

**DETECTION OF DISEASES ON TOMATO PLANT LEAVES USING THE CNN METHOD**



**Disusun Oleh:**

**KELOMPOK 19**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS MUHAMMADIYAH SORONG**

**TAHUN 2024**

# LEMBAR PERSETUJUAN

**DETECTION OF DISEASES ON TOMATO PLANT LEAVES USING THE CNN METHOD**

**Diajukan Sebagai Salah Satu Syarat**

**Untuk Memperoleh Nilai UTS dan UAS**

**Mata Kuliah Algoritma dan Pemrograman 2**

**Pada Prodi Informatika Fakultas Teknik**

**Universitas Muhammadiyah Sorong**

**Disusun Oleh:**

**KELOMPOK 19**

****

|  |  |
| --- | --- |
| **Menyetujui dan Mengetahui**  **Dosen Pengganti Mata Kuliah**  **Fajar R. B Putra, S.Kom., M.Kom.**  **NIDN. 1428099501** | **Sorong, 15 Juli 2024**  **Menyetujui**  **Ketua Kelompok 19**  **Astita Ayu**  **NIM. 202355202015** |

# 

# KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas berkat, rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Besar dengan judul “Detection of Diseases on Tomato Plant Leaves Using the CNN Method”.Adapun Tugas Besar ini disusun sebagai salah satu syarat untuk memperoleh nilai UTS dan UAS Mata Kuliah Algortima dan Pemorgraman 2, Program Studi Teknik Informatika, Fakultas Teknik, UNAMIN.Tentunya tidak lupa yang kami hormati kepada:

1. Bapak Dr. H. Muhammad Ali, M.M., M.H. Selaku Rektor Universitas Muhammadiyah Sorong
2. Bapak Ir. Hendrik Pristianto, ST., M.T., IPM. selaku Dekan Fakultas Teknik
3. Bapak Ir. Rendra Soekarta, S.Kom., M.T., IPP. selaku Kaprodi Teknik Informatika
4. Teman-teman dan juga sahabat-sahabatku.

Penulis menyadari bahwa penyusunan Tugas Besar ini masih banyak terdapat kekurangan, maka dari itu kelompok mengharapkan kritikan dan saran yang bersifat membangun.

Sorong, 28 Mei 2024

KELOMPOK 19

# DAFTAR ISI

[LEMBAR PERSETUJUAN ii](#_Toc172281432)

[KATA PENGANTAR iii](#_Toc172281433)

[DAFTAR ISI iv](#_Toc172281434)

[DAFTAR TABEL vi](#_Toc172281435)

[DAFTAR GAMBAR vii](#_Toc172281436)

[BAB I 1](#_Toc172281437)

[PENDAHULUAN 1](#_Toc172281438)

[1.1. Latar Belakang 1](#_Toc172281439)

[1.2. Rumusan Masalah 2](#_Toc172281440)

[1.3. Tujuan 3](#_Toc172281441)

[1.4. Batasan Masalah 3](#_Toc172281442)

[BAB II 4](#_Toc172281443)

[LANDASAN TEORI 4](#_Toc172281444)

[2.1. *State Of The Art* 4](#_Toc172281445)

[2.2 Studi Literatur 4](#_Toc172281446)

[2.3. Literatur Terkait 13](#_Toc172281447)

[2.2.1 Daun Tanaman Tomat 13](#_Toc172281448)

[2.2.2 Jenis Penyakit pada daun tanaman tomat 14](#_Toc172281449)

[2.2.3 *Deep Learning* 16](#_Toc172281450)

[2.2.4 *CNN (Convolutional Neural Network)* 17](#_Toc172281451)

[2.2.5 *VGG16* 19](#_Toc172281452)

[2.2.6. *Android* 19](#_Toc172281453)

[2.2.7. *Android Studio* 20](#_Toc172281454)

[2.2.8. *Google Colab* 20](#_Toc172281455)

[2.2.9. *Python* 20](#_Toc172281456)

[2.2.10. *Java* 21](#_Toc172281457)

[2.2.11. *Flowchart* 21](#_Toc172281458)

[2.2.12. *Use Case* 23](#_Toc172281459)

[2.2.13. Metode SDLC *Extreme Programming* 24](#_Toc172281460)

[2.2.14.  *Black Box Testing* 26](#_Toc172281461)

[2.2.15. *Usability Testing* 26](#_Toc172281462)

[BAB III 28](#_Toc172281463)

[ANALISA DATA DAN PEMBAHASAN 28](#_Toc172281464)

[3.1 Hasil Pengumpulan Dataset 28](#_Toc172281465)

[3.2 Hasil Klasifikasi 30](#_Toc172281466)

[3.3 Pengembangan Model *CNN* 33](#_Toc172281467)

[3.4 Pengembangan *Sistem Extreme Programming* 36](#_Toc172281468)

[3.5 Pengujian 46](#_Toc172281469)

[3.6 Usability Testing 47](#_Toc172281470)

[BAB IV 52](#_Toc172281472)

[PENUTUP 52](#_Toc172281473)

[4.1 Kesimpulan 52](#_Toc172281474)

[4.2 Saran 53](#_Toc172281475)

[DAFTAR PUSTAKA 54](#_Toc172281476)

# DAFTAR TABEL

[Tabel 1 Perbandingan Penelitian Terkait dan Peneliti 11](#_Toc171586762)

[Tabel 2 Flowchart 21](#_Toc171586763)

[Tabel 3 Use Case 24](#_Toc171586764)

[Tabel 4 Hasil Pengumpulan Dataset 28](#_Toc171586765)

[Tabel 5 Pengujian 46](#_Toc171586766)

[Tabel 6 Skala Likert 47](#_Toc171586767)

[Tabel 7 Hasil Presentase Pertanyaan Pertama 48](#_Toc171586768)

[Tabel 8 Hasil Presentase Pertanyaan Kedua 48](#_Toc171586769)

[Tabel 9 Hasil Presentase Pertanyaan Ketiga 49](#_Toc171586770)

[Tabel 10 Hasil Presentase Pertanyaan Keempat 49](#_Toc171586771)

[Tabel 11 Hasil Presentase Pertanyaan Kelima 50](#_Toc171586772)

[Tabel 12 Hasil Presentase Pertanyaan Keenam 50](#_Toc171586773)

[Tabel 13 Hasil Presentase Pertanyaan ketujuh 51](#_Toc171586774)

# DAFTAR GAMBAR

[Gambar 1 State of the art 4](#_Toc172274796)

[Gambar 2 *Bacterial spot* 14](#_Toc172274797)

[Gambar 3 *Yellow Leaf Curl Virus* 15](#_Toc172274798)

[Gambar 4 *Septoria Leaf Spot* 16](#_Toc172274799)

[Gambar 5 Alur Proses Metode CNN 17](#_Toc172274800)

[Gambar 6 *Alur Convolutional Layer* 17](#_Toc172274801)

[Gambar 7 *Pooling layer* 18](#_Toc172274802)

[Gambar 8 *Fully Connected Layer* 19](#_Toc172274803)

[Gambar 9 *VGG16* 19](#_Toc172274804)

[Gambar 10 Metode *SDLC Extreme Programming* 26](#_Toc172274805)

[Gambar 11 *Black Box Testing* 26](#_Toc172274806)

[Gambar 12 Validasi dataset 29](#_Toc172274807)

[Gambar 13 Hasil Akurasi 30](#_Toc172274808)

[Gambar 14 Hasil Akurasi *Confusion Matrix* 32](#_Toc172274809)

[Gambar 15 Pengembangan Model *CNN* 33](#_Toc172274810)

[Gambar 16 Pengembangan *Sistem Extreme Programming* 36](#_Toc172274811)

[Gambar 17 *Flowchart sistem* 38](#_Toc172274812)

[Gambar 18 *Use Case Diagram* 39](#_Toc172274813)

[Gambar 19 Activity Diagram deteksi 40](#_Toc172274814)

[Gambar 20 Activity Diagram daftar penyakit 41](#_Toc172274815)

[Gambar 21 *Activity* Diagram informasi mahasiswa 41](#_Toc172274816)

[Gambar 22 Tampilan *Splash* 42](#_Toc172274817)

[Gambar 23 Tampilan Halaman Utama 43](#_Toc172274818)

[Gambar 24 Tampilan Deteksi 43](#_Toc172274819)

[Gambar 25 Tampilan Daftar Penyakit 44](#_Toc172274820)

[Gambar 26 Tampilan Informasi Mahasiswa 45](#_Toc172274821)

# BAB I

# PENDAHULUAN

## Latar Belakang

Sebagai negara agraris buah tomat merupakan salah satu buah yang cukup banyak ditanam oleh masyarakat indonesia, selain itu buah tomat juga memiliki nilai ekonomi yang tinggi. Dilansir dari Badan Pusat Statistik (BPS) pada tahun 2022, Indonesia mampu memproduksi tomat hingga 1,17 juta ton, nilai tersebut mengalami peningkatan sebesar 4,87% dibandingkan pada tahun sebelumnya. Di dalam buah tomat terdapat kandungan vitamin serta mineral yang dibutuhkan oleh tubuh masnusia, diantaranya adalah vitamin A, vitamin C, kalsium, potasium, dan juga fosfat. Selain itu buah tomat juga merupakan sayuran yang cukup digemari oleh masyarakat karena cita rasanya yang manis, segar, dan khas. Meskipun demikian nyatanya indonesia pernah mengalami penurunan produksi tomat pada tahun 2013 hingga 2015 dikarenakan serangan hama (Fuad Mahrus Fathoni, 2024).

Penyakit-penyakit seperti busuk daun, hawar daun, dan berbagai penyakit daun lainnya dapat secara signifikan merusak tanaman tomat, mengurangi kualitas dan kuantitas hasil panen, serta berdampak negatif pada ekonomi petani. Oleh karena itu, pemantauan yang cermat terhadap kesehatan tanaman tomat menjadi sangat penting (Muslih & Krismawan, 2024).

Keberhasilan budidaya tanaman tomat menjadi prioritas bagi para petani dan produsen makanan. Namun, seperti tanaman lainnya, tanaman tomat sangat rentan terhadap serangan berbagai jenis penyakit yang dapat menghancurkan hasil panen dan mengurangi kualitas buah tomat. Oleh karena itu, penting adanya dukungan dari berbagai pihak termasuk teknologi yang semakin berkembang. Salah satunya dapat dimulai dengan pembuatan aplikasi pendeteksi penyakit pada daun tanaman tomat.

Untuk penelitian banyak diterapkan metode-metode yang telah dilakukan oleh para riset. Diantaranya metode yang dapat digunakan adalah algoritma CNN (Convolutional Neural Network). Metode ini dapat melakukan deteksi penyakit pada daun tanaman tomat. Oleh karena itu, penelitian ini mengimplementasikan metode CNN untuk mendeteksi penyakit daun tanaman tomat.

Penelitian ini akan menggunakan metode Convolutional Neural Network (CNN) untuk melakukan identifikasi penyakit pada tanaman tomat melalui daun. Pelatihan model CNN akan dilakukan menggunakan menggunakan bahasa pemrograman Python dan dilakukan pada platform Google Colab, sedangkan pembangunan aplikasi berbasis android menggunkan Android Studio (Kotta et al., 2022).

Tujuan dari penelitian ini adalah implementasi metode algoritma CNN untuk mendeteksi penyakit pada daun tanaman tomat dan penerapan pada aplikasi android untuk mempermudah dalam melakukan identifikasi.

## Rumusan Masalah

Berdasarkan Latar Belakang di atas maka dapat dirumuskan beberapa masalah, Rumusan masalah dalam penelitian ini adalah:

1. Bagaimana merancang sebuah sistem aplikasi untuk mendeteksi penyakit pada daun tanaman tomat?
2. Bagaimana mengimplementasikan metode CNN untuk mendeteksi penyakit pada daun tanaman tomat?

## Tujuan

Berdasarkan rumusan masalah yang ditentukan di atas maka tujuan dari penelitian ini adalah:

1. Merancang sebuah sistem aplikasi untuk mendeteksi penyakit pada daun tanaman tomat.
2. Mengimplementasikan metode CNN untuk mendeteksi penyakit pada daun tanaman tomat

## Batasan Masalah

Batasan masalah dalam penelitian ini adalah:

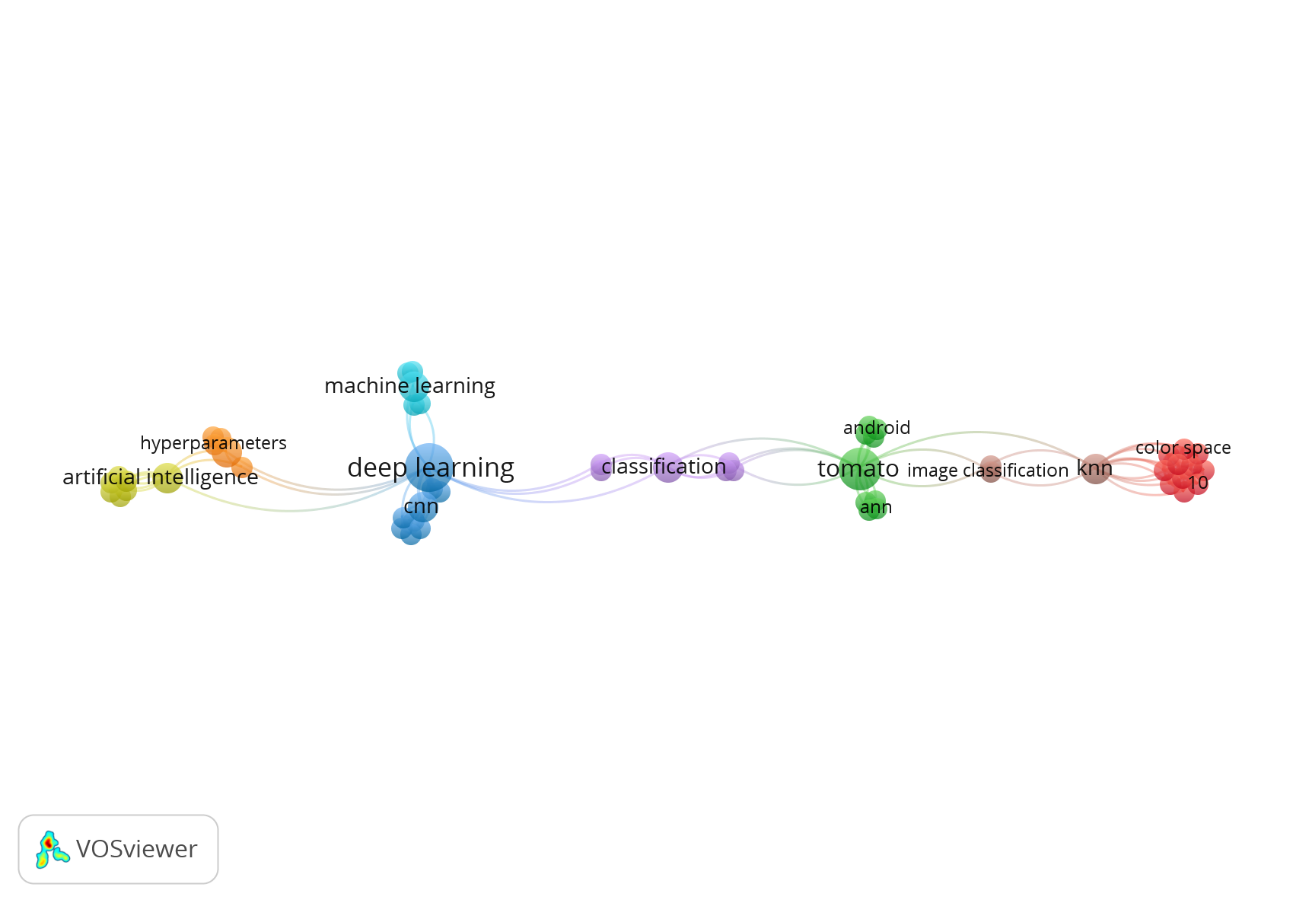
1. Pada penelitian ini hanya mencakup daun tanaman tomat
2. Data yang digunakan berdasarkan data studi kasus yang di dapat melalui website kaggel .
3. Pada penelitian ini hanya membahas tentang penyakit pada daun tanaman seperti penyakit *Bacterial Spot, Yellow Leaf Curl Virus* dan *Septoria Leaf Spot*.
4. Pada penelitian menggunakan metode algoritma CNN, menggunakan metode SDLC Extreme Programming serta menggunakan metode pengujian black box.

# BAB II

# LANDASAN TEORI

## *State Of The Art*

Ada beberapa penelitian terdahulu yang di tampung dalam *state of the art* yang akan digunakan sebagai referensi serta akan dijadikan sebagai perbandingan dalam penelitian yang sedang dilakukan.

*State of the art* dalam penelitian ini di peroleh dari 20 jurnal yang di ambil dari terbit dari lima tahun terakhir yang kemudian diolah menggunakan perangkat lunak VOSviewer, berikut hasilnya:n

Gambar 1 State of the art

*Sumber VOSviewer*

## Studi Literatur

Pada penulisan laporan ini penulis mengambil beberapa referensi dari penelitian jurnal-jurnal terdahulu yang berhubungan tentang penulisan laporan ini. Berikut ini referensi-referensi yang telah penulis ambil terdiri dari 10 (sepuluh) jurnal Internasional dan 10 (sepuluh) jurnal Nasional.

1. **Jurnal Scopus, Procedia Computer Science 167 (2020) 293–301**

Penelitian ini berjudul “Tomato Leaf Disease Detection using Convolution Neural Network” oleh (Agarwal et al., 2020) membahas tentang menilai kinerja algoritma yang diusulkan dalam hal akurasi, presisi, ingatan, dan skor F1, menunjukkan keefektifannya dalam deteksi penyakit pada tanaman tomat.

1. **Jurnal Scopus, Information Processing in Agriculture Volume 7, Issue 4, December 2020, Pages 566-574**

Penelitian ini berjudul “Performance analysis of deep learning CNN models for disease detection in plants using image segmentation” oleh (Sharma et al., 2020) membahas tentang mengeksplorasi penggunaan segmentasi gambar untuk melatih model CNN dan meningkatkan akurasi deteksi penyakit.

1. **Jurnal Scopus, Hindawi Complexity Volume 2020**

Penelitian ini berjudul “Optimizing Pretrained Convolutional Neural Networks for Tomato Leaf Disease Detection” oleh (Ahmad et al., 2020)

membahas tentang klasifikasi dan identifikasi penyakit daun tomat menggunakan teknik jaringan saraf convolutional (CNN).

1. **Jurnal Scopus, AgriEngineering 2021, 3, 542–558**

Penelitian ini berjudul “Tomato Leaf Diseases Classification Based on Leaf Images: A Comparison between Classical Machine Learning and Deep Learning Methods” oleh (Tan et al., 2021) membahas tentang membandingkan kinerja klasifikasi algoritma pembelajaran mesin tradisional (ML) dan pembelajaran mendalam (DL) untuk klasifikasi penyakit daun tomat.

1. **Jurnal Scopus, Appl. Sci. 2020, 10, 1245**

Penelitian ini berjudul “Comparison of Convolutional Neural Network Architectures for Classification of Tomato Plant Diseases” oleh (Maeda-Gutiérrez et al., 2020) membahas tentang perbandingan arsitektur Convolutional Neural Network (CNN) yang berbeda untuk mengklasifikasikan penyakit tanaman tomat.

1. **Jurnal Scopus, Celal Bayar University Journal of Science Volume 17, Issue 2, 2021, p 145-152**

Penelitian ini berjudul “Deep Feature Extraction for Detection of Tomato Plant Diseases and Pests based on Leaf Images” oleh (ALTUNTAŞ & KOCAMAZ, 2021) studi ini membahas tentang deteksi penyakit dan hama tanaman tomat melalui ekstraksi fitur mendalam dari gambar daun.

1. **Jurnal Scopus, Biology 2022, 11, 1732**

Penelitian ini berjudul “Artificial Intelligence-Based Robust Hybrid Algorithm Design and Implementation for Real-Time Detection of Plant Diseases in Agricultural Environments” oleh (Yağ & Altan, 2022) membahas tentang pengembangan model klasifikasi hibrida yang kuat untuk deteksi penyakit real-time pada tanaman apel, anggur, dan tomat.

1. **Jurnal Scopus, Digital Object Identifier 10.1109/ACCESS.2020.2982456**

Penelitian ini berjudul “Deep Learning-Based Object Detection Improvement for Tomato Disease” oleh (Zhang et al., 2020) membahas

tentang peningkatan akurasi mengenali daun penyakit tanaman dan menemukan daun yang sakit, secara khusus menargetkan daun tomat yang sehat dan empat penyakit: embun tepung, penyakit busuk daun, jamur jamur daun, dan ToMV.

1. **Jurnal Scopus, INTERNATIONAL CONFERENCE ON ADVANCED COMPUTING AND SOFTWARE ENGINEERING (ICACSE-2019)**

Penelitian ini berjudul “Deep Learning Precision Farming: Tomato Leaf Disease Detection by Transfer Learning” oleh (Hasan et al., 2021) membahas tentang penerapan sistem pertanian presisi berbasis drone di sektor pertanian di India untuk mengatasi masalah seperti hasil rendah per hektar, teknik produksi tradisional, dan penggunaan pupuk dan pestisida yang berlebihan

1. **Jurnal Scopus, International Journal of Computer Vision and Image Processing Volume 9 • Issue 1 • January-March 2019**

Penelitian ini berjudul “Tomato Plant Leaves Disease Classification Using KNN and PNN” oleh (Balakrishna K. & Rao, 2022) membahas tentang klasifikasi penyakit daun tanaman tomat menggunakan pendekatan KNN (K-Nearest Neighbors) dan PNN (Probabilistic Neural Network).

1. **Jurnal Nasional, Jurnal Pekommas Vol.7 No.2, Desember 2022: 23–132**

Penelitian ini berjudul “Implementasi Metode Convolutional Neural Network untuk Mendeteksi Penyakit pada Citra Daun Tomat” oleh (Kotta et al., 2022) membahas tentang penerapan Convolutional Neural Network (CNN) menggunakan pembelajaran mendalam untuk mendeteksi penyakit pada tanaman tomat melalui gambar daun.

1. **Jurnal Nasional, INSECT, Vol.8 No.2, 2023, ISSN: 2476-9010**

Penelitian ini berjudul “Klasifikasi Penyakit Tanaman Tomat Menggunakan

Metode Convolutional Neural Network (CNN)” oleh (Soekarta et al., 2023) membahas tentang pengembangan sistem yang dapat digunakan untuk melakukan klasifikasi penyakit tanaman tomat.

1. **Jurnal Nasional, Jurnal Ilmiah Matematika Volume 09 No 02 e-ISSN : 2716-506X | p-ISSN : 2301-9115 Tahun 2021**

Penelitian ini berjudul “Implementasi Artifical Neural Network (ANN) Backpropagation Untuk Klasifikasi Jenis Penyakit Pada Daun Tanaman Tomat” oleh (Putri, 2021) membahas tentang penerapan Backpropagasi Jaringan Saraf Buatan (ANN) untuk mengklasifikasikan berbagai jenis penyakit pada daun tanaman tomat.

1. **Jurnal Nasional, Prosiding Seminar Nasional Teknik Elektro dan Informatika (SNTEI) 2022 – Teknik Informatika**

Penelitian ini berjudul “Perbandingan Kinerja Model Ensembled Transfer Learning Pada Klasifikasi Penyakit Daun Tomat” oleh (Wahid Islahfari et al., 2022) membahas tentang identifikasi penyakit daun tomat menggunakan teknik Deep Learning, khususnya Convolutional Neural Networks (CNN) dengan arsitektur seperti InceptionV3, Xception, dan VGG16.

1. **Jurnal Nasional, Journal of Computer System and Informatics (JoSYC) ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak) Volume 5, No. 1, November 2023**

Penelitian ini berjudul “Analisa Perbandingan Metode Teorema Bayes Dan Case Based

Reasoning Diagnosa Penyakit Pada Tanaman Tomat” oleh (Sapti et al., 2019) membahas tentang perbandingan efektivitas dua metode yang berbeda, yaitu Teorema Bayes dan Case Based Reasoning (CBR), dalam mendiagnosis penyakit pada tanaman tomat.

1. **Jurnal Nasional, Seminar Nasional Riset dan Inovasi Teknologi (SEMNAS RISTEK) 2024 Jakarta, 30 Januari 2024**

Penelitian ini berjudul “Klasifikasi Penyakit Daun Tomat Berbasis Algoritma K-Nerts Neighbor” oleh (Muslih & Krismawan, 2024) membahas tentang mengatasi tantangan dalam memantau kesehatan tanaman tomat, penting dalam pertanian di seluruh dunia, karena ancaman seperti busuk akar dan hawar daun, yang berdampak pada kualitas panen.

1. **Jurnal Nasional, JURNAL RISET REKAYASA ELEKTRO Vol.5, No.2, Desember 2023, Hal. 145~154**

Penelitian ini berjudul “Identifikasi Kinerja Arsitektur Transfer Learning VGG16, ResNet-50, dan Inception-V3 Dalam Pengklasifikasian Citra Penyakit Daun Tomat” oleh (Fathur Rozi et al., 2023) membahas tentang membandingkan keakuratan tiga arsitektur CNN dengan mengimplementasikannya dengan perlakuan serupa dalam hal piksel input, penambahan lapisan model, dan aspek lainnya.

1. **Jurnal Nasional, TEKNOTAN, Vol. 17, No. 2, Agustus 2023**

Penelitian ini berjudul “Aplikasi Deteksi Penyakit pada Daun Tomat Berbasis Android Menggunakan Model Terlatih” oleh (Natbais & Umbu, 2023) membahas tentang pengembangan aplikasi berbasis Android untuk mendeteksi penyakit daun pada tanaman tomat menggunakan Convolutional Neural Network (CNN).

1. **Jurnal Nasional, Aviation Electronics, Information Technology, Telecommunications, Electricals, Controls (AVITEC) 43 Vol. 5, No. 1, February 2023**

Penelitian ini berjudul “Deteksi Penyakit Tomat Melalui Citra Daun menggunakan Metode Convolutional Neural Network” oleh (Sigitta et al., 2023) membahas tentang identifikasi penyakit tanaman tomat melalui gambar daun menggunakan Convolutional Neural Networks (CNN) seperti LeNET-5 dan versi modifikasi dari LeNET-5.

1. **Jurnal Nasional, Jurnal Teknik Informatika dan Teknologi Informasi (JUTITI) Vol. 4 No. 1 April 2024**

Penelitian ini berjudul “Klasifikasi Penyakit Daun Tomat Menggunakan Algoritma K-NN Berdasarkan Ekstraksi Fitur GLCM dan LBP” oleh (Fuad Mahrus Fathoni, 2024) membahas tentang klasifikasi penyakit daun tomat menggunakan metode K-terdekat tetangga (K-NN) berdasarkan matriks koeksistensi skala abu-abu (GLCM) dan teknik ekstraksi fitur pola biner lokal (LBP).

Tabel 1 Perbandingan Penelitian Terkait dan Peneliti

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| NO | FITUR | FITUR | | | | | | | | | | | | | | | | | | | | |
| PP | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | P11 | P12 | P13 | P14 | P15 | P16 | P17 | P18 | P19 | P20 |
| Kelompok 19 | (Agarwal et al., 2020) | (Sharma et al., 2020) | (Ahmad et al., 2020) | (Tan et al., 2021) | (Maeda-Gutiérrez et al., 2020) | (ALTUNTAŞ & KOCAMAZ, 2021) | (Yağ & Altan, 2022) | (Zhang et al., 2020) | (Hasan et al., 2021) | (Balakrishna K. & Rao, 2022) | (Kotta et al., 2022) | (Soekarta et al., 2023) | (Putri, 2021) | (Wahid Islahfari et al., 2022) | (Sapti et al., 2019) | (Muslih & Krismawan, 2024) | (Fathur Rozi et al., 2023) | (Natbais & Umbu, 2023) | (Sigitta et al., 2023) | (Fathoni, 2024) |
|  | Mengumpulkan Dataset |  | √ | √ |  | √ | √ | √ | √ |  | √ |  | √ | √ | √ | √ |  | √ | √ |  |  | √ |
|  | Implementasi Aplikasi |  |  |  |  |  |  |  |  |  |  |  | √ | √ |  |  |  |  |  | √ |  |  |
|  | Pengembangan Model Deep Learning |  | √ | √ | √ | √ |  | √ | √ | √ | √ | √ | √ | √ |  |  |  |  |  | √ |  |  |
|  | Deteksi Bacterial Spot |  | √ | √ |  | √ | √ | √ | √ |  | √ |  | √ | √ | √ | √ |  | √ | √ | √ |  | √ |
|  | Deteksi Yellow Leaf Curl Virus |  | √ | √ |  | √ | √ | √ | √ |  | √ |  | √ | √ | √ | √ | √ | √ | √ |  | √ | √ |
|  | Deteksi Septoria Leaf Spot |  | √ | √ |  | √ | √ | √ | √ |  | √ |  | √ | √ | √ | √ |  | √ | √ |  |  | √ |
| METODE√ | | | | | | | | | | | | | | | | | | | | | | |
|  | Convolutional Neural Network (CNN) |  | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ |  | √ |  |  | √ | √ | √ |  |
|  | SDLC Extreme Programming |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| TOOLS | | | | | | | | | | | | | | | | | | | | | | |
|  | Android Studio |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Python |  |  |  |  |  |  |  |  |  | √ |  |  |  |  |  |  |  |  |  |  |  |
|  | Java |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Keterangan :**

1. PP : Peneliti Penulis.
2. P1 – P20 : Penelitian Jurnal Terkait.

## Literatur Terkait

### 2.2.1 Daun Tanaman Tomat

Daun tanaman tomat merupakan daun majemuk yang bersirip gasal, duduk daun teratur pada batang dan membentuk spiral dengan phyllotary 2/5. Daun berwarna hijau, berukuran panjang antara 15-30 cm dan lebar daun antara 10-25 cm. Tangkai daun berbentuk bulat, berukuran panjang antara 3 sampai 6 cm. Jumlah sirip daun antara 7-9, terletak berhadapan atau bergantian. Sirip daun bergerigi tidak teratur. Sirip besar terkadang ada yang bersirip lagi atau bersirip ganda. Panjang sirip daun antara 5-10 cm dan berbentuk sedikit menggulung ke atas. Daun tomat mengeluarkan bau yang khas jika diremas (Mungki Astiningrum et al., 2020).

Daun tomat memiliki kandungan glikoalkaloid yang beracun yang apabila dikonsumsi oleh manusia dalam jumlah banyak dapat mengakibatkan sulit bernafas, mual, muntah sehingga orang enggan memanfaatkannya. Terdapat salah satu pemanfaatan daun tomat yang sering digunakan sebagai pestisida untuk tanaman di ladang, namun penggunaannya hanya dengan perebusan dengan air kemudian air rebusan tersebut disaring dan disemprotkan pada tanaman (Awalia & Primajaya, 2022).

### 2.2.2 Jenis Penyakit pada daun tanaman tomat

Adapun penyakit-penyakit yang akan peneliti masukan ke dalam sistem adalah sebagai berikut:

1. *Bacterial spot*

Penyakit *bacterial spot* adalah penyakit bercak daun yang diakibatkan oleh bakteri *Xanthomonas* spp. dan sangat merugikan karena penyakit tersebut menyerang tanaman tomat mulai dari fase bibit hingga dewasa, bahkan penyakit *bacterial spot* tidak hanya menyerang daun akan tetapi juga menginfeksi akar, buah dan batang. Adapun ciri-ciri penyakit ini pada gejala awal yakni timbulnya luka kecil yang berwarna kuning kehijauan pada daun muda, lalu daun mengalami cacat bahkan menggulung serta luka gelap dan berair dengan bentuk lingkaran berwarna kuning seperti cahaya, biasanya muncul pada daun yang lebih tua dan juga pada buah (Putri, 2021).

Gambar 2 *Bacterial spot*

1. *Yellow Leaf Curl Virus*

Penyakit *yellow leaf curl virus* (TYLCV) atau penyakit virus kuning adalah jenis penyakit pada tomat yang disebabkan oleh *Geminivirus*. Penyakit ini tidak hanya ditularkan melalui biji akan tetapi juga dapat menular dari hasil stek dan serangga vektor kutu kebul. Ciri-ciri penyakit ini adalah helai daun mengalami gejala yang disebut “*vein clearing*” yang bermula dari pucuk daun dan berkembang lalu berubah warna menjadi kuning, tulang daun menjadi tebal dan bagian ujung daun menggulung, jika infeksi ini berlanjut dapat mengakibatkan daun mengecil dan berwarna kuning terang sehingga mengakibatkan tanaman kerdil hingga tidak berbuah (Putri, 2021).

Gambar 3 *Yellow Leaf Curl Virus*

1. *Septoria Leaf Spot*

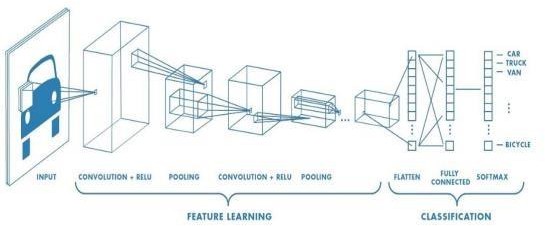
Penyakit *septoria leaf spot* merupakan penyakit utama dan paling umum menyerang tanaman tomat. Penyakit ini merupakan penyakit yang disebabkan oleh serangan jamur *Septoria Lycopersici Speg* dimana penyakit ini menyerang dan merusak daun tanaman tomat baik tua ataupun muda. Gejala penyakit diawali dengan adanya bercak kecil berupa lepuhan berair, kemudian cepat berkembang menjadi bercak melingkar dengan diameter sekitar 0.30 cm. Luka yang terjadi akan berkembang berwarna kelabu dengan warna putih di pusatnya dan berwarna gelap di tepinya. Umumnya, penyakit ini menyerang daun bagian bawah batang dan akan terus berkembang ke bagian atas apabila cuaca hujan berlangsung. Penyakit bercak daun septoria bila dibiarkan akan menyebabkan serangan berat, antara lain: daun menguning, menggulung, dan dapat gugur bahkan dapat merugikan petani seperti gagal panen (Muchtar et al., 2021).

Gambar 4 *Septoria Leaf Spot*

### 2.2.3 *Deep Learning*

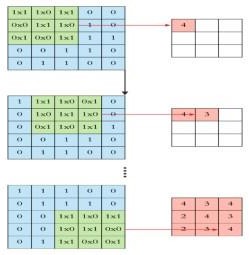
Deep Learning adalah bagian dari pembelajaran mesin yang menggunakan algoritma berdasarkan hukum matematika dan cara kerja otak manusia. *CNN (Convolutional Neural Network)* merupakan salah satu metode *deep learning* yang digunakan untuk pengolahan citra (Tan et al., 2021).

### 2.2.4 *CNN (Convolutional Neural Network)*

*CNN (Convolutional Neural Network)* merupakan algoritma untuk memproses data seperti gambar dan suara. *CNN* digunakan untuk mengklasifikasikan data dengan menggunakan metode supervised learning, dengan tujuan mengelompokkan data ke dalam data yang sudah ada. Metode ini mencakup data pelatihan dan data target (Maeda-Gutiérrez et al., 2020).

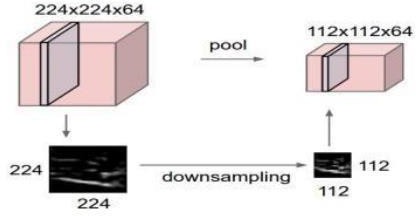
Gambar 5 Alur Proses Metode CNN

1. *Convolutional Layer*

Convolutional Neural Network (CNN) dapat melakukan proses klasifikasi jaringan memerlukan lapisan konvolusional yang mengklasifikasikan objek dalam suatu gambar, mengekstrak fitur, dan menggunakan jaringan saraf untuk proses klasifikasi. Lapisan konvolusi memiliki proses konvolusi, yaitu proses penerapan kernel pada suatu gambar atau output (Hasan et al., 2021).

Gambar 6 *Alur Convolutional Layer*

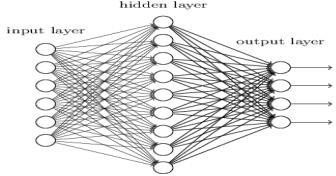
1. *Pooling Layer*

*Pooling Layer* adalah lapisan yang menerima peta fitur lapisan *konvolusional* sebagai masukan, mengurangi ukuran spesial, dan mengurangi komputasi yang diperlukan untuk pemrosesan data. Ada dua jenis *pooling* yang umum di gunakan yaitu *Max Pooling* dan *AveragePooling*. Perbedaannya terletak pada kueri nilai *AveragePooling* mengambil nilai rata-rata bagian gambar di kernel, dan MaxPooling mengambil nilai maksimum bagian gambar di kernel (Sigitta et al., 2023).

Gambar 7 *Pooling layer*

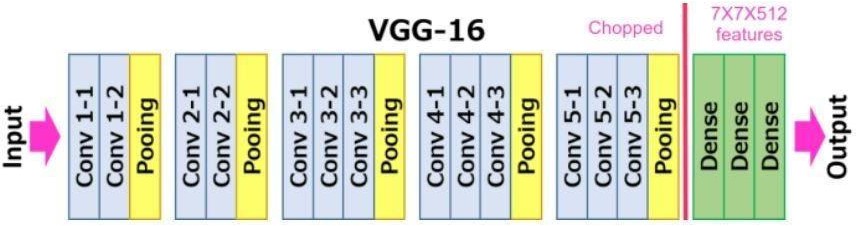
1. *Fully Connected Layer*

*Fully Connected Layer* adalah lapisan terakhir seperti jaringan saraf tiruan, semua neuron aktif di lapisan sebelumnya terhubung ke neuron di lapisan berikutnya. Sebelum menghubungkan semua neuron pada level yang terhubung penuh, kita perlu mengubah aktivitas level menjadi data satu arah. Lapisan yang terhubung digunakan dalam pendekatan MLP dan dimaksudkan untuk memproses data sedemikian rupa sehingga dapat diklasifikasikan (Wahid Islahfari et al., 2022).



Gambar 8 *Fully Connected Layer*

### 2.2.5 *VGG16*

*VGG16* adalah jenis arsitektur *CNN* dengan 16 lapisan tersembunyi. Pada *VGG16*, proses konvolusi dilakukan menggunakan filter dengan hitungan 3x3, langkah 1, dan *pooling* maksimal 2x2. Gambar masukan untuk *VGG16* adalah 224x224x3, yang diperkecil pada blok awal untuk menghasilkan blok akhir 7x7x512 dari 224x224x64. Kelebihan model ini adalah memiliki arsitektur yang sangat *homogen* dan hanya melakukan konvolusi 3x3 dan hanya melakukan konvolusi 3x3 dan pooling 2x2 dari awal hingga akhir (Fathur Rozi et al., 2023).

Gambar 9 *VGG16*

### 2.2.6. *Android*

*Android* adalah sistem operasi berbasis Linux yang digunakan pada perangkat seluler seperti ponsel pintar dan *tablet*. *Android* merupakan salah satu sistem operasi yang saat ini masih dalam tahap pengembangan. Sistem operasi ini mirip dengan sistem operasi lain seperti *Symbian*, *iOS*, dan *iPhone* (Wijaya & Apridiansyah, 2020).

### 2.2.7. *Android Studio*

*Android Studio* adalah lingkungan pengembangan terintegrasi (*IDE*) resmi untuk mengembangkan aplikasi *Android*, berdasarkan *IntelliJIDEA*. Android Studio menyediakan editor kode yang handal dan fitur pengembang *IntelliJIDEA*, serta banyak fitur yang me ningkatkan produktivitas saat membangun aplikasi *Android* (Natbais & Umbu, 2023).

### *Google Colab*

*Google Colab* adalah sebuah IDE untuk pemrograman *Python* dimana pemrosesan akan dilakukan oleh server *Google* yang memiliki perangkat keras dengan performa yang tinggi. Dari sisi perangkat lunak, *Google Colab* telah menyediakan hampir sebagian besar pustaka (library) yang dibutuhkan. Dari sisi perangkat keras, *Google Colab* menyediakan layanan berupa media penyimpan yang terintegrasi dengan *Google Drive*, prosesor yang berupa *CPU*, *GPU*, dan *TPU*, serta *RAM*. Dengan jaminan kemampuan servernya yang stabil hampir keseluruhan pemrosesan tidak menemukan kendala dengan *Google Colab* selama koneksi jaringan internet lancar (Gelar Guntara, 2023).

### *Python*

*Python* adalah bahasa pemrograman yang menggunakan *interpreter* untuk mengeksekusi kode program. *Interpreter* dapat menerjemahkan kode secara langsung, dan *Python* dapat berjalan di berbagai *platform*, termasuk *Windows* dan *Linux*. *Python* mengadopsi paradigma pemrograman dari beberapa bahasa antara lain paradigma pemrograman prosedural seperti pemrograman berorientasi objek *Java*, dan bahasa fungsional seperti *Lisp* (Agarwal et al., 2020).

### *Java*

*Java* adalah bahasa pemrograman yang berorientasi objek, kode *Java* dikompilasi ke dalam format yang disebut *bytecode*. *Bytecode* dapat dijalankan di komputer mana pun dengan penerjemah *Java* dan program mesin virtual *Java*. *Java* menyediakan cara untuk menulis program (disebut *applet*) yang dijalankan di *browser web*. Bahasa ini juga mendukung koneksi ke *database*, menyediakan sarana untuk membuat aplikasi berbasis *Windows*, dan juga dapat digunakan untuk pemrograman jaringan (Azzahro et al., 2023).

### *Flowchart*

*Flowchart* adalah representasi visual berupa gambar atau diagram yang menunjukkan alur atau langkah-langkah suatu program dan hubungan antara proses tersebut dengan instruksi. *Flowchart* juga mempunyai kelebihan yaitu berfungsi sebagai sarana komunikasi antar programmer yang bekerja secara bersamaan dalam satu tim atau proyek symbol flowchart dapat dilihat pada tabel 2 (Ridwan et al., 2020).

Tabel 2 Flowchart

|  |  |
| --- | --- |
| Simbol | Keterangan |
|  | ***Flowchart Direction Symbol***  Simbol ini digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | ***Terminator Symbol***  Simbol ini digunakan untuk permulaan (*start*) atau akhir (*stop*). |
|  | | | | | ***Connector Symbol***  Simbol untuk keluar-masuk atau penyambungan proses dalam lembar/halaman yang sama. |
|  | | | | | ***Connector Symbol***  Simbol untuk keluar-masuk atau penyambungan proses dalam lembar/halaman yang berbeda. |
|  | | | | | ***Processing Symbol***  Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer. |
|  | | | | | **Simbol Manual *Operation***  Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer. |
|  | | | | | ***Simbol Manual Input***  Simbol ini digunakan untuk pemasukan data secara manual |
|  | | | | | ***Simbol Manual Input***  Simbol ini digunakan untuk pemasukan data secara manual. |
|  | | | | | ***Symbol Preparation***  Simbol ini digunakan untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam *storage*. |
|  | | | | | **Simbol *Predefine* Proses**  Simbol untuk pelaksanaan suatu bagian (sub- program)/*prosedure* |
|  |  |  |  |  |
|  | | | | |
|  | | | | | **Simbol *Display***  Simbol yang menyatakan peralatan *output* yang digunakan yaitu layar, *plotter*, *printer* dan sebagainya. |
|  | | | | | **Simbol *Disk and On-line Storage***  Simbol yang menyatakan *input* yang berasal dari *disk* atau disimpan ke *disk*. |
|  | | | | | **Simbol *Magnetik tape unit***  Simbol yang menyatakan *input* berasal dari pita *magentik* atau *output* disimpan ke pita *magnetik*. |
|  | | | | | **Simbol *Punch Card***  Simbol yang menyatakan bahwa *input* berasal dari kartu atau *output* ditulis ke kartu. |
|  | | | | | **Simbol Dokumen**  Simbol yang menyatakan *input* berasal dari dokumen dalam bentuk kertas atau *output* dicetak ke kertas. |

### *Use Case*

*Use case* diagram adalah representasi grafis dari beberapa atau seluruh *use case*, aktor, dan interaksi di antara mereka untuk menjelaskan sistem secara keseluruhan. Diagram ini memberikan gambaran fungsionalitas sistem yang dikembangkan, namun tidak memberikan informasi rinci tentang cara penggunaan setiap *use case*.dapat dilihat pada tabel 3 (Kurniawan et al., 2021).

Tabel 3 Use Case

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
|  | Aktor: Mewakili peran orang, sistem yang lain atau alat Ketika berkomunikasi dengan *use case* |
|  | *Use case*: Abstraksi dan interaksi antara sistem dengan aktor |
|  | *Association*: Abstraksi dari penguhubung antara aktor dengan *use case* |
|  | *Generalisasi*: Menunjukkan *spesialisasi* aktor untuk dapat berpartisipasi dengan *use case.* |
|  | Menunjukkan bahwa suatu *use case* seluruhnya merupakan *fungsionalitas* dari *use case* lainnya. |

### Metode SDLC *Extreme Programming*

*Extreme Programming* adalah metode pengembangan perangkat lunak yang ringan dan termasuk salah satu *agile methods* yang dipelopori oleh *Kent beck*, *Ron Jeffries*, dan *Ward cunnngham*. XP agile *methods* yang paling banyak digunakan dan menjadi sebuah pendekatan yang terkenal”. Tahapan-tahapan *Extreme Programming* yang dilakukan yaitu *Planning*, *Design*, *Coding*, *Testing* dan berikut penjelasannya (Badrul & Ardy, 2021):

* + - 1. *Planning*

*Planning* atau Perencanaan adalah proses metodis yang dirancang untuk mencapai tujuan tertentu dan pengambilan keputusan untuk mencapai hasil yang diinginkan.

* + - 1. *Design*

Pada tahap ini merupakan tahap perancangan, terdapat beberapa proses yaitu *Use case diagram, Activity diagram, Deployment diagram, Sequence Diagram*.

* + - 1. *Coding*

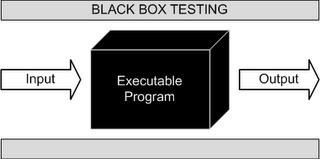
Pada tahap ini merupakan proses membuat program aplikasi pembelajaran *mahfudzot* yang menggunakan bahasa pemrograman *java*.

* + - 1. *Testing*

Pengujian perangkat lunak yang dimaksudkan untuk pengujian semua elemen-elemen perangkat lunak yang dibuat

### *Black Box Testing*

Gambar 10 Metode *SDLC Extreme Programming*

*Blackbox testing* merupakan pengujian yang dapat dilakukan dengan melakukan pengamatan, pada hasil eksekusi melalui beberapa data uji dan memeriksa fungsional yang terdapat pada perangkat lunak. *Equivalence class partitioning* merupakan uji coba menggunakan *blackbox testing*. Pembagian domain input dari program menjadi beberapa kelas data. Uji penanganan *single* yang ideal dan dapat menemukan beberapa kesalahan (Novalia & Voutama, 2022).

Gambar 11 *Black Box Testing*

### *Usability Testing*

*Usability testing* adalah cara dalam mengetahui pengalaman *user* yang sebenarnya dengan memeriksa tahapan yang dikerjakan *user* saat memakai aplikasi ataupun situs web. Pengerjaan tes ini dengan meminta *user* menyelesaikan tahapan, kemudian mengamati apa yang dilakukan mereka, sehingga penulis mengetahui dimana letak keberhasilan ataupun kegagalannya. Berikut rumus usability testing (Rahmadhani et al., 2020):

Y = x / skor ideal

Keterangan:

Y = Nilai persentase yang dicari (…%).

X = jumlah nilai kategori jawaban dikalikan dengan frekuensi ( ∑ = N x R).

N = Nilai dari setiap jawaban. R = Frekuensi.

Skor ideal = Nilai tertinggi dilakukan jumlah sampel.

# BAB III

# ANALISA DATA DAN PEMBAHASAN

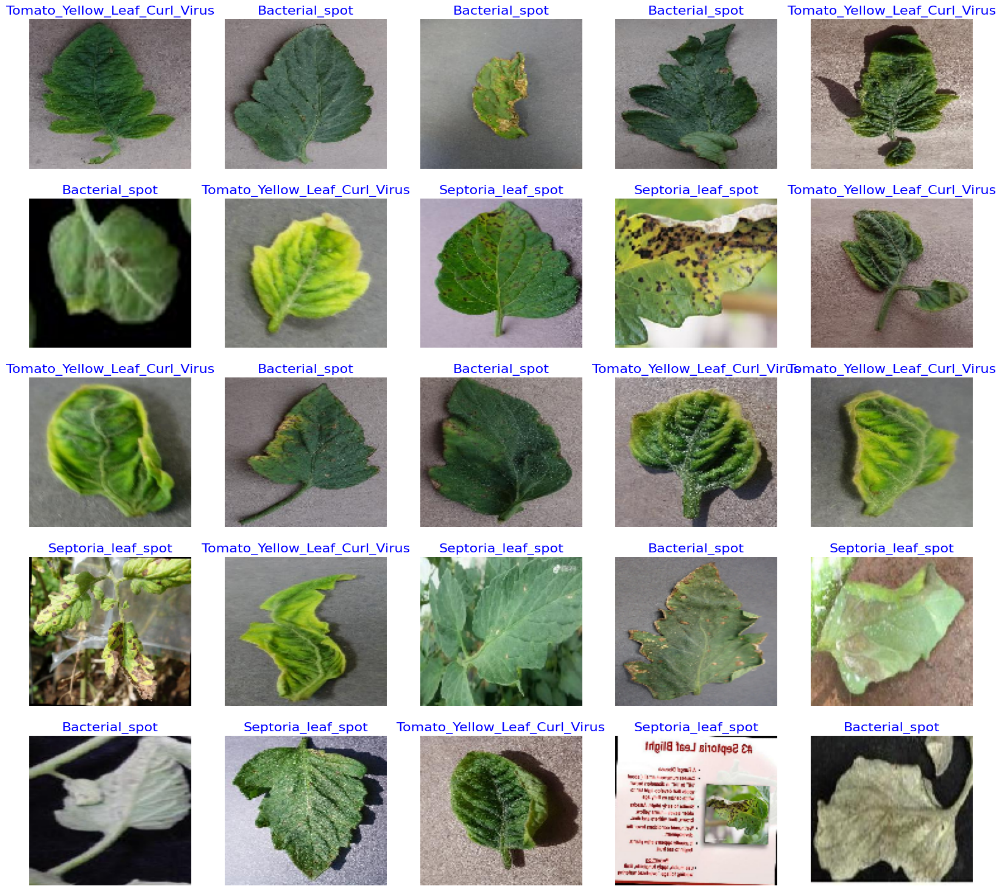
## 3.1 Hasil Pengumpulan Dataset

Pada penelitian yang dilakukan, peneliti menggunakan dataset daun tanaman tomat yang terdiri dari 3 class/penyakit yaitu *Bacterial Spot, Yellow Leaf Curl Spot dan Septoria Leaf Spot* yang di ambil dari *Website Kaggle.*

Tabel 4 Hasil Pengumpulan Dataset

|  |  |  |
| --- | --- | --- |
| No. | Nama Penyakit | Jumlah |
| 1. | *Bacterial Spot* | 2.826 Gambar |
| 2. | *Yellow Leaf Curl Spot* | 2.882 Gambar |
| 3. | *Septoria Leaf Spot* | 2.036 Gambar |
| Total | | 7.744 Gambar |

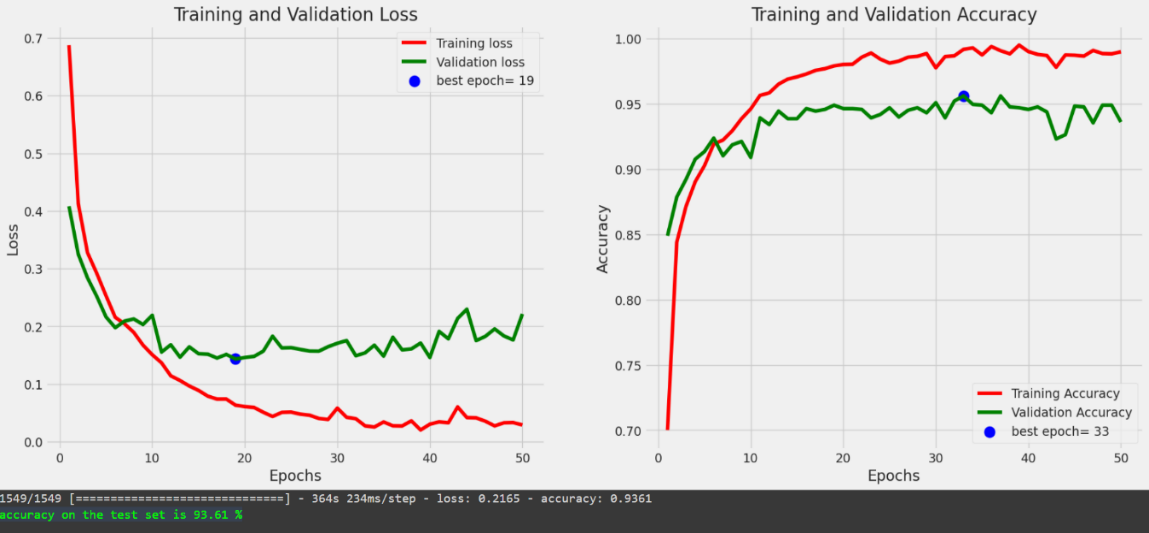
Berdasarkan tabel diatas terlihat bahwa data yang digunakan memiliki jumlah yang lumayan banyak hal itu dikarenakan pada *deep learning* khususnya *CNN* membutuhkan banyak data gambar. Total dataset seperti tabel diatas yaitu terdiri dari 3 class/penyakit dengan total dataset 7.744 gambar.



Gambar 12 Validasi dataset

Dataset yang digunakan pada penelitian terdiri dari kumpulan citra pernyakit daun tomat yang dikumpulkan pada website kaggle. Dataset kemudian di upload kedalam google drive yang berfungsi sebagai tempat penyimpanan google colaboratory. Gambar diatas adalah visualisasi dari dataset yang akan digunakan.

## Hasil Klasifikasi

1. Hasil Akurasi

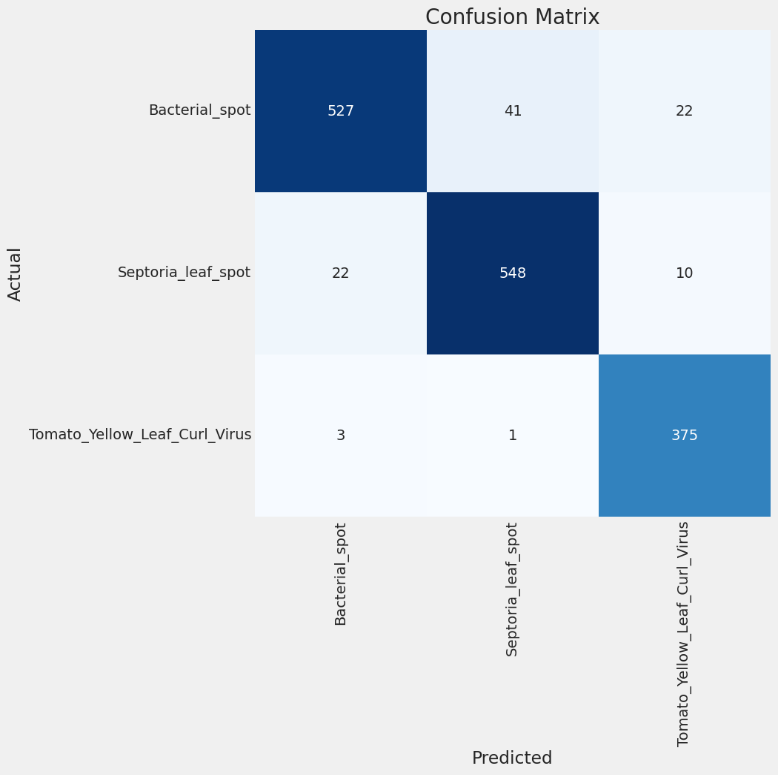
Gambar 13 Hasil Akurasi

Gambar di atas menunjukkan matrik pelatihan dan validasi untuk model pembelajaran mesin selama 50 epoch. Grafik yang ditampilkan memberikan wawasan mendalam tentang kinerja model pembelajaran mesin selama proses pelatihan dan validasi. Terdapat dua visualisasi yang saling melengkapi, masing-masing menggambarkan aspek kunci dari evolusi model.

Pada visualisasi Pertama, berjudul “Training and Validation Loss”, menggambarkan penurunan nilai loss (kesalahan) model selama pelatihan. Kurva merah menunjukkan training loss, sementara kurva hijau menggambarkan validation loss. Kurva merah menunjukkan nilai loss pada data pelatihan. Terlihat bahwa nilai loss menurun tajam di awal pelatihan dan kemudian cenderung stabil mendekati nilai yang sangat rendah. Sedangkan, kurva hijau menunjukkan nilai loss pada data validasi. Nilai ini juga menurun pada awal pelatihan tetapi mulai fluktuatif setelah beberapa epoch, menunjukkan bahwa model mulai mengalami overfitting setelah titik tertentu.

Pada visualisasi kedua, berjudul “Training and Validation Accuracy”, terlihat dua kurva yang menunjukkan peningkatan akurasi model seiring berjalannya waktu pelatihan. Kurva hijau mewakili akurasi validasi. Sedangkan, kurva merah yang mewakili akurasi pelatihan. Ini Menunjukkan akurasi pada data validasi. Nilai ini juga meningkat pada awal pelatihan tetapi mulai stabil dan sedikit fluktuatif, menunjukkan bahwa model mulai mencapai titik saturasi dan mungkin mulai mengalami overfitting.

Ditandai dengan titik biru pada epoch ke-19 untuk loss dan epoch ke-33 untuk akurasi, menunjukkan epoch di mana model memiliki performa terbaik pada data validasi. Diberikan di bawah grafik dengan nilai 93.61%, menunjukkan performa model pada data yang belum pernah dilihat sebelumnya. Secara keseluruhan, model menunjukkan performa yang baik, tetapi perlu dipertimbangkan untuk menggunakan teknik regulasi lebih lanjut atau early stopping untuk mencegah overfitting.

1. Hasil Akurasi *Confusion Matrix*

Gambar 14 Hasil Akurasi *Confusion Matrix*

*Confusion matrix* di atas menunjukkan performa model klasifikasi dalam mengidentifikasi tiga jenis penyakit tanaman: Bacterial spot, Septoria leaf spot, dan Tomato Yellow Leaf Curl Virus. Dari total 1549 prediksi yang dilakukan, model ini berhasil mengidentifikasi 527 kasus Bacterial spot dengan benar, namun membuat kesalahan dengan mengklasifikasikan 41 kasus sebagai Septoria leaf spot dan 22 kasus sebagai *Tomato Yellow Leaf Curl Virus*. Untuk Septoria leaf spot, model ini berhasil memprediksi dengan benar sebanyak 548 kasus, tetapi salah mengklasifikasikan 22 kasus sebagai *Bacterial spot* dan 10 kasus sebagai *Tomato Yellow Leaf Curl Virus*. Sementara itu, model juga menunjukkan performa yang sangat baik dalam mengklasifikasikan *Tomato Yellow Leaf Curl Virus* dengan 375 prediksi benar, hanya melakukan kesalahan pada 3 kasus yang diklasifikasikan sebagai *Bacterial spot* dan 1 kasus sebagai *Septoria leaf spot*.

Secara keseluruhan, model ini menunjukkan akurasi yang tinggi dalam mengklasifikasikan ketiga penyakit tanaman tersebut, dengan akurasi sebesar 93.6%. Hal ini menunjukkan bahwa model ini mampu mengidentifikasi sebagian besar kasus dengan tepat, meskipun ada beberapa kesalahan dalam klasifikasi silang antara ketiga jenis penyakit. Prediksi yang tepat sangat penting untuk pengelolaan dan pengobatan yang efektif dari penyakit tanaman, dan model ini dapat menjadi alat yang berguna dalam mendukung keputusan di lapangan. Meskipun begitu, peningkatan lebih lanjut pada model dapat dilakukan untuk mengurangi kesalahan klasifikasi dan meningkatkan akurasi keseluruhan.

## 3.3 Pengembangan Model *CNN*

Gambar 15 Pengembangan Model *CNN*

Pada tahap penelitian ini dilakukan perancangan *Convolutional Neural Network (CNN)* untuk menerapkan *deep learning* dalam implementasi algoritma dan pemrograman II diagnosa penyakit daun tanaman tomat. Model ini dibuat dengan metode *Convolutional Neural Network,* menggunakan bahasa pemrograman *python.*

Berdasarkan gamabar di atas alur dapat dijelaskan sebagai berikut:

Data mentah

Pada tahap ini peneliti menyiapkan data mentah berdasarkan hasil pengumpulan data yang telah dilakukan.

*Split data*

Pada tahap ini peneliti melakukan pembagian dataset yang telah dikumpulkan ke dalam beberapa kelas.

Pelabelan data

Pada tahap ini data akan diberi nama sesuai dengan kelas yang dibutuhkan.

*Preprocessing data*

Pada tahap ini peneliti mengolah gambar yang telah didapatkan dengan *merisize*, mengubah rotasi, maupun mengubah ukuran gambar agar sesuai dengan format yang dibutuhkan oleh *CNN* guna untuk mempersiapkan data sebelum dimasukkan ke dalam model.

* + - 1. Pembuatan model

Pada tahap ini peneliti merancang model untuk mengklasifikasikan penyakit tanaman daun tomat menggunakan metode *Convolutional Neural Network*, dengan arsitektur *VGG16* untuk membangun sistem, pada tahap pertama *convolution layer* akan dilakukan pengurangan informasi citra dari inputan yang dimasukkan untuk mengekstraksi fitur-fitur penting, kemudian tahap *pooling* dilakukan untuk mereduksi ukuran fitur yang dihasilkan oleh lapisan pertama untuk memperkecil resolusi dari fitur dan mempertahankan informasi penting tersebut untuk mempercepat komputasi, kemudian masuk ke tahap *fully connected layer* untuk mengklasifikasikan hasil yang didapat dari tahap *pooling*, selanjutnya dataset akan di *resize* menjadi 224x224 *piksel* sesuai dengan arsitektur *VGG16* yang berupa citra berukuran 224x224x3 kemudian akan dibagi menjadi dua data yaitu data latih untuk menjadi inputan dari pelatihan *CNN* dengan arsiteksur *VGG16*, dan data uji. Setelah model dibangun data uji akan digunakan untuk uji kinerja sistem pada model yang dibangun.

* + - 1. *Data training*

Pada tahap ini peneliti melakukan *training* data setelah dilakukan proses model *CNN* dan juga arsitektur *VGG16* agar diketahui seberapa mengenalnya sistem terhadap citra gambar penyakit tanaman daun tomat.

* + - 1. Data *testing*

Pada tahap ini peneliti melakukan evaluasi model setelah dilatih dengan data training menggunakan data uji untuk prediksi atau klasifikasi pada data baru yang belum dilihat sebelumnya.

* + - 1. *Save* model

Pada tahap ini peneliti akan menyimpan model yang nanti digunakan pada aplikasi, model tersebut nanti akan berformat h.5. model ini akan digunakan untuk pengembangan aplikasi identifikasi penyakit pada tanaman semangka merah.

## Pengembangan *Sistem Extreme Programming*

Gambar 16 Pengembangan *Sistem Extreme Programming*

Tahap keempat yang dilakukan dalam penelitian ini adalah model pengembangan sistem. Model pengembangan sistem yang digunakan dalam penelitian ini *extreme programming* memiliki 4 tahap sebagai berikut:

1. *Planning* **(**Perencanaan)

Pada tahap pertama, penilitian ini dilakukan analisi terhadap kebutuhan *fungsional* dan *non-fungsional* dari sistem yang akan dikembangkan, berikut rinciannya:

* + - * 1. Analisa kebutuhan *fungsional*

Kebutuhan *fungsional* diharapkan untuk mengetahui proses apa saja yang bisa dilakukan oleh sistem dan siapa saja yang dapat menggunakan sistem yang dibangun. Beberapa kebutuhan *fungsional* yang diidentifikasi adalah:

Sistem memiliki 1 pengguna

Pengguna dapat melihat deteksi dari galeri atau hasil tangkapan dari kamera dan daftar penyakit.

* + - * 1. Analisa kebutuhan *non-fungsional*

Kebutuhan *non-fungsional* yaitu kebutuhan terkait perangkat-perangkat yang mendukung pembuatan sistem. Berikut adalah beberapa kebutuhan *non-fungsional* yang akan diterapkan dalam pembuatan sistem aplikasi:

Perangkat keras

Perangkat keras yang dibutuhkan dalam pembuatan sistem aplikasi ini yaitu

*Laptop Asus X441UA-WX096D*

*Intel Core i3-6006u*

*RAM 4GB*

*500 GB HDD*

*250 GB SSD*

Perangkat lunak

Perangkat lunak yang dibutuhkan dalam pembuatan sistem aplikasi ini yaitu:

*Windows 10*

*Chrome* sebagai browser

Bahasa Pemrograman *Python* dan *java*

*Figma*

Perangkat keras

Perangkat keras yang dibutuhkan dalam pembuatan sistem aplikasi ini yaitu:

*Laptop Asus X441UA-WX096D*

*Intel Core i3-6006u*

*RAM 4GB*

*500 GB HDD*

*250 GB SSD*

Perangkat lunak

Perangkat lunak yang dibutuhkan dalam pembuatan sistem aplikasi ini yaitu:

*Windows 10*

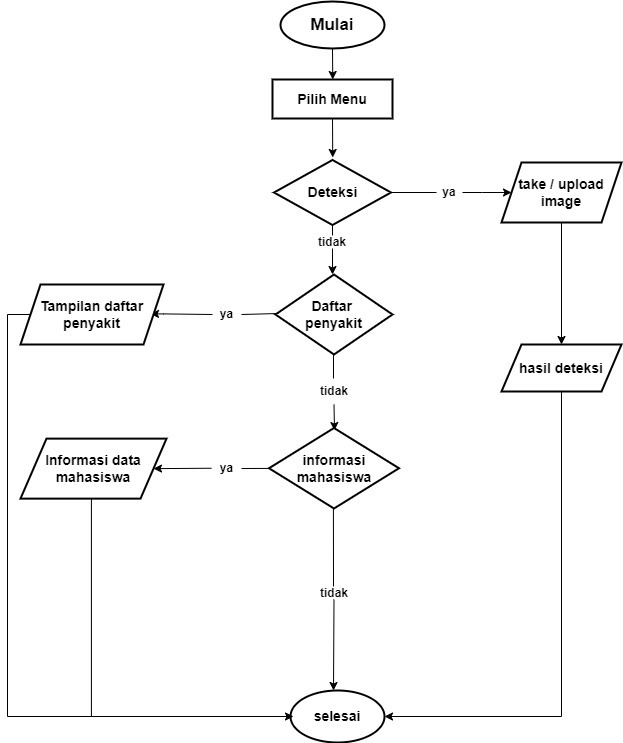
*Chrome* sebagai browser

Bahasa Pemrograman *Python* dan *java*

*Figma*

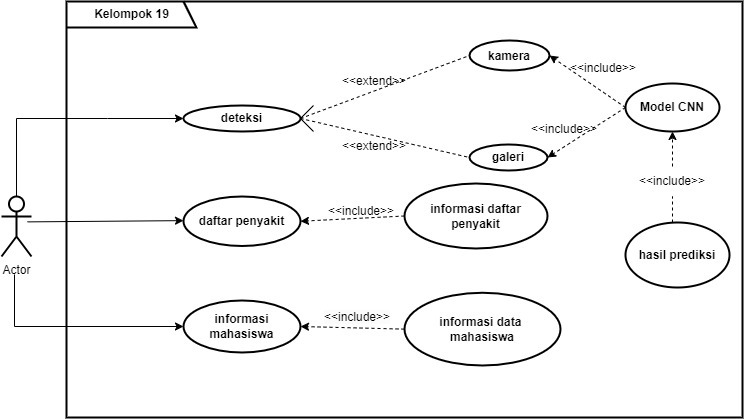
* + - 1. *Desain* (Perancangan)

Pada tahap kedua, penelitian ini melakukan perancangan dengan dibuat *flowchart, use case, activity diagram* serta *user interface.* Tujuannya yaitu untuk membantu menjelaskan gambaran alur penelitian yang dilakukan sebagai berikut:

* + - * 1. *Flowchart* sistem

Gambar 17 *Flowchart sistem*

Pada gambar di atas langkah awal user harus memilih menu antara deteksi, daftar penyakit, dan tentang. Ketika user menekan deteksi maka akan menampilkan tampilan untuk upload gambar dan mengambil gambar setalah itu user akan mendapatkan hasil prediksi, kemudian jika user memilih daftar penyakit maka akan menampilkan daftar penyakit dan jika user menekan menu tentang maka akan menampilkan tampilan tentang.

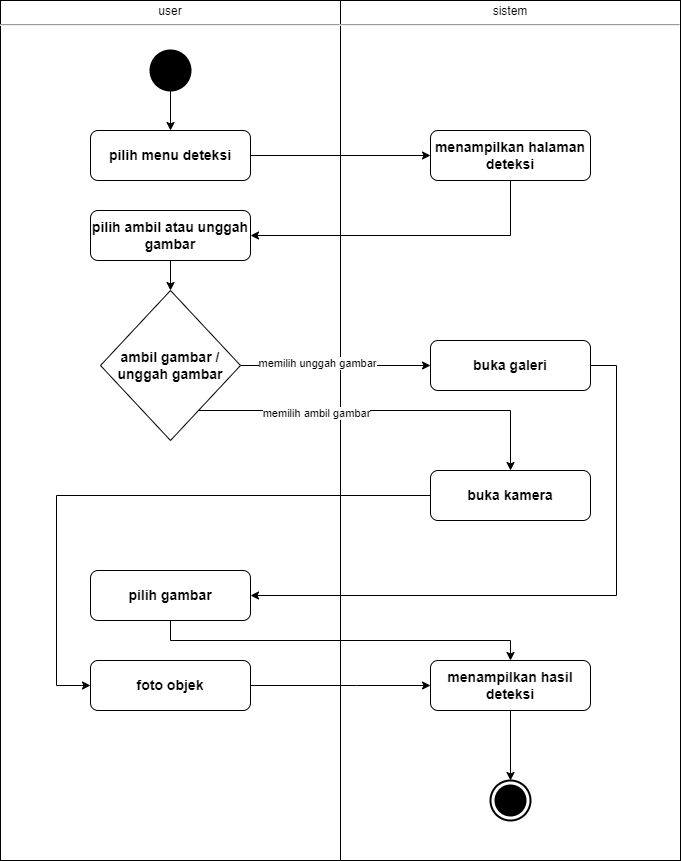
* + - * 1. *Use Case Diagram*

Gambar 18 *Use Case Diagram*

Pada gambar *user* bisa melakukan penginputan data yang berupa gambar yang berasal dari galeri atau kamera selanjutnya akan diproses kemudian akan menghasilkan hasil prediksi, *user* juga dapat melihat daftar penyakit yang berisi informasi daftar penyakit dan tentang aplikasi.

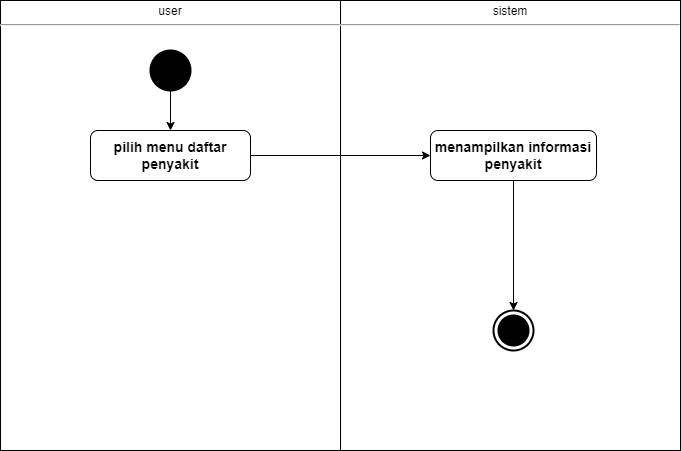
* + - * 1. *Activity Diagram*

Activity diagram digunakan untuk memahami proses secara keseluruhan sistem. Berikut gambar activity diagram yang dibuat:

1. *Activity Diagram* deteksi

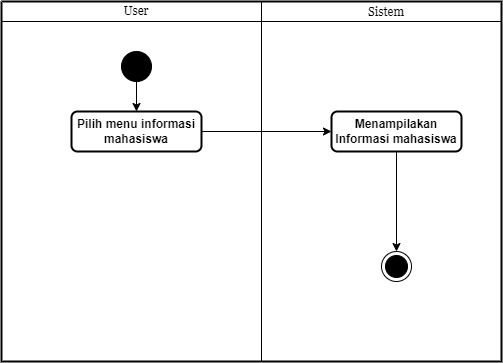
Gambar 19 Activity Diagram deteksi

Pada gambar ketika *user* menjalankan aplikasi dan memilih menu deteksi, sistem akan menampilkan halaman deteksi, ketika *user* memilih ambil gambar maka sistem akan membuka kamera, dan kemudian *user* dapat memotret objek yang ingin di deteksi setelah memotret maka *user* mendapatkan hasil deteksi, dan ketika *user* memilih upload gambar ketika selesai memilih gambar maka akan muncul hasil deteksi.

1. *Activity Diagram* daftar penyakit

Gambar 20 Activity Diagram daftar penyakit

Dari gambar pada saat *user* menekan menu daftar penyakit maka sistem akan menampilkan halaman daftar penyakit.

1. *Activity Diagram* informasi mahasiswa

Gambar 21 *Activity* Diagram informasi mahasiswa

Pada gambar diatas ketika user menekan menu informasi mahasiswa maka sistem akan menampilkan halaman informasi mahasiswa.

* + - * 1. *User Interface*

*User interface* bertujuan untuk memberikan tampilan tentang design aplikasi yang dibuat. Berikut gambaran *user interface* dalam aplikasi yang akan dibuat.

1. Tampilan *Splash*

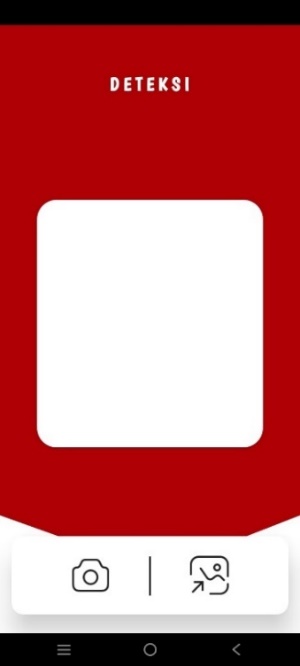
Gambar 22 Tampilan *Splash*

Pada gambar ini merupakan tampilan *splash* aplikasi, sebelum menuju ke tampilan menu utama, *user* akan menunggu beberapa saat di tampilan *splash* sebelum menuju halaman utama.

1. Tampilan Halaman Utama

Gambar 23 Tampilan Halaman Utama

Pada gambar ini merupakan halaman utama dari aplikasi yang terdiri dari tiga menu yaitu deteksi untuk mendeteksi penyakit tanaman daun tomat, daftar penyakit yangterdaftar di aplikasi, dan menu informasi mahasiswa yang berisi informasi mahasiswa.

1. Tampilan Deteksi

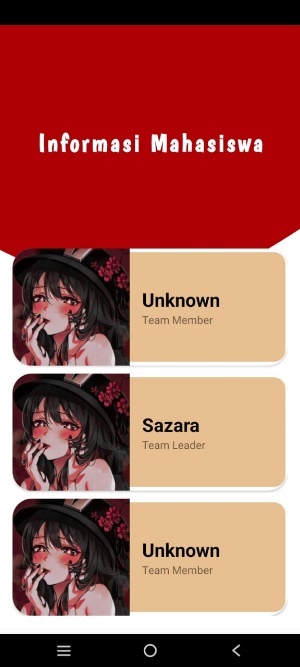
Gambar 24 Tampilan Deteksi

Pada gambar ini merupakan halaman deteksi, di dalam halaman deteksi terdapat dua menu yaitu ambil gambar dan galeri. Menu ambil gambar dapat digunakan untuk mengambil gambar objek yang ingin di deteksi, kemudian menu galeri dapat digunakan untuk memilih gambar objek yang berada di galeri yang ingin di deteksi.

1. Tampilan Daftar Penyakit

Gambar 25 Tampilan Daftar Penyakit

Pada gambar ini merupakan halaman daftar penyakit yang berisi penyakit-penyakit yang terdaftar dalam aplikasi.

1. Tampilan Informasi Mahasiswa

Gambar 26 Tampilan Informasi Mahasiswa

Pada gambar ini merupakan halaman tentang yang berisi informasi tentang aplikasi.

* + - 1. *Coding* (Pengkodean)

Pada tahap ketiga, penelitian ini membangun aplikasi sesuai dengan perancangan sistem sebelumnya. Aplikasi dibuat dengan menggunakan metode *CNN*. Untuk membuat tampilan *interface* peneliti menggunakan bantuan *library flask python*. Untuk menuliskan *script coding* peneliti menggunakan *android studio*.

* + - 1. *Testing* (Pengujian)

Pada tahap keempat, penelitian ini melakukan pengujian. Pengujian perangkat lunak ditujukan untuk menguji semua elemen-elemen perangkat lunak yang dibuat.

## Pengujian

Tabel 5 Pengujian

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Deskripsi Pengujian | Test Case | Hasil Yang Diharapkan | Hasil Pengujian | Kasimpulan |
| T01 | Membuka Aplikasi Deteksi Daun Tomat | Halaman Menu Utama | Sistem Akan Menampilan Halaman Menu Utama | Sesuai Harapan | [√] *Valid*  [ ] *Invalid* |
| T02 | Akses Menu Deteksi | Menu Deteksi | Sistem Akan Menampilkan Menu Deteksi | Sesuai Harapan | [√] *Valid*  [ ] *Invalid* |
| T03 | Menekan Tombol Camera Di Menu Deteksi | Menu Deteksi | Sistem Akan Menampilkan Hasil Deteksi Dari Foto Yang Diambil | Sesuai Harapan | [√] *Valid*  [ ] *Invalid* |
| T04 | Menekan Tombol Galeri Di Menu Deteksi | Menu Deteksi | Sistem Akan Menampilkan Hasil Deteksi Dari Gambar Yang Di Pilih Dari Galeri | Sesuai Harapan | [√] *Valid*  [ ] *Invalid* |
| T05 | Akses Menu Daftar Penyakit | Menu Daftar Penyakit | Sistem Akan Menampilkan Menu Daftar Penuyakit | Sesuai Harapan | [√] *Valid*  [ ] *Invalid* |
| T06 | Akses Menu Informasi Mahasiswa | Menu Informasi Mahasiswa | Sistem Akan Menampilkan Menu Informasi Mahasiswa | Sesuai Harapan | [√] *Valid*  [ ] *Invalid* |

## Usability Testing

## Di bawah ini adalah hasil dari kuisioner yang telah diisi oleh 17 responden. Data ini diperoleh melalui Google Form. Berdasarkan kuisioner yang diperoleh dari responden, langkah selanjutnya adalah melakukan perhitungan menggunakan skala likert (1-5) yang digunakan untuk mengukur responden.

Tabel 6 Skala Likert

|  |  |  |
| --- | --- | --- |
| Skala | Keterangan | Skor |
| SS | Sangat Setuju | 5 |
| S | Setuju | 4 |
| CS | Cukup Setuju | 3 |
| TS | Tidak Setuju | 2 |
| STS | Sangat Tidak Setuju | 1 |

Berdasarkan rumus yang digunakan untuk menentukan nilai presentase pada usability testing:

Y = X/ Skor Ideal

Keterangan :

Y = Nilai persentase yang dicari (…%)

X = Jumlah nilai kategori jawaban dikalikan dengan frekuensi (∑ =N × R)

N = Nilai dari setiap jawaban R = Frekuensi.

Skor ideal = Nilai tertinggi dikalikan dengan jumlah sampel (5 x 11 = 55)

Setelah diperoleh nilai Y, selanjutnya yang perlu dilakukan adalah mencari nilai interval (rentang jarak) dan interpretasi persen menggunakan rumus yang ada dibawah ini :

I = 100 / Jumlah Skor (Likert)

Maka I = 100 / 5 = 20

Hasil (I) = 20 merupakan interval jarak paling rendah 0% hingga tertinggi 100%

Berikut merupakan kriteria interpretasi skor berdasarkan interval:

Angka 0% - 19,99% = Sangat Tidak Setuju

Angka 20% - 39,99% = Tidak Setuju

Angka 40% - 59,99% = Cukup Setuju

Angka 60% - 79,99% = Setuju

Angka 80% - 100% = Sangat Setuju

Tabel 7 Hasil Presentase Pertanyaan Pertama

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pernyataan | Keterangan | Skor  (N) | Responden  (R) | N x R |
| Aplikasi ini mudah di pahami | SS | 5 | 4 | 20 |
| S | 4 | 13 | 52 |
| CS | 3 | 0 | 0 |
| TS | 2 | 0 | 0 |
| STS | 1 | 0 | 0 |
| Jumlah | | | 17 | 72 |
| Persentase *Y* = 72 𝑥 100  85 | | | | 85% |

Tabel 8 Hasil Presentase Pertanyaan Kedua

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pernyataan | Keterangan | Skor  (N) | Responden  (R) | N x R |
| Aplikasi mudah digunakan | SS | 5 | 9 | 45 |
| S | 4 | 6 | 24 |
| CS | 3 | 2 | 6 |
| TS | 2 | 0 | 0 |
| STS | 1 | 0 | 0 |
| Jumlah | | | 17 | 75 |
| Persentase *Y* = 75 𝑥 100  85 | | | | 88% |

Tabel 9 Hasil Presentase Pertanyaan Ketiga

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pernyataan | Keterangan | Skor  (N) | Responden  (R) | N x R |
| Proses deteksi aplikasi ini dapat berjalan dengan baik | SS | 5 | 6 | 30 |
| S | 4 | 7 | 28 |
| CS | 3 | 4 | 12 |
| TS | 2 | 0 | 0 |
| STS | 1 | 0 | 0 |
| Jumlah | | | 17 | 70 |
| Persentase *Y* = 70 𝑥 100  85 | | | | 82% |

Tabel 10 Hasil Presentase Pertanyaan Keempat

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pernyataan | Keterangan | Skor  (N) | Responden  (R) | N x R |
| Proses deteksi aplikasi ini terdapat kesalahan saat identifikasi  gambar | SS | 1 | 0 | 0 |
| S | 2 | 0 | 0 |
| CS | 3 | 1 | 3 |
| TS | 4 | 8 | 32 |
| STS | 5 | 8 | 40 |
| Jumlah | | | 17 | 75 |
| Persentase *Y* = 75 𝑥 100  85 | | | | 88% |

Tabel 11 Hasil Presentase Pertanyaan Kelima

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pernyataan | Keterangan | Skor  (N) | Responden  (R) | N x R |
| Aplikasi ini memiliki fungsi yang diharapkan | SS | 5 | 9 | 45 |
| S | 4 | 8 | 32 |
| CS | 3 | 0 | 0 |
| TS | 2 | 0 | 0 |
| STS | 1 | 0 | 0 |
| Jumlah | | | 17 | 77 |
| Persentase *Y* = 75 𝑥 100  85 | | | | 90% |

Tabel 12 Hasil Presentase Pertanyaan Keenam

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pernyataan | Keterangan | Skor  (N) | Responden  (R) | N x R |
| Aplikasi ini memiliki tampilan aplikasi yang mudah dipahami | SS | 5 | 5 | 25 |
| S | 4 | 9 | 36 |
| CS | 3 | 3 | 9 |
| TS | 2 | 0 | 0 |
| STS | 1 | 0 | 0 |
| Jumlah | | | 17 | 70 |
| Persentase *Y* = 70 𝑥 100  85 | | | | 82% |

Tabel 13 Hasil Presentase Pertanyaan ketujuh

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pernyataan | Keterangan | Skor  (N) | Responden  (R) | N x R |
| Aplikasi deteksi penyakit daun tanaman tomat ini sudah ada sebelumnya di  kota sorong | SS | 1 | 0 | 0 |
| S | 2 | 0 | 0 |
| CS | 3 | 3 | 9 |
| TS | 4 | 7 | 28 |
| STS | 5 | 7 | 35 |
| Jumlah | | | 17 | 72 |
| Persentase *Y* = 72 𝑥 100  85 | | | | 85% |

Berdasarkan hasil perhitungan diatas maka peneliti menghitung nilai rata-rata sebagai berikut :

88+85+82+88+90+ 82+85 = 86%

7

Berdasarkan hasil perhitungan rata-rata dengan menggunakan rumus *usability testing* pada 7 pernyataan, dapat ditarik kesimpulan bahwa pembuatan aplikasi deteksi jenis penyakit daun tanaman tomat berbasis *Android* sudah sesuai, dengan presentase yang didapat berdasarkan kriteria interpretasi skor mencapai dari 86% dengan keterangan sangat setuju.

# BAB IV

# PENUTUP

## 4.1 Kesimpulan

Berdasarkan implementasi dan pengujian yang telah dilakukan pada deteksi penyakit tanaman daun tomat menggunakan *Convolutional Neural Network (CNN)* berbasis *android*, maka dapat diperoleh beberapa kesimpulan sebagai berikut :

* + 1. Dari hasil implementasi sistem berbasis android dengan menggunakan metode *convolutional neural network (CNN)* mencapai skor *usability testing* sebesar 86% dapat ditarik kesimpulan bahwa penelitian ini berhasil dalam merancang sistem deteksi penyakit tanaman semangka.
    2. Dari hasil pengujian sistem menggunakan metode *blackbox testing* maka diperoleh kesimpulan bahwa sistem telah berjalan sebagaimna mestinya.
    3. Hasil implementasi *deep learning* dalam deteksi penyakit tanaman daun tomat yang dilakukan dalam penelitian ini menggunakan *convolutional neural network (CNN)* dengan arsitektur *VGG16* dapat mendeteksi penyakit tanaman daun tomat dengan tingkat akurasi sebesar 93.61%, untuk *precission* dan 93.6% *Confusion Matrix* F1-*score..* Hasil yang didapat menunjukkan bahwa proses deteksi penyakit tanaman semangka berhasil.

## 4.2 Saran

Penelitian yang telah dilakukan masih terdapat kekurangan dan masih dapat dikaji lebih lanjut. Saran yang dapat diberikan untuk peningkatan sistem deteksi penyakit tanaman daun tomat menggunakan CNN berbasis android. Meskipun sistem mencapai skor usability sebesar 86%, pengujian lebih lanjut dengan kelompok pengguna yang lebih beragam dapat memberikan wawasan tambahan untuk perbaikan antarmuka dan pengalaman pengguna. Selain itu, optimasi performa algoritma CNN dan arsitektur VGG16 diperlukan untuk meningkatkan efisiensi komputasi pada perangkat mobile. Penambahan fitur diagnostik yang memberikan rekomendasi perawatan tanaman berdasarkan jenis penyakit juga akan memberikan nilai tambah bagi pengguna. Uji lapangan yang lebih luas dengan berbagai kondisi lingkungan dan jenis tomat yang berbeda penting untuk memastikan keandalan sistem dalam berbagai situasi nyata. Pembaruan dan perluasan dataset gambar daun tomat yang digunakan untuk melatih model secara berkala sangat dianjurkan, serta integrasi dengan teknologi IoT untuk pemantauan tanaman secara real-time dan otomatis akan meningkatkan efektivitas dan efisiensi sistem.

# DAFTAR PUSTAKA

Agarwal, M., Singh, A., Arjaria, S., Sinha, A., & Gupta, S. (2020). ToLeD: Tomato Leaf Disease Detection using Convolution Neural Network. *Procedia Computer Science*, *167*(2019), 293–301. https://doi.org/10.1016/j.procs.2020.03.225

Ahmad, I., Hamid, M., Yousaf, S., Shah, S. T., & Ahmad, M. O. (2020). Optimizing pretrained convolutional neural networks for tomato leaf disease detection. *Complexity*, *2020*. https://doi.org/10.1155/2020/8812019

ALTUNTAŞ, Y., & KOCAMAZ, F. (2021). Deep Feature Extraction for Detection of Tomato Plant Diseases and Pests based on Leaf Images. *Celal Bayar Üniversitesi Fen Bilimleri Dergisi*, *17*(2), 145–157. https://doi.org/10.18466/cbayarfbe.812375

Awalia, N., & Primajaya, A. (2022). Identifikasi Penyakit Leaf Mold Daun Tomat Menggunakan Model DenseNet-121. *Jurnal Ilmiah Ilmu Komputer*, *8*(1), 49–54. http://ejournal.fikom-unasman.ac.id

Azzahro, N. F., Suprapto, Y., & Sudrajat. (2023). Rancangan E-Logbook Berbasis Android Menggunakan Indikator Peringatan Peralatan Navigasi Penerbangan Di Perum Lppnpi Cabang Semarang. *PROSIDING, Seminar Nasional Inovasi Teknologi Penerbangan (SNITP)*, *7*(7), 1–13.

Badrul, M., & Ardy, R. (2021). Penerapan Metode Waterfall pada Perancangan Sistem Informasi Pendaftaran Siswa Baru. *Jurnal Sains Komputer & Informatika (J-SAKTI*, *5*(1), 52–61.

Balakrishna K., & Rao, M. (2022). Tomato Plant Leaves Disease Classification Using KNN and PNN. *International Journal of Computer Vision and Image Processing*, *9*(1), 51–63. https://doi.org/10.4018/ijcvip.2019010104

Fathoni, F. M. (2024). Klasifikasi Penyakit Daun Tomat Menggunakan Algoritma K-NN Berdasarkan Ekstraksi Fitur GLCM dan LBP. *Jurnal Teknik Informatika Dan Teknologi Informasi*, *4*(1), 39–50. https://doi.org/10.55606/jutiti.v4i1.3417

Fathur Rozi, M. I., Adiwijaya, N. O., & Swasono, D. I. (2023). Identifikasi Kinerja Arsitektur Transfer Learning Vgg16, Resnet-50, Dan Inception-V3 Dalam Pengklasifikasian Citra Penyakit Daun Tomat. *Jurnal Riset Rekayasa Elektro*, *5*(2), 145. https://doi.org/10.30595/jrre.v5i2.18050

Fuad Mahrus Fathoni. (2024). Klasifikasi Penyakit Daun Tomat Menggunakan Algoritma K-NN Berdasarkan Ekstraksi Fitur GLCM dan LBP. *Jurnal Teknik Informatika Dan Teknologi Informasi*, *4*(1), 39–50. https://doi.org/10.55606/jutiti.v4i1.3417

Gelar Guntara, R. (2023). Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, *5*(1), 55–60. https://doi.org/10.47233/jteksis.v5i1.750

Hasan, M., Tanawala, B., & Patel, K. J. (2021). Deep Learning Precision Farming: Tomato Leaf Disease Detection by Transfer Learning. *SSRN Electronic Journal*, 1–5. https://doi.org/10.2139/ssrn.3349597

Kotta, C. R., Paseru, D., & Sumampouw, M. (2022). Implementasi Metode Convolutional Neural Network untuk Mendeteksi Penyakit Pada Citra Daun Tomat. *Jurnal Pekommas*, *7*(2), 123–132. https://doi.org/10.56873/jpkm.v7i2.4961

Kurniawan, H., Apriliah, W., Kurnia, I., & Firmansyah, D. (2021). Penerapan Metode Waterfall Dalam Perancangan Sistem Informasi Penggajian Pada Smk Bina Karya Karawang. *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi Dan Komunikasi*, *14*(4), 13–23. https://doi.org/10.35969/interkom.v14i4.78

Maeda-Gutiérrez, V., Galván-Tejada, C. E., Zanella-Calzada, L. A., Celaya-Padilla, J. M., Galván-Tejada, J. I., Gamboa-Rosales, H., Luna-García, H., Magallanes-Quintanar, R., Guerrero Méndez, C. A., & Olvera-Olvera, C. A. (2020). Comparison of convolutional neural network architectures for classification of tomato plant diseases. *Applied Sciences (Switzerland)*, *10*(4). https://doi.org/10.3390/app10041245

Muchtar, K., Chairuman, Yudha Nurdin, & Afdhal Afdhal. (2021). Pendeteksian Septoria pada Tanaman Tomat dengan Metode Deep Learning berbasis Raspberry Pi. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *5*(1), 107–113. https://doi.org/10.29207/resti.v5i1.2831

Mungki Astiningrum, Arhandi, P. P., & Ariditya, N. A. (2020). Identifikasi Penyakit Pada Daun Tomat Berdasarkan Fitur Warna Dan Tekstur. *Jurnal Informatika Polinema*, *6*(2), 47–50. https://doi.org/10.33795/jip.v6i2.320

Muslih, M., & Krismawan, A. D. (2024). Klasifikasi Penyakit Daun Tomat Berbasis Algoritma K-Nearest Neighbor. *Semnas Ristek (Seminar Nasional Riset Dan Inovasi Teknologi)*, *8*(01), 103–107. https://doi.org/10.30998/semnasristek.v8i01.7141

Natbais, Y. H., & Umbu, A. B. S. (2023). Aplikasi Deteksi Penyakit pada Daun Tomat Berbasis Android Menggunakan Model Terlatih Tensorflow Lite. *Teknotan*, *17*(2), 83. https://doi.org/10.24198/jt.vol17n2.1

Novalia, E., & Voutama, A. (2022). Black Box Testing dengan Teknik Equivalence Partitions Pada Aplikasi Android M-Magazine Mading Sekolah. *Syntax : Jurnal Informatika*, *11*(01), 23–35. https://doi.org/10.35706/syji.v11i01.6413

Putri, A. W. (2021). Implementasi Artificial Neural Network (ANN) Backpropagation Untuk Klasifikasi Jenis Penyakit Pada Daun Tanaman Tomat. *MATHunesa: Jurnal Ilmiah Matematika*, *9*(2), 344–350. https://doi.org/10.26740/mathunesa.v9n2.p344-350

Rahmadhani, A., Fauziah, F., & Aningsih, A. (2020). Sistem Pakar Deteksi Dini Kesehatan Mental Menggunakan Metode Dempster-Shafer. *Sisfotenika*, *10*(1), 37. https://doi.org/10.30700/jst.v10i1.747

Ridwan, A., Faisol, A., & Santi Wahyuni, F. (2020). Penerapan Metode Least Square Untuk Prediksi Penjualan Berbasis Web Pada Doni Sport Malang. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *4*(1), 129–136. https://doi.org/10.36040/jati.v4i1.2745

Sharma, P., Berwal, Y. P. S., & Ghai, W. (2020). Performance analysis of deep learning CNN models for disease detection in plants using image segmentation. *Information Processing in Agriculture*, *7*(4), 566–574. https://doi.org/10.1016/j.inpa.2019.11.001

Sigitta, R. C., Saputra, R. H., & Fathulloh, F. (2023). Deteksi Penyakit Tomat melalui Citra Daun menggunakan Metode Convolutional Neural Network. *Avitec*, *5*(1), 43. https://doi.org/10.28989/avitec.v5i1.1404

Soekarta, R., Nurdjan, N., & Syah, A. (2023). Klasifikasi Penyakit Tanaman Tomat Menggunakan Metode Convolutional Neural Network (CNN). *Insect (Informatics and Security): Jurnal Teknik Informatika*, *8*(2), 143–151. https://doi.org/10.33506/insect.v8i2.2356

Tan, L., Lu, J., & Jiang, H. (2021). Tomato Leaf Diseases Classification Based on Leaf Images: A Comparison between Classical Machine Learning and Deep Learning Methods. *AgriEngineering*, *3*(3), 542–558. https://doi.org/10.3390/agriengineering3030035

Wahid Islahfari, M., Lawi, A., & Siddik, A. M. A. (2022). Perbandingan Kinerja Model Ensembled Transfer Learning Pada Klasifikasi Penyakit Daun Tomat. *Seminar Nasional Teknik Elektro Dan Informatika (SNTEI)*, *8*(1), 286–291. http://118.98.121.208/index.php/sntei/article/view/3630

Wijaya, A., & Apridiansyah, Y. (2020). Penerapan Algortima Fisher Yates Shuflle Pada Media Pembelajaran Mapel Agama Islam Berbasis Android. *Jurnal Informatika Upgris*, *6*(1). https://doi.org/10.26877/jiu.v6i1.5747

Yağ, İ., & Altan, A. (2022). Artificial Intelligence-Based Robust Hybrid Algorithm Design and Implementation for Real-Time Detection of Plant Diseases in Agricultural Environments. *Biology*, *11*(12). https://doi.org/10.3390/biology11121732

Zhang, Y., Song, C., & Zhang, D. (2020). Deep Learning-Based Object Detection Improvement for Tomato Disease. *IEEE Access*, *8*, 56607–56614. https://doi.org/10.1109/ACCESS.2020.2982456

Lampiran 1 Evaluasi Pengerjaan Tugas Besar

Kelompok 19:

|  |  |
| --- | --- |
| Astita Ayu | : BAB 1, BAB 2, BAB 3, BAB 4, Jurnal Internasional, Jurnal Nasional, Develop Aplikasi. |
| 2. Nini Wara | : BAB 3, Desain Interface, Desain Cover, Develop Aplikasi. |

Mengetahui Dosen Mata Kuliah

Mata Kuliah Algoritma Pemrograman 2

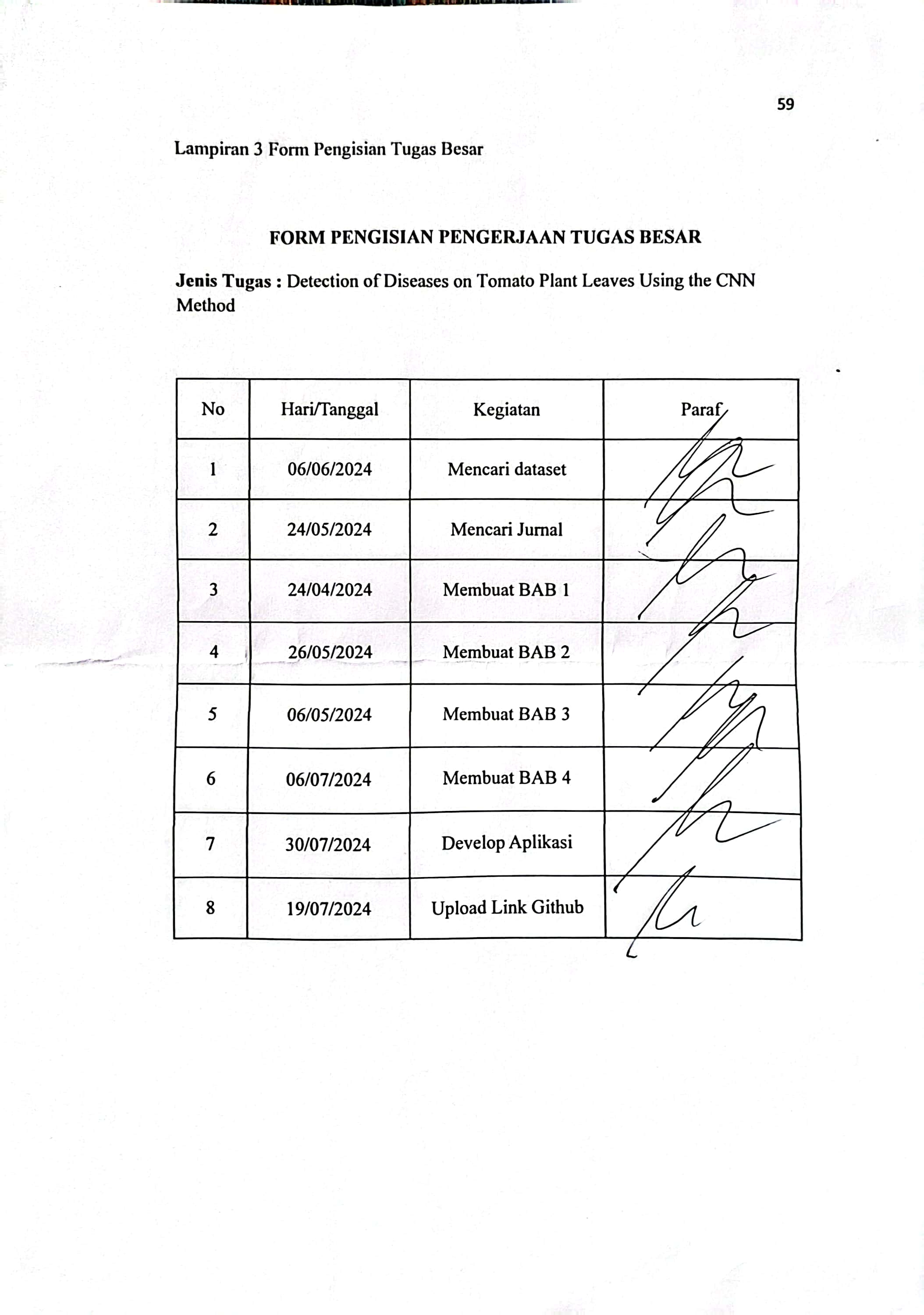
FAJAR R. B PUTRA, S.Kom., M.Kom.

NIDN. 1428099501

Lampiran 2 Dokumentasi







Lampiran 4 link Github

<https://github.com/astitaayu/Tugas_algoritma>