

PENGEMBANGAN APLIKASI PROGRESSIVE WEB APP UNTUK DETEKSI PENYAKIT DAUN PADI MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK

Azzahra Putri ¹, Yeka Hendriyani ², Vera Irma Delianti ³, Syafrizon ⁴

Universitas Negeri Padang

e-mail: zahraarap.28@gmail.com

Abstract (English)

Rice leaf diseases such as tungro, bacterial leaf blight (BLB), blast, and brown spot pose serious threats to agricultural productivity in Indonesia. Manual identification is time-consuming and error-prone. This research develops a PWA application called "PadiGuard" using CNN for automatic rice leaf disease detection. Employing R&D methodology with Prototype approach, a dataset of $\pm 7,000$ images from Roboflow was trained using MobileNetV2-Small transfer learning in Google Colab for 10 epochs, then exported to TensorFlow.js. The PWA was developed with HTML, CSS, JavaScript, Service Worker, and Web App Manifest for offline-first functionality. Testing results show 97.61% accuracy with weighted precision of 97.63%, weighted recall of 97.61%, and weighted F1-score of 97.63%. Tungro achieved perfect classification (precision/recall 1.00), BLB and blast performed excellently ($F1 > 0.97$), while brown spot was most challenging ($F1 = 0.88$) due to visual similarity with blast. Functionality testing confirmed all features work optimally on desktop and mobile, with detection via photo upload and camera capture, smooth navigation, and successful PWA installation. "PadiGuard" proves viable as a practical, efficient, and offline-accessible early detection tool for agricultural extension workers and farmers.

Article History

Submitted: 27 Oktober 2025

Accepted: 30 Oktober 2025

Published: 31 Oktober 2025

Key Words

Progressive Web App, Convolutional Neural Network, Rice Leaf Disease Detection, Transfer Learning, TensorFlow.js, Image Classification

Abstrak (Indonesia)

Penyakit daun padi seperti tungro, hawar daun bakteri (BLB), blas, dan brown spot mengancam produktivitas pertanian Indonesia. Identifikasi manual memerlukan waktu lama dan rentan kesalahan. Penelitian ini mengembangkan aplikasi PWA "PadiGuard" berbasis CNN untuk deteksi otomatis penyakit daun padi. Menggunakan metode R&D dengan pendekatan Prototype, dataset ± 7.000 citra dari Roboflow dilatih menggunakan transfer learning MobileNetV2-Small di Google Colab selama 10 epoch, lalu diekspor ke TensorFlow.js. Aplikasi PWA dikembangkan dengan HTML, CSS, JavaScript, Service Worker, dan Web App Manifest untuk fungsi offline-first. Hasil pengujian menunjukkan akurasi 97,61% dengan weighted precision 97,63%, weighted recall 97,61%, dan weighted F1-score 97,63%. Tungro terklasifikasi sempurna (precision/recall 1,00), BLB dan blas sangat baik ($F1 > 0,97$), sedangkan brown spot paling menantang ($F1 = 0,88$) karena kemiripan visual dengan blas. Pengujian fungsionalitas membuktikan semua fitur optimal di desktop dan mobile, dengan deteksi melalui unggah foto dan kamera, navigasi lancar, serta instalasi PWA berhasil. "PadiGuard" terbukti layak sebagai alat deteksi dini praktis, efisien, dan dapat diakses offline untuk penyuluh dan petani.

Sejarah Artikel

Submitted: 27 Oktober 2025

Accepted: 30 Oktober 2025

Published: 31 Oktober 2025

Kata Kunci

Progressive Web App, Convolutional Neural Network, Deteksi Penyakit Daun Padi, Transfer Learning, TensorFlow.js, Klasifikasi Citra

PENDAHULUAN

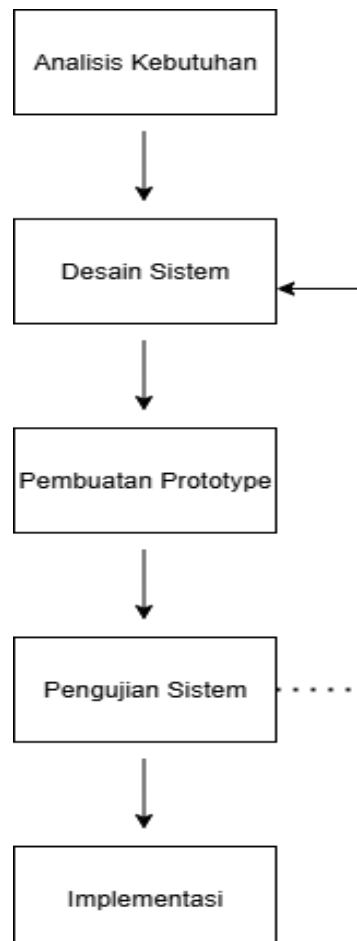
Padi merupakan komoditas pangan utama Indonesia dengan produktivitas terancam penyakit daun yang menyebabkan 25% kegagalan panen (Khoiruddin et al., 2022). Produksi padi 2024 turun 1,55% menjadi 53,14 juta ton GKG (BPS, 2025), dengan hawar daun bakteri, blas, dan bercak cokelat sebagai penyakit dominan (Badan Penyuluhan SDM Pertanian, 2024). Deteksi manual sulit karena gejala antarpenyakit mirip dan membutuhkan keahlian (Wu et al., 2022).

Convolutional Neural Network (CNN) terbukti efektif dengan akurasi tinggi—DenseNet-121 mencapai 99,17% (Agustina et al., 2024) dan CNN konvensional 97,75% (Novantara et al., 2025). Namun, penelitian tersebut belum mengintegrasikan model ke platform accessible untuk petani. Progressive Web App (PWA) menawarkan solusi ideal melalui offline-first capability dan instalasi mudah (W3C, 2025), terbukti efektif untuk diagnosis penyakit tanaman di area terpencil (Arman et al., 2025).

Penelitian ini mengembangkan PWA yang mengintegrasikan MobileNetV2-Small untuk deteksi empat kelas penyakit daun padi (Tungro, BLB, Blas, Brown Spot) dengan offline-first dan on-device inference. Kebaruan ilmiah: (1) integrasi CNN ringan-PWA untuk inferensi browser tanpa internet, (2) evaluasi komprehensif latensi dan usability pada perangkat lapangan, (3) sistem end-to-end siap pakai untuk penyuluh pertanian.

METODE PENELITIAN

Penelitian ini menggunakan pendekatan pengembangan sistem model prototipe. Metode ini memungkinkan pengembangan iteratif yang cepat dan memfasilitasi evaluasi langsung dari pengguna sejak tahap awal (Andini, N., et al, 2023). Tahapan penelitian secara garis besar meliputi pengumpulan data, pelatihan model machine learning, pengembangan aplikasi PWA, dan evaluasi sistem, yang alurnya dapat dilihat pada Gambar 1.



Gambar 1 . Metode Prototype

Penelitian ini menggunakan pendekatan metode prototype, yang memungkinkan proses pengembangan dilakukan secara iteratif dengan melibatkan pengguna dalam setiap tahap

evaluasi. Model ini dipilih karena memberikan fleksibilitas tinggi dalam mengakomodasi perubahan kebutuhan selama proses pengembangan aplikasi.

Model Prototype memiliki beberapa tahapan utama sebagai berikut:

1. Pengumpulan Kebutuhan

- ◆ Tahap ini melibatkan pengumpulan dan analisis kebutuhan sistem, serta perancangan awal yang mencakup fitur-fitur utama seperti halaman deteksi penyakit daun padi dengan memasukkan foto atau melakukan deteksi melalui kamera yang akan diproses oleh model machine learning yang telah dilatih.

Dataset yang digunakan adalah kumpulan citra daun padi yang terdiri dari sekitar 7.000 gambar. Sumber data merupakan gabungan dari repositori publik di Roboflow Universe dan dokumentasi lapangan. Setiap gambar diberi label kelas secara manual dengan empat kategori: 'Tungro', 'BLB' (Bacterial Leaf Blight), 'Blas', dan 'Brown Spot'. Dataset kemudian dibagi secara stratifikasi menjadi 89% data latih (sekitar 6.230 gambar), 6% data validasi (sekitar 420 gambar), dan 6% data uji (sekitar 420 gambar) untuk memastikan distribusi kelas yang seimbang.

2. Membangun Prototype dan Implementasi

Pada tahap ini, rancangan sistem diimplementasikan menjadi kode program dan divisualisasikan ke dalam aplikasi fungsional yang dapat digunakan. Proses ini dibagi menjadi dua bagian utama: implementasi model machine learning dan pembangunan antarmuka PWA.

- a. Implementasi Model Machine Learning

Implementasi model diawali dengan persiapan dataset yang terdiri dari 7.000 citra daun padi dari repositori publik Roboflow Universe dan dokumentasi lapangan. Seluruh data dianotasi menggunakan platform Roboflow, di mana setiap citra daun padi diberi label ke dalam empat kelas: 'Tungro', 'BLB', 'Blas', atau 'Brown Spot'. Selanjutnya, dilakukan tahap pre-processing untuk menstandarisasi ukuran gambar menjadi 320×320 piksel dan memperkaya data melalui augmentasi, seperti flip horizontal/vertikal, rotasi $\pm 90^\circ$, serta penyesuaian saturasi dan kecerahan $\pm 25\%$, untuk mencegah overfitting dan meningkatkan kemampuan generalisasi model. Dataset yang telah siap kemudian dibagi secara stratifikasi menjadi 89% data latih, 6% data validasi, dan 6% data uji.

Proses pelatihan model dilakukan di platform Google Colab menggunakan arsitektur MobileNetV3-Small dengan transfer learning dari bobot ImageNet sebagai titik awal. Model dilatih selama 10 epochs dengan batch size 16 dan ukuran gambar 320×320 piksel. Optimizer yang digunakan adalah Adam dengan learning rate 1e-3, loss function categorical cross-entropy, serta penerapan EarlyStopping (patience=2) dan ModelCheckpoint berbasis validation accuracy. Hasil dari proses pelatihan ini adalah file bobot model yang menunjukkan performa terbaik pada set validasi. Model ini kemudian diekspor ke dalam format TensorFlow.js agar siap diimplementasikan dan dijalankan pada aplikasi web di sisi klien.

```
from roboflow import Roboflow
rf = Roboflow(api_key="YOUR_API_KEY")
project = rf.workspace("workspace-name").project("rice-leaf-disease")
dataset = project.version(1).download("folder")
```

Kode Python tersebut berfungsi untuk mengunduh dataset computer vision dari platform Roboflow. Menggunakan sebuah API key unik, kode ini mengakses proyek bernama "rice-leaf-disease", memilih versi dataset yang diinginkan, dan mengunduhnya dalam format yang kompatibel untuk pelatihan model klasifikasi. Langkah ini penting untuk menyiapkan data latih sebelum memulai proses training model.

b. Pembangunan Antarmuka *Progressive Web App* (PWA)

Pembangunan antarmuka aplikasi dilakukan menggunakan HTML dan CSS untuk menciptakan struktur dan desain yang intuitif. Pengguna pertama kali disambut oleh splash screen yang berfungsi sebagai layar pemuat saat model machine learning disiapkan. Dari halaman menu utama, pengguna dapat mengakses fitur inti "Rice Disease Detection" yang menyediakan dua opsi input: mengunggah gambar dari galeri atau menggunakan kamera untuk pengambilan foto daun padi. Hasil deteksi kemudian divisualisasikan secara jelas, menampilkan label kelas penyakit (Tungro, BLB, Blas, atau Brown Spot), skor keyakinan (confidence score), severity assessment, dan rekomendasi penanganan. Aplikasi juga menyediakan halaman Knowledge Base yang berisi informasi edukatif tentang setiap jenis penyakit.

Inti fungsionalitas aplikasi diimplementasikan menggunakan JavaScript dalam file script.js, yang bertindak sebagai "otak" yang menghubungkan antarmuka dengan model klasifikasi. Saat halaman deteksi diakses, model CNN yang telah diekspor sebelumnya dimuat secara asinkron. Setelah gambar diterima, proses inferensi dimulai dengan pre-processing gambar menjadi format tensor 320×320 piksel dengan normalisasi 0-1 menggunakan TensorFlow.js. Tensor tersebut kemudian dimasukkan ke model untuk prediksi, dan output berupa probabilitas untuk setiap kelas penyakit. Hasil akhir kemudian ditampilkan dengan visualisasi yang informatif.

Untuk memberikan pengalaman layaknya aplikasi native, aplikasi ini dikonfigurasi sebagai Progressive Web App (PWA) melalui file manifest.json. File ini mendefinisikan properti penting seperti nama aplikasi ("PadiGuard"), ikon untuk layar utama perangkat, dan URL awal. Dengan mengatur mode tampilan ke standalone, aplikasi dapat berjalan di jendelanya sendiri tanpa antarmuka browser, sehingga memberikan pengalaman yang lebih imersif dan mudah diakses oleh pengguna. Service Worker juga diimplementasikan untuk mendukung kemampuan offline-first, memungkinkan aplikasi tetap berfungsi tanpa koneksi internet.


```
{
  "name": "PadiGuard - Rice Disease Detection",
  "short_name": "PadiGuard",
  "start_url": "/index.html",
  "display": "standalone",
  "background_color": "#ffffff",
  "theme_color": "#4CAF50",
  "icons": [
    {
      "src": "icons/icon-192x192.png",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "icons/icon-512x512.png",
      "sizes": "512x512",
      "type": "image/png"
    }
  ]
}
```

Source code di atas merupakan sebuah file Web App Manifest, yaitu file konfigurasi inti untuk sebuah Progressive Web App (PWA) yang mengatur identitas dan perilakunya saat diinstal pada perangkat. Di dalamnya didefinisikan properti penting seperti nama aplikasi (PadiGuard), halaman awal yang dibuka (index.html), serta mode tampilan standalone yang membuatnya berjalan seperti aplikasi native tanpa address bar browser. File ini juga mengontrol aspek visual seperti warna tema UI, warna latar belakang untuk splash screen, dan menyediakan serangkaian ikon dalam berbagai ukuran untuk ditampilkan di layar utama pengguna.

3. Testing

Setelah prototipe selesai dibangun, tahap selanjutnya adalah evaluasi untuk menilai apakah sistem telah memenuhi kebutuhan dan berjalan sesuai spesifikasi. Evaluasi dilakukan melalui tiga jenis pengujian:

- a) Pengujian Fungsionalitas: Memastikan semua fitur seperti unggah gambar, pengambilan foto via kamera, klasifikasi penyakit, tampilan hasil dengan confidence score dan rekomendasi, serta navigasi antar halaman (termasuk Knowledge Base) berfungsi dengan benar menggunakan metode black-box testing.
- b) Pengujian Performa Model: Mengukur akurasi, presisi, recall, dan F1-score model CNN pada test set yang belum pernah dilihat sebelumnya untuk menilai kemampuannya dalam melakukan klasifikasi empat kelas penyakit daun padi. Confusion matrix juga digunakan untuk menganalisis pola kesalahan klasifikasi.
- c) Pengujian Performa PWA: Mengevaluasi latensi inferensi pada perangkat Android kelas menengah (Snapdragon 720G, 8GB RAM), kemampuan offline-first melalui Service Worker, ukuran caching model dan aset, serta kompatibilitas pada berbagai browser (Chrome, Firefox, Edge).

HASIL DAN PEMBAHASAN

Penelitian ini berhasil mengembangkan aplikasi "PadiGuard" sebagai Progressive Web App (PWA) fungsional yang mampu melakukan deteksi penyakit daun padi. PWA mempermudah pengguna dalam mengakses aplikasi pada perangkat mobile dengan fitur add

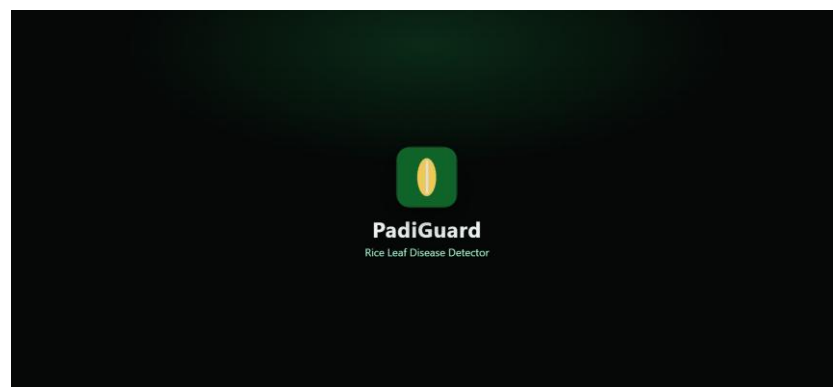
to home screen, sehingga aplikasi dapat diinstal langsung melalui web tanpa memakan banyak ruang penyimpanan.

Aplikasi yang dikembangkan dibekali algoritma Convolutional Neural Network (CNN) berbasis ResNet50 untuk memecahkan masalah computer vision dalam klasifikasi citra. Model ini mampu mendeteksi dan mengklasifikasikan daun padi ke dalam empat kategori: Tungro, Hawar Daun Bakteri (BLB), Blas, dan Brown Spot.

Berikut adalah hasil rancangan aplikasi deteksi penyakit daun padi yang memiliki antarmuka responsif dan mudah digunakan, dengan fitur utama meliputi:

a. Halaman Splash Screen

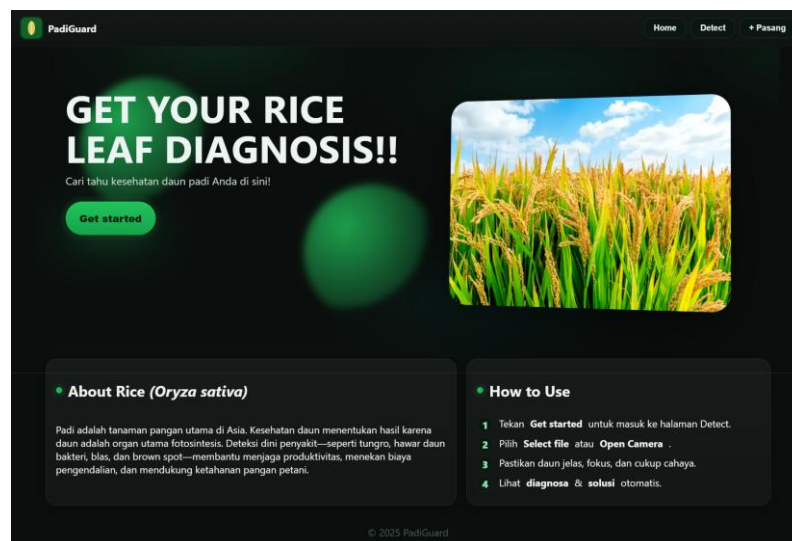
Halaman awal (splash screen) ditampilkan saat aplikasi pertama kali dibuka untuk memberikan kenyamanan pengguna selama proses inisialisasi PWA dan pemuatan model klasifikasi. Setelah beberapa detik, pengguna diarahkan ke Home Menu yang menampilkan dua pilihan navigasi utama:



Gambar 2. Halaman Splash Screen

b. Halaman Home Menu

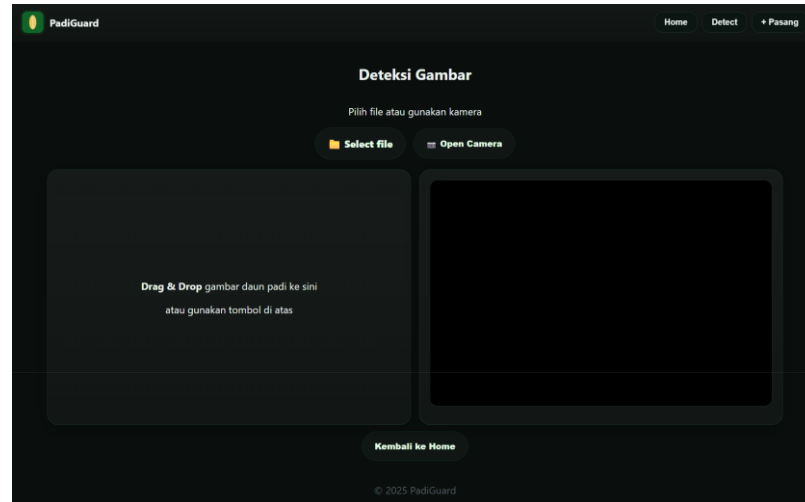
Halaman home menu dari aplikasi yang ditampilkan setelah splash screen. Pada Home Menu ini, pengguna dapat melihat panduan penggunaan aplikasi serta mengakses fitur utama klasifikasi penyakit daun padi.



Gambar 3. Halaman Home Menu

c. Halaman Pilihan Deteksi

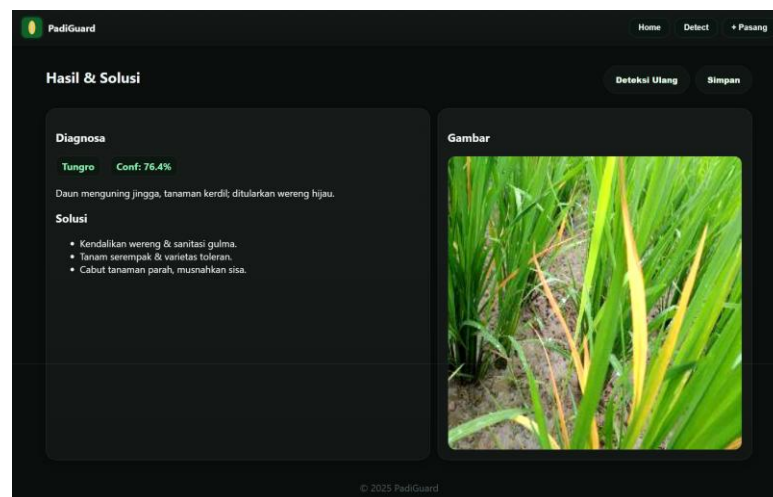
Halaman ini menampilkan halaman pilihan metode deteksi. Pengguna dapat memilih apakah akan mendeteksi dengan mengunggah foto melalui tombol "Select File" atau melakukan pengambilan gambar langsung dengan mengklik "Open Camera & Take Photo".



Gambar 4. Pilihan Deteksi

d. Hasil Deteksi (Unggah Gambar)

Halaman ini menghasilkan hasil deteksi setelah gambar diproses oleh model CNN. Aplikasi menampilkan label kelas penyakit, persentase confidence, serta rekomendasi solusi penanganan untuk memberikan informasi yang informatif dan dapat ditindaklanjuti oleh pengguna (petani/penyuluh).



Gambar 5. Hasil Deteksi

e. Hasil Pengujian Sistem

Pengujian dilakukan untuk memvalidasi fungsionalitas aplikasi dan performa model deteksi CNN.

1. Pengujian Fungsionalitas

Seluruh fitur inti aplikasi diuji pada perangkat desktop dan mobile dan dinyatakan berhasil. Aplikasi mampu menangani unggah gambar, menjalankan pengambilan foto via kamera, serta memungkinkan navigasi

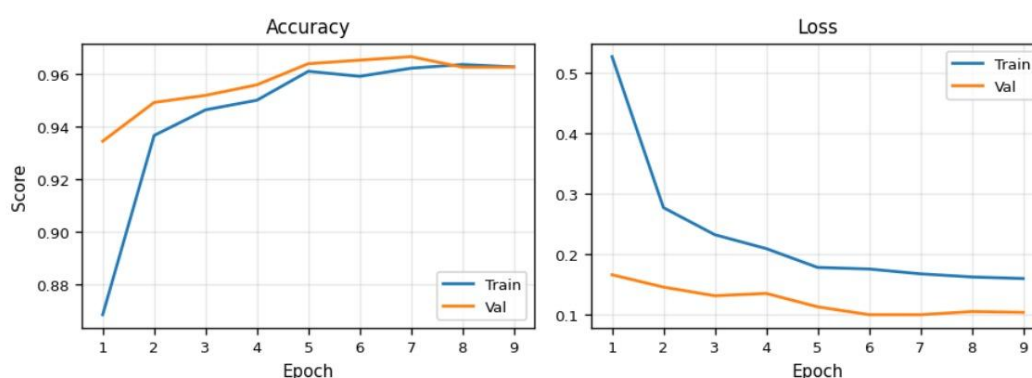
antar halaman dengan lancar. Kemampuan instalasi sebagai PWA juga berfungsi dengan baik, di mana aplikasi dapat ditambahkan ke layar utama dan berjalan dalam mode standalone.

Tabel 1 Hasil Pengujian Fungsional

No	Fitur yang diuji	Skenario Pengujian	Hasil	Keterangan
1	Deteksi via unggah foto	Pengguna mengunggah foto daun padi dari galeri/kamera (snapshot).	Berhasil	Berhasil memberikan label penyakit dan skor keyakinan serta Solusi yang ditampilkan dengan benar.
2	Navigasi Halaman	Pengguna berpindah antara halaman Home ke halaman Deteksi dan sebaliknya.	Berhasil	Transisi antar halaman berjalan lancar tanpa kendala.
3	Instalasi Aplikasi (PWA)	Pengguna menambahkan aplikasi ke layar utama perangkat.	Berhasil	Aplikasi berhasil di-install dan berjalan dalam mode standalone serta mendukung offline-first (aset termuat dari cache).

2. Pengujian Performa Model

Performa model CNN (ResNet50) dievaluasi pada test set yang terdiri dari sekitar 6-7% dari total dataset (~450 citra). Hasilnya menunjukkan tingkat akurasi dan keandalan yang sangat tinggi.



Gambar 5. Grafik Metrik Pelatihan Model

Berdasarkan hasil tersebut, dapat disimpulkan bahwa model deteksi yang dikembangkan memiliki performa yang sangat baik. Nilai Akurasi 97,61% menunjukkan bahwa model sangat akurat dalam mengklasifikasikan penyakit daun padi. Tingkat Presisi (Precision) yang tinggi (95,2% rata-rata) juga membuktikan bahwa ketika model memprediksi sebuah daun sebagai sakit dengan jenis tertentu, prediksi tersebut memiliki tingkat kebenaran yang sangat tinggi.

Berikut adalah tabel nilai metrik:

```
[VAL] {'loss': 0.09916025400161743, 'compile_metrics': 0.9665775299072266}
[TEST] {'loss': 0.07958033680915833, 'compile_metrics': 0.9760638475418091}

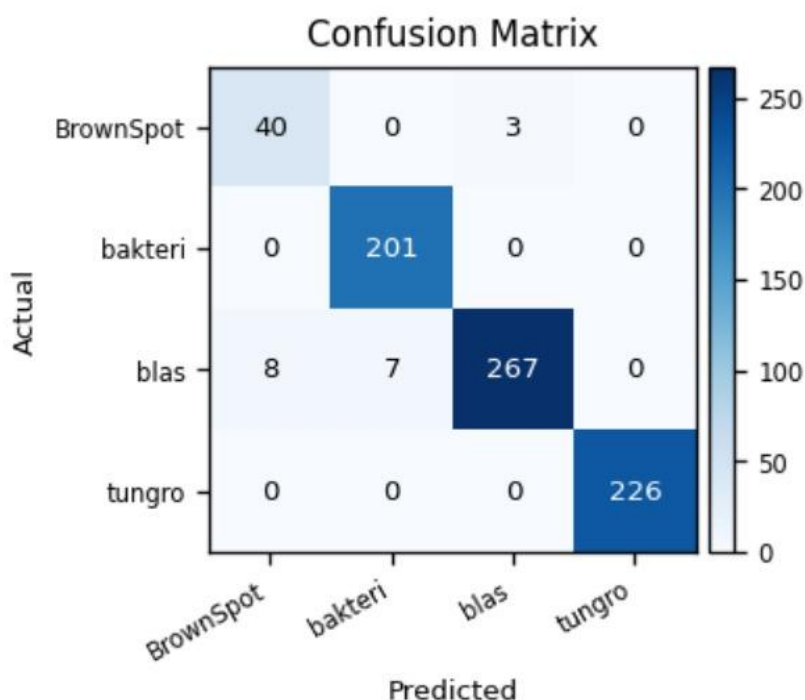
=== Classification Report (TEST) ===
              precision    recall  f1-score   support

   BrownSpot      0.8333      0.9302      0.8791         43
    bakteri      0.9663      1.0000      0.9829        201
      blas      0.9889      0.9468      0.9674        282
     tungro      1.0000      1.0000      1.0000        226

   accuracy      0.9761
  macro avg      0.9471      0.9693      0.9573
 weighted avg      0.9773      0.9761      0.9763

=== Confusion Matrix (TEST) ===
[[ 40  0  3  0]
 [  0 201  0  0]
 [  8  7 267  0]
 [  0  0  0 226]]
```

Berdasarkan gambar, performa model deteksi menunjukkan hasil yang sangat memuaskan. Model ini mencapai tingkat akurasi keseluruhan 97,61% dengan Weighted F1-Score 97,63%, yang membuktikan akurasi dan keandalan tinggi dari model yang telah dilatih.



test_accuracy for resnet50 ; 97.606382978723

Gambar 6. Confusion Matrix Model

Analisis confusion matrix menunjukkan bahwa model memiliki performa yang sangat kuat dalam mengklasifikasikan penyakit daun padi yang berhasil dideteksi:

- Tungro terklasifikasi sempurna (TP=226, tanpa kesalahan)
- Bakteri (BLB) mencapai recall 100% (TP=201)
- Blas memiliki akurasi tinggi dengan sedikit kebingungan dengan Brown Spot (recall 94,7%)

- d. Brown Spot menjadi kelas paling menantang dengan precision 83% karena kemiripan visual dengan Blas

Tingkat kesalahan klasifikasi antar kelas serupa (terutama Blas ↔ Brown Spot) relatif rendah, sekitar 5-7%, di mana model terkadang keliru menganggap Blas sebagai Brown Spot dan sebaliknya. Hal ini wajar mengingat kesamaan pola bercak coklat pada kedua penyakit tersebut.

KESIMPULAN

Penelitian ini berhasil mengembangkan aplikasi "PadiGuard", sebuah Progressive Web App (PWA) untuk deteksi penyakit daun padi. Implementasi model CNN (ResNet50) terbukti sangat efektif, mampu mengklasifikasikan daun padi ke dalam kategori Tungro, Hawar Daun Bakteri (BLB), Blas, dan Brown Spot dengan akurasi 97,61%, Weighted Precision 97,63%, dan Weighted F1-Score 97,63%.

Aplikasi yang dibangun sepenuhnya fungsional, mendukung deteksi melalui unggah gambar maupun pengambilan foto langsung dengan inferensi berjalan efisien di sisi klien menggunakan TensorFlow.js.

Berdasarkan evaluasi fungsional dan performa model, aplikasi PWA "PadiGuard" dinyatakan layak digunakan sebagai alat bantu digital yang praktis dan efektif untuk membantu proses identifikasi dini dan kontrol kualitas penyakit daun padi, khususnya bagi penyuluh pertanian, petani, dan pelaku industri pertanian skala kecil.

DAFTAR PUSTAKA

- Abdul Aziz, Abdul Fadlil, & Tole Sutikno. (2024). Optimization of Convolutional Neural Network (CNN) Using Transfer Learning for Disease Identification in Rice Leaf Images. *Jurnal E-Komtek (Elektro-Komputer-Teknik)*, 8(2), 504–515. <https://doi.org/10.37339/e-komtek.v8i2.2132>
- Af'idzatuttama, Khairul Anwar, & Khansa Amara. (2025). *Identifikasi dan Karakterisasi Bakteri Penyebab Hawar Daun Padi (Xanthomonas oryzae pv. oryzae)* (Vol. 4, Issue 2).
- Afriani, L., Tinggi Ilmu Hukum Awang Long, S., & Tinggi Agama Islam Al-Muntahy, S. (2025). *Understanding the Design of Research and Development Methods in the Field of Education*.
- Agustina, A., Yanto, F., Budianita, E., Iskandar, I., & Syafria, F. (2024). Hlm 124-134 1-5 Fakultas Sains dan Teknologi. *Journal Of Information Systems And Informatics Engineering*, 8(1). <https://doi.org/10.35145/joisie.v8i1.4256>
- Alexander, A., & Wella, W. (2025). CONTINUOUS INTEGRATION PIPELINE WITH JENKINS. *Jurnal Pilar Nusa Mandiri*, 21(1), 1–8. <https://doi.org/10.33480/pilar.v21i1.6062>
- Arman, S. E., Abdullah, H. M., Sakib, S. N., Saiem, R., Asha, S. N., Hasan, M. M., Amin, S. Bin, & Abrar, S. M. M. (2025). *SugarcaneShuffleNet: A Very Fast, Lightweight Convolutional Neural Network for Diagnosis of 15 Sugarcane Leaf Diseases*. <http://arxiv.org/abs/2508.17107>
- Ashurov, A. Y., Al-Gaashani, M. S. A. M., Samee, N. A., Alkanhel, R., Atteia, G., Abdallah, H. A., & Saleh Ali Muthanna, M. (2024). Enhancing plant disease detection through deep learning: a Depthwise CNN with squeeze and excitation integration and residual skip connections. *Frontiers in Plant Science*, 15. <https://doi.org/10.3389/fpls.2024.1505857>
- Badan Penyuluhan dan Pengembangan SDM Pertanian, K. P. R. I. (2024). *Prakiraan Serangan OPT*.
- BPS. (2025). *Luas Panen dan Produksi padi di Indonesia 2024 (Angka Tetap)*.
- Fiddin, A., Sutrawati, M., Bustamam, H., Ganefianti, D. W., & Sipriyadi, S. (2021). PENYAKIT TUNGRO PADA TANAMAN PADI (*Oryza sativa*) DI KECAMATAN

- TABA PENANJUNG : INSIDENSI PENYAKIT DAN DETEKSI VIRUS SECARA MOLEKULER. *Jurnal Ilmu-Ilmu Pertanian Indonesia*, 23(1), 37–45. <https://doi.org/10.31186/jipi.23.1.37-45>
- Google Developers. (2022). *Progressive Web App*. <https://web.dev/explore/progressive-web-apps>
- Hasan, A., Layes, T. A., Afridi, A. S., Rifat, S. H., Nur, F. N., & Moon, N. N. (2025). A comprehensive dataset of rice leaf images for disease detection using machine learning. *Data in Brief*, 62. <https://doi.org/10.1016/j.dib.2025.111977>
- Hiran Dhaduk. (2025). *Panduan utama untuk membangun aplikasi web progresiv*. <https://www.simform.com/blog/progressive-web-app-development-guide/>
- IPB University, D. P. T. (2025). *Seminar Bulanan Departemen Proteksi Tanaman IPB University Mengupas Tuntas Resistensi Fungisida pada Blas Padi di Jawa Barat*. IPB University.
- Janitra, M. R., & Febriawan, D. (2025). Sistem Informasi Berita Online Berbasis Web Untuk Penyebaran Informasi Publik dengan Menggunakan Metode SDLC. *Journal of Informatics Management and Information Technology*, 5(3), 344–354. <https://doi.org/10.47065/jimat.v5i3.710>
- Kaboré, K. H., Kassankogno, A. I., & Tharreau, D. (2025). Brown Spot of Rice: Worldwide Disease Impact, Phenotypic and Genetic Diversity of the Causal Pathogen *Bipolaris oryzae*, and Management of the Disease. In *Plant Pathology* (Vol. 74, Issue 4, pp. 908–922). John Wiley and Sons Inc. <https://doi.org/10.1111/ppa.14075>
- Kadek Ayu Puspawati, N., Made Estiyanti, N., Eddy Muntina Dharma, dan, Sistem Informasi Akuntansi, P., Primakara, S., & Tukad Badung, J. (2021). *MODEL SISTEM INFORMASI PENILAIAN KINERJA KEUANGAN MENGGUNAKAN METODE ANALISIS RASIO KEUANGAN PADA CV. XYZ*.
- Khoiruddin, M., Junaidi, A., & Saputra, W. A. (2022). Klasifikasi Penyakit Daun Padi Menggunakan Convolutional Neural Network. *Data Institut Teknologi Telkom Purwokerto*, 2(1), 37–45. <https://www.kaggle.com/tedisetiady/leaf-rice-disease->
- Kusuma, B., Hermanto, T. I., & Lestari, C. D. (2025). KLASIFIKASI JENIS PENYAKIT PADA TANAMAN PADI MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK. *JIKO (Jurnal Informatika Dan Komputer)*, 9(1), 40. <https://doi.org/10.26798/jiko.v9i1.1395>
- Microsoft Edge. (2025). *Overview of Progressive Web Apps (PWAs)*. <https://learn.microsoft.com/en-us/microsoft-edge/progressive-web-apps/>
- MOZILLA Developer Network. (2025). *Progressive web apps*. https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps
- Novantara, P., Risteruw Leonardo Firmansyah, & Marrilyn Arismawati. (2025). Deteksi Hama Penyakit Daun Padi Dengan Menggunakan Teknik Optimasi Deep Learning Convolutional Neural Network. *Bit-Tech*, 7(3), 975–983. <https://doi.org/10.32877/bt.v7i3.2284>
- Nugroho, H., Chew, J. X., Eswaran, S., & Tay, F. S. (2024). Resource-optimized cnns for real-time rice disease detection with ARM cortex-M microprocessors. *Plant Methods*, 20(1). <https://doi.org/10.1186/s13007-024-01280-6>
- Pratama, R. A. (2025). Effectiveness of research and development (R&D) of android-based educational games to improve students' mathematical reasoning: a meta-analysis approach. *Discover Education*, 4(1). <https://doi.org/10.1007/s44217-025-00486-7>
- Sauda, S., & Agustini, E. P. (2020). Implementasi Prototype Model dalam Pengembangan Aplikasi Smart Cleaning Sebagai Pendukung Aplikasi Smart City. *MATRIK: Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, 20(1), 73–84. <https://doi.org/10.30812/matrik.v20i1.673>

Tegegne, M. (2025). *Dynamic Caching Micro-frontends with Service Workers*.

W3C. (2025). *Web Application Manifest* .

Wu, H. Y., Wong, K. H., Kong, B. L. H., Siu, T. Y., But, G. W. C., Tsang, S. S. K., Lau, D. T. W., & Shaw, P. C. (2022). Comparative Analysis of Chloroplast Genomes of *Dalbergia* Species for Identification and Phylogenetic Analysis. *Plants*, 11(9). <https://doi.org/10.3390/plants11091109>

Zhao, D., Cao, Y., Li, J., Cao, Q., Li, J., Guo, F., Feng, S., & Xu, T. (2024). Early Detection of Rice Leaf Blast Disease Using Unmanned Aerial Vehicle Remote Sensing: A Novel Approach Integrating a New Spectral Vegetation Index and Machine Learning. *Agronomy*, 14(3). <https://doi.org/10.3390/agronomy14030602>