditampilkan pada Gambar 6. Pada halaman utama menampilkan judul aplikasi, *preview*

gambar yang telah diambil, dua tombol untuk memilih metode ambil gambar, serta satu tombol yang akan mengarahkan ke halaman hasil. Pada halaman hasil deteksi menampilkan gambar penyakit tanaman padi yang sesuai dengan hasil deteksi, teks hasil deteksi, deskripsi penyakit, gejala penyakit, serta solusi yang dapat diterapkan, serta tombol kembali ke halaman utama.

#### 2.5.3 Penulisan Kode

Setelah data dikumpulkan dan desain sistem dibuat, langkah berikutnya adalah menyusun kode program yang akan digunakan untuk mengembangkan aplikasi. Bahasa pemrograman yang digunakan adalah kotlin, sedangkan editor kode yang digunakan adalah Android Studio.

#### 2.5.4 Pengujian

Tahapan pengujian aplikasi bertujuan untuk mendeteksi potensi kesalahan dalam aplikasi dan menilai apakah aplikasi tersebut memenuhi standar yang diperlukan sebelum dirilis atau memerlukan perbaikan lebih lanjut (Afriyantari & Putri, 2019). Pengujian aplikasi ini diuji menggunakan dua cara pengujian yaitu menguji fungsionalitas dari aplikasi menggunakan *black box testing* serta pengujian menggunakan *System Usability Scale* (SUS). *Black box testing* dipilih karena pengujian dilakukan oleh pengguna akhir yang tidak memiliki pengetahuan mendalam tentang bahasa pemrograman yang digunakan dalam pembuatan aplikasi (Afriyantari & Putri, 2019). Pengujian SUS digunakan untuk mengukur seberapa efektif dan efisien sebuah aplikasi menurut penggunanya dengan mengisi survey yang disediakan. Survey terdiri dari 10 pertanyaan, dan setiap tanggapan memiliki poin 1-5. SUS menghasilkan skor 0 hingga 100, skor lebih tinggi berarti lebih baik.

#### 2.5.5 Pemeliharaan

Pemeliharaan program adalah tahap terakhir dari pengembangan perangkat lunak. Tujuan dari pemeliharaan program adalah untuk memperbaiki bug dan menjaga program tetap relevan. Dengan pemeliharaan program, pengembang juga dapat menambahkan fitur baru dan memastikan pengalaman pengguna yang baik. Pemeliharaan sistem secara berkala dapat mengurangi kemungkinan kegagalan sistem dan memperpanjang umur perangkat lunak (Rahayu & Nurgiyatna, 2022).

### 3. HASIL DAN PEMBAHASAN

Pada tahapan ini, penulis akan membahas mengenai hasil dari proses pelatihan model *machine learning*, proses perancangan aplikasi, dan hasil survey yang telah diberikan kepada pengguna.

#### 3.1 Hasil Pelatihan Model

Dalam kode yang terdapat pada Gambar 7, metode *evaluate()* digunakan untuk mengevaluasi model menggunakan dataset testing *test\_ds*. Hasil evaluasi tersebut menunjukkan bahwa model mencapai *loss* sebesar 0.8290 dan akurasi sebesar 83.49% pada data *testing* dengan jumlah 2090 citra yang tidak termasuk ke dalam dataset pelatihan.

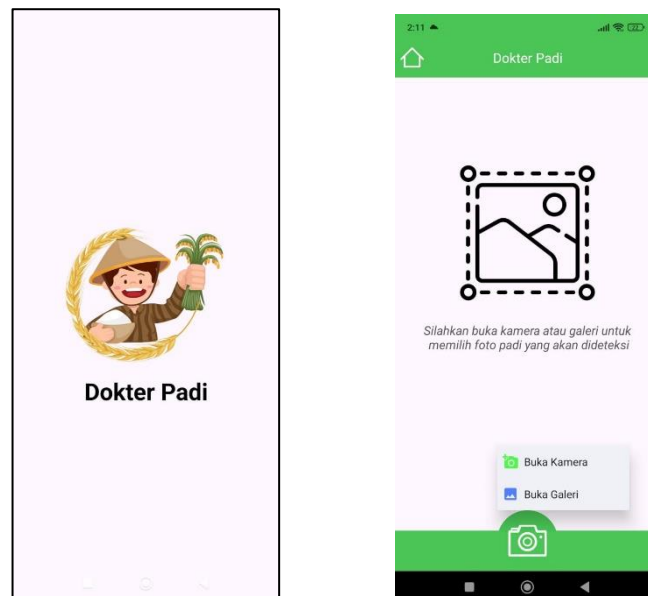
```
model.evaluate(test_ds)

105/105 [=====] - 5s 43ms/step - loss: 0.8290 - accuracy: 0.8349
[0.8289604187011719, 0.8349282145500183]
```

Gambar 7. Hasil evaluasi model

Gambar 7 menunjukkan bahwa model mampu melakukan klasifikasi dengan baik pada data yang belum pernah dilihat sebelumnya, dengan tingkat akurasi yang cukup tinggi. Performa model ini diuji menggunakan dataset *testing* untuk memastikan bahwa model tidak hanya mampu mempelajari pola dari data pelatihan, tetapi juga mampu menggeneralisasi pengetahuannya ke data yang baru.

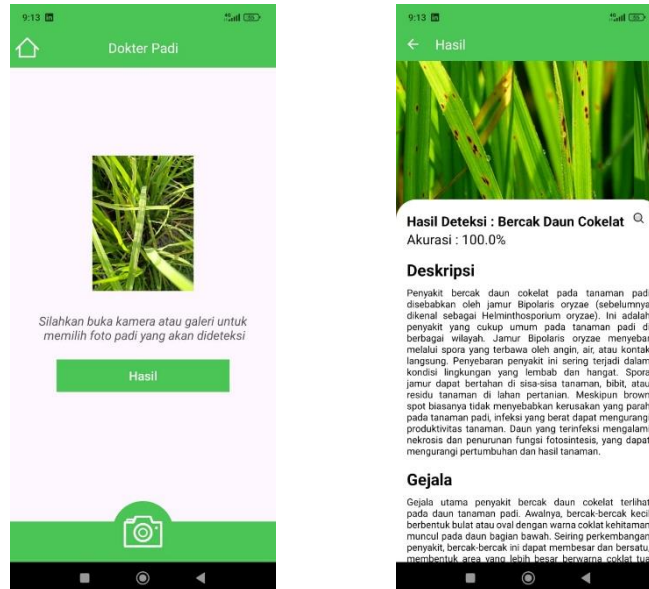
#### 3.2 Hasil Tampilan Aplikasi



Gambar 8. Halaman *splash screen* dan halaman utama

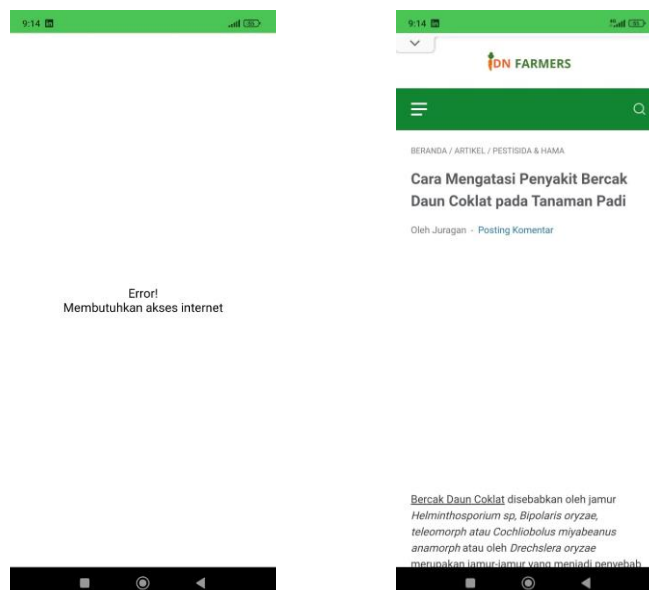
Gambar 8 menunjukkan halaman *splash screen* yang akan tampil pertama kali saat

aplikasi dijalankan dan kemudian secara otomatis berganti ke halaman utama aplikasi. Pada halaman utama aplikasi menunjukkan ikon foto, teks instruksi, serta tombol ikon kamera yang ketika ditekan akan memunculkan popup pilihan buka kamera atau buka galeri.



Gambar 9. Halaman utama dan halaman hasil

Setelah pengguna mengambil gambar atau memilih gambar dari galeri, gambar akan ditampilkan pada halaman utama dan menggantikan ikon gambar semula. Apabila gambar yang dipilih merupakan gambar padi, maka akan muncul tombol hasil dan sebaliknya, apabila gambar bukan tanaman padi, tombol hasil akan menghilang. Ketika tombol hasil ditekan akan berpindah ke halaman hasil yang memberikan hasil deteksi, akurasi, deskripsi penyakit, gejala, serta solusi.



Gambar 10. Halaman webview

Gambar 10 menunjukkan tampilan dari halaman – halaman yang ada pada aplikasi, diantaranya adalah splash screen, halaman utama sebelum gambar dibuka, halaman utama setelah gambar dibuka, halaman hasil, serta halaman webview ketika gagal dibuka dan berhasil dibuka.

### 3.3 Pengujian *Black Box*

Pengujian *black box* dilakukan pada setiap halaman aplikasi untuk memastikan bahwa semua fungsinya berjalan dengan benar. Hasil uji coba *black box* untuk aplikasi ini disajikan dalam Tabel 1.

Tabel 1. Hasil Pengujian *Black Box*

| Bagian Aplikasi       | Nama Pengujian       | Bentuk Pengujian                    | Hasil yang Diharapkan                                                                 | Hasil Pengujian |
|-----------------------|----------------------|-------------------------------------|---------------------------------------------------------------------------------------|-----------------|
| Halaman splash screen | <i>splash screen</i> | Membuka aplikasi untuk pertama kali | Muncul halaman <i>splash screen</i> selama tiga detik dan berpindah ke halaman utama. | Valid           |
| Halaman Utama         | Menu pop-up          | Klik ikon kamera                    | Muncul popup pilihan buka kamera dan buka galeri.                                     | Valid           |
|                       | Menu buka kamera     | Klik menu popup “buka kamera”       | Membuka kamera dan dapat mengambil gambar dengan kamera.                              | Valid           |
|                       | Menu buka galeri     | Klik menu popup “buka galeri”       | Membuka galeri dan dapat memilih gambar dari galeri.                                  | Valid           |
|                       | Menu pilih gambar    | Memilih gambar dari galeri          | Kembali ke halaman utama dan menampilkan gambar.                                      | Valid           |
|                       | Menu ambil gambar    | Mengambil gambar dengan kamera      | Kembali ke halaman utama dan menampilkan gambar.                                      | Valid           |
|                       | Menu hasil           | Klik tombol “hasil”                 | Berpindah ke halaman hasil dan menampilkan prediksi.                                  | Valid           |
| Halaman hasil         | Menampilkan          | Membuka halaman hasil               | Menampilkan data deskripsi, gejala, dan solusi sesuai                                 | Valid           |

| Bagian Aplikasi | Nama Pengujian    | Bentuk Pengujian                              | Hasil yang Diharapkan                                          | Hasil Pengujian |
|-----------------|-------------------|-----------------------------------------------|----------------------------------------------------------------|-----------------|
|                 | data              |                                               | dengan hasil prediksi.                                         |                 |
|                 | Menu webview      | Klik ikon kaca pembesar                       | Berpindah ke halaman webview.                                  | Valid           |
| Halaman webview | Menampilkan data  | Membuka halaman webview dengan akses internet | Menampilkan halaman web sesuai dengan penyakit yang dideteksi. | Valid           |
|                 | Menampilkan error | Membuka halaman webview tanpa akses internet  | Menampilkan halaman error.                                     | Valid           |

Berdasarkan hasil dari pengujian *black box* pada Tabel 1, dapat disimpulkan bahwa semua fungsi yang ada di aplikasi pendeteksi penyakit padi ini dapat berjalan dengan baik dan benar tanpa adanya kesalahan.

### 3.4 Pengujian SUS

Pada tahap pengujian SUS, evaluasi dilakukan dengan menggunakan sepuluh pertanyaan standar yang dirancang secara internasional untuk menilai tingkat kegunaan sistem secara subjektif. Hasil pengujian SUS ditampilkan dalam Tabel 2, memberikan informasi penting tentang bagaimana memahami kebutuhan dan harapan pengguna serta memberikan panduan untuk langkah-langkah selanjutnya yang perlu diambil.

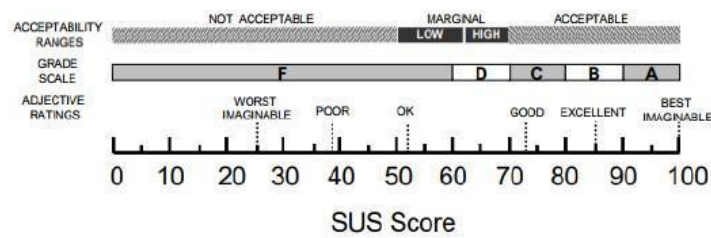
Tabel 2. Pengujian SUS

| Responden | Jawaban Kuesioner |    |    |    |    |    |    |    |    |     | SUS Score |
|-----------|-------------------|----|----|----|----|----|----|----|----|-----|-----------|
|           | Q1                | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 |           |
| 1         | 3                 | 2  | 4  | 2  | 5  | 2  | 4  | 2  | 4  | 2   | 75        |
| 2         | 5                 | 1  | 5  | 1  | 5  | 1  | 5  | 1  | 5  | 1   | 100       |
| 3         | 5                 | 1  | 5  | 1  | 5  | 1  | 5  | 1  | 5  | 1   | 100       |
| 4         | 5                 | 1  | 4  | 2  | 5  | 1  | 5  | 2  | 5  | 1   | 92,5      |
| 5         | 5                 | 1  | 5  | 1  | 5  | 1  | 5  | 1  | 5  | 1   | 100       |

| Responden          | Jawaban Kuesioner |    |    |    |    |    |    |    |    |     | SUS   |
|--------------------|-------------------|----|----|----|----|----|----|----|----|-----|-------|
|                    | Q1                | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 | Score |
| 6                  | 5                 | 1  | 5  | 1  | 5  | 1  | 5  | 1  | 5  | 1   | 100   |
| 7                  | 4                 | 1  | 4  | 2  | 5  | 1  | 4  | 1  | 5  | 2   | 87,5  |
| 8                  | 5                 | 1  | 4  | 1  | 5  | 2  | 5  | 1  | 5  | 1   | 95    |
| 9                  | 5                 | 2  | 5  | 1  | 5  | 2  | 5  | 2  | 5  | 2   | 90    |
| 10                 | 5                 | 1  | 5  | 1  | 5  | 1  | 5  | 1  | 5  | 1   | 100   |
| 11                 | 5                 | 2  | 5  | 2  | 5  | 2  | 5  | 1  | 5  | 2   | 90    |
| 12                 | 5                 | 1  | 5  | 1  | 5  | 1  | 5  | 1  | 5  | 1   | 100   |
| 13                 | 5                 | 1  | 5  | 1  | 5  | 1  | 5  | 1  | 5  | 1   | 100   |
| 14                 | 5                 | 1  | 5  | 1  | 5  | 1  | 5  | 1  | 5  | 1   | 100   |
| 15                 | 4                 | 2  | 5  | 1  | 4  | 2  | 4  | 2  | 5  | 1   | 85    |
| 16                 | 5                 | 1  | 5  | 1  | 5  | 1  | 5  | 1  | 5  | 1   | 100   |
| 17                 | 3                 | 2  | 4  | 4  | 5  | 2  | 5  | 2  | 3  | 3   | 67,5  |
| 18                 | 3                 | 3  | 4  | 3  | 4  | 2  | 3  | 2  | 4  | 4   | 60    |
| 19                 | 5                 | 3  | 4  | 3  | 4  | 3  | 5  | 2  | 5  | 3   | 72,5  |
| 20                 | 3                 | 2  | 4  | 5  | 3  | 3  | 3  | 3  | 2  | 4   | 45    |
| 21                 | 4                 | 2  | 4  | 2  | 4  | 2  | 3  | 2  | 4  | 4   | 67,5  |
| 22                 | 5                 | 1  | 5  | 4  | 3  | 3  | 5  | 1  | 4  | 2   | 77,5  |
| 23                 | 3                 | 4  | 3  | 2  | 5  | 2  | 3  | 1  | 4  | 5   | 60    |
| 24                 | 4                 | 3  | 3  | 2  | 4  | 3  | 4  | 2  | 3  | 3   | 62,5  |
| 25                 | 4                 | 4  | 3  | 5  | 5  | 3  | 2  | 3  | 3  | 5   | 42,5  |
| 26                 | 3                 | 3  | 4  | 3  | 3  | 4  | 4  | 3  | 3  | 5   | 47,5  |
| 27                 | 4                 | 1  | 3  | 3  | 3  | 3  | 4  | 3  | 5  | 5   | 60    |
| 28                 | 2                 | 1  | 5  | 1  | 5  | 3  | 5  | 1  | 5  | 2   | 85    |
| 29                 | 2                 | 4  | 3  | 4  | 3  | 4  | 2  | 3  | 2  | 5   | 30    |
| 30                 | 3                 | 4  | 2  | 5  | 4  | 3  | 1  | 3  | 2  | 4   | 32,5  |
| Rata-rata skor SUS |                   |    |    |    |    |    |    |    |    |     | 77,5  |

Dari pengujian SUS, sistem mendapatkan skor rata-rata 77,5. Hasil tersebut diperoleh dengan mengimplementasikan metode perhitungan SUS yaitu untuk setiap pertanyaan dengan nomor ganjil, skor dari responden dikurangi satu. Kemudian untuk skor yang diberikan responden pada setiap pertanyaan dengan nomor genap, bernilai lima dikurangi jawaban kuesioner. Sehingga berdasarkan pedoman penilaian SUS yang ditampilkan pada Gambar 11, sistem memiliki nilai C yaitu *Good*, hal ini

menunjukkan bahwa sistem layak dan mudah digunakan.



Gambar 11. Penilaian *System Usability Scale*

### 3.5 Uji Validasi

Tahap uji validasi dilakukan dengan mencoba aplikasi akhir untuk mengidentifikasi tanaman padi menggunakan gambar tanaman padi yang diambil secara langsung dari sawah. Dari sepuluh percobaan, didapatkan bahwa sembilan dari sepuluh percobaan menunjukkan hasil yang sesuai, dengan tingkat *confidence* stabil di antara 80% sampai 100%, sedangkan satu percobaan menunjukkan hasil berbeda dengan *confidence* 64%. Hal ini menunjukkan bahwa sistem dapat mengidentifikasi tanaman padi dengan cukup baik.

## 4. PENUTUP

Pada penelitian ini, aplikasi android pendeteksi penyakit padi berhasil dikembangkan dengan baik. Aplikasi ini dapat mendeteksi sembilan kelas penyakit padi dan satu kelas padi normal dengan akurasi sebesar 83,49%. Dengan adanya aplikasi ini petani dapat terbantu untuk mengidentifikasi jenis penyakit padi secara akurat serta dapat menemukan informasi – informasi penting tentang penyakit yang sedang menjangkiti tanaman padi, sehingga petani dapat mengantisipasi kerusakan tanaman padi sejak dini. Selama pengujian *black box*, semua fungsi sistem ini berjalan dengan baik. Selain itu juga dilakukan pengujian *System Usability Scale* yang mendapatkan skor rata-rata 77,5, atau termasuk dalam kategori *Acceptable*, dan mendapatkan nilai C atau *Good*. Dengan demikian, sistem ini layak digunakan, dapat diterima oleh pengguna, dan siap untuk membantu petani dalam mengidentifikasi jenis penyakit tanaman padi.

## DAFTAR PUSTAKA

- Afriyantari, D., & Putri, P. (2019). Rancang Bangun Media Pembelajaran Bahasa Arab untuk Anak Usia Dini Berbasis Android. In *Technologia* (Vol. 10, Issue 3).
- Anggoro, D. A. (2020). Comparison of Accuracy Level of Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) Algorithms in Predicting Heart Disease. *International*

- Journal of Emerging Trends in Engineering Research*, 8(5), 1689–1694.  
<https://doi.org/10.30534/ijeter/2020/32852020>
- Anggoro, D. A., Marzuki, A. A. T., & Supriyanti, W. (2023). Classification of Solo Batik patterns using deep learning convolutional neural networks algorithm. *Telkomnika (Telecommunication Computing Electronics and Control)*, 22(1), 232–240.  
<https://doi.org/10.12928/TELKOMNIKA.v22i1.24598>
- Ariska, F. M., & Qurniawan, B. (2021). PERKEMBANGAN IMPOR BERAS DI INDONESIA. *Journal of Agriculture and Animal Science*, 1(1), 27–34.  
<https://doi.org/10.47637/agrimals.v1i1.342>
- Badan Pusat Statistik. (2024, March 1). *Luas Panen dan Produksi Padi di Indonesia 2023*.  
<https://www.bps.go.id/id/pressrelease/2024/03/01/2375/pada-2023--luas-panen-padi-mencapai-sekitar-10-21-juta-hektare-dengan-produksi-padi-sebesar-53-98-juta-ton-gabah-kering-giling--gkg-.html>
- Gunawan, D., Ar Raniri, I. A., Setyawan, R. N., & Prasetya, Y. D. (2021). WEB-BASED LIBRARY INFORMATION SYSTEM IN MADRASAH IBTIDAIYAH NEGERI SURAKARTA. *Jurnal Teknik Informatika (Jutif)*, 2(1), 33–41.  
<https://doi.org/10.20884/1.jutif.2021.2.1.44>
- Jaisyurahman, U., Wirnas, D., Trikoesoemaningtyas, & Purnamawati, D. H. (2020). Dampak Suhu Tinggi terhadap Pertumbuhan dan Hasil Tanaman Padi. *Jurnal Agronomi Indonesia (Indonesian Journal of Agronomy)*, 47(3), 248–254.  
<https://doi.org/10.24831/jai.v47i3.24892>
- Khan, M. E., Shadab, S. G. M., & Khan, F. (2020). Empirical Study of Software Development Life Cycle and its Various Models. *International Journal of Software Engineering (IJSE)*, 8(2), 16–26.
- Listyarini, S. N., & Anggoro, D. A. (2021). Analisis Sentimen Pilkada di Tengah Pandemi Covid-19 Menggunakan Convolution Neural Network (CNN). *Jurnal Pendidikan Dan Teknologi Indonesia*, 1(7), 261–268. <https://doi.org/10.52436/1.jpti.60>
- Mawan, A. H., & Anggoro, D. A. (2021). Inventory Information System in Benostore Stores. *Emitor: Jurnal Teknik Elektro*, 21(1), 54–58.  
<https://doi.org/10.23917/emitor.v21i1.13262>
- Naufal, M. F. (2021). Analisis Perbandingan Algoritma SVM, KNN, dan CNN untuk Klasifikasi Citra Cuaca. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 8(2), 311–318.  
<https://doi.org/10.25126/jtiik.2021824553>
- Paddy Doctor, Pandarasamy Arjunan (Samy), & Petchiammal. (2022). *paddy-disease-classification*. Kaggle. <https://kaggle.com/competitions/paddy-disease-classification>
- Rahayu, A. W., & Nurgiyatna, N. (2022). Sistem Administrasi Dokumen Berbasis Web pada Forum Human Capital Indonesia. *Emitor: Jurnal Teknik Elektro*, 22(1), 47–54.  
<https://doi.org/10.23917/emitor.v21i2.13657>

- Rino, R. B. (2023). ANDROID APP FOR DIAGNOSING PADDY DISEASE. *International Research Journal of Modernization in Engineering Technology and Science*, 05(03). [www.irjmets.com](http://www.irjmets.com)
- Santosa, A. A., Fu'adah, R. Y. N., & Rizal, S. (2023). Deteksi Penyakit pada Tanaman Padi Menggunakan Pengolahan Citra Digital dengan Metode Convolutional Neural Network. *JOURNAL OF ELECTRICAL AND SYSTEM CONTROL ENGINEERING*, 6(2), 98–108. <https://doi.org/10.31289/jesce.v6i2.7930>
- Veeramreddy, R., Gnanasekaran, A., & Kunchala, S. B. (2024). Detection of diseases in rice leaf using convolutional neural network with transfer learning based on ResNeXt. *International Journal of Electrical and Computer Engineering (IJECE)*, 14(2), 1739. <https://doi.org/10.11591/ijece.v14i2.pp1739-1749>
- Zhou, G., Zhang, W., Chen, A., He, M., & Ma, X. (2019). Rapid Detection of Rice Disease Based on FCM-KM and Faster R-CNN Fusion. *IEEE Access*, 7, 143190–143206. <https://doi.org/10.1109/ACCESS.2019.2943454>