

**KLASIFIKASI TANAMAN AIR BERDASARKAN KEBUTUHAN CAHAYA  
PADA AQUASCAPE MENGGUNAKAN ALGORITMA YOLO**

**SKRIPSI**

**MAILAN RONI SAPUTRA**

**171401135**



**PROGRAM STUDI S1 ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2024**

KLASIFIKASI TANAMAN AIR BERDASARKAN KEBUTUHAN CAHAYA PADA  
AQUASCAPE MENGGUNAKAN ALGORITMA YOLO

SKRIPSI

MAILAN RONI SAPUTRA

171401135



PROGRAM STUDI S1 ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI UNIVERSITAS  
SUMATERA UTARA

MEDAN

2024

## PERSETUJUAN

Judul : KLASIFIKASI TANAMAN AIR BERDASARKAN  
KEBUTUHAN CAHAYA PADA AQUASCAPE  
MENGUNAKAN ALGORITMA YOLO

Kategori : SKRIPSI

Nama : MAILAN RONI SAPUTRA

Nomor Induk Mahasiswa : 171401135

Program Studi : SARJANA ( S1 ) ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI  
INFORMASI UNIVERISTAS SUMATERA UTARA

Tanggal Sidang : 9 Juli 2024

Komisi Pembimbing

Pembimbing 2



Ivan Jaya S.Si., M.Kom.

NIP. 198407072015041001

Pembimbing 1




Sri Melvani Hardi S.Kom., M.Kom.

NIP. 198805012015042006

Diketahui / disetujui oleh

Program Studi S1 Ilmu Komputer

Ketua

  
Dr. Amalia ST. MT  
NIP. 197812212014042001

**PERNYATAAN****KLASIFIKASI TANAMAN AIR BERDASARKAN KEBUTUHAN CAHAYA  
PADA AQUASCAPE MENGGUNAKAN ALGORITMA YOLO****SKRIPSI**

Dengan ini saya mengakui bahwasanya skripsi yang saya tulis ini merupakan hasil karya saya sendiri, kecuali beberapa ringkasan serta kutipan yang masing-masing telah dicantumkan asal sumbernya.

Medan, 16 Juli 2024



Mailan Roni Saputra

171401135

### UCAPAN TERIMA KASIH

Segala puji dan rasa syukur penulis panjatkan kepada ALLAH SWT karna atas limpahan rahmat dan hidayat-Nya penulis akhirnya mampu menyelesaikan penyusunan skripsi ini yang merupakan bagian dari proses penulis untuk memperoleh gelar Sarjana Komputer.

Tak lupa juga penulis menyampaikan rasa terima kasih dan rasa syukur terhadap segala pihak yang telah berkontribusi sehingga skripsi ini dapat terselesai dengan lancar. Pihak-pihak tersebut antara lain:

1. Bapak Prof. Dr. Muryanto Amin, S.Sos., M.Si. Rektor Universitas Sumatera Utara.
2. Ibu Dr. Maya Silvi Lydia B.Sc. M.Sc., Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
3. Ibu Dr. Amalia ST., M.T., Ketua Program Studi S-1 Ilmu Komputer Universitas Sumatera Utara.
4. Ibu Sri Melvani Hardi, S.Kom. M.Kom., sebagai Dosen Pembimbing I penulis yang telah banyak memberikan bimbingan, arahan, masukan, serta dukungan dalam penulisan skripsi ini.
5. Bapak Ivan Jaya, S.Si., M.Kom. sebagai Dosen Pembimbing II penulis yang telah banyak memberikan bimbingan, arahan, masukan, serta dukungan dalam penulisan skripsi ini.
6. Seluruh dosen dan staf pegawai Program Studi S-1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara yang telah memberikan ilmu pengetahuan yang berguna selama masa perkuliahan.
7. Orang tua tercinta Papa Eri Yanto, Mama Anik Hidayah, Kakak Popi Dwi Patrojani S.Sos., Adik M. Jejen Tivana dan Abiyu Akram.
8. Teman-teman yang selalu memberikan motivasi dan semangat kepada penulis, Ferouz Syahalam, M. Raihansyah Pohan, Arya Duta Dias, dan Azhar Bazla Rabbani.
9. Teman-teman Kom C Ilmu Komputer 2017, Fadli, Jaka, Egi, Doni, Zidane, dan teman-teman penulis lainnya selama masa perkuliahan.

10. Teman-teman Gamer Go Brrrr, M. Aulia Ramadhan, Charli, Rizky Ramadhan Tanjung dan teman-teman lainnya yang tidak bisa saya sebutkan satu persatu.
11. Teman-teman mabar Minecraft, Jeremy Michael Sinaga dan Alfatih Rizka.
12. Teman-teman penulis, M.Wally Gilbrant Althaf, M. Havid Abdillah, Joni Saputra dan teman-teman lainnya yang tdaapat disebutkan satu persatu.
13. Dan juga pihak-pihak lainnya yang mana telah membantu penulis baik secara langsung atau tidak langsung yang tidak dapat penulis sebutkan namanya satu persatu.

Semoga Allah SWT memberikan rahmat-Nya terhadap segala pihak dan teman-teman yang telah berkontribusi terhadap penulis dalam menyelesaikan penulisan skripsi. Penulis sadar bahwasanya didalam skripsi ini mungkin terdapat beberapa kesalahan dan kekurangan. Namun penulis berharap agar skripsi ini bisa memberikan manfaat yang baik bagi para pembaca.

## ABSTRAK

### KLASIFIKASI TANAMAN AIR BERDASARKAN KEBUTUHAN CAHAYA PADA AQUASCAPE MENGGUNAKAN ALGORITMA YOLO

Tanaman air adalah tanaman yang media hidupnya berada di lingkungan perairan. Tanaman air juga sering digunakan sebagai salah satu komponen untuk menghias aquascape dikarenakan tampilan dan bentuknya yang menarik. Aquascape adalah seni yang menggabungkan penggunaan tanaman air, ikan hias, kayu, batu, dan ornamen lainnya dalam akuarium untuk menciptakan ekosistem alami di dalam air yang sangat indah. Tanaman air membutuhkan cahaya agar dapat tumbuh dengan baik. Setiap jenis tanaman air memerlukan tingkat kebutuhan cahaya yang berbeda. Maka dibangunlah sistem yang dapat mengklasifikasikan citra tanaman air. Ada beberapa metode yang dapat digunakan dalam melakukan klasifikasi dalam bidang pengolahan citra, salah satunya yaitu algoritma YOLO. Pertama dilakukan pengumpulan data sebanyak 226 citra tanaman air dengan perbedaan tingkat kebutuhan cahaya sebagai kelasnya. Lalu dilakukan pra pemrosesan citra dengan menggunakan platform roboflow. Roboflow mampu melakukan pra pemrosesan data dari pembagian data citra hingga proses augmentasi. Hasil dari pra pemrosesan di platform roboflow yaitu dataset yang siap digunakan untuk proses pelatihan. Pelatihan dilakukan menggunakan Google Colab Notebooks menggunakan metode YOLO. Untuk mencari model terbaik, dilakukan juga fine-tuning pada nilai hyperparameter. Model terbaik hasil latih lalu digunakan dalam sistem yang dibangun untuk melakukan klasifikasi terhadap tanaman air. Hasil pengujian menunjukkan bahwa algoritma YOLO mampu melakukan klasifikasi kebutuhan cahaya pada tanaman air dengan akurasi yang cukup tinggi yaitu 80,95%. Metode ini dapat digunakan untuk membantu pegiat aquascape dalam memilih tanaman air berdasarkan kebutuhan cahayanya.

**Kata kunci:** tanaman air, aquascape, kebutuhan cahaya, algoritma YOLO, klasifikasi, roboflow

## ABSTRACT

### AQUATIC PLANTS CLASSIFICATION BASED ON LIGHT REQUIREMENTS IN AQUASCAPE USING THE YOLO ALGORITHM

Aquatic plants are plants that live in aquatic environments. These plants are often used as components to decorate aquascapes because of their attractive appearance and forms. Aquascaping is an art that involves combining aquatic plants, ornamental fish, wood, rocks, and other decorations within an aquarium to create a beautifully integrated natural underwater ecosystem. Aquatic plants require light to grow well, and each type has different light requirements. Therefore, a system was developed to classify images of aquatic plants. Several methods can be used for image classification in this field, one of which is the YOLO algorithm. First, data collection was conducted, gathering 226 images of aquatic plants categorized by their light requirements. Next, image preprocessing was performed using the Roboflow platform. Roboflow is capable of data preprocessing, from image data splitting to augmentation. The result of preprocessing on the Roboflow platform was a dataset ready for training. Training was carried out using Google Colab Notebooks with the YOLO method. To find the best model, fine-tuning of the hyperparameters was also performed. The best-trained model was then used in the developed system to classify aquatic plants. Test results showed that the YOLO algorithm could classify the light requirements of aquatic plants with quite high accuracy, achieving 80.95%. This method can assist aquascape enthusiasts in selecting aquatic plants based on their light needs.

**Keywords:** aquatic plants, aquascape, light requirements, YOLO algorithm, classification, Roboflow



## DAFTAR ISI

	Halaman
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
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah.....	3
1.3. Batasan Masalah .....	3
1.4. Tujuan Penelitian .....	4
1.5. Manfaat Penelitian .....	4
1.6. Metodologi Penelitian.....	4
1.7. Sistematika Penulisan .....	6
BAB 2 LANDASAN TEORI.....	7
2.1. Aquascape .....	7
2.2. Kebutuhan Cahaya pada Tanaman Air .....	8
2.2.1. Kadar Cahaya Rendah .....	8
2.2.2. Kadar Cahaya Sedang.....	8
2.2.3. Kadar Cahaya Tinggi.....	8
2.3. <i>Machine Learning</i> .....	9

2.4.	<i>Deep Learning</i> .....	9
2.5.	YOLO .....	10
2.6.	Python .....	12
2.7.	Flask.....	13
2.8.	Roboflow.....	14
2.9.	<i>Confusion Matrix</i> .....	14
2.9.1.	<i>Accuracy</i> .....	15
2.9.2.	<i>Precision</i> .....	15
2.9.3.	<i>Recall</i> .....	16
BAB 3 ANALISIS DAN PERANCANGAN SISTEM .....		17
3.1.	Analisis Sistem.....	17
3.1.1.	Analisis Masalah .....	17
3.1.2.	Analisis Kebutuhan .....	18
3.2.	Pemodelan Sistem.....	19
3.2.1.	<i>General Architecture</i> .....	19
3.2.2.	<i>Use Case Diagram</i> .....	28
3.2.3.	<i>Activity Diagram</i> .....	28
3.2.4.	<i>Sequence Diagram</i> .....	29
3.3.	Flowchart .....	30
3.3.1.	Flowchart Training Dataset .....	30
3.3.2.	Flowchart Klasifikasi.....	31
3.3.3.	Perancangan Antarmuka ( <i>Interface</i> ).....	32
3.3.4.	Rancangan Halaman Depan .....	33
BAB 4 IMPLEMENTASI DAN PENGUJIAN .....		35
4.1.	Implementasi Sistem.....	35

4.1.1.	Implementasi Data.....	35
4.1.2.	Implementasi <i>Training</i> .....	37
4.1.3.	Implementasi Aplikasi.....	38
4.2.	Pengujian Sistem.....	41
4.2.1.	Pengujian <i>Black Box</i> .....	41
4.2.2.	Pengujian Validitas .....	42
BAB 5 KESIMPULAN DAN SARAN .....		46
5.1.	Kesimpulan .....	46
5.2.	Saran .....	46
DAFTAR PUSTAKA .....		48

**DAFTAR TABEL**

<b>Tabel 4. 1</b>	Pembagian Data primer/mentah .....	36
<b>Tabel 4. 2</b>	Tabel data sintesis .....	36
<b>Tabel 4. 3</b>	Validasi.....	38
<b>Tabel 4. 4</b>	Pengujian Black Box .....	41
<b>Tabel 4. 5</b>	Pengujian Validitas.....	42
<b>Tabel 4. 6</b>	Tabel Confusion Matrix .....	44
<b>Tabel 4. 7</b>	Tabel Recall.....	45
<b>Tabel 4. 8</b>	Tabel Presisi .....	45

## DAFTAR GAMBAR

<b>Gambar 2. 1</b> Ilustrasi Aquascape (sumber: lifestyle.kompas.com) .....	7
<b>Gambar 2. 2</b> Arsitektur YOLOv5 (sumber: researchgate.net).....	11
<b>Gambar 2. 3</b> Tabel Confusion Matrix.....	15
<b>Gambar 3. 1</b> Diagram Ishikawa.....	18
<b>Gambar 3. 2</b> Diagram General Architecture.....	20
<b>Gambar 3. 3</b> Resizing Citra Tanaman Air .....	21
<b>Gambar 3. 4</b> Flipping Citra Tanaman Air .....	22
<b>Gambar 3. 5</b> Rotate Citra Tanaman Air.....	23
<b>Gambar 3. 6</b> Saturasi Citra Tanaman Air .....	24
<b>Gambar 3. 7</b> Kecerahan Citra Tanaman Air .....	24
<b>Gambar 3. 8</b> Blur Citra Tanaman Air .....	25
<b>Gambar 3. 9</b> Noise Citra Tanaman Air.....	25
<b>Gambar 3. 10</b> Feature Map .....	26
<b>Gambar 3. 11</b> Feature Pyramid Network.....	27
<b>Gambar 3. 12</b> Use Case Diagram .....	28
<b>Gambar 3. 13</b> Activity Diagram .....	29
<b>Gambar 3. 14</b> Sequence Diagram .....	30
<b>Gambar 3. 15</b> Flowchart Training Dataset .....	31
<b>Gambar 3. 16</b> Flowchart Klasifikasi.....	32
<b>Gambar 3. 17</b> Halaman Depan Klasifikasi .....	33
<b>Gambar 3. 18</b> Rancangan Halaman Hasil.....	34
<b>Gambar 3. 19</b> Rancangan Halaman Tentang Aquascape .....	34
<b>Gambar 4. 1</b> Gambar dataset .....	37
<b>Gambar 4. 2</b> Grafik Hasil Pelatihan Dataset .....	38
<b>Gambar 4. 3</b> Halaman Antarmuka.....	39
<b>Gambar 4. 4</b> Halaman saat citra dimasukkan .....	39
<b>Gambar 4. 5</b> Halaman Hasil .....	40
<b>Gambar 4. 6</b> Penjelasan Tanaman Air .....	40
<b>Gambar 4. 7</b> Contoh Tanaman Air .....	41

## **BAB 1**

### **PENDAHULUAN**

#### **1.1. Latar Belakang**

Tanaman air merupakan tanaman yang hidup di lingkungan perairan. Ciri-ciri tanaman air meliputi kutikula tipis, stomata yang tidak aktif, peningkatan jumlah stomata pada satu sisi daun, daun datar, dan adanya udara untuk pengapungan. Tanaman akuatik, juga dikenal sebagai *hidrophytic* atau *hydrophytes*, merupakan sebuah tanaman yang telah menyesuaikan dirinya untuk tinggal dan tumbuh di lingkungan perairan. Tanaman ini memerlukan banyak adaptasi khusus untuk tumbuh di lingkungan yang dikategorikan basah atau di dalam air. Tanaman yang hidup di air dan lingkungan sekitarnya juga disebut tanaman akuatik. (Widjaja, 2013). Tanaman air juga sering digunakan sebagai salah satu komponen untuk menghias aquascape dikarenakan tampilan dan bentuknya yang menarik.

Aquascape merupakan seni dengan menggabungkan penggunaan tanaman air, ikan hias, kayu, batu, dan ornamen lainnya dalam akuarium untuk menciptakan ekosistem alami di dalam air yang sangat indah. Salah satu unsur yang menunjang pembuatan seni aquascape adalah pemilihan dan penggunaan material tanaman air yang tepat dan sesuai dengan kondisi wadah, media air, dan material lainnya (Widjaja, 2013). Ada beragam jenis tanaman air yang bisa digunakan untuk mengisi aquascape. Menurut Taufik (2013), tanaman air memerlukan cahaya untuk digunakan sebagai pengganti sinar matahari dalam menunjang pertumbuhannya. Maka dari itu digunakanlah cahaya yang bersumber dari lampu dengan tingkat kecerahan disesuaikan terhadap kebutuhan tanaman dan tingkat kedalaman akuarium. Secara garis besar, tanaman air dapat dibagi 3 berdasarkan kebutuhan cahayanya, yaitu tanaman yang berkebutuhan cahaya tinggi, tanaman yang berkebutuhan cahaya sedang, dan tanaman berkebutuhan cahaya rendah. Maka dibangunlah suatu sistem yang mampu mengklasifikasi tanaman air sehingga kita dapat mengetahui

tanaman air mana yang cocok untuk mengisi aquascape berdasarkan kebutuhan cahayanya.

Klasifikasi adalah sebuah metode dan menjadi tahap yang penting dari data mining yang melibatkan pengelompokan data atau objek baru ke dalam sebuah kelas atau label yang didasarkan pada atribut tertentu. Teknik dalam klasifikasi bekerja dengan mengamati variabel dari data-data yang sudah ada. Tujuannya adalah untuk memprediksi kelas dari objek yang belum diketahui sebelumnya.

Ada beberapa algoritma yang dapat digunakan untuk melakukan klasifikasi, salah satunya yaitu algoritma YOLO. *You Only Look Once* (YOLO) merupakan algoritma untuk deteksi objek dengan berbasis *Region-based Convolutional Neural Network* (R-CNN). Model ini dapat banyak melakukan banyak objek, melakukan prediksi kelas yang dibuat, serta melakukan indentifikasi lokasi dari objek tersebut. YOLO juga diklaim sebagai arsitektur yang terbilang akurat dan cepat (Dawami et al., 2023). Algoritma YOLO memiliki beberapa versi yang cukup populer digunakan untuk deteksi dan klasifikasi objek, salah satunya yaitu algoritma YOLOv5.

YOLOv5 pertama kali diperkenalkan oleh Glenn Jocher, pendiri Ultralytics. Tidak seperti pendahulunya yang menggunakan darknet dan bahasa C, YOLOv5 ditulis dalam bahasa Python dengan framework PyTorch. Salah satu dari keunggulan PyTorch adalah komunitasnya yang terbilang besar, berkat kemudahan dan kesederhanaan dalam pengimplementasiannya. Selain itu, PyTorch juga memudahkan dalam persiapan dan integrasi dengan perangkat IoT (Internet of Things) pada masa depan. (Setiana Riva, 2023) . Model jaringan ini memiliki keakuratan deteksi yang terbilang tinggi dengan inferensi yang memiliki kecepatan yang baik, dengan kecepatan deteksi dapat mencapai hingga 140 frame per detik. Selain itu, ukuran file dari bobot model YOLOv5 sangat kecil, hampir 90% lebih kecil dibandingkan dengan YOLOv4. Hal ini dapat menunjukkan bahwa YOLOv5 mampu dan dapat untuk diterapkan pada perangkat yang tertanam dan memerlukan deteksi waktu yang nyata. Oleh karena itu, keunggulan YOLOv5 adalah dari segi

tingkat akurasi deteksi yang tinggi, kecepatan deteksi yang cepat secara bersamaan, meskipun memiliki sifat yang ringan (Akhyar et al., 2022).

Menurut (Wibowo et al., 2023) dalam penelitian mengenai Implementasi Algoritma *Deep Learning You Only Look Once* (YOLOv5) yang digunakan dalam Mendeteksi Buah Segar Dan Busuk disimpulkan bahwa algoritma tersebut mampu mengenali objek dengan akurasi yang tinggi secara konsisten yaitu mencapai 90%. Hasil akurasi tinggi dengan algoritma YOLOv5 yang dioptimasi mencapai 92,57% juga didapatkan oleh penelitian Deteksi Penyakit Tanaman berbasis metode Klasifikasi yang dilakukan oleh (Wang et al., 2022) dan dapat dinyatakan bahwa algoritma ini memiliki hasil yang akurat. Berdasarkan hasil dari penelitian diatas yang telah dijabarkan, maka penulis ingin melakukan penelitian dengan judul ‘Klasifikasi Tanaman Air Berdasarkan Kebutuhan Cahaya pada Aquascape menggunakan Algoritma YOLOv5’.

## **1.2. Rumusan Masalah**

Beragamnya jenis tanaman air yang ada pada aquascape seringkali menimbulkan kebingungan bagi para penggiat aquascape. Apalagi setiap tanaman air juga memiliki kebutuhan cahaya yang berbeda, sehingga peggiat aquascape kesulitan dalam memilih tanaman air yang tepat untuk dibudidayakan pada aquascape mereka. Maka dibutuhkan sebuah sistem yang dapat membantu pengguna dalam mengklasifikasikan tanaman air berdasarkan kebutuhan cahayanya dalam aquascape untuk memaksimalkan pertumbuhan dari tanaman.

## **1.3. Batasan Masalah**

Dalam melakukan penelitian ini, penulis merumuskan apa saja batasan terhadap masalah yang diteliti. Batasan masalah pada penelitian ini yaitu:



1. Mengklasifikasikan tanaman air di dalam aquascape berdasarkan kebutuhan kadar cahayanya, yaitu berkebutuhan tinggi, sedang, maupun rendah.
2. Metode yang diterapkan adalah algoritma YOLOv5.
3. Sistem yang dibangun akan berbasis web aplikasi, dan memakai bahasa pemrograman JavaScript dan Python, serta menggunakan framework Flask.
4. Data yang digunakan berupa citra gambar.
5. Data diambil dari laman [www.tropica.com](http://www.tropica.com) beserta keterangan kebutuhan cahayanya serta dari toko aquascape terdekat sekitaran kota Medan.

#### **1.4. Tujuan Penelitian**

Tujuan yang peneliti harapkan dari sistem yang dibangun adalah untuk mengklasifikasikan tanaman air berdasarkan kebutuhan cahayanya di dalam aquascape menggunakan algoritma YOLO.

#### **1.5. Manfaat Penelitian**

Manfaat yang peneliti harapkan dari sistem yang dibangun adalah untuk memudahkan para pemilik aquascape dalam menentukan tanaman yang akan ditata dalam aquascape sesuai dengan ketersediaan cahayanya.

#### **1.6. Metodologi Penelitian**

Berikut merupakan tahap-tahap yang dilakukan penulis dalam penelitian ini antara lain yaitu :

##### **1. Studi Pustaka**

Pada tahapan studi pustaka, peneliti memulai dengan mencari referensi berupa buku, artikel, jurnal dan berbagai bentuk penelitian tertulis lainnya, yang berhubungan terhadap topik yang dibahas dalam penelitian ini, yaitu Klasifikasi Tanaman Air pada Aquascape dan Algoritma YOLO. Referensi yang digunakan berupa buku, artikel, jurnal, maupun penelitian terdahulu yang berhubungan dengan penelitian ini.

## 2. Analisis

Tahap analisis ini berguna untuk memproses data-data yang telah dikumpulkan saat studi Pustaka, setelah itu dilakukan analisa terhadap hal-hal yang diperlukan dalam penelitian dan perancangan sistem klasifikasi yang menggunakan Algoritma YOLO sehingga menjadi sistem yang baik dan terstruktur.

## 3. Perancangan

Pada proses perancangan ini dilakukan implementasi dari hasil analisa kebutuhan pembuatan sistem dengan merancang Arsitektur umum sistem, seperti *Flowchart*, dan pembuatan *User Interface* aplikasi yang nantinya berbasis WEB.

## 4. Implementasi

Pada tahapan implementasi, penulis akan membuat sebuah sistem dengan memanfaatkan bahasa pemrograman yang sesuai dengan *flowchart* yang telah dirancang.

## 5. Pengujian

Pada tahapan pengujian, penulis akan menguji apakah sistem yang dibangun sesuai Bahasa pemrograman yang sesuai dengan *flowchart* telah berhasil bekerja sesuai dengan kebutuhan yang diharapkan sebelumnya dan jika terdapat kesalahan, maka perbaikan akan dilakukan oleh penulis.

## 6. Dokumentasi

Pada tahap ini, seluruh proses penelitian mulai pada saat analisa sampai pengujian sistem akan dalam bentuk skripsi untuk menunjukkan hasil penelitian ini.

### **1.7. Sistematika Penulisan**

Berikut ini merupakan sistematika dalam penulisan skripsi ini yang terdiri dari 5 bab, yaitu:

#### **BAB 1 : PENDAHULUAN**

Bab ini akan membahas tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian dan sistematika penulisan skripsi.

#### **BAB 2 : LANDASAN TEORI**

Bab ini akan membahas mengenai landasan teori yang berkaitan dengan tanaman air pada aquascape, algoritma YOLO, serta membahas metode yang dipakai dalam penelitian

#### **BAB 3 : ANALISIS DAN PERANCANGAN**

Didalam bab ini penulis membahas mengenai analisis masalah yang diteliti serta perancang sistem dan metode yang digunakan sebagai solusi dari masalah tersebut.

#### **BAB 4: IMPLEMENTASI DAN PENGUJIAN**

Didalam bab ini penulis membahas mengenai implementasi dari metode yang digunakan beserta pengujian dan hasil analisis dari sistem yang telah dibangun.

#### **BAB 5 : KESIMPULAN DAN SARAN**

Didalam bab ini penulis membahas mengenai kesimpulan dan hasil penelitian yang dilakukan beserta saran terhadap penelitian terkait yang diharapkan akan bermanfaat bagi pihak terkait maupun untuk pengembangan penelitian selanjutnya.

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1. Aquascape**

Aquascape, yang oleh orang Jepang lebih dikenal sebagai nature aquarium, adalah taman alami yang dibuat di dalam akuarium. Aquascape bukan hanya sekadar taman biasa, melainkan representasi hidup dari miniatur alam yang penuh sinergi. Di dalam akuarium, tanaman mendapatkan nutrisi dari media tanam, sementara ikan menghasilkan CO<sub>2</sub> yang diperlukan oleh tanaman. Sebagai timbal balik, tanaman memproduksi oksigen yang dibutuhkan oleh ikan, udang pemakan lumut, dan mikroorganisme lainnya. Semua komponen ini bekerja harmonis serta saling membutuhkan satu sama lain. Karena hal tersebut, aquascape sering dianggap sebagai "guru sejati" yang mengajarkan tentang keseimbangan hidup.(Widjaja, 2013). Terdapat beragam variasi tanaman air yang dapat dipilih untuk mengisi aquascape. Namun, sebelum menambahkannya ke dalam akuarium, penting untuk memilih tanaman air yang sesuai dengan ukuran pertumbuhan maksimalnya, kebutuhan akan cahaya, serta jenis media tanam yang akan digunakan. Ilustrasi aquascape dapat dilihat pada gambar 2.1 berikut.



**Gambar 2. 1** Ilustrasi Aquascape  
(sumber: lifestyle.kompas.com)

## **2.2. Kebutuhan Cahaya pada Tanaman Air**

Salah satu syarat penting dalam membuat aquascape adalah memenuhi kebutuhan cahaya di dalam akuarium. Tanaman memerlukan cahaya untuk menggantikan sinar matahari dalam menyokong pertumbuhannya. Cahaya ini dihasilkan dari lampu, yang kapasitasnya harus disesuaikan dengan kebutuhan tanaman dan kedalaman akuarium.. Namun, di dalam seni membuat aquascape, penataan cahaya tidak hanya perihal penerangan semata, tetapi juga bagaimana menciptakan pencahayaan yang dapat menjangkau seluruh bagian akuarium dan menjadikan akuarium tampak lebih memesona (Widjaja, 2013).

Bagi para pegiat aquascape yang memiliki sumber cahaya yang konstan dan tidak berubah, maka hal yang perlu diperhatikan ialah memilih tanaman air yang sesuai berdasarkan kebutuhan cahayanya. Menurut (Widjaja, 2013), secara garis besar, tanaman air dapat dibagi 3 berdasarkan kebutuhan cahayanya, yaitu:

### **2.2.1. Kadar Cahaya Rendah**

Tanaman yang tumbuh secara alami di tempat teduh atau jauh dari sinar matahari langsung adalah jenis yang membutuhkan sedikit cahaya. Oleh karena itu, tanaman ini tidak memerlukan banyak lampu dalam perawatannya, menjadikannya pilihan yang baik untuk pemula. Tanaman seperti Kadaka dan Anubias termasuk dalam kategori low light, mampu tumbuh dengan kadar cahaya rendah.

### **2.2.2. Kadar Cahaya Sedang**

Kadar cahaya medium merupakan kadar cahaya tidak terik. Tanaman seperti Cryptocorine dan Echinodorus memerlukan kadar sinar medium yang dua kali lebih banyak dibandingkan dengan kadar cahaya rendah agar daunnya dapat tumbuh besar dan bebas lumut.

### **2.2.3. Kadar Cahaya Tinggi**

Kadar cahaya tinggi merupakan kadar cahaya yang sangat terang dan diperlukan oleh tanaman dengan daun berwarna cerah, seperti kuning, jingga, dan merah. Umumnya, semua tanaman yang bisa diperbanyak dengan stek dan berdaun kecil memerlukan sinar yang banyak. Contohnya, Hemiantus Cuba dan Rotala

marcandra. Kedua tanaman ini memerlukan kadar cahaya tinggi sehingga harus diletakkan tepat di bawah lampu dengan ketinggian akuarium tidak lebih dari 50 cm.

### 2.3. *Machine Learning*

Menurut Arthur Samuel, *Machine Learning* merupakan bidang studi yang memungkinkan komputer belajar dengan sendirinya tanpa harus diarahkan secara langsung oleh pengguna secara eksplisit. Arthur Samuel terkenal karena program catur yang dibuatnya. *Machine Learning* (ML) digunakan dalam hal mengajarkan mesin mengenai cara mengelola data secara lebih efisien. Terkadang, setelah melihat data, terdapat kesulitan dalam menginterpretasikan informasi yang diekstraksi melalui data tersebut. Dalam situasi seperti ini, kita menerapkan machine learning. Dengan semakin banyaknya dataset yang tersedia akan mengakibatkan terjadinya peningkatan permintaan pada *machine learning*. Banyak industri menggunakan *machine learning* untuk melakukan ekstraksi data yang mereka anggap relevan. Tujuan *machine learning* adalah belajar melalui data. Telah banyak penelitian yang dilakukan untuk mengembangkan cara supaya mesin dapat belajar sendiri tanpa harus diarahkan secara eksplisit oleh pengguna. Banyak matematikawan serta programmer mencoba berbagai pendekatan dalam menemukan solusi terkait permasalahan ini dengan melibatkan dataset yang besar (Mahesh, 2018).

### 2.4. *Deep Learning*

Dalam beberapa tahun terakhir, machine learning telah menjadi semakin populer dalam penelitian dan telah diintegrasikan dalam sejumlah besar aplikasi, termasuk pengambilan konsep multimedia, klasifikasi gambar, rekomendasi video, analisis jaringan sosial, penambangan teks, dan sebagainya. Di antara berbagai algoritma *Machine Learning*, "*Deep Learning*," juga dikenal sebagai representation learning, secara luas digunakan dalam aplikasi-aplikasi ini. Pertumbuhan yang pesat dan ketersediaan data, ditambah dengan kemajuan luar biasa dalam teknologi perangkat keras, telah mendorong kemunculan studi-studi baru dalam pembelajaran terdistribusi dan pembelajaran mendalam. *Deep learning*, yang berkembang dari

jaringan saraf konvensional, secara signifikan mengungguli pendahulunya. Ini memanfaatkan teknologi graf dengan transformasi antar neuron untuk mengembangkan model pembelajaran berlapis-lapis. Banyak teknik *deep learning* terbaru telah diperkenalkan dan menunjukkan hasil yang menjanjikan dalam berbagai aplikasi, seperti *Natural Language Processing* (NLP), pemrosesan data visual, pemrosesan suara dan audio, serta banyak aplikasi terkenal lainnya. (Pouyanfar et al., 2018).

## 2.5. YOLO

YOLO (*You Only Look Once*) merupakan sebuah algoritma deteksi objek yang sangat populer dalam bidang computer vision. Algoritma ini dikembangkan oleh Joseph Redmon, Santosh Divvala, Ross Girshick, dan Ali Farhadi. Berbeda dengan pendekatan deteksi objek lainnya yang membagi gambar menjadi beberapa grid dan melakukan deteksi pada setiap grid secara terpisah, YOLO menggunakan pendekatan end-to-end. YOLO membagi gambar menjadi grid dan secara simultan memprediksi kotak pembatas (*bounding box*) serta probabilitas kelas objek dalam satu proses penghitungan.

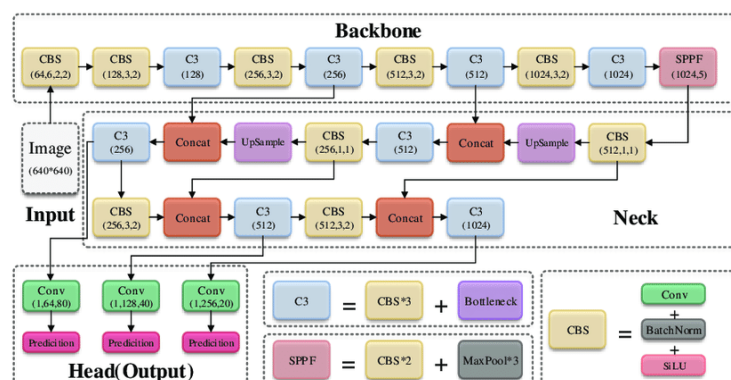
(Liu et al., 2022) menjelaskan untuk algoritma deteksi objek pada versi terdahulu, misalnya YOLOv1, YOLOv2, dll., struktur jaringannya sering kali berupa tumpukan beberapa lapisan konvolusi dan lapisan koneksi penuh. Hanya peta fitur dengan ukuran tetap yang dihitung, yang sangat membatasi kemampuan deteksi pada model multiskala. Hal ini menyebabkan akurasi algoritma pendeteksian untuk target kecil kurang ideal. Untuk memecahkan masalah tersebut, metode yang efektif adalah dengan mengadopsi FPN (*Feature Pyramid Network*), yang dapat meningkatkan kemampuan deteksi multiskala model, namun biaya komputasi yang tinggi dan kecepatan inferensi membatasi pengembangan strategi piramida gambar dalam algoritma deteksi objek.

Berdasarkan metode yang disebutkan di atas, FPN mengambil sampel peta fitur dan kemudian menggabungkannya dengan keluaran yang sesuai di jaringan backbone dalam proses ini, dan menghasilkan beberapa peta fitur baru untuk deteksi akhir. Modifikasi ini meningkatkan kemampuan ekspresi fitur algoritma

dan selanjutnya meningkatkan efek algoritma berdasarkan kemampuan deteksi waktu nyata dan deteksi multiskala. Sementara itu, FPN mengadopsi jalur fusi fitur top-down, yang juga memperkuat kinerja algoritma pendeteksian objek kecil.

Melalui kombinasi backbone yang dirancang dengan baik dan jaringan fusi fitur, algoritma pendeteksian objek saat ini dapat memastikan akurasi pendeteksian dan kemampuan waktu nyata, serta memiliki kemampuan untuk mendeteksi berbagai skala objek, namun untuk desain spesifik, algoritma ini sering kali berfokus pada berbagai skenario kehidupan yang kompleks. Tujuan awalnya adalah untuk menjamin bahwa model matematika memiliki generalisasi dan kemampuan deteksi yang sempurna untuk target dengan skala berbeda, tetapi mengabaikan optimalisasi deteksi dalam beberapa adegan skala tertentu. Dalam konteks ini, merancang algoritma yang sederhana, efisien, ringan, dan mudah diterapkan untuk deteksi objek kecil memiliki signifikansi studi yang penting.

YOLOv5 pertama kali diperkenalkan oleh Glenn Jocher, pendiri Ultralytics. Tidak seperti pendahulunya yang menggunakan darknet dan bahasa C, YOLOv5 ditulis dalam bahasa Python menggunakan framework PyTorch. PyTorch memiliki keunggulan berupa komunitasnya yang besar, berkat kesederhanaan dan kemudahan dalam pengimplementasiannya. Ini juga mempermudah persiapan dan integrasi dengan perangkat Internet of Things (IoT) di masa depan. (Setiana Riva, 2023). Berikut merupakan arsitektur YOLOv5 secara umum pada gambar 2.2.



**Gambar 2. 2** Arsitektur YOLOv5  
(sumber: researchgate.net)



Proses yang terdapat dalam arsitektur YOLOv5 yakni dimulai dengan memasukkan citra ke dalam komponen *backbone*, yakni CSPDarknet53, berguna dalam hal ekstraksi berbagai fitur yang berasal dari citra. Pada setiap fitur yang telah dilakukan ekstraksi kemudian digabungkan menuju PANet sebelum pada akhirnya disalurkan ke komponen utama atau disebut *head*. Pada bagian *head* terjadi proses yang memiliki tugas dalam hal mendeteksi objek-objek dalam gambar. YOLOv5 memiliki tiga tahap deteksi yang berbeda untuk memastikan deteksi objek dari skala kecil hingga besar. YOLOv5 menggunakan tiga skala berbeda didalam setiap grid, yaitu 8, 16, dan 32. Misalnya, untuk gambar berukuran 320 x 320 piksel, dihasilkan grid berukuran 40 x 40 untuk mendeteksi objek kecil, 20 x 20 untuk objek sedang, dan 10 x 10 untuk objek besar. Setiap skala ini memiliki grid yang menghasilkan 3 *anchor box* dengan ukuran berbeda, sehingga total *bounding box* yang dihasilkan adalah  $((40 \times 40) + (20 \times 20) + (10 \times 10)) \times 3 = 6,300$ . Hasil deteksi dari ketiga tahap ini pada akhirnya akan digabungkan serta dilakukan *Non-Max Suppression* (NMS) dengan tujuan untuk memilih satu *bounding box* dengan *confidence* tertinggi jika terdapat beberapa *bounding box* untuk satu objek.

## 2.6. Python

Python yakni bahasa pemrograman komputer yang sudah banyak digunakan dalam hal membangun analisis data, pengembangan perangkat lunak, pengembangan web, dan berbagai keperluan lainnya. Python mempunyai sintaksis yang terbilang sederhana dan mudah untuk dipahami, sehingga membuatnya menjadi salah satu bahasa pemrograman populer dan banyak digemari.

Python saat ini menjadi pilihan yang utama dalam bidang ilmu data. Melalui Bahasa pemrograman ini analis data mampu melakukan perhitungan statistik yang kompleks, melakukan pembuatan visualisasi data, hingga melakukan implementasi algoritma *machine learning*. Selain itu, Python juga dapat digunakan dalam hal manipulasi hingga analisis data, serta menyelesaikan berbagai tugas yang berkaitan dengan data lainnya. Melalui Python, pembuatan berbagai jenis hal visualisasi data

sangat memungkinkan, terutama berkaitan dengan grafik garis, diagram lingkaran, grafik batang, histogram, dan sebagainya.

Python juga sering dipilih untuk pengembangan bagian *back-end* sebuah situs atau aplikasi, yaitu bagian yang tidak terlihat oleh pengguna. Perannya dalam pengembangan web meliputi pengiriman data dari dan ke server, pemrosesan data, komunikasi dengan server, manajemen URL, serta pengamanan situs. Selain itu, Python juga menyediakan beberapa *framework* untuk pengembangan web seperti Flask dan Django.

## 2.7. Flask

Flask adalah sebuah kerangka kerja atau bisa disebut *web framework* yang ditulis dalam bahasa pemrograman Python dan termasuk dalam kelompok *microframework*. Flask memiliki peran sebagai kerangka kerja yang berfungsi dalam membangun serta menampilkan situs web. Melalui Flask serta Python, pengembang dapat melakukan pembuatan situs web dengan terstruktur serta mampu mengatur perilaku situs lebih mudah. Flask disebut *microframework* sebab tampilan yang sangat ringan dan sederhana serta tidak memerlukan alat atau pustaka tertentu dalam penggunaannya.

Sebagian besar fungsi dan komponen umum seperti validasi formulir, database, dan lainnya tidak disertakan secara default dalam Flask. Hal ini karena fungsi-fungsi dan komponen-komponen tersebut telah disediakan oleh pihak ketiga dalam bentuk ekstensi, memungkinkan Flask untuk menggunakan fitur-fitur tersebut seolah-olah telah diimplementasikan oleh Flask sendiri. Meskipun Flask dikategorikan sebagai *microframework*, ini tidak berarti bahwa Flask memiliki keterbatasan dalam hal fungsionalitas.

*Microframework* pada Flask merujuk pada prinsip bahwa Flask dirancang untuk menyediakan inti aplikasi sesederhana mungkin, tetapi masih memungkinkan untuk dengan mudah menambahkan fitur-fitur tambahan. Dengan demikian, fleksibilitas dan skalabilitas Flask relatif tinggi dibandingkan dengan *framework* lainnya.

## 2.8. Roboflow

Roboflow merupakan platform yang memungkinkan pengembang dengan mudah membangun, mengelola, dan melatih model deteksi objek dan pengenalan gambar menggunakan dataset mereka sendiri. Platform ini menyediakan berbagai fitur, termasuk pengelolaan dataset, augmentasi data, pelatihan model, serta konversi dan distribusi model. Dengan Roboflow, pengguna dapat mengimpor dataset mereka, melakukan augmentasi data untuk meningkatkan kualitas model, dan melatih model deteksi objek menggunakan beberapa arsitektur model yang telah ada seperti YOLO, SSD, dan Faster R-CNN.

Salah satu keunggulan Roboflow adalah kemudahan penggunaannya. Platform ini menyediakan antarmuka yang intuitif dan ramah pengguna, sehingga memungkinkan pengembang untuk dengan cepat memulai dan melatih model deteksi objek tanpa memerlukan pengetahuan mendalam tentang machine learning atau computer vision. Selain itu, Roboflow juga menyediakan integrasi dengan berbagai platform seperti TensorFlow, PyTorch, dan ONNX untuk memudahkan penggunaan model yang telah dilatih pada suatu aplikasi maupun proyek.

## 2.9. Confusion Matrix

Confusion Matrix merupakan alat evaluasi kinerja berbentuk sebuah tabel yang berguna dalam hal mengklasifikasikan data dalam *machine learning*, hingga melakukan evaluasi kinerja secara menyeluruh terkait hasil prediksi model. Tabel berikut membandingkan prediksi model dengan nilai aktual terhadap data uji. *Confusion Matrix* memberikan sebuah gambaran yang lebih detail mengenai kinerja model, termasuk kesalahan yang dibuat. Tabel *Confusion Matrix* dapat dilihat pada gambar 2.3 berikut.

		Nilai Aktual	
		Positive	Negative
Nilai Prediksi	Positive	TP	FP
	Negative	FN	TN

**Gambar 2. 3** Tabel *Confusion Matrix*

*Confusion Matrix* terdiri dari empat komponen utama:

1. *True Positives* (TP) : Prediksi benar untuk kelas positif.
2. *True Negatives* (TN) : Prediksi benar untuk kelas negatif.
3. *False Positives* (FP) : Prediksi salah untuk kelas positif (dikenal sebagai *Type I error*).
4. *False Negatives* (FN) : Prediksi salah untuk kelas negatif (dikenal sebagai *Type II error*).

Berikut merupakan perhitungan yang berguna untuk mengukur kinerja suatu model dari tabel *confusion matrix*.

### 2.9.1. Accuracy

Akurasi merupakan sebuah *matrix* yang mengukur tingkat seberapa sering sebuah model membuat prediksi yang benar. Rumusnya adalah:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

### 2.9.2. Precision

Presisi digunakan dalam hal pengukuran mengenai banyaknya prediksi positif yang benar dibandingkan dari keseluruhan prediksi positif yang telah dibuat. Rumusnya adalah:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

### 2.9.3. *Recall*

*Recall* atau yang juga sering disebut sebagai Sensitivitas atau *True Positive Rate* merupakan sebuah kemampuan dalam mengukur banyaknya data positif yang terdeteksi secara benar oleh model dari semua data positif yang ada. Rumusnya adalah:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

## **BAB 3**

### **ANALISIS DAN PERANCANGAN SISTEM**

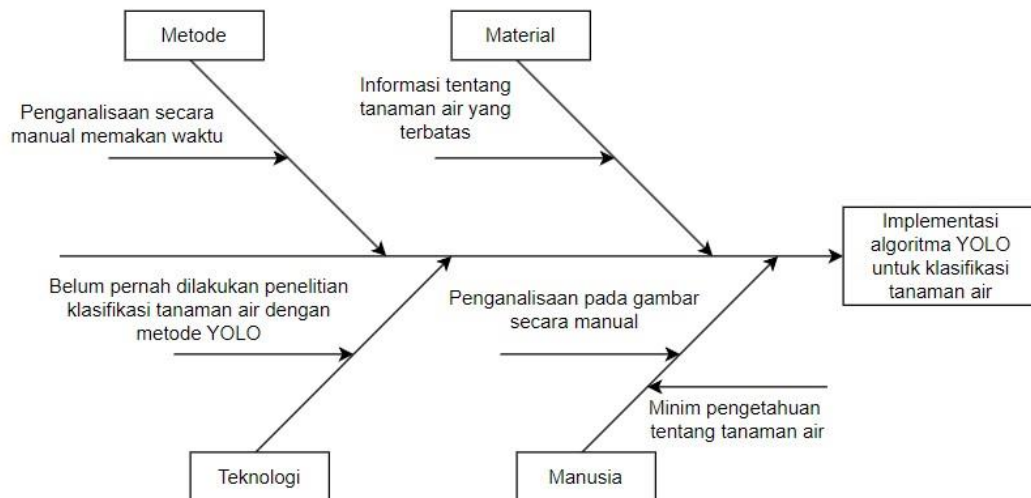
#### **3.1. Analisis Sistem**

Analisis sistem merupakan suatu langkah dasar dalam perancangan dan Pembangunan sistem di mana permasalahan suatu sistem dipecah menjadi beberapa bagian yang lebih sederhana. Tujuan dari analisis sistem adalah untuk memahami dan memperbaiki permasalahan yang ada, sehingga kinerja sistem dapat ditingkatkan menjadi lebih efisien. Proses analisis sistem terdiri dari beberapa tahap, antara lain analisis masalah serta analisis kebutuhan

##### **3.1.1. Analisis Masalah**

Analisis masalah merupakan proses identifikasi kasus berdasarkan penyebab yang terjadi selama proses pembuatan sistem dan menemukan alternatif solusi untuk mengatasi masalah tersebut. Analisis masalah diperlukan untuk mengurangi kesalahan dalam perancangan sistem sehingga sistem dapat beroperasi dengan optimal. Dalam hal ini yaitu mengimplementasikan aplikasi web yang dapat mengklasifikasikan tanaman air berdasarkan kebutuhan cahayanya dan dirancang untuk terwujudnya kemudahan bagi para penggunanya.

Untuk memudahkan identifikasi masalah, maka digunakan media berupa diagram tulang ikan (*fishbone* diagram atau diagram Ishikawa) yang akan menggambarkan dan mengidentifikasi hubungan antara penyebab dan akibat dari analisis permasalahan yang akan dibahas dalam penelitian ini. Berikut adalah diagram Ishikawa untuk analisis permasalahan tersebut.



**Gambar 3. 1** Diagram Ishikawa

### 3.1.2. Analisis Kebutuhan

Untuk menyediakan aplikasi yang memberikan informasi lengkap, dibutuhkan komponen penunjang dalam aplikasi tersebut. Analisis kebutuhan akan mengevaluasi berbagai kebutuhan yang harus dipenuhi supaya sistem dapat berfungsi dengan baik. Berikut ini adalah analisis kebutuhan berdasarkan kategori-kategori yaitu:

#### 1. Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang terkait dengan beberapa fungsi yang harus ada dalam perangkat lunak. Berikut adalah kebutuhan fungsional dari system:

- a. Sistem menampilkan beberapa informasi umum tentang tanaman air pada aquascape.
- b. Sistem dapat melakukan perhitungan dengan mengimplementasikan metode YOLO.
- c. Sistem mampu memberi prediksi tentang tingkat kebutuhan cahaya tanaman air berdasarkan gambar yang telah diunggah.

#### 2. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan kebutuhan yang mencakup batasan layanan yang terkait dengan kemampuan yang diberikan melalui aplikasi. Berikut adalah kebutuhan non-fungsional pada aplikasi:

- a. Aplikasi harus memiliki antarmuka yang sederhana agar mudah dipahami oleh pengguna.
- b. Aplikasi tidak memakan biaya tambahan dan dapat berjalan dengan baik dalam pengoperasiannya.
- c. Aplikasi mudah dan gampang diakses oleh pengguna.

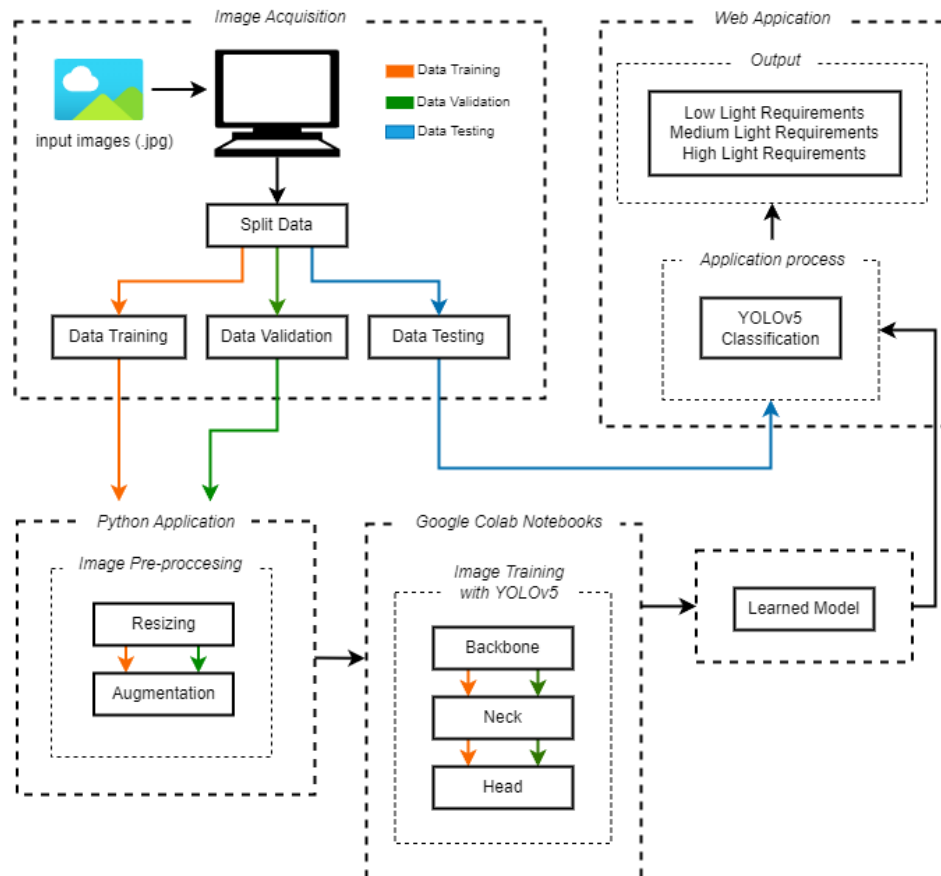
### **3.2. Pemodelan Sistem**

Pemodelan sistem digunakan untuk menggambarkan peran dan kondisi dari setiap pengguna terhadap sistem yang dibangun. Pemodelan sistem ini menggunakan *Unified Modeling Language* (UML) untuk desain dan perancangan sistem. Pemodelan dalam penelitian ini meliputi *general architecture*, *use case diagram*, *activity diagram*, dan *sequence diagram*.

#### **3.2.1. General Architecture**

General architecture adalah salah satu jenis pemodelan sistem yang menggambarkan struktur umum atau arsitektur keseluruhan dari sistem yang dibangun. Diagram arsitektur umum aplikasi yang dibangun bisa dilihat dari gambar 3.2 berikut.





**Gambar 3. 2** Diagram General Architecture

### 3.2.1.1 Image Acquisition

Dalam tahapan ini penulis pengumpulan data gambar (citra) yang akan digunakan pada penelitian ini. Data yang digunakan merupakan citra tanaman air berdasarkan kebutuhannya dalam aquascape. Pertama data citra yang telah dikumpulkan akan dipisahkan menjadi 3 kelas. Kemudian data citra dibagi menjadi 3 bagian yaitu *Data Training*, *Data Testing* dan *Data Validation*.

*Data training* adalah kumpulan data yang digunakan untuk melatih model dan digunakan untuk mempelajari pola, fitur, dan hubungan dalam data, sehingga model dapat membuat prediksi atau pengambilan keputusan yang akurat pada data baru. *Data testing* adalah data yang digunakan untuk menguji kinerja dan akurasi model setelah model tersebut dilatih. *Data validation* adalah kumpulan data yang digunakan untuk menilai kinerja model *machine learning* selama proses pelatihan

agar nantinya dapat disimpulkan nilai *hyperparameter* yang tepat dalam mencari model terbaik. Data validasi membantu memastikan bahwa model yang sedang dilatih tidak hanya bekerja dengan baik pada data training, tetapi juga dapat menggeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya.

#### 3.2.1.2. Image Pre-processing

Sebelum data diproses lebih lanjut, citra tanaman air akan memasuki *image pre-processing* terlebih dahulu. Dalam proses ini data citra akan melalui tahap *resizing* dan *augmentation*. Berikut merupakan tahap-tahap dari *image pre-processing* tersebut.

##### 3.2.1.2.1. Resizing

*Resizing* adalah proses mengubah ukuran gambar atau citra digital. Proses ini melibatkan penyesuaian dimensi (lebar dan tinggi) gambar agar sesuai dengan kebutuhan atau persyaratan tertentu yang dikehendaki. *Resizing* yang diterapkan yaitu mengubah ukuran data citra yang beragam menjadi ukuran 640 x 640. Berikut contoh citra yang telah di-*resizing*.



**Gambar 3. 3** Resizing Citra Tanaman Air

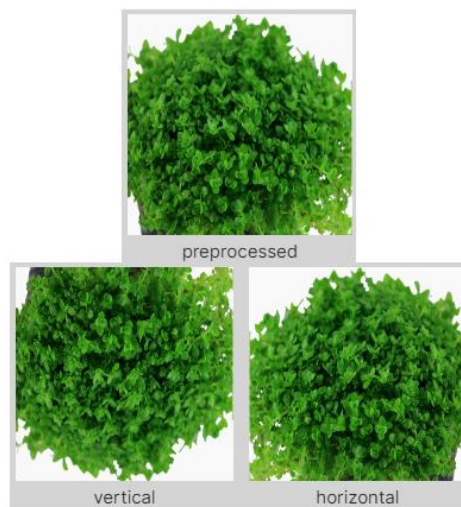
##### 3.2.1.2.2. Augmentation

*Augmentation* atau augmentasi data adalah teknik yang digunakan untuk memperbesar atau memperkaya dataset dengan menciptakan variasi baru dari data yang ada. *Augmentation* membantu meningkatkan kinerja model dengan menyediakan lebih banyak data untuk dilatih, yang dapat mengurangi *overfitting*.

dan meningkatkan kemampuan generalisasi model. Berikut metode yang digunakan dalam proses augmentasi data.

a. *Flip*

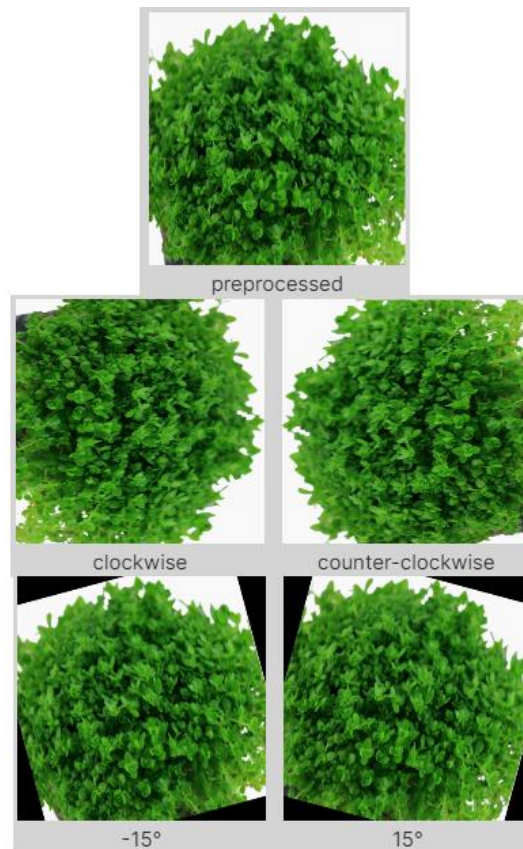
*Flip* atau *Flipping* merupakan proses untuk membalikkan data citra secara horizontal atau vertical. Berikut contoh data citra yang telah melalui proses *flipping*.



**Gambar 3. 4** Flipping Citra Tanaman Air

b. *Rotate*

Rotate atau rotasi yaitu proses memutar gambar pada sudut tertentu untuk menciptakan variasi dalam orientasi. Disini penulis melakukan beberapa rotasi yaitu pada sudut  $90^\circ$  searah jarum jam, rotasi pada sudut  $90^\circ$  berlawanan jarum jam, rotasi pada sudut  $-15^\circ$ , dan rotasi pada sudut  $15^\circ$ . Berikut merupakan citra yang telah melalui proses rotasi.



**Gambar 3. 5** Rotate Citra Tanaman Air

c. *Saturation*

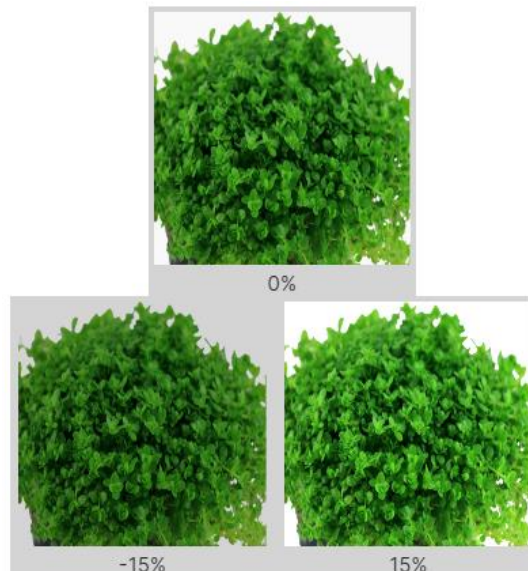
*Saturation* atau saturasi yaitu proses untuk mengubah Tingkat ketajaman warna pada data citra. Disini penulis menerapkan nilai saturasi dari -25% hingga 25%. Berikut contoh penerapan saturasi pada citra.



**Gambar 3. 6** Saturasi Citra Tanaman Air

d. *Brightness*

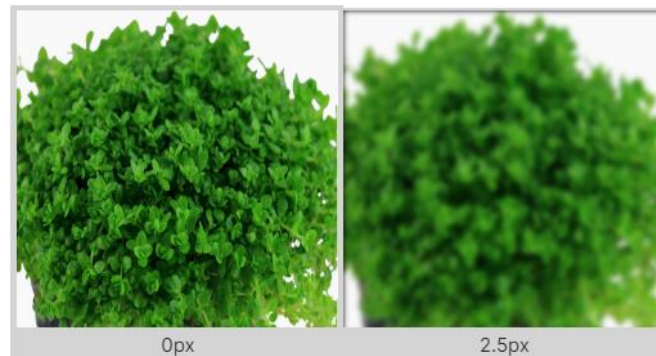
*Brightness* merupakan Tingkat kecerahan suatu data citra. Modifikasi pada nilai kecerahan berguna untuk mensimulasikan kondisi cahaya tertentu yang berguna untuk meningkatkan akurasi dari data citra tersebut. Penulis menerapkan nilai dari rentang -15% hingga 15% untuk nilai kecerahan data citra. Penerapannya pada data citra dapat di lihat pada gambar 3.7 berikut.



**Gambar 3. 7** Kecerahan Citra Tanaman Air

e. *Blur*

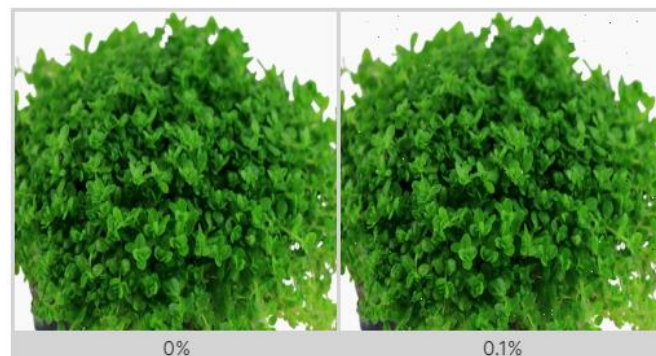
*Blur* merupakan teknik dalam pengolahan citra yang digunakan untuk mengurangi ketajaman dan detail dalam gambar, menghasilkan efek yang lebih halus atau buram. Disini penulis menggunakan Gaussian Blur dengan nilai sebesar 2,5px. Berikut penerapan *blur* pada data citra.



**Gambar 3. 8** Blur Citra Tanaman Air

f. *Noise*

Menambahkan *noise* atau gangguan acak berguna untuk membuat model lebih tahan terhadap gangguan di dunia nyata. Penulis menerapkan 0,1% untuk nilai gangguan acak. Berikut merupakan contoh penerapan *noise* pada data citra.



**Gambar 3. 9** Noise Citra Tanaman Air

### 3.2.1.3. Image Training with YOLOv5

Dataset citra gambar yang telah melalui proses augmentasi akan melalui proses training menggunakan pra pembelajaran algoritma YOLOv5. Tahap ini bertujuan untuk melatih algoritma sehingga dapat dihasilkan model terspesifikasi yang dapat mengklasifikasikan tanaman air berdasarkan kebutuhan cahayanya.



Seperti yang dapat dilihat pada gambar (arsitektur YOLOv5), dalam proses pendeteksian objek, algoritma YOLOv5 dibagi atas 3 bagian utama, yaitu backbone, neck, dan head.

#### 3.2.1.3.1. *backbone*

Struktur Focus pada CSPDarknet melibatkan operasi empat irisan. Pada gambar input berukuran  $224 \times 224 \times 3$ , keempat gambar tersebut akan diiris menjadi empat bagian berukuran  $112 \times 112 \times 3$ . Melalui operasi concat, keempat irisan ini akan digabungkan sehingga menghasilkan output berukuran  $112 \times 112 \times 12$ . Berikutnya, gambar tersebut akan melewati lapisan konvolusi dengan 32 kernel dan menghasilkan output berupa *feature map* berukuran  $208 \times 208 \times 32$ . Maka hasil ini akan dimasukkan ke lapisan konvolusi berikutnya melalui proses batch normalization dan leaky ReLU. Pada tahap ini, fitur-fitur penting pada citra gambar tanaman air akan diekstraksi dan dikirim ke modul neck. Prosesnya secara sederhana dapat dilihat pada gambar 3.10 berikut.

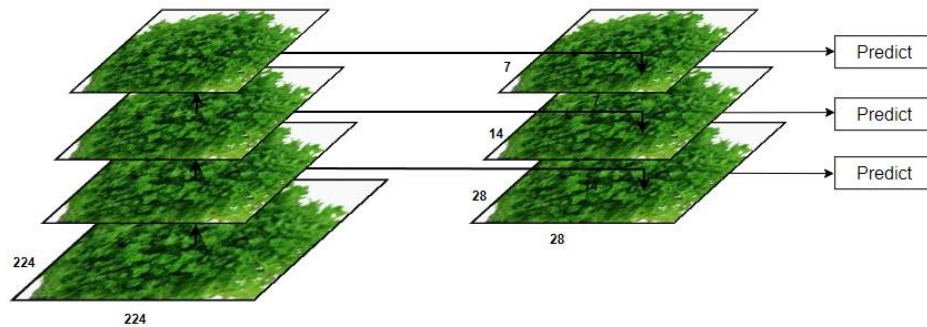


**Gambar 3. 10** *Feature Map*

#### 3.2.1.3.2. *Neck*

Setiap tahap dari *feature pyramid network* algoritma YOLOv5 menggunakan *feature map* dari tahap sebelumnya sebagai input dan memprosesnya dengan lapisan konvolusi  $3 \times 3$ . Output dari proses ini berikutnya ditambahkan ke peta fitur dari tahap yang sama pada jalur top-down melalui koneksi secara horizontal. Hal ini menghasilkan piramida fitur dengan tiga ukuran berbeda berdasarkan faktor skala 32, 16, dan 8. Dengan demikian, untuk gambar input berukuran  $224 \times 224$ , piramida fitur yang dihasilkan memiliki skala masing-masing

7 x 7, 14 x 14, dan 28 x 28. Prosesnya secara sederhana dapat dilihat pada gambar 3.11 berikut.



**Gambar 3. 11** *Feature Pyramid Network*

#### 3.2.1.3.3. *Head*

Lapisan YOLO pada modul *head* memprediksi *feature map* dengan tiga ukuran berbeda (7 x 7, 14 x 14, dan 28 x 28), memungkinkan model untuk mendeteksi objek dalam berbagai ukuran kecil, sedang, dan besar. Pada bagian ini, kotak deteksi dihasilkan, yang menunjukkan kategori, koordinat, dan *confidence score*.

#### 3.2.1.4. *Learned Model*

Setelah melewati proses training menggunakan *pretrained* algoritma YOLO, maka akan dihasilkan file akhir berupa .pt. Model ini nantinya akan digunakan untuk melakukan proses klasifikasi dalam suatu aplikasi terhadap tanaman air berdasarkan kebutuhan cahayanya.

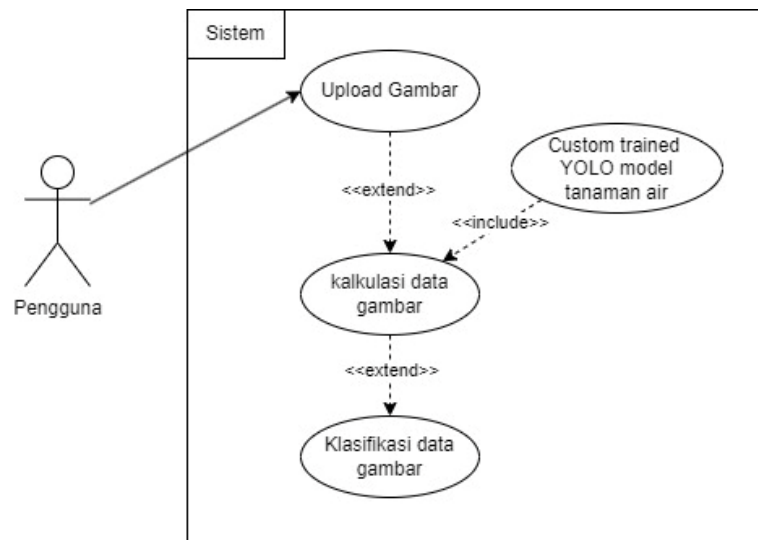
#### 3.2.1.5. *Web Application*

Berikutnya model akan digunakan dalam aplikasi untuk melakukan proses klasifikasi terhadap tanaman air. Aplikasi yang didesain berbasis web dan menggunakan bahasa pemrograman Python dengan menggunakan Flask framework. Output dari aplikasi ini yaitu berupa hasil prediksi beserta nilai akurasi terhadap citra tanaman air berkebutuhan cahaya rendah, sedang, ataupun tinggi.



### 3.2.2. Use Case Diagram

*Use case diagram* berguna untuk memberikan gambaran tentang fungsionalitas sistem dari sudut pandang pengguna atau aktor. Diagram *use case* membantu dalam pemahaman kebutuhan pengguna dan menyediakan pandangan yang jelas tentang fungsi-fungsi yang harus disediakan oleh sistem.

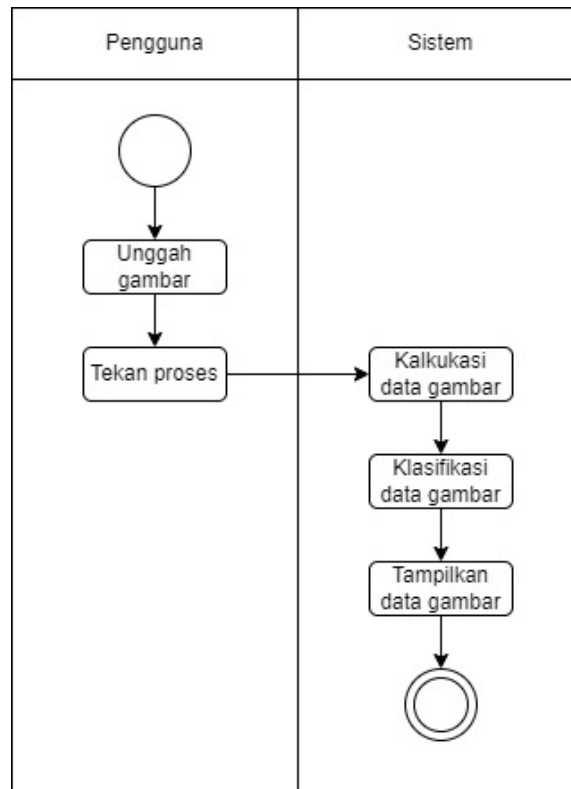


**Gambar 3. 12** *Use Case Diagram*

Berdasarkan gambar 3.12 diatas, proses dimulai dengan pengguna mengunggah gambar. Kemudian system akan menggunakan metode YOLO yang telah di training menggunakan dataset tanaman air. Data gambar kemudian akan dilakukan perhitungan berdasarkan model YOLO yang telah dikostumasi. Barulah system akan mengeluarkan hasil kalkulasi yaitu berupa Tingkat kebutuhan cahaya dari data gambar yang diunggah sebelumnya.

### 3.2.3. Activity Diagram

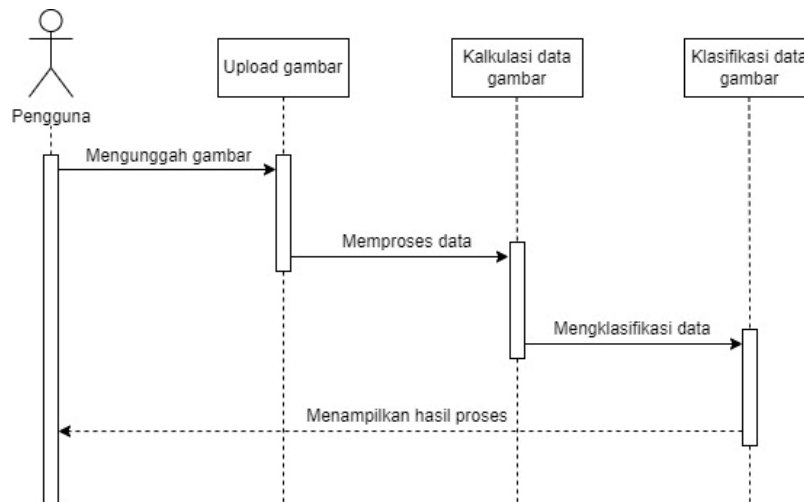
*Activity Diagram* merupakan diagram yang menggambarkan alur kerja atau aktivitas dalam suatu proses. Disini dijelaskan urutan langkah-langkah yang harus dilakukan dalam suatu proses, serta keputusan atau percabangan yang mungkin terjadi selama proses. Diagram *Activity* sistem ini bisa dilihat pada gambar 3.13 berikut.



**Gambar 3. 13** *Activity Diagram*

#### 3.2.4. *Sequence Diagram*

*Sequence diagram* berguna untuk menggambarkan interaksi antara objek-objek dalam sebuah sistem berdasarkan urutan waktu tertentu. *Sequence diagram* menjelaskan bagaimana proses dari objek-objek yang saling berinteraksi didalam sebuah skenario atau proses. *Sequence diagram* dari sistem ini dapat dilihat pada gambar 3.14 berikut.



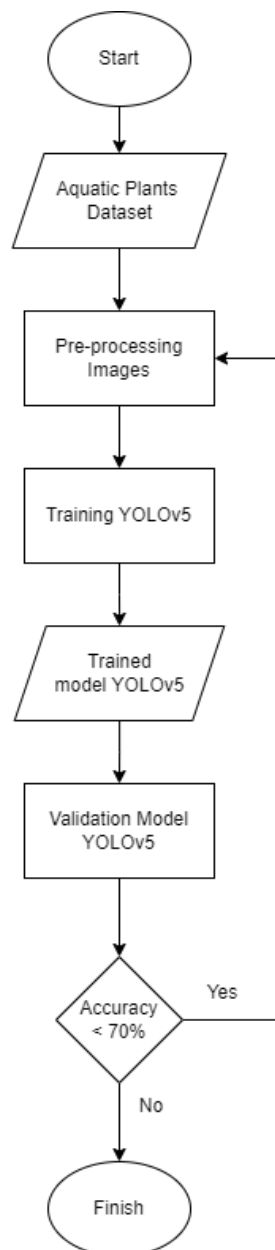
**Gambar 3. 14** *Sequence Diagram*

### 3.3. Flowchart

Flowchart adalah representasi visual dari urutan langkah-langkah atau proses. Flowchart menggunakan simbol-simbol grafis untuk menggambarkan alur kerja atau prosedur dari awal hingga akhir. Flowchart digunakan untuk menggambarkan proses algoritma atau prosedur secara visual dan mudah dipahami.

#### 3.3.1. Flowchart Training Dataset

Flowchart pada Gambar 3.15 menjelaskan proses *training dataset*. Proses diawali dengan mengumpulkan berbagai citra tanaman air yang akan dijadikan dataset. Kemudian dataset tanaman air tersebut akan masuk dan melalui pra-pemrosesan gambar. Selanjutnya dataset tersebut akan memasuki algoritma YOLOv5 untuk melakukan proses pembelajaran lebih lanjut. Setelah proses pembelajaran selesai, dilakukan validasi terhadap model yang telah dilatih. Setelah itu, dilakukan proses pengecekan akurasi. Jika akurasi dibawah 70%, maka akan dilakukan proses pembelajaran model kembali, jika akurasi sudah diatas 70%, maka proses pembelajaran model selesai.

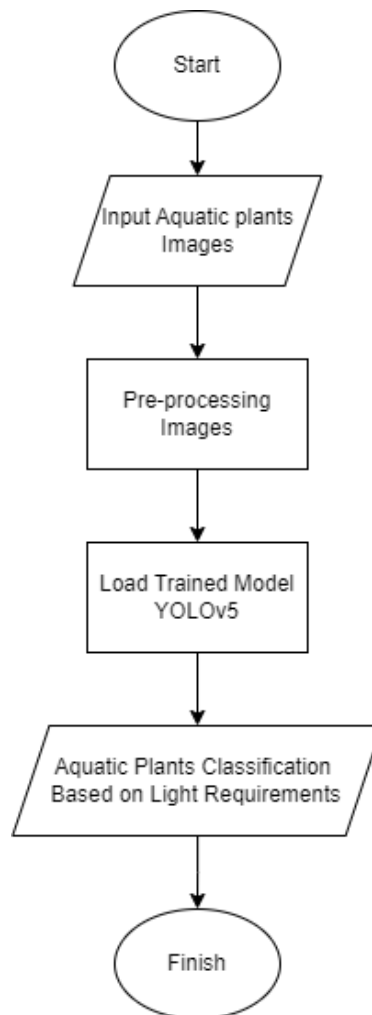


**Gambar 3. 15** Flowchart Training Dataset

### 3.3.2. Flowchart Klasifikasi

Flowchart pada gambar 3.16 menjelaskan sistem klasifikasi menggunakan model YOLOv5 yang telah dilatih. Sistem diawali dengan menginput citra tanaman air. Citra akan melalui pra-pemrosesan terlebih dahulu untuk ekstraksi fitur. Selanjutnya barulah data yang telah diekstraksi memasuki model YOLOv5 yang telah dilatih. Terakhir akan dihasilkan informasi hasil klasifikasi terhadap citra yang

diuji berdasarkan kebutuhan cahayanya baik itu berkebutuhan cahaya tinggi, sedang, atau rendah.



**Gambar 3. 16** Flowchart Klasifikasi

### 3.3.3. Perancangan Antarmuka (*Interface*)

Perancangan antarmuka (*interface*) merupakan tahap yang sangat krusial sebelum pembuatan sistem dimulai. Perancangan antarmuka (*interface*) memberikan gambaran tentang tampilan sistem serta komponen-komponen yang ada di dalamnya. Dengan perancangan antarmuka (*interface*) yang baik, pengguna akan lebih mudah dalam menggunakan sistem.

### 3.3.4. Rancangan Halaman Depan

Halaman depan atau yang biasa disebut *homepage* merupakan halaman yang pertama kali dilihat saat pengunjung mengakses Alamat suatu situs web. Rancangan halaman depan ini akan menampilkan informasi seperti tombol navigasi “Klasifikasi” dan “Tanaman Air”. Dibawahnya terdapat form untuk penginputan citra tanaman air. Form “Preview citra” akan menampilkan citra gambar jika pengguna telah menginputkan data citranya. Berikutnya terdapat “Tombol proses” yang berguna untuk memproses data citra yang telah dimasukkan. Berikut merupakan rancangan halaman depan dari aplikasi klasifikasi citra tanaman air.

Klasifikasi	Tanaman Air	
<div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: 80%;"> <div style="border: 1px solid black; height: 80px; margin-bottom: 10px; display: flex; align-items: center; justify-content: center;">           Preview citra jika telah di input         </div> <div style="border: 1px solid black; height: 40px; margin-bottom: 10px; display: flex; align-items: center; justify-content: center;">           Form input citra         </div> <div style="border: 1px solid black; height: 30px; display: flex; align-items: center; justify-content: center;">           Tombol proses         </div> </div>		

**Gambar 3. 17** Halaman Depan Klasifikasi

### 3.3.1. Rancangan Halaman Hasil

Rancangan halaman ini akan berisi tampilan sederhana hasil data citra yang telah diproses dari halaman klasifikasi. Halaman ini akan menampilkan klasifikasi dari citra tanaman air yang telah diproses beserta nilai *confidence*-nya. Rancangan halaman hasil dapat dilihat pada gambar 3.18 berikut.

Klasifikasi	Tanaman Air	
Menampilkan hasil klasifikasi citra beserta nilai confidence-nya		

**Gambar 3. 18** Rancangan Halaman Hasil

### 3.3.2. Rancangan Halaman Ringkasan Seputar Aquascape

Pada halaman ini akan ditampilkan pengetahuan umum tentang *aquascape* beserta beberapa contoh tanaman air yang populer digunakan untuk menghiasinya. Dibawah tombol navigasi akan dirancang halaman untuk menjelaskan secara ringkas apa itu *aquascape*. Lalu dibawahnya akan ditampilkan beberapa contoh citra tanaman air berdasarkan kebutuhan cahayanya. Rancangannya dapat dilihat pada gambar 3.19 berikut.

Klasifikasi	Tanaman Air	
Keterangan umum tentang tanaman air dalam aquascape		
contoh tanaman air beserta atributnya	contoh tanaman air beserta atributnya	contoh tanaman air beserta atributnya

**Gambar 3. 19** Rancangan Halaman Tentang Aquascape

## BAB 4

### IMPLEMENTASI DAN PENGUJIAN

#### 4.1. Implementasi Sistem

Implementasi sistem merupakan tahap dimana penulis membangun dan melakukan penerapan terhadap desain yang telah dirancang sebelumnya. Beberapa komponen perangkat keras dan perangkat lunak menjadi kebutuhan penting dalam proses implementasi sistem. Berikut merupakan perangkat yang penulis gunakan dalam penelitian ini.

1. Laptop Asus Vivobook M409D
2. Processor AMD Ryzen 3 3200U with Radeon Vega Mobile Gfx 2.60 GHz
3. Storage SSD 256 GB
4. RAM 12 GB
5. Sistem Operasi Windows 10 Pro
6. Visual Studio Code
7. Google Colab Notebooks
8. Python 3.10.0
9. Library : Flask, Tensorflow, Torch, Numpy, Matplotlib, Pandas

##### 4.1.1. Implementasi Data

Data yang digunakan pada penelitian ini merupakan citra tanaman air yang dibagi berdasarkan kebutuhan cahayanya dalam aquascape. Proses pengumpulan citra yaitu dengan melakukan penangkapan gambar terhadap citra tanaman air menggunakan kamera *smartphone* di toko aquascape terdekat disekitaran kawasan Universitas Sumatra Utara. Untuk menambahkan data citra yang masih kurang lengkap terhadap beberapa jenis tanaman air tertentu, maka penulis juga melakukan pengumpulan citra gambar pada toko *online* dari pencarian di internet. Pembagian data mentah atau data primer dapat dilihat pada tabel tabel 4.1 berikut.



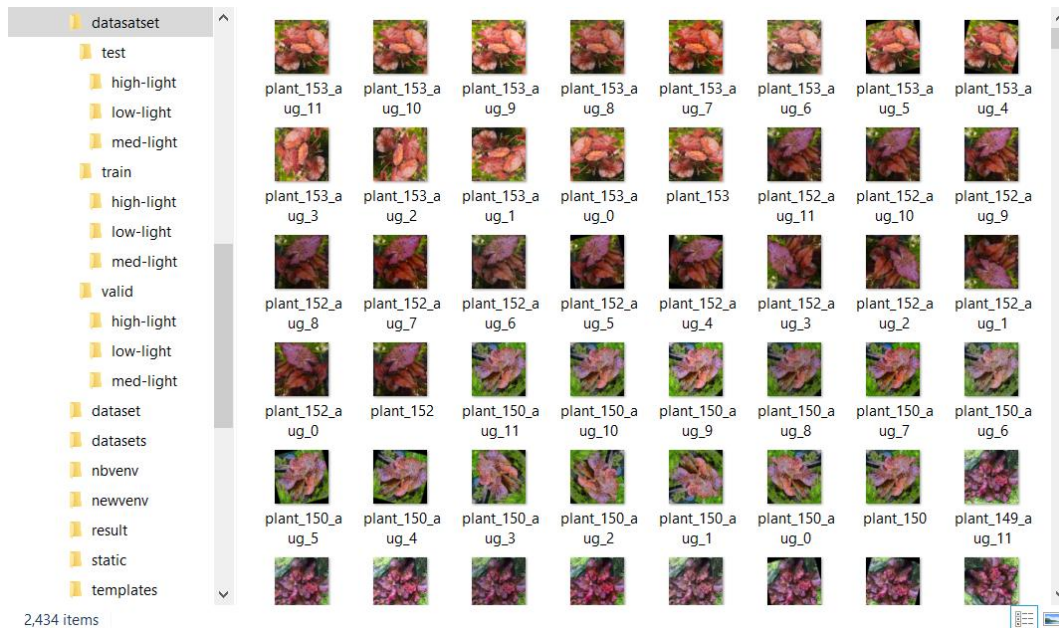
**Tabel 4. 1** Pembagian Data primer/mentah

No.	Tanaman Air	Jumlah Data
1.	Berkebutuhan Cahaya Rendah	75
2.	Berkebutuhan Cahaya Sedang	78
3.	Berkebutuhan Cahaya Tinggi	73
Total		226

Berikutnya dilakukan pra pemrosesan gambar untuk memperbanyak data. Untuk data test diambil sebanyak 20% dari 226 data primer, yaitu 42 citra untuk data test. Kemudian sisa 184 citra memasuki tahap augmentasi dan menghasilkan data sebanyak 2392 citra, yang mana sebanyak 20% atau 480 citra merupakan data valid, dan 80% atau 1912 citra merupakan data train. Persebaran data sintetis dapat dilihat pada tabel 4.2 berikut.

**Tabel 4. 2** Tabel data sintesis

	Train	Valid	Test	Total
Cahaya Rendah	634	159	14	807
Cahaya Sedang	665	167	14	846
Cahaya Tinggi	613	154	14	781
total	1912	480	42	2434



**Gambar 4. 1** Gambar dataset

#### 4.1.2. Implementasi *Training*

Implementasi *training* atau pelatihan menggunakan algoritma YOLOv5. Algoritma ini dipilih karna cukup populer dan telah banyak dilakukan deteksi citra dengan akurasi yang baik. Untuk proses pelatihan datanya sendiri penulis menggunakan Google Colab Notebooks. Pengaturan *fine-tuning* dan *hyperparameter* penulis menggunakan nilai sebagai berikut:

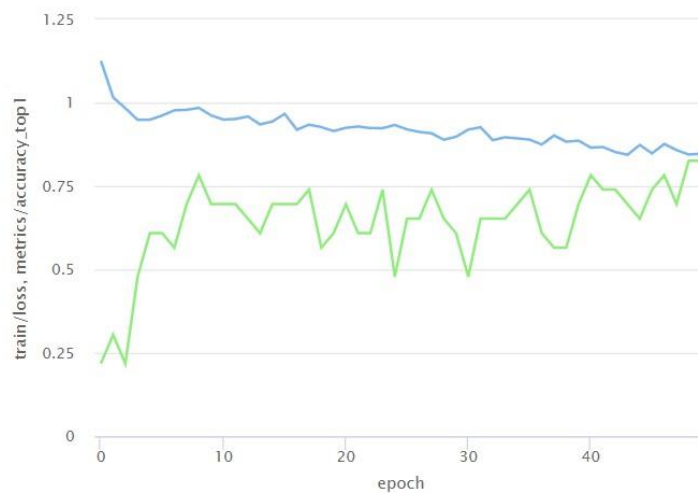
1. Model : yolov5s-cls.pt
2. Epochs : 50
3. Image size : 224
4. Learning Rate : 0.001

Untuk mencari pelatihan mana yang menghasilkan model terbaik, penulis melakukan *fine-tuning* pada *hyperparameter batch size*, yaitu dengan nilai 32, 64, dan 128. Nilai *top1\_accuracy* tertinggi terhadap masing-masing *batch size* dapat dilihat pada tabel 4.2 berikut.

**Tabel 4. 3** Validasi

No.	<i>Batch size</i>	<i>Top1 accuracy</i>
1.	32	0.81
2.	64	0.762
3.	128	0.738

Berdasarkan tabel, maka penulis memutuskan untuk menggunakan model latih dengan *batch size* 32 dikarenakan telah menghasilkan validasi *top1 accuracy* dengan nilai 0.81. Model terbaik dengan *batch size* 32 didapatkan pada *epoch* ke-48 dengan nilai *top1 accuracy* 0.81 dan *train/loss* 0.884. Grafik hasil pelatihan dataset dapat dilihat pada gambar 4.2 berikut.

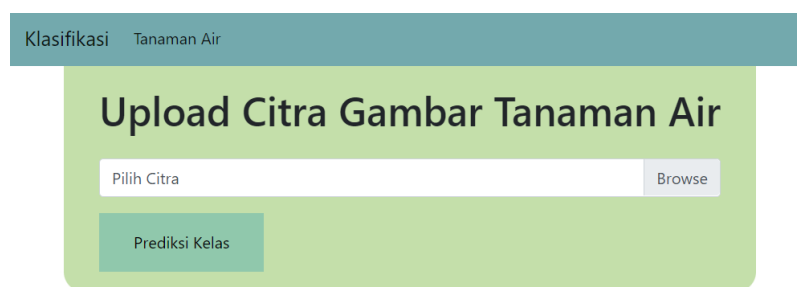
**Gambar 4. 2** Grafik Hasil Pelatihan Dataset

#### 4.1.3. Implementasi Aplikasi

Implementasi aplikasi merupakan tahapan dimana penulis menerapkan rancangan sistem yang telah didesain pada bab 3. Rancangan sistem yang dibangun terdiri dari 3 komponen utama, yaitu halaman utama yang merupakan halaman klasifikasi citra, halaman hasil untuk menampilkan output klasifikasi, dan halaman keterangan umum seputar tanaman air pada aquascape.

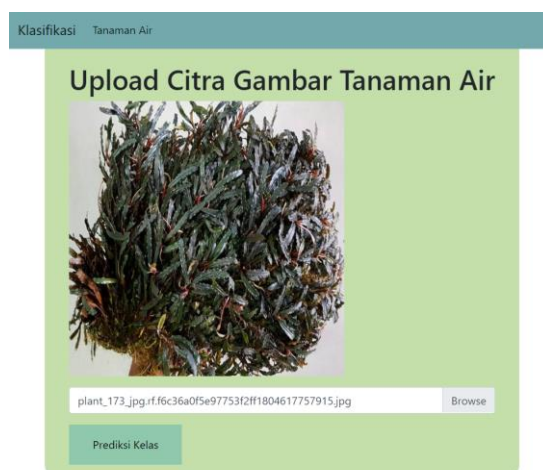
#### 4.1.3.1. Antarmuka Halaman Klasifikasi

Halaman utama atau biasa disebut *homepage* pada aplikasi web ini merupakan halaman klasifikasi. Halaman ini berguna untuk menginputkan citra tanaman air yang nantinya akan diproses agar diketahui hasil klasifikasinya. Halaman ini terdiri dari 2 tombol navigasi di atasnya. Di bawahnya terdapat form untuk menginputkan gambar. Tampilan halaman antarmuka dapat dilihat pada gambar 4.3 berikut.



**Gambar 4. 3** Halaman Antarmuka

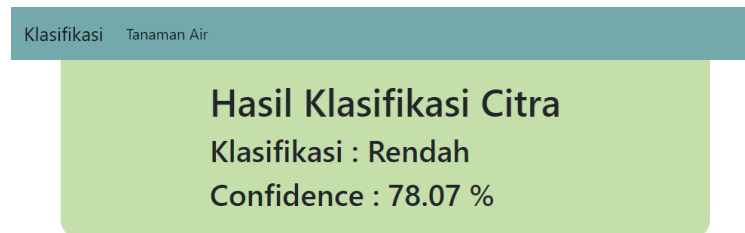
Jika pengguna menginputkan gambar, maka gambar yang diinputkan akan ditampilkan terlebih dahulu untuk memastikan bahwa apa gambar yang dimasukkan telah benar. Tampilan saat gambar telah diinput dapat dilihat pada gambar 4.4 berikut.



**Gambar 4. 4** Halaman saat citra dimasukkan

#### 4.1.3.2. Antarmuka Halaman Hasil

Halaman ini menampilkan hasil dari klasifikasi citra tanaman air yang dimasukkan pada halaman klasifikasi. Halaman ini akan menampilkan hasil prediksi dari kelas tanaman air yang dimasukkan beserta nilai *confidence*-nya. Tampilan halaman ini dapat dilihat pada gambar 4.5 berikut.



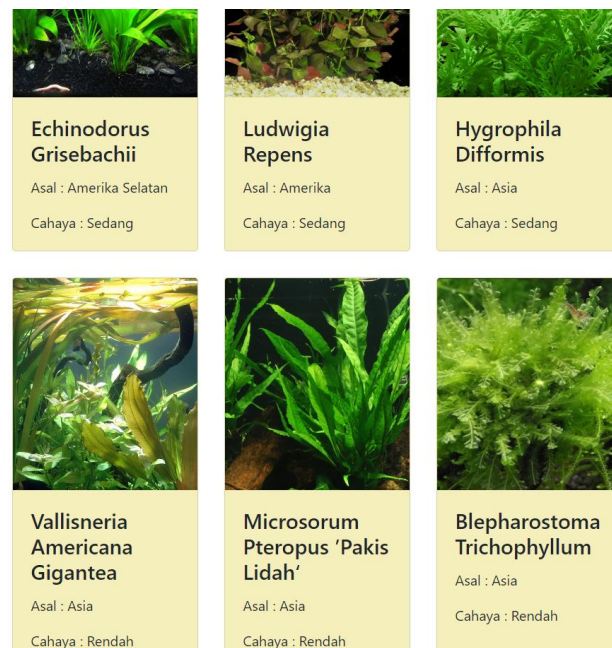
**Gambar 4. 5** Halaman Hasil

#### 4.1.3.3. Antarmuka Halaman Tentang Tanaman Air

Halaman ini menampilkan informasi seputar tanaman air pada aquascape. Halaman ini terdiri dari penjelasan secara umum apa itu aquascape, dan menampilkan beberapa jenis tanaman air yang populer beserta atributnya seperti asal tanaman tersebut serta Tingkat kebutuhan cahayanya. Antarmuka halaman dapat dilihat pada gambar 4.6 dan gambar 4.7 berikut.



**Gambar 4. 6** Penjelasan Tanaman Air



**Gambar 4. 7** Contoh Tanaman Air

## 4.2. Pengujian Sistem

Pengujian sistem yang akan penulis lakukan dalam penelitian ini diantaranya pengujian *black box* dan pengujian validitas terhadap hasil yang diharapkan pada sistem yang diuji.

### 4.2.1. Pengujian *Black Box*

Pengujian *black box* merupakan suatu pengujian yang digunakan untuk mengetahui apakah terdapat kesalahan pada sistem yang dibangun. Pengujian ini berguna untuk mengetahui nilai fungsionalitas apakah sistem sudah berjalan dengan baik atau belum. Berikut tabel pengujian *black box* pada sistem klasifikasi tanaman air pada aquascape berdasarkan kebutuhan cahayanya menggunakan algoritma YOLO. Hasil pengujian dapat dilihat pada tabel 4.4 berikut.

**Tabel 4. 4** Pengujian *Black Box*

No.	Input	Proses	Output	Hasil
1.	Menekan tombol proses	Melakukan proses walaupun citra belum diunggah	Menampilkan notifikasi bahwa	Berhasil

	sebelum citra diunggah		citra belum diunggah	
2.	Unggah citra dengan cara drag and drop	Citra ditarik dari folder dan diarahkan ke form input aplikasi web	Citra masuk dan langsung ditampilkan di aplikasi web	Berhasil
3.	Menekan tombol proses setelah citra diunggah	Melakukan proses prediksi kelas pada sistem aplikasi	Menampilkan hasil prediksi dari citra yang diunggah	Berhasil
4.	Menekan tombol navigasi	Berganti antarmuka antara halaman klasifikasi dan halaman penjelasan umum	Halaman antarmuka berganti	Berhasil

#### 4.2.2. Pengujian Validitas

Pengujian validitas merupakan pengujian terhadap sistem untuk membandingkan antara nilai faktual dengan nilai yang dihasilkan oleh sistem yang dibangun. Data citra terdapat 3 kelas yang masing-masing memiliki sekitar 70 citra. Peneliti mengambil 20% dari total citra per kelasnya secara acak sebagai data uji. Diambilah 14 citra tanaman air secara random per kelasnya, maka total data ujinya adalah  $14 \times 3$  yaitu 42 citra tanaman air untuk divalidasi. Berikut hasil pengujian citra tanaman air untuk kelas berkebutuhan cahaya rendah pada tabel 4.5 berikut.

**Tabel 4. 5** Pengujian Validitas

No.	Citra	Hasil Aktual	Hasil Sistem	Hasil Validitas
1.	plant_010	Rendah	Rendah	Benar
2.	plant_013	Rendah	Tinggi	Salah
3.	Plant_017	Rendah	Rendah	Benar
4.	Plant_020	Rendah	Sedang	Salah
5.	Plant_029	Rendah	Rendah	Benar

No.	Citra	Hasil Aktual	Hasil Sistem	Hasil Validitas
6.	Plant_079	Rendah	Rendah	Benar
7.	Plant_099	Rendah	Sedang	Salah
8.	Plant_103	Rendah	Sedang	Salah
9.	Plant_173	Rendah	Rendah	Benar
10.	Plant_179	Rendah	Rendah	Benar
11.	Plant_187	Rendah	Rendah	Benar
12.	Plant_190	Rendah	Rendah	Benar
13.	Plant_195	Rendah	Rendah	Benar
14.	Plant_200	Rendah	Rendah	Benar
15.	Plant_005	Sedang	Sedang	Benar
16.	Plant_006	Sedang	Sedang	Benar
17.	Plant_033	Sedang	Sedang	Benar
18.	Plant_044	Sedang	Sedang	Benar
19.	Plant_056	Sedang	Sedang	Benar
20.	Plant_065	Sedang	Sedang	Benar
21.	Plant_070	Sedang	Sedang	Benar
22.	Plant_077	Sedang	Sedang	Benar
23.	Plant_088	Sedang	Sedang	Benar
24.	Plant_093	Sedang	Sedang	Benar
25.	Plant_111	Sedang	Sedang	Benar
26.	Plant_115	Sedang	Sedang	Benar
27.	Plant_146	Sedang	Sedang	Benar
28.	Plant_151	Sedang	Sedang	Benar
29.	Plant_048	Tinggi	Tinggi	Benar
30.	Plant_061	Tinggi	Sedang	Salah
31.	Plant_122	Tinggi	Tinggi	Benar
32.	Plant_129	Tinggi	Tinggi	Benar
33.	Plant_133	Tinggi	Tinggi	Benar
34.	Plant_156	Tinggi	Tinggi	Benar



No.	Citra	Hasil Aktual	Hasil Sistem	Hasil Validitas
35.	Plant_164	Tinggi	Tinggi	Benar
36.	Plant_168	Tinggi	Sedang	Salah
37.	Plant_171	Tinggi	Tinggi	Benar
38.	Plant_204	Tinggi	Tinggi	Benar
39.	Plant_208	Tinggi	Sedang	Salah
40.	Plant_213	Tinggi	Sedang	Salah
41.	Plant_220	Tinggi	Tinggi	Benar
42.	Plant_226	Tinggi	Tinggi	Benar

Setelah melakukan validasi terhadap data citra pada tabel, maka selanjutnya dilakukan pengujian evaluasi performa menggunakan *confusion matrix*. *Confusion Matrix* berguna untuk mengukur performa berupa sensitif, spesifik, presisi dan akurasi. Hasil pengujian validitas terhadap metode YOLO untuk klasifikasi dapat dilihat pada tabel berikut.

**Tabel 4. 6** Tabel Confusion Matrix

Kelas Aktual	Kelas Prediksi			
		Rendah	Sedang	Tinggi
	Rendah	10	3	1
	Sedang	0	14	0
	Tinggi	0	4	10

$$TP = 10 + 14 + 10 = 34$$

$$Accuracy = \frac{34}{42} \times 100\% = 80,95\%$$

**Tabel 4. 7** Tabel Recall

	Rendah	Sedang	Tinggi
TP	10	14	10
FN	4	0	4
Recall	$10 / (10 + 4) = 0,71$	$14 / (14 + 0) = 1$	$10 / (10 + 4) = 0,71$

$$Recall = \frac{0,71 + 1 + 0,71}{3} \times 100\% = 80,67\%$$

**Tabel 4. 8** Tabel Presisi

	Rendah	Sedang	Tinggi
TP	10	14	10
FP	0	7	1
Precision	$10 / (10 + 0) = 1$	$14 / (14 + 7) = 0,67$	$10 / (10 + 1) = 0,9$

$$Precision = \frac{1 + 0,67 + 0,9}{3} \times 100\% = 85,67\%$$

Berdasarkan persebaran pengujian data yang berjumlah 42 citra pada tabel *confusion matrix* maka didapatkanlah nilai *acuracy* sebesar 80,95%, nilai *recall* sebesar 80,67%, dan nilai *precision* sebesar 85,67%.

## **BAB 5**

### **KESIMPULAN DAN SARAN**

#### **5.1. Kesimpulan**

Berdasarkan hasil pengujian dari penelitian tentang klasifikasi tanaman air pada aquscapes dengan metode YOLO, maka diperoleh beberapa kesimpulan yaitu:

1. Sistem klasifikasi tanaman air berdasarkan kebutuhan cahayanya telah berhasil dibangun menggunakan metode YOLO.
2. Jumlah dan pemilihan dataset sangat penting dalam meningkatkan kualitas dari data yang direpresentasikan.
3. Validasi kinerja model penting untuk menentukan pemilihan nilai *hyperparameter* yang tepat sehingga menghasilkan model yang baik.
4. Walaupun hasil prediksi benar, namun ada beberapa citra dengan nilai *confidence* yang masih rendah dikarenakan beberapa tanaman air contohnya seperti tanaman air yang berkebutuhan cahaya rendah ada yang memiliki warna cenderung terang maupun sebaliknya.
5. Hasil pengujian sistem dengan data sampel berjumlah 42 citra dapat menunjukkan nilai *accuracy* sebesar 80,95%, nilai *recall* sebesar 80,67%, dan nilai *precision* sebesar 85,67%.

#### **5.2. Saran**

Dari hasil penelitian yang telah penulis lakukan, terdapat beberapa saran sebagai acuan untuk pengembangan dan perbaikan pada penelitian berikutnya, yaitu:

1. Pada penelitian selanjutnya diharapkan menggunakan metode YOLO generasi terbaru.
2. Dataset terkait tanaman air dapat diperbanyak lagi untuk penelitian selanjutnya agar diharapkan dapat menghasilkan nilai *confidence* yang lebih baik.
3. Menerapkan metode *deep learning* lainnya untuk membandingkan mana hasil yang lebih baik.

4. Sistem dapat melakukan klasifikasi secara *realtime* dengan platform yang beragam.

## DAFTAR PUSTAKA

- Akhyar, F., Novamizanti, L., & Riantiarni, T. (2022). Sistem Inspeksi Cacat pada Permukaan Kayu menggunakan Model Deteksi Obyek YOLOv5. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 10(4), 990. <https://doi.org/10.26760/elkomika.v10i4.990>
- Dawami, H., Rachmawati, E., & Sulistiyo, M. D. (2023). Deteksi Penggunaan Masker Wajah Menggunakan YOLOv5. *E-Proceeding of Engineering*, 10, 1746–1764.
- Liu, H., Sun, F., Gu, J., & Deng, L. (2022). SF-YOLOv5: A Lightweight Small Object Detection Algorithm Based on Improved Feature Fusion Mode. *Sensors*, 22(15). <https://doi.org/10.3390/s22155817>
- Mahesh, B. (2018). Machine Learning Algorithms-A Review. *International Journal of Science and Research*. <https://doi.org/10.21275/ART20203995>
- Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., Shyu, M. L., Chen, S. C., & Iyengar, S. S. (2018). A survey on deep learning: Algorithms, techniques, and applications. In *ACM Computing Surveys* (Vol. 51, Issue 5). Association for Computing Machinery. <https://doi.org/10.1145/3234150>
- Setiana Riva, L. (2023). Deteksi Penyakit Tanaman Cabai Menggunakan Algoritma YOLOv5 Dengan Variasi Pembagian Data. *Jurnal Pengembangan IT (JPIT)*, 8(3), 248–254.
- Wang, H., Shang, S., Wang, D., He, X., Feng, K., & Zhu, H. (2022). Plant Disease Detection and Classification Method Based on the Optimized Lightweight YOLOv5 Model. *Agriculture (Switzerland)*, 12(7). <https://doi.org/10.3390/agriculture12070931>
- Wibowo, A., Lusiana, L., & Dewi, T. K. (2023). Implementasi Algoritma Deep Learning You Only Look Once (YOLOv5) Untuk Deteksi Buah Segar Dan

Busuk. *Paspalum: Jurnal Ilmiah Pertanian*, 11(1), 123–130.  
<https://doi.org/10.35138/paspalum.v11i1.489>

Widjaja, T. (2013). *Aquascape, Pesona Taman dalam Aquarium*.

## **CURRICULUM VITAE**

### **DATA DIRI**

---



Nama : Mailan Roni Saputra  
Tempat/Tanggal Lahir : Tabek Patah, 9 Mei 1998  
Jenis Kelamin : Laki-laki  
Agama : Islam  
Kewarganegaraan : Indonesia  
Alamat : Jl. Raya Batusangkar-Bukittinggi  
Km 16, Pos 27263  
Telp/Hp : 085830666630  
Email : mailan.ronny09@gmail.com

### **RIWAYAT PENDIDIKAN**

---

2004 – 2010 : SD Negeri 04 Tabek Patah  
2010 – 2013 : SMP Negeri 1 Batusangkar  
2013 – 2016 : SMA Negeri 1 Batusangkar  
2017 – 2024 : S-1 Ilmu Komputer Universitas Sumatera Utara

### **PRESTASI**

---

2020 : Juara 3 (Perunggu) GemasTIK XIII Divisi Pengembangan Aplikasi Permainan

### **PENGALAMAN ORGANISASI**

---

2017 : Anggota Ikatan Mahasiswa Imam Bonjol (IMIB) USU  
2019 : Anggota Game Dev ITIKOM USU

## **PENGALAMAN KERJA**

---

2020 : Praktik Kerja Lapangan di BP2JK (Balai Pelaksana  
Pemilihan Jasa Konstruksi) Sumatera Utara

## **KEAHLIAN**

---

Bahasa : Indonesia, Inggris

Pemrograman : Python, HTML, JavaScript

Aplikasi : Unity, MS. Office, Visual Studio Code