

**PREDIKSI PERTUMBUHAN TANAMAN TOMAT BERDASARKAN
PENGAMBILAN CITRA OLEH ROBOT MENGGUNAKAN
*GRADIENT BOOSTING***

SKRIPSI

NURUL ANDINI

171402046



**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2024**

PREDIKSI PERTUMBUHAN TANAMAN TOMAT BERDASARKAN
PENGAMBILAN CITRA OLEH ROBOT MENGGUNAKAN
GRADIENT BOOSTING

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah
Sarjana Teknologi Informasi

NURUL ANDINI

171402046



PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2024

PERSETUJUAN

Judul : PREDIKSI PERTUMBUHAN TANAMAN TOMAT
BERDASARKAN PENGAMBILAN CITRA OLEH
ROBOT MENGGUNAKAN *GRADIENT BOOSTING*

Kategori : Skripsi

Nama Mahasiswa : Nurul Andini

Nomor Induk Mahasiswa : 171402046

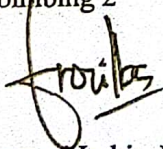
Program Studi : Sarjana (S-1) Teknologi Informasi

Fakultas : Ilmu Komputer dan Teknologi Informasi
Universitas Sumatera Utara

Medan, 04 Juli 2024

Komisi Pembimbing:

Pembimbing 2



Fahrurrozi Lubis, B.IT., M.Sc.IT
NIP. 198610122018052001

Pembimbing 1

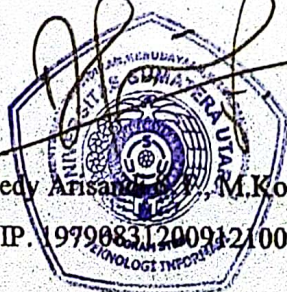


Baihaqi Siregar, S.Si., MT.
NIP. 197901082012121002

Diketahui/disetujui oleh

Program Studi S-1 Teknologi Informasi

Ketua,


Dedy Arisandi, M.Kom.
NIP. 197908312009121002

PERNYATAAN

**PREDIKSI PERTUMBUHAN TANAMAN TOMAT BERDASARKAN
PENGAMBILAN CITRA OLEH ROBOT MENGGUNAKAN
*GRADIENT BOOSTING***

SKRIPSI

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing – masing telah disebutkan sumbernya.

Medan, 04 Juli 2024

Nurul Andini

171402046

UCAPAN TERIMA KASIH

Puji dan syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat dan karunia-Nya kepada penulis, sehingga dapat menyelesaikan penyusunan skripsi yang berjudul “Prediksi Pertumbuhan Tanaman Tomat Berdasarkan Pengambilan Citra Oleh Robot Menggunakan *Gradient Boosting*” yang mana menjadi salah satu syarat untuk mendapatkan gelar Sarjana Komputer di Program Studi Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.

Pada kesempatan ini, penulis ingin menyampaikan ucapan terimakasih kepada semua pihak yang selama ini telah membantu dan mendukung penulis dalam melakukan penelitian ini, antara lain :

1. Orang tua penulis yaitu Bapak M. Adnan dan Ibu Rifmirika serta saudara penulis Kak Dara, Kak Amel, Kak Dwi dan Raja yang telah memberikan dukungan, kasih sayang serta doa kepada penulis.
2. Bapak Baihaqi Siregar S.Si, M.T., sebagai Dosen Pembimbing I dan Bapak Fahrurrozi Lubis, B.IT., M.Sc.IT, sebagai Dosen Pembimbing II yang dengan sabar dan Ikhlas dalam memberikan arahan, bimbingan, dukungan, saran serta motivasi kepada penulis.
3. Ibu Dr. Maya Silvi Lydia B.Sc., M.Sc., sebagai Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
4. Bapak Dedy Arisandi S.T., M.Kom., sebagai Ketua Program Studi S-1 Teknologi Informasi Universitas Sumatera Utara.
5. Seluruh dosen serta Staff Pegawai Program Studi S-1 Teknologi Informasi Universitas Sumatera Utara.
6. Sahabat seperjuangan penulis yaitu Nadia Siti Namira, Allia Rania, Tria Riskiani, Lisa Ayuning Tias, Ali Hidayat, Jackie Chandra, M. Bayhaqi Daulay, Fahmi Rizal, M. Ihsan Maulana, M. Ulwan Azmi, M. Wahyu Pratama dan Fakhri Rizha Ananda yang telah memberikan dukungan, doa serta motivasi kepada penulis.

7. Keluarga besar penulis yang memberikan semangat, motivasi, doa serta dukungan kepada penulis.
8. Seluruh pihak yang tidak dapat disebutkan semua namanya, yang berperan dalam mendukung dan memberikan motivasi kepada penulis dalam menyelesaikan skripsi ini.

Penulis mengakui bahwa dalam penyusunan skripsi ini, terdapat beberapa kekurangan dan kesalahan yang mungkin ada. Oleh karena itu, setiap kritik dan saran sangat dihargai, dan diharapkan dapat menjadi bekal berharga untuk perbaikan dan pengembangan di masa depan. Terima kasih atas semua dukungan yang telah diberikan, semoga semuanya akan memberikan manfaat yang berlipat ganda bagi penulis dan orang-orang yang telah mendukung penulisan skripsi ini.

Medan, 04 Juli 2024

Nurul Andini

171402046

ABSTRAK

Pada metode pertanian konvensional, petani harus terlibat secara langsung dalam pemantauan pertumbuhan tanaman di lapangan. Apabila terjadi kelalaian dalam proses pemantauan, hal ini dapat menghambat pertumbuhan tanaman itu sendiri. Kadar kelembapan dan tingkat keasaman (pH) tanah adalah dua faktor krusial yang dapat memengaruhi pertumbuhan tanaman. Salah satu cara untuk mengatasi masalah tersebut yaitu, dengan menggunakan sistem prediksi pertumbuhan tanaman. Penelitian ini bertujuan untuk memprediksi pertumbuhan luas citra tanaman tomat dan luas citra buah tomat berdasarkan kelembapan tanah dan pH tanah. Pada prosesnya, penulis dibantu oleh robot pemantau pertumbuhan tanaman yang dilengkapi dengan kamera dan sensor untuk mengambil citra tanaman dan buah tomat secara berkala, serta data sensor pH tanah dan kelembapan tanah. Tahap pertama yang akan dilalui adalah image *pre-processing*, yang meliputi *cropping*, *gamma correction* dan *hsv color space*. Tahap berikutnya adalah proses ekstraksi fitur menggunakan length features extraction untuk mengukur luas citra tanaman dan luas citra buah. Hasil analisis citra yang telah diperoleh akan digabungkan dengan dataset sensor kelembapan dan pH tanah, yang nantinya digunakan sebagai masukan untuk model prediksi berbasis *Gradient Boosting*. Model ini memungkinkan prediksi pertumbuhan tanaman selama masa pertumbuhan dan dapat dipantau melalui aplikasi desktop. Prediksi pertumbuhan tanaman pada penelitian ini memiliki akurasi sebagai berikut: (a) kecocokan model untuk luas citra tanaman tomat sebesar 88,04%, dan (b) kecocokan model untuk luas citra buah tomat sebesar 92%.

Kata Kunci : *artificial intelligence*, *gradient boosting*, *internet of think*, prediksi, luas tanaman, robot, kelembapan tanah, pH tanah.

TOMATO PLANTS GROWTH PREDICTION BASED ON IMAGES BY ROBOTS USING GRADIENT BOOSTING

ABSTRACT

In conventional farming methods, farmers must be directly involved in monitoring the growth of plants in the field. Any negligence in the monitoring process can hinder the growth of the plants themselves. Soil moisture levels and acidity (pH) are two crucial factors that can affect plant growth. One way to address these issues is by using a plant growth prediction system. This study aims to predict the growth of tomato plant images and tomato fruit images based on soil moisture and soil pH. In the process, the author is assisted by a plant growth monitoring robot equipped with a camera and sensors to periodically capture images of the plants and fruits, as well as soil pH and moisture sensor data. The first stage involves image pre-processing, which includes cropping, gamma correction, and HSV color space. The next stage is feature extraction using length features extraction to measure the area of the plant images and fruit images. The obtained image analysis results will be combined with the soil moisture and pH sensor dataset, which will then be used as input for a Gradient Boosting-based prediction model. This model allows for the prediction of plant growth during the growth period and can be monitored through a desktop application. The accuracy of plant growth prediction in this study is as follows: (a) the model's accuracy for the area of tomato plant images is 88.04%, and (b) the model's accuracy for the area of tomato fruit images is 92%.

Keywords: artificial intelligence, gradient boosting, internet of think, prediction, plant area, robot, soil moisture, soil pH.

DAFTAR ISI

PERSETUJUAN	ii
PERNYATAAN	iii
UCAPAN TERIMA KASIH	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
BAB 1 PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Batasan Masalah	2
1.5 Manfaat Penelitian	3
1.6 Metodologi Penelitian	3
1.7 Sistematika Penulisan	3
BAB 2 LANDASAN TEORI	
2.1 Robot Pertanian	5
2.2 Kecerdasan Buatan (<i>Artificial Intelligence</i>)	5
2.3 Kelembapan Tanah	6
2.4 pH Tanah	6
2.5 Deep Learning	7
2.6 <i>Gradient Boosting</i>	7
2.7 Penelitian Terdahulu	10

BAB 3 ANALISIS DAN PERANCANGAN

3.1	Data yang digunakan	14
3.2	Analisis Sistem	14
3.2.1	<i>Arsitektur Umum</i>	14
3.2.2	<i>Data Collection</i>	15
3.2.3	<i>Data Pre-processing</i>	16
3.2.3.1	<i>Image cropping</i>	16
3.2.3.2	<i>Gamma Correction</i>	17
3.2.3.3	<i>HSV color space</i>	18
3.2.4	<i>Feature Extraction</i>	19
3.2.5	<i>Merging Dataset</i>	20
3.2.6	<i>Gradient Boosting</i>	20
3.2.6.1	<i>Inisialisasi Model</i>	20
3.2.6.2	<i>Residual Error</i>	21
3.2.6.3	<i>Weak Learner</i>	21
3.2.6.4	<i>Update Model</i>	22
3.2.7	<i>Learned Model</i>	23
3.2.8	<i>Prediciton</i>	24
3.2.9	<i>Output</i>	24
3.3	Perancangan Sistem	24
3.3.1	Perancangan Halaman Beranda	25
3.3.2	Perancangan Halaman <i>Import Data</i>	26
3.3.3	Perancangan Halaman <i>Pre-Processing Data</i>	27
3.3.4	Perancangan Halaman <i>Train Model</i>	27
3.3.5	Perancangan Halaman Prediksi dan Hasil Prediksi	28

BAB 4 IMPLEMENTASI DAN PENGUJIAN

4.1	Spesifikasi Perangkat Keras dan Perangkat Lunak	31
4.2	Impelentasi Sistem	31
4.2.1	Impelentasi <i>Data Collection</i>	31

4.2.2	Implementasi <i>Data Pre-processing</i>	32
4.2.2.1	Implementasi <i>Image Cropping</i>	32
4.2.2.2	Implementasi <i>Gamma Correction</i>	33
4.2.2.3	Implementasi <i>HSV Color Space</i>	34
4.2.3	Implementasi <i>Feattrue Extraction</i>	36
4.2.4	Implementasi <i>Merging Dataset</i>	37
4.2.5	Implementasi <i>Gradient Boosting</i>	38
4.2.5.1	Implementasi Inisialisasi Model	38
4.2.5.2	Implementasi <i>Residual Error</i>	39
4.2.5.3	<i>Weak Learner</i>	40
4.2.5.4	<i>Update Model</i>	41
4.2.6	Implementasi Perancangan Sistem	42
4.2.6.1	Implementasi Halaman Beranda	42
4.2.6.2	Implementasi Halaman <i>Import Data</i>	43
4.2.6.3	Implementasi Halaman <i>Pre-Processing Data</i>	44
4.2.6.4	Implementasi Halaman <i>Train Model</i>	45
4.2.6.5	Implementasi Halaman Prediksi dan Hasil Prediksi	46
4.3	Pengujian Sistem	47
BAB 5 KESIMPULAN DAN SARAN		
5.1	Kesimpulan	56
5.2	Saran	56
DAFTAR PUSTAKA		57

DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu	11
Tabel 3. 1 Perhitungan <i>Residual Error</i>	21
Tabel 4. 1 Implementasi <i>Length Feature Extraction</i> Pada Citra	37
Tabel 4. 2 Hasil Merging Dataset	38
Tabel 4. 3 Implementasi Inisialisasi Model	39
Tabel 4. 4 Implementasi <i>Residual Error</i>	40
Tabel 4. 5 Implementasi <i>Update Model</i>	41
Tabel 4. 6 Hasil Pengujian Data Luas Tanaman	48
Tabel 4. 7 Hasil Validasi Model	53

DAFTAR GAMBAR

Gambar 2. 1 Agribot / Robot Pertanian	5
Gambar 2. 2 Arsitektur <i>Gradient Boosting</i>	8
Gambar 3. 1 Arsitektur Umum	15
Gambar 3. 2 Contoh Gambar Tanaman dan Buah Tomat	16
Gambar 3. 3 Sebelum dan Sesudah <i>Cropping</i>	17
Gambar 3. 4 Sebelum dan Sesudah <i>Gamma Correction</i>	18
Gambar 3. 5 <i>HSV Color Space</i>	19
Gambar 3. 6 Perancangan Halaman Beranda	25
Gambar 3. 7 Perancangan Halaman <i>Import</i>	26
Gambar 3. 8 Perancangan Halaman <i>Pre-processing</i> Data	27
Gambar 3. 9 Perancangan Halaman Data <i>Train</i>	28
Gambar 3. 10 Perancangan Halaman Prediksi	29
Gambar 3. 11 Perancangan Halaman Hasil Prediksi	30
Gambar 4. 1 Implementasi Pengambilan Citra Tanaman Tomat	32
Gambar 4. 2 Implementasi Sebelum dan Sesudah <i>Cropping</i>	33
Gambar 4. 3 Implementasi <i>Gamma Correction</i>	34
Gambar 4. 4 Implementasi <i>HSV Color Space</i>	36
Gambar 4. 5 Contoh Proses Pembangunan <i>Weak Learner</i>	40
Gambar 4. 6 Implementasi Halaman Beranda	42
Gambar 4. 7 Implementasi Halaman <i>Import</i> Data	43
Gambar 4. 8 Implementasi Halaman <i>Pre-processing</i>	44
Gambar 4. 9 Implementasi Halaman <i>Data Train / Train Model</i>	45
Gambar 4. 10 Implementasi Halaman Prediksi	46
Gambar 4. 11 Implementasi Halaman Hasil Prediksi	47
Gambar 4. 12 <i>Featur Importance</i> pada besar buah dan luas tanaman	54
Gambar 4. 13 Gambar Tanaman Lewat <i>Frame</i> Kamera	55

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pemantauan pertumbuhan tanaman merupakan kegiatan yang sangat penting bagi petani untuk keberlangsungan hidup tanaman tersebut. Penyemaian dan penanaman yang presisi meningkatkan ukuran rata-rata tanaman dan keseragaman kematangan tanaman. Penggunaan teknik penyiraman dan pemupukan presisi yang melibatkan pemberian air dan nutrisi tanaman hanya pada waktu dan posisi optimal dapat mengurangi jumlah pupuk dan air yang diperlukan, sehingga lebih efisien dan menghasilkan tanaman yang lebih baik (Tremblay *et al.*, 2011). Tidak banyak pula tanaman yang mengalami gagal panen dikarenakan kurangnya pemantauan tanaman yang dipengaruhi berbagai aspek, terutama dari faktor perubahan lingkungan.

Perkembangan teknologi semakin pesat seiring berjalannya waktu terhadap semua bidang kehidupan, terutama pada bidang pertanian. Perkembangan teknologi yang menjadi pedoman di bidang pertanian saat ini adalah revolusi industri 4.0. Revolusi industri 4.0 adalah transformasi komprehensif dari keseluruhan aspek produksi di industri melalui penggabungan teknologi digital dan internet dengan industri konvensional (Merkel, 2014). Salah satu contoh dari perkembangan revolusi industri 4.0 pada bidang pertanian, yaitu Robot Pertanian. Robot Pertanian merupakan sebuah robot perkebunan yang bekerja secara otomatis dengan bantuan kecerdasan buatan sesuai dengan kondisi tanaman. Kegiatan produktivitas pertanian konvensional, dimana budidaya dan pengelolaan tanaman dilakukan secara manual oleh petani, dapat ditingkatkan secara signifikan dengan menggunakan mesin yang cerdas (Xia *et al.*, 2015).

Penerapan *artificial intelligence* atau kecerdasan buatan sangat membantu dalam pemantauan keadaan suatu hal berdasarkan data yang sudah dilatih sebelumnya. Kecerdasan buatan atau *artificial intelligence* merupakan bidang ilmu komputer yang berfokus pada pengembangan sistem yang dapat bekerja secara otomatisasi dan

menampilkan perilaku cerdas (Kristianto, 2004). Dengan otomatisasi, mesin pertanian dapat beroperasi dengan lebih efisien, dapat diandalkan dan memiliki tingkat presisi yang lebih tinggi yang pada gilirannya dapat mengurangi keterlibatan pekerja manusia dalam berbagai tugas pertanian. Hal ini dapat menyebabkan peningkatan keseluruhan produktivitas dan efisiensi operasional dalam sektor pertanian (Schueller *et al.*, 2006).

Beberapa penelitian yang berkaitan dengan prediksi pertumbuhan tanaman yang dilakukan dengan beberapa metode, yaitu penelitian oleh Pohan *et al.* (2020) dengan judul “ *Backpropagation Artificial Neural Network for Prediction Plant Seedling Growth* ” membangun sebuah sistem berbasis dekstop untuk memprediksi pertumbuhan bibit tanaman berdasarkan data intensitas cahaya, suhu, kelembapan udara, kelembapan tanah, jumlah daun, lebar tanaman dan panjang batang dengan menggunakan metode *Backpropagation Artificial Neural Network*. Adapun hasil dari pengaplikasian algoritma ini untuk memprediksi pertumbuhan bibit tanaman di rumah kaca menunjukkan hasil yang baik, dengan Mean Squared Error (MSE) dari iterasi pertama sebesar 0.0112, waktu komputasi sebesar 0.0193 detik, dan tingkat akurasi data sebesar 92.79%.

Penelitian selanjutnya dilakukan oleh Sudana *et al.* (2017) dengan judul “*Prediction System Of Hydroponic Plant Growth And Development Using Algorithm Fuzzy Mamdani Method*”, membangun sebuah sistem prediksi pertumbuhan tanaman hidroponik menggunakan *Fuzzy Mamdani Method*. Penelitian ini menghasilkan kesimpulan bahwa variabel tingkat nutrisi memberikan pengaruh sebesar 22,5%, variabel tingkat keasaman (pH) memberikan pengaruh sebesar -42,5%, dan variabel suhu memberikan pengaruh sebesar 31,7% terhadap pertumbuhan dan perkembangan tanaman hidroponik sawi caisim.

Selanjutnya, penelitian oleh Sakurai *et al.* (2019) dengan judul “ *Plant growth prediction using convolutional LSTM* ” membangun sebuah sistem prediksi pertumbuhan tanaman dengan memprediksikan frame tanaman yang akan datang berdasarkan frame tanaman sebelumnya menggunakan algoritma *Long Short Term Memory* (LSTM). Penelitian ini juga menggunakan fungsi *difference loss* dan *centroid loss* untuk mendapatkan hasil yang lebih maksimal.

Penelitian berikutnya dilakukan oleh Widiana *et al.* (2020) dengan judul “ *Plant Seeds Growth Prediction on Greenhouse Using Adaptive Neuro Fuzzy Inference System (ANFIS) Method* ” yang membangun suatu sistem prediksi

pertumbuhan bibit tanaman pada *green house* berdasarkan intensitas cahaya, suhu, kelembapan udara, kelembapan tanah, jumlah daun, lebar tanaman dan panjang tanaman. Penelitian ini menerapkan metode Sistem Inferensi Fuzzy Neuro Adaptif (ANFIS), yang menggabungkan prinsip-prinsip fuzzy logic dan jaringan saraf tiruan dengan menghasilkan akurasi sebesar 93,3333% dengan *Mean Absolute Deviation* (MAD) sebesar 64,3391 dari 15 data prediksi sebesar 4,2893, Mean Absolute Percentage Error (MAPE) sebesar 5,3485 dari 15 data prediksi sebesar 0,35657, Mean Square Deviation (MSD) sebesar 9,159 dari 15 data prediksi sebesar 0,6106.

Oleh karena itu, penulis melakukan sebuah penelitian yang bertujuan untuk memprediksi pertumbuhan pada tanaman tomat menggunakan algoritma *Gradient Boosting* dengan judul “Prediksi Pertumbuhan Tanaman Tomat Berdasarkan Pengambilan Citra oleh Robot Menggunakan *Gradient Boosting*”.

1.2 Rumusan Masalah

Metode pertanian konvensional secara umum melakukan pemantauan pertumbuhan secara langsung yang mengharuskan petani terjun langsung ke area tanam. Apabila terjadi kelalaian dalam proses pemantauan terhadap pertumbuhan tanaman akan mengakibatkan terjadinya penghambatan dalam pertumbuhan tanaman itu sendiri.

1.3 Tujuan Penelitian

Tujuan penelitian ini adalah memprediksi pertumbuhan tanaman tomat berdasarkan pengambilan citra oleh robot menggunakan *gradient boosting*.

1.4 Batasan Masalah

Adapun beberapa batasan masalah yang diberikan pada penelitian ini adalah sebagai berikut :

1. Prediksi pertumbuhan tanaman yang dilakukan, yaitu prediksi luas tanaman dan ukuran buah dalam satuan pixel.
2. Citra tanaman disimpan dalam ekstensi file .png
3. Pengambilan citra tanaman dilakukan dengan pencahayaan yang baik.
4. Parameter tambahan yang digunakan untuk prediksi pertumbuhan adalah kelembapan dan pH tanah.
5. Ukuran area tanam yaitu 1,8 m x 4 m, dengan jarak antara kamera dengan tanaman sebesar 50 cm.

6. Prediksi pertumbuhan tanaman dipantau melalui aplikasi berbasis *desktop*.

1.5 Manfaat Penelitian

Adapun manfaat yang terdapat pada penelitian ini adalah mempermudah petani dalam memprediksi pertumbuhan tanaman tanpa harus terjun langsung ke area tanam.

1.6 Metodologi Penelitian

1. Studi Literatur

Tahapan awal yang dilakukan oleh peneliti dengan mencari dan mengumpulkan informasi yang relevan dari berbagai buku, jurnal, skripsi, serta sumber informasi lainnya untuk mendukung penelitian ini.

2. Analisis Permasalahan

Tahapan yang dilalui oleh peneliti dengan cara menganalisis sumber – sumber informasi yang telah terkumpul untuk menemukan sebuah metode yang sesuai dalam menyelesaikan permasalahan yang sedang diteliti.

3. Perancangan

Pada tahap ini, peneliti merancang sistem yang dapat memecahkan masalah yang sedang diteliti. Adapun hal yang dirancang yaitu mulai dari arsitektur umum yang menggambarkan alur proses sistem, hingga desain antar muka sistem yang akan digunakan.

4. Implementasi

Peneliti mengimplementasikan desain antar muka sistem dan alur proses yang telah dirancang sebelumnya kedalam kode - kode pemrograman, sehingga menjadi sebuah sistem yang dapat memecahkan permasalahan yang sedang diteliti.

5. Pengujian

Melakukan sebuah pengujian dan analisis terhadap sistem yang telah dibangun.

6. Penyusunan Laporan

Pada tahap ini, penulis akan mengumpulkan semua hasil dokumentasi terhadap penelitian yang nantinya akan disusun kedalam bentuk sebuah laporan.

1.7 Sistematika Penulisan

Adapun sistematika penulisan pada penelitian kali ini terdiri dari 5 bagian, yaitu:

Bab 1. Pendahuluan

Bab ini penulis membahas banyak bagian, yaitu latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

Bab 2. Landasan Teori

Pada bab ini penulis menjelaskan teori - teori yang terkait dengan permasalahan pada penelitian ini agar nantinya mudah dipahami.

Bab 3. Analisa dan Perancangan

Bab ini penulis membahas mengenai arsitektur umum, perancangan aplikasi dan tampilan, tahapan yang dilakukan pada penelitian, dan metode yang digunakan serta penerapannya.

Bab 4. Implementasi dan Pengujian

Bab ini berisi penjelasan mengenai implementasi dari rancangan yang sudah dibuat pada bab sebelumnya. Pada bab ini penulis juga membahas hasil yang didapat dari pengujian aplikasi apakah sesuai dengan tujuan penelitian.

Bab 5. Kesimpulan dan Saran

Pada bab ini penulis membahas kesimpulan dan rangkuman yang diperoleh dari penjelasan yang sudah diuraikan pada bab-bab sebelumnya, baik itu perancangan, implementas, hasil pengujian, dan lain-lain. Penulis juga menuliskan saran – saran yang sekiranya digunakan untuk pengembangan dan penelitian selanjutnya agar lebih baik lagi.

BAB 2

LANDASAN TEORI

2.1 Robot Pertanian



Gambar 2. 1 Agribot / Robot Pertanian

Robot pertanian adalah sebuah hasil implementasi antara kecerdasan buatan dan teknologi penginderaan yang bertujuan untuk memantau pertumbuhan dan menjaga keseimbangan ekosistem area tanam. Dengan otomatisasi, mesin pertanian dapat beroperasi dengan lebih efisien, dapat diandalkan dan memiliki tingkat presisi yang lebih tinggi yang pada gilirannya dapat mengurangi keterlibatan pekerja manusia dalam berbagai tugas pertanian. Hal ini dapat menyebabkan peningkatan keseluruhan produktivitas dan efisiensi operasional dalam sektor pertanian (Schueller *et al.*, 2006).

2.2 Kecerdasan Buatan (*Artificial Intelligence*)

Kecerdasan buatan adalah sebuah sistem dimana komputer dapat menyelesaikan suatu masalah yang seharusnya diselesaikan oleh manusia dengan melakukan beberapa latihan untuk mendapatkan hasil atau keputusan dalam menyelesaikan masalah tersebut. Kecerdasan buatan atau artificial intelligence merupakan bidang ilmu komputer yang berfokus pada pengembangan sistem yang dapat bekerja secara otomatisasi dan menampilkan perilaku cerdas (Kristianto, 2004). Salah satu contoh

sederhana dalam penerapan kecerdasan buatan pada kegiatan sehari-hari, yaitu adanya robot pembersih lantai atau biasa disebut *vacuum cleaner robot*, dimana robot tersebut dapat membersihkan lantai yang terdapat kotoran secara otomatis.

2.3 Kelembapan Tanah

Kelembapan tanah adalah jumlah air yang terdapat dalam tanah pada suatu waktu tertentu dan merupakan faktor kritis yang mempengaruhi pertumbuhan tanaman, penyerapan nutrisi, serta keseimbangan air dalam tanaman. Faktor-faktor yang menentukan kelembapan tanah adalah curah hujan, jenis tanah, dan laju evapotranspirasi, dimana kelembapan tanah akan menentukan ketersediaan air dalam tanah bagi pertumbuhan tanaman (Djumali & Mulyaningsih, 2014). Pengukuran kelembapan tanah dapat dilakukan dengan berbagai metode seperti gravimetrik, sensor kapasitif dan resistif, serta tensiometer, yang memberikan informasi penting untuk pengelolaan irigasi dan praktik pertanian berkelanjutan (Muñoz-Carpena *et al.*, 2010). Sebagai contoh, tanaman tomat membutuhkan kelembapan tanah yang konsisten dan optimal berada pada kisaran 60-70% dari kapasitas lapang untuk mendukung pertumbuhan dan produksi buah yang maksimal (Patane & Cosentino, 2010).

2.4 pH Tanah

pH tanah adalah ukuran keasaman atau kebasaan relatif tanah yang mempengaruhi ketersediaan nutrisi tanaman. Tanah yang terlalu asam (pH rendah) dapat menghambat penyerapan nutrisi penting seperti fosfor dan magnesium, sedangkan tanah yang terlalu basa (pH tinggi) dapat mengurangi ketersediaan unsur hara seperti besi dan mangan. Tanaman tomat (*Solanum lycopersicum*) umumnya tumbuh optimal pada rentang pH tanah antara 6 hingga 7.5, di mana nutrisi lebih mudah diserap dan proses biokimia dalam tanaman berjalan efisien. Perubahan signifikan dalam pH tanah dapat mengganggu pertumbuhan tanaman tomat, mempengaruhi produktivitasnya secara langsung melalui akses terhadap nutrisi esensial (Brady & Weil, 2008).

2.5 Deep Learning

Deep Learning merupakan sebuah metode yang menjadi bagian dari *artificial intelligence* dan *machine learning*, dimana dalam prosesnya menggunakan *neural network multiple layer* untuk menyelesaikan masalah secara akurat dan melakukan representasi data secara otomatis. *Deep learning* dibagi menjadi 3 kategori (Deng & Yu 2014) , yaitu :

- *Deep Networks for Unsupervised or Generative Learning*

Terdapat tingkat korelasi yang tinggi terhadap data yang akan diamati atau analisis pola ketika informasi label kelas target tidak tersedia.

- *Deep Networks for Supervised Learning*

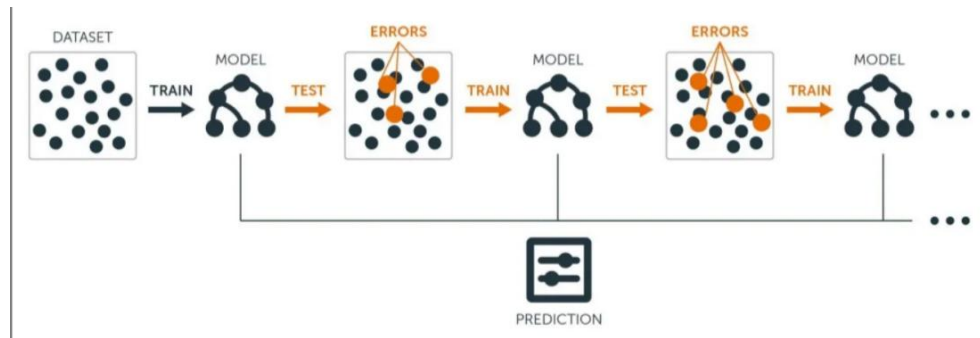
Dalam menentukan klasifikasi pola dilakukan tindakan secara diskriminatif dengan memberikan karakteristik dari distribusi posterior kelas yang disesuaikan dengan data yang diamati.

- *Hybrid Deep Networks*

Mendiskriminasi secara signifikan terhadap hasil yang terdapat pada *unsupervised learning*.

2.6 Gradient Boosting

Gradient Boosting merupakan sebuah metode atau teknik yang mengkombinasikan model yang lemah menjadi model yang lebih kuat. Algoritma ini membangun model dengan melakukan penyesuaian secara berurutan pada fungsi parameter sederhana terhadap “pseudo” residual yang bertujuan untuk mengurangi residual yang terjadi disetiap iterasinya, sehingga menghasilkan menghasilkan prosedur-prosedur yang memiliki kinerja yang baik, stabil, mudah diinterpretasikan baik untuk memprediksi nilai numerik (regresi) maupun untuk mengklasifikasikan data (Friedman, 2001). Bentuk arsitekturnya sendiri memiliki kemiripan dengan pola pada *decision tree*, dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Arsitektur *Gradient Boosting*

(sumber : <https://medium.com/analytics-vidhya/introduction-to-the-gradient-boosting-algorithm-c25c653f826b>)

Alur *Gradient Boosting* sendiri melalui beberapa tahapan, yaitu:

1. Inisialisasi Model

Merupakan langkah awal pada *gradient boosting* dengan menghitung rata-rata dari target. Adapun output pada tahap ini dapat dihitung menggunakan persamaan 2.1 (Friedman, 2001).

$$F_0(x) = \operatorname{argmin}_{\gamma} \sum_{i=1}^n L(y_i, \gamma) \quad (2.1)$$

Keterangan :

$F_0(x)$: Prediksi awal.

L : *Loss function*.

y_i : Data aktual

γ : Data prediksi

2. Residual Error

Pada tahap ini dilakukan perhitungan residual error yang dapat dilihat pada persamaan 2.2 (Friedman, 2001).

$$r_{im} = - \left[\frac{\partial L(y_i, F(x))}{\partial F(x)} \right]_{F(x) = F_{m-1}(x)} \quad (2.2)$$

Keterangan :

F_{m-1} : Prediksi sebelumnya

∂L : Diferensial dari *loss function*

r_{im} : Residual error

y_i : Nilai Aktual

Berdasarkan persamaan di atas, perhitungan residu dilakukan dengan mengambil turunan dari *loss function* terhadap prediksi sebelumnya F_{m-1} dan dikali dengan -1.

Residu ini, dinotasikan sebagai r_{im} yang merupakan gradien negatif yang memberikan arah dan besaran dimana *loss function* dapat diminimalkan. Pada iterasi pertama, F_{m-1} mengacu pada prediksi dari langkah sebelumnya, yaitu F_0 . Langkah ini bertujuan untuk menyelesaikan persamaan untuk mendapatkan nilai r_{im} , untuk lebih mudahnya kita dapat menggunakan rumus turunannya yang dapat dilihat pada persamaan 2.3 (Friedman, 2001).

$$\begin{aligned}\frac{\partial L}{\partial \gamma} &= -(y_i - \gamma_i) = -(Observed - Predicted) \\ &= (Observed - Predicted)\end{aligned}\quad (2.3)$$

3. *Weak Learner*

Dalam tahap ini, weak learner dikembangkan sebagai model yang mampu memperkirakan kesalahan dari model sebelumnya. Model lemah ini umumnya berupa decision tree sederhana yang hanya memiliki 1 atau 2 cabang. Dengan kata lain, weak learner ini dirancang untuk fokus pada pola-pola sederhana dalam kesalahan prediksi yang dilakukan oleh model sebelumnya (Friedman, 2001).

4. *Update Model*

Pada langkah ini, model baru yang dibangun dengan menggunakan decision tree untuk memprediksi residu digabungkan dengan model sebelumnya, menghasilkan model yang lebih unggul. Dengan menggabungkan model-model ini, kita memanfaatkan kekuatan masing-masing model untuk meningkatkan kinerja secara keseluruhan. Proses ini terus diulangi dalam beberapa iterasi, dengan setiap iterasi berfokus pada memperbaiki kesalahan prediksi model sebelumnya. Dengan demikian, setiap langkah tambahan dalam proses ini bertujuan untuk meningkatkan akurasi dan kinerja model secara keseluruhan. Adapun rumus perhitungannya dapat dilihat pada persamaan 2.4 (Friedman, 2001).

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \quad (2.4)$$

Keterangan :

- F_m : Prediksi model gabungan setelah iterasi ke-m.
- F_{m-1} : Prediksi model gabungan sebelumnya setelah iterasi ke-(m-1).
- γ : *learning rate* yang berkisar antara 0 dan 1.
- m : Indeks iterasi
- $h_m(x)$: Prediksi dari pohon keputusan yang dibuat pada iterasi ke-m untuk data input x.

2.7 Penelitian Terdahulu

Beberapa penelitian yang berkaitan dengan prediksi pertumbuhan tanaman, yaitu penelitian oleh Pohan *et al.* (2020) dengan judul “ *Backpropagation Artificial Neural Network for Prediction Plant Seedling Growth* ” membangun sebuah sistem berbasis dekstop untuk memprediksi pertumbuhan bibit tanaman berdasarkan data intensitas cahaya, suhu, kelembapan udara, kelembapan tanah, jumlah daun, lebar tanaman dan panjang batang dengan menggunakan metode *Backpropagation Artificial Neural Network*. Adapun hasil dari pengaplikasian algoritma ini untuk memprediksi pertumbuhan bibit tanaman di rumah kaca menunjukkan hasil yang baik, dengan Mean Squared Error (MSE) dari iterasi pertama sebesar 0.0112, waktu komputasi sebesar 0.0193 detik, dan tingkat akurasi data sebesar 92.79%.

Penelitian selanjutnya dilakukan oleh Sudana *et al.* (2017) dengan judul “*Prediction System Of Hydroponic Plant Growth And Development Using Algorithm Fuzzy Mamdani Method*”, membangun sebuah sistem prediksi pertumbuhan tanaman hidroponik menggunakan *Fuzzy Mamdani Method*. Penelitian ini menghasilkan kesimpulan bahwa variabel tingkat nutrisi memberikan pengaruh sebesar 22,5%, variabel tingkat keasaman (pH) memberikan pengaruh sebesar -42,5%, dan variabel suhu memberikan pengaruh sebesar 31,7% terhadap pertumbuhan dan perkembangan tanaman hidroponik sawi caisim.

Selanjutnya, penelitian oleh Sakurai *et al.* (2019) dengan judul “ *Plant growth prediction using convolutional LSTM* ” membangun sebuah sistem prediksi pertumbuhan tanaman dengan memprediksikan frame tanaman yang akan datang berdasarkan frame tanaman sebelumnya menggunakan algoritma *Long Sort Term Memory* (LSTM). Penelitian ini juga menggunakan fungsi *difference loss* dan *centroid loss* untuk mendapatkan hasil yang lebih maksimal.

Penelitian berikutnya dilakukan oleh Widiana *et al.* (2020) dengan judul “ *Plant Seeds Growth Prediction on Greenhouse Using Adaptive Neuro Fuzzy Inference System (ANFIS) Method* ” yang membangun suatu sistem prediksi pertumbuhan bibit tanaman pada *green house* berdasarkan intensitas cahaya, suhu, kelembapan udara, kelembapan tanah, jumlah daun, lebar tanaman dan panjang tanaman. Penelitian ini menerapkan metode *Sistem Inferensi Fuzzy Neuro Adaptif* (ANFIS), yang menggabungkan prinsip-prinsip *fuzzy logic* dan jaringan saraf tiruan dengan menghasilkan akurasi sebesar 93,3333% dengan *Mean Absolute Deviation*

(MAD) sebesar 64,3391 dari 15 data prediksi sebesar 4,2893, *Mean Absolute Percentage Error* (MAPE) sebesar 5,3485 dari 15 data prediksi sebesar 0,35657, *Mean Square Deviation* (MSD) sebesar 9, 159 dari 15 data prediksi sebesar 0,6106.

Penelitian ini berbeda dari penelitian sebelumnya dalam beberapa aspek. Pohan et al. (2020) menggunakan jaringan saraf tiruan *backpropagation* untuk memprediksi pertumbuhan bibit tanaman berdasarkan data lingkungan, seperti intensitas cahaya dan suhu. Sudana et al. (2017) menggunakan metode fuzzy Mamdani untuk memprediksi pertumbuhan tanaman hidroponik berdasarkan parameter nutrisi dan pH. Penelitian penulis menggunakan citra tanaman dan data sensor pH serta kelembapan yang diambil oleh robot, serta algoritma *gradient boosting* untuk prediksi pertumbuhan. Berbeda dengan Sakurai et al. (2019) yang menggunakan LSTM untuk memprediksi frame tanaman, penelitian ini fokus pada citra statis dan analisis datanya melalui *gradient boosting*. Selain itu, penelitian ini memperkenalkan penggunaan robot untuk akuisisi data, memberikan pendekatan inovatif dalam pengumpulan data visual tanaman, berbeda dengan Widiana et al. (2020) yang menggunakan ANFIS untuk prediksi di rumah kaca.

Tabel 2. 1 Penelitian Terdahulu

No	Judul Penelitian	Peneliti	Tahun	Keterangan
1	<i>Backpropagation Artificial Neural Network for Prediction Plant Seedling Growth</i>	<ul style="list-style-type: none"> • S Pohan • B Warsito • S Suryono 	2020	Memprediksi pertumbuhan bibit tanaman berdasarkan data intensitas cahaya, suhu, kelembapan udara, kelembapan tanah, jumlah daun, lebar tanaman dan panjang batang dengan menggunakan metode <i>Backpropagation Artificial Neural Network</i> . Adapun hasil dari pengaplikasian algoritma ini untuk memprediksi pertumbuhan bibit tanaman di rumah kaca menunjukkan hasil yang baik, dengan <i>Mean Squared Error</i> (MSE) dari iterasi pertama sebesar 0.0112, waktu komputasi sebesar 0.0193 detik, dan tingkat akurasi data sebesar 92.79%.

Tabel 2. 2 Penelitian Terdahulu (Lanjutan)

No	Judul Penelitian	Peneliti	Tahun	Keterangan
2	<i>Prediction System Of Hydroponic Plant Growth And Development Using Algorithm Fuzzy Mamdani Method.</i>	<ul style="list-style-type: none"> • I Made Sudana • Okta Purnawirawan • Ulfa Mediaty Arief 	2017	Membangun sebuah sistem prediksi pertumbuhan tanaman hidroponik menggunakan Fuzzy Mamdani Method. Penelitian ini menghasilkan kesimpulan bahwa variabel tingkat nutrisi memberikan pengaruh sebesar 22,5%, variabel tingkat keasaman (pH) memberikan pengaruh sebesar -42,5%, dan variabel suhu memberikan pengaruh sebesar 31,7% terhadap pertumbuhan dan perkembangan tanaman hidroponik sawi caisim.
3	<i>Plant Growth Prediction using Convolutional LSTM</i>	<ul style="list-style-type: none"> • Shunsuke Sakurai • Hideaki Uchiyama • Atshushi Shimada • Rin-ichiro Taniguchi 	2020	Membangun sebuah sistem prediksi pertumbuhan tanaman dengan memprediksikan frame tanaman yang akan datang berdasarkan frame tanaman sebelumnya menggunakan algoritma Long Short Term Memory (LSTM). Penelitian ini juga menggunakan fungsi difference loss dan centroid loss untuk mendapatkan hasil yang lebih maksimal.

Tabel 2. 1 Penelitian Terdahulu (Lanjutan)

No	Judul Penelitian	Peneliti	Tahun	Keterangan
4	<i>Plant Seeds Growth Prediction on Greenhouse Using Adaptive Neuro Fuzzy Inference System (ANFIS) Method.</i>	<ul style="list-style-type: none"> • Siska Ayu Widiana • Suryono • Budi Warsito 	2020	Membangun suatu sistem prediksi pertumbuhan bibit tanaman pada <i>green house</i> berdasarkan intensitas cahaya, suhu, kelembapan udara, kelembapan tanah, jumlah daun, lebar tanaman dan panjang tanaman. Penelitian ini menggunakan metode <i>Adaptive Neuro Fuzzy Inference System</i> (ANFIS) yang mana gabungan antara metode <i>fuzzy</i> dan <i>neural network</i> . Metode ini menghasilkan akurasi sebesar 93,3333% dengan <i>Mean Absolute Deviation</i> (MAD) sebesar 64,3391 dari 15 data prediksi sebesar 4,2893, <i>Mean Absolute Percentage Error</i> (MAPE) sebesar 5,3485 dari 15 data prediksi sebesar 0,35657, <i>Mean Square Deviation</i> (MSD) sebesar 9,159 dari 15 data prediksi sebesar 0,6106.

BAB 3

ANALISIS DAN PERANCANGAN

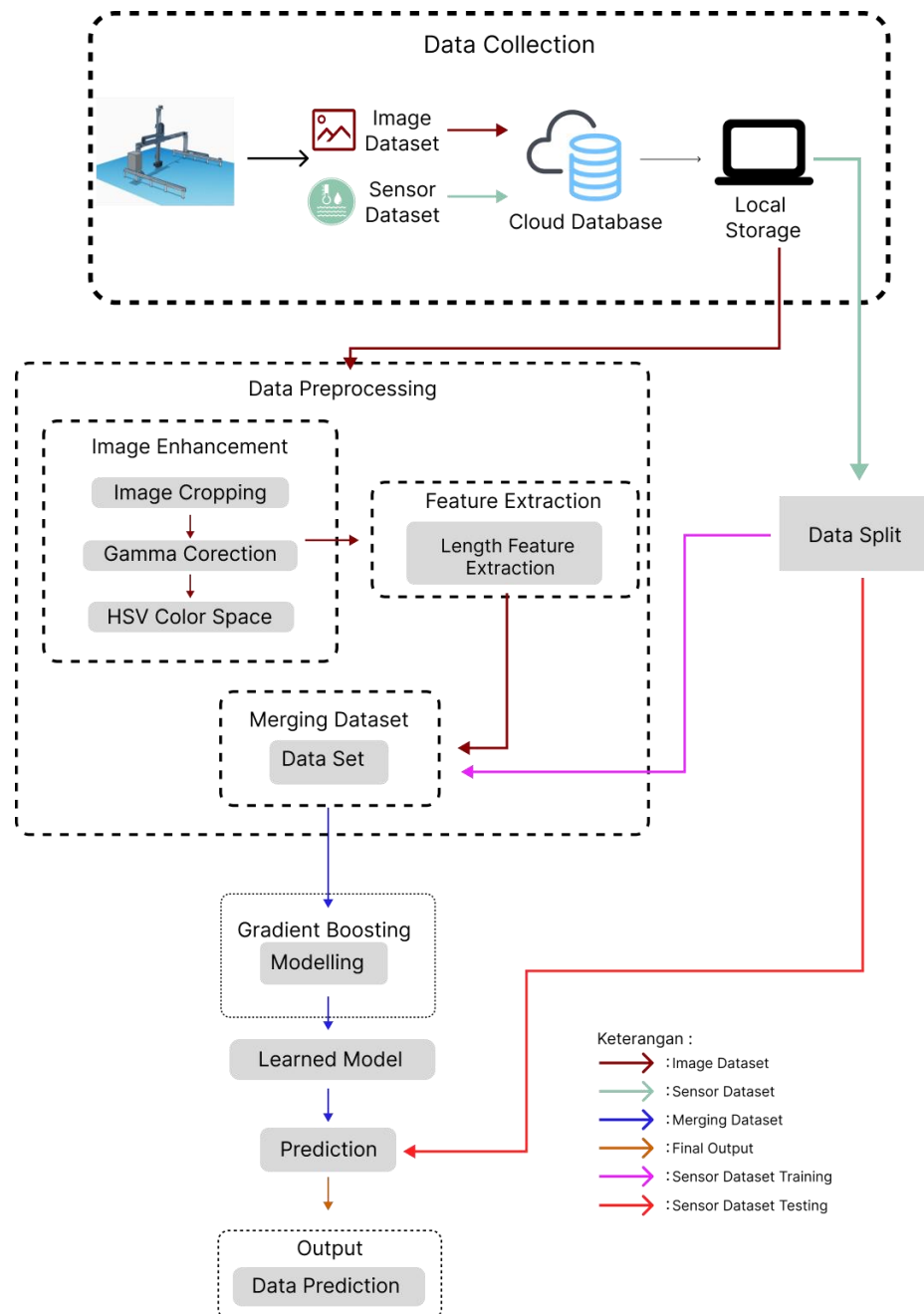
3.1 Data yang digunakan

Data yang digunakan penulis dalam mendukung penelitian ini meliputi data luas tanaman, luas buah, waktu / hari, kelembapan tanah, dan pH tanah. Data luas tanaman dan buah diperoleh dari citra tanaman dan buah tomat dengan menghitung nilai piksel yang bukan 0, pengambilan citra dilakukan sebanyak empat kali sehari pada pukul 09.00 WIB, 12.00 WIB, 15.00 WIB, dan 17.00 WIB menggunakan kamera webcam Logitech, kemudian disimpan dalam format PNG (*Portable Network Graphics*) ke dalam cloud server. Teknik pengambilan citra tanaman dilakukan dengan mengambil foto tanaman dari samping, di mana kamera webcam ditempatkan tepat di sisi kiri lahan tanam. Data yang digunakan meliputi luas tanaman dan buah, hari ke-, pH tanah dan kelembapan tanah, yang kemudian digabungkan ke dalam sebuah file dengan format CSV (*Comma Separated Values*) untuk diolah dan dijadikan model prediksi pertumbuhan tanaman. Penelitian ini menggunakan jumlah data sebanyak 473 untuk dataset tanaman dan 494 untuk dataset buah. Data tersebut kemudian dilakukan pembagian untuk *data training* 80% dan *data testing* 20%.

3.2 Analisis Sistem

3.2.1 Arsitektur Umum

Arsitektur umum merujuk pada suatu diagram atau representasi visual yang menggambarkan proses dari sistem yang sedang dikembangkan. Adapun arsitektur umum dari penelitian ini dapat dilihat pada gambar 3.1.



Gambar 3. 1 Arsitektur Umum

3.2.2 Data Collection

Tahap pertama merupakan proses pengumpulan data gambar dan data sensor. Data – data tersebut berasal dari data yang ditangkap oleh kamera dan sensor yang terdapat pada robot pertanian. Nantinya data terbagi dua pada tahap *data split*, yaitu *data training* dan *data testing*. *Data training* merupakan data yang telah dikumpulkan untuk dilatih dan *data testing* merupakan data yang digunakan untuk pengujian pada sistem. Contoh citra tanaman dapat dilihat pada Gambar 3.2



Gambar 3. 2 Contoh Gambar Tanaman dan Buah Tomat

3.2.3 *Data Pre-processing*

Pada tahap *pre-processing* data terbagi menjadi 3 proses, yaitu ;

3.2.3.1 *Image cropping*

Citra akan dipotong sesuai dengan ukuran yang dibutuhkan untuk mempermudah proses pengolahan data. Adapun cara *cropping* citra menggunakan bantuan *library* scikit-image, dimana pemotongan citra dilakukan berdasarkan titik koordinat yang telah disesuaikan. Proses pemotongan citra dan hasil pemotongan dapat dilihat pada Gambar 3.3 serta penjelasan dibawah ini ;

```
# Deklarasi variabel gambar
deklarasikan gambar_tanaman
deklarasikan gambar_buah

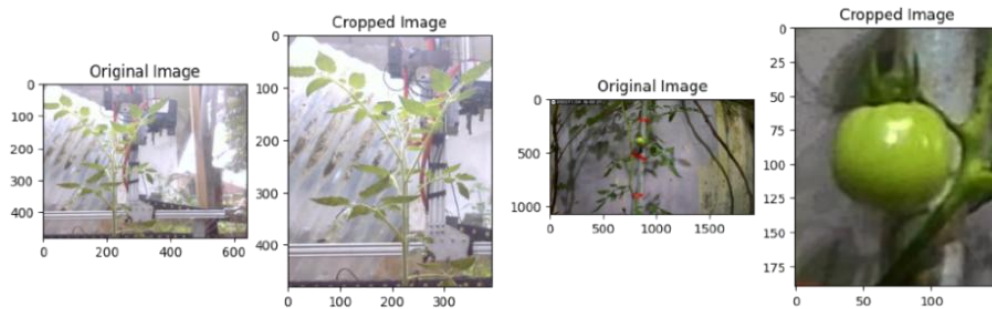
# Memotong gambar tanaman
# X : 0 sampai 189
# Y : semua baris
gambar_tanaman_potong = gambar_tanaman [ semua baris, kolom 0 sampai 189 ]

# Memotong gambar buah
# X : 770 sampai 919
# Y : 300 sampai 489
gambar_buah_potong = gambar_buah [baris 300 sampai 489, kolom 770 sampai 919]
```

Penjelasan :

Pada rumusan diatas terdapat enam variable yaitu image, cropped_image, y1, y2, x1 dan x2. Image merupakan citra awal yang belum dilakukan pemotongan, sedangkan

`cropped_image` merupakan hasil dari citra yang telah dipotong. `Y1` dan `y2` merupakan batas awal dan batas akhir pada koordinat `y` atau secara vertikal yang akan dipotong pada citra. `X1` dan `x2` merupakan batas awal dan batas akhir pada koordinat `x` atau secara horizontal yang dipotong pada citra. Pada penelitian ini citra dipotong berdasarkan nilai koordinat yang ada pada sumbu `x` atau secara horizontal saja yaitu 0 : 190 untuk tanaman. Adapun hasil proses *cropping* dapat dilihat pada gambar 3.3 yang mana *original image* merupakan gambar sebelum *cropping* dan *cropped_image* merupakan gambar setelah *cropping*.



Gambar 3. 3 Sebelum dan Sesudah *Cropping*

3.2.3.2 *Gamma Correction*

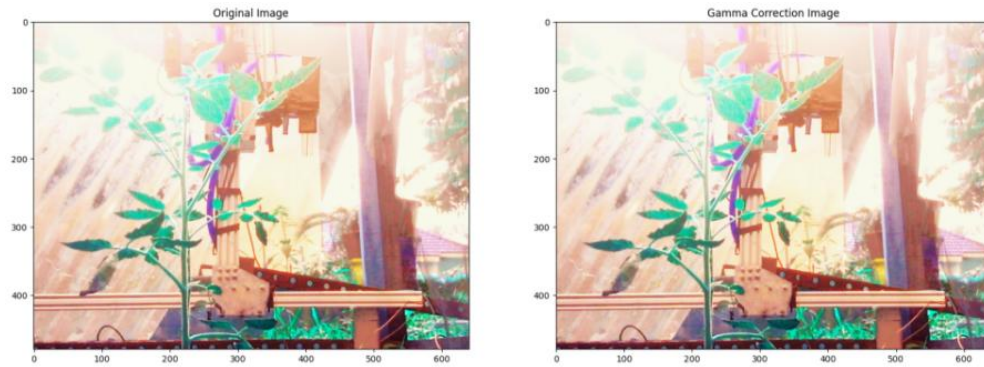
Setelah proses *cropping*, selanjutnya citra akan memasuki tahapan *gamma correction* yang dilakukan untuk memperbaiki kecerahan, kontras, dan reproduksi warna dalam citra tanaman. Dengan bantuan *library scikit-image* dan melakukan import *exposure* untuk dapat menggunakan fungsi `exposure.adjust_gamma(image, 1)`.

```
# Deklarasi variabel gambar tomat
deklarasikan gambar_tomat

# Koreksi gamma pada gambar tomat dengan faktor gamma 1
gambar_koreksi_gamma = exposure.adjust_gamma(gambar_tomat, 1)
```

Keterangan :

<code>gamma_corrected_image</code>	: Variabel yang menyimpan citra hasil dari proses <i>gamma correction</i> .
<code>exposure.adjust_gamma</code>	: Fungsi <i>gamma correction</i> pada citra.
<code>tomato_image</code>	: Variabel yang menyimpan citra asli.
1	: Nilai gamma



Gambar 3. 4 Sebelum dan Sesudah *Gamma Correction*

Adapun gambar 3.4 merupakan hasil implementasi dari proses *gamma correction*, yang mana *original image* merupakan gambar sebelum *gamma correction* dan *gamma correction image* merupakan gambar setelah *gamma correction*.

3.2.3.3 *HSV color space*

Pendeteksian warna yang sudah ditentukan pada citra agar dapat diseleksi bagian yang ingin diolah. Pada penelitian kali ini menggunakan bantuan *HSV color space*.

```
# Deklarasi variabel gambar
deklarasikan gambar

# Konversi gambar dari RGB ke HSV
hsv = cv2.cvtColor(gambar, cv2.COLOR_RGB2HSV)

# Deklarasi batas atas dan batas bawah warna biru
batas_atas_biru = array([75, 255, 255])
batas_bawah_biru = array([24, 38, 49])

# Buat mask untuk warna biru
mask = cv2.inRange(hsv, batas_bawah_biru, batas_atas_biru)

# Terapkan mask pada gambar asli
hasil = cv2.bitwise_and(gambar, gambar, mask=mask)
```

Penjelasan :

1. `hsv = cv2.cvtColor(im, cv2.COLOR_RGB2HSV)`

Tahapan ini mengonversi citra `im` dari RGB ke HSV menggunakan fungsi `cv2.cvtColor` dari OpenCV. Citra dalam HSV memungkinkan pemrosesan yang lebih mudah untuk pemisahan warna berdasarkan hue, saturation, dan value.

2. `upper_blue = np.array([75,255,255]), lower_blue = np.array([24,38, 49])`

Baris ini mendefinisikan nilai batas atas dan batas bawah warna biru dalam HSV, tersimpan dalam bentuk array yang memiliki tiga nilai yang mewakili Hue (nilai warna), Saturation (nilai kejenuhan), dan Value (nilai kecerahan) yang merupakan representasi dari warna biru yang diinginkan.

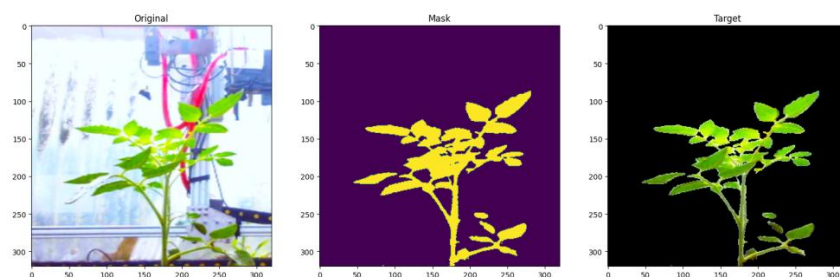
3. `mask = cv2.inRange(hsv, lower_blue, upper_blue)`

Fungsi `cv2.inRange` untuk membuat *masking* yang mencakup piksel-piksel dalam rentang warna biru yang ditentukan oleh `lower_blue` dan `upper_blue` di dalam citra `hsv`.

4. `res = cv2.bitwise_and(im, im, mask=mask)`

Fungsi `cv2.bitwise_and` untuk menerapkan *masking* yang telah dibuat sebelumnya (*mask*) pada citra asli `im`. Ini menghasilkan citra `res` yang hanya menampilkan bagian dari citra asli yang berada dalam rentang warna biru yang ditentukan.

Adapun hasil proses *CSV Color Space* dapat dilihat pada gambar 3.5, yang mana menampilkan hasil gambar *original*, gambar hasil proses *masking* dan gambar target yang merupakan hasil akhir dari proses *hsv color space* yang bertujuan untuk memisahkan objek dengan *background* gambar.



Gambar 3. 5 HSV Color Space

3.2.4 Feature Extraction

Untuk ekstraksi fitur pada penelitian ini, penulis menggunakan *length feature extraction* atau fitur ekstraksi Panjang dengan menghitung jumlah piksel yang nilainya bukan 0 yang ada pada gambar. Kemudian penulis menggunakan fungsi `countNonZero` yang terdapat pada *library* `cv2`.

```
# Deklarasi variabel mask
deklarasikan mask

# Hitung jumlah piksel tidak nol dalam mask
jumlah_tidak_nol = cv2.countNonZero(mask)
```

Keterangan :

Jumlah_tidak_nol : Hasil perhitungan jumlah piksel dari citra tanaman.

Mask : File citra tanaman

3.2.5 Merging Dataset

Pada tahap ini dilakukan penggabungan antara data pixel gambar dan data sensor kedalam sebuah file dataset dengan format *comma separated values* (CSV).

3.2.6 Gradient Boosting

Setelah penggabungan dataset, proses selanjutnya akan berlanjut dengan membangun sebuah model menggunakan teknik *gradient boosting*, di mana beberapa model lemah digabungkan menjadi model baru yang lebih kuat. Adapun tahapan yang akan dilalui adalah sebagai berikut:

3.2.6.1 Inisialisasi Model

Adapun contoh perhitungan untuk inisialisasi model diambil berdasarkan persamaan 2.1., dimana kita akan menggunakan rumus turunannya yang dapat dilihat pada persamaan 3.1.

$$F_0(x) = \frac{1}{n} \sum_{i=0}^n y_i \quad (3.1)$$

Diketahui:

$$X = [1, 2, 3, 4, 5]$$

$$Y = [3, 6, 7, 8, 11]$$

$$\begin{aligned} F_0(x) &= \frac{1}{5} (3 + 6 + 7 + 8 + 11) \\ &= \frac{35}{5} = 7 \end{aligned}$$

Keterangan :

$F_0(x)$: Prediksi dari model dasar.

X : Fitur.

Y : target.

n : Jumlah data observasi.

I : Jumlah iterasi.

3.2.6.2 *Residual Error*

Pada tahap ini dilakukan perhitungan residual error yang berdasarkan persamaan 2.3 berikut:

Tabel 3. 1 Perhitungan *Residual Error*

X	Y	$F_0(x)$	$r_{im} = (y_i - \gamma_i)$
1	3	7	$3 - 7 = -4$
2	6	7	$6 - 7 = -1$
3	7	7	$7 - 7 = 0$
4	8	7	$8 - 7 = 1$
5	11	7	$11 - 7 = 4$

Keterangan :

$F_0(x)$: Prediksi dari model dasar.

X : Fitur.

Y : target.

r_{im} : *Residual error*

y_i : Nilai Aktual

γ_i : Nilai Prediksi

3.2.6.3 *Weak Learner*

Dalam tahap ini, weak learner dikembangkan sebagai model yang mampu memperkirakan kesalahan dari model sebelumnya. Fit weak learner pada residual error. Misalkan kita menggunakan decision stump yang memilih split di $X = 3$. sebagai berikut:

Diketahui:

- Jika $X \leq 3$, prediksi residual *error* rata-rata dari $X - 3$.
- Jika $X > 3$, prediksi residual *error* rata - rata dari X.

1. Rata - rata residual error untuk $X \leq 3$

$$\frac{-4 + (-1) + 0}{3} = \frac{-5}{3} = -1.67$$

2. Rata - rata residual error untuk $X > 3$

$$\frac{1+4}{2} = \frac{5}{2} = 2.5$$

Maka dari itu, *weak learner* $h_1(x)$ adalah

$$\begin{cases} -1.67, & \text{jika } x \leq 3 \\ 2.5, & \text{jika } x > 3 \end{cases}$$

Keterangan

1. Rata-rata residual error untuk $X < 3$:
2. *Update Model*
3. *Weak Learner*
4. *Update Model*

3.2.6.4 *Update Model*

Pada tahap ini, model baru yang dibuat dengan menggunakan decision tree yang memprediksi residu digabungkan dengan model sebelumnya untuk membentuk model yang lebih baik. Dengan menggabungkan model-model ini, kita memanfaatkan kekuatan masing-masing model untuk meningkatkan kinerja secara keseluruhan. Proses ini terus diulangi dalam beberapa iterasi, dengan setiap iterasi berfokus pada memperbaiki kesalahan prediksi model sebelumnya. Dengan demikian, setiap langkah tambahan dalam proses ini bertujuan untuk meningkatkan akurasi dan kinerja model secara keseluruhan. Adapun untuk contoh perhitungannya berdasarkan persamaan 2.4 sebagai berikut:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

1. Untuk $X \leq 3$

$$F_1(x) = 7 + 0,1 \times (-1.67) = 6.833$$

2. Untuk $X > 3$

$$F_1(x) = 7 + 0,1 \times 2.5 = 7.25$$

Maka dari itu, *weak learner* $h_1(x)$ adalah

$$\begin{cases} 6.833, & \text{jika } x \leq 3 \\ 7.25, & \text{jika } x > 3 \end{cases}$$

Keterangan :

F_m : Prediksi model gabungan setelah iterasi ke-m.

F_{m-1} : Prediksi model gabungan sebelumnya setelah iterasi ke-(m-1).

- γ : *learning rate* yang berkisar antara 0 dan 1.
 m : Indeks iterasi
 $h_m(x)$: Prediksi dari pohon keputusan yang dibuat pada iterasi ke- m .

3.2.7 Learned Model

Model yang terlatih yaitu model yang didapatkan setelah melalui tahap pelatihan menggunakan algoritma *gradient boosting* dan telah divalidasi menggunakan *Mean Square Error* (MSE), *Mean Absolute Error* (MAE) dan R^2 , adapun perhitungannya dapat dilihat di bawah ini:

a. *Mean Square Error* (MSE)

$$MSE = \frac{1}{n} \sum_{i=0}^n (y_i - \gamma_i)^2 \quad (3.1)$$

Diketahui:

1. Nilai sebenarnya (y_{true}): [3, -0.5, 2, 7]
2. Nilai prediksi (y_{pred}): [2.5, 0.0, 2, 8]

$$\begin{aligned} MSE &= \frac{1}{4} ((3 - 2.5)^2 + (-0.5 - 0.0)^2 + (2 - 2)^2 + (7 - 8)^2) \\ &= \frac{1}{4} (0.25 + 0.25 + 0 + 1) = \frac{1}{4} \times 1.5 = 0.375 \end{aligned}$$

Keterangan :

- n : Jumlah observasi
 y_i : Nilai aktual
 γ_i : Nilai prediksi

b. *Mean Absolute Error* (MAE)

$$MAE = \frac{1}{n} \sum_{i=0}^n |y_i - \gamma_i| \quad (3.2)$$

$$\begin{aligned} MAE &= \frac{1}{4} (|3 - 2.5| + |-0.5 - 0.0| + |2 - 2| + |7 - 8|) \\ &= \frac{1}{4} (0.5 + 0.5 + 0 + 1) = \frac{1}{4} \times 2 = 0.5 \end{aligned}$$

Keterangan :

- n : Jumlah observasi
 y_i : Nilai aktual
 γ_i : Nilai prediksi

c. R^2

$$R^2 = 1 - \frac{\sum_{i=0}^n (y_i - \gamma_i)^2}{\sum_{i=0}^n (y_i - \bar{y})^2} \quad (3.3)$$

- Rata-rata nilai aktual (\bar{y})

$$\bar{y} = \frac{3 + (-0.5) + 2 + 7}{4} = \frac{11.5}{4} = 2.875$$

- Sum of squares of residuals (SS_{res}):

$$SS_{res} \sum_{i=1}^n (y_i - \gamma_i)^2 = 0.25 + 0.25 + 0 + 1 = 1.5$$

- Total sum of squares (SS_{tot}):

$$\begin{aligned} SS_{tot} \sum_{i=1}^n (y_i - \bar{y})^2 &= (3 - 2.875)^2 + (-0.5 - 2.875)^2 + (2 - 2.875)^2 + (7 - 2.875)^2 \\ &= 0.0156 + 11.3906 + 0.7656 + 17.0156 = 29.1875 \end{aligned}$$

- R^2

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{1.5}{29.1875} = 1 - 0.0514 = 0.9486$$

Keterangan :

n : Jumlah observasi

y_i : Nilai aktual

γ_i : Nilai prediksi

3.2.8 Predicition

Model yang telah dilatih digunakan untuk membuat prediksi luas tanaman ataupun buah berdasarkan data input berupa hari, kelembapan tanah dan pH tanah.

3.2.9 Output

Adapun *output* dari penerapan metode *gradient boosting* ini merupakan hasil prediksi luas tanaman dan luas buah yang didapatkan berdasarkan jumlah pixel yang bukan 0 dan menampilkan grafik model persamaan *gradient boosting*, yang nantinya dapat digunakan untuk analisis lebih lanjut atau pengambilan keputusan.

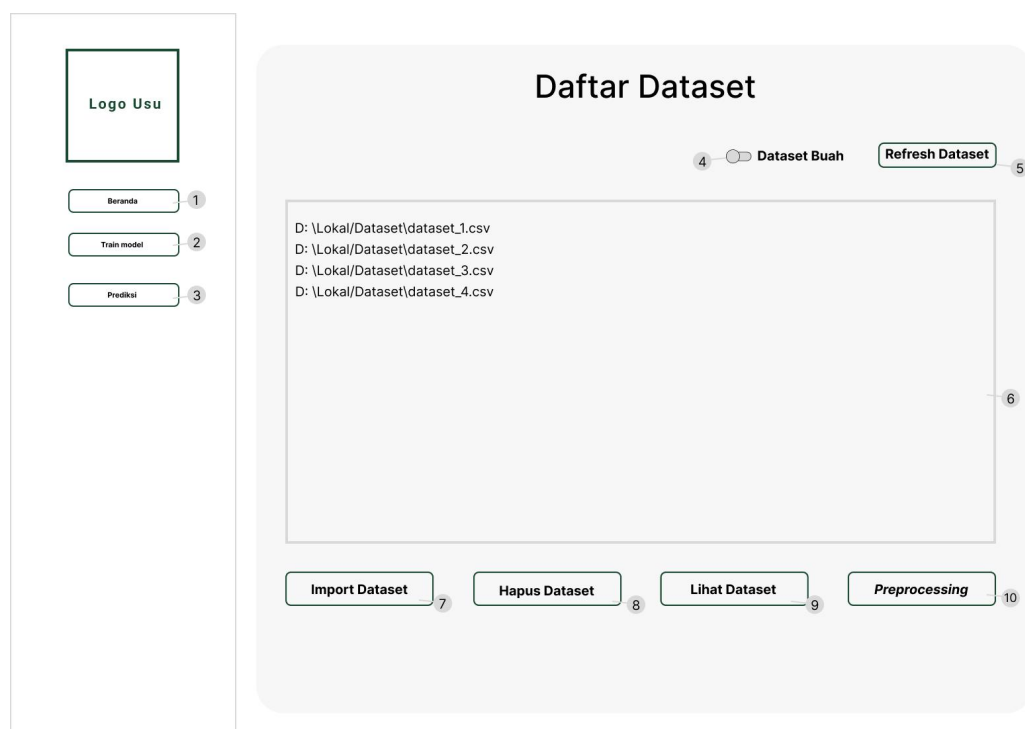
3.3 Perancangan Sistem

Perancangan sistem prediksi luas tanaman dan produksi buah tomat bertujuan untuk menyediakan platform yang memungkinkan pengguna untuk memperoleh estimasi luas tanaman dan buah yang diperlukan. Sistem ini akan memiliki antarmuka pengguna yang ramah pengguna, terdiri dari beberapa halaman yaitu, halaman

beranda, halaman *import data*, halaman *pre-processing*, halaman *training data* dan halaman prediksi.

3.3.1 Perancangan Halaman Beranda

Halaman beranda dari sistem prediksi luas tanaman dan produksi buah tomat bertujuan untuk memberikan pengguna informasi yang jelas dan ringkas tentang tujuan serta fitur-fitur yang tersedia dalam sistem. Desain antarmuka halaman beranda akan dioptimalkan untuk kejelasan informasi, dengan layout yang bersih dan mudah dipahami, sehingga pengguna dapat dengan cepat mendapatkan pemahaman tentang fitur-fitur yang ditawarkan oleh sistem.



Gambar 3. 6 Perancangan Halaman Beranda

Keterangan :

1. Tombol beranda, menampilkan halaman beranda.
2. Tombol *train model*, menampilkan halaman *train model*.
3. Tombol prediksi, menampilkan halaman prediksi.
4. Tombol dataset Buah, mengubah dataset tanaman menjadi dataset buah.
5. Tombol *refresh* Dataset, melakukan *refresh* pada dataset.
6. Daftar dataset, menampilkan daftar dataset yang ingin digunakan.
7. Tombol *Import*, menampilkan halaman *import data*.
8. Tombol hapus dataset, menghapus dataset.

9. Tombol lihat dataset, melihat dataset.
10. Tombol *pre-processing*, menampilkan halaman *pre-processing*

3.3.2 Perancangan Halaman Import Data

Halaman import data dalam perancangan sistem prediksi luas tanaman dan produksi buah tomat akan memungkinkan pengguna untuk mengimpor data yang diperlukan untuk melakukan prediksi. Tujuan dari halaman import data adalah untuk memberikan pengguna kemudahan dalam mengatur dan mempersiapkan data mereka untuk analisis lebih lanjut dalam sistem.



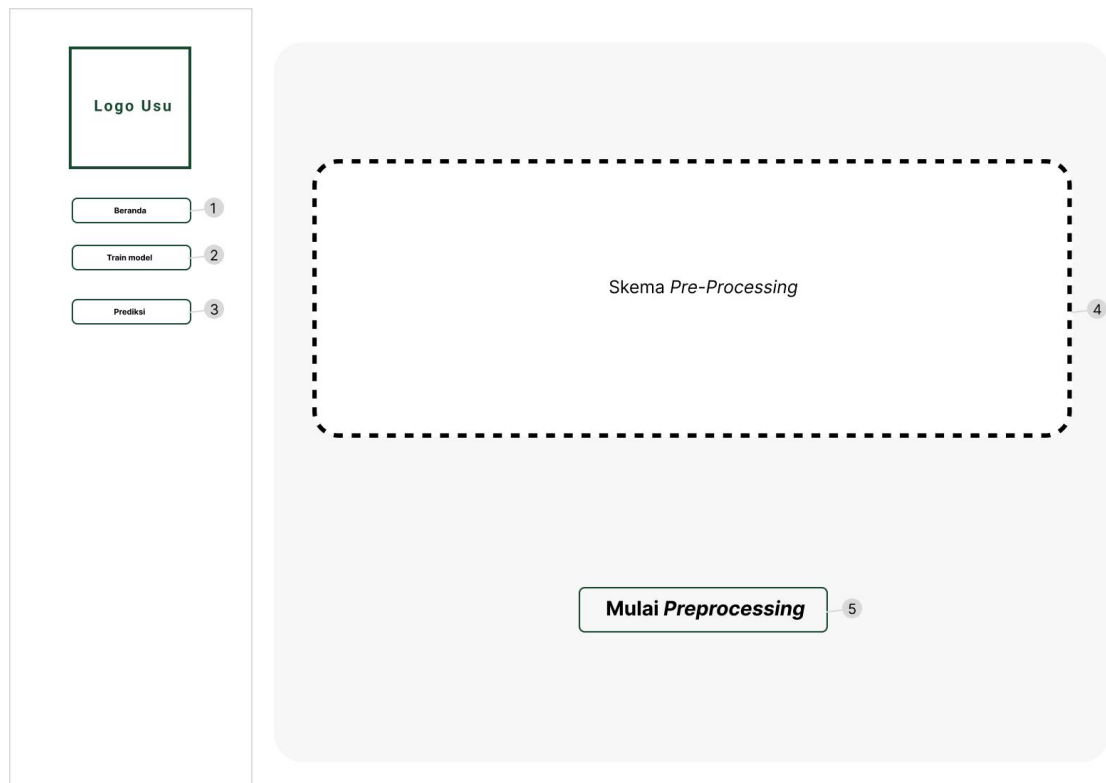
Gambar 3. 7 Perancangan Halaman Import

Keterangan :

1. Tombol beranda, menampilkan halaman beranda.
2. Tombol *train model*, menampilkan halaman *train model*.
3. Tombol prediksi, menampilkan halaman prediksi.
4. Skema *import* data, menampilkan skema *import* data.
5. Tombol tanggal awal, memilih tanggal awal dari data yang ingin diunduh.
6. Tombol tanggal akhir, memilih tanggal akhir dari data yang ingin diunduh.
7. Tombol *import* data, mengunduh data gambar dan data sensor.

3.3.3 Perancangan Halaman *Pre-Processing Data*

Pada halaman ini, pengguna akan memiliki akses untuk melakukan berbagai praproses terhadap data yang telah diunggah. Tujuan dari halaman ini adalah untuk memastikan bahwa data yang digunakan untuk prediksi luas tanaman dan produksi buah tomat adalah data yang bersih, terstruktur, dan siap untuk dianalisis dengan akurat oleh sistem.



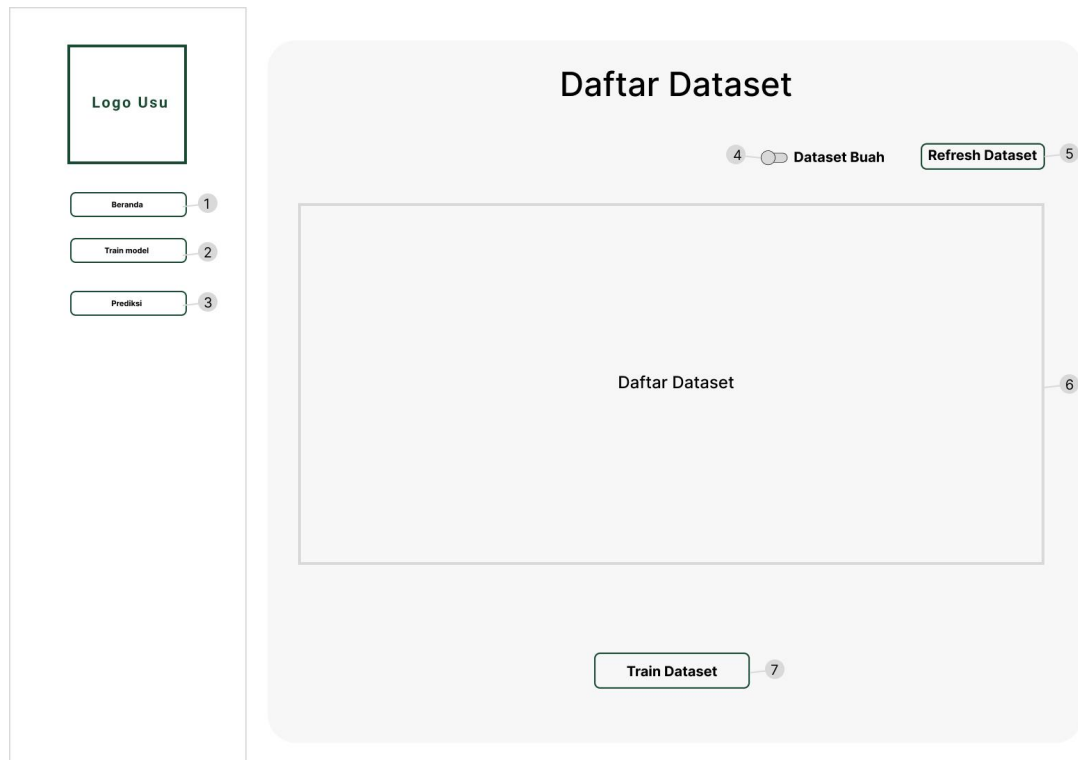
Gambar 3. 8 Perancangan Halaman *Pre-processing Data*

Keterangan :

1. Tombol beranda, menampilkan halaman beranda.
2. Tombol *train model*, menampilkan halaman *train model*.
3. Tombol prediksi, menampilkan halaman prediksi.
4. Skema *pre-processing*, menampilkan skema *pre-processing*.
5. Tombol mulai *pre-processing*, memulai proses *pre-processing*.

3.3.4 Perancangan Halaman *Train Model*

Pada halaman ini, pengguna akan memilih dataset yang ingin digunakan dan memulai proses pelatihan. Antarmuka halaman training model akan menyediakan opsi untuk memantau kemajuan pelatihan dan menampilkan metrik evaluasi kinerja model.



Gambar 3. 9 Perancangan Halaman Data *Train*

Keterangan :

1. Tombol beranda, menampilkan halaman beranda.
2. Tombol *train model*, menampilkan halaman *train model*.
3. Tombol prediksi, menampilkan halaman prediksi.
4. Tombol dataset Buah, mengubah dataset tanaman menjadi dataset buah.
5. Tombol *refresh* Dataset, melakukan *refresh* pada dataset.
6. Daftar dataset, menampilkan daftar dataset yang ingin digunakan.
7. Tombol *train* dataset, melakukan proses *train* dataset.

3.3.5 Perancangan Halaman Prediksi dan Hasil Prediksi

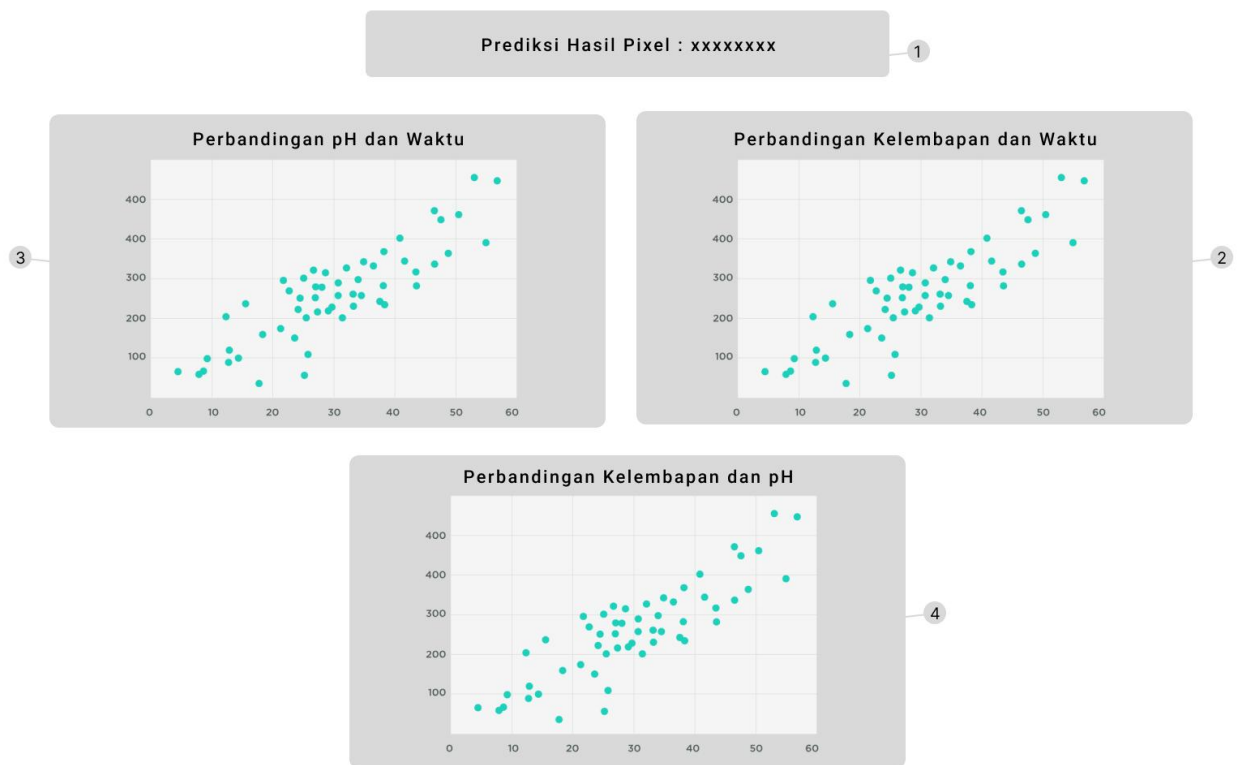
Pada halaman ini, pengguna akan diminta untuk memasukkan parameter-parameter tertentu, seperti hari, kelembapan tanah dan pH tanah. Setelah parameter dimasukkan, sistem akan menggunakan model yang telah dilatih sebelumnya untuk menghasilkan prediksi luas tanaman yang diperlukan serta luas buah tomat yang diharapkan berdasarkan parameter-parameter yang diberikan. Hasil prediksi kemudian akan ditampilkan kepada pengguna, mungkin dengan grafik atau laporan yang menjelaskan hasil secara detail.



Gambar 3. 10 Perancangan Halaman Prediksi

Keterangan :

1. Tombol beranda, menampilkan halaman beranda.
2. Tombol *train model*, menampilkan halaman *train model*.
3. Tombol prediksi, menampilkan halaman prediksi.
4. Skema skrema proses, menampilkan skema proses pengolahan data.
5. *Form input* hari, *input* data hari ke- yang ingin diprediksi.
6. *Form input* kelembapan tanah, *input* nilai kelembapan yang ingin diprediksi.
7. Tombol dataset Buah, mengubah dataset tanaman menjadi dataset buah.
8. *Form input* pH tanah, *input* nilai pH tanah yang ingin diprediksi.
9. Tombol lihat prediksi, melakukan proses prediksi data luas tanaman atau buah.



Gambar 3. 11 Perancangan Halaman Hasil Prediksi

Keterangan:

1. Prediksi hasil pixel, menampilkan hasil prediksi luas tanaman atau buah berdasarkan satuan pixel dan hasil konversi nilai pixel ke cm.
2. Perbandingan kelembapan dan waktu, menampilkan grafik perbandingan prediksi dengan kelembapan dan waktu.
3. Perbandingan pH dan waktu, menampilkan grafik perbandingan prediksi dengan pH dan waktu.
4. Perbandingan pH dan kelembapan, menampilkan grafik perbandingan prediksi dengan pH dan kelembapan

BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Spesifikasi Perangkat Keras dan Perangkat Lunak

Pengimplementasian metode *gradient boosting* untuk memprediksi luas tanaman memerlukan dukungan dari perangkat keras dan perangkat lunak yang sesuai. Bahasa pemrograman yang digunakan untuk implementasi sistem ini adalah Python, yang memiliki berbagai pustaka seperti scikit-learn untuk mengimplementasikan algoritma *gradient boosting*. Sistem aplikasi ini dikembangkan sebagai aplikasi berbasis desktop. Adapun spesifikasi perangkat keras yang digunakan untuk mendukung penelitian ini meliputi:

1. Komputer yang digunakan yaitu, Asus Tuf Gaming A15.
2. *Processor* AMD Ryzen 7 6800H with Radeon Graphics 3.20 GHz.
3. Kapasitas memori RAM 16 GB.
4. Kapasitas SSD 1.5 TB.

Adapun spesifikasi perangkat lunak yang digunakan untuk mendukung penelitian ini meliputi:

1. Sistem Operasi *Windows 11 Home Single Language*.
2. Python 3.11.0.
3. *Visual Studio Code* 1.89.1.
4. *customtkinter* 4.5.1
5. Figma

4.2 Impelentasi Sistem

4.2.1 Impelentasi *Data Collection*

Data yang digunakan penulis dalam mendukung penelitian ini yaitu data luas tanaman, luas buah, data kelembapan tanah, dan data pH. Data luas tanaman dan buah berasal dari citra tanaman dan buah tomat yang diambil empat kali sehari pada jam 09.00

WIB, 12.00 WIB, 15.00 WIB dan jam 17.00 WIB menggunakan kamera web cam logitech dan disimpan dalam format PNG (*Portable Network Graphics*) ke dalam cloud server. Adapun teknik pengambilan citra tanaman yaitu, mengambil foto tanaman dari samping tanaman dimana kamera webcam ditempel tepat di bagian sisi kiri lahan tanam. Data – data yang digunakan meliputi data luas tanaman dan buah, pH, dan kelembapan, yang kemudian digabungkan ke dalam sebuah file dengan format CSV (*Comma Separated Values*) untuk diolah dan dijadikan model untuk prediksi pertumbuhan tanaman. Tahap pertama merupakan proses pengumpulan data gambar dan data sensor yang ditangkap oleh kamera dan sensor yang terdapat pada robot pertanian. Nantinya data terbagi dua, yaitu data training dan data testing. Data training merupakan data yang telah dikumpulkan untuk dilatih dan data testing merupakan data yang digunakan untuk pengujian pada sistem. Adapun ukuran area tanam yang diterapkan pada penelitian kali ini yaitu 1,8 m x 4 m. Contoh citra tanaman dapat dilihat pada Gambar 4.1.



Gambar 4. 1 Implementasi Pengambilan Citra Tanaman Tomat

4.2.2 Implementasi Data Pre-processing

4.2.2.1 Implementasi Image Cropping

Citra akan dipotong sesuai dengan ukuran yang dibutuhkan untuk mempermudah proses pengolahan data. Adapun cara *cropping* citra menggunakan bantuan *library* scikit-image, dimana pemotongan citra dilakukan berdasarkan titik koordinat yang telah disesuaikan. Proses pemotongan citra dan hasil pemotongan dapat dilihat pada gambar 4.2 serta *pseudo code* dibawah ini:

```
# Deklarasi variabel gambar
deklarasikan gambar_tanaman
deklarasikan gambar_buah
```

```

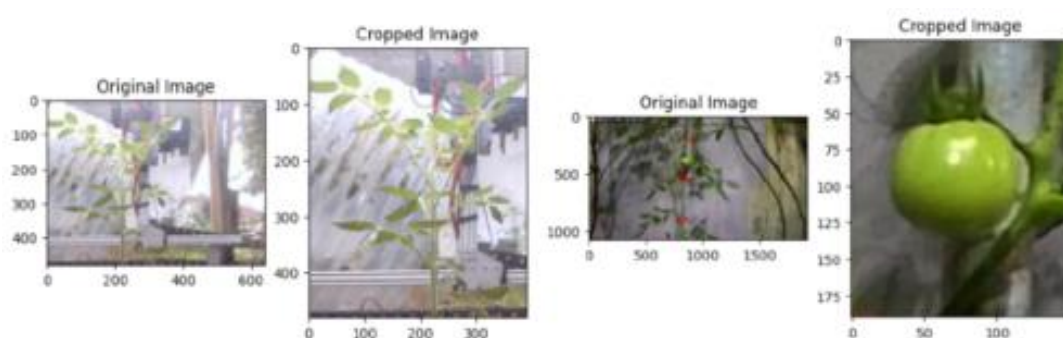
# Memotong gambar tanaman
# X : 0 sampai 189
# Y : semua baris
gambar_tanaman_potong = gambar_tanaman [ semua baris, kolom 0 sampai 189 ]

# Memotong gambar buah
# X : 770 sampai 919
# Y : 300 sampai 489
gambar_buah_potong = gambar_buah [baris 300 sampai 489, kolom 770 sampai 919]

```

Penjelasan :

Pada rumusan diatas terdapat enam variable yaitu *image*, *cropped_image*, *y1*, *y2*, *x1* dan *x2*. *Image* merupakan citra awal yang belum dilakukan pemotongan, sedangkan *cropped_image* merupakan hasil dari citra yang telah dipotong. *Y1* dan *y2* merupakan batas awal dan batas akhir pada koordinat *y* atau secara vertikal yang akan dipotong pada citra. *X1* dan *x2* merupakan batas awal dan batas akhir pada koordinat *x* atau secara horizontal yang dipotong pada citra. Pada penelitian ini citra dipotong berdasarkan nilai koordinat yang ada pada sumbu *x* atau secara horizontal saja yaitu 0 : 190 untuk tanaman . Adapun hasil proses cropping dapat dilihat pada gambar 4.2 yang mana *original image* merupakan gambar sebelum *cropping* dan *cropped_image* merupakan gambar setelah *cropping* .



Gambar 4. 2 Implementasi Sebelum dan Sesudah Cropping

4.2.2.2 Implementasi Gamma Correction

Setelah proses *cropping*, selanjutnya citra akan memasuki tahapan *gamma correction* yang dilakukan untuk memperbaiki kecerahan, kontras, dan reproduksi warna dalam citra tanaman. Dengan bantuan *library scikit-image* dan melakukan import *exposure*

untuk dapat menggunakan fungsi `exposure.adjust_gamma(image, 1)`, adapun *pseudo code* dari proses *gamma correction* sebagai berikut:

```
# Deklarasi variabel gambar tomat
deklarasikan gambar_tomat

# Koreksi gamma pada gambar tomat dengan faktor gamma 1
gambar_koreksi_gamma = exposure.adjust_gamma(gambar_tomat, 1)
```

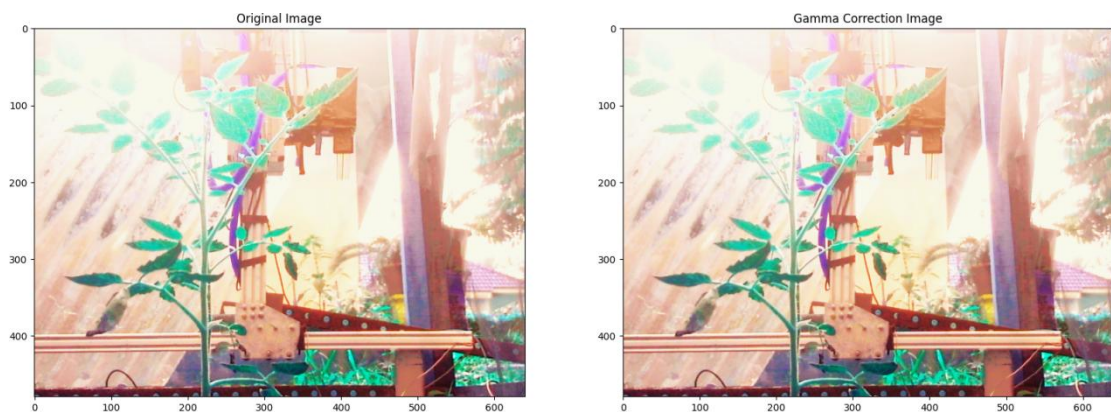
Keterangan :

`gamma_corrected_image` : Variabel yang menyimpan citra hasil dari proses *gamma correction*.

`exposure.adjust_gamma` : Fungsi *gamma correction* pada citra.

`tomato_image` : Variabel yang menyimpan citra asli.

`1` : Nilai gamma



Gambar 4.3 Implementasi Gamma Correction

Adapun gambar 4.3 merupakan hasil implementasi dari proses *gamma correction*, yang mana *original image* merupakan gambar sebelum *gamma correction* dan *gamma correction image* merupakan gambar setelah *gamma correction*.

4.2.2.3 Implementasi HSV Color Space

Pendeteksian warna yang sudah ditentukan pada citra agar dapat diseleksi bagian yang ingin diolah. Pada penelitian kali ini menggunakan bantuan *HSV color space*. adapun *pseudo code* dari proses *HSV color space* sebagai berikut:

```

# Deklarasi variabel gambar
deklarasikan gambar

# Konversi gambar dari RGB ke HSV
hsv = cv2.cvtColor(gambar, cv2.COLOR_RGB2HSV)

# Deklarasi batas atas dan batas bawah warna biru
batas_atas_biru = array([75, 255, 255])
batas_bawah_biru = array([24, 38, 49])

# Buat mask untuk warna biru
mask = cv2.inRange(hsv, batas_bawah_biru, batas_atas_biru)

# Terapkan mask pada gambar asli
hasil = cv2.bitwise_and(gambar, gambar, mask=mask)

```

Penjelasan :

1. `hsv = cv2.cvtColor(im, cv2.COLOR_RGB2HSV)`

Tahapan ini mengonversi citra `im` dari RGB ke HSV menggunakan fungsi `cv2.cvtColor` dari OpenCV. Citra dalam HSV memungkinkan pemrosesan yang lebih mudah untuk pemisahan warna berdasarkan hue, saturation, dan value.

2. `upper_blue = np.array([75,255,255]), lower_blue = np.array([24,38, 49])`

Baris ini mendefinisikan nilai batas atas dan batas bawah warna biru dalam HSV, tersimpan dalam bentuk array yang memiliki tiga nilai yang mewakili Hue (nilai warna), Saturation (nilai kejenuhan), dan Value (nilai kecerahan) yang merupakan representasi dari warna biru yang diinginkan.

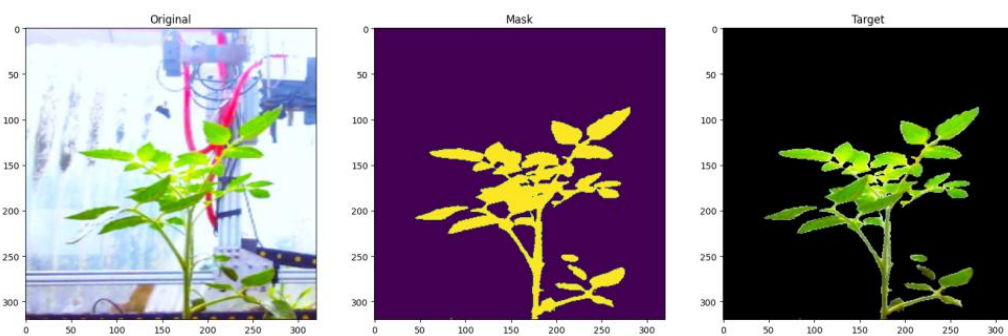
3. `mask = cv2.inRange(hsv, lower_blue, upper_blue)`

Fungsi `cv2.inRange` untuk membuat *masking* yang mencakup piksel-piksel dalam rentang warna biru yang ditentukan oleh `lower_blue` dan `upper_blue` di dalam citra `hsv`.

4. `res = cv2.bitwise_and(im, im, mask=mask)`

Fungsi `cv2.bitwise_and` untuk menerapkan *masking* yang telah dibuat sebelumnya (*mask*) pada citra asli *im*. Ini menghasilkan citra *res* yang hanya menampilkan bagian dari citra asli yang berada dalam rentang warna biru yang ditentukan.

Adapun hasil proses *CSV Color Space* dapat dilihat pada gambar 4.4, yang mana menampilkan hasil gambar *original*, gambar hasil proses *masking* dan gambar target yang merupakan hasil akhir dari proses *hsv color space* yang bertujuan untuk memisahkan objek dengan *background* gambar.



Gambar 4. 4 Implementasi HSV Color Space

4.2.3 Implementasi Featrue Extraction

Untuk ekstraksi fitur pada penelitian ini, penulis menggunakan *length feature extraction* atau fitur ekstraksi Panjang dengan menghitung jumlah pixel yang nilainya bukan 0 yang ada pada gambar. Kemudian penulis menggunakan fungsi `countNonZero` yang terdapat pada *library cv2*.

```
# Deklarasi variabel mask
deklarasikan mask
# Hitung jumlah piksel tidak nol dalam mask
jumlah_tidak_nol = cv2.countNonZero(mask)
```

Keterangan :

`jumlah_tidak_nol` : Hasil perhitungan jumlah piksel dari citra tanaman.

`Mask` : File citra tanaman

Data pixel luas tanaman atau buah yang telah dihitung, selanjutnya disimpan ke dalam dataset dalam bentuk csv. Adapun data hasil perhitungan pixel dapat dilihat pada Tabel 4.1.

Tabel 4. 1 Implementasi *Length Feature Extraction* Pada Citra

File	Luas Tanaman (Pixel)
farmbot_capture_77767479369311670205616.png	1256
farmbot_capture_34162274582721670206809.png	1044
farmbot_capture_50530266333871670216409.png	1221
farmbot_capture_5718867438321670217608.png	1193
farmbot_capture_49463918825671670218810.png	1107
farmbot_capture_70312715602461670220009.png	1389
farmbot_capture_50581019543901670221208.png	1246
farmbot_capture_17467465575681670222410.png	1210
farmbot_capture_26446800331021670223609.png	1222
farmbot_capture_32068010998641670224808.png	1131
.....
farmbot_capture_36693769687491672652410.png	32152
farmbot_capture_8983595683531672711210.png	24259
farmbot_capture_3538987191581672712408.png	23463
farmbot_capture_42380491168411672718408.png	23589
farmbot_capture_67898099577171672725608.png	32718
farmbot_capture_41903448479531672729208.png	31695
farmbot_capture_49124408969981672730408.png	42075
farmbot_capture_49200087626771672731610.png	41412

4.2.4 Implementasi Merging Dataset

Pada tahap ini, dilakukan penggabungan antara data pixel gambar dan data sensor kedalam sebuah file dataset dengan format *comma separated values* (CSV) yang dapat dilihat pada Tabel 4.2.

Tabel 4. 2 Hasil *Merging Dataset*

file	Hari ke-	kelembapan	pH	Luas Tanaman (Pixel)
farmbot_capture_77767479369311670205616.png	1	39.9	6.24	1256
farmbot_capture_70312715602461670220009.png	1	35.5	6.27	1389
farmbot_capture_50581019543901670221208.png	1	35.5	6.31	1246
farmbot_capture_17467465575681670222410.png	1	36.4	6.31	1210
farmbot_capture_83463104622701670304008.png	2	43.7	5.7	2103
farmbot_capture_8276159681251670305210.png	2	42.6	5.7	1973
farmbot_capture_87752558589461670306409.png	2	48.3	5.88	1961
farmbot_capture_3056111123701670307609.png	2	48.3	5.88	2069
.....
farmbot_capture_8397448303021672647608.png	29	44.1	7.07	29893
farmbot_capture_18736377343581672648808.png	29	41.9	6.42	36050
farmbot_capture_11853458213221672650008.png	29	41.9	5.87	29113
farmbot_capture_40020614282811672651209.png	29	41.9	6.58	32576
farmbot_capture_36693769687491672652410.png	29	41.9	7.73	32152
farmbot_capture_8983595683531672711210.png	29	47.9	6.35	24259
farmbot_capture_3538987191581672712408.png	30	47.9	7.05	23463
farmbot_capture_42380491168411672718408.png	30	47.9	7.75	23589
farmbot_capture_67898099577171672725608.png	30	45.3	6.78	32718

4.2.5 Implementasi Gradient Boosting

Saat semua parameter tercatat seperti waktu, pH tanah, kelembapan tanah dan luas tanaman, langkah selanjutnya adalah melakukan prediksi luas tanaman dengan menerapkan model yang telah disiapkan menggunakan *gradient boosting*.

4.2.5.1 Implementasi Inisialisasi Model

Berdasarkan data pada Tabel 4.2 Hasil *Merging Dataset*, dimana nilai pixel putih merupakan nilai target, maka kita akan melakukan tahapan inisialisasi model berdasarkan persamaan 2.1, dimana kita akan menggunakan rumus turunannya yang diterangkan pada persamaan 3.1, sebagai berikut:

$$F_0(x) = \frac{1}{n} \sum_{i=0}^n y_i$$

$$F_0(x) = \frac{1}{378} (1389 + 1193 + 1210 + \dots + 42075 + 32718 + 41412)$$

$$F_0(x) = \frac{1}{378} (6890426)$$

$$F_0(x) = 18228,64021$$

Adapun hasil perhitungan inisialisasi model dapat dilihat pada tabel 4.3 pada kolom $fo(x)$.

Tabel 4. 3 Implementasi Inisialisasi Model

File	Hari ke-	pH	kelembapan	Luas Tanaman (Pixel)	fo(x)
farmbot_capture_34162274582721670206809.png	1	6.06	39.9	1044	18228.64021
farmbot_capture_87752558589461670306409.png	2	6.36	48.3	1961	18228.64021
farmbot_capture_90818902810161670308809.png	2	6.37	48.3	2241	18228.64021
farmbot_capture_71600467389251670298021.png	2	6.07	45.5	2253	18228.64021
.....
farmbot_capture_8397448303021672647608.png	29	6.67	44.1	29893	18228.64021
farmbot_capture_18736377343581672648808.png	29	6.8	41.9	36050	18228.64021
farmbot_capture_36693769687491672652410.png	29	6.69	41.9	32152	18228.64021
farmbot_capture_41903448479531672729208.png	30	6.5	45.9	31695	18228.64021
farmbot_capture_67898099577171672725608.png	30	6.57	45.3	32718	18228.64021
farmbot_capture_49200087626771672731610.png	30	6.75	45.9	41412	18228.64021

4.2.5.2 Implementasi Residual Error

Pada tahap ini dilakukan perhitungan residual error yang berdasarkan persamaan 2.3 berikut:

$$\begin{aligned}
 \frac{\partial L}{\partial y} &= -(y_i - \gamma_i) = (Observed - Predicted) \\
 &= (1389 - 18228.6402116402) \\
 &= -16839.64
 \end{aligned}$$

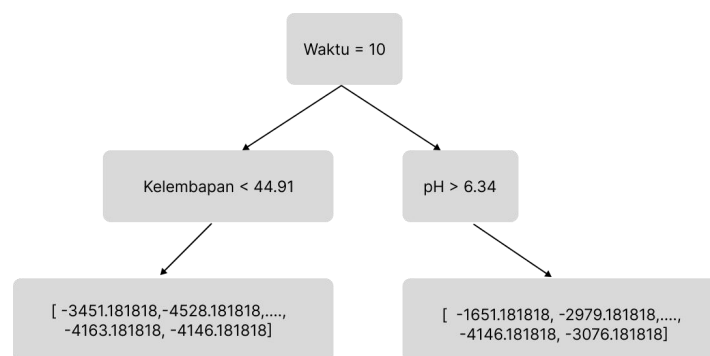
Adapun hasil implementasi dari residual *error* dapat dilihat pada tabel 4.4 yang terdapat pada kolom r_1 .

Tabel 4. 4 Implementasi *Residual Error*

File	Hari ke-	pH	kelembapan	Luas Tanaman (Pixel)	fo(x)	r_1
farmbot_capture_34162274582721670206809.png	1	6.06	39.9	1044	18228.64021	-17184.64021
farmbot_capture_87752558589461670306409.png	2	6.36	48.3	1961	18228.64021	-16267.64021
farmbot_capture_90818902810161670308809.png	2	6.37	48.3	2241	18228.64021	-15987.64021
farmbot_capture_71600467389251670298021.png	2	6.07	45.5	2253	18228.64021	-15975.64021
farmbot_capture_8397448303021672647608.png	29	6.67	44.1	29893	18228.64021	11664.35979
farmbot_capture_18736377343581672648808.png	29	6.8	41.9	36050	18228.64021	17821.35979
farmbot_capture_36693769687491672652410.png	29	6.69	41.9	32152	18228.64021	13923.35979
farmbot_capture_41903448479531672729208.png	30	6.5	45.9	31695	18228.64021	13466.35979
farmbot_capture_67898099577171672725608.png	30	6.57	45.3	32718	18228.64021	14489.35979
farmbot_capture_49200087626771672731610.png	30	6.75	45.9	41412	18228.64021	23183.35979

4.2.5.3 *Weak Learner*

Dalam tahap ini, weak learner dikembangkan sebagai model yang mampu memperkirakan kesalahan dari model sebelumnya. Model lemah ini umumnya berupa decision tree sederhana yang hanya memiliki 1 atau 2 cabang. Dengan kata lain, weak learner ini dirancang untuk fokus pada pola-pola sederhana dalam kesalahan prediksi yang dilakukan oleh model sebelumnya. Adapun contoh proses pembangunan *weak learner* dapat dilihat pada gambar 4.5.

**Gambar 4. 5** Contoh Proses Pembangunan *Weak Learner*

4.2.5.4 Update Model

Pada tahap ini, model baru yang dibuat dengan menggunakan decision tree yang memprediksi residu digabungkan dengan model sebelumnya untuk membentuk model yang lebih baik. Dengan menggabungkan model-model ini, kita memanfaatkan kekuatan masing-masing model untuk meningkatkan kinerja secara keseluruhan. Proses ini terus diulangi dalam beberapa iterasi, dengan setiap iterasi berfokus pada memperbaiki kesalahan prediksi model sebelumnya. Dengan demikian, setiap langkah tambahan dalam proses ini bertujuan untuk meningkatkan akurasi dan kinerja model secara keseluruhan. Adapun untuk contoh perhitungannya berdasarkan persamaan 2.4 sebagai berikut:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

$$F_1(1) = 18269,18182 + 0,1 * (-17048,18182)$$

$$= 9745,090909$$

Keterangan :

- F_m : Prediksi model gabungan setelah iterasi ke-m.
- F_{m-1} : Prediksi model gabungan sebelumnya setelah iterasi ke-(m-1).
- γ : *learning rate* yang berkisar antara 0 dan 1.
- m : Indeks iterasi
- $h_m(x)$: Prediksi dari pohon keputusan yang dibuat pada iterasi ke-m.

Adapun hasil dari implementasi update model dapat dilihat pada tabel 4.5 kolom *f1*.

Tabel 4. 5 Implementasi *Update Model*

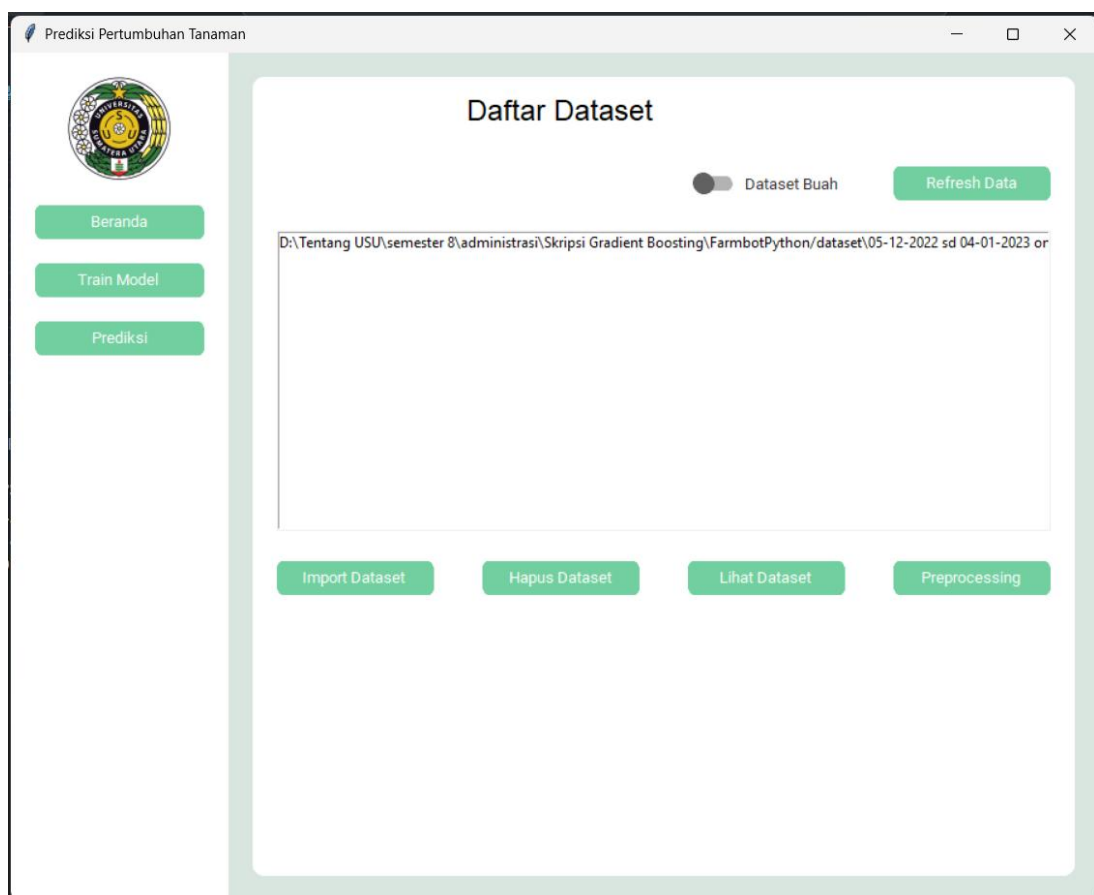
File	Hari ke-	pH	kelembapan	Luas Tanaman (Pixel)	fo(x)	r_1	<i>f1</i>
farmbot_capture_34162274582721670206809.png	1	6.06	39.9	1044	18228.64021	-17184.64021	16510.17619
farmbot_capture_87752558589461670306409.png	2	6.36	48.3	1961	18228.64021	-16267.64021	16601.87619
farmbot_capture_90818902810161670308809.png	2	6.37	48.3	2241	18228.64021	-15987.64021	16629.87619
farmbot_capture_71600467389251670298021.png	2	6.07	45.5	2253	18228.64021	-15975.64021	16631.07619
farmbot_capture_8397448303021672647608.png	29	6.67	44.1	29893	18228.64021	11664.35979	19395.07619
farmbot_capture_18736377343581672648808.png	29	6.8	41.9	36050	18228.64021	17821.35979	20010.77619
farmbot_capture_36693769687491672652410.png	29	6.69	41.9	32152	18228.64021	13923.35979	19620.97619
farmbot_capture_41903448479531672729208.png	30	6.5	45.9	31695	18228.64021	13466.35979	19575.27619
farmbot_capture_67898099577171672725608.png	30	6.57	45.3	32718	18228.64021	14489.35979	19677.57619
farmbot_capture_49200087626771672731610.png	30	6.75	45.9	41412	18228.64021	23183.35979	20546.97619

4.2.6 Implementasi Perancangan Sistem

Pada bagian ini, penulis akan menampilkan hasil implementasi dari perancangan tampilan antarmuka sistem yang telah dijelaskan sebelumnya.

4.2.6.1 Implementasi Halaman Beranda

Halaman beranda dari sistem prediksi luas tanaman dan produksi buah tomat bertujuan untuk memberikan pengguna informasi yang jelas dan ringkas tentang tujuan serta fitur-fitur yang tersedia dalam sistem.



Gambar 4. 6 Implementasi Halaman Beranda

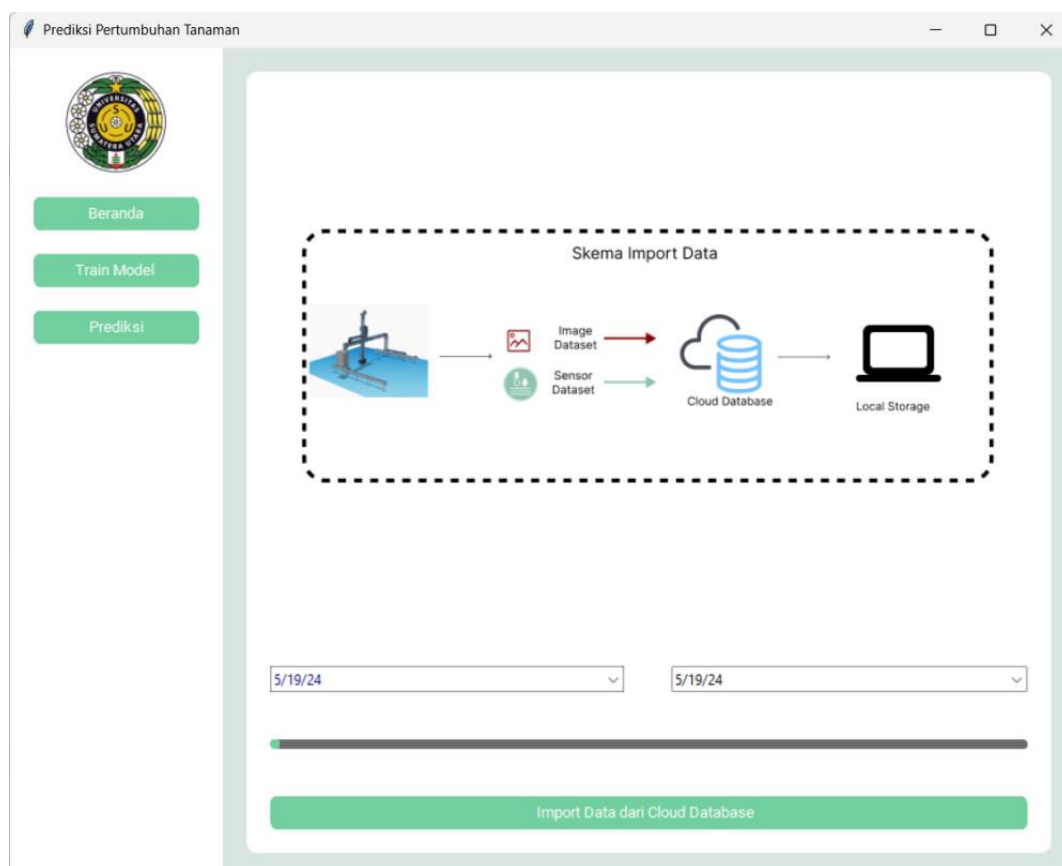
Keterangan :

1. Tombol beranda, menampilkan halaman beranda.
2. Tombol *train model*, menampilkan halaman *train model*.
3. Tombol prediksi, menampilkan halaman prediksi.
4. Tombol dataset Buah, mengubah dataset tanaman menjadi dataset buah.
5. Tombol *refresh* Dataset, melakukan *refresh* pada dataset.

6. Daftar dataset, menampilkan daftar dataset yang ingin digunakan.
7. Tombol *Import*, menampilkan halaman *import data*.
8. Tombol hapus dataset, menghapus dataset.
9. Tombol lihat dataset, melihat dataset
10. Tombol *pre-processing*, menampilkan halaman *pre-processing*

4.2.6.2 Implementasi Halaman *Import Data*

Halaman import data dalam implementasi sistem prediksi luas tanaman dan produksi buah tomat akan memungkinkan pengguna untuk mengimpor data yang diperlukan untuk melakukan prediksi. Tujuan dari halaman import data adalah untuk memberikan pengguna kemudahan dalam mengatur dan mempersiapkan data mereka untuk analisis lebih lanjut dalam sistem.



Gambar 4. 7 Implementasi Halaman *Import Data*

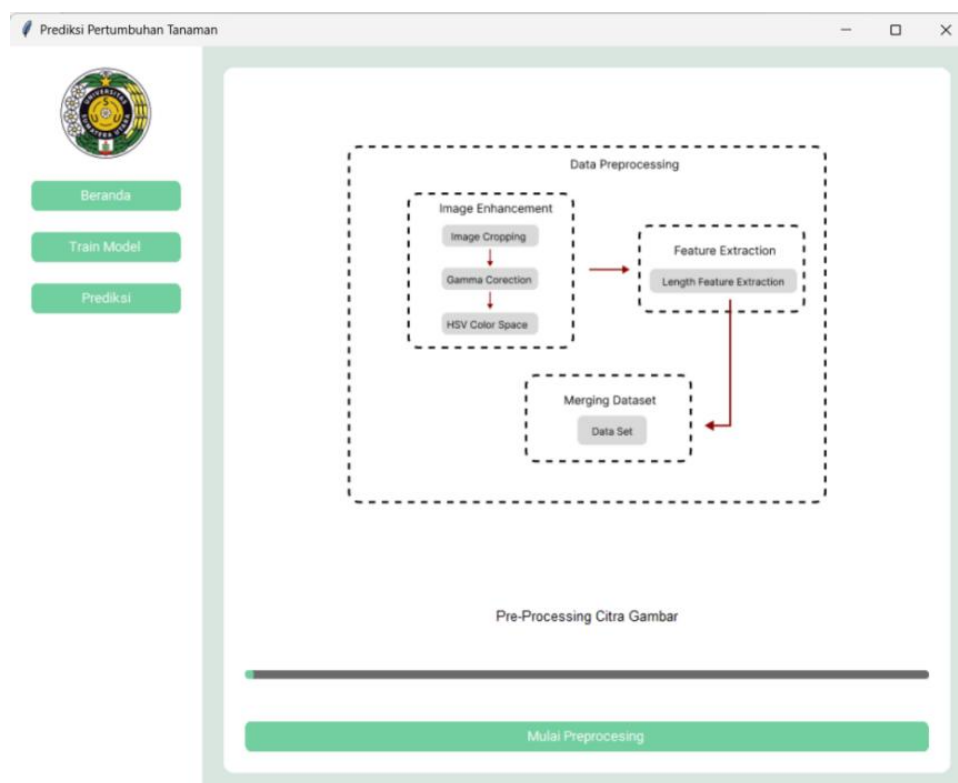
Keterangan :

1. Tombol beranda, menampilkan halaman beranda.
2. Tombol *train model*, menampilkan halaman *train model*.

3. Tombol prediksi, menampilkan halaman prediksi.
4. Skema *import* data, menampilkan skema *import* data.
5. Tombol tanggal awal, memilih tanggal awal dari data yang ingin diunduh.
6. Tombol tanggal akhir, memilih tanggal akhir dari data yang ingin diunduh.
7. Tombol *import* data, mengunduh data gambar dan data sensor.

4.2.6.3 Implementasi Halaman Pre-Processing Data

Pada halaman ini, pengguna akan memiliki akses untuk melakukan berbagai *pre-processing* terhadap data yang akan diproses. Tujuan dari halaman ini adalah untuk memastikan bahwa data yang digunakan untuk prediksi luas tanaman dan luas buah tomat adalah data yang bersih, terstruktur, dan siap untuk dianalisis dengan akurat oleh sistem.



Gambar 4. 8 Implementasi Halaman *Pre-processing*

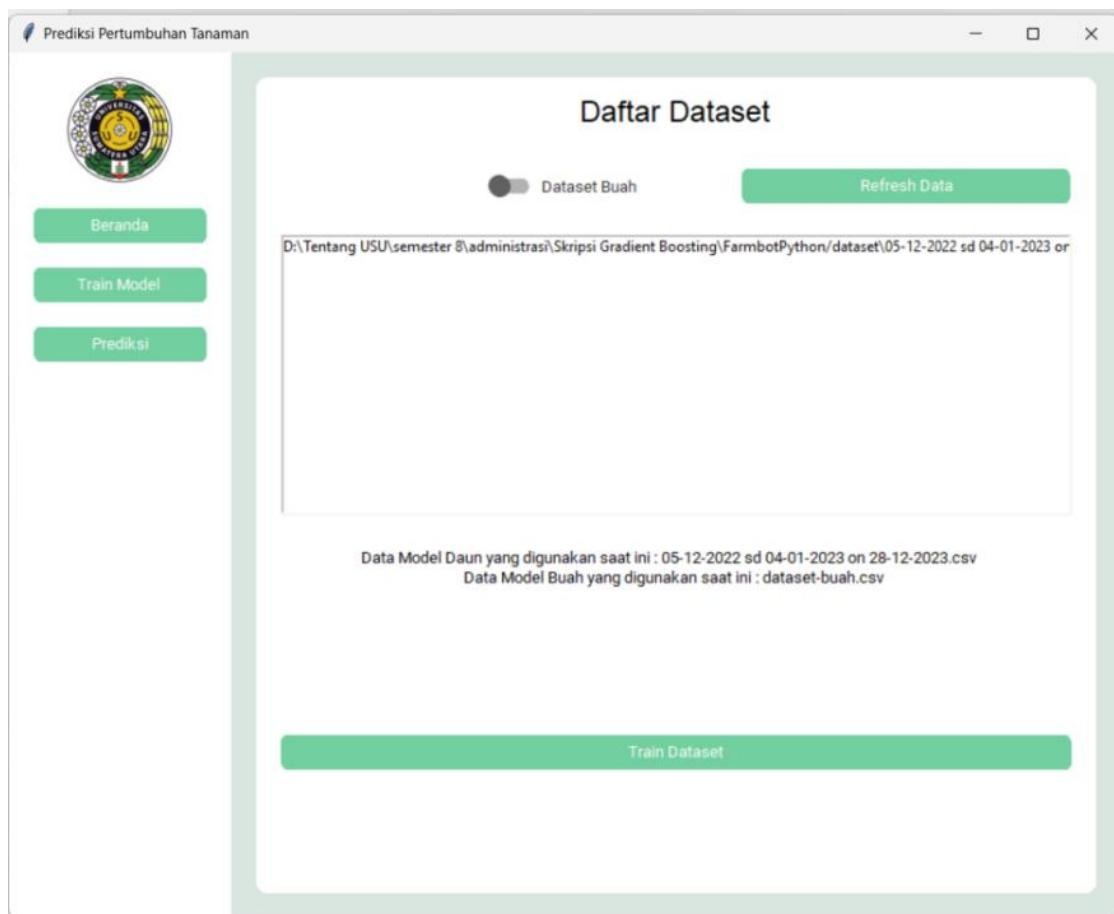
Keterangan :

1. Tombol beranda, menampilkan halaman beranda.
2. Tombol *train model*, menampilkan halaman *train model*.
3. Tombol prediksi, menampilkan halaman prediksi.

4. Skema *pre-processing*, menampilkan skema *pre-processing*.
5. Tombol mulai *pre-processing*, memulai proses *pre-processing*.

4.2.6.4 Implementasi Halaman Train Model

Pada halaman ini, pengguna akan memilih dataset yang ingin digunakan dan memulai proses pelatihan. Antarmuka halaman training model akan menyediakan opsi untuk memantau kemajuan pelatihan dan menampilkan metrik evaluasi kinerja model.



Gambar 4. 9 Implementasi Halaman *Data Train / Train Model*

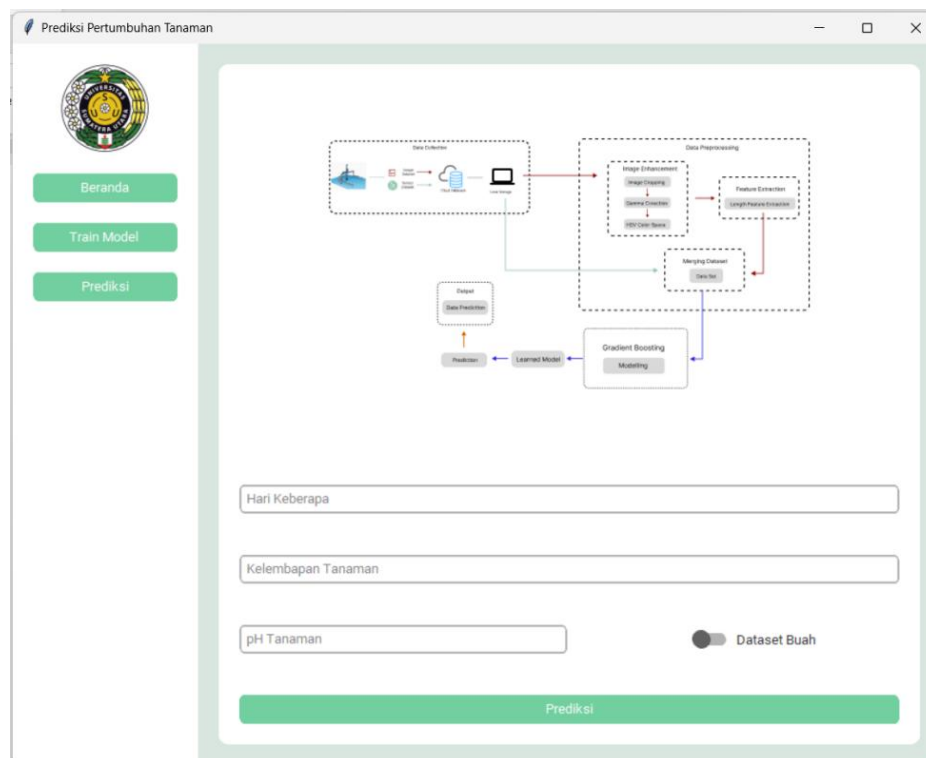
Keterangan :

1. Tombol beranda, menampilkan halaman beranda.
2. Tombol *train model*, menampilkan halaman *train model*.
3. Tombol prediksi, menampilkan halaman prediksi.
4. Tombol dataset Buah, mengubah dataset tanaman menjadi dataset buah.
5. Tombol *refresh* Dataset, melakukan *refresh* pada dataset.
6. Daftar dataset, menampilkan daftar dataset yang ingin digunakan.

7. Tombol *train* dataset, melakukan proses *train* dataset.

4.2.6.5 Implementasi Halaman Prediksi dan Hasil Prediksi

Pada halaman ini, pengguna dapat melakukan prediksi luas tanaman dengan memasukkan data hari, nilai kelembapan tanah serta pH tanah. Setelah parameter dimasukkan, sistem akan menggunakan model yang telah dilatih sebelumnya untuk menghasilkan prediksi luas tanaman yang diperlukan serta luas buah tomat yang diharapkan berdasarkan parameter-parameter yang diberikan. Adapun masukkan data untuk pH tanah sendiri dimulai dari 5,8 sampai dengan 6,8 dan untuk masukkan data kelembapan tanah dimulai dari 32,8 % hingga 61,5 %. Adapun untuk halaman prediksi dapat dilihat pada Gambar 4.10.



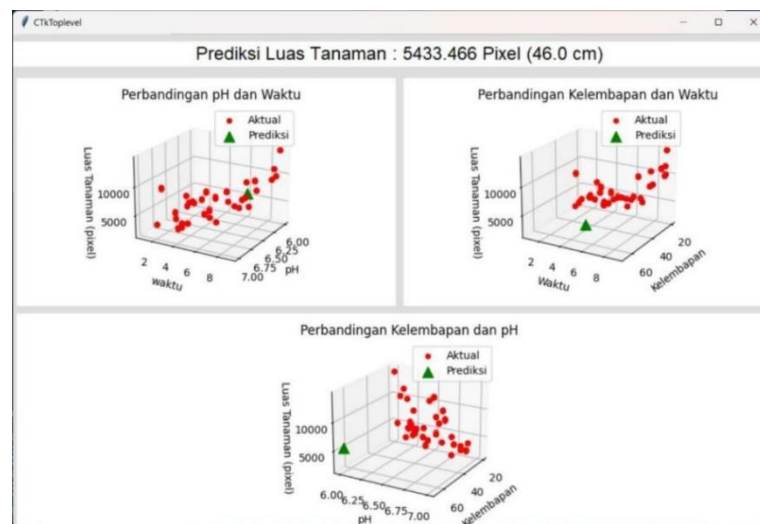
Gambar 4. 10 Implementasi Halaman Prediksi

Keterangan :

1. Tombol beranda, menampilkan halaman beranda.
2. Tombol *train model*, menampilkan halaman *train model*.
3. Tombol prediksi, menampilkan halaman prediksi.
4. Skema proses, menampilkan skema proses pengolahan data.
5. *Form input* hari, *input* data hari ke- yang ingin diprediksi.

6. *Form input* kelembapan tanah, *input* nilai kelembapan yang ingin diprediksi.
7. Tombol dataset Buah, mengubah dataset tanaman menjadi dataset buah.
8. *Form input* pH tanah, *input* nilai pH tanah yang ingin diprediksi.
9. Tombol lihat prediksi, melakukan proses prediksi data luas tanaman atau buah

Adapun implementasi halaman hasil prediksi dapat dilihat pada Gambar 4.11.



Gambar 4. 11 Implementasi Halaman Hasil Prediksi

Keterangan:

1. Prediksi hasil pixel, menampilkan hasil prediksi luas tanaman atau buah berdasarkan satuan pixel dan hasil konversi nilai pixel ke cm.
2. Perbandingan kelembapan dan waktu, menampilkan grafik perbandingan prediksi dengan kelembapan dan waktu.
3. Perbandingan pH dan waktu, menampilkan grafik perbandingan prediksi dengan pH dan waktu.
4. Perbandingan pH dan kelembapan, menampilkan grafik perbandingan prediksi dengan pH dan kelembapan

4.3 Pengujian Sistem

Setelah tahap implementasi selesai, langkah berikutnya adalah melakukan evaluasi terhadap kinerja model prediksi berbasis *Gradient Boosting* untuk melakukan prediksi luas tanaman. Adapun hasil akhir dari pengujian sistem ini yaitu, nilai prediksi luas tanaman dalam satuan pixel. Pengujian dilakukan dengan memberikan nilai input pada

kolom waktu, kelembapan dan pH sebanyak 95 data uji. Adapun hasil pengujian data dapat dilihat pada Tabel 4.6.

Tabel 4. 6 Hasil Pengujian Data Luas Tanaman

Hari ke-	kelembapan	pH	Luas Tanaman (Pixel)	Luas Tanaman (cm)
3	51.4	6.2	3832	32.44
5	45.9	6.24	6093	51.59
3	35.6	5.9	3262	27.62
23	41	6.76	31446	266.24
15	50.1	5.42	16194	137.11
18	39	7.73	26599	225.21
12	53.7	6.78	12370	104.73
1	43.3	6.25	1131	9.58
7	50.2	6.52	9722	82.31
5	35.5	6	5663	47.95
5	45.7	7	5791	49.03
10	57.2	7.8	15193	128.63
21	38.9	5.84	26657	225.70
9	43.4	7.15	12047	102.00
29	49	7.58	20294	171.82
18	42.5	5.27	28564	241.84
8	52.4	5.59	11426	96.74
6	44.9	6.36	7281	61.65
26	42.2	7.37	26000	220.13
5	42.4	6.82	6041	51.15
1	39.9	6.24	1256	10.63

Tabel 4. 6 Hasil Pengujian Data Luas Tanaman (Lanjutan)

Hari ke-	kelembapan	pH	Luas Tanaman (Pixel)	Luas Tanaman (cm)
30	47.9	7.75	23589	199.72
1	38.6	6.25	1096	9.28
8	48.7	5.98	11650	98.64
5	46.6	5.06	6408	54.25
19	41	5.29	26268	222.40
25	43.1	5.53	30553	258.68
5	45	6.5	4993	42.27
2	37.9	6.2	2534	21.45
29	47.9	6.35	24259	205.39
6	41.2	6.65	7975	67.52
11	58.1	7.16	12315	104.27
6	45.1	6.83	7635	64.64
24	39.3	6.88	31778	269.05
2	45.5	5.6	2070	17.53
3	47.8	6.2	3087	26.14
17	37.4	6.65	24582	208.13
30	47.9	7.05	23463	198.65
8	45.7	6.56	11862	100.43
23	39.6	5.2	29511	249.86
22	40.6	6.88	32320	273.64
21	40.3	5.44	26443	223.88
22	38.7	6.82	29042	245.89
26	42.6	5.48	28587	242.04

Tabel 4. 6 Hasil Pengujian Data Luas Tanaman (Lanjutan)

Hari ke-	kelembapan	pH	Luas Tanaman (Pixel)	Luas Tanaman (cm)
20	47.7	5.43	31608	267.62
12	49	5.07	14990	126.92
18	36.7	7.17	28678	242.81
5	39	6.2	6098	51.63
5	46	6.26	6448	54.59
16	49.2	6.14	30040	254.34
25	43.1	6.77	30833	261.05
17	35	5.98	24545	207.82
16	49.2	7.4	33509	283.71
26	42.2	6.63	25041	212.01
16	34.9	6.61	24029	203.45
23	37.3	7.37	32968	279.13
5	45.7	7.19	6116	51.78
27	45.6	5.73	31841	269.59
4	46.3	5.6	3967	33.59
28	42.9	6.83	34734	294.08
7	46.5	6.28	9700	82.13
18	36.7	5.09	27787	235.26
29	41.9	6.58	32576	275.81
9	43.4	7.56	12580	106.51
5	45.7	7.83	5978	50.61
23	39.7	6.33	27623	233.88
27	43	5.68	25907	219.35

Tabel 4. 6 Hasil Pengujian Data Luas Tanaman (Lanjutan)

Hari ke-	kelembapan	pH	Luas Tanaman (Pixel)	Luas Tanaman (cm)
28	44.2	6.6	25886	219.17
19	39.1	7.53	27212	230.40
26	44.6	6.04	27903	236.25
25	41.8	6.31	26988	228.50
2	46.2	4.93	2074	17.56
19	41.1	6.01	29311	248.17
23	36.4	5.33	26388	223.42
2	45.5	5.79	1877	15.89
18	36.7	7.52	27058	229.09
19	38.3	6.21	28413	240.56
4	45	5.8	4744	40.17
12	50.4	6.96	10146	85.90
7	41.2	7.13	8740	74.00
27	45.3	7.05	18854	159.63
25	46.8	5.7	27628	233.92
22	36.5	6.4	32613	276.12
24	39.6	5.26	27055	229.07
2	48.3	5.8	2086	17.66
20	37.8	5.12	32016	271.07
22	38.8	5.71	32380	274.15
17	42.5	5.62	22513	190.61
3	44.5	6.31	3183	26.95
10	54	5.7	13604	115.18

Tabel 4. 6 Hasil Pengujian Data Luas Tanaman (Lanjutan)

Hari ke-	kelembapan	pH	Luas Tanaman (Pixel)	Luas Tanaman (cm)
20	42.4	5.96	28314	239.73
7	51.4	6.82	9960	84.33
29	44.1	5.23	27260	230.80
13	47.6	6.72	21281	180.18
12	50.4	7.62	18932	160.29

Setelah memperoleh hasil prediksi tanaman, langkah selanjutnya adalah melakukan validasi model dengan menguji akurasi. Proses ini menerapkan metode *Mean Square Error* (MSE), *Mean Absolute Error* (MAE) dan koefisien determinasi (R^2).

1. Perhitungan *Mean Square Error* (MSE)

a. Luas Tanaman

$$MSE = \frac{1}{n} \sum_{i=0}^n (y_i - \gamma_i)^2$$

$$\begin{aligned}
 MSE &= \frac{1}{95} ((3165.74 - 3832)^2 + (6377.59 - 6093)^2 + \dots + \\
 &\quad (22377.55 - 21281)^2 + (15846.40 - 18932)^2) \\
 &= 14182473.63
 \end{aligned}$$

b. Besar Buah

$$MSE = \frac{1}{n} \sum_{i=0}^n (y_i - \gamma_i)^2$$

$$\begin{aligned}
 MSE &= \frac{1}{99} ((65824.61 - 66526)^2 + (2706.59 - 1526)^2 + \dots + \\
 &\quad (4124.18 - 3434)^2 + (63608.18 - 65888)^2) \\
 &= 4958183.438
 \end{aligned}$$

2. Perhitungan *Mean Absolute Error* (MAE)

a. Luas Tanaman

$$MAE = \frac{1}{n} \sum_{i=0}^n |y_i - \gamma_i|$$

$$\begin{aligned}
 MAE &= \frac{1}{95} (|3165.74 - 3832| + |6377.59 - 6093| + \dots + \\
 &\quad |22377.55 - 21281| + |15846.40 - 18932|) \\
 &= 2303.666085
 \end{aligned}$$

b. Besar Buah

$$MAE = \frac{1}{n} \sum_{i=0}^n |y_i - \gamma_i|$$

$$MAE = \frac{1}{99} (|65824.61 - 66526| + |2706.59 - 1526| + \dots + |4124.18 - 3434| + |63608.18 - 65888|)$$

$$= 1517.674296$$

3. Perhitungan Koefisien Determinasi (R^2)

a. Luas Tanaman

$$R^2 = 1 - \frac{\sum_{i=0}^n (y_i - \gamma_i)^2}{\sum_{i=0}^n (y_i - \bar{y})^2}$$

$$R^2 = 1 - \frac{((3165.74 - 3832)^2 + (6377.59 - 6093)^2 + \dots + (4124.18 - 3434)^2 + (63608.18 - 65888)^2)}{((3165.74 - 18843.09)^2 + (6377.59 - 18843.09)^2 + \dots + (4124.18 - 18843.09)^2 + (63608.18 - 18843.09)^2)}$$

$$R^2 = 1 - \frac{1347334995}{11136289539}$$

$$R^2 = 0.879014012$$

b. Besar Buah

$$R^2 = 1 - \frac{\sum_{i=0}^n (y_i - \gamma_i)^2}{\sum_{i=0}^n (y_i - \bar{y})^2}$$

$$R^2 = 1 - \frac{((65824.61 - 66526)^2 + (2706.59 - 1526)^2 + \dots + (4124.18 - 3434)^2 + (63608.18 - 65888)^2)}{((65824.61 - 16362.19005)^2 + (2706.59 - 16362.19005)^2 + \dots + (4124.18 - 16362.19005)^2 + (63608.18 - 16362.19005)^2)}$$

$$R^2 = 1 - \frac{490860160.4}{58272824121}$$

$$R^2 = 0.991576517$$

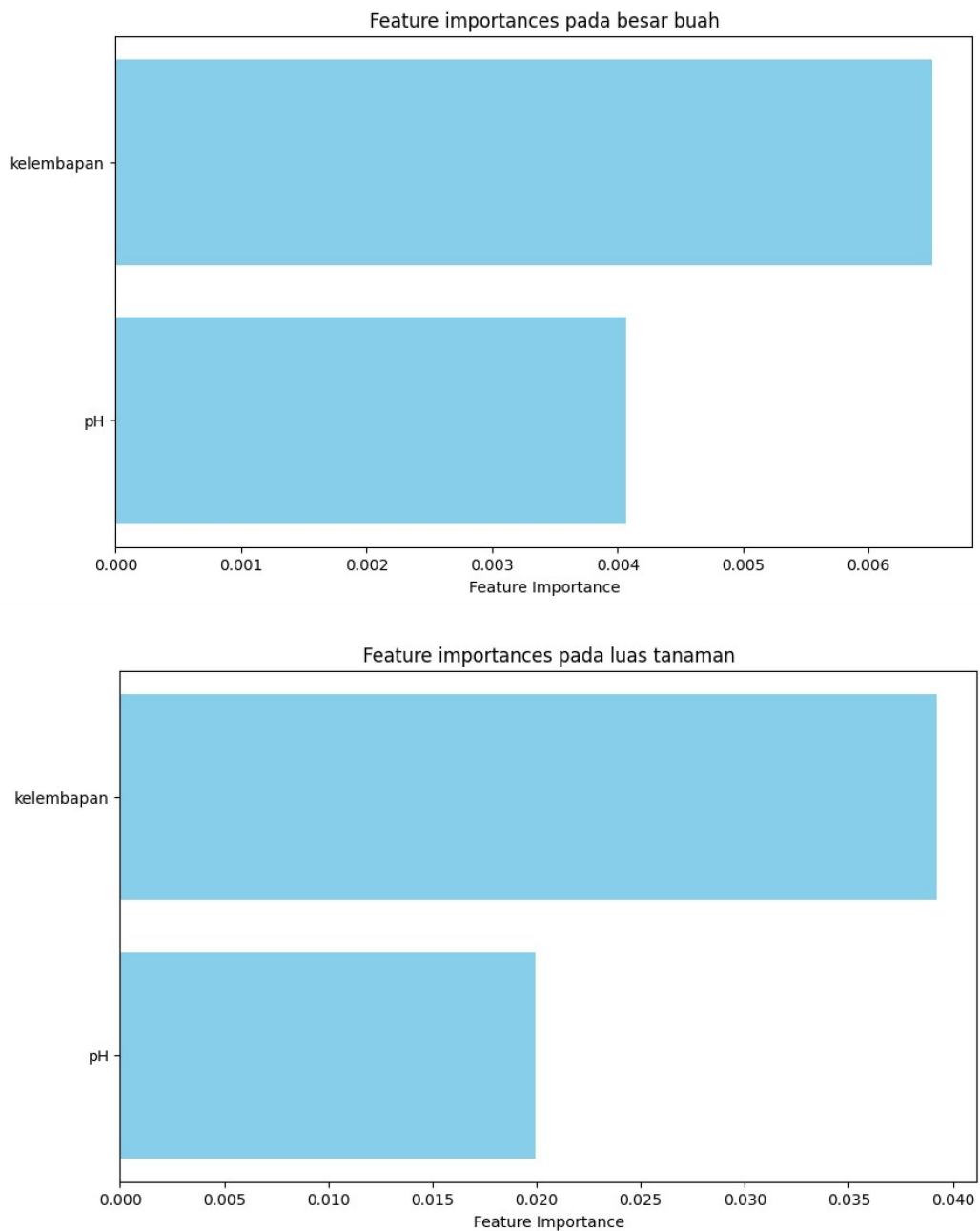
Adapun hasil validasi akurasi untuk masing-masing model disajikan dalam Tabel 4.7.

Tabel 4. 7 Hasil Validasi Model

Model	Mean Square Error (MSE)	Mean Absolute Error (MAE)	Koefisien Determinasi (R^2)
Tanaman	14182473.63	2303.666085	0.879014012
Buah	4958183.438	1517.674296	0.991576517

Berdasarkan Tabel 4.7 hasil *Mean Square Error* (MSE) untuk model tanaman sebesar 14182473.63 dan model buah sebesar 4958183.438, *Mean Absolute Error* (MAE) untuk model tanaman sebesar 2303.666085 dan untuk model buah sebesar 1517.674296, Koefisien Determinasi (R^2) pada model tanaman sebesar 0.8790 dengan akurasi sebesar 88,04% dan untuk model buah sebesar 0.9915 dengan akurasi sebesar

92%, yang mana menunjukkan bahwa metode *gradient boosting* mampu melakukan prediksi luas tanaman dan buah.



Gambar 4. 12 *Featur Importance* pada besar buah dan luas tanaman

Adapun penjelasan untuk gambar 4.12 yaitu hasil *feature importance* yang disediakan oleh algoritma *gradient boosting* untuk mengetahui parameter mana yang memiliki pengaruh yang lebih besar dari data yang ada. Dalam penelitian ini, parameter yang memiliki pengaruh lebih besar terhadap pertumbuhan luas tanaman dan besar buah yaitu kelembapan.

Penelitian ini memiliki beberapa kekurangan, di antaranya adalah pengaturan jarak antara kamera dan tanaman yang belum optimal serta kualitas kamera yang belum memadai. Gambar yang diambil dengan jarak yang tidak tepat dapat menyebabkan detail penting dari tanaman tidak terekam dengan baik serta mengakibatkan objek yang diteliti keluar dari *frame* kamera dalam jangka waktu yang lebih singkat. Selain itu, jarak yang tidak optimal bisa mengakibatkan distorsi atau kesalahan dalam pengukuran parameter tanaman, seperti ukuran daun, warna, atau tekstur yang penting untuk penelitian. Di samping itu, kualitas kamera yang digunakan dalam penelitian ini belum optimal. Kamera dengan resolusi rendah atau kemampuan fokus yang terbatas dapat mengurangi kualitas gambar yang dihasilkan. Keterbatasan dalam kemampuan kamera untuk menangkap detail halus dan warna yang akurat menjadi hambatan dalam mengidentifikasi parameter penting secara jelas. Oleh karena itu, penelitian ini memerlukan penyesuaian lebih lanjut dalam pengaturan jarak antara kamera dan tanaman serta peningkatan kualitas kamera untuk memastikan data gambar yang diperoleh lebih akurat dan konsisten, mendukung hasil penelitian yang lebih valid. Adapun contoh gambar tanaman yang keluar dari *frame* kamera dan kualitas yang kurang optimal sehingga tidak dapat dilakukan pemisahan antara objek dan *background* gambar dapat dilihat pada gambar 4.13.



Gambar 4. 13 Gambar Tanaman Lewat *Frame* Kamera

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Adapun kesimpulan yang didapatkan berdasarkan prediksi pertumbuhan tanaman tomat berdasarkan pengambilan citra oleh robot menggunakan *gradient boosting* sebagai berikut:

1. Metode *gradient boosting* yang diterapkan dalam proses prediksi luas tanaman sebesar 88,04% dan prediksi luas buah sebesar 92%.
2. Metode *HSV Color Space* merupakan metode yang cukup akurat dalam proses pemisahan objek dengan *background* berdasarkan nilai *hue*, *saturation* dan *value*.
3. Kelembapan tanah memiliki pengaruh yang cukup penting terhadap luas tanaman dan besar buah dibandingkan dengan pH tanah.
4. Menjaga kestabilan kelembapan dan pH tanah merupakan suatu tindakan yang dapat menjaga keseimbangan pertumbuhan tanaman.

5.2 Saran

Berikut adalah saran-saran yang dapat bermanfaat untuk pengembangan penelitian berikutnya:

1. Penambahan parameter lainnya untuk membuat model lebih baik lagi seperti, intensitas cahaya, suhu maupun NPK.
2. Penggunaan kamera dengan kualitas yang lebih baik untuk mendapatkan citra gambar yang lebih maksimal.
3. Pengaturan jarak yang lebih optimal antara kamera dan tanaman untuk mendapatkan data gambar yang lebih baik lagi.
4. Melakukan tes dengan objek tanaman yang berbeda.
5. Pengumpulan data bisa lebih lama lagi, sehingga data yang dikumpulkan untuk menjadi model dataset lebih maksimal
6. Menambahkan fitur yang dapat memprediksi luas tanaman secara real-time

DAFTAR PUSTAKA

- Brady, N. C., & Weil, R. R. (2008). *The Nature and Properties of Soils (14th ed.)*. Upper Saddle River, NJ: Pearson Prentice Hall.
- Deng, L., & Yu, D. (2014). Deep Learning: Methods and Applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387.
- Djumali, & Mulyaningsih, S. (2014). Pengaruh Kelembaban Tanah terhadap Karakter Agronomi, Hasil Rajangan Kering dan Kadar Nikotin Tembakau (*Nicotiana tabacum* L; Solanaceae) Temanggung pada Tiga Jenis Tanah. *Balai Penelitian Tanaman Pemanis dan Serat. Berita Biologi*, 13(1).
- Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *The Annals of Statistics*, 29(5) : 1189–1232.
- Kristanto, A. (2004). *Jaringan Syaraf Tiruan (Konsep Dasar, Algoritma, dan Aplikasi)*. Yogyakarta: Gaya Media.
- Merkel, A. (den 14 Februari 2014). *Speech by Federal Chancellor Angela Merkel to the OECD Conference*. Hämtat från bundesregierung.de: <https://www.bundesregierung.de/breg-en/service/archive/archive/speech-by-federal-chancellor-angela-merkel-to-the-oecd-conference-477432>
- Muñoz-Carpena, R., Dukes, M. D., Li, Y. C., & Klassen, W. (2005). Field Comparison of Tensiometer and Granular Matrix Sensor Automatic Drip Irrigation on Tomato. *HortTechnology*, 15(3).
- Patane, C., & Cosentino, S. (2010). Effects of soil water deficit on yield and quality of processing tomato under a Mediterranean climate. *Agricultural Water Management*, 97(1) : 131-138.
- Pohan, S., Warsito, B., & Suryono, S. (2020). Backpropagation artificial neural network for prediction plant. *Journal of Physics: Conference Series* (s. 1524). Central Java: IOP Publishing.
- Sakurai, S., Uchiyama, H., Shimada, A., & Taniguchi, R.-i. (2019). Plant Growth Prediction using Convolutional LSTM. *VISIGRAPP 2019 - Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications* (ss. 105-113). Prague: SciTePress.
- Schueller, J. K., Burks, T. F., & Sivaraman, B. (2006). Using Modern Robot Synthesis and Analysis Tools for the Design of Agricultural Manipulators. *Agricultural Engineering International: the CIGR Ejournal. Invited Overview*, 8(2).
- Sudana, I. M., Purnawirawan, O., & Arief, U. M. (2017). Prediction System Of Hydroponic Plant Growth and Development Using Algorithm Fuzzy Mamdani

Method. *ENGINEERING INTERNATIONAL CONFERENCE (EIC) 2016: Proceedings of the 5th International Conference on Education, Concept, and Application of Green Technology* (ss. Vol 1818, issue 1). Semarang: AIP Conference Proceedings.

Tremblay, N., Fallon, E., & Ziadi, N. (2011). Sensing of Crop Nitrogen Status: Opportunities, Tools, Limitations, and Supporting Information Requirements. *HortTechnology*, 21(30) : 274–281 .

Widiana, S. A., Suryono, S., & Warsito, B. (2020). Plant Seeds Growth Prediction on Greenhouse Using Adaptive Neuro Fuzzy Inference System (ANFIS) Method. *The 5th International Conference on Energy, Environmental and Information System (ICENIS 2020)* (s. 202(8)). E3S Web of Conferences.

Xia, C., Wang, L., Chung, B.-K., & Lee, J.-M. (2015). In Situ 3D Segmentation of Individual Plant Leaves Using a RGB-D Camera for Agricultural Automation. *sensors*, 15, 20463-20479.



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
UNIVERSITAS SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

Jalan Universitas No. 9A Gedung A, Kampus USU Medan 20155, Telepon: (061) 821007
Laman: <http://Fasilkomti.usu.ac.id>

KEPUTUSAN
DEKAN FAKULTAS ILMU KOMPUTER
DAN TEKNOLOGI INFORMASI
NOMOR :2360/UN5.2.14.D/SK/SPB/2024
DEKAN FAKULTAS ILMU KOMPUTER
DAN TEKNOLOGI INFORMASI UNIVERSITAS SUMATERA UTARA

- Membaca : Surat Permohonan Mahasiswa Fasilkom-TI USU tanggal 3 Juli 2024 perihal permohonan ujian skripsi:
Nama : NURUL ANDINI
NIM : 171402046
Program Studi : Sarjana (S-1) Teknologi Informasi
Judul Skripsi : Prediksi Pertumbuhan Tanaman Tomat Berdasarkan Pengambilan Citra Oleh Robot Menggunakan Gradient Boosting
- Memperhatikan : Bahwa Mahasiswa tersebut telah memenuhi kewajiban untuk ikut dalam pelaksanaan Meja Hijau Skripsi Mahasiswa pada Program Studi Sarjana (S-1) Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara TA 2023/2024.
- Menimbang : Bahwa permohonan tersebut diatas dapat disetujui dan perlu ditetapkan dengan surat keputusan
- Mengingat : 1. Undang-undang Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional.
2. Peraturan Pemerintah Nomor 17 tahun 2010 tentang pengelolaan dan penyelenggara pendidikan.
3. Keputusan Rektor USU Nomor 03/UN5.1.R/SK/SPB/2021 tentang Peraturan Akademik Program Sarjana Universitas Sumatera Utara.
4. Surat Keputusan Rektor USU Nomor 1876/UN5.1.R/SK/SDM/2021 tentang pengangkatan Dekan Fasilkom-TI USU Periode 2021-2026

MEMUTUSKAN

- Menetapkan :
Pertama : Membentuk dan mengangkat Tim Penguji Skripsi mahasiswa sebagai berikut:
Ketua : Ivan Jaya S.Si., M.Kom.
NIP: 198407072015041001
Sekretaris : Umayra Ramadhani Putri Nasution S.TI, M.Kom.
NIP: 199104112021022001
Anggota Penguji : Baihaqi Siregar S.Si., MT.
NIP: 197901082012121002
Anggota Penguji : Fahrurrozi Lubis B.IT., M.Sc.IT
NIP: 198610122018052001
Moderator : -
Panitera : -
- Kedua : Segala biaya yang diperlukan untuk pelaksanaan kegiatan ini dibebankan pada Dana Penerimaan Bukan Pajak (PNPB) Fasilkom-TI USU Tahun 2024.
- Ketiga : Keputusan ini berlaku sejak tanggal ditetapkan dengan ketentuan bahwa segala sesuatunya akan diperbaiki sebagaimana mestinya apabila dikemudian hari terdapat kekeliruan dalam surat keputusan ini.

- Tembusan :
- 1. Ketua Program Studi Sarjana (S-1) Teknologi Informasi
 - 2. Yang bersangkutan
 - 3. Arsip

Medan, 04 Juli 2024
Ditandatangani secara elektronik oleh:
Dekan



Maya Silvi Lydia
NIP 197401272002122001