****

**PENERAPAN ALGORITMA CONVULUTIONAL NEURAL NETWORK DAN ARSITEKTUR NASNETMOBILE PADA APLIKASI DETEKSI PENYAKIT DAUN TOMAT**

**SKRIPSI**

**M HASANUDIN**

**19552011129**

**PROGRAM SARJANA DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS INDUSTRI KREATIF UNIVERSITAS TEKNOLOGI BANDUNG**

**APRIL 2024**

****

**PENERAPAN ALGORITMA CONVULUTIONAL NEURAL NETWORK DAN ARSITEKTUR NASNETMOBILE PADA APLIKASI DETEKSI PENYAKIT DAUN TOMAT**

**SKRIPSI**

Diajukan sebagai salah satu syarat untuk mencapai gelar Sarjana

**M HASANUDIN**

**19552011129**

**PROGRAM SARJANA DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS INDUSTRI KREATIF UNIVERSITAS TEKNOLOGI BANDUNG**

**APRIL 2024**

**Lembar Pengesahan**

**SEMINAR TERBUKA SKRIPSI**

**PENERAPAN ALGORITMA CONVULUTIONAL NEURAL NETWORK DAN ARSITEKTUR NASNETMOBILE PADA APLIKASI DETEKSI PENYAKIT DAUN TOMAT**

Diajukan sebagai salah satu syarat untuk mencapai gelar Sarjana

M Hasanudin

19552011129

|  |  |
| --- | --- |
| Pembimbing I  ( ) | Pembimbing II  ( ) |
| Kepala Departemen Teknik Informatika  ( ) | |

**PROGRAM SARJANA DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS INDUSTRI KREATIF UNIVERSITAS TEKNOLOGI BANDUNG**

**APRIL 2024**

**HALAMAN PERNYATAAN**

Dengan ini saya, M HASANUDIN, menyatakan bahwa Skripsi dengan judul “Penerapan Algoritma Convulutional Neural Network Dan Arsitektur Nasnetmobile Pada Aplikasi Deteksi Penyakit Daun Tomat” adalah benar-benar karya saya sendiri. Saya tidak melakukan penjiplakan kecuali melalui pengutipan sesuai dengan etika keilmuan yang berlaku. Saya bersedia menanggung risiko/sanksi yang dijatuhkan kepada saya apabila ditemukan pelanggaran terhadap etika keilmuan dalam Skripsi saya ini.

Bandung, 01 April 2024

Yang membuat pernyataan,

Materai 10.000

M HASANUDIN

19552011129

**Lembar Pengesahan**

**YUDISIUM**

**PENERAPAN ALGORITMA CONVULUTIONAL NEURAL NETWORK DAN ARSITEKTUR NASNETMOBILE PADA APLIKASI DETEKSI PENYAKIT DAUN TOMAT**

**SKRIPSI**

Diajukan sebagai salah satu syarat untuk mencapai gelar Sarjana

M Hasanudin

19552011129

|  |  |
| --- | --- |
| Pembimbing I  ( ) | Pembimbing II  ( ) |
| Kepala Departemen Teknik Informatika  ( ) | |

**PROGRAM SARJANA DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS INDUSTRI KREATIF UNIVERSITAS TEKNOLOGI BANDUNG**

**APRIL 2024**

**TANDA PENGESAHAN PENGUJI SKRIPSI**

|  |  |  |
| --- | --- | --- |
| NIM | : | 19552011129 |
| Nama | : | M Hasanudin |
| Jenjang Studi | : | Strata Satu (S1) |
| Departemen | : | Teknik Informatika |
| Fakultas | : | Fakultas Industri Kreatif |
| Judul Skripsi | : | Penerapan Algoritma Convulutional Neural Network Dan Arsitektur NasNetMobile Pada Aplikasi Deteksi Penyakit Daun Tomat |

Bandung,..........

Dipertahankan di depan Tim Penguji Skripsi Departemen Teknik Informatika Fakultas Industri Kreatif Universitas Teknologi Bandung

Pada Tanggal …….

Dewan Penguji

|  |  |  |
| --- | --- | --- |
| **No.** | **NAMA** | **TANDA TANGAN** |
| 1. | Nama Pembimbing I |  |
| 2. | Nama Penguji I |  |
| 3. | Nama Penguji II |  |

Mengetahui,

Kepala Departemen Teknik Informatika

Ttd dan Cap

(Nama Kepala Departemen)

**HALAMAN PERNYATAAN PERSETUJUAN AKADEMIS**

Sebagai civitas akademik Universitas Teknologi Bandung, saya yang bertanda tangan di bawah ini:

Nama : ………......................................................................................................

NIM : ……………………………......................................................................

Departemen : ..................................................................................................................

Fakultas : ………………………………………………………………………….

Jenis karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Teknologi Bandung **Hak Bebas Royalti Non Eksklusif (*Non-exclusive Royalty- Free Right*)** atas karya ilmiah saya yang berjudul :

………………………........................................................................................................

............................................................................................................................................

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non Eksklusif ini Universitas Teknologi Bandung berhak menyimpan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta. Demikian pernyataan ini saya buat dengan sebenarnya.

Tempat, Tanggal Bulan Tahun

# KATA PENGANTAR

Alhamdulillah, segala puji syukur penulis panjatkan kehadirat Allah SWT karena atas limpahan rahmat dan hidayah-Nya penulis dapat menyelesaikan Skripsi dengan judul “Penerapan Algoritma Convulutional Neural Network Dan Arsitektur NasNetMobile Pada Aplikasi Deteksi Penyakit Daun Tomat”. Tujuan penulisan Skripsi ini yaitu untuk memenuhi salah satu syarat kelulusan jenjang pendidikan S-1 Departemen Teknik Industri, Fakultas Industri Kreatif Universitas Teknologi Bandung.

Dalam penelitian ini, penulis banyak mendapatkan bimbingan, kritik, saran, dan motivasi yang sangat besar dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Muchammad Naseer, S.Kom., M.T. Rektor Universitas Teknologi Bandung.
2. Rina Indrayani, S.E., M.M. Wakil Rektor Bidang Pembelajaran dan Kemahasiswaan Universitas Teknologi Bandung.
3. Rika Andriyanti Dinata, S.T., M.T. selaku Dekan Fakultas Industri Kreatif Universitas Teknologi Bandung.
4. Yasti Aisyah Primianjani, S.Kom. selaku Kepala Departemen Teknik Informatika Universitas Teknologi Bandung.
5. Danny Aidil Rismayadi, S.SI., M.Kom. selaku Pembimbing 1 yang telah membimbing penulis untuk menyelesaikan penelitian skripsi.
6. Pak Ade Rukmana. selaku ketua koordinator Desa Tani Dompet Dhuafa
7. Pak Ujang Priatna. selaku pembimbing di Desa Tani Dompet Dhuafa
8. Kedua orang tua penulis yaitu Bapak Asep Soleh dan Ibu Enah Sukaenah yang selalu memberikan dukungan serta selalu berdo’a untuk kelancaran penulisan skripsi penulis
9. Teman yang selalu memberikan semangat dan berbagi ilmu diantaranya Muhammad Ikhwan Fatullah, Erryck Norrys serta teman kerja di PUSDATIN Universitas Teknologi Bandung yang memberikan dukungan dan bantuan kepada penulis.

Penulis menyadari bahwa penulisan Skripsi ini masih jauh dari kesempurnaan. Dengan segenap kerendahan hati, penulis berharap semoga segala kekurangan yang ada pada Skripsi ini dapat dijadikan bahan pembelajaran untuk penelitian yang lebih baik dimasa yang akan datang.

Bandung, 01 April 2024

M HASANUDIN

19552011129

# DAFTAR ISI

[KATA PENGANTAR iii](#_Toc145446481)

[DAFTAR ISI iv](#_Toc145446482)

[DAFTAR TABEL vi](#_Toc145446483)

[DAFTAR GAMBAR vii](#_Toc145446484)

[BAB I PENDAHULUAN 1](#_Toc145446485)

[1.1. Latar Belakang 1](#_Toc145446486)

[1.2. Rumusan Masalah Penelitian 3](#_Toc145446487)

[1.3. Tujuan Penelitian 3](#_Toc145446488)

[1.4. Manfaat Penelitian 4](#_Toc145446489)

[1.5. Ruang Lingkup 4](#_Toc145446490)

[1.6. Penelitian Sistematika Penulisan 5](#_Toc145446491)

[BAB II TINJAUAN PUSTAKA 7](#_Toc145446492)

[2.1. Landasan Teori 7](#_Toc145446493)

[2.1.1. Tanaman Tomat 7](#_Toc145446494)

[2.1.2. *Deep Learning* 7](#_Toc145446495)

[2.1.3. Metode *Convolutional Neural Network CNN* 8](#_Toc145446496)

[2.1.4. Arsitektur *NASNetMobile* 10](#_Toc145446497)

[2.1.5. Diagram Unified Modeling Language (UML) 10](#_Toc145446498)

[2.1.6. Model Pengembangan Perangkat Lunak (SDLC) 16](#_Toc145446499)

[2.1.7. *Extreme Programing* 17](#_Toc145446500)

[2.1.8. Bahasa Pemograman Python 18](#_Toc145446501)

[2.1.9. Analisis Kebutuhan Sistem 19](#_Toc145446502)

[2.1.10. *Flutter* 19](#_Toc145446503)

[2.1.11. Bahasa Pemograman *Dart* 20](#_Toc145446504)

[2.1.12. Android 20](#_Toc145446505)

[2.1.13. Keras 20](#_Toc145446506)

[2.1.14. Metode *In Depth Interview* 20](#_Toc145446507)

[2.1.15. Black Box Testing 20](#_Toc145446508)

[2.1.16. Pengujian Alpha 20](#_Toc145446509)

[2.1.17. Pengujian Beta 20](#_Toc145446510)

[2.1.18. Matlab 20](#_Toc145446511)

[2.1.19. Technology Acceptance Model (TAM) 20](#_Toc145446512)

[2.1.20. Skala Likert 20](#_Toc145446513)

[2.2. Penelitian Terdahulu 21](#_Toc145446514)

[2.2.1. State Of The Art 35](#_Toc145446515)

[BAB III ANALISIS DAN PERANCANGAN 36](#_Toc145446516)

[3.1. Metode Penelitian 36](#_Toc145446517)

[3.2. Metode Pengumpulan Data 36](#_Toc145446518)

[3.3. Metode Pengembangan Perangkat Lunak 37](#_Toc145446519)

[3.3.1. Evaluasi *Prototype* **Error! Bookmark not defined.**](#_Toc145446520)

[3.3.2. Mengkodekan Sistem **Error! Bookmark not defined.**](#_Toc145446521)

[3.3.3. Pengujian Sistem **Error! Bookmark not defined.**](#_Toc145446522)

[3.3.4. Evaluasi Sistem **Error! Bookmark not defined.**](#_Toc145446523)

[3.3.5. Menggunakan Sistem **Error! Bookmark not defined.**](#_Toc145446524)

[BAB IV IMPLEMENTASI DAN PENGUJIASN SISTEM 79](#_Toc145446525)

[DAFTAR PUSTAKA 80](#_Toc145446526)

[LAMPIRAN 82](#_Toc145446527)

# DAFTAR TABEL

Tabel 2.1 Simbol-simbol Use Case Diagram 11

Tabel 2.2 Simbol-simbol Class Diagram 13

Tabel 2.3 Simbol-simbol Activity Diagram 14

Tabel 2.4 Simbol-simbol Sequence Diagram 16

Tabel 2.5 Penelitian Terdahulu 21

Tabel 3.1 Komponen 37

Tabel 3.2 Kebutuhan Fungsional 41

Tabel 3.3 Kebutuhan Non Fungsional 42

Tabel 3.4 Use Case Scenario Kelola Dasboard 44

Tabel 3.5 Use Case Scenario Kelola Data cuaca 45

Tabel 3.6 Use Case Scenario Prediksi Data cuaca 46

Tabel 3.7 Use Case Scenario Menampilkan saran dosis pupuk 48

Tabel 3.8 Use Case Scenario Mengelola akses token alat 49

Tabel 3.9 Use Case Scenario Data visual grafik 50

Tabel 3.10 Tabel visualisasi\_data 61

Tabel 3.11 Tabel data\_cuaca 61

Tabel 3.12 Tabel sensor 61

Tabel 3.13 Tabel sensor\_pemantauan 61

Tabel 3.14 Tabel sensor\_kendali 62

Tabel 3.15 Tabel akses\_token 62

Tabel 3.16 Tabel prediksi\_cuaca 63

Tabel 3.17 Tabel data\_dosis\_pupuk 63

Tabel 3.18 Rencana Pengujian Dasboard 65

Tabel 3.19 Rencana Pengujian Akses Token 66

Tabel 3.20 Rencana Pengujian Saran Dosis Pupuk 66

Tabel 3.21 Rencana Pengujian Database Cuaca 67

Tabel 3.22 Rencana Pengujian Prediksi Cuaca 67

Tabel 3.23 Rencana Pengujian Beta 68

Tabel 4.1 Jadwal Kerja 71

# DAFTAR GAMBAR

Gambar 1.1 Konsumsi Cabai 2017-2021 1

Gambar 1.2 Bagan pertumbuhan penggunaan Internet of Things 2

Gambar 3.1 Proses Model Prototype 35

Gambar 3.2 Arsitektur jaringan Greenhouse 36

Gambar 3.3 Desain arsitektur jaringan sistem 38

Gambar 3.4 Alur proses yang sedang berjalan 39

Gambar 3.5 Alur proses yang diusulkan 40

Gambar 3.6 Use Case Diagram 44

Gambar 3.8 Activity diagram Kelola Dasboard 51

Gambar 3.9 Activity diagram Kelola data cuaca 52

Gambar 3.10 Activity diagram Prediksi cuaca 52

Gambar 3.11 Activity diagram Saran dosis pupuk 53

Gambar 3.12 Sequence Diagram Dasboard 54

Gambar 3.13 Sequence Diagram Saran dosis pupuk 55

Gambar 3.14 Sequence Diagram kelola data cuaca 55

Gambar 3.15 Sequence Diagram prediksi cuaca 56

Gambar 3.16 Class Diagram 57

Gambar 3.17 Mockup Dasboard 57

Gambar 3.18 Mockup Akses Token 58

Gambar 3.19 Mockup Saran Dosis Pupuk 58

Gambar 3.20 Mockup Data Cuaca 59

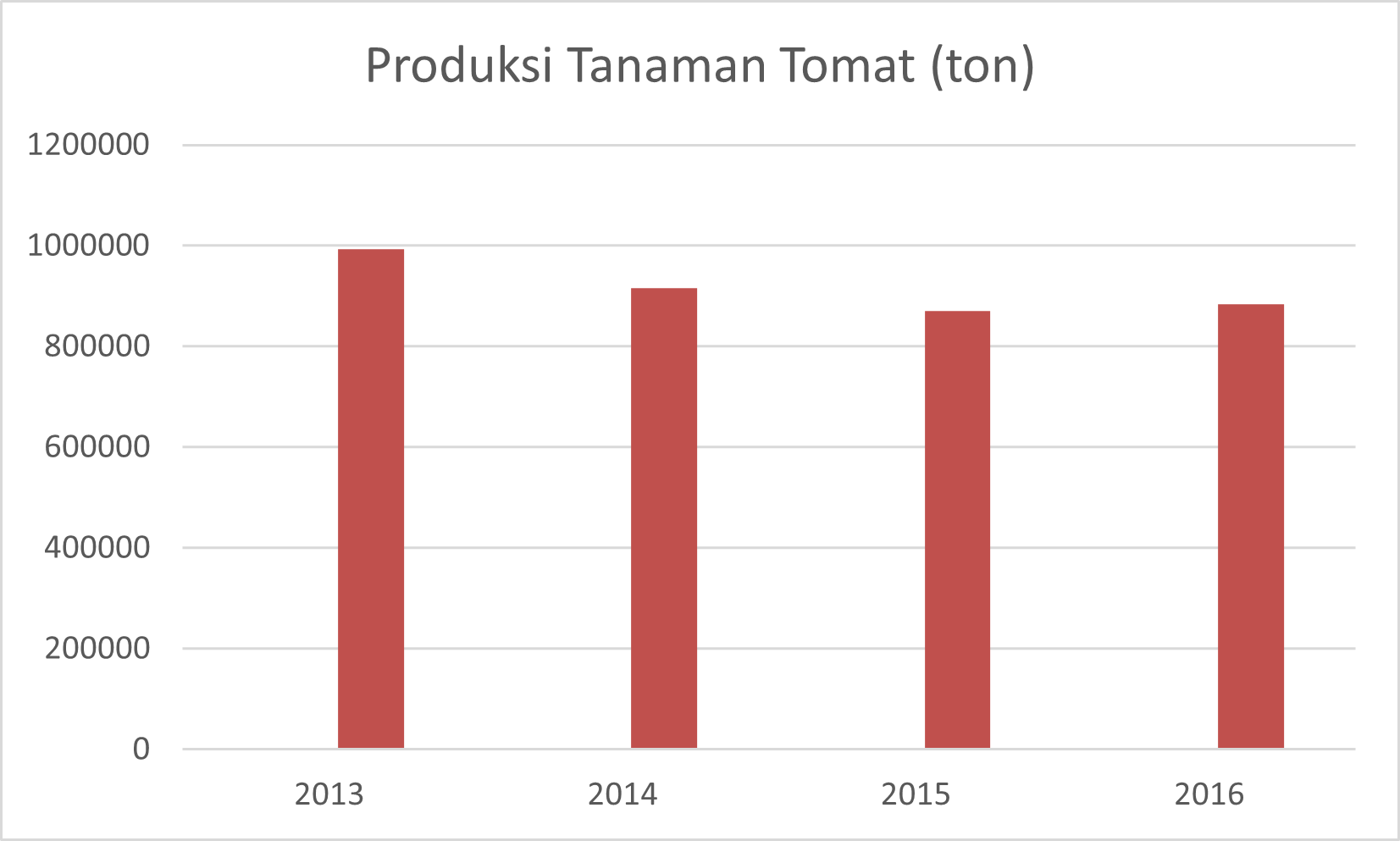
Gambar 3.21 Mockup Grafik Data Cuaca 59

Gambar 3.22 Mockup Prediksi Cuaca 60

# BAB I PENDAHULUAN

Latar Belakang

Negara agraris merupakan negara yang sebagian penduduknya bekerja di sektor pertanian, salah satu negara agraris yang cukup besar adalah negara Indonesia. Jenis komoditas sayuran hortikultura yang dibudidayakan di negara Indonesia salah satunya adalah tanaman tomat. Buah tomat memiliki nilai ekonomis cukup tinggi dan sangat potensial untuk dikembangkan dari segi produksinya, karena nilai konsumsi buah ini yang luas (Sharon Patricia Siahaan, dkk. 2020). Produksi tanaman tomat di Indonesia pada tahun 2016 mencapai 883.242 ton, angka tersebut sudah mengalami kenaikan kurang lebih 13 ton dari tahun sebelumnya (Badan Pusat Statistik, produksi tanaman sayuran. 2018). Bagan produksi tomat pada tahun 2013-2016 dapat dilihat pada gambar 1.1 di bawah ini:



Gambar 1.1 Produksi tomat 2013-2016

(sumber *website* Badan Pusat Statistik)

Meskipun mengalami kenaikan, hal tersebut masih rendah dibandingkan dengan jumlah konsumsi tomat. Pada tahun 2016 data nasional menunjukkan bahwa konsumsi tomat mencapai 1.152.790 ton (Kahlil Muchtar, dkk. 2021). Dapat disimpulkan bahwa terdapat perbedaan yang cukup tinggi antara produksi tomat dengan permintaan atau konsumsi tomat.

Metode yang digunakan pada penelitian ini adalah *Convolutional Neural Network* (CNN) dengan arsitektur *NasNetMobile*. Metode CNN merupakan salah satu metode *machine learning* yang berada pada cabang klasifikasi, kelebihan dari metode ini yaitu secara otomatis mengekstrasi input tanpa kehilangan informasi ciri penting dari tiap citra gambar, sehingga dapat menambah tingkat akurasi algoritma CNN (Nadhifa Sofia. 2018). NasNetMobile ialah arsitektur jaringan saraf tiruan yang menggunakan metode CNN, tingkat akurasi arsitektur ini dapat dikatakan cukup baik dengan ukuran penggunaan memori yang kecil untuk diterapkan pada aplikasi mobile serta kinerja arsitektur ini cukup cepat (Anwar F dan Aries S. 2022).

Tanaman tomat merupakan salah satu jenis sayuran hortikultura yang rentan terhadap serangan hama dan penyakit. Penyakit pada tanaman tomat bisa disebabkan oleh virus, jamur, bakteri dan hama. Pada umumnya penyakit dan hama yang menyerang tanaman tomat bisa ditemukan pada daun tanaman, batang dan buah tomat, serta dapat dikenali secara visual. Setiap gejala penyakit memiliki ciri warna dan tekstur yang khas serta berbeda. Penyakit pada tanaman tomat bisa di deteksi dengan pendekatan biologi, pendekatan ini dikenal sebagai teknik molekuler. Penelitian yang dilakukan Resti Fajarfika, dkk tentang pendeteksian penyakit kuning pada tanaman tomat, menyatakan pengujian penelitian ini melalui tahap RT-PCR(*reverse transcription-polymerase chain reactions*), hasilnya menunjukkan identifikasi virus yang menyebabkan penyakit kuning pada daun tomat satu kelompok dengan jenis virus yang ada di negara Jepang, Amerika Utara Dan Eroupa (Resti Fajarfika, dkk. 2015). Pendeteksian melalui pendekatan biologi dilakukan pada laboratorium dengan membutuhkan biaya lebih dan waktu yang tidak sedikit (Akbar Hidayatuloh, Thesis. 2018).

Desa Tani merupakan salah satu program koperasi yang dijalankan oleh produsen Argonatik Pratama Indonesia yang bekerja sama dengan berbagai pihak salah satunya dengan Dompet Dhuafa. Salah satu hasil produksi pertaniannya adalah tanaman tomat. Kondisi naik dan turunnya produksi tomat ini juga dirasakan oleh koperasi Desa Tani ini. Salah satu faktor penghambat yang berpengaruh cukup besar terhadap peningkatan produksi tomat adalah penyakit dan hama.

Berdasarkan hasil wawancara dengan salah satu petani dan juga sebagai tim petugas koordinator di Desa Tani, Pak Jajang. Menurut beliau ada banyak kendala pada budidaya tanaman tomat. Salah satu yang utama adalah terserang berbagai penyakit dan hama. Hal tersebut bisa diakibatkan oleh petani dan petugas telat memberikan nutrisi yang dibutuhkan oleh tanaman tomat. Selain itu juga para petani kesulitan dalam mendiagnosis penyakit yang menyerang tanaman, masing-masing dari petani memiliki perkiraannya masing-masing terhadap menentukan jenis penyakit, hal tersebut menjadikan penanganan yang di lakukan tidak efektif. Pendiagnosaan penyakit tomat yang dilakukan petani di Desa tani yaitu dengan melihat langsung dari gejala penyakitnya, hal ini di sampaikan oleh salah satu petani tomat, Pak Mamat.

Oleh karena itu salah satu cara untuk meningkatkan produksi tomat yaitu membantu petani dalam mengidentifikasi penyakit tomat dengan biaya rendah, maka diperlukan bantuan teknologi. Teknologi yang akan dibangun berupa aplikasi yang dapat mengidentifikasi penyakit tanaman tomat melalui citra gambar daun tomat, serta memberikan solusi berupa penanganan dan jumlah dosis obat berdasarkan penyakit tomat yang telah teridentifikasi. Pengimplementasian metode CNN pada penelitian ini untuk proses identifikasi atau diagnosa penyakit tomat berdasarkan citra daun tomat. Dataset yang digunakan berupa citra daun tomat dengan 5 kelas penyakit. Jenis penyakit yang diambil yaitu Bercak kering, Embun tepung, dan Tengorok Daun*.* Berdasarkan penuturan permasalahan yang sudah dijabarkan, penulis mengambil judul penelitian **“*PENERAPAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK DAN ARSITEKTUR NASNETMOBILE PADA APLIKASI DETEKSI PENYAKIT DAUN TOMAT*”.**

Rumusan Masalah Penelitian

Berdasarkan latar belakang di atas dan dari hasil wawancara yang dilakukan, maka tersusun rumusan masalah sebagai berikut:

1. Terdapat beberapa petani yang mengalami kesulitan dalam mendiagnosis jenis penyakit tomat yang dilakukan petani dengan melihatnya tanpa alat bantu, sehingga kurang optimalnya proses pengenalan penyakit tanaman tomat oleh petani.
2. Setiap petani memiliki pendapatnya masing-masing terkait jenis penyakit yang dilihat mereka, sehingga terdapat solusi yang kurang efektif dalam penanganannya.

Tujuan Penelitian

Tujuan dari penelitian ini sebagai berikut:

1. Merancang dan membangun aplikasi deteksi penyakit tanaman tomat melalui citra daun berbasis android.
2. Menerapkan Algoritma *Convolutional Neural Network (CNN)* dalam klasifikasi jenis penyakit tanaman tomat dan hasil diagnosa disertai juga dengan solusi penanganan.

Manfaat Penelitian

Manfaat penelitian ini adalah sebagai berikut:

1. Bagi Petani Desa Tani Dompet Dhuafa

Manfaat yang bisa dirasakan oleh petani di Desa Tani Dompet Dhuafa yaitu dapat membantu petani dalam mendeteksi penyakit tanaman tomat berdasarkan hasil klasifikasi yang dihasilkan oleh aplikasi yang dibangun.

1. Bagi Peneliti

Diharapkan peneliti mendapatkan pengetahuan mengenai identifikasi penyakit tomat berdasarkan data citra daunnya tomat. Selanjutnya penelitian bisa mengetahui proses penerapan metode *deep learning* dengan model *convolutional neural network* dan arsitektur *NasNetMobile* untuk klasifikasi penyakit pada tanaman tomat.

1. Bagi Sekolah Tinggi Teknologi Bandung

Mengimplementasikan hasil pembelajaran secara langsung terhadap objek studi dan mengembangkan materi pembelajaran yang telah didapatkan di kelas. Selain itu dapat mewujudkan Tri Dharma perguruan tinggi yaitu penelitian dan pengembangan, serta pengabdian kepada masyarakat.

Ruang Lingkup

Ruang lingkup pada penelitian ini ditetapkan agar penelitian tidak terlalu luas, ruang lingkup ini bertujuan untuk memudahkan dan penelitian bisa terarah dalam penjelasannya.

1. *Deployment* algoritma CNN pada aplikasi *mobile* dengan memanfaatkan *TFlite.*
2. Penelitian ini berfokus untuk mendeteksi satu jenis tanaman yaitu tanaman tomat.
3. Dataset yang digunakan untuk proses *training* dan *testing* berasal dari data gambar daun tomat yang terkena penyakit.
4. Pengambilan data gambar langsung dari lapangan di Desa Tani Dompet Dhuafa, selanjutnya untuk dataset pendukung didapatkan pada website kaggel*.*
5. Penelitian ini berfokus pada 4 kelas yang akan digunakan untuk klasifikasi penyakit tanaman tomat, yaitu bercak kering, embun tepung, tengorok daun dan daun sehat
6. Jenis berkas dalam gambar citra yaitu JPG (*Join Photographic Group)*
7. Metode yang diambil yaitu *Convolutional Neural Network CNN* dengan arsitektur *NASNetMobile*

Penelitian Sistematika Penulisan

Sistematika penulisan yang memuat uraian secara garis besar isi proposal skripsi untuk tiap-tiap Bab.

BAB I PENDAHULUAN

Pada bab ini berisi latar belakang penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, ruang lingkup penelitian dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab dua menjelaskan tentang tinjauan pustaka dan landasan teori yang berhubungan dengan topik yang akan dibahas dalam penelitian ini.

BAB III ANALISA DAN PERANCANGAN

Bagian ini menjelaskan tentang metode penelitian yang digunakan dalam menyelesaikan penelitian ini. Pada bab ini juga akan menjelaskan tentang metodologi pengembangan perangkat lunak (SDLC). Adapun yang penulis uraikan sebagai pelengkap pada bab ini adalah analisis kebutuhan sistem, tahapan desain menggunakan visualisasi *Unified Modelling Language* (UML) yang terdiri dari *Use Case Diagram, Use Case Scenario, Activity Diagram* dan *Sequence Diagram*, hingga ke rancangan desain tampilan sistem.

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Bagian ini berisikan tentang hasil dan pengujian dalam pembangunan aplikasi

BAB V KESIMPULAN DAN SARAN

Bagian ini berisikan tentang kesimpulan dan saran dari hasil pembangunan aplikasi serta pengujian aplikasi.

# BAB II TINJAUAN PUSTAKA

* 1. Landasan Teori

Ringkasan teori dari buku atau jurnal, penjelasan tentang konsep dasar yang berkaitan dengan penelitian ini. Landasan teori harus sudah jelas antara teori yang dipakai karena teori ini berfungsi untuk memperjelas masalah yang diteliti.

### Tanaman Tomat

Tomat tergolong dalam tanaman sayuran yaitu family Solanaceae. Tomat merupakan tanaman Hortikultura yang sangat dikenal masyarakat dan mempunyai banyak manfaat, baik sebagai sayuran, bahan baku industri obat-obatan dan kosmetik, maupun sebagai bahan baku pengolahan makanan. Di samping itu, buah tomat mengandung gizi yang tinggi, yaitu vitamin A, vitamin C, protein, karbohidrat, kalsium, natrium, fosfor, vitamin, riboflavin, niasin, dan askorbik (Cahyono, 2000).

Tanaman tomat banyak ditanam pada dataran tinggi dan dataran rendah. Tanaman tomat termasuk tanaman semusim yang berumur sekitar 3-4 bulan (Surtinah, 2007), serta dapat ditanam sepanjang tahun. Namun waktu yang paling baik untuk menanam tomat adalah musim kemarau yang dibantu dengan penyiraman secukupnya (Pracaya, 1994)

Buah tomat merupakan salah satu bahan pangan yang memiliki kandungan antioksidan yang cukup tinggi(Mu’nisa A, 2012). Tomat memiliki senyawa polifenol, karotenoid, asam askorbat, potasium, vitamin A, dan vitamin C yang dapat bertindak sebagai antioksidan. Polifenol pada tomat sebagian besar terdiri dari flavonoid, sedangkan jenis karotenoid yang dominan adalah pigmen likopen (Eveline dkk, 2014).

### *Deep Learning*

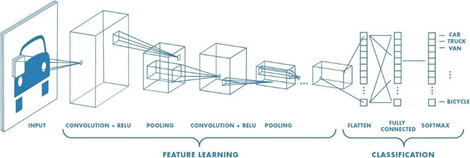
*Deep learning*  adalah metode *representation learning* dengan tingkat representasi yang majemuk, yang diperoleh dengan menyusun modul yang sederhana tetapi non linier yang masing-masing mentransformasikan dari level sederhana ke tingkat yang lebih tinggi, yaitu tingkat yang lebih abstrak (M Ibnu Choldun R dan Kridanto Surendro. 2018).

Salah satu pengembangan dari *machine learning* adalah *deep learning*. Pengembangan ini terdiri dari banyak layer tersembunyi (*hidden layer)* yang merupakan bagian dari kecerdasan buatan. Contoh penerapan *deep learning* adalah pemrosesan gambar digital (Brian Dwi Hartomo. 2021). *Deep learning* merupakan aspek jaringan tiruan(*artificial* *neural network*) memiliki tujuan untuk meniru pembelajaran *machince learning* yang banyak digunakan manusia untuk mencari jenis pengetahuan tertentu.

### Metode *Convolutional Neural Network CNN*

*Convolution Neural Network* (CNN) pertama kali dikembangkan dengan nama Neocognitron oleh Kunihiko Fukushima, seorang peneliti dari NHK *Broadcasting Science Research Laboratories*, Kinuta, Setagaya, Tokyo, Jepang. (K. Fukushima, "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position," Biological Cybernetics, 1980). *Convolutional Neural Network* (CNN) adalah salah satu algoritma *deep learning* yang digunakan untuk kasus-kasus penggunaan *computer vision* seperti mengklasifikasikan gambar atau video dan mendeteksi objek di dalam gambar atau bahkan wilayah dalam gambar (Moolayil, 2019). Cara kerja dari CNN dengan memanfaatkan proses konvolusi dengan cara mengubah-ubah posisi sebuah *kernel* konvolusi (filter) berukuran tertentu pada sebuah gambar. Selanjutnya komputer akan mendapatkan informasi representatif baru dari hasil perkalian pada bagian gambar dengan menggunakan filter tadi (L Qolbiyatul, 2019).

Menurut Keiron O’Shea dan Ryan Nash dalam penelitiannya menyatakan bahwa metode CNN berfokus pada klasifikasi dengan masukan berupa gambar, oleh karena itu metode CNN memiliki arsitektur yang diatur sedemikian rupa untuk menangani tipe data berupa data citra (Keiron O’Shea & Ryan Nash, 2015). Arsitektur CNN dibagi menjadi 2 bagian besar yaitu *Feature Extraction* Layer dan *Fully Connected* Layer (MLP), arsitektur CNN dapat dilihat pada gambar 2.1 di bawah ini:



Gambar 2.1 Arsitektur CNN

(Sumber artikel Qolbiyatul Lina, pada *website* Medium.com)

1. *Feature Extraction Layer*

Proses dari arsitektur ini disebut dengan “*encoding*”. Proses *encoding* akan dilakukan dari sebuah gambar dan di rubah menjadi *feature* yang berupa angka-angka. Angka-angka tersebut merepresentasikan sebuah gambar (*Feature Extraction*). Arsitektur *Feature Extraction* Layer terdiri dari dua bagian yaitu *Convolutional Layer* dan *Pooling Layer*. Tapi terkadang ada beberapa penelitian yang menggunakan *Convolutional layer* saja.

1. *Convolutional Layer* (*Conv. Layer*)

*Conv. Layer* terdiri dari banyak neuron yang tersusun membentuk sebuah filter yang memiliki panjang dan tinggi (*pixel*) serta terdapat jumlah *channel* warna RGB pada gambar tersebut. Sebagai contoh, terdapat masukan berupa gambar. Selanjutnya masuk pada layer pertama pada *feature extraction*, di mana *conv. Layer* dengan ukuran 32x32x3. Jadi panjang 32 *pixel*, lebar 32 *pixel* dan tebal/jumlah *channel* 3 buah.

Ketika filter tersebut akan bergeser pada seluruh bagian gambar. Proses pergeseran akan dilakukan operasi “dot” antara *input* dan nilai dari filter sehingga menghasilkan nilai *output* atau biasa disebut dengan *feature* map.

1. *Pooling Layer*

Hal terpenting dari pembuatan metode CNN adalah dengan memiliki banyak lapisan *pooling* layer. *Polling* layer berada setelah proses *conv. layer*. Pada dasarnya *pooling* layer terdiri dari sebuah filter yang memiliki ukuran dan *stride* tertentu dan bergeser pada seluruh area *feature* map. Terdapat 2 jenis *Pooling* yang biasa digunakan yaitu *Max Pooling* dan *Average Pooling*. Tujuan *pooling* layer adalah mengurangi dimensi dari *feature* map (*downsampling*) dan mengatasi *overfitting*. Penggunaan *pooling* layer akan mempercepat penyelesaian algoritma CNN, karena parameter yang harus di *update* menjadi semakin sedikit.

1. *Stride*

Dalam pergeseran filter pada proses *conv. layer* jumlah pergeseran dapat ditentukan dengan parameter *stride*. Jadi ketika nilai *stride* adalah 1, maka proses *conv. layer* filter akan bergeser sebanyak 1 *pixel* secara *horizontal* lalu *vertical*. Jika ingin mendapatkan informasi yang lebih detail dari sebuah *input* maka nilai *stride* harus semakin kecil. Namun menggunakan nilai *stride* kecil tidak akan selalu mendapatkan performa yang bagus.

### Arsitektur *NASNetMobile*

*NasNetMobile* merupakan arsitektur yang dikembangkan oleh *Google Inc*. Pengembangan dari arsitektur ini lebih difokuskan pada *embedded system* dan perangkat seluler, NasNetmobile adalah salah satu varian arsitektur *NasNet* dengan parameter lebih sedikit sehingga memiliki kinerja yang cepat dan cocok diimplementasikan untuk menyelesaikan masalah klasifikasi gambar pada perangkat *mobile* (F Anwar & S Aries, 2022).

### Diagram Unified Modeling Language (UML)

*Unified Modeling Language* atau bisa disebut Diagram UML merupakan pemodelan dan komunikasi dari sistem yang dibuat. Seperti namanya Diagram UML, pemodelan yang disajikan menggunakan diagram. Diagram-diagram ini digunakan sebagai representasi visual objek, kondisi proses pada sebuah sistem. Tujuan dari Diagram UML ini agar pengguna dan pembuat sistem dapat memahami struktur dan perilaku sebuah sistem.

1. *Use Case Diagram*

*Use Case Diagram* merupakan pemodelan untuk mendeskripsikan sebuah interaksi dari satu atau lebih aktor dengan sistem yang dibuat. *Use case* digunakan untuk mengetahui fungsi yang terdapat pada sistem dan siapa saja yang dapat menggunakan fungsi tersebut. Berikut simbol-simbol yang ada pada *Use Case Diagram* (Heri Oktovan. 2019):

Tabel 2.1 Simbol-simbol *Use Case Diagram*

|  |  |  |
| --- | --- | --- |
| No. | Simbol | Deskripsi |
| 1. | Use case | Fungsi yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar aktor atau antar unit. Isi *use case* berupa kata kerja yang diawali nama *use case* |
| 2. | Aktor | Aktor biasanya digambarkan dengan orang, akan tetapi gambar tersebut tidak hanya menjelaskan aktor saja, bisa sistem lain atau juga proses tambahan yang berinteraksi dengan sistem |
| 3. | Asosiasi | Garis tanpa anak panah merupakan asosiasi, memiliki fungsi sebagai jalur komunikasi antara aktor dan *use case.* |
| 4. | Ekstensi/e*xtend*  <<extend>> | *Extend* merupakan *use case* tambahan yang dapat berdiri sendiri. Biasanya anak panah dari *use case* tambahan menuju *use case* utama. Sebagai contoh dapat dilihat dibawah ini: |
| 5. | *Include*  <<include>> |  |
| 6. | Generalisasi | Hubungan spesialisasi antar dua buah *use case* dengan *use case* utama. Sebagai contoh pada gambar berikut:    Pada contoh diatas *use case* utama berada di tengah (*use case* generalisasi/umum), anak panah menuju *use case* umum. |

*(sumber Heri Oktovan 2019:18-24. “Perancangan Sistem Informasi Geografis Pemetaan Pada Daerah Yang Memiliki Potensi Sumber Daya Alam Di Wilayah Provinsi Sumatera Barat Berbasis Web”)*

1. *Class Diagram*

*Class diagram* merupakan penggambaran struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat pada sistem. Pada kelas terdapat atribut yang merupakan variabel-variabel yang dimiliki kelas tersebut, selain itu pada kelas juga terdapat operasi yang merupakan fungsi-fungsi yang terdapat pada kelas.

Struktur kelas dapat bagi menjadi: Kelas main, merupakan kelas yang pertama dieksekusi ketika sistem dijalankan; Kelas *Controller*, merupakan kelas yang diambil dari pendefinisian *use case*; Kelas *view,* merupakan pendefinisian pengaturan tampilan sistem; dan Kelas model, merupakan pendefinisian dari data dan memiliki tujuan untuk membungkus data menjadi sebuah kesatuan baik untuk diambil atau di simpan ke basis data. Berikut merupakan simbol-simbol pada *Class Diagram:*

Tabel 2.2 Simbol-simbol *Class Diagram*

|  |  |  |
| --- | --- | --- |
| No. | Simbol | deskripsi |
| 1. | Kelas | Kelas yang terdapat pada struktur sistem |
| 2. | Antarmuka    nama\_interface | Konsep *interface* sama dengan konsep pada pemrograman berorientasi objek |
| 3. | Asosiasi | Asosiasi dengan garis tanpa anak panah menggambarkan relasi umum dan biasanya disertai dengan *multiplicity* |
| 4. | Asosiasi berarah | Asosiasi dengan anak panah menggambarkan relasi yang bermakna antar kelas, biasanya di sertai *multiplicity* |
| 5. | Generalisasi | Relasi antar kelas memiliki makna umum ke khusus |
| 6. | Kebergantungan | Relasi antar kelas yang memiliki ketergantungan |
| 7. | Agregrasi | Relasi antar kelas memiliki makna semua-bagian. |

*(sumber Heri Oktovan 2019:18-24. “Perancangan Sistem Informasi Geografis Pemetaan Pada Daerah Yang Memiliki Potensi Sumber Daya Alam Di Wilayah Provinsi Sumatera Barat Berbasis Web”)*

1. *Activity Diagram*

Menurut Rosa dan Shalahuddin (2016:161) dalam penelitian Heri Oktovan bahwa activity diagram merupakan aliran kerja (*workflow)* dari sebuah sistem atau menu yang ada pada sistem. Simbol actity diagram dapat dilihat pada tabel 2.3 dibawah ini:

Tabel 2.3 Simbol-simbol *Activity Diagram*

|  |  |  |
| --- | --- | --- |
| No. | Simbol | deskripsi |
| 1. | Status awal | Pada diagram aktivitas memiliki status awal, seperti gambar disamping |
| 2. | Aktivitas | Simbol aktivias disamping biasanya dilengkapi dengan kata kerja |
| 3. | Percabangan/*decision* | Jika ada pilihan aktivitas lebih dari satu dapat digambarkan dengan simbol disamping |
| 4. | Penggabungan/*join* | Ketika terdapat lebih dari satu aktivitas digabungkan maka menggunkana simbol *join* |
| 5. | *Swimlane* | Berfungsi untuk memisahkan organisasi aktivitas sistem yang bertanggung jawab terhadap aktivitas tersebut. |
| 6. | Status akhir | Terdapat status awal, maka akan di akhiri dengan status akhir. |

*(sumber Heri Oktovan 2019:18-24. “Perancangan Sistem Informasi Geografis Pemetaan Pada Daerah Yang Memiliki Potensi Sumber Daya Alam Di Wilayah Provinsi Sumatera Barat Berbasis Web”)*

1. *Sequence Diagram*

Menurut Rosa dan Shalahuddin (2016:161) dalam penelitian Heri Oktovan bahwa, *Sequence Diagram* berfungsi untuk menggambarkan waktu hidup objek dan message yang dikirim atau diterima antar objek. Simbol dari *Sequence Diagram* dapat dilihat pada tabel 2.4 dibawah ini:

Tabel 2.4 Simbol-simbol Sequence Diagram

|  |  |  |
| --- | --- | --- |
| No. | Simbol | deskripsi |
| 1. | Aktor    Atau    Tanpa waktu aktif | Menggambarkan orang, proses atau sistem lain yang berinteraksi dengan sistem yang dibuat. Simbol aktor ini tidak selalu menggambarkan orang. Biasanya diawali dengan kata benda |
| 2. | *Lifeline* | Menyatakan objek yang sedang berinteraksi dengan pesan |
| 3. | Pesan tipe *create* | Menyatakan sebuah objek membuat objek lain, arah anak panah menuju pada objek yang dibuat |
| 4. | Pesan tipe *send* | Menyatakan bahwa objek mengirim data atau masukan kepada objek lain, arah anak panah mengarah pada objek yang dituju |
| 5. | Waktu aktif | Menyatakan keaktifan objek dan saling berinteraksi dengan terhubung pada waktu aktif ini, misal    Maka cekStatusLogin() dan open() berada didalam metode login(). |
| 6. | Pesan tipe *call* | Menyatakan objek yang memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. misal pemanggilan operasi/metode pada diri sendiri:    Anak panah mengarah pada objek yang memiliki operasi/metode. Metode/operasi ini hendaknya harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi. |
| 7. | Pesan tipe *return* | Menyatakan objek yang menjalankan suatu operasi/metode menghasilkan suatu kembalian |
| 8. | Pesan tipe *destroy* | Menyatakan suatu objek mengakhiri objek lain, arah panah menuju pada objek yang akan diakhiri. Biasanya ketika ada *create* maka ada *destroy* |

*(sumber Heri Oktovan 2019:18-24. “Perancangan Sistem Informasi Geografis Pemetaan Pada Daerah Yang Memiliki Potensi Sumber Daya Alam Di Wilayah Provinsi Sumatera Barat Berbasis Web”)*

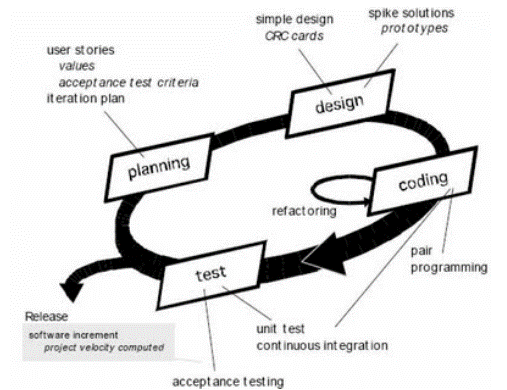
### Model Pengembangan Perangkat Lunak (SDLC)

Software Development Life Cycle (SDLC) merupakan proses pengembangan perangkat lunak yan dirancang untuk berjalan secara sistematis sehingga dapat menghasilkan perangkat lunak yang berkualitas. SDLC juga berfungsi untuk membagi peran dan tanggung secara jelas antara designer, business analyst dan project manager. Fungsi lain dari SDLC juga dapat memberikan gambaran jelas tentang input dan otput dari satu tahap ke tahap berikutnya. Secara umum ada enam tahapan dalam SDLC yaitu perencanaan sistem, analisis sistem, perancangan sistem, impementasi sistem, pengujian sistem, dan pemeliharaan sistem (Deni Murdiani dan Muhamad Sobirin, 2022).

Metode SDLC model waterfall merupakan metode tahap pengembangan yang mempunyai ciri 5 tahapan yaitu tahap planning, analisis, desain, pengkodean program, maintenance. Sehingga masing-masing fase tersebut harus harus dikerjakan secara berurutan dan tidak bisa diacak (P. T. Damai dkk, 2013).

### *Extreme Programing*

*Extreme Programming* (XP) adalah metode pengembangan perangkat lunak yang sederhana dan mencakup salah satu metode tangkas yang dipelopori oleh Kent Beck, Ron Jeffries, dan Ward Cunningham. *Extreme Programming* (XP) merupakan sebuah proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek dan sasaran dari metode ini adalah tim yang dibentuk dalam skala kecil sampai medium serta metode ini juga sesuai jika tim dihadapkan dengan kebutuhan yang tidak jelas maupun terjadi perubahan–perubahan kebutuhan yang sangat cepat (S A Prabowo dkk, 2013). Gambar .... menunjukkan proses pengembangan perangkat lunak dari metode XP.



Gambar .... Proses *Extreme Programming*

(Sumber Jurnal Teknik Pomits, Vol. 2 (3), 2013)

Gambar ... di atas menunjukkan 4 proses dari metode *Extreme Programming* yang bisa disebut sebagai siklus besar (Pressman, 2010). Berikut merupakan penjelasan dari proses metode *Extreme Programming:*

1. *Planning*/Perencanaan, pada tahap ini pengembang mulai melakukan pengumpulan kebutuhan yang membantu memahami konteks bisnis dari sebuah aplikasi. Tahapan pertama ini juga akan mendefinisikan hasil berupa fitur yang akan dimiliki aplikasi dan fungsi dari aplikasi.
2. *Design*/Perancangan, tahapan kedua menekankan dari segi desain aplikasi yang mudah dipahami. Dalam tahapan desain aplikasi dapat menggunakan konsep *Class Responsibility Collaborator* (CRC) cards untuk tujuan mengidentifikasi dan mengatur *class* pada *object-oriented.*
3. *Coding*/Pengkodean, pada tahap ini dilakukan implementasi dari tahapan desain ke kode program. Konsep utama dari tahapan pengkodean adalah pair programming, yang artinya terdapat 2 orang dalam menyusun kode yang saling bekerja sama dalam penyusunan kode dari desain yang ada serta menyusun kode dengan mengikuti standar pengkodean.
4. *Testing*/Pengujian, pada tahapan sebelum liris aplikasi, pengembang akan fokus pada pengujian fitur dan fungsionalitas lainnya dari aplikasi.

### Bahasa Pemrograman Python

Python merupakan salah satu bahasa pemrograman yang banyak digunakan oleh perusahaan besar maupun para developer untuk mengembangkan berbagai macam aplikasi berbasis dekstop, web dan mobile. Python diciptakan oleh Guido Van Rossum di Belanda pada tahun 1990 dan namanya diambil dari televisi kesukaan. Guido Monty Python’s Flying Circus Van Rossum mengembangkan Python sebagai hobi, kemudian Python menjadi bahasa pemrograman yang dipakai secara luas dalam industri dan pendidikan karena sederhana, ringkas, sintak sintuitif dan memiliki pustaka yang luas (Schuerer & Maufrais, 2010). Pada sistem operasi linux bahasa pemrograman ini menjadi standarisasi untuk disertakan dalam paket distribusinya (A Muhardian, 2017).

### Analisis Kebutuhan Sistem

Pada analisis kebutuhan sistem terdapat dua kategori yaitu kebutuhan fungsional dan kebutuhan non fungsional. Menurut Nih Luh Putri Srinadi dan Nih Nyoman Harini Puspita pada penelitiannya menyebutkan bahwa Kebutuhan Fungsional adalah kebutuhan dari segi layanan sistem yang disediakan, bagaimana sistem yang akan dibuat akan memberi aksi terhadap *input* dari pengguna dan seperti apa proses serta *output* yang diberikan (Nih Luh Putri Srinadi dan Nih Nyoman Harini Puspita. 2018).

Menurut Agus Setiawan dkk, menyatakan bahwa kebutuhan fungsional mempunyai tujuan untuk mendapatkan informasi terkait kebutuhan sistem berdasarkan proses-proses apa saja yang nantinya dilakukan. Sedangkan Analisis kebutuhan non-fungsional mempunyai perbedaan, yaitu untuk mendapatkan informasi terkait dengan kebutuhan-kebutuhan sistem yang meliputi properti perilaku-perilaku yang dimilikinya (Agus Setiawan dkk, 2016).

### *Flutter*

*Flutter* merupakan sebuah SDK (*Software* *Development* *Kit*) yang digunakan untuk mengembangkan aplikasi mobile yang dibuat oleh Google. *Flutter* dikembangkan untuk membuat aplikasi yang mempunyai *performance* yang tinggi dan dapat dipublikasikan untuk platform Android dan IOS dari *codebase* tunggal (Utomo Ardy dkk, 2018).

*Flutter* menggunakan bahasa pemrograman C, C++, *Dart*, dan *Skia* (2D *Rendering* *Engine*). Komponen Utama *Flutter* (Cato Chandra dkk, 2019):

1. *Flutter Engine*, memberikan dukungan *rendering* tingkat rendah dengan menggunakan *library graph Skia* milik Google dan dapat berinteraksi dengan *tools* pengembangan perangkat lunak yang disediakan Android dan iOS.
2. *Foundation Library,* menyediakan fungsi dan *class* dasar yang digunakan dalam membangun aplikasi dengan *Flutter* dan juga API untuk berkomunikasi dengan *Engine.*

### Bahasa Pemograman *Dart*

Bahasa pemrograman *Dart* merupakan bahasa pemrograman *general-purpose,* berbasis *class* dan berorientasi terhadap objek dengan menggunakan sintaks bahasa pemrograman Cyang dibuat oleh Google dan dirancang oleh Lars Bak dan Kasper Lund. Bahasa pemrograman ini dapat digunakan untuk mengembangkan berbagai macam platform termasuk di dalamnya adalah web, aplikasi mobile, server, dan perangkat yang mengusung teknologi *Internet of Things. Dart* juga dapat digunakan untuk mengembangkan aplikasi dari *codebase* tunggal menjadi aplikasi Android maupun iOS. Bahasa pemrograman ini dirilis secara *open source* oleh Google dengan lisensi BSD dan juga bertipe dinamis (W Suprantono, 2019).

### Android

Android adalah sistem operasi berbasis Linux yang di peruntukan bagi telepon seluler, dengan menyediakan platform open source bagi para pengembang android dapat menciptakan aplikasi yang digunakan bermacam peranti bergerak seperti *smartphone* dan tablet. Dalam pembuatan aplikasi berbasis android bahasa pemrograman yang dipakai yaitu bahasa C, C++ dan Java sebagai desain tampilannya. Pengelolaan paket yang disediakan oleh Play Store sebagai penyedia aplikasi resmi dari sistem operasi Android (Usamah Ash-Shidiq, dkk. 2017).

### Keras

Keras merupakan *library* yang fokus mempermudah proses pengembangan pada jaringan syaraf tiruan tingkat tinggi, dibuat menggunakan bahasa pemrograman *python* dan mampu beroperasi di atas *TensorFlow.* Ekosistem Keras mencakup setiap langkah alur kerja *Machine Learning* mulai dari pengolahan data, pelatihan *hyperparameter* hingga solusi penerapan(Aditya S dan Gunawan A. 2018).

Keras juga dapat disebut sebagai API yang dirancang untuk manusia, karena proses kerjanya mengikuti praktik terbaik untuk mengurangi kesulitan manusia. Keras meminimalkan jumlah tindakan dalam penggunaan umum, memberikan pesan kesalahan yang jelas, dapat ditindak lanjuti dengan mudah dan ekosistem API yang konsisten dan sederhana (Keras.io).

### Flask

*Flask* merupakan *framework* ringan dibandingkan dengan *framework* lainnya yang dibuat berbasis *python*. Pelopor yang mengembangkan *framework* ini adalah Armin Ronacher dengan ide awalnya untuk menyederhanakan inti *framework*-nya seminimal mungkin tapi tidak mempengaruhi performanya, hal ini juga menjadi *tagline* dari *framework* ini “*Web development, one drop at a time*” (Robby Reza dkk, 2016). Aplikasi yang dibuat menggunakan Flask akan disimpan dalam satu berkas *.py.* Framework Flask memang sederhana tetapi dapat diperluas dengan beragam *library* tambahan sesuai kebutuhan pengguna (Endang T dan Eko B S, 2017).

### Metode *In Depth Interview*

Wawancara mendalam (in-depth interview) merupakan bagian dari metode kualitatif. Maksud dari wawancara mendalam adalah proses memperoleh informasi untuk tujuan penelitian secara mendalam dengan cara tanya jawab sambil bertatap muka antara pewawancara dengan informan (orang yang di wawancarai), biasanya pada metode wawancara mendalam terdapat daftar pertanyaan yang telah dibuat sebelumnya oleh pewawancara berdasarkan tujuan dan maksud diadakannya wawancara tersebut (Noor Wahyuni, 2014).

### Black Box Testing

Pengujian *software* sangat diperlukan agar sesuai dengan yang diharapkan oleh pengembang dan juga pengguna. Black Box Testing merupakan salah satu metode yang dapat dipakai untuk menguji kelayakan dari sebuah sistem. Pengujian pada metode ini hanya memeriksa nilai keluaran berdasarkan masukan yang dilakukan pengguna pada sistem. Proses pengujian dengan cara mencoba menjalankan *software* yang telah dibuat pada setiap *form-*nya (Fadhila Cahya Ningrum, dkk. 2019).

Black Box Testing merupakan pengujian perangkat lunak yang menitik beratkan hanya mengamati hasil eksekusi sistem melalui data uji dan memeriksa fungsional dari sistem aplikasi tersebut, dengan kata lain Black Box Testing ini menitik beratkan pengujiannya pada fungsi sistem. Secara umum pengujian ini bertujuan untuk mengetahui bahwa setiap bagian dalam sistem aplikasi telah benar menampilkan output/pesan-pesan kesalahan dalam masukan data (Umi Hanifah, dkk. 2016).

### Pengujian Alpha

Pengujian alpha adalah proses pengujian dengan tujuan memastikan dari segi fungsionalitasnya dapat berjalan dengan lancar tanpa ada gangguan berupa error atau bug. Pada umumnya pengujian alpha dapat di lakukan dengan teknik Black Box Testing dengan hasilnya berupa kolom hasil pengujian dan kesimpulannya, dimana hasil tersebut dapat ditulis pada tabel (S Masripah dan L Ramayanti. 2020).

### Pengujian Beta

Pengujian beta merupakan jenis pengujian yang bersifat langsung di lapangan. Proses pengujian beta dilakukan dengan cara menyebarkan kuesioner berdasarkan pengalaman penggunaan sistem aplikasi, hasil dari kuesioner berupa angka yang akan dihitung untuk menentukan kesimpulan penilaian sejauh mana sistem aplikasi berjalan sesuai tujuan (S Masripah dan L Ramayanti. 2020).

### Matlab

Matlab merupakan perangkat lunak yang mampu melakukan proses komputasi matematik, menganalisis data, mengembangkan algoritma, melakukan simulasi dan pemodelan serta menghasilkan tampilan grafik dan antar-muka grafikal. Penggunaan matlab hanya dengan mengetikan perintah pada comand prompt di aplikasi matlab (R.H. Sianipar, 2013).

### *Technology Acceptance Model* (TAM)

*Technology Acceptance Model* (TAM) adalah suatu teori sistem informasi yang bertujuan untuk memodelkan bagaimana pengguna bisa, mulai dan menggunakan suatu teknologi. TAM sendiri diperkenalkan pertama kali oleh Davis pada tahun 1986, kemudian dikembangkan sebagai model penerimaan dari pengguna pada suatu sistem informasi di tahun 1989. TAM menyebutkan bahwa penggunaan sistem paling di pengaruhi oleh intensi perilaku (keinginan menggunakan kembali). Persepsi pengguna terhadap manfaat (perceived usefulness) dan persepsi pengguna terhadap kemudahan (perceived ease of use) merupakan bagian penting dari model TAM (Feronica M, dkk. 2011).

### Skala *Likert*

Skala *Likert* merupakan metode yang di kembangkan pada tahun 1932, pengembangnya adalah Likert. Skala *Likert* adalah metode untuk mengukur skala kepuasan pengguna. Metode ini umumnya digunakan pada kuesioner dengan pertanyaan-pertanyaan yang di kombinasikan sehingga membentuk nilai yang merepresentasikan pengetahuan, sikap dan perilaku. Berikut tabel yang menggambarkan Skala *Likert* dapat di lihat pada tabel ... (Ryan dan Walter. 2018).

Tabel .... Skala Likert

|  |  |
| --- | --- |
| Skala | Kategori |
| 1 | Sangat Setuju |
| 2 | Sejuta |
| 3 | Cukup |
| 4 | Kurang Setuju |
| 5 | Tidak Setuju |

* 1. Penelitian Terdahulu

Penyusunan skripsi ini mengambil referensi dari penelitian terdahulu dan dikembangkan, berikut beberapa penelitian terdahulu yang berkaitan dengan penelitian ini dapat dilihat pada tabel 2.5.

Tabel 2.5 Penelitian Terdahulu

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Judul | Peneliti, Media Publikasi dan Tahun | Tujuan Penelitian | Kesimpulan | Saran atau Kelemahan | Perbandingan |
| 1 | Pemantauan Dan Pengontrolan Sistem Pemupukan Budidaya Tanaman Tomat Menggunakan *Convolutional Neural Network* | Sharon Patricia Siahaan, Dr. Eng. Willy Anugrah Cahyadi dan Ir. Porman Pangaribuan. Jurnal *e-Proceeding of Engineering*. 2020 | Memberikan efisiensi dalam pemupukan agar menjaga kualitas produksi, penghematan pemberian pupuk dan kelestarian lingkungan | -Data citra yang digunakan pada penelitian ini sebanyak 400 buah.  -jumlah layer yang digunakan 28 layer dan jumlah *epoch* 300. Kedua jumlah ini mempengaruhi tingkat akurasi deteksi.  -nilai akurasi dari pengujian deteksi daun sebesar 87% | -jumlah dataset tiap kelas sebaiknya di perbanyak  -selain nilai layer dan epoch yang mempengaruhi nilai akurasi, jenis arsitektur dari CNN juga dapat mempengaruhi nilai keakurasian | Pada penelitian yang akan dibangun penulis memiliki jumlah *class* 4 buah dengan jumlah dataset tiap *class* 200 buah gambar.  Penelitian yang dilakukan penulis akan lebih fokus untuk mendeteksi penyakit tanaman tomat berdasarkan gambar daun tomat. |
| 2 | Implementasi CNN dan SVM untuk  Identifikasi Penyakit Tomat via Daun | Felix, Said Faisal, Theresia F M Butarbutar dan Pahala Sirait. Jurnal Sifo Mikroskill. 2019 | Menentukan *classifier* yang lebih baik dalam mendeteksi penyakit tanaman tomat antara metode CNN dan SVM | -Hasil pengujian pada penelitian ini sangat dipengaruhi oleh ekstraksi fitur tekstur (energi dan entropi)  -dataset yang digunakan berjumlah 200 citra daun tomat, dengan 160 citra digunakan sebagai data latih dan 40 citra sebagai data uji  -algoritma yang digunakan yaitu SVM (Support Vector Machine) dan CNN  -Hasil akurasi dari metode SVM 90.83% dan hasil akurasi dari metode CNN sebesar 97.5%  -Perbandingan hasil akurasi dari kedua algoritma didapati bahwa CNN merupakan *classifier* yang lebih baik dibandingkan dengan algorita SVM | -Jumlah dataset tiap kelas sebaiknya di perbanyak  -Hasil dari perhitungan dapat dibuat berbasis mobile | Pada penelitian yang dilakukan penulis memiliki jumlah *class* 4 buah dengan jumlah dataset tiap *class* 200 buah gambar.  Aplikasi akan dibuat berbasis mobile  Fokus proses prediksi menggunakan 1 algoritma yaitu CNN |
| 3 | Pendeteksian Septoria pada Tanaman Tomat dengan Metode Deep Learning berbasis Raspberry Pi | Kahlil Muchtar, Chairuman, Yudha Nurdin dan Afdhal. Jurnal RESTI. 2021 | Merancang alat yang dapat digunakan untuk mendeteksi penyakit bercak daun *septoria* berbasis *deep learning* dengan menggunakan *Raspberry Pi* sebagai *mikrokontroler* dan *Intel Movidius Neural Computing Stick (NCS)* yang berfungsi untuk mempercepat proses komputasi | -Penelitian ini dapat mendeteksi penyakit bercak daun *septoria* pada tanaman tomat berbasis *Raspberry pi* dan menggunakan model *InceptionV3* yang mendeteksi penyakit bercak daun *septoria* pada tanaman tomat dengan akurat.  -Hasil akurasi didapati antara 84.22% sampai 100% dengan rata-rata 95.85% diperoleh dari 30 sampel daun tomat | Klasifikasi deteksi penyakit dapat dikembangkan untuk beberapa penyakit tomat | Penelitian yang dilakukan penulis dapat melakukan proses klasifikasi deteksi penyakit tomat dengan 4 jenis penyakit.  Implementasi hasil klasifikasi dibuat berbasis mobile |
| 4 | Identifikasi Penyakit Tanaman Tomat Melalui Daun Dengan Metode Convolutional Neural Network Arsitektur Squeezenet | Akbar Hidayatuloh. Thesis UPI. 2018 | Penelitian ini bertujuan untuk mengimplementasikan metode CNN dengan arsitektur *squeezenet* dalam identifikasi penyakit tanaman tomat berbasis android | Dalam mendeteksi penyakit tanaman tomat melalui daun menggunakan metode CNN dengan arsitektur *squeezenet*  Aplikasi deteksi di bangun berbasis Mobile menggunakan Android Studio  Hasil akurasi arsitektur *squeezenet* sebesar 86.92%, hasil ini menunjukkan bahwa model sudah cukup baik untuk mendeteksi penyakit tanaman tomat melalui daun dari data gambar daun yang belum ada pada dataset. |  | Pada penelitian yang dilakukan penulis, akan melakukan observasi terhadap arsitektur dari metode CNN yang akan di gunakan. Lalu akan memilih arsitektur yang lebih baik. |
| 5 | Penerapan Algoritma *Random Forest* dengan Kombinasi Ekstraksi  Fitur Untuk Klasifikasi Penyakit Daun Tomat | Umi khultsum dan Agus subekti. Jurnal Media Informatika Budidarma. 2021 | Tujuan penelitian ini berdasarkan metode yang digunakan adalah untuk mengembangkan teknik dalam mengidentifikasi penyakit daun pada tanaman tomat dengan  meningkatkan akurasi klasifikasi. | -menerapkan 3 fitur dalam mendukung klasifikasi dari metode *Random Forest* yaitu fitur warna, fitur Hu-Moment, dan fitur haralick  -hasil akurasi dari kombinasi ekstraksi fitur dengan metode Random Forest, sebesar 96% |  | Fokus penelitian yang dilakukan penulis yaitu mengimplementasikan hasil prediksi dari metode CNN pada aplikasi mobile  Data citra daun di ambil langsung pada objek penelitian. |
| 6 | Identifikasi Penyakit Pada Tanaman Tomat Berdasarkan Warna Dan  Bentuk Daun Dengan Metode Naive Bayes  Classifier Berbasis Web | Fahri Alviansyah, Ikhwan Ruslianto dan Muhammad Dipenogoro. Jurnal Coding Sistem Komputer Untan. 2017 | Pada penelitian ini bertujuan untuk membuat sebuah aplikasi identifikasi penyakit pada  tanaman tomat dengan memanfaatkan pengolahan citra digital melalui proses ekstraksi fitur warna  HSV dan bentuk menggunakan metode deteksi tepi operator *sobel* dan metode *Naive Bayes Classifier* sebagai proses identifikasi | -Nilai rata-rata pixel warna HSV didapatkan dari konversi nilai RGB ke HSV. Sedangkan nilai rata-rata deteksi tepi sobel didapatkan dari konversi RGB ke grayscale, selanjutnya menghitung konvolusi menggunakan operator sobel.  -Data pengujian berjumlah 47 dengan hasil 39 pengujian dapat teridentifikasi dan 8 pengujian tidak teridentifikasi.  -Presentase dari hasil identifikasi sebesar 82,98% dan kegagalan sebesar 17,02%  -Fitur ekstraksi citra dipengaruhi oleh cahaya dan *noise.* Jika cahaya dan noise makin tinggi maka objek akan semakin tidak jelas. | Dataset gambar daun dapat diperbanyak dan juga perbanyak dari data latihnya  Aplikasi untuk implementasi hasil dari klasifikasi bisa dibuat berbasis mobile  Metode klasifikasi bisa menggunakan CNN | Metode yang digunakan penulis dalam penelitian menggunakan metode CNN  Pada penelitian yang dilakukan penulis memiliki jumlah *class* 5 buah dengan jumlah dataset tiap *class* 200 gambar daun. |
| 7 | Deteksi Penyakit pada Daun Kentang  Menggunakan Pengolahan Citra  dengan Metode *Convolutional Neural Network* | Abdul Jalil Rozaqi, dkk. *Citec Jounal.* 2021 | Tujuan dari penelitian ini adalah membuat sistem yang dapat mengidentifikasi penyakit pada tanaman kentang dengan memanfaatkan data gambar daun kentang. | Berdasarkan implementasi metode CNN pada data gambar daun kentang, dengan pembagian *dataset* 80% data *training* (922 gambar)dan 20% data *testing* (230 gambar), serta menggunakan ukuran gambar 150x150 *pixels .* Maka diperoleh nilai akurasi 95% dan akurasi validasi sebesar 94% | -Sebaiknya data daun sehat bisa ditambah, sehingga dataset dari ketiga kelas (daun sehat, *late blight* dan *ealy blight*) memiliki jumlah yang sama | -Penelitian yang dilakukan penulis lebih fokus pada identifikasi penyakit pada tanaman tomat  -Pembagian kelas memiliki 5 kelas, dengan masing-masing kelas memiliki data gambar daun tomat sebesar 200 gambar.  -Implementasi dari hasil klasifikasi akan di tampilkan pada aplikasi berbasis android. |
| 8 | Identifikasi Penyakit Pada Daun Tomat Berdasarkan Fitur Warna dan Tekstur | Mungki Astiningrum, dkk. Jurnal Informatika Polinema. 2020 | Tujuan dari penelitian ini untuk merancang sistem identifikasi penyakit pada daun tomat dengan menggunakan fitur ekstraksi warna *RGB* dan fitur ekstraksi tekstur *Gray Level Co-Occurance Matrix (GLCM).* Hasil dari identifikasi kedua fitur tersebut selanjutnya akan di klasifikasi menggunakan metode *K-Nearest Neighbor (KNN).* | Berdasarkan penelitian ini, bahwa implementasi sistem identifikasi penyakit tanaman tomat menggunakan fitur RGB dan GLCM serta proses klasifikasi menggunakan metode KNN dapat dilakukan secara baik, dengan nilai K yang di uji adalah k=1 dan nilai kurasi sebesar 92,89%.  Data gambar pada penelitian ini berjumlah 750 gambar dengan dibagi menjadi 3 kelas (Sehat, Bercak Bakteri dan Busuk Daun). Data training berjumlah 600 dan data testing berjumlah 150 citra daun. | -Data citra daun dapat di perbanyak, sehingga setiap kelas memiliki data citra dengan jumlah yang sama. | -Pembagian kelas memiliki 5 kelas, dengan masing-masing kelas memiliki data gambar daun tomat sebesar 200 gambar.  -Implementasi dari hasil klasifikasi akan di tampilkan pada aplikasi berbasis android |
| 9 | Klasifikasi Penyakit Tanaman Kedelai Melalui Tekstur Daun dengan Metode Gabor Filter | Rangga Pahlevi Putra, dkk. Jurnal EECCIS. 2018 | Penelitian ini memiliki tujuan untuk mengembangkan aplikasi pengolah citra digatal untuk identifikasi penyakit tanaman kedelai berdasarkan masukan berupa citra daun kedelai yang terkena penyakit. | Metode *Gabor filter* dapat digunakan untuk mengenali penyakit tanaman kedelai melalui citra daun.  Penelitian ini, untuk nilai frekuensi pada nilai pertengahan yaitu 0,176 menghasilkan keluaran *Gabor filter* dengan tingkat kecerahan cukup.  Sedangkan pada orientasi yang terlihat paling mudah dikenali adalah sudut 45 ̊dan 135 ̊. Sehingga dengan kemampuannya mengisolasi frekuensi dan orientasi tertentu dari citra, maka Gabor filter bisa digunakan sebagai salah satu metode dalam proses ekstraksi tekstur dalam mengenali kondisi daun yang terkena penyakit | Pada penelitian ini menggunakan dataset citra daun berjumlah 90 citra daun kedelai, dengan rincian 60 data latih dibagi menjadi tiga kelas dan 30 data testing dengan dibagi menjadi tigas kelas. Maka dari itu sebaiknya dataset yang digunakan dapat di perbanyak agar hasil dari metode Gabor filter jauh lebih baik. | -Penelitian yang dilakukan penulis lebih fokus pada identifikasi penyakit pada tanaman tomat  -Pembagian kelas memiliki 5 kelas, dengan masing-masing kelas memiliki data gambar daun tomat sebesar 200 gambar. Jadi total dataset semuanya adalah 1000 citra daun tanaman tomat.  -penelitian penulis menggunakan metode CNN untuk identifikasi penyakit tanaman tomat  -implementasi dari hasil klasifikasi akan di bangun pada sistem berbasis android. |
| 10 | Perancangan Dan Implementasi Untuk Penyortiran Buah Tomat (Lycopersicum Esculentum) Dengan Metode Learning Vector Quantizationprototype | Gayuh Erlanggono G, dkk. Jurnal *e-Proceeding of Engineering.* 2017 | Tujuan penelitian ini adalah membuat sistem untuk identifikasi tingkat kematangan buah tomat berdasarkan RGB pada citra buah tomat. Identifikasi dilakukan dengan menggunakan jaringan saraf tiruan dengan jenis algoritma *Learning Vector Quantization* | Tingkat akurasi dari algoritma *Learning Vector Quantization* untuk mendeteksi kadar tomat dari hasil threshold RGB sebesar 76,67% | Dapat dikembangkan dengan algoritma lain, seperti KNN dan CNN. | Penelitian penulis melakukan identifikasi penyakit pada tanaman tomat berdasarkan citra daun tomat  Implementasi dari hasil identifikasi berbasis android  Pengumpulan data citra gambar dilakukan dengan pengambilan gambar secara langsung |

### State Of The Art

Berdasarkan beberapa penelitian terdahulu yang telah dilakukan, peneliti melakukan perkembangan dari beberapa aspek. Hal ini dapat menjadikan pembeda antara penelitian yang dilakukan penulis dan penelitian terdahulu. Adapun beberapa perbedaannya antara lain sebagai berikut:

* + 1. Sistem yang dibangun berbasis android
    2. Dataset menggunakan gambar daun tomat perpaduan antara gambar yang diambil di lapangan dan di dalam ruangan
    3. Kelas yang ditentukan berjumlah 4 kelas, dengan jenis penyakit tomat tenggorok daun, bercak kering, embun tepung dan daun sehat
    4. *Deployment* menggunakan *Tensorflow lite*
    5. Metode klasifikasi menggunakan metode CNN dengan arsitektur *NASNetMobile.*

# BAB III ANALISIS DAN PERANCANGAN

* 1. Metode Penelitian

Penelitian yang dilakukan penulis dalam pengumpulan data menggunakan metode kuantitatif dengan berfokus pada elemen manusia dan objek yang di teliti, serta hubungan dari berbagai elemen yang terkait dengan tujuan untuk memahami peristiwa, perilaku, persepsi, fenomena atau tindakan. Sedangkan dalam metode pengembangan sistem penulis menggunakan model *Extreme Programming.*

* 1. Metode Pengumpulan Data

Pada penelitian hal yang penting dilakukan adalah proses pengumpulan data, dengan strategi/proses ini penelitian bisa di golongkan ke dalam penelitian ilmiah. Terdapat beberapa metode pengumpulan data yang dilakukan yaitu sebagai berikut:

1. Studi Literatur

Studi literatur dilakukan dengan pencarian terhadap berbagai sumber, bisa berupa buku-buku, arsip, artikel dan jurnal atau sumber-sumber lainnya yang relevan dengan penerapan deep learning menggunakan metode klasifikasi. Adapun studi literatur yang dilakukan yaitu melalui jurnal, buku dan *thesis*.

1. Observasi

Metode Observasi yang dilakukan dengan cara pengamatan secara langsung pada objek yang menjadi rujukan untuk diteliti, yaitu pada lahan pertanian Desa Tani Dompet Dhuafa, Suntenjaya, Kec. Lembang. Observasi pada penelitian ini akan dilakukan observasi pada proses deteksi penyakit tanaman tomat pada tempat objek studi.

1. Wawancara

Wawancara dilakukan peneliti dengan petugas koordinator di Desa Tani Dompet Dhuafa yaitu Pak Jajang dan seorang petani tomat Pak Mamat. Hasil laporan wawancara dapat dilihat pada lampiran.

* 1. Metode Pengembangan Perangkat Lunak

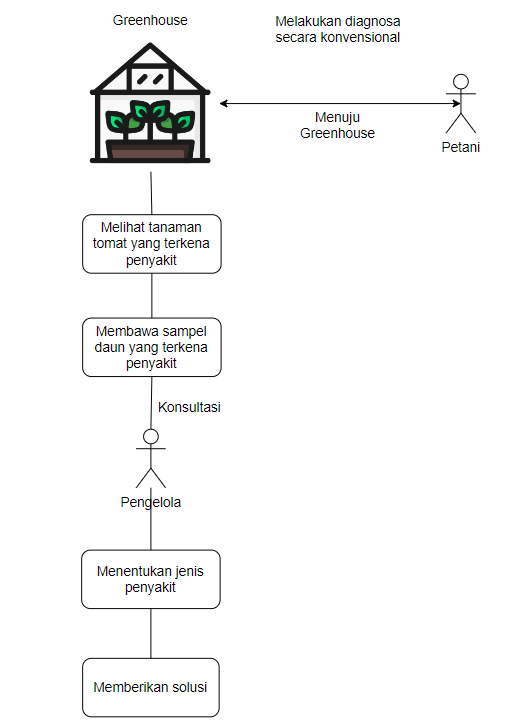
Penelitian yang dilakukan penulis membutuhkan suatu perancangan untuk membuat sistem berjalan dengan semestinya. Sistem yang dibuat mengacu pada model *Extreme Programming.* Alasan penulis menggunakan metodologi ini karena memungkinkan pengguna mengetahui seperti apa sistem yang dibuat sehingga sistem dapat berjalan dengan baik sesuai dengan kebutuhan pengguna dan juga metode ini dapat menghemat waktu dalam pengembangan sistem**.** Berikut merupakan alur model *Extreme Programming*:

* + 1. *Planning* (Perencanaan)

Tahapan perencanaan akan menjelaskan mengenai kebutuhan sistem yang akan dibangun dalam proses pengembangan aplikasi.

* + - 1. Proses Yang Sedang Berjalan

Alur proses diagnosa penyakit tanaman tomat yang sedang berjalan, petani harus melakukan diagnosa secara konvensional. Alur proses di gambarkan melalui diagram sederhana di bawah ini:

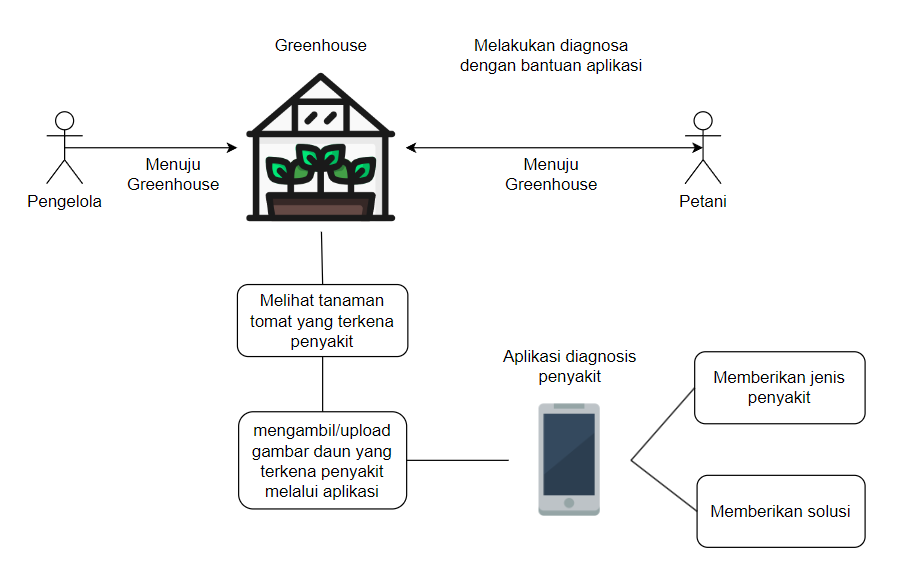


Gambar ... Proses yang sedang berjalan

(Sumber Pribadi)

* + - 1. Proses Yang Diusulkan

Alur Proses yang penulis ajukan pada Desa Tani Dompet Dhuafa akan mendiagnosis penyakit pada tanaman tomat. Berikut alur prosesnya:



Gambar ... Proses yang diusulkan

(Sumber Pribadi)

* + - 1. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional akan menjelaskan perihal fitur dan fungsi yang disediakan oleh sistem. Analisis kebutuhan fungsional sistem dapat dilihat pada tabel di bawah ini:

Tabel ... Kebutuhan Fungsional

(Sumber Pribadi)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kebutuhan Pengguna | Kebutuhan dan Prioritas Fungsional | | | |
| Prioritas Tinggi | Prioritas Sedang | Prioritas Rendah | Tidak ada Prioritas |
| Fitur Diagnosa penyakit tanaman tomat | Pengguna dapat melakukan diagnosis penyakit tanaman tomat melalui gambar daun yang di *upload* pada sistem | Pengguna dapat melakukan diagnosis penyakit tanaman tomat melalui gambar daun yang di foto melalui aplikasi |  |  |
| Fitur informasi penggunaan aplikasi | Pengguna dapat melihat informasi bagaimana cara menggunakan aplikasi secara keseluruhan | Pengguna dapat melihat informasi cara pengambilan gambar untuk hasil prediksi yang lebih akurat |  |  |
| Fitur detail penyakit dan solusi | Setelah melakukan prediksi pengguna dapat melihat lebih lanjut tentang penyakit yang telah diprediksi beserta solusinya |  |  |  |
| Kelola pengambilan gambar | Pengguna dapat mengambil gambar dengan kamera *handphone* atau mengambil gambar dari penyimpanan |  |  |  |

* + - 1. Analisis Kebutuhan Non-Fungsional

Pada tahapan analisis ini akan menentukan atribut dan kualitas yang harus dimiliki sistem yang akan dibangun. Berikut merupakan tabel dari hasil analisis kebutuhan non fungsional pada sistem:

Tabel ... Kebutuhan Non-Fungsional

(Sumber Pribadi)

|  |  |  |
| --- | --- | --- |
| Kualitas Atribut | Kebutuhan Non-Fungsional dan Prioritas | |
| Definisi Kebutuhan | Ruang Lingkup/Bagaimana |
| *Availability* | Sistem dapat digunakan pada kondisi jaringan terputus dengan internet | Membangun aplikasi yang dapat diakses secara *offline* maupun *online* |
| *Performansi* | Aplikasi memiliki batasan OS | Pembuatan aplikasi mempunyai spesifikasi OS mulai dari Android OS versi 4.0 hingga android OS versi 11 |
| *Usability* | Rancangan *interface* yang mudah dipahami pengguna | Implementasi tampilan yang mudah diingat pengguna dan menerapkan tampilan sederhana |

* + - 1. Iterasi Pengembangan Sistem

Tahapan Iterasi pengembangan sistem dilakukan agar pengembang mampu mengembangkan aplikasi secepat mungkin dengan risiko yang rendah. Pada penelitian ini penulis akan melakukan 2 kali iterasi, dengan fokus pada pengembangan aplikasi.

* 1. Iterasi Pertama

Iterasi pertama adalah membangun fitur utama sistem yang akan digunakan pengguna. Adapun perancangan fitur utama sistem adalah seperti yang digambarkan pada tabel ... berikut ini:

Tabel ... Iterasi Pertama

(Sumber Pribadi)

|  |  |
| --- | --- |
| Splash screen | Fungsi yang disediakan adalah tampilan awal sebelum masuk pada sistem |
| Header/AppBar | Label nama aplikasi |
| Navigation bar | Menampilkan menu yang dapat di akses pengguna:  -Menu Beranda  -Menu diagnosis penyakit |
| Halaman Prediksi | Halaman ini memuat:  -Label halaman  -Tombol kamera  -Tombol *upload file* gambar  -Tampilan memuat gambar  -Tombol Proses diagnosa |
| Halaman Hasil Prediksi | Halaman ini terdapat:  -Tampilan gambar  -Hasil diagnosis (berupa nama jenis penyakit)  -Informasi penyakit dan solusi dari penyakit |
| Halaman Beranda | Menampilkan informasi yang berhubungan dengan apliksai yang dibangun yaitu slogan aplikasi, tutorial penggunaan aplikasi, informasi dataset yang dikumpulkan dan informasi cara pengambilan gambar yang baik. |

* 1. Iterasi Kedua

Iterasi kedua merupakan pembuatan model deep learning menggunakan algoritma CNN dengan arsitektur *transfer learning* NasNetMobile dan implementasi model pada sistem aplikasi. Adapun perancangan fitur ini, seperti yang digambarkan pada tabel ... berikut ini:

Tabel ... Iterasi Kedua

(Sumber Pribadi)

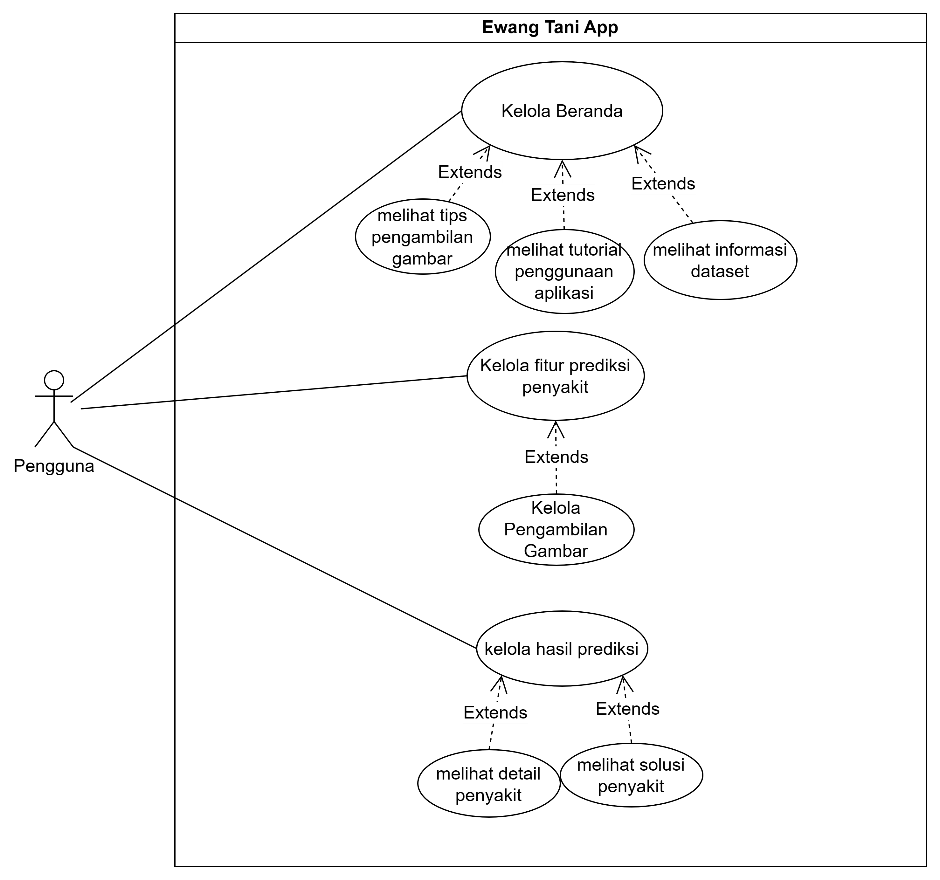
|  |  |
| --- | --- |
| Mengelola Data | Data yang digunakan pada model deep learning yaitu gambar daun tanaman tomat yang terkena penyakit dan daun yang sehat. Sebelum data digunakan akan dilakukan proses pengelolaan data sesuai kebutuhan |
| Membangun model | Setelah data siap untuk digunakan, implementasikan data tersebut pada model deep learning menggunakan algoritma CNN serta dilakukan evaluasi pada model yang sudah dibangun |
| Save model dan konverter model | Model yang sudah dibangun akan disimpan dan selanjutnya akan di konverter menjadi file dengan format .tflite |
| Implementasi model pada sistem aplikasi mobile | Membangun struktural kode untuk menambahkan model pada aplikasi agar dapat dijalankan. |

* + 1. *Design* (Desain)

Pada tahapan Desain menjelaskan mengenai kerangka aplikasi yang akan dibuat. Rancangan kerangka aplikasi akan dibuat dengan metode pemodelan UML (*Unified Modelling Language*) diagram dan tampilan *interface* sistem.

1. *Use Case Diagram*

Diagram ini menjelaskan aktor yang terlibat dengan sistem dan tahapan apa saja yang dapat dilakukan pengguna terhadap sistem. Aktor yang terlibat yaitu satu pengguna. Diagram yang dimaksudkan dapat dilihat pada gambar ... di bawah ini:



Gambar .... *Use Case* Diagram Sistem

(Sumber Pribadi)

1. *Use Case Scenario*

Pada *Use Case Scenario* akan mendeskripsikan proses pada *Use Case Diagram* yang telah dibuat dengan terperinci, serta diagram yang dijelaskan pada tabel di bawah ini akan menjelaskan alternatif kesalahan atau perilaku yang akan ditampilkan apabila sistem mengalami kesalahan.

1. *Use case Scenario* Kelola Beranda

*Case* Kelola Beranda memiliki aktor utama yaitu Pengguna dan memiliki hubungan *extend* dengan *use case* Akses Informasi Penyakit Tanaman Tomat. Lebih jelasnya dapat dilihat pada tabel .... dibawah ini:

Tabel .... *Use Case Scenario* Mengelola Beranda

(Sumber Pribadi)

|  |  |  |
| --- | --- | --- |
| **Nama *Use Case* :** Mengelola Beranda | **IDE :** UC 01 | **Level :** Prioritas |
| **Aktor utama:** Pengguna  **Aktor kedua:** | | **Tipe *Use Case*:** UC Utama |
| **Tujuan:**  Menandakan pengguna dapat mengakses sistem dan dapat mengelola informasi yang ada, serta sistem terindikasi berjalan dengan baik | | |
| **Deskripsi:**  Dalam *use case* ini pengguna dapat melihat tampilan awal sistem yaitu beranda dan pengguna dapat mengakses informasi-informasi yang terdapat beranda | | |
| **Penyebab:** Pengguna ingin masuk ke dalam sistem dan pengguna ingin melihat informasi pada halaman beranda  ***Type*:** *External* | | |
| **Hubungan:**  **Asosiasi:** Pengguna  ***Include*:** -  ***Extend*:** melihat informasi tips pengambilan gambar, melihat tutorial penggunaan aplikasi, melihat informasi dataset | | |
| **Alur Normal:**   1. Membuka aplikasi 2. Sistem menampilkan *splash screen* 3. Pengguna masuk pada halaman beranda | | |
| **Alur Alternatif:**  2A. Tampilan *loading screen* tidak muncul  3A. Pengguna tidak dapat masuk aplikasi | | |

1. *Use case Scenario* Kelola Fitur Prediksi Penyakit

Pada *case* ini aktor pengguna mengakses fitur prediksi penyakit serta *case* ini memiliki hubungan *extend* dengan *case* kelola pengambilan gambar. Penjelasan *case* ini dapat dilihat pada tabel ... di bawah ini:

Tabel .... *Use Case Scenario* Kelola Fitur Prediksi Penyakit

(Sumber Pribadi)

|  |  |  |
| --- | --- | --- |
| **Nama *Use Case* :**  Kelola Fitur Diagnosa Penyakit | **IDE :** UC 02 | **Level :** Prioritas |
| **Aktor utama:** Pengguna  **Aktor kedua:** | | **Tipe *Use Case*:** UC Utama |
| **Tujuan:**  Pengguna dapat melakukan prediksi penyakit tanaman tomat melalui gambar daun tomat yang di ambil melalui kamera atau yang di unggah pada penyimpanan galeri. | | |
| **Deskripsi:**  Dalam *use case* ini menjelaskan bahwa pengguna dapat melakukan prediksi penyakit tanaman tomat melalui gambar daun tomat. | | |
| **Penyebab:** Pengguna ingin melakukan prediksi penyakit tanaman tomat melalui gambar daun tomat.  ***Type*:** *External* | | |
| **Hubungan:**  **Asosiasi:** Pengguna  ***Include*:** -  ***Extend*:** Kelola pengambilan gambar | | |
| **Alur Normal:**   1. Membuka aplikasi 2. Pilih menu diagnosa pada *navigation bar* 3. Tekan tombol Pilih Gambar 4. Pilih salah satu metode pengambilan gambar 5. Setelah selesai tekan tombol Prediksi Gambar untuk mulai proses prediksi 6. Menampilkan hasil prediksi | | |
| **Alur Alternatif:**  2A. Pengguna pilih menu beranda  5A. Pengguna kembali pada halaman beranda | | |

1. *Use Case Scenario* Kelola Hasil Prediksi

Pada *scenario* ini akan menjelaskan *Use Case* utama yaitu kelola hasil prediksi*. Use Case* ini memuat informasi hasil dari proses prediksi. Penjelasan lebih lanjut dapat dilihat pada tabel .... di bawah ini:

Tabel .... *Use Case Scenario* Kelola Hasil Prediksi

(Sumber Pribadi)

|  |  |  |
| --- | --- | --- |
| **Nama *Use Case* :**  Kelola Hasil Prediksi | **IDE :** UC 03 | **Level :**  Prioritas |
| **Aktor utama:** Pengguna  **Aktor kedua:** | | **Tipe *Use Case*:**  UC Utama |
| **Tujuan:**  Pengguna dapat melihat informasi penyakit dari gambar yang diprediksi serta nilai persentase keakuratan prediksi | | |
| **Deskripsi:**  Dalam *use case* ini menjelaskan bahwa pengguna dapat melihat hasil dari proses prediksi gambar dengan dilengkapi tombol untuk melihat detail penyakit serta solusi dari penyakit. | | |
| **Penyebab:** Pengguna ingin melihat hasil proses prediksi  ***Type*:** *External* | | |
| **Hubungan:**  **Asosiasi:** Pengguna  ***Include*:** -  ***Extend*:** - | | |
| **Alur Normal:**   1. Membuka aplikasi 2. Pilih menu prediksi melalui *navigation bar* 3. Pilih gambar yang akan diprediksi 4. Tekan tombol Prediksi Gambar untuk memulai proses prediksi 5. Pengguna melihat informasi hasil dari prediksi | | |
| **Alur Alternatif:**  2A. Pengguna pilih menu lain  3A. Pengguna kembali pada halaman beranda | | |

1. *Use Case Scenario* Kelola pengambilan gambar

Pada *scenario* ini akan menjelaskan *Use Case* tambahan yaitu kelola pengambilan gambar. Penjelasan lebih lanjut dapat dilihat pada tabel .... di bawah ini:

Tabel .... *Use Case Scenario* Kelola Pengambilan Gambar

(Sumber Pribadi)

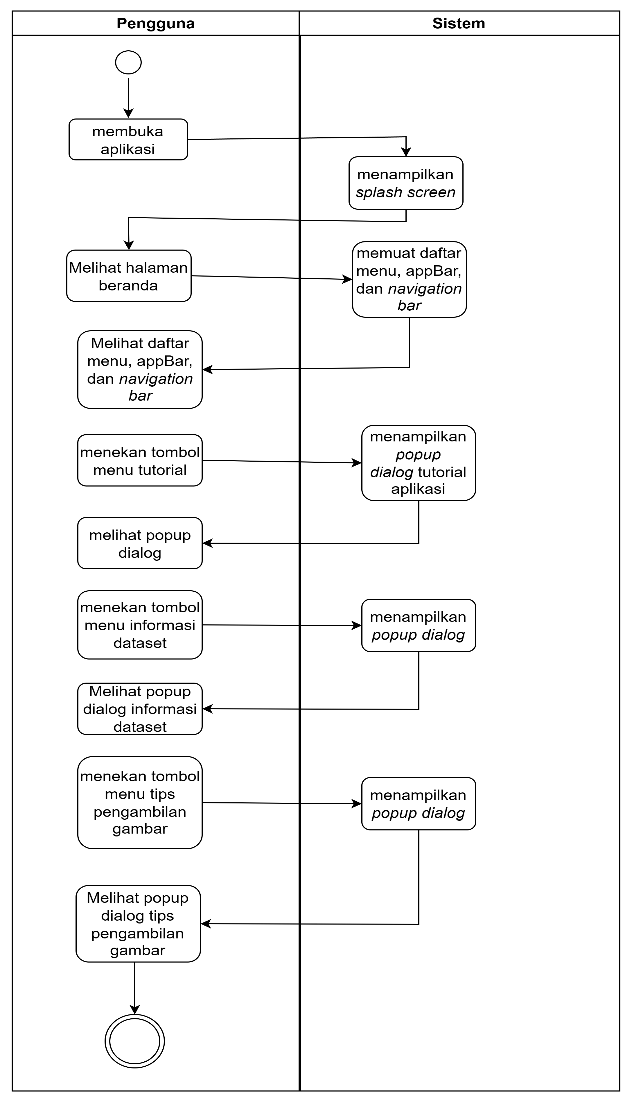
|  |  |  |
| --- | --- | --- |
| **Nama *Use Case* :**  Kelola pengambilan Gambar | **IDE :** UC 06 | **Level :**  Prioritas |
| **Aktor utama:** Pengguna  **Aktor kedua:** | | **Tipe *Use Case*:**  UC Tambahan |
| **Tujuan:**  Pengguna dapat memilih metode pengambilan gambar | | |
| **Deskripsi:**  Dalam *use case* ini menjelaskan bahwa pengguna dapat memilih metode pengambilan gambar. Metode dapat berupa unggah gambar dari penyimapanan galeri atau pengambilan gambar melalui kamera | | |
| **Penyebab:** Pengguna ingin melakukan prediksi penyakit  ***Type*:** *External* | | |
| **Hubungan:**  **Asosiasi:** Pengguna, Kelola fitur prediksi penyakit  ***Include*:** -  ***Extend*:** - | | |
| **Alur Normal:**   1. Membuka aplikasi 2. Pilih menu prediksi 3. Memilih metode pengambilan gambar | | |
| **Alur Alternatif:**  2A. Pengguna keluar dari fitur prediksi | | |

1. *Activity Diagram*

Pada diagram ini akan menjelaskan aktivitas pengguna secara bertahap pada aplikasi dan menjelaskan alur kerja dari aktivitas aplikasi sesuai dengan *Use Case* yang telah ada.

1. *Activity* Kelola Beranda

*Activity* ini menjelaskan tentang alur dari *Case* Kelola Beranda. Pada case Kelola Beranda melibatkan aktor/pengguna dan juga sistem. Aktor dapat melihat informasi yang berkaitan dengan aplikasi seperti tutorial menggunakan aplikasi, informasi dataset dan tips pengambilan gambar daun tomat. Lebih lanjutnya dapat dilihat pada gambar .... berikut ini:

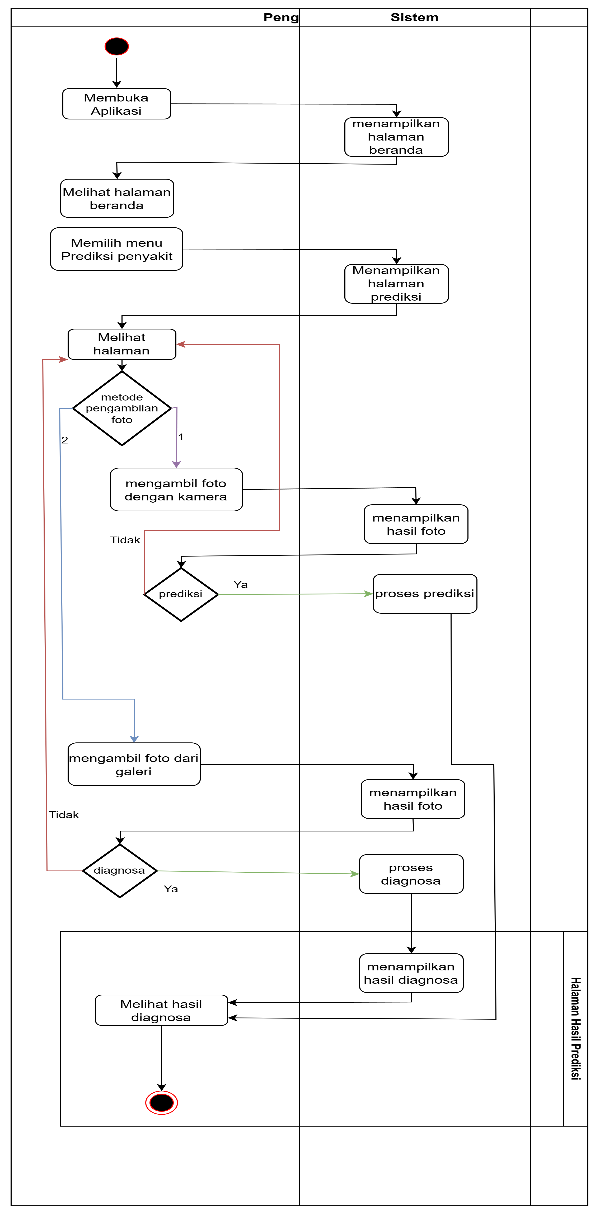


Gambar .... *Activity* Kelola Beranda

(Sumber Pribadi)

1. *Activity* Kelola Fitur Prediksi Penyakit

*Activity* ini menjelaskan tentang alur dari *Case* Kelola Fitur Prediksi Penyakit, melibatkan aktor dengan sistem. Aktor dapat melakukan diagnosa penyakit tanaman tomat dengan memberi *input*-an berupa gambar yang di unggah pada aplikasi. Metode *input* gambar bisa berupa pengambilan gambar dengan kamera dan bisa dengan unggah gambar dari galeri. Lebih jelasnya dapat dilihat pada gambar .... berikut ini:



Gambar .... *Activity* Kelola Fitur Prediksi Penyakit

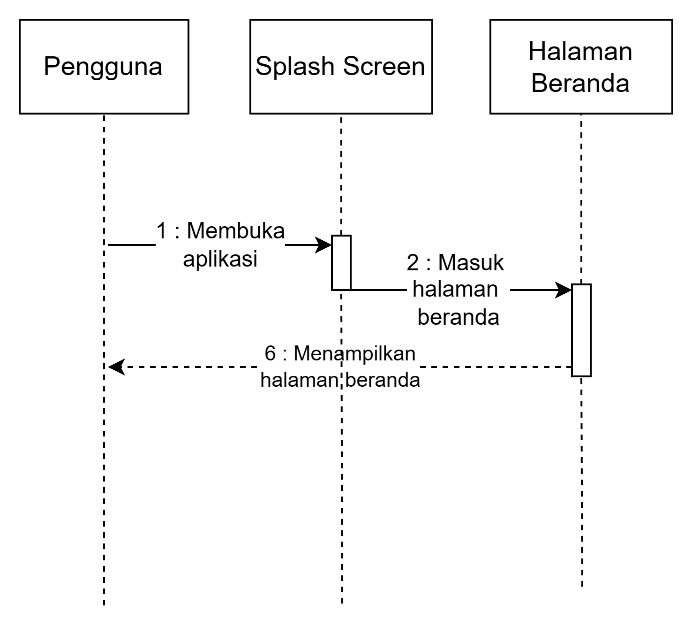
(Sumber Pribadi)

1. *Sequence Diagram*

Pada bagian ini menjelaskan perilaku sistem terhadap urutan waktu yang akan dilalui pengguna. Diagram ini melibatkan antara pengguna dan sistem di setiap prosesnya, serta melibatkan *database* sebagai alur untuk kirim atau terima data dari maupun ke sistem. Proses yang berjalan digambarkan dengan garis petunjuk arah waktu.

1. *Sequence* Beranda

Pada proses ini menjelaskan mengenai alur proses interaksi antara objek dalam sistem saat pengguna masuk aplikasi akan menampilkan halaman beranda. Lebih jelasnya dapat dilihat pada gambar .... di bawah ini:

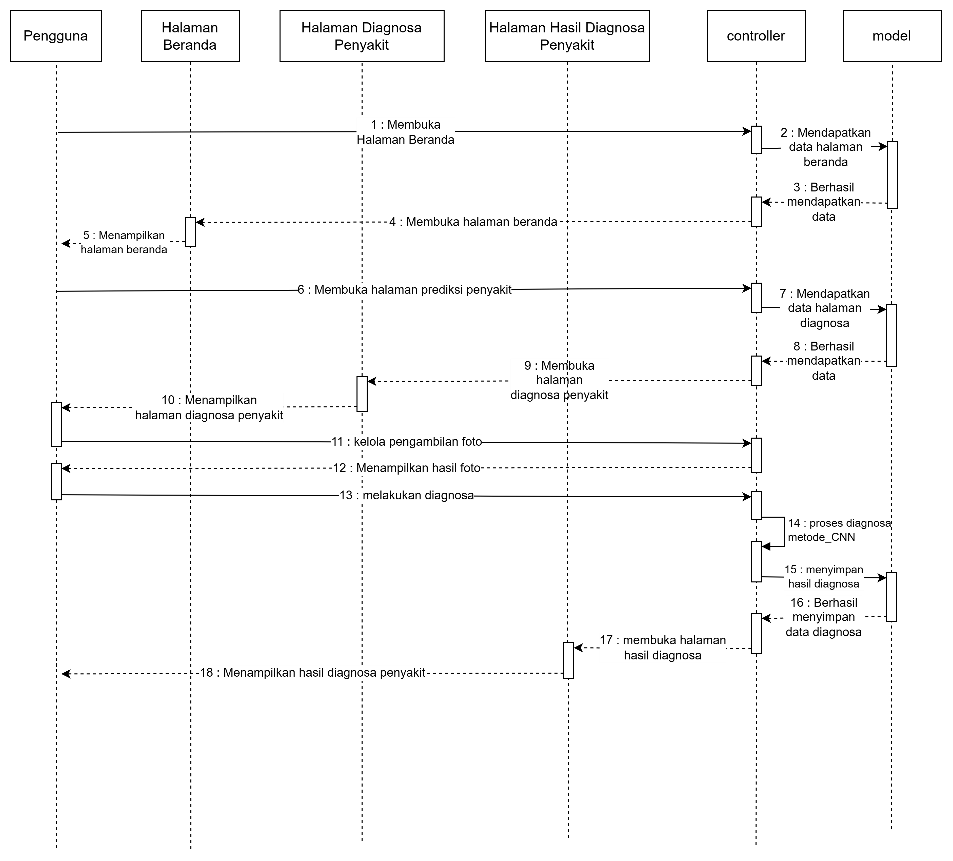


Gambar .... *Sequence* Beranda

(Sumber Pribadi)

1. *Sequence* Diagnosa Penyakit

Diagram *Sequence* ini menjelaskan alur proses yang di jalankan aplikasi untuk mendiagnosa penyakit. Model klasifikasi yang dipakai yaitu *Convolutional Neural Network (CNN)* dengan proses dilakukan pada objek controller, dapat dilihat pada gambar .... berikut:



Gambar .... *Sequence* Diagnosa Penyakit

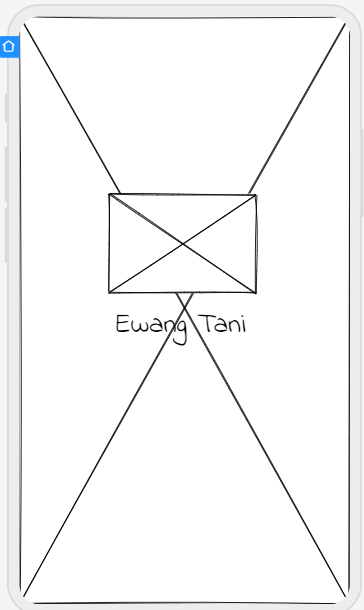
(Sumber Pribadi)

1. *Mockup Design*

Perancangan antar muka sistem bertujuan untuk menjelaskan secara visual bagaimana nantinya desain sistem yang akan peneliti buat.

1. *Mockup Spalsh Screen*

Berikut ini merupakan tampilan *splass* *screen* pada sistem yang dibangun, dapat dilihat pada gambar .... di bawah ini:



Gambar .... *Mockup Splash Screen*

(Sumber Pribadi)

1. Mockup Beranda

Berikut ini merupakan tampilan Beranda pada sistem yang dibangun, dapat dilihat pada gambar .... di bawah ini:

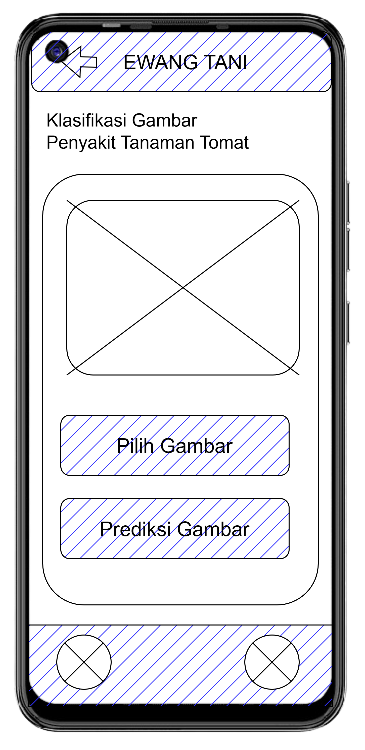


Gambar .... Mockup Beranda

(Sumber Pribadi)

1. Mockup Prediksi

Berikut ini merupakan tampilan Prediksi pada sistem yang dibangun, dapat dilihat pada gambar .... di bawah ini:



Gambar .... Mockup Prediksi

(Sumber Pribadi)

1. Mockup Hasil Prediksi

Berikut ini merupakan tampilan Hasil Prediksi pada sistem yang dibangun, dapat dilihat pada gambar .... di bawah ini:



Gambar .... Mockup Hasil Diagnosa

(Sumber Pribadi)

1. Mockup *Popup* Menu Tutorial Prediksi

Berikut ini merupakan tampilan *popup* untuk menu tutorial prediksi pada aplikasi yang dibangun, dapat dilihat pada gambar .... di bawah ini:



Gambar .... Mockup Popup Menu Tutorial Prediksi

(Sumber Pribadi)

1. *Mockup* *Popup* Menu Dataset

Berikut ini merupakan tampilan dari *popup* menu dataset pada sistem yang dibangun, dapat dilihat pada gambar .... di bawah ini:

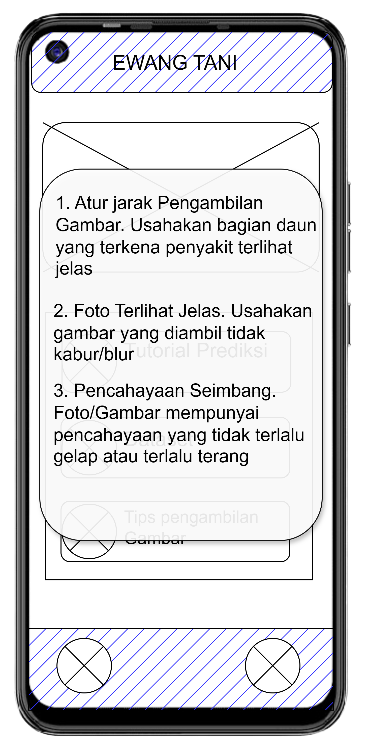


Gambar .... *Mockup* *Popup* Menu Dataset

(Sumber Pribadi)

1. *Mockup Popup* Menu Tips Pengambilan Gambar

Berikut ini merupakan tampilan dari popup menu tips pengambilan gambar yang ada di halaman beranda, dapat dilihat pada gambar .... di bawah ini:



Gambar .... *Mockup* *Popup* Menu Tips Pengambilan Gambar

(Sumber Pribadi)

1. *Mockup Popup* Menu Detail dan Solusi Penyakit

Berikut ini merupakan tampilan dari *popup* menu detail penyakit yang ada pada halaman hasil prediksi, dapat dilihat pada gambar .... di bawah ini :

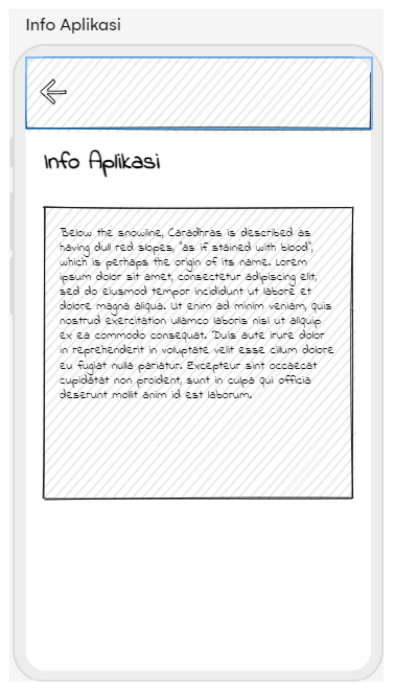


Gambar .... *Mockup Popup* Menu Detail Penyakit

(Sumber Pribadi)

1. *Mockup Popup* Info Aplikasi

Berikut ini merupakan tampilan dari *popup* info aplikasi pada sistem yang dibangun, dapat dilihat pada gambar .... di bawah ini :

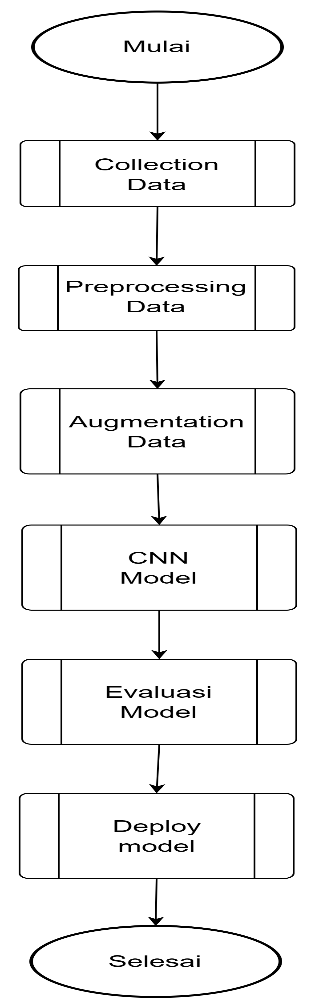


Gambar .... *Mockup Popup* Info Aplikasi

(Sumber Pribadi)

1. Perancangan Metode *Convolutional Neural Network (CNN)*

Penggunaan Metode CNN oleh penulis ini untuk diterapkan pada proses prediksi penyakit tanaman tomat berdasarkan masukan berupa gambar daun tomat. Hasil dari proses prediksi berupa keterangan penyakit yang prediksi, dengan batasan 4 kelas penyakit yang dapat di prediksi oleh pengguna, yaitu Bercak kering, Embun Tepung, Tengorok Daun dan Daun Sehat. Berikut gambar .... di bawah ini adalah *flowchart* yang menggambarkan proses dari Metode CNN:



Gambar .... Alur proses Metode *Convolutional Neural Network (CNN)*

Berdasarkan gambar di atas, berikut ini penjelasan dari masing-masing proses pada Metode CNN yaitu:

1. *Collection* Data

*Collection* data merupakan tahapan pengumpulan dataset yang akan digunakan pada model. Penelitian yang dilakukan bersifat *supervised learning* dimana artinya proses klasifikasi menggunakan data yang sudah ada yang memiliki label dari masing-masing kelas yang sudah ditentukan. Dataset pada penelitian ini memiliki label atau kelas yang dibagi menjadi empat yaitu bercak kering, embun tepung, tengorok daun dan daun sehat.

1. *Preprocessing* Data

Proses ini bertujuan untuk memperbaiki masalah dalam data. Pada dataset yang sudah dikumpulkan, penulis menganalisis kembali parameter atau jenis penyakit apa saja yang banyak dialami tanaman tomat di lapangan, selanjutnya menentukan parameter apa saja yang akan di masukan pada proses modeling, hal terakhir yang penulis lakukan adalah melakukan pembersihan data dengan cara *cropping* gambar agar terlihat jelas jenis penyakit yang ada pada daun tomat.

1. Data *Augmentation*

*Augmentasi* data merupakan proses mengubah atau memodifikasi gambar yang ada agar mendapatkan data tambahan untuk data latih. Tujuan dari proses ini untuk membuat model yang dapat melakukan generalisasi dengan lebih baik.

1. CNN Model

Pembuatan model pada penelitian ini menggunakan metode CNN dibantu dengan Keras Application. Arsitektur yang dipilih yaitu NasNetMobile, dengan epoch 25, optimizer Adam serta satu proses Convulition layer.

1. Evaluasi Model

Tahapan ini secara garis besar melakukan interpretasi terhadap hasil dalam proses pemodelan. Tahap evaluasi dilakukan dengan tujuan agar model yang ditentukan dapat menjawab rumusan masalah yang ada. Proses Evaluasi pada penelitian ini menggunakan *Confusion Matrik.*

1. *Deployment*/*Deploy* model

Proses untuk membuat suatu model dapat tersedia pada lingkungan produksi, artinya model yang sudah dibangun sebelumnya dapat memberikan diagnosa pada perangkat lunak lain. Penelitian ini menggunakan *framework Flask* sebagai *Deploy* modelnya.

* + 1. *Coding* (Pengkodean)

Tahapan Coding ini penulis akan melakukan perancangan sebuah aplikasi dari masalah yang telah diteliti sebelumnya.

1. Pengkodean

Pengkodean adalah proses menerjemahkan desain yang sudah dibuat untuk diolah menjadi suatu bahasa pemrograman yang dimengerti komputer. Sistem yang dibangun menggunakan bahasa pemrograman *Dart* dengan menggunakan *framework* *Flutter* dan *library* untuk implementasi model CNN menggunakan *tflite*\_*fluter*.

1. Perancangan Kamus Data

Perancangan kamus data yang dilakukan peneliti yaitu menggunakan struktur data.

1. Tabel *penyakit*

Rancangan kamus data pada tabel *penyakit* memiliki rancangan yang dapat dilihat pada tabel ... berikut ini:

Tabel .... *penyakit*

(Sumber Pribadi)

|  |  |  |
| --- | --- | --- |
| Nama Field | Type Data | Keterangan |
| penyakit\_id | Big Int(20) | Primary key |
| nama\_penyakit | String(100) |  |
| img\_penyakit | String(100) |  |
| detail\_penyakit | String(100) |  |

1. Tabel *solusi*

Rancangan kamus data pada tabel *solusi* memiliki rancangan yang dapat dilihat pada tabel ... berikut ini:

Tabel .... *solusi*

(Sumber Pribadi)

|  |  |  |
| --- | --- | --- |
| Nama Field | Type Data | Keterangan |
| penyakit\_id | Big Int(20) | Primary key |
| solusi\_id | String(100) | Foreign key |
| jenis\_obat | String(100) |  |
| dosis\_obat | String(100) |  |

* + 1. *Testing* (Pengujian)

Tahapan Testing ini akan dilakukan pengujian perangkat lunak dengan menggunakan teknik *alpha testing* dan *beta testing*.

1. Pengujian Alpha

Pada penelitian ini, penulis melakukan pengujian *alpha* menggunakan metode Black Box Testing atau dikenal sebagai *behavioral testing*. Metode ini cocok diterapkan pada pengujian *alpha* atau pengujian fungsional, karena metode ini melakukan pengujian terhadap fungsi atau fitur spesifik sebuah sistem (Humaria Aliya, 2020). Tahapan pengujian digambarkan dengan skenario pengujian *alpha* yang dapat dilihat pada tabel .... sampai .... sebagai berikut:

Tabel .... Rencana Pengujian *Splash Screen*

(Sumber Pribadi)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | *Scenario* Uji | Hasil yang Diharapkan | Hasil Pengujian | |
| Sesuai | Tidak Sesuai |
| **1** | Membuka aplikasi Ewang Tani | Aplikasi menampilkan halaman *splash screen* selamat 3 detik dengan memuat logo aplikasi |  |  |

Tabel ... Rencana Pengujian Beranda

(Sumber Pribadi)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | *Scenario* Uji | Hasil yang Diharapkan | Hasil Pengujian | |
| Sesuai | Tidak Sesuai |
| **1** | Masuk halaman Beranda | Aplikasi akan menampilkan foto banner yang memuat slogan dari aplikasi. Menampilkan 3 menu dengan bentuk *card widget* yaitu menu Tutorial Prediksi, Dataset dan Prediksi |  |  |
| **2** | Menekan tombol *card* menu Tutorial Prediksi | Aplikasi akan menampilkan *alertDialog* dengan isi informasi tips apa saja dalam pengambilan foto untuk di prediksi |  |  |
| **3** | Menekan tombol Dataset | Aplikasi akan menampilkan *alert dialog widget* dengan isi sebuah informasi dimana saja dataset didapatkan. |  |  |
| **4** | Menekan tombol Prediksi Sekarang | Aplikasi akan membawa pengguna menuju halaman prediksi |  |  |

Tabel ... Rencana Pengujian Prediksi

(Sumber Pribadi)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | *Scenario* Uji | Hasil yang Diharapkan | Hasil Pengujian | |
| Sesuai | Tidak Sesuai |
| **1** | Masuk halaman prediksi | Aplikasi akan menampilkan halaman prediksi dengan dilengkapi icon gambar kosong dan terdapat 2 tombol, yaitu tombol Pilih Gambar serta tombol Prediksi Gambar. |  |  |
| **2** | Menekan Pilih gambar | Aplikasi akan memunculkan *alertDialog* pada bagian bawah tampilan dengan isian berupa pilihan untuk memilih metode pengambilan gambar. |  |  |
| **3** | Menekan pilihan Kamera dari tombol Pilih Gambar | Aplikasi akan menampilkan fungsi kamera yang diakses dari *handphone* pengguna untuk dapat mengambil gambar yang hendak akan diprediksi |  |  |
| **4** | Menekan pilihan Galeri dari tombol Pilih Gambar | Aplikasi akan membawa pengguna menuju penyimpanan galeri pada *handphone* pengguna |  |  |
| **5** | Ketika foto sudah ditentukan dan di upload | Foto yang sudah dipilih akan ditampilkan diatas tombol Pilih Gambar dan Prediksi sekarang, menggantikan icon gambar kosong. |  |  |
| **6** | Pengguna melakukan prediksi dengan menekan tombol Prediksi Gambar | Aplikasi akan memproses gambar yang akan dideteksi melalui model CNN. selanjutnya hasil prediksi akan diteruskan pada halaman Hasil Prediksi. |  |  |

Tabel .... Rencana Pengujian Hasil Prediksi

(Sumber Pribadi)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | *Scenario* Uji | Hasil yang Diharapkan | Hasil Pengujian | |
| Sesuai | Tidak Sesuai |
| **1** | Masuk halaman Hasil Prediksi | Setelah proses prediksi selesai selanjutnya hasil dari prediksi akan ditampilkan pada halaman ini. Dengan menampilkan gambar penyakit, jenis nama penyakit dan nilai persentase kepercayaan. Selain itu terdapat tombol Detail Penyakit |  |  |
| **2** | Menekan tombol Detail Penyakit | Aplikasi akan menampilkan *alertDialog* dengan isi informasi detail dari penyakit yang telah diprediksi dan dilengkapi dengan keterangan solusi dari penyakit. |  |  |

1. Pengujian Beta

Pada pengujian Beta penulis menggunakan pendekatan *Technology Acceptance Model* (TAM). Rencana pengujian beta ini akan dilakukan dengan menyebarkan kuesioner kepada objek studi yaitu kelompok petani Di Desa Tani, Kab. Bandung Barat. Sebagai data pendukung akan disiapkan demo sistem yang dibangun agar memudahkan petani untuk memahami sistem. Dalam pengujian terdapat 3 variabel yang akan di uji yaitu pengaruh Persepsi Kemudahan Penggunaan (*Perceived Ease of Use*), Persepsi Kegunaan (*Perceived Usefullness*) dan penerimaan teknologi (*Acceptance of IT*). Adapun yang akan melakukan pengujian beta pada sistem yang dibangun adalah petani tomat dan hasil pengujian akan dikelola menggunakan aplikasi IBM SPSS. Berikut pertanyaan yang terdapat pada kuesioner dapa t dilihat pada tabel ... di bawah ini.

Tabel .... Rencana Pengujian Beta

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Pertanyaan | | Alternatif Jawaban | | | | | | | | | Sumber | |
| STS | TS | | | N | S | | SS | |
| *Perceived Ease of Use* (PEU) | | | | | | | | | | | | | |
| 1 | Apakah aplikasi ini mudah untuk dipelajari? | |  |  | | |  |  | |  | | (Flourensia, 2017) | |
| 2 | Apakah aplikasi ini mudah untuk dipahami/dimengerti? | |  |  | | |  |  | |  | | (Flourensia, 2017) | |
| 3 | Apakah aplikasi ini mudah untuk digunakan? | |  |  | | |  |  | |  | | (Flourensia, 2017) | |
| 4 | Apakah aplikasi ini mudah untuk diingat? | |  |  | | |  |  | |  | | (Flourensia, 2017) | |
| *Perceived Usefulness* (PU) | | | | | | | | | | | | | |
| 1 | Apakah aplikasi ini memberikan informasi dan saran yang membantu Anda membudidayakan tanaman tomat? | |  |  | | |  |  | |  | | (Flourensia, 2017) | |
| 2 | Apakah aplikasi ini memberikan solusi yang efektif untuk masalah penyakit tanaman tomat? | |  |  | | |  |  | |  | | (Flourensia, 2017) | |
| 3 | Apakah aplikasi ini membuat proses diagnosa penyakit tanaman tomat menjadi lebih cepat dan lebih mudah? | |  |  | | |  |  | |  | | (Flourensia, 2017) | |
| *Attitude Toward Using* (ATU) | | | | | | | | | | | | | |
| 1 | Apakah anda merasa senang untuk menggunakan aplikasi ini? | |  |  | | |  |  | |  | | (Agus Mulyanto, 2020) | |
| 2 | Apakah anda menikmati alur dari aplikasi ini? Sehingga berencana untuk menggunakan aplikasi ini? | |  |  | | |  |  | |  | | (Agus Mulyanto, 2020) | |
| 3 | Apakah anda merasa bosan menggunakan aplikasi ini? | |  |  | | |  |  | |  | | (Agus Mulyanto, 2020) | |
| 4 | Apakah anda tidak suka dengan aplikasi ini? | |  |  | | |  |  | |  | | (Agus Mulyanto, 2020) | |
| *Behavioral Intention* (BI) | | | | | | | | | | | | | |
| 1 | | Apakah Anda berniat menggunakan aplikasi ini? |  | |  |  | | |  | |  | |  |
| 2 | | Apakah anda akan selalu menggunakan Aplikasi ini untuk melakukan proses budidaya tomat? |  | |  |  | | |  | |  | |  |
| 3 | | Apakah anda akan menggunakan Aplikasi ini kapan saja saat butuh untuk melakukan proses budidaya tomat? |  | |  |  | | |  | |  | |  |
| 4 | | Apakah anda akan tetap menggunakan Aplikasi ini untuk proses budidaya tomat? |  | |  |  | | |  | |  | |  |
| *Actual System Use* (AU) | | | | | | | | | | | | | |
| 1 | | Secara keseluruhan apakah anda merasa puas dengan kinerja Aplikasi ini? |  | |  |  | | |  | |  | |  |
| 2 | | Apakah anda menghabiskan waktu cukup lama dalam menggunakan Aplikasi ini? |  | |  |  | | |  | |  | |  |

* + 1. *Release System*

Tahapan terakhir adalah Rilis sistem Tahapan ini dapat dilakukan setelah semua tahapan selesai, mulai dari *planning*, *desain*, *coding*, *testing* *Alpha* dan *testing* *Beta*. Semua tahapan tersebut dilakukan untuk memastikan semua fitur pada sistem sudah dapat dipakai dan sesuai dengan kebutuhan pengguna. Setelah melewati tahapan-tahapan tersebut sistem dapat di rilis untuk digunakan oleh pengguna*,* dengan cara di *deploy* dan di *hosting* ke server.

# BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

* 1. **Implementasi Sistem**

Sub bab implementasi sistem akan menjabarkan hasil pengembangan aplikasi, pada tahapan ini implementasi dibagi menjadi dua bagian yaitu implementasi model Machine Learning dan implementasi perangkat lunak.

* + 1. Implementasi Model

Sub bab ini menjabarkan implementasi dari alur model yang digunakan. Dalam penelitian ini terdapat empat tahapan dimulai dari *collection* data (pengumpula data), preprocessing data (persiapan data sebelum diporses), augmentation data (memodifikasi dataset gambar), CNN model (perancangan dan pelatihan model), evaluasi model dan yang terakhir tahap deploy model. Penjelasan dari tiap-tiap tahapan akan dijelaskan sebagai berikut:

1. Collection data

Dataset yang digunakan pada penelitian ini berupa gambar daun tanaman tomat yang ditentukan menjadi empat kelas yaitu bercak kering, embun tepung, tengorok daun dan daun sehat. Dataset ini didapatkan dari tempat objek studi Desa Tani dan dari website Kaggle. Berikut detail dataset yang telah dikumpulkan dapat dilihat pada tabel ... dibawah ini:

Tabel ... Detail Dataset

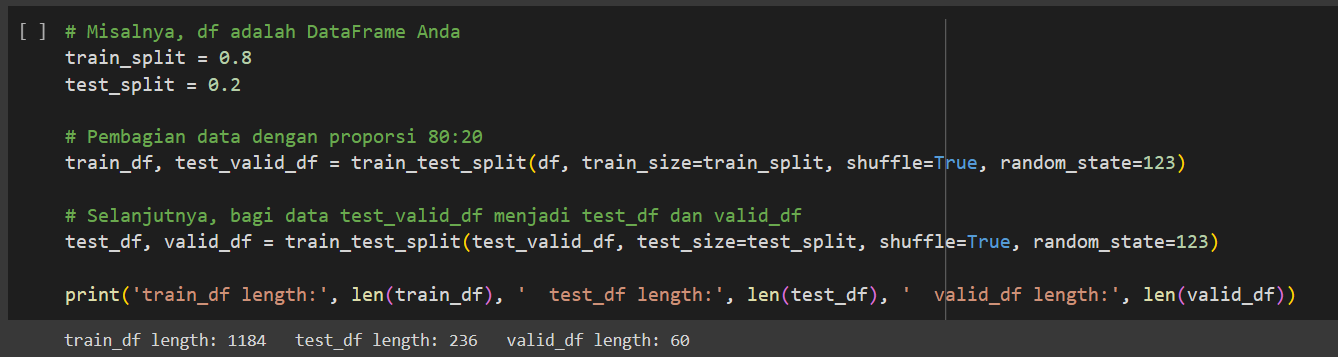
|  |  |
| --- | --- |
| **Kelas** | **Jumlah data** |
| Bercak Kering | 370 |
| Embun Tepung | 370 |
| Tengorok Daun | 370 |
| Daun Sehat | 370 |
| **Total** | **1480** |

1. Preprocessing data

Preprocessing data merupakan tahap persiapan sebelum dataset digunakan untuk proses klasifikasi. Tahapan ini akan dilakukan pembagian data untuk digunakan pada data training dan data testing. Berikut detail pembagian data dapat dilihat pada tabel ... .

|  |  |  |
| --- | --- | --- |
| **Dataset** | **Data Training**  **80%** | **Data Testing**  **20%** |
| Bercak Kering | 296 | 74 |
| Embun Tepung | 296 | 74 |
| Tengorok Daun | 296 | 74 |
| Daun Sehat | 296 | 74 |
| **Total** | **1184** | **296** |

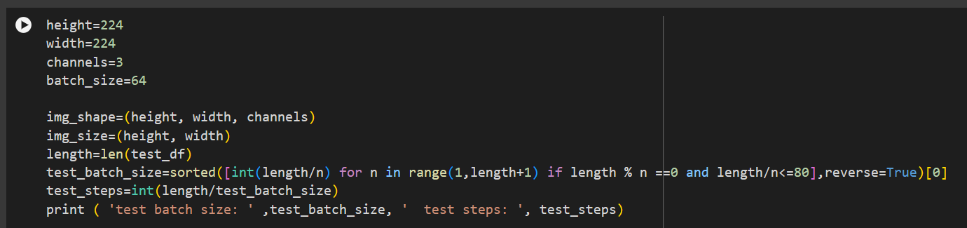
Kode untuk processing data dalam pembagian data untuk di latih dan di uji dapat dilihat pada gambar ... dibawah ini:



Gambar ... Kode pembagian dataset

(Sumber Pribadi)

Selanjutnya akan dilakukan oversampling terhadap ukuran gambar, semua gambar yang digunakan sebagai dataset akan dirubah menjadi 224x224 guna untuk mempercepat pemrosesan klasifikasi. Kode oversampling terhadap ukuran gambar dapat dilihat pada gambar ... .

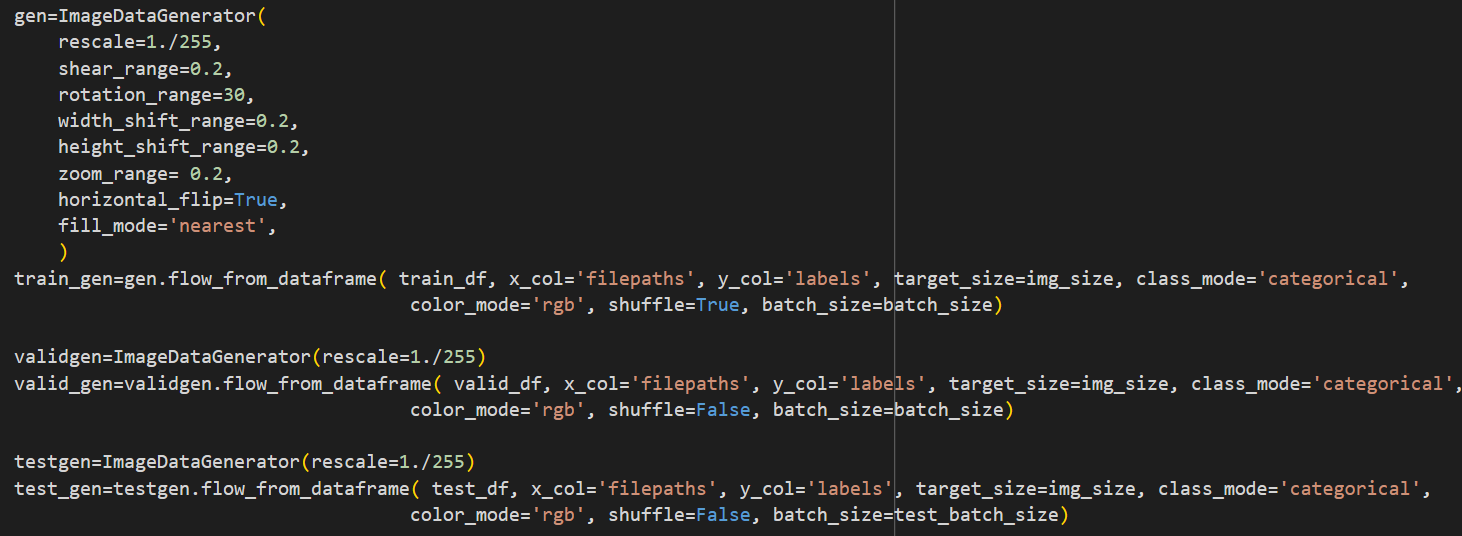


Gambar ... Kode *oversampling* ukuran gambar

(Sumber Pribadi)

1. *Augmentation* data

Tahap *augmentation* data bertujuan untuk memperbanyak data dengan ditambahkan beberapa fitur. Pada penelitian ini tahap *augmentation* data meliputi *rescale, shear range, rotation range, width shift range, height shift range, zoom range, horizontal flip* dan *fill mode* untuk data latih. Sedangkan untuk data uji hanya menggunakan *rescale*. Untuk lebih jelasnnya dapat dilihat kode pada gambar ... .



Gambar ... Kode *augmentation* data

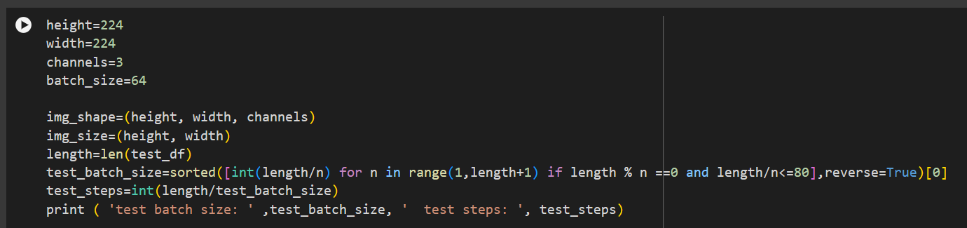
(Sumber Pribadi)

1. CNN model

Struktur kerja model machine learning pada penelitian ini penulis menggunakan algoritma CNN sebagai metode untuk memproses dan menganalisis dataset daun tanaman tomat. Secara garis besar algoritma CNN terdiri dari beberapa layer di antaranya *convulutional layer, pooling layer, dropout layer, Flatten layer dan fully connected layer*. Agar memudahkan memahami cara kerja setiap layer berikut dibawah ini penjelasanya:

1. Input layer

Citra gambar yang telah melalui tahapan preproccesing data dengan dimensi 224x224 dan channel RGB. Kemudian dimasukan kedalam proses *input layer*. Proses ini akan memberikan nilai dimensi 224x224 dengan nilai matriks [-1,0,1] dan *channel* RGB. Berikut kode dari *input layer* dapat dilihat pada gambar ... .

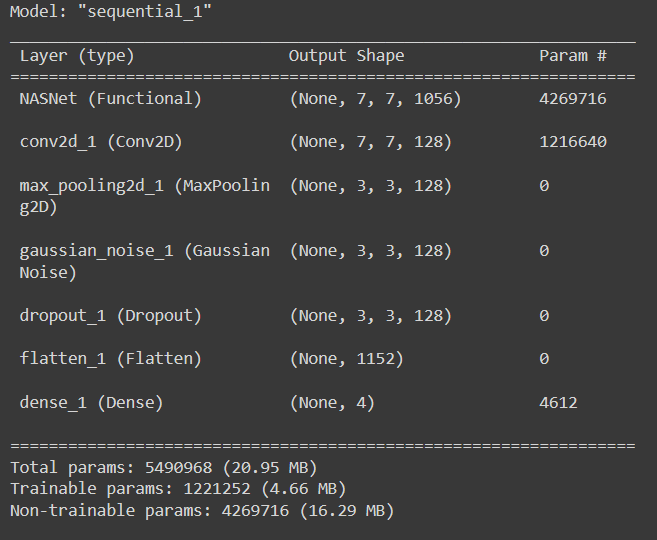


Gambar ... Kode *input layer*

(Sumber Pribadi)

1. Menambahkan model NasNetMobile

NasNetMobile adalah salah satu jenis arsitektur models CNN yang tersedia di *keras application*. Kelebihan dari models ini yaitu pemrosesan training memerlukan beban komputasi yang rendah, serta model yang dihasilkan lebih kecil namun dengan performa terbaik sehingga cocok untuk di deploy pada aplikasi berbasis mobile dengan kinerja aplikasi tidak berat. Berikut tampilan total parameter dari model NasNetMobile dapat dilihat pada gambar ...

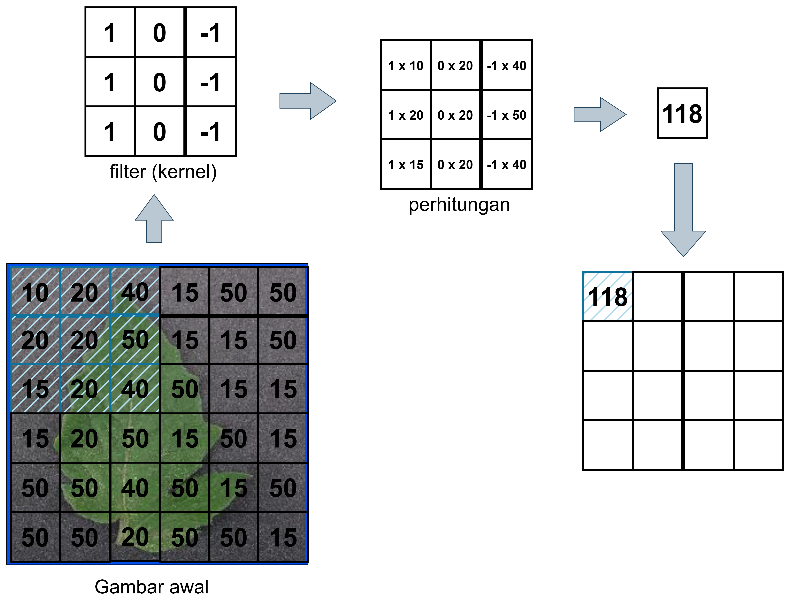


Gambar ... Output tahapan penambahan model NasNetMobile

(Sumber Pribadi)

1. Proses Konvulusi

Tahapan konvulusi (convulution layer) pada penelitian ini memanfaatkan inputan yang didapatkan dari proses penambahan model NasnetMobile yaitu 7x7x1056. Proses konvulusi menggunakan filter 64, kernel 3x3, padding same (0), activation relu, strides 1. Untuk memudahkan pemahaman pada proses ini, berikut proses konvulusi dengan memberikan nilai pada filter dan matrix dapat dilihat pada gambar ... .



Gambar ... Proses Konvulusi

(Sumber Pribadi)

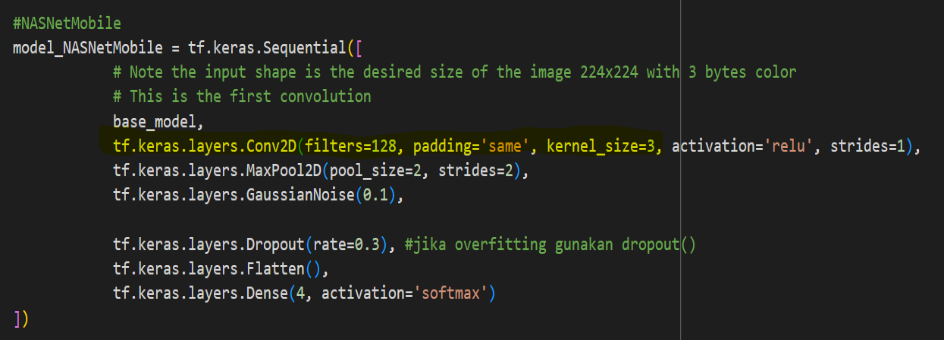
Perhitungan dilakukan pada seluruh matix dengan kernel 3x3, kernel bergeser sesuai strides yang telah ditentukan yaitu 1, artinya pertama kernel akan bergeser secara horizontal dengan 1 langkah selanjutnya kernel juga akan bergeser secara vertical 1 langkah pada seluruh matix. Sehingga ketika perhitungan keseluruhan selesai didapatkan hasil setelah konvulusi baru dapat dilihat pada gambar ... .



Gambar ... Hasil Konvulusi

(Sumber Pribadi)

Proses konvulusi dalam kode dilakukan proses konvulusi satu kali dengan menggunakan 128 filter dengan kernel 3x3 pixel. Berikut gambar penerapan proses konvulsi pada kode dapat dilihat pada gambar ... .

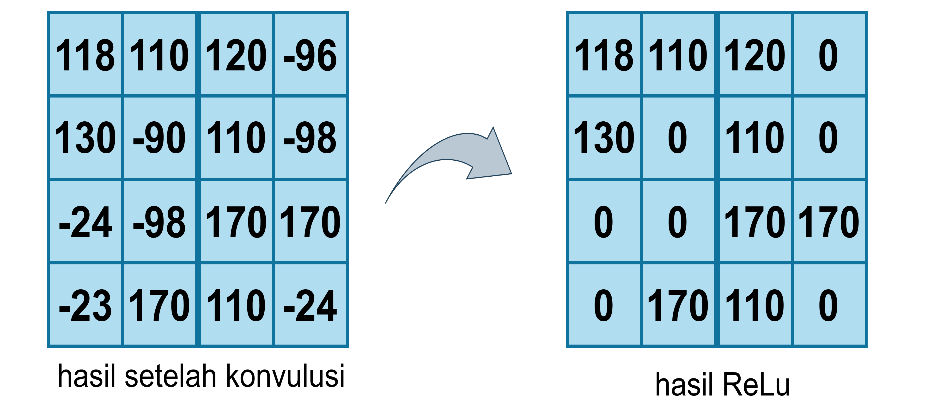


Gambar ... Kode proses konvulusi

(Sumber Pribadi)

1. Proses ReLu

Hasil dari proses konvulusi selanjutnya akan diaktivasi menggunakan fungsi ReLu (Rectified Linear Unit). Fungsi ReLu akan mengonversi nilai-nilai pada setiap pixel yang bernilai negatif menjadi nol, sementara nilai positif tidak akan dirubah. Proses ini dilakukan agar jaringan dapat menangkap relasi yang lebih kompleks di antara fitur-fitur yang di ekstrasi. Berikut proses ReLu dapat dilihat pada gambar ... .

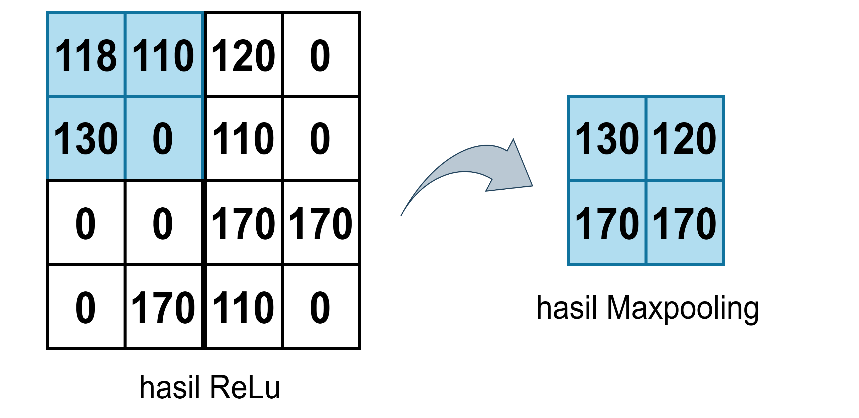


Gambar ... Proses ReLu

(Sumber Pribadi)

1. Proses Pooling

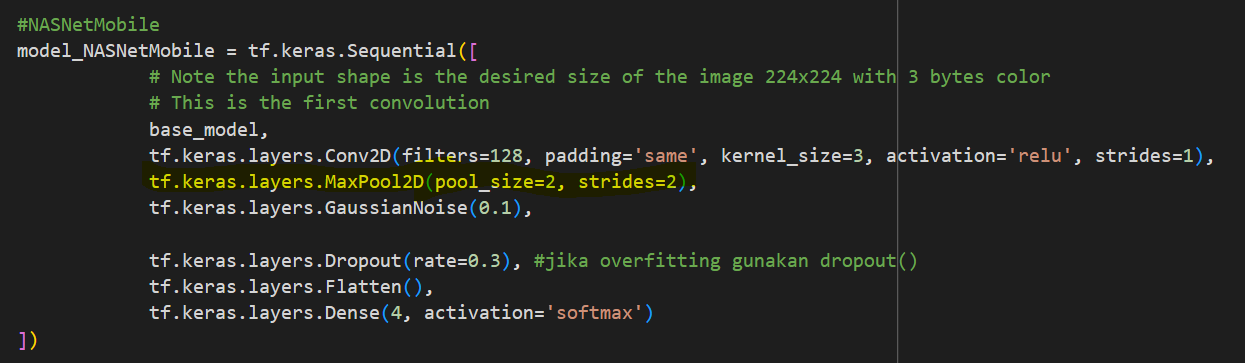
Dari hasil proses sebelumnya selanjutnya akan dijadikan input pada pooling layer, pada tahapan ini penulis menggunakan maxpooling dengan pool size 2x2 dengan strides 2. Maxpolling digunakan untuk mengambil nilai maksimum dari setiap wilayah. Gambar ilustrasi max pooling dapat dilihat pada gambar ... .



Gambar ... Proses pooling

(Sumber Pribadi)

Pada gambar ... dibawah ini merupakan penerapan kode untuk proses pooling dengan menggunakan maxpooling dengan pool size 2x2 dan strides 2.

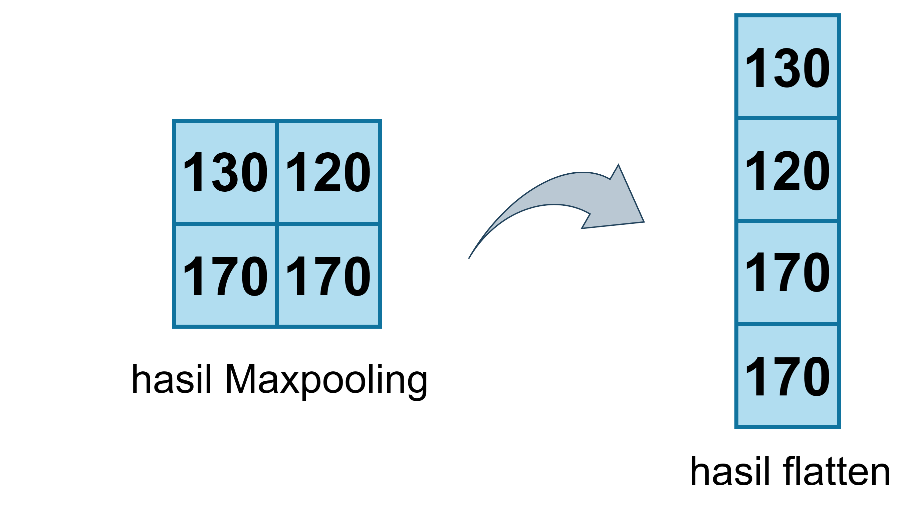


Gambar ... Kode Proses Pooling Layer

(Sumber Pribadi)

1. Proses Flatten

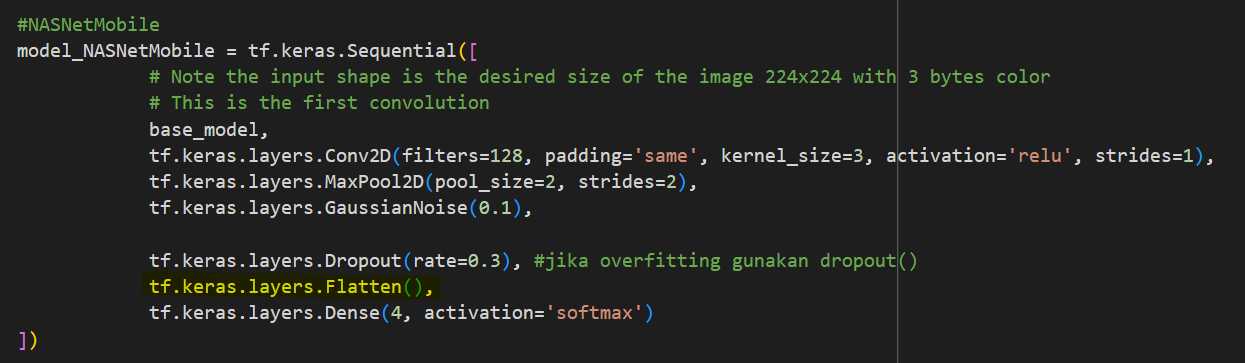
Proses flatten akan mengubah bentuk matrixs kebentuk vector/matrix 1 dimensi. Dari output proses sebelum akan didapatkan bentuk vektor baru, gambar ilustrasi dapat dilihat pada gambar ... dibawah ini.



Gambar ... Proses Flatten

(Sumber Pribadi)

Pada gambar ... dibawah ini merupakan penerapan kode untuk proses flatten.

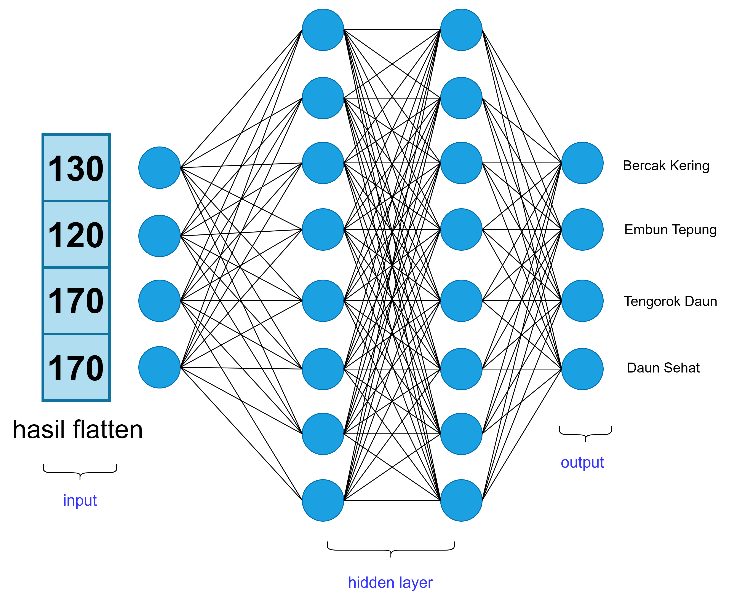


Gambar ... Kode Proses Flatten

(Sumber Pribadi)

1. Proses *Fully Connected Layer*

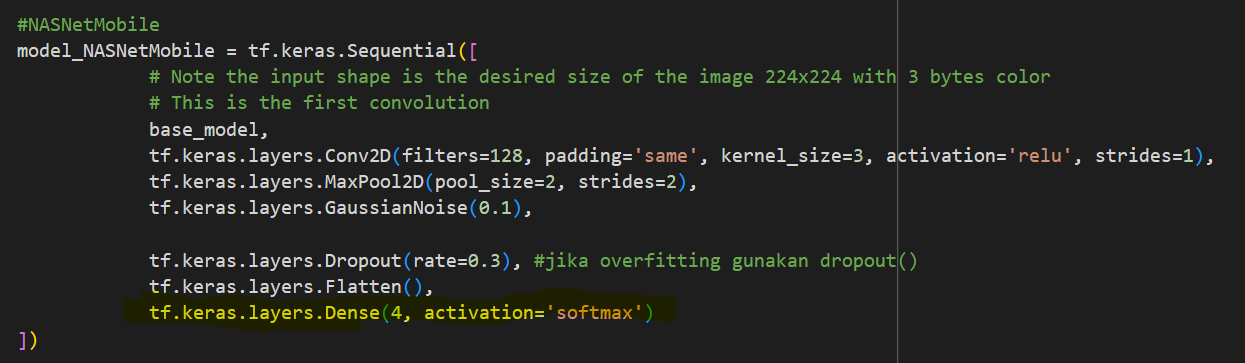
Proses *fully connected layer* merupakan tahapan layer terakhir pada algoritma CNN. Proses ini akan berjalan berdasarkan output dari proses sebelumnya dengan tujuan untuk memungkinkan model dalam memahami fitur-fitur yang diekstraksi secara global dan lebih kompleks. Berikut ilustrasi fully connected layer seperti gambar ... berikut ini:



Gambar ... Proses Fully Connected Layer

(Sumber Pribadi)

Dalam penerapan proses fully connected layer pada kode melibatkan pemrosesan di output layer yang terdiri dari 4 neuron sesuai dengan jumlah kelas dalam model. Pemrosesan melibatkan pencocokan dari output layer sebelumnya menggunakan aktivasi softmax. Pencocokan dilakukan untuk mendapatkan presentasi kecocokan tertinggi atau tingkat kepercayaan (confidence) sebagai hasil akhir. Berikut gambar penerapan proses fully connected layer pada kode dapat dilihat pada gambar ... dibawah ini:

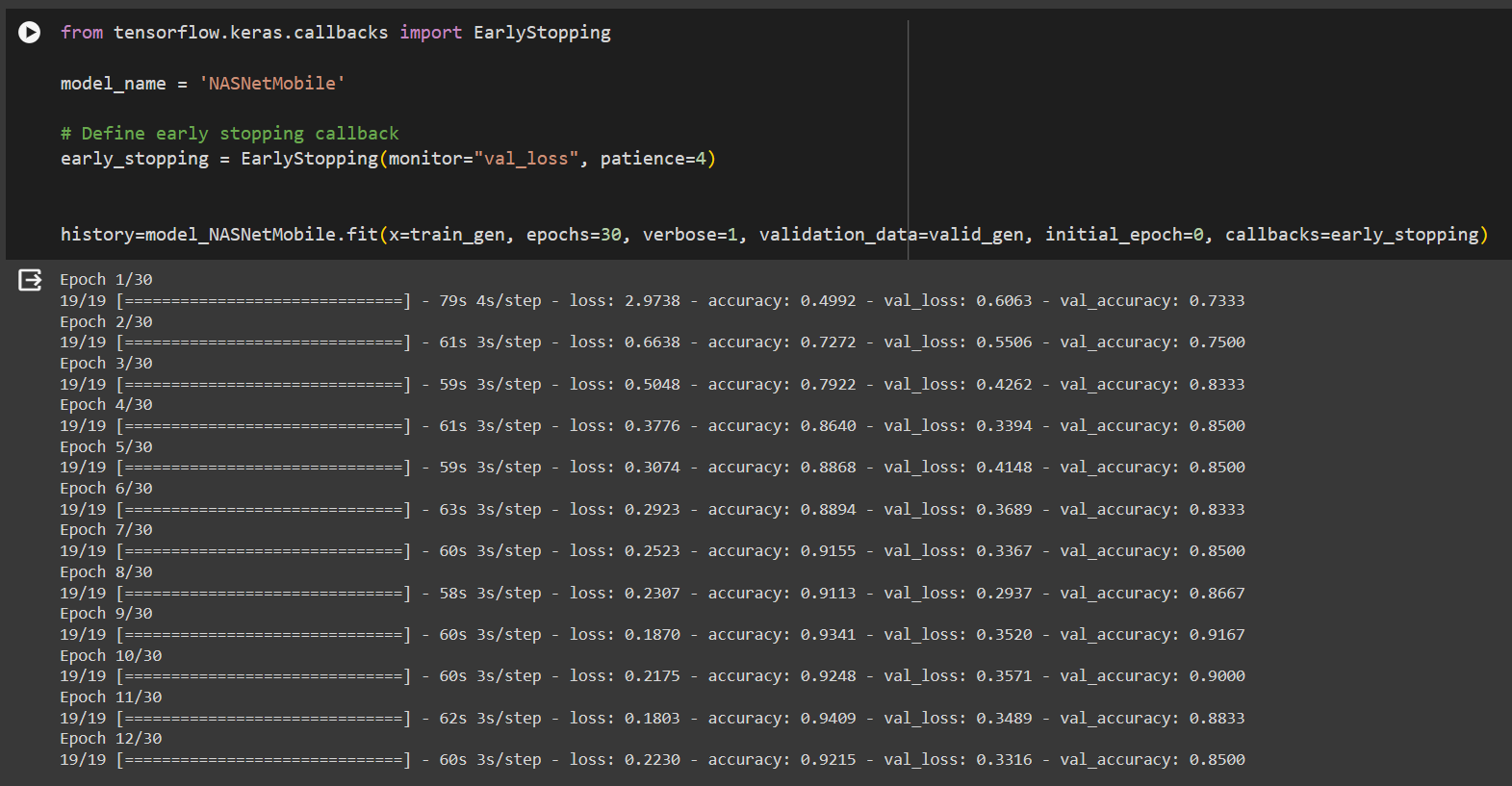


Gambar ... Kode Proses *Fully Connected Layer*

(Sumber Pribadi)

1. Akurasi Algoritma CNN dengan arsitektur NasNetMobile

Pada tahap ini model akan melalui tahapan pelatihan model dengan hasil nilai akurasi dari algoritma CNN. pada tahapan pelatihan menggunakan epoch 50 atau 50 kali iterasi, namun penulis juga menggunakan *callback* dengan jenis *early* *stopping* untuk menghentikan proses pelatihan jika sudah tidak ada peningkatan akurasi pada prosesnya. Berikut hasil proses akurasi algoritma CNN dapat dilihat pada gambar ... dibawah ini:



Gambar ... Akurasi Algoritma CNN

1. Evaluasi model

Evaluasi model pada penelitian ini menggunakan *confusion* *matrix* untuk mendapatkan nilai akurasi, presisi, *recal* dan hasil f1. Hasil dari proses ini akan dijadikan landasan apakah model perlu dilakukan pelatihan ulang atau tidak bergantung pada hasil yang didapat. Perhitungan dari confusion matrix menggunakan persamaan berikut :

*Accuracy* (....)

*Precision* (....)

*Recall* (....)

*F1-score* (....)

Keterangan :

*True Positif* (TP) : Data positif diprediksi positif

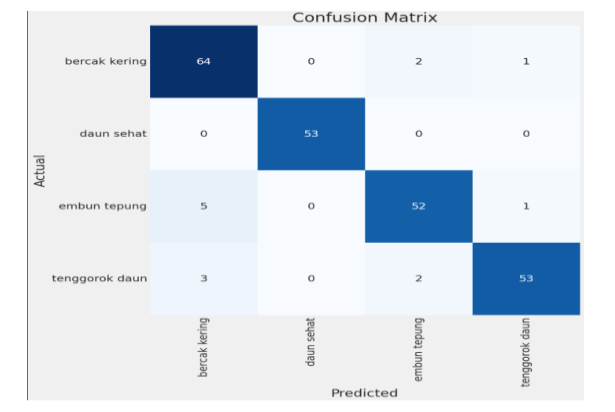
*True Negatif* (TN) : Data negatif diprediksi negatif

*False Positif* (FP) : Data negatif namun diprediksi sebagai data positif

*False Negatif* (FN) : Data positif namun diprediksi sebagai data negatif

Presisi merupakan akurasi data dengan hasil prediksi dari model yang sudah dibangun menggunakan algoritma CNN dan arsitektur NasNetMobile. *Recall* menggambarkan keberhasilan dari suatu model dalam menemukan kembali sebuah informasi. Sedangkan *f1-score* adalah perbandingan rata-rata dari nilai presisi dan *recall* yang dibobotkan.

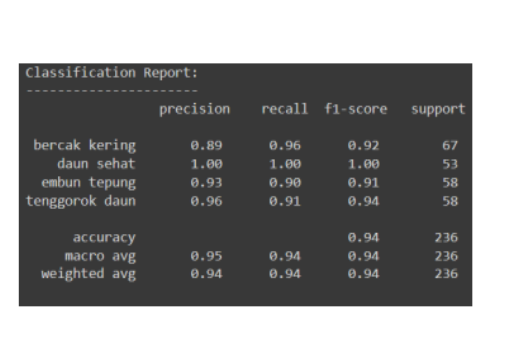
Pada model CNN dengan arsitektur dari *transfer learning* menggunakan NasNetMobile diberikan dataset sebanyak 236. Adapun hasilnya seperti gambar ... berikut.



Gambar ... Hasil Prediksi model CNN dengan NasNetMobile

(Sumber Pribadi)

Dari gambar .... diatas didapatkan hasil dari kelas bercak kering denga TP sebesar 64 gambar, FP sebesar 3 dan FN sebesar 8. Pada kelas daun sehat didapatkan hasil TP sebesar 53, FP sebesar 0 dan FN sebesar 0. Pada kelas embun tepung didapatkan hasil TP sebesar 52, FP sebesar 6 dan FN sebesar 4. Dan terakhir pada kelas tengorok daun didapatkan hasil TP sebesar 53, FP sebesar 5 dan FN sebesar 2. Jadi didapatkan perhitungan *confusion matrix* dari model CNN seperti pada gambar ... berikut :



Gambar ... Hasil perhitungan Confusion Matrix

(Sumber Pribadi)

Dapat disimpulkan hasil pengujian yang telah dilakukan berdasarkan sample data uji sebanyak 236 yaitu didapatkan hasil presentasi akurasi pengujian klasifikasi dalam mengindentifikasi objek sebesar 94.00%.

1. Deploy model

Proses deployment Machine Learning di aplikasi mobile pada penelitian ini menggunakan layanan tensorflow lite. Hasil dari model yang telah dibangun akan disimpan terlebih dahulu dengan format .h5, kemudian model tersebut akan di konverter menjadi file dengan format .tflite. File dengan .tflite akan di tambahkan pada aplikasi mobile dengan perantara library tflite\_flutter, sehingga aplikasi mobile yang dibangun akan dapat melakukan proses prediksi. Berikut proses konverter file .h5 menjadi file .tflite dapat dilihat pada gambar ... berikut :

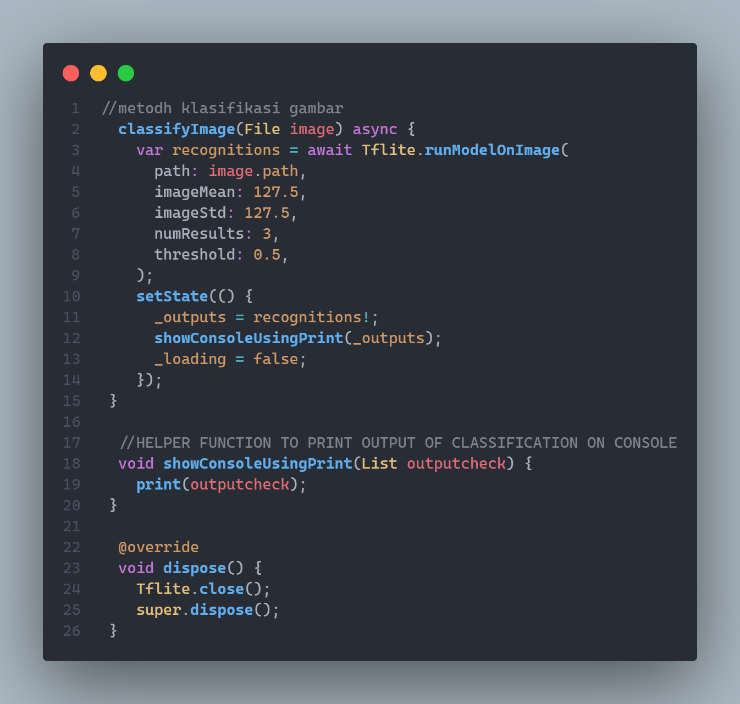


Gambar ... Konverter file model .h5 menjadi .tflite

(Sumber Pribadi)

Selanjutnya implementasi kode untuk menambahkan model pada aplikasi mobile dengan menggunakan library/pustaka tflite\_flutter dapat dilihat pada gambar ... dibawah ini :





Gambar .... Kode Menambahkan model pada Aplikasi Mobile

(Sumber Pribadi)

* + 1. Implementasi perangkat lunak

Sub bab ini menjabarkan implementasi dari perangkat lunak yang akan diisi oleh dokumentasi tangkapan layar dari aplikasi yang sudah dibangun dan dokumentasi kode pada setiap halaman aplikasi.

1. Halaman Splass Screen

Halaman Splass Screen akan ditampilkan saat aplikasi pertama kali dibuka bertujuan untuk memberikan informasi aplikasi dengan diberikan gambar logo aplikasi. Berikut tampilan splass screen pada gambar ... .

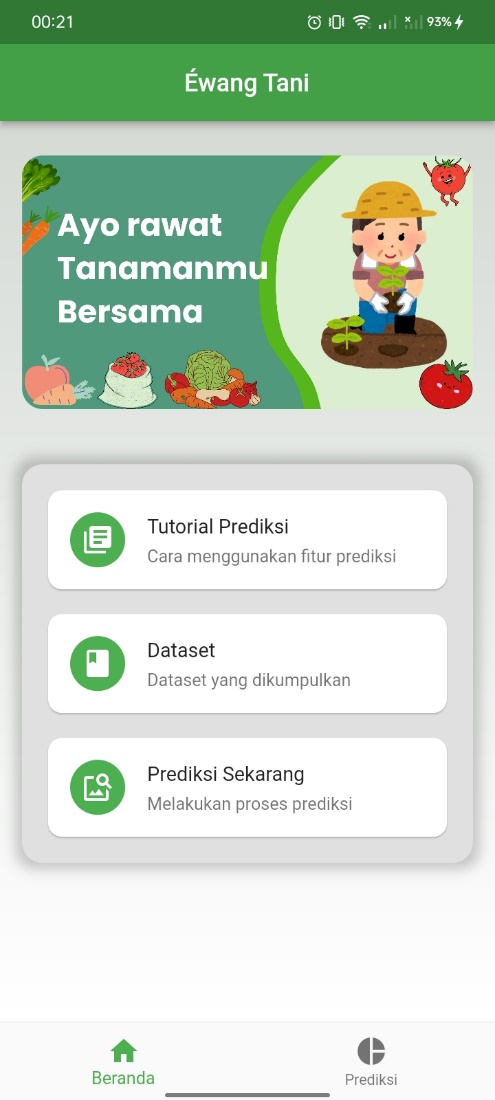


Gambar ... Halaman Splass Screen

(Sumber Pribadi)

1. Halaman Beranda

Halaman Beranda akan muncul ketika halaman sebelumnya selesai dimuat. Pada halaman ini pengguna akan diberikan informasi berupa tiga buah menu pada *body* halaman beranda, *app bar* dan *navigation* *bar*. Pada Menu terdiri dari Diagnosis Tips, Dataset dan Prediksi. Pada *app bar* terdapat nama aplikasi serta pada *navigation bar* terdapat navigasi ke halaman Beranda dan ke halaman Prediksi. Berikut tampilan halaman beranda dapat dilihat pada gambar ... .

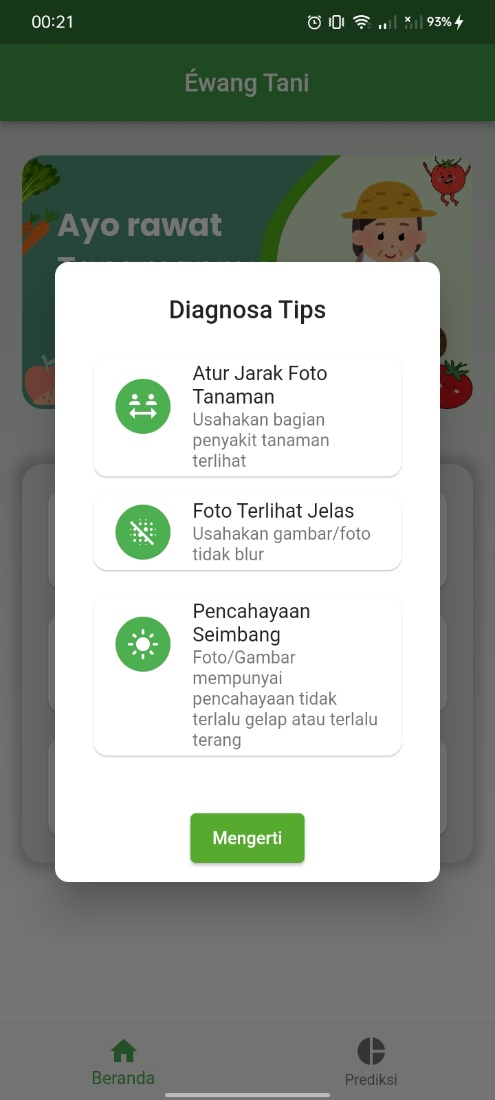


Gambar .... Halaman Beranda

(Sumber pribadi)

1. Halaman Beranda (Tips Diagnosa)

Pada menu Tips Diagnosa yang terdapat pada halaman beranda memuat informasi bagaimana pengguna menggunakan fitur diagnosa pada aplikasi ini. Informasi akan diberikan melalui *popup diaglog,* berikut tampilan menu Tips Diagnosa dapat dilihat pada gambar ... .

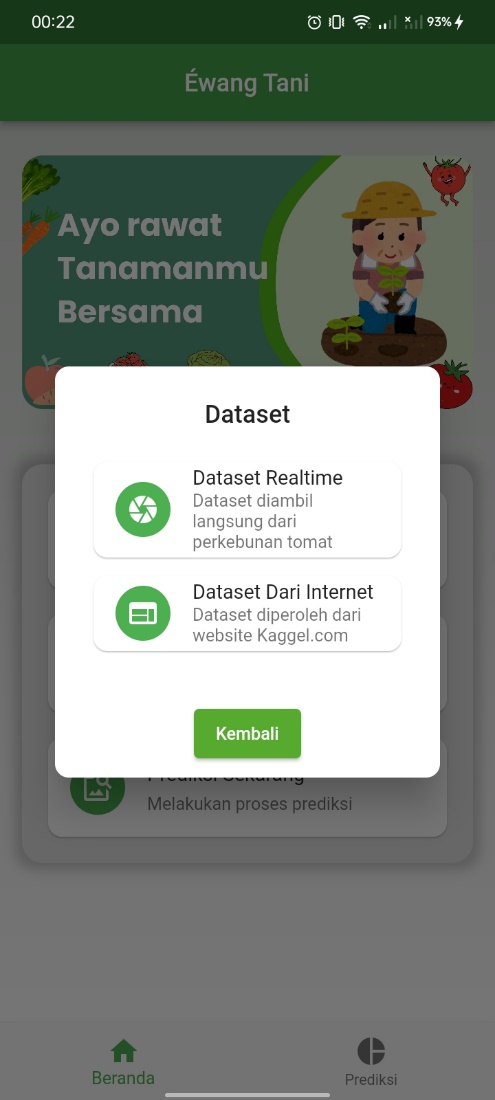


Gambar ... Menu Tips Diagnosa

(Sumber Pribadi)

1. Halaman Beranda (Dataset)

Selanjutnya pada halaman beranda terdapat menu Dataset yang bertujuan untuk memberikan informasi bagaimana aplikasi ini mendapatkan dataset sebagai bahan untuk melatih model machine learning. Dataset pada aplikasi ini didapatkan dari pengambilan gambar secara langsung di objek studi dan di perkuat dengan pengambilan gambar pada website penyedia dataset *kaggle.* Berikut tampilan menu Dataset dapat dilihat pada gambar ... .

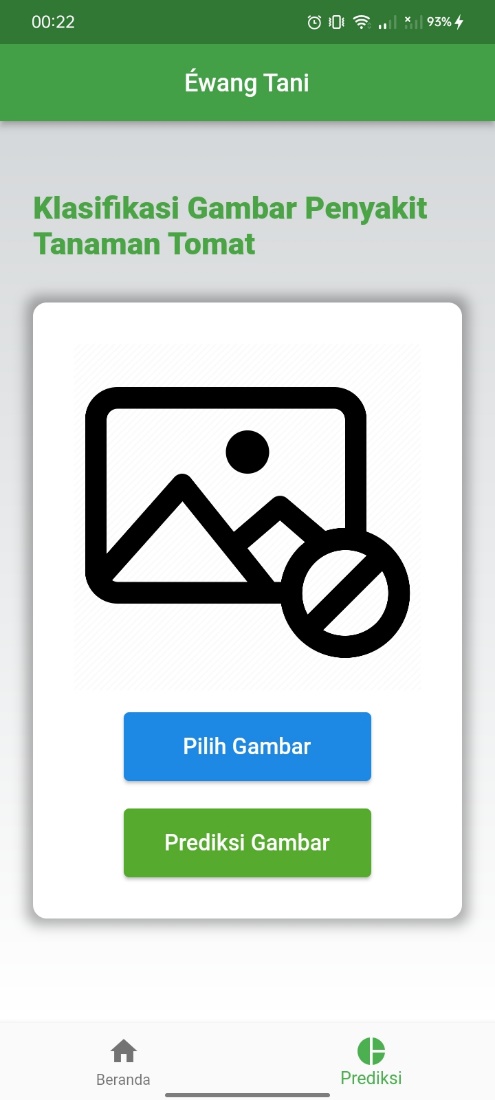


Gambar ... Menu Dataset

(Sumber Pribadi)

1. Halaman Prediksi

Pada halaman prediksi dapat diakses ketika pengguna menekan tombol *navigation bar* prediksi, di halaman ini pengguna harus memuat terlebih dahulu gambar daun tomat yang hendak diprediksi dengan cara pengguna menekan tombol pilih gambar. Setelah gambar dimuat pada halaman ini, selanjutnya pengguna dapat menekan tombol prediksi untuk melakukan proses diagnosa penyakit tanaman tomat. Berikut tampilan halaman Prediksi dapat dilihat pada gambar ... .



Gambar ... Halaman Prediksi

(Sumber Pribadi)

1. Halaman Prediksi (Metode Pengambilan Gambar dari Galeri)

Pada saat pengguna menekan tombol pilih gambar dengan opsi dari Galeri, aplikasi akan membuka halaman penyimpanan lokal pada *handphone* milik pengguna, selanjutnya pengguna dapat memilih foto yang akan diprediksi melalui aplikasi. Berikut tampilan opsi pengambilan gambar melalui galeri dapat dilihat pada gambar .... .

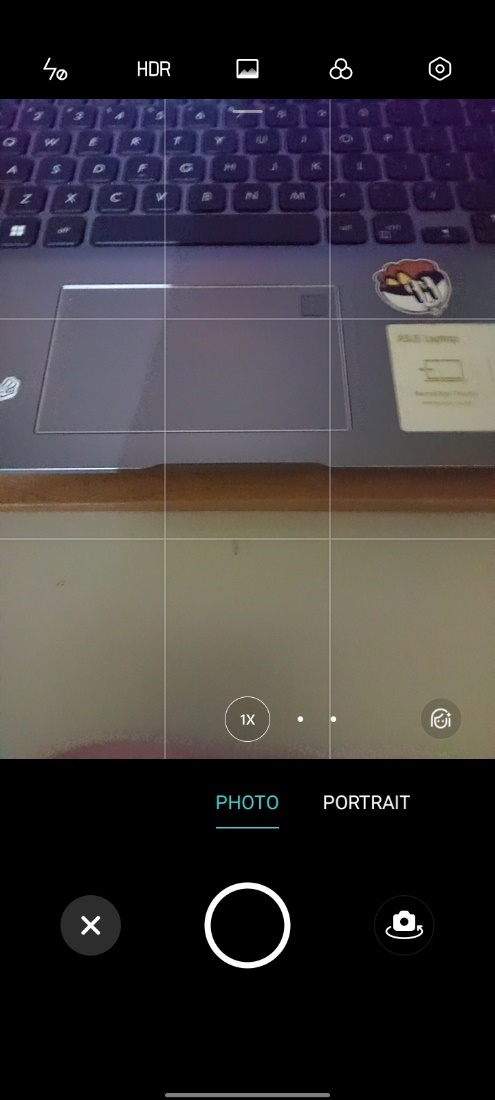


Gambar .... Tampilan Pengambilan Gambar dari Galeri

(Sumber Pribadi)

1. Halaman Prediksi (Metode Pengambilan Gambar dengan Kamera)

Pada saat pengguna menekan tombol pilih gambar dengan opsi dari Kamera, aplikasi akan membuka halaman kamera pada *handphone* milik pengguna, selanjutnya pengguna dapat mengambil foto daun tanaman tomat yang akan diprediksi penyakitnya melalui aplikasi. Berikut tampilan opsi pengambilan gambar melalui kamera dapat dilihat pada gambar .... .

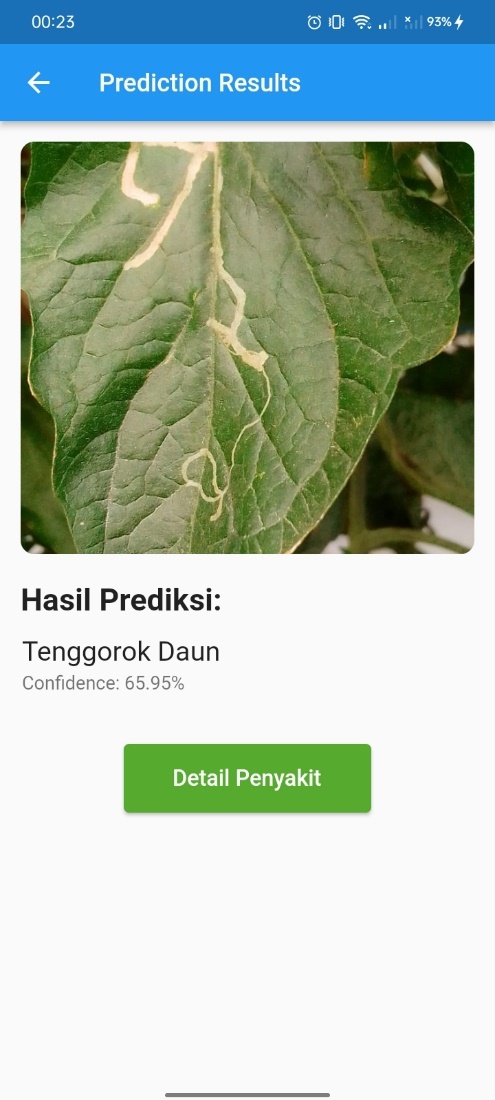


Gambar ... Tampilan Pengambilan Gambar dari Kamera

(Sumber Pribadi)

1. Halaman Hasil Prediksi

Pada halaman hasil prediksi pengguna akan mendapatkan informasi hasil diagnosa penyakit berdasarkan gambar yang dimuat pengguna sebelumnya. Informasi tersebut antara lain jenis penyakit dan jumlah presentasi prediksi penyakit. Jika ingin melihat lebih detail tentang hasil penyakit yang didapatkan, pengguna dapat menekan tombol Detail Penyakit yang ada di bawah hasil prediksi. Berikut tampilan halaman hasil prediksi dapat dilihat pada gambar ... .

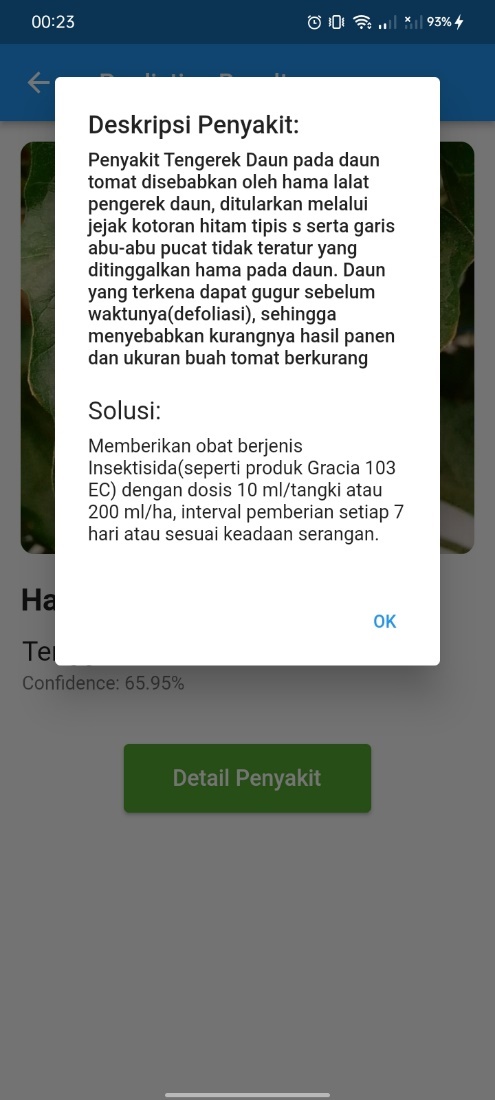


Gambar ... Halaman Hasil Prediksi

(Sumber Pribadi)

1. Halaman Hasil Prediksi (Detail Penyakit)

Pada halaman prediksi dengan aksi pengguna menekan tombol Detail Penyakit akan menampilkan deskripsi penyakit yang didapatkan dari hasil prediksi dan juga solusi penanganan penyakit. Informasi ini akan ditampilkan melalui *popup diaglog.* Berikut tampilan *popup diaglog* detail penyakit dapat dilihat pada gambar ... .



Gambar ... Tampilan *popup diaglog* Detail Penyakit

(Sumber Pribadi)

* 1. **Pengujian Sistem**

Setelah proses pengembangan aplikasi selesai. Selanjutnya sistem harus diuji sebelum digunakan, guna mengetahui tingkat kelayakan aplikasi terhadap proses mendeteksi penyakit tanaman tomat melalui daun. Proses pengujian akan melalui 2 tahapan, yaitu pengujian model machine learning, hal ini dilakukan untuk mengetahui tingkat akurasi model yang dibangun dalam mendeteksi masukan berupa data gambar daun tomat yang memiliki penyakit. Selanjutnya pengujian perangkat lunak, yang akan diuji melalui dua sudut pandang yaitu pengujian terhadap fungsionalitas perangkat lunak menggunakan teknik black-box testing dan pengujian terhadap penerimaan sistem oleh pengguna menggunakan metode beta testing.

* + 1. Pengujian *Alpha*

Pengujian *Alpha* dilakukan oleh penulis dengan menggunakan Teknik BlackBox Testing. Jenis pengujian *blackbox* testing yang dilakukan hanya menggunakan jenis *fungsional testing*. Berikut hasil dari pengujian yang telah dilakukan penulis sebagai pihak pengembang aplikasi.

1. Pengujian *splass screen*

Hasil pengujian tampilan splash screen dari aplikasi yang sudah dibangun dapat dilihat pada tabel ... berikut :

Tabel .... Hasil Pengujian Splash Screen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | *Scenario* Uji | Hasil yang Diharapkan | Hasil Pengujian | |
| Sesuai | Tidak Sesuai |
| **1** | Membuka aplikasi Ewang Tani | Aplikasi menampilkan halaman *splash screen* selama 3 detik dengan memuat logo aplikasi | v |  |

1. Pengujian Beranda

Hasil pengujian halaman beranda dari aplikasi yang sudah dibangun dapat dilihat pada tabel ... dibawah ini :

Tabel .... Hasil Pengujian Beranda

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | *Scenario* Uji | Hasil yang Diharapkan | Hasil Pengujian | |
| Sesuai | Tidak Sesuai |
| **1** | Masuk halaman Beranda | Aplikasi akan menampilkan foto banner yang memuat slogan dari aplikasi. Menampilkan 3 menu dengan bentuk *card widget* yaitu menu Tutorial Prediksi, Dataset dan Prediksi | v |  |
| **2** | Menekan tombol *card* menu Tutorial Prediksi | Aplikasi akan menampilkan *alertDialog* dengan isi informasi tips apa saja dalam pengambilan foto untuk di prediksi | v |  |
| **3** | Menekan tombol Dataset | Aplikasi akan menampilkan *alert dialog widget* dengan isi sebuah informasi dimana saja dataset didapatkan. | v |  |
| **4** | Menekan tombol Prediksi Sekarang | Aplikasi akan membawa pengguna menuju halaman prediksi | v |  |

1. Pengujian Prediksi

Hasil pengujian halaman predikisi dari aplikasi yang sudah di bangun dapat dilihat pada tabel ... berikut :

Tabel .... Hasil Pengujian Prediksi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | *Scenario* Uji | Hasil yang Diharapkan | Hasil Pengujian | |
| Sesuai | Tidak Sesuai |
| **1** | Masuk halaman prediksi | Aplikasi akan menampilkan halaman prediksi dengan dilengkapi icon gambar kosong dan terdapat 2 tombol, yaitu tombol Pilih Gambar serta tombol Prediksi Gambar. | v |  |
| **2** | Menekan Pilih gambar | Aplikasi akan memunculkan *alertDialog* pada bagian bawah tampilan dengan isian berupa pilihan untuk memilih metode pengambilan gambar. | v |  |
| **3** | Menekan pilihan Kamera dari tombol Pilih Gambar | Aplikasi akan menampilkan fungsi kamera yang diakses dari *handphone* pengguna untuk dapat mengambil gambar yang hendak akan diprediksi | v |  |
| **4** | Menekan pilihan Galeri dari tombol Pilih Gambar | Aplikasi akan membawa pengguna menuju penyimpanan galeri pada *handphone* pengguna | v |  |
| **5** | Ketika foto sudah ditentukan dan di upload | Foto yang sudah dipilih akan ditampilkan diatas tombol Pilih Gambar dan Prediksi sekarang, menggantikan icon gambar kosong. | v |  |
| **6** | Pengguna melakukan prediksi dengan menekan tombol Prediksi Gambar | Aplikasi akan memproses gambar yang akan dideteksi melalui model CNN. selanjutnya hasil prediksi akan diteruskan pada halaman Hasil Prediksi. | v |  |

1. Pengujian Hasil Prediksi

Hasil pengujian dari halaman Hasil Prediksi pada aplikasi yang telaht di bangun dapat dilihat pada tabel ... berikut :

Tabel .... Pengujian Hasil Prediksi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | *Scenario* Uji | Hasil yang Diharapkan | Hasil Pengujian | |
| Sesuai | Tidak Sesuai |
| **1** | Masuk halaman Hasil Prediksi | Setelah proses prediksi selesai selanjutnya hasil dari prediksi akan ditampilkan pada halaman ini. Dengan menampilkan gambar penyakit, jenis nama penyakit dan nilai persentase kepercayaan. Selain itu terdapat tombol Detail Penyakit | v |  |
| **2** | Menekan tombol Detail Penyakit | Aplikasi akan menampilkan *alertDialog* dengan isi informasi detail dari penyakit yang telah diprediksi dan dilengkapi dengan keterangan solusi dari penyakit. | v |  |

* + 1. Pengujian Beta

Pengujian Beda dilakukan dengan menyebar kuesioner ke petani tomat di Desa Tani Dompet Dhuafa maupun yang memiliki pengetahuan mengenai pertanian khususnya tanaman tomat. Media kuesioner yang dibuat memiliki 17 pertanyaan dengan dibagi menjadi lima instrumen pertanyaan, yakni segi kemudahan pengguna (*perceived ease of use*) terdapat 4 pertanyaan, manfaat penggunaan (*perceived usefulness*) terdapat 3 pertanyaan, sikap pengguna (*attitude toward using*) terdapat 4 pertanyaan, niat untuk menggunakan (*behavior intention to use*) terdapat 4 pertanyaan dan perilaku penggunaan sistem (*actual usage behavior*) terdapat 2 pertanyaan. Pada pengujian beta juga dilengkapi dengan uji validitas, uji realibilitas dan perhitungan skala likert.

1. Menentukan minimal responden dengan Rumus Slovin

Pengisian kuesioner ini dilakukan oleh petani tomat di Desa Tani maupun orang yang memiliki pengetahuan tentang pertanian khusus pada tanaman tomat. Sebelum penyebaran dilakukan secara daring, perlu dilakukan penarikan sampel terlebih dahulu menggunakan rumus Slovin, Persamaan Slovin dapat dituliskan sebagai berikut:

(....)

Keterangan :

n = Jumlah minimal sampel

N = Jumlah populasi

e = *Margin of error*

Dari persamaan slovin diatas, penulis menetapkan nilai dari jumlah populasi 60 dengan margin of error 10%. Margin of error adalah tingkat kesalahan untuk mendapatkan batas minimal sampel yang diperlukan. Maka hasil perhitungan berdasarkan rumus diatas adalah sebagai berikut :

Berdasarkan perhitungan diatas maka didapatkan sampel yang diujikan dari jumlah populasi 60 dengan tingkat kesalahan sebesar 10% adalah sebanyak 38 responden.

1. Uji Validitas
2. Uji Realibitas
3. Skala Likert
   1. ***Realese* Sistem**

Setelat dilakukannya semua tahapan dari metode Extreme Programming, serta telah dilakukan testing Alpha dan Beta. Maka telah dipastikan semua fitur sudah dapat dipakai dan sesuai dengan kebutuhan pengguna. Selanjutnya sistem telah di rilis dengan link download aplikasi yang dapat di akses oleh pengguna adalah google.play/../.. .

# BAB V KESIMPULAN DAN SARAN

# DAFTAR PUSTAKA

Penulis Tuga Mauritsius dan Faisal Binsar. Website Binus University bagian *Knowledge dan Reserch*. 2023. [online]. Tersedia di <https://mmsi.binus.ac.id/2020/09/18/cross-industry-standard-process-for-data-mining-crisp-dm/> [diakses pada 2023]

Badan Pusat Statistika (BPS), 2023 [Online]. Tersedia pada: <https://www.bps.go.id/indicator/55/61/5/produksi-tanaman-sayuran.html> [Diakses pada Agustus 2023]

Humaira Aliya, 2020 [online]. Tersedia pada: <https://glints.com/id/lowongan/perbedaan-alpha-dan-beta-testing/> [Diakses pada 04 Oktober 2023]

Sofia Nadhifa, 2018, "Convolutional Neural Network", tersedia di https://medium.com/@nadhifasofia/1-convolutional-neural-network-convolutional-neural-network-merupakan-salah-satu-metode-machine-28189e17335b [diakses pada 10 Okt 2023]

Keras, "Introducing Keras Core: Keras for TensorFlow, JAX and PyTorch", tersedia di https://keras.io/keras\_core/announcement/ [diakses pada 24 Oktober 2023]

Noor Wahyuni, 2014, “In-Depth Interview (Wawancara Mendalam)”, Binus University Quality Management Center [online], tersedia di https://qmc.binus.ac.id/2014/10/28/in-depth-interview-wawancara-mendalam/ [diakses pada 30 Oktober 23]

Sharon Patricia Siahaan, Dkk. (2009). “Pemantauan Dan Pengontrolan Sistem Pemupukan Budidaya Tanaman Tomat Menggunakan *Convolutional Neural Network*”, *eProceedings of Engineering*, vol. 7. no. 3. hal. 8647-8661.

Kahlil Muchtar Dkk. (2015). “Pendeteksian *Septoria* pada Tanaman Tomat dengan Metode *Deep Learning* berbasis *Raspberry Pi*”, Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi), vol. 5. No. 1. Hal 107-113

Resti Fajarfika. Dkk. (2019). “Deteksi Molekuler Penyebab Penyakit Kuning (*Tomato Chlorosis Virus* dan *tomato Infectious Chlorosis Virus*) Pada Tanaman Tomat”., Jurnal Perlindungan Tanaman Indonesia, vol. 25. no. 2.

Akbar Hidayatuloh. (2018). “Identifikasi Penyakit Tanaman Tomat Melalui Daun Dengan Metode *Convolutional Neural Network* Arsitektur *Squeezenet*”, Skripsi, Universitas Pendidikan Indonesia

Bambang Cahyono. (2000). Tomat. Budidaya dan Analisis Usaha Tani. Kanisius, Yogyakarta.

Pracaya. I. (1994). Bertanam Tomat. Kanisius, Yogyakarta.

Surtinah. (2007). “Kajian Tentang Hubungan Pertumbuhan Vegetatif Dengan Produksi Tanaman Tomat (Lycopersicum esculentum, Mill ) PS”. Agronomi, Staf Pengajar Fakultas Pertanian Universitas Lancang Kuning , Vol. 4 No 1.

Rosa dan Salahuddin, M., “Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek”. Bandung. Informatika Bandung. 2018.

Reza Robby dkk. (2016), “Perancangan dan Implementasi Penggunaan *Cloud* *Server* PAAS (Platform As A Service) Sebagai Penyimpanan Data Pengunjung dan Rekaman Video Pengunjung Pada *Building Security System* Berbasis *Embedded*”, e-Proceeding of Engineering, vol. 3, no. 1.

Tirtana E dan Setiawan Budi. (2017), “Aplikasi Pengendali Kamera DSLR *Nirkabel* *Tipe Low End* Berbasis Android”, ULTIMA Computing, vol. 9, no. 1.

Ash-Shidiq Usamah, dkk. (2017). “Perancangan Aplikasi *Smartphone* Android Sebagai Pengendali Jarak Jauh Pada Sistem *Smart Baby Monitoring*”. e-Proceeding of Engineering, vol. 4, no. 2.

Santoso Aditya dan Ariyanto Gunawan. (2018). “Implementasi Deep Learning Berbasis Keras untuk Pengenalan Wajah”, Skripsi, Universitas Muhammadiyah Surakarta.

N Cahya Fadhila, dkk. (2019). “Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik *Equivalence Partitions*”. Jurnal Informatika Universitas Pamulang, vol. 4, no. 4, hal 125-130.

Hanifah Umi, dkk. (2016). “Penggunaan Metode Black Box Testing pada Pengujian Sistem Informasi Surat Keluar Masuk”. SCAN Jurnal Teknologi Informasi dan Komunikasi, vol. 11, no. 2, hal 33-40.

Masripah Siti dan Ramayanti Linda. (2020). “Penerapan Pengujian Alpha dan Beta pada Aplikasi Penerimaan Siswa Baru”. Jurnal SWABUMI, vol. 8, no. 1, hal 100-105.

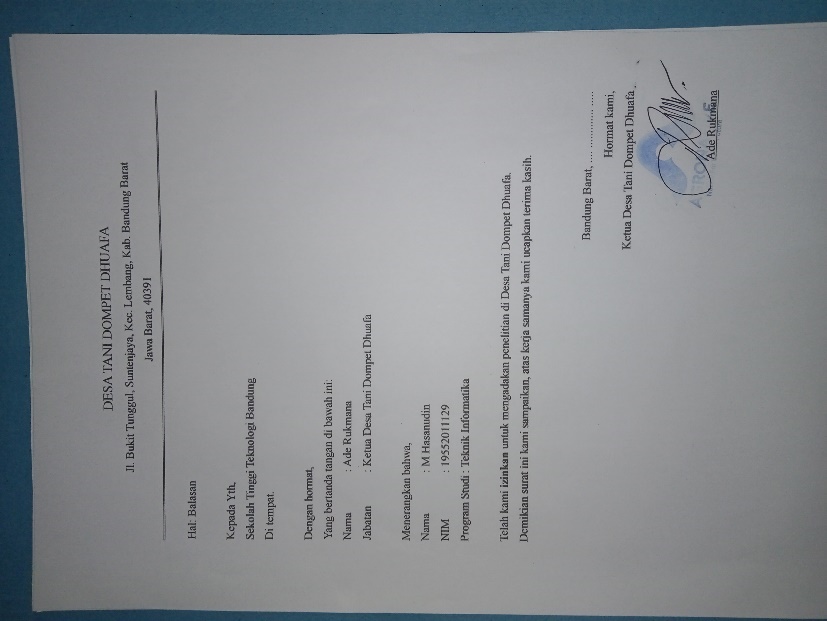
Setyawan, Ari Ryan dan Walter F. Atapukan. (2018). “Pengukuran Usability Website E-Commerce Sambal Nyoss Menggunakan Metode Skala Likert”. Jurnal Compiler, vol. 7, no. 1, hal 54-61.

Sianipar, Rismon Hasiholan. 2013. “Pemograman MATLAB Dalam Contoh dan Penerapan”. Penerbit INFORMATIKA.

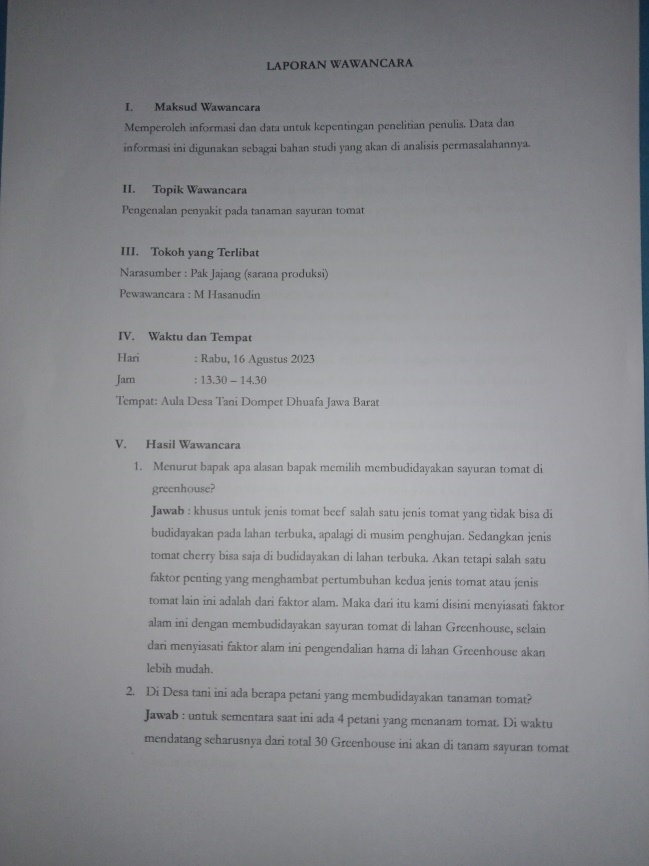
Mayasari, Ferinica dkk. 2011. “Anteseden dan Konsekuen Sikap Nasabah dalam Menggunakan Internet *Banking* dengan Menggunakan Kerangka *Technology Acceptance Model* (TAM), Survei pada Pengguna KlikBCA”. SEMATIK, vol. 1, no. 1.

# LAMPIRAN

1. **Surat Balasan Penelitian**

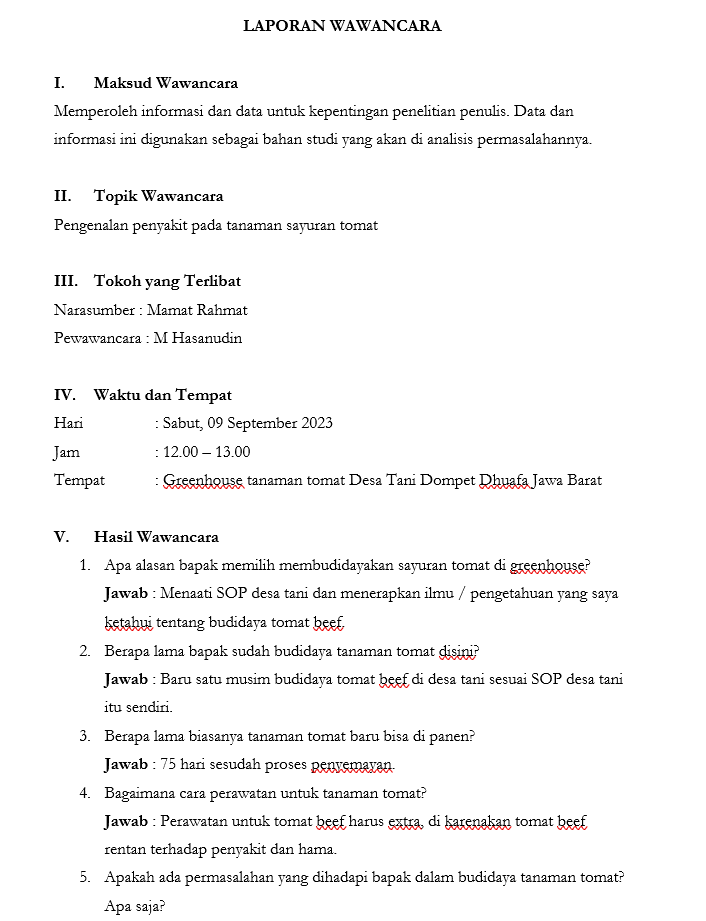
****

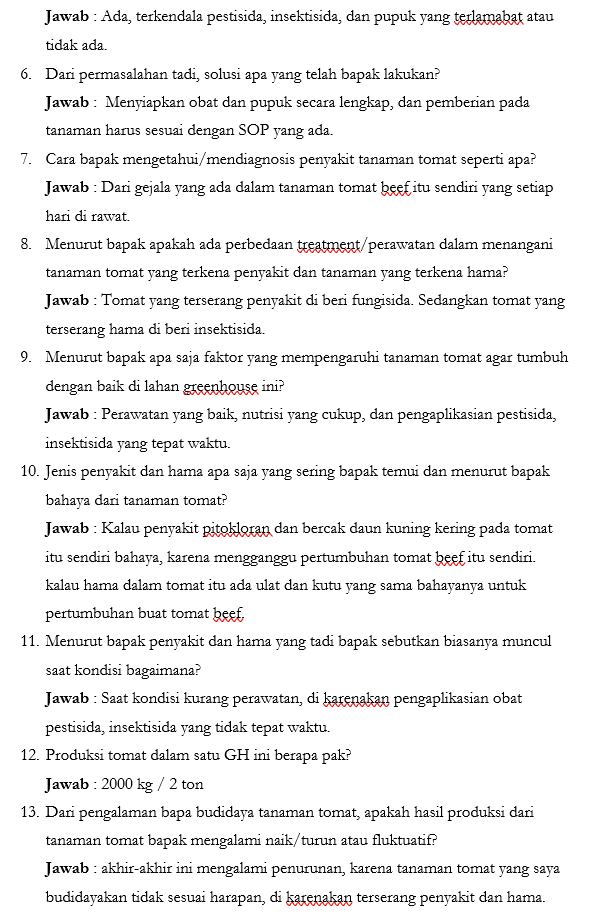
1. **Laporan Wawancara I**

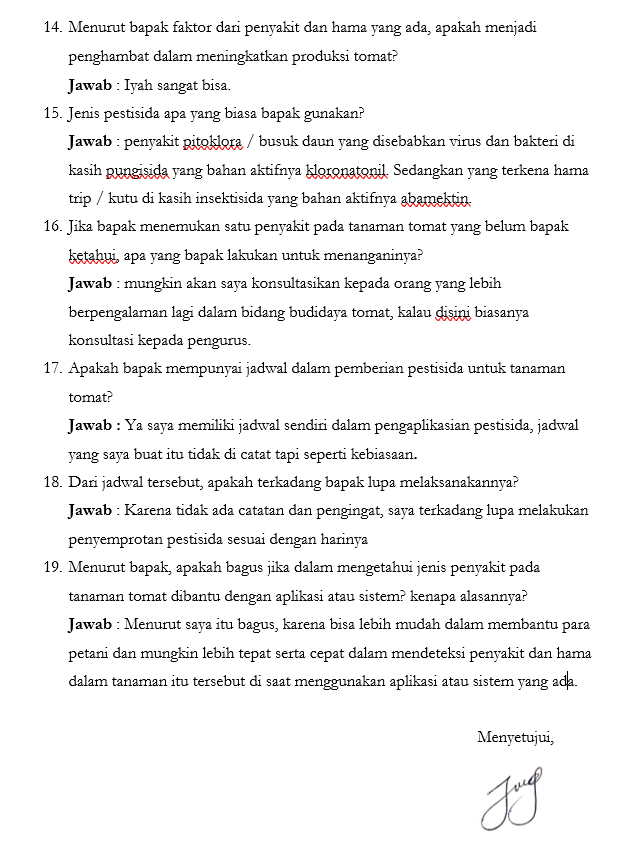
****

****

1. **Laporan Wawancara II**







1. **Dokumentasi**

|  |
| --- |
|  |
| Proses Perizinan penelitian dengan Pak Ade |

|  |
| --- |
|  |
| Wawancara dengan pak Jajang |

|  |
| --- |
|  |
| Keterangan: Lahan Pertanian Desa Tani |