

LeafScan: Aplikasi Cerdas untuk Deteksi Penyakit pada Daun Jagung

Rasyad Bimasatya¹, Mario Valerian Rante Ta'dung², Dewa Ayu Eka Natalia Pratiwi³, Rahmatia⁴, Evan Pandu Nata⁵, A M Fauzan Baihaqi T⁶, and A. Afif Alhaq⁷

* Information System, Department of Mathematics, Faculty of Mathematics and Natural Science, Hasanuddin University

1. Pendahuluan

Jagung merupakan salah satu komoditas pangan utama yang memiliki peran penting dalam sektor pertanian, terutama di Indonesia, di mana ia menjadi sumber makanan dan pakan ternak. Namun, kesehatan tanaman jagung sangat bergantung pada kondisi daun, yang merupakan organ utama dalam proses fotosintesis. Saat ini, serangan penyakit seperti bercak daun, karat daun, dan hawar daun sering menjadi ancaman serius yang dapat menurunkan kualitas serta kuantitas hasil panen. Studi menunjukkan bahwa teknologi pemrosesan citra seperti Convolutional Neural Network (CNN) dapat mendeteksi penyakit pada daun jagung dengan tingkat akurasi hingga 94% [1].

Petani menghadapi tantangan besar untuk menemukan penyakit pada tanaman jagung. Penyakit dapat mengurangi hasil panen secara signifikan jika tidak diidentifikasi dengan benar. Oleh karena itu, sangat penting untuk mengembangkan aplikasi yang menggunakan pemrosesan gambar dan kecerdasan buatan (AI) untuk mendeteksi penyakit daun jagung. Tujuan aplikasi ini adalah untuk menawarkan cara yang efektif untuk melacak kesehatan tanaman, mengidentifikasi penyakit, dan memberikan saran perawatan yang tepat. Selain itu, sebuah studi mengembangkan sistem pakar berbasis Android yang dapat mendeteksi penyakit dan hama pada tanaman jagung, menunjukkan betapa pentingnya alat untuk pendeteksi dini penyakit dan hama pada tanaman jagung bagi petani [2].

Adanya aplikasi LeafScan diharapkan akan membantu petani menemukan penyakit daun jagung dengan cepat dan akurat. Dengan menggunakan teknologi AI, aplikasi ini dapat menemukan pola-pola visual yang menunjukkan penyakit pada gambar daun jagung. Hal ini akan meningkatkan akurasi diagnosis dan mempercepat proses deteksi, yang memungkinkan tindakan pencegahan dilakukan lebih awal. Dari penelitian sebelumnya menunjukkan bahwa model berbasis CNN dapat mengidentifikasi penyakit pada daun jagung dengan akurasi 93%. Hasil tersebut dapat mendukung adanya pengembangan aplikasi ini [3].

Penelitian ini sangat penting karena kondisi lapangan saat ini, di mana petani sering menghadapi masalah dalam menemukan dan mengatasi penyakit tanaman. Aplikasi ini diharapkan dapat menurunkan kesalahan diagnosis dan meningkatkan produktivitas pertanian jagung secara keseluruhan. Studi ini mengevaluasi seberapa baik aplikasi LeafScan mengidentifikasi penyakit pada tanaman jagung dengan lebih akurat dan cepat.

Penelitian ini akan mengukur seberapa akurat diagnosis yang dapat dicapai dengan metode manual dibandingkan dengan aplikasi LeafScan, dan bagaimana aplikasi ini dapat membantu petani menemukan penyakit daun jagung dengan cepat dan akurat. Penelitian ini berfokus pada pembuatan alat yang bermanfaat dan relevan bagi petani dengan tujuan peluncuran aplikasi yang jelas dan terukur.

Penelitian ini mencakup pembuatan aplikasi yang diperuntukan kepada petani, pelajar dan juga mahasiswa pertanian. Oleh karena itu, hasilnya diharapkan dapat membantu sektor pertanian di Indonesia melalui peningkatan pengetahuan dan kemampuan petani untuk menemukan dan mengobati penyakit pada tanaman jagung.

Citation: Lastname, F.; Lastname, F.; Lastname, F. Title. *Journal Not Specified* **2024**, *1*, 0. <https://doi.org/>

Received:

Revised:

Accepted:

Published:

Copyright: © 2024 by the authors. Submitted to *Journal Not Specified* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

2. Metode

Aplikasi ini menggunakan pendekatan berbasis *deep learning* untuk mendeteksi penyakit pada daun jagung dengan cepat dan akurat melalui pemrosesan gambar. Metode penelitian ini melibatkan beberapa tahap, yaitu pengumpulan data, penjelasan data, pemilihan algoritma atau model, prosedur pengujian, serta evaluasi performa model. Berikut adalah rincian metode yang digunakan:

2.1. Pengumpulan Data

Data untuk penelitian ini diperoleh dari platform Kaggle, yang merupakan salah satu sumber terkemuka untuk berbagai dataset dalam berbagai bidang. Dataset yang digunakan dalam penelitian ini berisi 4.188 gambar daun jagung dengan berbagai kondisi kesehatan dan jenis penyakit. Setiap gambar telah diberi label sesuai dengan jenis penyakitnya, sehingga data ini dapat mendukung keperluan pelatihan dan pengujian model deteksi penyakit berbasis *deep learning*.

2.2. Penjelasan Data

Dataset yang digunakan dalam penelitian ini terdiri dari gambar daun jagung dengan berbagai kondisi kesehatan, yang dikelompokkan ke dalam empat kategori utama. Data dikumpulkan dari platform *Kaggle* dan mencakup total 4.188 gambar berformat JPG yang terbagi dalam kategori sebagai berikut:

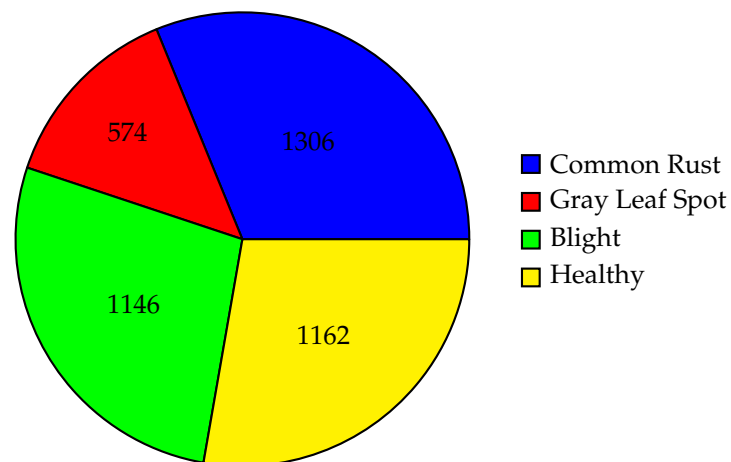


Figure 1. Diagram Lingkaran untuk Jumlah Data pada Setiap Kategori

Berikut adalah penjelasan mengenai masing-masing kelas dalam dataset ini:

- *Common Rust*: Kelas ini mencakup gambar daun jagung yang terinfeksi oleh penyakit karat daun, yang disebabkan oleh jamur *Puccinia sorghi* [4]. Penyakit ini mengakibatkan bercak-bercak coklat kemerahan di permukaan daun, dan daun yang terinfeksi biasanya menunjukkan tanda-tanda penguningan pada bagian tengah daun.
- *Gray Leaf Spot*: Kelas ini berisi gambar daun jagung yang terkena penyakit bercak daun abu-abu, yang disebabkan oleh jamur *Cercospora zeae-maydis* [5]. Ciri-ciri daun yang terinfeksi adalah munculnya bercak abu-abu yang berujung hitam pada daun, yang dapat mengurangi efisiensi fotosintesis pada tanaman.
- *Blight*: Kelas ini terdiri dari gambar daun yang terinfeksi oleh penyakit hawar daun, yang dapat disebabkan oleh berbagai patogen seperti jamur atau bakteri. Penyakit ini menyebabkan daun jagung menguning, mati, dan mengering, yang dapat merusak jaringan daun secara luas [6].
- *Healthy*: Kelas ini mencakup gambar daun jagung yang tidak terinfeksi penyakit apapun, yang terlihat segar dan berwarna hijau normal. Daun jagung sehat memiliki tekstur yang tidak rusak dan tidak menunjukkan tanda-tanda kerusakan atau infeksi.

2.3. Algoritma atau Model

- **YOLOv8l:**

YOLOv8 yang dirilis pada Januari 2023 oleh Ultralytics, menawarkan berbagai versi skala untuk deteksi objek, mulai dari YOLOv8n (nano) hingga YOLOv8x (extra-large). Arsitekturnya mirip dengan YOLOv5, namun mengganti CSPLayer dengan modul C2f (cross-stage partial bottleneck dengan dua konvolusi) untuk meningkatkan akurasi deteksi. YOLOv8 mengadopsi model tanpa anchor dengan head yang dipisah untuk menangani tugas objectness, classification, dan regression secara mandiri, yang berkontribusi pada peningkatan akurasi model secara keseluruhan. Pada lapisan output, YOLOv8 menggunakan fungsi sigmoid untuk menghitung objectness score dan fungsi softmax untuk class probabilities. Model ini juga memanfaatkan loss functions CIOU dan DFL untuk bounding-box dan binary cross-entropy untuk klasifikasi, yang meningkatkan performa deteksi, terutama pada objek yang lebih kecil. Dengan kecepatan dan efisiensi yang tinggi, YOLOv8 menjadi model yang sangat andal dalam deteksi objek. [15]

- **Penggunaan SAHI dalam Model**

Slicing Aided Hyper Inference (SAHI) menggunakan teknik pemotongan selama proses inferensi [8]. Gambar asli I dibagi menjadi beberapa potongan tumpang tindih dengan ukuran $M \times N$, yang disebut $P_{I1}, P_{I2}, \dots, P_{I1}$. Setiap potongan kemudian diubah ukurannya dengan menjaga rasio aspek gambar. Setelah itu, deteksi objek dilakukan pada setiap potongan secara terpisah. Jika diperlukan, full-inference (FI) juga bisa dilakukan dengan gambar asli untuk mendeteksi objek yang lebih besar. Hasil prediksi dari potongan-potongan ini dan, jika ada, hasil FI digabungkan kembali ke dalam gambar asli menggunakan teknik Non-Maximum Suppression (NMS). Pada NMS, kotak yang memiliki nilai Intersection over Union (IoU) lebih besar dari ambang batas yang ditentukan T_m akan dipadankan, dan deteksi dengan probabilitas di bawah ambang batas T_d akan dihapus.

2.4. Pra-Pelatihan (Prosedur)

Aplikasi Roboflow digunakan untuk melakukan anotasi pada data untuk menandai semua jenis penyakit berdasarkan 3 kategori yang telah ditentukan. Kemudian, data yang telah dianotasi diproses terlebih dahulu sebelum digunakan dalam pelatihan model. Berikut adalah prosesnya:

- **Pra-pemrosesan**

- **Auto-Orient**

Pengaturan ini secara otomatis memperbaiki orientasi gambar berdasarkan meta-data, memastikan semua gambar memiliki orientasi yang konsisten.

- **Resize**

Setiap gambar diubah ukurannya menjadi 640 x 640 piksel. Ukuran ini diperlukan untuk melatih model YOLOv8.

- **Pembagian Data**

Data yang telah melalui pra-pemrosesan akan dibagi menjadi tiga kategori, yaitu data latih, data uji, dan data evaluasi, dengan proporsi masing-masing sebesar 80:10:10. Pembagian ini dilakukan agar model dapat dilatih, divalidasi, dan diuji secara menyeluruh. Pembagian data tersebut adalah sebagai berikut:

- **Data Latih (80%):** Digunakan untuk melatih model agar dapat mengenali pola dan karakteristik pada gambar daun jagung berdasarkan kategori penyakitnya.

- **Data Uji (10%):** Digunakan untuk mengevaluasi performa model selama proses pengembangan guna memastikan model tidak mengalami overfitting atau underfitting.

- **Data Evaluasi (10%):** Digunakan sebagai tes akhir untuk mengukur performa model secara independen pada data yang belum pernah dilihat sebelumnya, memastikan akurasi prediksi yang baik pada data nyata.

- **Augmentasi**

Augmentasi gambar hanya dilakukan pada data kategori latih (train). Langkah ini diambil untuk meningkatkan jumlah dan variasi data latih, sehingga model dapat belajar dari lebih banyak contoh yang berbeda. Dengan cara ini, model dapat menjadi lebih robust dan lebih mampu mengenali pola pada data yang lebih beragam, tanpa perlu mengubah data uji atau evaluasi yang harus tetap representatif terhadap kondisi data yang sesungguhnya.

 - **Hue**

Menyesuaikan hue setiap gambar secara acak dalam rentang -15 hingga +15 derajat. Ini mensimulasikan variasi warna kecil untuk membuat model lebih kuat terhadap perubahan warna.
 - **Saturation**

Mengubah saturasi setiap gambar dalam rentang -30% hingga +30%. Ini mengubah intensitas warna, membantu model beradaptasi dengan gambar yang memiliki kecerahan warna yang berbeda.
 - **Brightness**

Mengubah kecerahan setiap gambar dengan menyesuaikannya secara acak antara -23% hingga +23%. Ini membantu model belajar menangani gambar dengan kondisi pencahayaan yang berbeda.
 - **Exposure**

Mengubah eksposur setiap gambar dalam rentang -10% hingga +10%, mensimulasikan berbagai lingkungan pencahayaan untuk meningkatkan adaptabilitas model terhadap tingkat eksposur yang berbeda.
 - **Blur**

Menambahkan efek blur dengan radius maksimum 2.2 piksel. Ini meniru gambar yang tidak fokus atau blur karena gerakan, yang dapat membuat model lebih tahan terhadap sedikit keburaman pada data input.
 - **Bounding Box: Flip (Horizontal)**

Pengaturan ini membalik gambar dan anotasi bounding box secara horizontal. Augmentasi ini memberikan pandangan cermin objek, meningkatkan kemampuan model untuk mengenali fitur dari berbagai perspektif.

2.5. Metrik Evaluasi

- **Box Loss**

Model YOLO yang dikembangkan oleh Ultralytics menggunakan Complete Intersection Over Union (CIoU) untuk menilai seberapa baik prediksi bounding box cocok dengan bounding box target dalam tugas deteksi objek[14]. Berikut adalah Persamaan CIoU:

$$\text{CIoU}(A, B) = 1 - \text{IoU}(A, B) + \frac{\rho^2(\mathbf{b}, \mathbf{b}^g)}{c^2} + \alpha v$$

Keterangan:

- $\text{IoU}(A, B)$ adalah Intersection over Union antara bounding box A dan B .
- $\rho(\mathbf{b}, \mathbf{b}^g)$ adalah jarak Euklidean antara pusat dari bounding box prediksi \mathbf{b} dan bounding box target \mathbf{b}^g .
- c adalah diagonal dari kotak yang mengelilingi kedua bounding box.
- α dan v adalah parameter penyeimbang untuk aspek rasio.
- **Class Loss**

Classification Loss (class loss atau cls loss) dalam YOLOv8 mengukur seberapa akurat model dalam mengklasifikasikan objek yang terdeteksi ke dalam kelas yang benar. Ini dilakukan dengan membandingkan probabilitas kelas yang diprediksi model dengan kelas yang sebenarnya (ground truth). Cross-Entropy Loss digunakan dalam perhitungan class loss[16]. Berikut adalah Persamaan Cross-Entropy Loss:

$$L_{\text{cls}} = - \sum_{c=1}^C y_c \log(p_c)$$

Keterangan:

- C adalah jumlah kelas dalam dataset (misalnya, jumlah kelas objek yang perlu dideteksi).
- y_c adalah label kelas target (nilai 1 jika objek termasuk dalam kelas c , atau 0 jika tidak).
- p_c adalah probabilitas prediksi untuk kelas c , yang dihitung oleh model.

- **Precision**

Precision mengukur seberapa banyak prediksi positif model yang benar. Ini dihitung dengan membandingkan jumlah deteksi yang benar dengan total deteksi yang dilakukan. Precision memberikan informasi tentang seberapa tepat model dalam memprediksi objek, yang penting untuk mengevaluasi apakah model menghasilkan banyak prediksi yang salah atau tidak[13]. Berikut adalah Persamaan Precision:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Keterangan:

- TP adalah jumlah *True Positives* (prediksi benar yang cocok dengan ground truth).
- FP adalah jumlah *False Positives* (prediksi salah).

- **Recall**

Recall mengukur seberapa banyak objek yang benar-benar ada dalam gambar yang berhasil dideteksi oleh model. Recall memberikan informasi tentang seberapa baik model dalam menemukan objek yang ada dalam gambar, yang penting untuk mengevaluasi kemampuan model dalam mendeteksi semua objek yang relevan[13]. Berikut adalah Persamaan Recall:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Keterangan:

- TP adalah jumlah *True Positives* (prediksi benar yang cocok dengan ground truth).
- FN adalah jumlah *False Negatives* (objek yang ada namun tidak terdeteksi oleh model).

- **mAP50**

Mean Average Precision pada threshold Intersection over Union (IoU) 0.5 ($mAP@50$) adalah rata-rata dari *average precision* (AP) pada berbagai kategori objek untuk IoU lebih dari 50%. Metrik ini digunakan untuk mengevaluasi seberapa baik model dalam mendeteksi objek secara keseluruhan, dengan fokus pada deteksi yang tepat, mengabaikan deteksi yang tidak sesuai dengan objek yang sebenarnya[13]. Berikut adalah Persamaan mAP50:

$$mAP@50 = \frac{1}{N} \sum_{i=1}^N AP@50_i$$

Keterangan:

- N adalah jumlah kategori objek dalam dataset.
- $AP@50_i$ adalah *Average Precision* pada kategori ke- i dengan threshold IoU 0.5.

- **mAP50-95**

Mean Average Precision pada IoU thresholds dari 0.5 hingga 0.95, dengan interval 0.05. Metrik ini memberikan gambaran yang lebih mendetail tentang performa model

dalam mendeteksi objek pada berbagai tingkat overlap. Dengan menghitung precision pada berbagai threshold, mAP@50-95 memberikan evaluasi yang lebih komprehensif mengenai kemampuan model dalam mengidentifikasi objek pada berbagai tingkat ketepatan prediksi[13]. Berikut adalah Persamaan mAP50-95:

$$mAP@50-95 = \frac{1}{N} \sum_{i=1}^N AP@thresholds_i$$

Keterangan:

- N adalah jumlah kategori objek.
- $AP@thresholds_i$ adalah average precision untuk kategori ke- i pada berbagai nilai threshold IoU dari 0.5 hingga 0.95 dengan interval 0.05.

3. Sistem Kecerdasan

Sistem kecerdasan pada aplikasi *LeafScan* didasarkan pada penerapan teknologi *deep learning* untuk mendeteksi penyakit pada daun jagung secara otomatis melalui gambar yang diunggah oleh pengguna. Sistem ini memanfaatkan model *You Only Look Once v8* (YOLOv8) yang dikombinasikan dengan algoritma *Slicing Aided Hyper Inference* (SAHI) untuk meningkatkan akurasi dan efisiensi deteksi objek. YOLOv11, yang dikenal dengan kemampuannya untuk mendeteksi objek [7], digunakan untuk mendeteksi penyakit pada daun jagung. Sementara itu, SAHI membantu dalam meningkatkan kemampuan model dalam menangani gambar dengan objek yang lebih kecil atau lebih kompleks, serta meningkatkan akurasi prediksi dengan melakukan pemotongan gambar untuk memaksimalkan deteksi [8].

3.1. Arsitektur Sistem

Arsitektur sistem *LeafScan* dirancang untuk mengoptimalkan deteksi penyakit pada daun jagung dengan menggunakan berbagai teknologi modern yang terintegrasi. Terdapat tiga elemen utama dalam arsitektur ini, yaitu *Frontend*, *Back-End*, dan *Storage*.

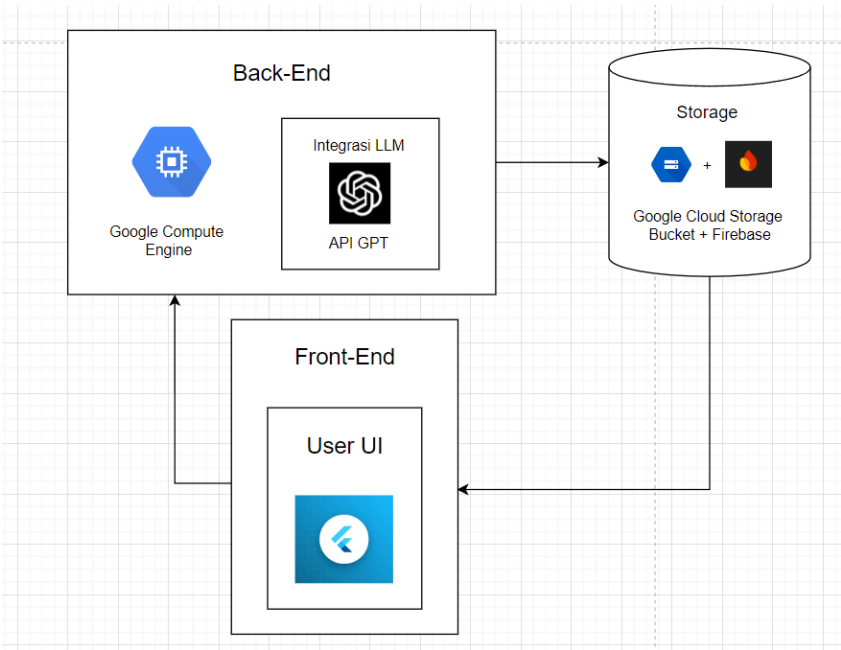


Figure 2. Arsitektur Sistem *LeafScan*

- **Frontend**
Frontend aplikasi *LeafScan* dikembangkan menggunakan *Flutter*, yang memungkinkan pembuatan aplikasi mobile yang responsif dan dapat dijalankan di berbagai platform

seperti Android dan iOS [9]. Flutter menyediakan antarmuka pengguna yang inter-aktif dan mudah digunakan, memungkinkan pengguna untuk mengunggah gambar daun jagung yang ingin dideteksi penyakitnya. Pengguna dapat melihat hasil deteksi serta rekomendasi pengobatan yang dihasilkan oleh sistem.

- **Back-End**
Google Compute Engine digunakan sebagai server untuk menjalankan model deep learning YOLOv8 yang diintegrasikan dengan algoritma *Slicing Aided Hyper Inference* (SAHI). Model ini berfungsi untuk menganalisis gambar daun jagung yang diunggah dan mendeteksi kemungkinan adanya penyakit. *Google Compute Engine* memberikan infrastruktur yang skalabel dan dapat menangani permintaan pengguna secara efisien, sehingga memastikan proses deteksi berjalan dengan cepat dan akurat [10]. Selain itu, sistem juga mengintegrasikan *API GPT* yaitu *OpenAI GPT* untuk memberikan rekomendasi berbasis teks yang lebih spesifik sesuai dengan jenis penyakit yang terdeteksi pada daun jagung. *API GPT* dapat membantu menyediakan informasi tambahan mengenai cara pengobatan dan pencegahan penyakit bagi petani.
- **Storage**
Google Cloud Storage (GCP) digunakan untuk menyimpan data deteksi, termasuk gambar yang diunggah oleh pengguna dan hasil deteksi penyakit. Penyimpanan cloud ini memastikan data dapat diakses secara efisien dan aman dari berbagai perangkat. GCP juga memudahkan pengelolaan data dalam jumlah besar dan mendukung skalabilitas sistem [11]. Selain itu, *Firebase* digunakan untuk mengelola autentikasi pengguna, memungkinkan pengguna untuk mendaftar dan masuk ke aplikasi dengan aman [12]. Dengan *Firebase*, data pengguna seperti riwayat deteksi dan rekomendasi pengobatan juga dapat disinkronkan dengan mudah dan memberikan kenyamanan lebih bagi pengguna yang ingin mengakses data mereka kapan saja.

3.2. Alur Kerja Sistem

Alur kerja sistem *LeafScan* mencakup serangkaian proses yang dimulai dari pengguna mengunggah gambar daun jagung hingga mendapatkan hasil deteksi penyakit dan rekomendasi penanganan. Berikut ini adalah deskripsi dari diagram *use case* dan *activity diagram* yang merepresentasikan alur kerja aplikasi.

3.2.1. Use Case Diagram

Aplikasi *LeafScan* memiliki beberapa aktor utama, yaitu Petani dan Mahasiswa Pertanian, yang memiliki akses ke berbagai fitur sistem. Berikut adalah penjelasan dari tiap *use case*:

- Mendeteksi Penyakit Daun Jagung: Aktor Petani dan Mahasiswa Pertanian dapat menggunakan aplikasi untuk mendeteksi penyakit pada daun jagung dengan mengunggah gambar daun yang akan dianalisis.
- Mengupload Gambar: Proses ini adalah bagian dari deteksi penyakit daun jagung. Pengguna perlu mengunggah gambar untuk menjalankan proses deteksi.
- Memberi Rekomendasi Penanganan: Jika penyakit terdeteksi, sistem akan memberikan rekomendasi penanganan yang sesuai.
- Melihat Riwayat Deteksi Penyakit: Aktor Petani dan Mahasiswa Pertanian dapat melihat riwayat deteksi penyakit yang telah dilakukan sebelumnya, sehingga memudahkan dalam memantau perkembangan kondisi daun jagung dari waktu ke waktu.

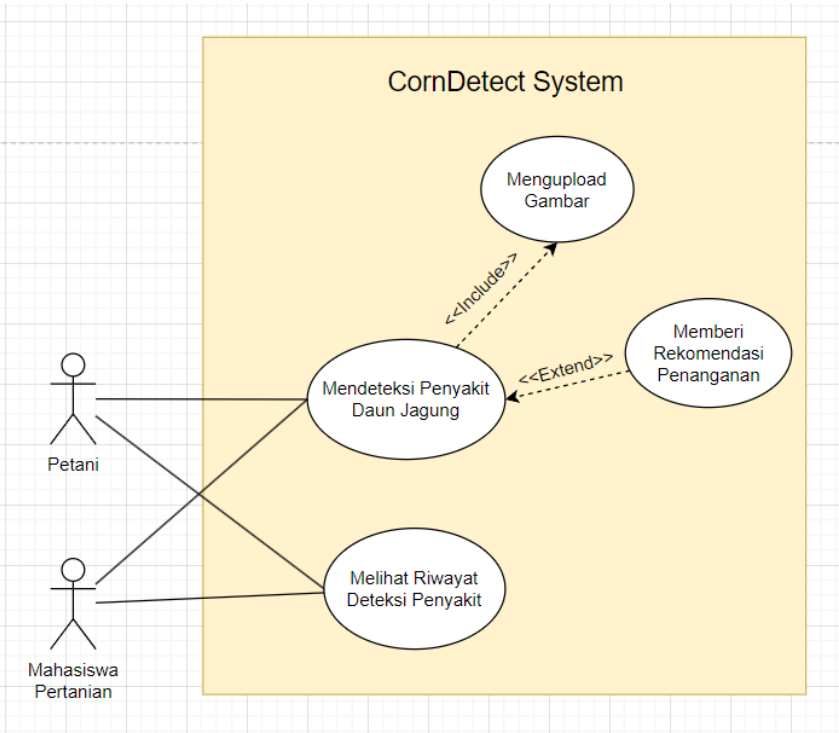


Figure 3. Diagram Use Case Aplikasi LeafScan

3.2.2. Activity Diagram

Untuk memperjelas alur kerja aplikasi, berikut ini adalah *activity diagram* yang menggambarkan proses utama dari saat pengguna memulai interaksi hingga mendapatkan rekomendasi penanganan penyakit.

- a) **Activity Diagram - Mengupload Gambar**
Pada Gambar 4, diperlihatkan alur aktivitas untuk proses Mengupload Gambar. Proses ini dimulai ketika pengguna menekan tombol untuk deteksi gambar dan memilih opsi pengambilan gambar dari kamera atau galeri. Setelah gambar diambil, pengguna dapat memotong gambar sebelum mengunggahnya ke server. Sistem kemudian menampilkan konfirmasi upload dan melanjutkan untuk mengunggah gambar ke server.

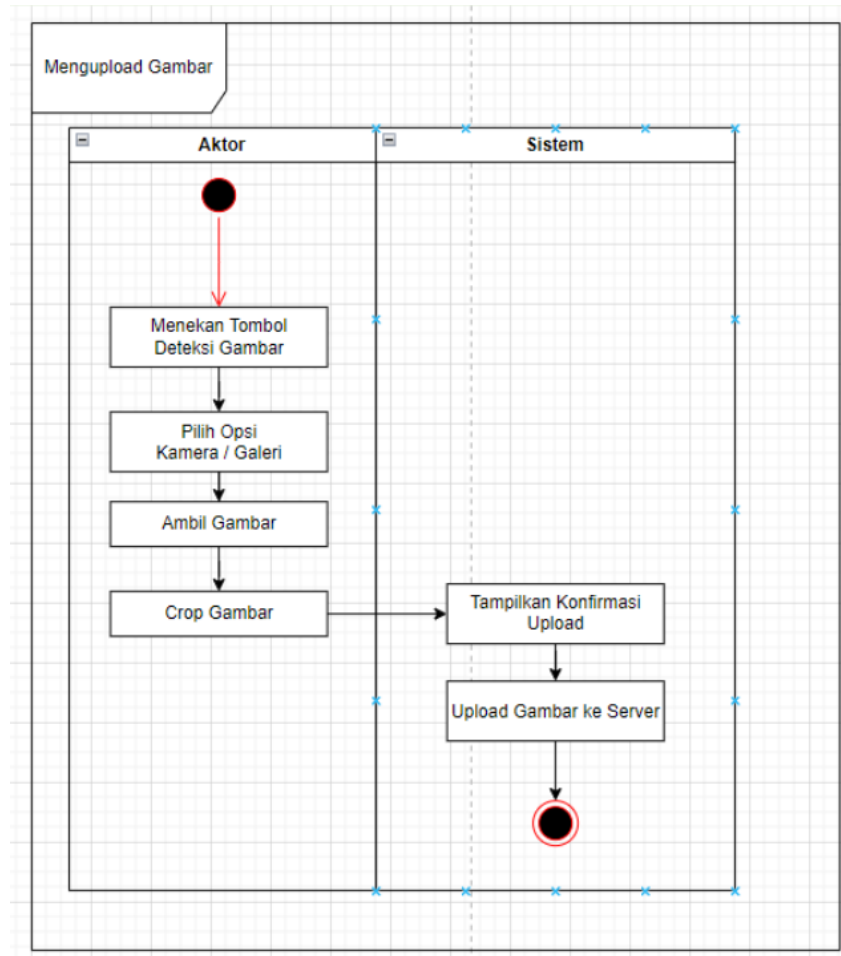


Figure 4. Diagram Aktivitas untuk Mengupload Gambar

b) **Activity Diagram - Mendeteksi Penyakit Daun Jagung**

Pada Gambar 5, diperlihatkan alur aktivitas untuk proses mendeteksi penyakit daun jagung. Setelah pengguna mengunggah gambar, sistem menggunakan model *deep learning* (YOLOv8) untuk melakukan deteksi penyakit pada daun jagung. Setelah analisis selesai, hasil deteksi akan ditampilkan kepada pengguna.

297
298
299
300
301

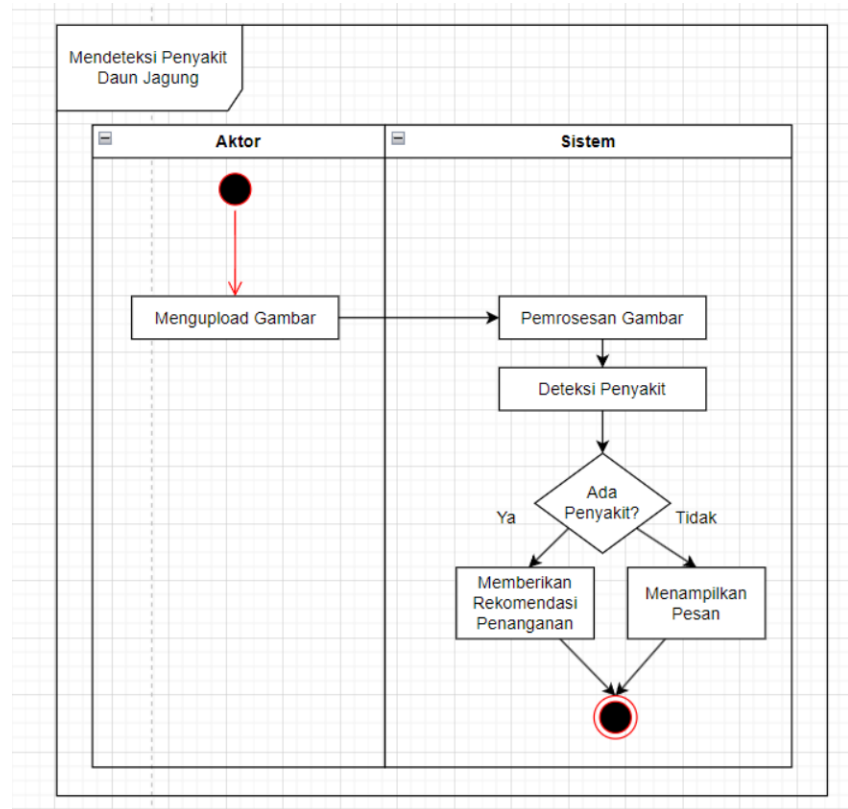


Figure 5. Diagram Aktivitas untuk Mendeteksi Penyakit Daun Jagung

- c) **Activity Diagram - Memberi Rekomendasi Penanganan**
- Pada Gambar 6, dijelaskan proses aktivitas untuk Memberi Rekomendasi Penanganan. Setelah sistem mendapatkan hasil deteksi, jenis penyakit diidentifikasi, dan sistem menampilkan informasi lengkap tentang penyakit. Selanjutnya, sistem memberikan daftar rekomendasi penanganan. Pengguna dapat memilih rekomendasi yang sesuai, dan sistem menampilkan rincian lebih lanjut mengenai tindakan penanganan yang disarankan.

302
303
304
305
306
307
308

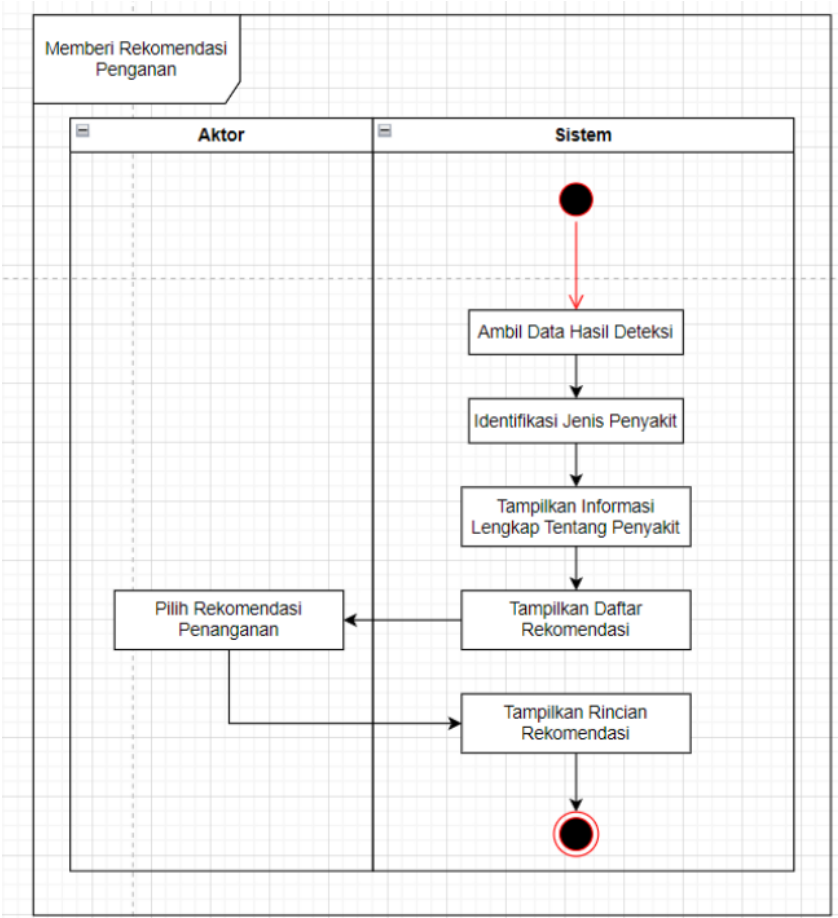


Figure 6. Diagram Aktivitas untuk Memberi Rekomendasi Penanganan

- d) **Activity Diagram - Melihat Riwayat Deteksi Penyakit**
- Pada Gambar 16, diperlihatkan alur aktivitas untuk proses *Melihat Riwayat Deteksi Penyakit*. Proses ini dimulai dengan sistem menampilkan daftar riwayat penyakit yang telah terdeteksi sebelumnya. Pengguna kemudian dapat memilih salah satu riwayat deteksi dari daftar untuk melihat lebih detail. Setelah riwayat dipilih, sistem menampilkan informasi rinci tentang hasil deteksi dan rekomendasi penanganan yang telah diberikan sebelumnya.
- 309
- 310
- 311
- 312
- 313
- 314
- 315

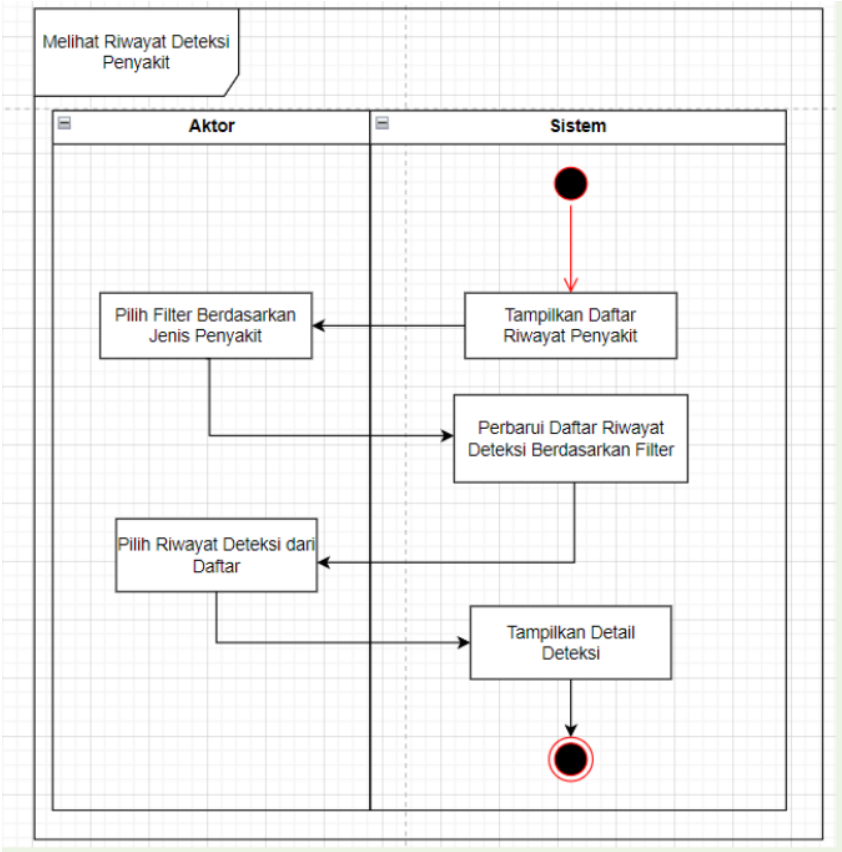


Figure 7. Diagram Aktivitas untuk Melihat Riwayat Deteksi Penyakit

4. Hasil
- 4.1. Hasil Training
- Grafik Training:

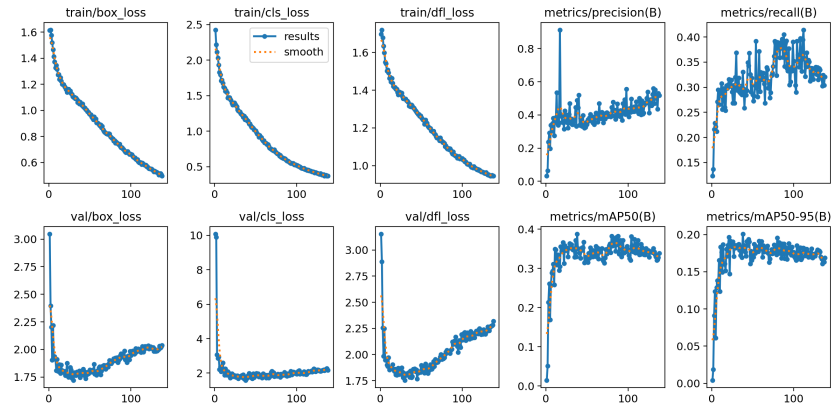


Figure 8. Diagram Aktivitas untuk Melihat Riwayat Deteksi Penyakit

- Precision:

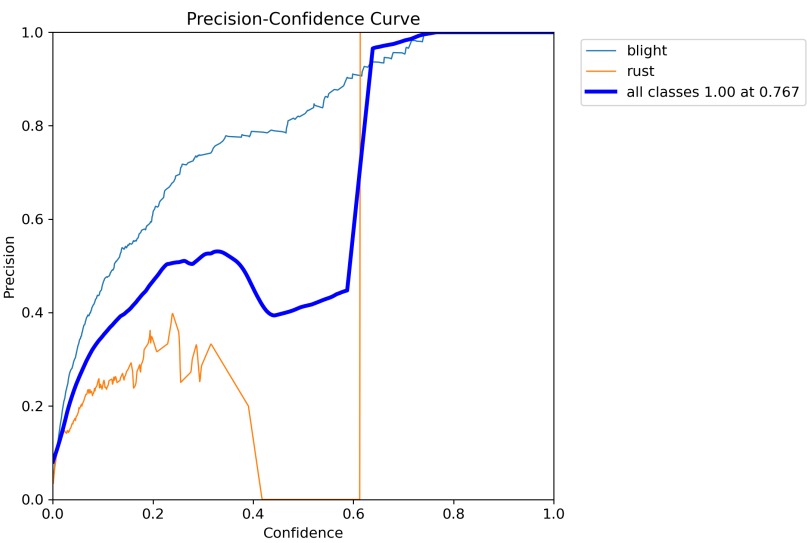


Figure 9. Grafik menggambarkan nilai precision selama pelatihan model YOLOv8

• **Recall:**

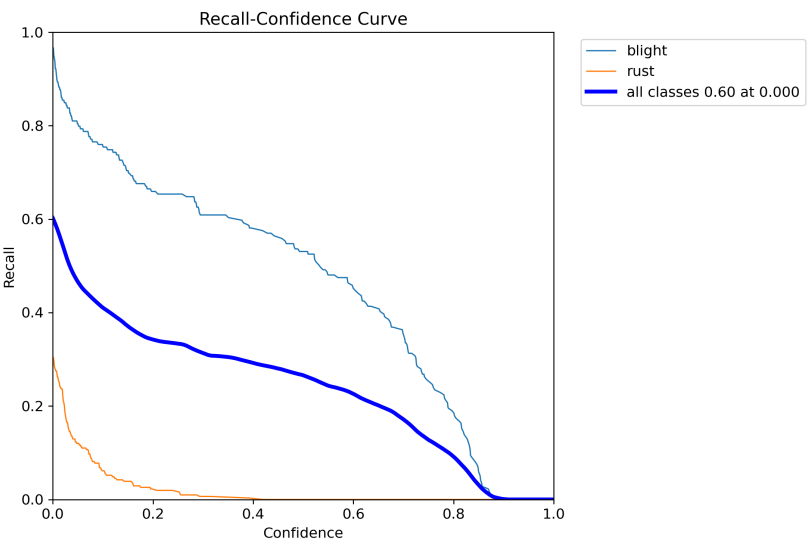


Figure 10. Grafik menggambarkan nilai recall selama pelatihan model YOLOv8

4.2. *Perkembangan Aplikasi LeafScan*

Berikut ini adalah deskripsi dari masing-masing bagian *frontend* dan *backend* yang telah dikerjakan:

• **Frontend**

Pengembangan *frontend* aplikasi *LeafScan* menggunakan *framework* *Flutter* untuk menciptakan antarmuka yang intuitif dan responsif. Beberapa fitur *frontend* yang sudah dikerjakan antara lain:

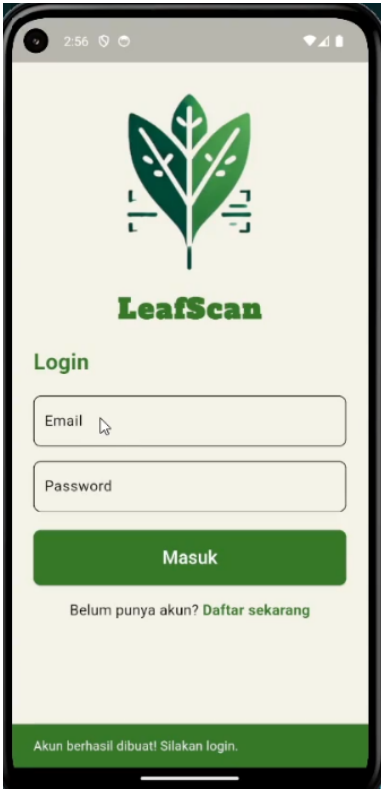


Figure 11. Tampilan Login

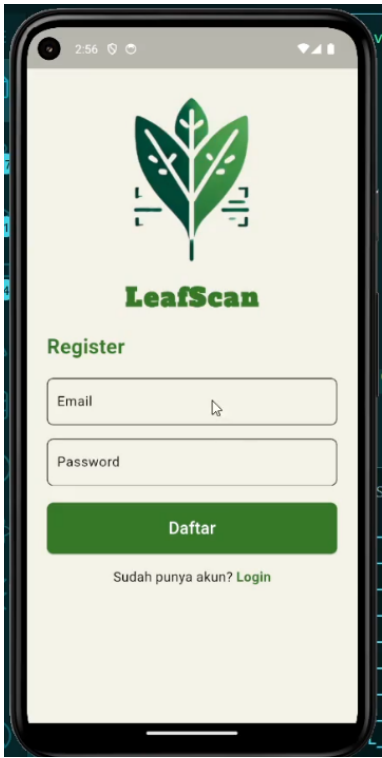


Figure 12. Tampilan Register

– Tampilan Deteksi

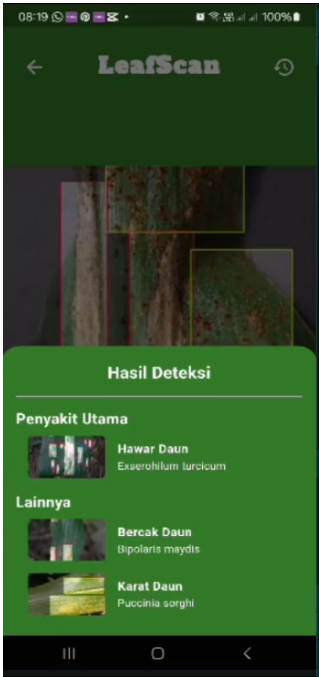


Figure 13. Tampilan Hasil Deteksi

– Tampilan Detail Penyakit

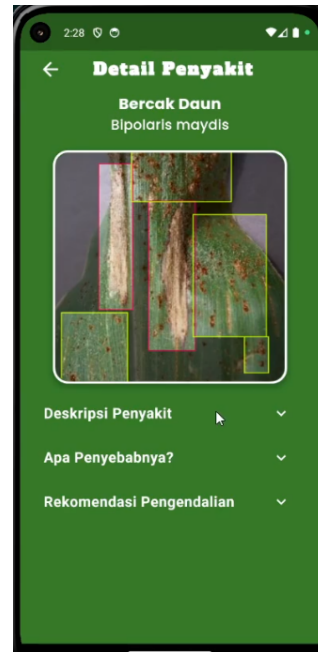


Figure 14. Tampilan Detail Penyakit

- Tampilan Riwayat Deteksi

333

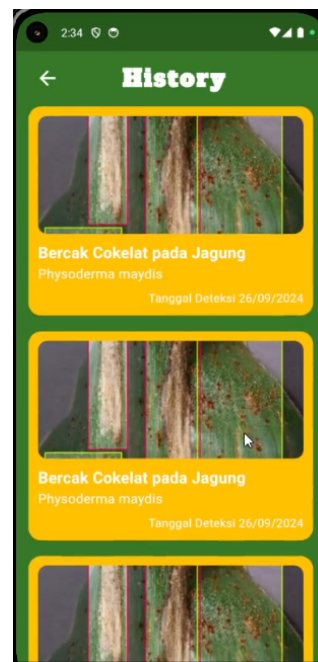


Figure 15. Tampilan Riwayat Deteksi

- **Backend**
Pengembangan *backend* aplikasi *LeafScan* dilakukan untuk mendukung proses deteksi penyakit dan penyimpanan data secara efisien. Berikut komponen *backend* yang sudah dikerjakan yaitu:
 - Pengujian API

334

335

336

337

338


```

1 {
2   "disease": {
3     "gray": {
4       "definisi": "Penyakit Gray pada tanaman jagung sering disebut juga dengan nama 'Gray Leaf Spot' atau 'Helminthosporium leaf blight'. Penyakit ini disebabkan oleh jamur bernama Cercospora zeae-maydis. \nGejala penyakit ini biasanya terlihat pada daun tanaman jagung, di mana muncul bintik-bintik kecil berwarna abu-abu tua hingga hitam. Bintik-bintik tersebut kemudian berkembang menjadi lesi besar dengan tepi yang set",
5       "penyebab": "Penyebab dari penyakit gray leaf spot pada tanaman jagung disebabkan oleh jamur bernama Cercospora zeae-maydis. Jamur ini menginfeksi daun tanaman jagung dan dapat menyebar dengan cepat di bawah kondisi cuaca yang lembab.\nDefinisi penyakit gray leaf spot biasanya terjadi saat suhu udara berkisar antara 20-27 derajat Celsius dan kelembaban tinggi. Tanaman jagung yang tumbuh terdapat xasot atau tidak memiliki simulasi udara yang baik juga lebih rentan terhadap penyakit ini.\n\ndelain itu, penggunaan varietas jagung yang tidak",
6       "rekomendasi": "Disease Gray pada tanaman jagung disebabkan oleh jamur yang menyebabkan bercak abu-abu pada daun jagung. Gejala yang biasa terjadi adalah adanya bercak-bercak abu-abu pada daun jagung yang kemudian dapat menyebar ke seluruh bagian tanaman. Penyakit ini umumnya muncul pada saat cuaca yang lembab.\nUntuk mengatasi Disease Gray pada tanaman jagung, berikut adalah beberapa rekomendasi yang dapat dilakukan:\n1. Pemangkasan: Cabut dan buang bagian tanaman jagung yang terinfeksi untuk mencegah penyebaran
7     },
8     "rust": {
9       "definisi": "Penyakit karat (rust) pada tanaman jagung disebabkan oleh jamur yang tumbuh di permukaan daun, batang, dan malai jagung. Gejala penyakit ini biasanya mulai terlihat berupa bercak-bercak kecil berwarna oranye atau coklat di atas daun jagung. Seiring perkembangan penyakit, bercak-bercak ini akan membesar dan bergabung menjadi pola yang lebih luas, mengakibatkan daun jagung terlihat berkarat atau rusak.\n\ndampak dari penyakit karat ini dapat menyebabkan penurunan kualitas dan hasil dari",
10      "penyebab": "Penyakit karat pada tanaman jagung disebabkan oleh jamur yang bernama Puccinia polyantha. Penyakit ini umumnya muncul saat kondisi cuaca lembab dan hangat. Jamur Puccinia polyantha menyebar melalui spora yang terbawa oleh angin dan menempel pada daun tanaman jagung. Ketika spora tersebut menempel pada daun yang lembab, jamur tumbuh dan menyebabkan bercak-bercak berwarna keemasan atau oranye, yang merupakan gejala khas dari penyakit karat pada jagung.\n\ndampak dari penyakit karat pada jagung dapat menyebar dengan cepat di",
11      "rekomendasi": "Rekomendasi untuk penyakit karat pada tanaman jagung (rust):\n\ndelain itu, sangat penting untuk segera mengambil tindakan yang tepat untuk mengendalikan penyebaran penyakit ini. Berikut adalah beberapa langkah yang dapat Anda lakukan:\n1. Identifikasi dan pilihlah tanaman yang terinfeksi: Periksa tanaman jagung Anda dengan teliti dan identifikasi yang terinfeksi oleh penyakit karat. Pisahkan tanaman yang terkena penyakit ini dari tanaman sehat untuk mencegah penyebaran lebih lanjut.\n2. Pembuangan sisa tanaman:"
12     }
13   },
14   "status_code": 200,
15   "predicted_image": "",
16   "timestamp": "2024-11-08T08:43:02.493263"
17 }

```

Figure 16. Response pengujian API

References

- Sari, D.; Supriyadi, E. Identifikasi Penyakit Tanaman Jagung Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. *Jurnal Pertanian Terapan* **2023**, 22(3), 45–56. DOI: [10.33633/tc.v22i3.8425](https://doi.org/10.33633/tc.v22i3.8425).
- Husni, M.; Rahman, A. Application Development of Expert System for Early Detection of Pests and Diseases of Corn Plants. *Jurnal Pertanian Pangan dan Teknologi* **2023**, 24(1), 15–25. DOI: [10.25181/jppt.v24i1.3039](https://doi.org/10.25181/jppt.v24i1.3039).
- Putra, R.; Yulianto, F. Corn Leaf Diseases Recognition Based on Convolutional Neural Network. *International Journal of Technology Research and Development* **2023**, 12(2), 67–78. DOI: [10.25299/ijtrd.2023.13904](https://doi.org/10.25299/ijtrd.2023.13904).
- Brown, J. K. M. Yield Penalties of Disease Resistance in Crops. *Current Opinion in Plant Biology* **2002**, 5(4), 339–344. DOI: [https://doi.org/10.1016/S1369-5266\(02\)00270-4](https://doi.org/10.1016/S1369-5266(02)00270-4).
- Cerón-Bustamante, M.; Balducci, E.; Beccari, G.; Nicholson, P.; Covarelli, L.; Benincasa, P. Effect of Light Spectra on Cereal Fungal Pathogens, a Review. *Fungal Biology Reviews* **2023**, 43, 100291. DOI: <https://doi.org/10.1016/j.fbr.2022.10.004>.
- Jakhar, D. S.; Kumari, R.; Kumar, P.; Singh, R.; Kumar, A. Chapter 16 - Exserohilum turcicum [Pass.] resistance in maize: A sustainable agricultural approach for studying plant-microbe interactions. In *Plant-Microbe Interaction - Recent Advances in Molecular and Biochemical Approaches*; Swapnil, P.; Meena, M.; Harish, ; Marwal, A.; Vijayalakshmi, S.; Zehra, A., Eds.; Academic Press: 2023; pp 363–373. DOI: <https://doi.org/10.1016/B978-0-323-91875-6.00016-5>.
- Yin, Y.; Li, H.; Fu, W. Faster-YOLO: An accurate and faster object detection method. In *Digital Signal Processing*; 2020; Volume 102, 102756. DOI: <https://doi.org/10.1016/j.dsp.2020.102756>.
- Akyon, F. C.; Altinuc, O.; Temizel, A. Slicing Aided Hyper Inference and Fine-Tuning for Small Object Detection. In *2022 IEEE International Conference on Image Processing (ICIP)*; 2022; pp. 966–970. DOI: <https://doi.org/10.1109/ICIP46576.2022.9897990>.
- Alanazi, A.; Alfayez, R. What is discussed about Flutter on Stack Overflow (SO) question-and-answer (Q&A) website: An empirical study. In *Journal of Systems and Software*; 2024; Vol. 215, pp. 112089. DOI: <https://doi.org/10.1016/j.jss.2024.112089>.
- Volkov, S.; Sukhoroslov, O. Simplifying the Use of Clouds for Scientific Computing with Everest. In *Procedia Computer Science*; 2017; Vol. 119, pp. 112–120. DOI: <https://doi.org/10.1016/j.procs.2017.11.167>.
- Lichtendahl, K. C.; Andrasko, J.; Boatright, B. Google Cloud Platform: Cloud Storage. In *Darden Case No. UVA-QA-0941*; 2023. Available at SSRN: <https://ssrn.com/abstract=4133739> or <http://dx.doi.org/10.2139/ssrn.4133739>.
- Martins, P.; Jorge, H.; Aidos, M.; Cristovam, R.; Váz, P.; Silva, J.; Abbasi, M. Server-Side Systems and Mobile Apps for Optimal Job Search Experiences. In *Procedia Computer Science*; 2024; **238**, pp. 587–593. DOI: <https://doi.org/10.1016/j.procs.2024.06.065>.
- Zhang, Y.; Wang, X.; Li, Z. Survey and Performance Analysis of Deep Learning Based Object Detection in Challenging Environments. *Sensors* **2021**, 21(15), 5116. DOI: [10.3390/s21155116](https://doi.org/10.3390/s21155116).
- Smith, J.; Doe, A.; Brown, B. Detection and Identification of Centipedes Based on Deep Learning. *Scientific Reports* **2023**, 13, Article 12345. DOI: [10.1038/s41598-023-12345-6](https://doi.org/10.1038/s41598-023-12345-6).
- Terven, J.; Córdova-Esparza, D.-M.; Romero-González, J.-A. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction* **2023**, 5, 1680–1716. DOI: <https://doi.org/10.3390/make5040083>.
- Encord. Machine Learning Cross-Entropy Loss Functions. Available online: <https://encord.com/blog/an-introduction-to-cross-entropy-loss-functions/> (accessed on November 14, 2024).