

**IDENTIFIKASI KEBUTUHAN CO₂ TANAMAN AIR PADA AQUASCAPE
BERDASARKAN CITRA GAMBAR MENGGUNAKAN
ALGORITMA *K-NEAREST NEIGHBORS***

SKRIPSI

FEROUZ SYAHALAM

171401108



**PROGRAM STUDI S1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2023**

IDENTIFIKASI KEBUTUHAN CO₂ TANAMAN AIR PADA AQUASCAPE
BERDASARKAN CITRA GAMBAR MENGGUNAKAN
ALGORITMA *K-NEAREST NEIGHBORS*

SKRIPSI

Diajukan untuk melengkapi Tugas Akhir dan memenuhi syarat memperoleh Ijazah
Sarjana S-1 Ilmu Komputer

FEROUZ SYAHALAM

171401108



PROGRAM STUDI S1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2023

PERSETUJUAN

Judul : IDENTIFIKASI KEBUTUHAN CO₂ TANAMAN AIR PADA
AQUASCAPE BERDASARKAN CITRA GAMBAR
MENGUNAKAN ALGORITMA *K-NEAREST NEIGHBORS*

Kategori : SKRIPSI

Nama : FEROUZ SYAHALAM

Nomor Induk Mahasiswa : 171401108

Program Studi : SARJANA (S-1) ILMU KOMPUTER

Departemen : ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITASI SUMATERA UTARA

Komisi Pembimbing :

Pembimbing II



Ivan Jaya S.Si., M.Kom.

NIP. 19840707 201504 1 001

Pembimbing I

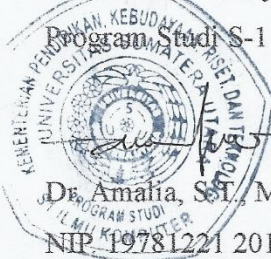


Sri Melvani Hardi, S.Kom., M.Kom.

NIP. 19880501 201504 2 006

Diketahui / Disetujui oleh

Program Studi S-1 Ilmu Komputer Ketua,



Dr. Amalia, S.T., M.T.

NIP. 19781221 201404 2 001

PERNYATAAN

IDENTIFIKASI KEBUTUHAN CO₂ TANAMAN AIR PADA AQUASCAPE
BERDASARKAN CITRA GAMBAR MENGGUNAKAN ALGORITMA *K-NEAREST*
NEIGHBORS

SKRIPSI

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

Medan, 27 Oktober 2023



Ferouz Syahalam

171401108

UCAPAN TERIMA KASIH

Segala puji atas kehadiran Allah SWT atas segala limpahan rahmat, karunia serta hidayah-Nya dan syukur yang tiada hentinya diucapkan sehingga penulis mampu menyelesaikan penyusunan skripsi ini sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer.

Pada kesempatan ini penulis telah banyak menerima dukungan dan arahan serta bimbingan dari berbagai pihak, maka dari itu penulis juga ingin mengucapkan banyak terimakasih kepada:

1. Bapak Dr. Muryanto Amin, S.Sos., M.Si. selaku Rektor Universitas Sumatera Utara.
2. Ibu Dr. Maya Silvi Lydia, B.Sc., M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.
3. Ibu Dr. Amalia, S.T., M.T. selaku Kepala Program Studi S-1 Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.
4. Ibu Sri Melvani Hardi, S.Kom. M.Kom., selaku Dosen Pembimbing I yang telah banyak memberikan bimbingan, saran, dan dukungan dalam pengerjaan skripsi ini.
5. Bapak Ivan Jaya, S.Si., M.Kom. selaku Dosen Pembimbing II yang telah banyak memberikan bimbingan, saran, dan dukungan dalam pengerjaan skripsi ini.
6. Bapak Herriyance, S.T., M.Kom. selaku Dosen Penguji I yang telah banyak memberikan bimbingan, saran, dan dukungan dalam pengerjaan skripsi ini.
7. Ibu Dian Rachmawati, S.Si., M.Kom. selaku Dosen Penguji II yang telah banyak memberikan bimbingan, saran, dan dukungan dalam pengerjaan skripsi ini.
8. 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 bermanfaat selama penulis mengikuti proses perkuliahan.
9. Orang tua tersayang Mama Laili Martini, S.E., Papa dr. Arif Yurisman, Sp.S., serta Abang Faridz Afdillah, S.H., M. Kn., dan Adik Fathira Arifah. Terima kasih untuk segala doa, serta dukungan yang selalu diberikan kepada penulis dalam menyelesaikan skripsi ini.

10. Teman-teman yang selalu memberi doa, semangat motivasi dan keceriaan kepada penulis, Charli, Rafley Husein Alwi, dan Anne Aisya Gebriella.
11. Teman-teman Gamer Go Brrrr, Aulia Ramadhan, Rizki Tanjung, Mailan Roni, Raihansyah, Kristov, dan teman-teman lain yang tidak bisa saya sebutkan satu per satu.
12. Teman-teman Kom C Ilmu Komputer 2017, Azhar Bazla, Zidane, Egi, Jeremy, dan teman-teman lain yang tidak bisa saya sebutkan satu per satu.
13. Teman-teman Online, Ihsan Aditya, Martha, Givan dan teman-teman lainnya yang tidak bisa saya sebutkan satu per satu.
14. Teman-teman penulis, Adytias Agung, Sella Septia, Siti Nur Azizah, Fanny Andira, Zahra Rusyda, M. Endi Hardanu, Nabila Husna, dan teman-teman lainnya yang tidak bisa saya sebutkan satu per satu.
15. Dan semua pihak yang terlibat baik secara langsung maupun tidak langsung yang telah banyak membantu penulis dan tidak dapat penulis sebutkan namanya satu persatu.

Semoga Allah SWT memberikan balasan kepada semua pihak dan teman-teman yang telah membantu penulis dalam menyelesaikan skripsi ini. Akhir kata, penulis berharap agar skripsi ini dapat memberikan manfaat yang baik bagi para pembaca.

ABSTRAK

Aquascape merupakan seni taman air yang menggabungkan elemen air, tanaman, dan hardscape untuk menciptakan lingkungan yang indah dan seimbang. Identifikasi tanaman air dalam aquascape dapat menjadi tugas yang rumit karena perlu memperhatikan kebutuhan masing-masing tanaman, salah satunya kebutuhan CO₂, tidak banyak masyarakat yang mengetahui hal tersebut. Pemberian CO₂ dalam pertumbuhan tanaman yang tepat mampu mengoptimalkan pertumbuhan tanaman dan efeknya tanaman tersebut akan tumbuh cepat dan tampak segar, untuk melakukan hal ini kita harus mengetahui terlebih dahulu kebutuhan CO₂ pada masing masing tanaman. Pertama, dilakukan pengumpulan data tentang tanaman air yang umum digunakan dalam aquascape beserta kebutuhan CO₂ masing-masing tanaman. Data ini kemudian dibagi menjadi dua subset, yaitu data latih dan data uji. Data latih berjumlah 45, sedangkan data yang diuji berjumlah 10. Lalu, algoritma KNN diterapkan pada data latih. Dalam langkah ini, KNN mengukur jarak antara titik data latih yang ada dengan titik data uji yang baru. Titik data uji kemudian diklasifikasikan berdasarkan mayoritas kelas dari k tetangga terdekatnya. Pemilihan nilai k yang optimal dilakukan dengan menggunakan teknik validasi silang. Setelah melatih algoritma KNN dengan data latih, dilakukan pengujian menggunakan data uji. Hasil penelitian ini menunjukkan bahwa algoritma KNN mampu mengidentifikasi kebutuhan CO₂ tanaman air pada aquascape berdasarkan citra gambar dengan akurasi yang tinggi yaitu 90%. Metode ini dapat digunakan sebagai alat bantu bagi para penghobi aquascape dalam memilih tanaman air yang sesuai dengan kebutuhan CO₂ yang tersedia.

Kata kunci: aquascape, tanaman air, CO₂, K-Nearest Neighbors, identifikasi.

ABSTRACT

IDENTIFICATION OF CO₂ REQUIREMENTS FOR AQUATIC PLANTS IN AQUASCAPE BASED ON IMAGE DATA USING K-NEAREST NEIGHBORS ALGORITHM

Aquascaping is the art of water gardening which combines the elements of water, plants, and hardscape to create a beautiful and balanced environment. Identification of aquatic plants in an aquascape can be a complicated task because it is necessary to pay attention to the needs of each plant, one of which is the need for CO₂, not many people know about this. Giving CO₂ to the right plant growth can optimize plant growth and the effect is that these plants will grow fast and look fresh, to do this we must first know the CO₂ needs of each plant. First, data collection was carried out on water plants that are commonly used in aquascapes along with the CO₂ needs of each plant. This data is then divided into two subsets, namely training data and test data. There are 45 training data and 10 tested data. Then, the KNN algorithm is applied to the training data. In this step, KNN measures the distance between the existing training data points and the new test data points. The test data points are then classified based on the majority of the classes from their k nearest neighbors. Selection of the optimal value of k is done by using cross-validation techniques. After training the KNN algorithm with training data, testing is carried out using the test data. The results of this study indicate that the KNN algorithm is able to identify CO₂ requirements of aquatic plants in aquascapes based on image data with a high accuracy of 90%. This method can be used as a tool for aquascape hobbyists in selecting aquatic plants that suit the available CO₂ needs.

Keywords: aquascape, aquatic, plants, CO₂, K-Nearest Neighbors , identification.

DAFTAR ISI

PERSETUJUAN	iii
PERNYATAAN	iv
UCAPAN TERIMA KASIH	v
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
1. BAB 1 PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah	3
1.4. Tujuan Penelitian	3
1.5. Manfaat Penelitian	4
1.6. Metode Penelitian	4
1.7. Sistematika Penulisan	5
2. BAB 2 LANDASAN TEORI	6
2.1. Aquascape	6
2.2. Fotosintesis pada Tanaman Air	7
2.3. Pengolahan Citra	8
2.3.1. Citra Biner (Monokrom)	9
2.3.2. Citra Grayscale	9
2.3.3. Citra Warna	10

2.4.	MobileNets	11
2.5.	Machine Learning.....	12
2.6.	K-Nearest Neighbors (KNN).....	15
2.7.	Javascript	18
2.8.	Framework CodeIgniter 3	19
2.9.	Confusion Matrix	19
3.	BAB 3 ANALISIS DAN PERANCANGAN.....	21
3.1.	Analisis Sistem.....	21
3.1.1.	Analisis Masalah	21
3.1.2.	Analisis Kebutuhan (<i>Requirement Analysis</i>).....	22
3.2.	Pemodelan Sistem	23
3.2.1.	General Architecture	23
3.2.1.1.	Unggah Gambar	24
3.2.1.2.	Ekstraksi Fitur	25
3.2.1.3.	<i>K-Nearest Neighbors</i>	28
3.2.1.4.	<i>Euclidean Distance</i>	29
3.2.2.	Use Case Diagram.....	31
3.2.3.	Activity Diagram	32
3.2.4.	Sequence Diagram	33
3.3.	Flowchart.....	34
3.3.1.	Flowchart Klasifikasi dengan K-Nearest Neighbors	34
3.4.	Perancangan Antarmuka (Interface).....	35
3.4.1.	Rancangan Halaman Depan.....	36
3.4.2.	Rancangan Halaman Klasifikasi Gambar	37

4.	BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM	38
4.1.	Implementasi Sistem	38
4.1.1.	Antarmuka Halaman Depan.....	39
4.1.2.	Antarmuka Halaman Klasifikasi Gambar	39
4.2.	Pengujian Sistem	42
4.2.1.	Pengujian <i>Black Box</i>	42
4.2.2.	Pengujian Validitas.....	43
5.	BAB 5 KESIMPULAN DAN SARAN.....	48
5.1.	Kesimpulan.....	48
5.2.	Saran	48
6.	DAFTAR PUSTAKA	49

DAFTAR GAMBAR

Gambar 2.1 Aquascape yang berada di dalam sebuah aquarium.....	6
Gambar 2.2 <i>Alternanthera reineckii</i> 'Rosanervig'	7
Gambar 2.3 Tiga Contoh Citra Digital.....	9
Gambar 2.4 Citra Biner / Monokrom.....	9
Gambar 2.5 Model Ruang Warna Pengolahan Citra	10
Gambar 2.6 Model MobileNets	12
Gambar 2.7 <i>Machine Learning</i> merupakan bagian dari <i>Artificial Intellegent</i>	13
Gambar 2.8 Kategori <i>Machine Learning</i>	13
Gambar 2.9 Konsep Algoritma <i>Supervised Learning</i>	15
Gambar 2.10 Konsep <i>K-Nearest Neighbors</i>	16
Gambar 2.11 Ilustrasi Pengukuran Jarak <i>Euclidean</i>	17
Gambar 2.12 Ilustrasi <i>Confusion Matrix</i>	20
Gambar 3.1 Diagram Ishikawa	22
Gambar 3.2 <i>General Architecture</i>	24
Gambar 3.3 Unggah Gambar	24
Gambar 3.4 Proses Perhitungan <i>Convolutional Layer Filter</i>	26
Gambar 3.5 Posisi Filter saat Proses Konvolusi.	27
Gambar 3.6 <i>K-Nearest Neighbors</i>	29
Gambar 3.7 Ilustrasi <i>K-Nearest Neighbors</i> (www.geospasialis.com).....	29
Gambar 3.8 <i>Use Case Diagan</i>	32
Gambar 3.9 <i>Activity Diagram</i>	33
Gambar 3.10 <i>Sequence Diagram</i>	34
Gambar 3.11 <i>Flowchart</i> Klasifikasi dengan <i>K-Nearest Neighbors</i>	35
Gambar 3.12 Halaman Depan	36
Gambar 3.13 Halaman Klasifikasi Gambar	37
Gambar 4.1 <i>Training Data</i>	38
Gambar 4.2 Halaman Depan	39
Gambar 4.3 Halaman Klasifikasi Gambar	40

Gambar 4.4 <i>Drag and Drop</i> Gambar pada Halaman Klasifikasi Gambar	40
Gambar 4.5 Tampilan Setelah Dilakukan <i>Drag and Drop</i>	41
Gambar 4.6 Data Sedang Diproses	41
Gambar 4.7 Data Selesai Diproses.....	42

DAFTAR TABEL

Tabel 4.1 Pengujian <i>Black Box</i>	42
Tabel 4.2 Pengujian Validitas dengan nilai $k = 2$	43
Tabel 4.3 Pengujian Validitas dengan nilai $k = 4$	43
Tabel 4.4 Pengujian Validitas dengan nilai $k = 6$	44
Tabel 4.5 Pengujian Validitas dengan nilai $k = 8$	44
Tabel 4.6 Pengujian Validitas dengan nilai $k = 10$	45
Tabel 4.7 Pengujian Validitas dengan nilai $k = 12$	45
Tabel 4.8 Tabel <i>Confusion Matrix</i>	46
Tabel 4.9 Tabel Presisi	47
Tabel 4.10 Tabel <i>Recall</i>	47

BAB 1

PENDAHULUAN

1.1. *Latar Belakang*

Aquascape atau sering kali kita kenal dengan akuarium yang ditata dengan pasir, batu karang, kayu bahkan dengan tanaman air di dalamnya. Aquascape ini sangat berbeda dengan akuarium biasa. Aquascape bukan sekadar taman biasa, tetapi lebih kepada pemaknaan hidup dari miniatur alam yang penuh dengan kesinergian (Widjaja, 2013). Aquascape ini memerlukan perawatan khusus yang tentunya berbeda dengan akuarium biasa. Fenomena aquascape semakin berkembang dari waktu ke waktu. Proses penanaman aquascape ini bukanlah perkara yang mudah. Menanam tanaman aquascape membutuhkan ketelitian yang tinggi. Tanaman yang ingin dipelihara dalam aquascape sangat penting untuk selalu diperhatikan setiap proses perkembangannya. Mulai dari penataan tata letak tanaman, proses pertumbuhan (Siska et al., 2020), dan juga kebutuhan tanaman tersebut terhadap CO₂.

Tanaman memiliki kemampuan untuk berfotosintesis, proses fotosintesis tanaman memerlukan gas CO₂ sebagai bahan bakunya dan hasil fotosintesis tersebut berupa gas oksigen dan zat zat makanan yang diperlukan oleh tanaman dan makhluk hidup yang lain (Sukmawati et al., 2015). Seperti tanaman pada umumnya, berbagai macam jenis tanaman yang dapat hidup pada aquascape bergantung pada CO₂ memiliki kebutuhan CO₂ yang berbeda, ada jenis tanaman yang membutuhkan kadar CO₂ yang tinggi, tanaman yang kadar sedang, dan ada pula yang memiliki kadar kebutuhan CO₂ rendah. Namun tidak banyak masyarakat yang mengetahui hal tersebut. Pemberian CO₂ dalam pertumbuhan tanaman yang tepat mampu mengoptimalkan pertumbuhan tanaman dan efeknya tanaman tersebut akan tumbuh cepat dan tampak segar, untuk melakukan hal ini kita harus mengetahui terlebih dahulu kebutuhan CO₂ pada masing masing tanaman, dengan bantuan teknologi yang berkembang pesat saat ini kita dapat mencari informasi tersebut dengan lebih mudah. Maka karena itu, dibutuhkan sebuah sistem identifikasi kebutuhan CO₂ tanaman pada

aquascape, yang mana sistem ini menggunakan metode dari algoritma *K-Nearest Neighbors* yang dapat membantu identifikasi kebutuhan CO₂ tanaman pada aquascape.

KNN atau *K-Nearest Neighbors* merupakan salah satu metode yang digunakan untuk mengambil keputusan menggunakan pembelajaran terawasi. KNN pun merupakan algoritma *supervised learning* dimana hasil dari *query instance* yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada algoritma KNN. Menurut (Herfina, 2013), dalam penelitiannya mengenai “Pengenalan Pola Bentuk Bunga” menjelaskan bahwa algoritma *K-Nearest Neighbors* (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data *training* yang jaraknya paling dekat dengan objek tersebut. Algoritma ini memiliki beberapa keunggulan seperti cepat men-*training* data, sederhana dan mudah dipelajari lalu kuat dalam men-*training* data yang memiliki derau dan efektif jika men-*training* data yang besar. Menurut (Khamdani et al., 2021), yang melakukan penelitian dengan “Metode *K-Nearest Neighbors* Untuk Mendiagnosis Penyakit Tanaman Bawang Merah” menyatakan bahwa metode ini memiliki hasil akurasi rata-rata sebesar 85,385%. Hasil percobaan identifikasi yang mendapatkan akurasi sebesar 83,33% pada penelitian “Identifikasi Tumbuhan Obat Herbal Berdasarkan Citra Daun Menggunakan Algoritma *Gray Level Co-occurrence Matrix* dan *K-Nearest Neighbors*” oleh (Ni'mah et al., 2018) dan hasil akurasi tinggi dengan algoritma *K-Nearest Neighbors* yang mencapai 87,00% juga didapatkan oleh penelitian “Penerapan Algoritma *K-Nearest Neighbors* Pada Analisis Sentimen” (Ernawati & Wati, 2018) dan dapat dinyatakan bahwa algoritma ini memiliki hasil yang akurat.

Berdasarkan hasil di atas yang telah diuraikan, maka penulis melakukan penelitian dengan judul “Identifikasi CO₂ Tanaman Air Pada Aquascape Berdasarkan Citra Gambar Menggunakan Algoritma *K-Nearest Neighbors*”.

1.2. *Rumusan Masalah*

Tidak banyak masyarakat yang tahu tentang kebutuhan tanaman aquascape terhadap CO₂ sehingga seringkali terjadi kasus tanaman tidak tumbuh secara optimal dikarenakan kekurangan atau kelebihan CO₂ seperti yang dikutip oleh (Legia, 2021) yang mengatakan “Dalam perawatan ekosistem mini ini, salah satu tantangan yang sering dialami adalah melimpahnya kandungan CO₂ (karbon dioksida) dalam akuarium. CO₂ merupakan masalah utama dalam akuarium yang berasal dari sisa pakan dan kotoran ikan”. Untuk mengatasi hal tersebut dibutuhkan sistem program yang dapat mengidentifikasi kebutuhan CO₂ tanaman aquascape berdasarkan citra gambar agar dapat meminimalisir kekurangan ataupun berlebihan kandungan CO₂ pada aquascape.

1.3. *Batasan Masalah*

Dalam melakukan penelitian ini, penulis membatasi ruang masalah yang akan diteliti agar tidak terjadi kesalahan saat penelitian. Batasan masalah pada penelitian ini adalah:

1. Mengidentifikasi kebutuhan CO₂ tanaman berdasarkan citra gambar.
2. Algoritma yang digunakan dalam sistem identifikasi ini adalah *K-Nearest Neighbors*.
3. Proses klasifikasi diimplementasikan dengan bahasa pemrograman JavaScript dengan *framework* CodeIgniter 3, dan berbasis *web*.
4. Data yang diambil melalui laman www.tropica.com berupa gambar tanaman air dan keterangan kebutuhan CO₂ tanaman air tersebut.
5. Citra gambar yang diinput bersifat jernih dan minim noise.

1.4. *Tujuan Penelitian*

Adapun tujuan yang diharapkan dalam penelitian ini yaitu mengidentifikasi kebutuhan CO₂ tanaman air pada aquascape berdasarkan citra gambarnya menggunakan algoritma *K-Nearest Neighbors*.

1.5. *Manfaat Penelitian*

Manfaat yang diharapkan dari penelitian ini adalah:

1. Memudahkan para pemilik aquascape dalam merawat tanaman dengan mengetahui kadar kebutuhan CO₂.
2. Mencegah tumbuhan aquascape tidak tumbuh secara maksimal dengan memberi asupan CO₂ yang sesuai.

1.6. *Metode Penelitian*

Metode penelitian yang dilakukan dalam penelitian ini adalah:

1. Studi Pustaka

Pada tahap ini dilakukan pengumpulan referensi yang diperlukan dalam penelitian tersebut. Hal ini dilakukan untuk memperoleh informasi dan data yang diperlukan untuk penulisan penelitian tersebut. Referensi yang digunakan dapat berupa buku, buku digital, jurnal, artikel ilmiah yang berhubungan dengan algoritma *K-Nearest Neighbors*.

2. Analisis dan Perancangan

Pada tahap ini dilakukan pembuatan sistem berbasis *website* dengan bahasa pemrograman PHP menggunakan CodeIgniter 3 untuk penulis melakukan analisa terhadap apa saja yang dibutuhkan dalam penelitian tersebut dalam bentuk sebuah diagram alir (*flowchart*), arsitektur umum, *use case diagram*, *activity diagram*, *sequence diagram*, serta perancangan antarmuka.

3. Implementasi

Pada tahap ini dilakukan pembuatan sistem berbasis *website* dengan bahasa pemrograman PHP untuk mengkomputasi metode *K-Nearest Neighbors*.

4. Pengujian

Pada tahap ini dilakukan pengujian terhadap kinerja sistem yang telah dibangun sesuai kebutuhan yang sudah ditetapkan pada perancangan sistem tersebut.

5. Dokumentasi

Pada tahap ini dilakukan dokumentasi dan penulisan laporan dari hasil tahapan sebelumnya yang bertujuan untuk menunjukkan proses dan hasil penelitian dalam bentuk skripsi.

1.7. Sistematika Penulisan

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

BAB 1 : PENDAHULUAN

Bab ini menjelaskan tentang latar belakang dari penelitian yang dilakukan, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, penelitian yang relevan, metodologi penelitian dan sistematika penulisan skripsi.

BAB 2 : LANDASAN TEORI

Bab ini menjelaskan tentang teori-teori yang berhubungan dengan algoritma *K-Nearest Neighbors* serta menjelaskan metode yang digunakan dalam penelitian.

BAB 3 : ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan tentang analisis masalah yang terdapat pada sistem dan menganalisis hal-hal yang diperlukan dalam perancangan sistem yang akan dibuat.

BAB 4 : IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini menjelaskan tentang implementasi sistem yang dibuat berdasarkan pada perancangan yang telah dibuat. Kemudian akan dilanjutkan dengan pengujian untuk melihat apakah sistem yang telah dibuat sudah berjalan sesuai dengan perancangan, serta dapat menemukan kekurangan dan kesalahan yang terdapat pada sistem tersebut.

BAB 5 : KESIMPULAN DAN SARAN

Bab ini menjelaskan tentang kesimpulan yang diperoleh dari pembahasan sampai hasil penelitian selesai dilakukan dan saran sebagai masukan untuk penelitian terkait yang diharapkan akan bermanfaat bagi pihak yang berkaitan maupun penelitian berikutnya.

BAB 2

LANDASAN TEORI

2.1. *Aquascape*

Aquascape menurut (Hariyatno et al., 2018) merupakan seni dalam menata komponen batu, kerang, pasir, kayu, dan tanaman air dalam akuarium. Aquascape dapat diterapkan dalam pemeliharaan ikan hias air tawar atau air laut di dalam akuarium tersebut, dan tujuan utama dari aquascape adalah untuk menciptakan pemandangan dibawah permukaan air pada akuarium, sehingga akuarium akan terlihat lebih cantik.

Aquascape merupakan salah satu hobi, memelihara suatu ekosistem air tawar pada media aquarium, hobi ini disebut dengan istilah Aquascape sedangkan untuk ekosistem air laut disebut Oceanirium. Sejarah aquascape bermula pada tahun 1369, yang merupakan ide seorang kaisar China membuatkan wadah untuk memelihara ikan hias.

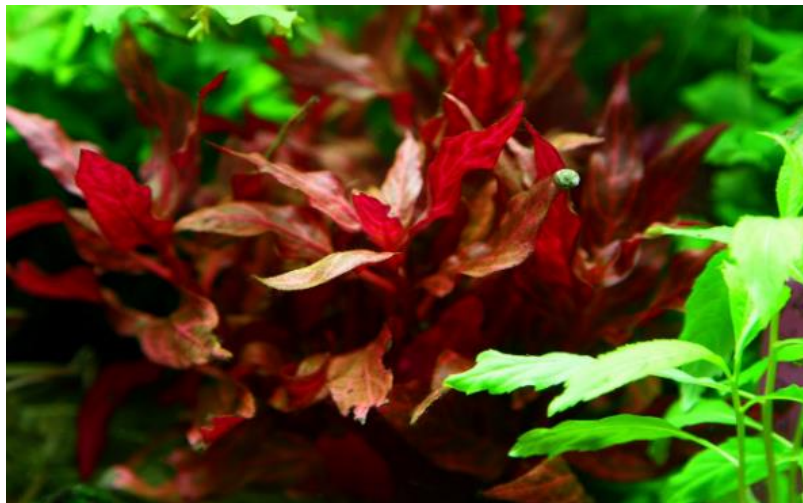


Gambar 2.1 Aquascape yang berada di dalam sebuah aquarium
(sumber: www.entrepreneur.bisnis.com)

Tujuan utama dari hobi aquascaping adalah untuk menciptakan pemandangan lanskap seperti bawah air yang cantik, untuk aspek pemeliharaan tanaman dan hewan air

juga wajib dipertimbangkan. Untuk memastikan keberhasilan proses aquascape, banyak sekali faktor pendukung yang harus seimbang.

Ada terdapat beberapa aspek teknis yang berkenaan dengan pemeliharaan tanaman aquascape yang harus dipertimbangkan, salah satunya faktor yang meliputi pengontrolan pemberian asupan CO₂ dalam aquascape.



Gambar 2.2 *Alternanthera reineckii* 'Rosanervig'
(sumber: www.tropica.com)

Dibalik banyaknya masyarakat sekarang yang memiliki aquascape, ada beberapa masyarakat yang terdapat sedikit kesulitan dalam mengatur asupan CO₂ kepada tanaman aquascape-nya.

2.2. *Fotosintesis pada Tanaman Air*

Seperti tanaman pada umumnya, tanaman air pada aquascape ini juga membutuhkan energi, energi tersebut didapat dari proses fotosintesis. Fotosintesis sendiri merupakan proses sintesis karbohidrat dari bahan-bahan anorganik yaitu CO₂ dan H₂O pada tumbuhan berpigmen dengan bantuan dari cahaya matahari.

Fotosintesis sendiri terdiri dari 2 fase yaitu fase pertama yang menghasilkan ATP dan NADPH₂ serta fase kedua yang menghasilkan karbohidrat (Nuryadi et al., 2021). Proses fotosintesis pada tanaman air selain membutuhkan pencahayaan, fotosintesis juga membutuhkan CO₂ yang juga sangat berpengaruh dalam peranan proses tersebut, maka

untuk mendapatkan hasil yang maksimal pada fotosintesis, tumbuhan aquascape perlu dilakukan perawatan secara intens dan rutin (Raharjo et al., 2018).

2.3. Pengolahan Citra

Pengolahan citra ini merupakan sebuah bentuk pemrosesan sebuah citra atau gambar dengan proses numerik dari gambar tersebut, dalam hal ini, objek yang diproses adalah masing masing pixel atau titik dari gambar tersebut (Gazali et al., 2012). Dan juga ilmu yang mempelajari hal-hal berkaitan dengan perbaikan kualitas terhadap suatu gambar (meningkatkan kontras, perubahan warna, restorasi citra), transformasi gambar (translasi, rotasi transformasi, skala, geometrik), melakukan pemilihan citra ciri (*feature images*) yang optimal untuk tujuan analisis, melakukan penyimpanan data yang sebelumnya dilakukan reduksi dan kompresi, transmisi data, dan waktu proses data (Munantri et al., 2019). Dalam perkembangannya, pengolahan citra pun sangat membantu untuk menyelesaikan masalah yang dihadapi manusia pada umumnya, dalam pengolahan citra, *image processing* dibutuhkan ketepatan dan keakuratan data karena pengolahan image ini berhubungan dengan pengolahan data (Putri, 2016).

Citra digital merupakan representatif dari citra yang diambil oleh mesin dengan bentuk pendekatan berdasarkan sampling dan kuantisasi. Sampling menyatakan besarnya kotak-kotak yang disusun dalam baris dan kolom. Dengan kata lain, sampling pada citra menyatakan besar kecilnya ukuran pixel (titik) pada citra, dan kuantisasi menyatakan besarnya nilai tingkat kecerahan yang dinyatakan dalam nilai tingkat keabuan (*grayscale*) sesuai dengan jumlah bit biner yang digunakan oleh mesin, dengan kata lain kuantisasi pada citra menyatakan jumlah warna yang ada pada citra.

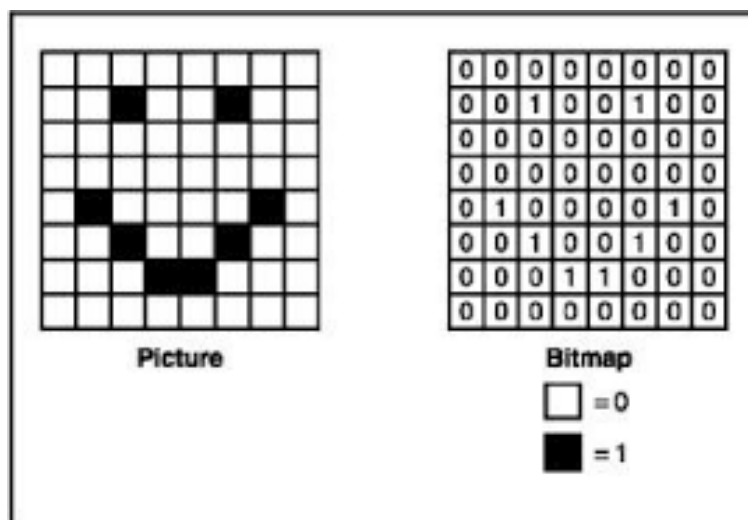
Adapun beberapa jenis citra digital yang sering digunakan adalah citra biner, citra grayscale dan citra warna:



Gambar 2.3 Tiga Contoh Citra Digital
(sumber: www.majalah1000guru.net)

2.3.1. Citra Biner (Monokrom)

Banyaknya dua warna, yaitu hitam dan putih. Dibutuhkan 1 bit di memori untuk menyimpan kedua warna ini.



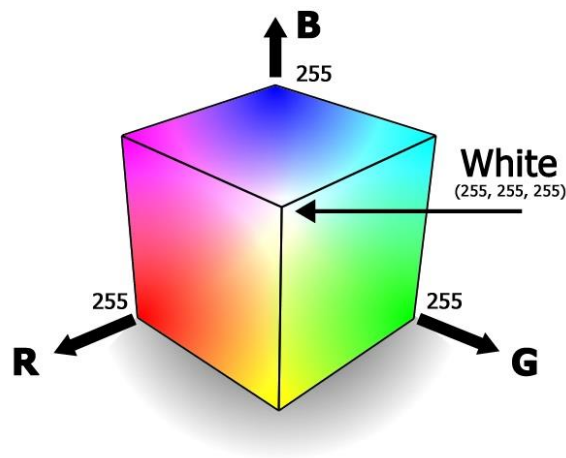
Gambar 2.4 Citra Biner / Monokrom
(sumber: www.mitchellkscscomputing.wordpress.com)

2.3.2. Citra Grayscale

Banyaknya warna tergantung pada jumlah bit yang disediakan di memori untuk menampung kebutuhan warna ini. Citra 2 bit mewakili 4 warna, citra 3 bit mewakili 8 warna, dan seterusnya. Semakin besar jumlah bit warna yang disediakan di memori, semakin halus gradasi warna yang terbentuk.

2.3.3. Citra Warna

Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar (RGB = *Red Green Blue*). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 byte, yang berarti setiap warna mempunyai gradasi sebanyak 255 warna. Berarti setiap piksel mempunyai kombinasi warna sebanyak $28 \times 28 \times 28 = 224 = 16$ juta warna lebih. Itulah sebabnya format ini dinamakan true color karena mempunyai jumlah warna yang cukup besar sehingga bisa dikatakan hampir mencakup semua warna di alam.



Gambar 2.5 Model Ruang Warna Pengolahan Citra
(sumber: www.support.medialooks.com)

Citra digital memiliki beberapa elemen:

1. *Brightness*

Brightness merupakan intensitas cahaya yang dipancarkan piksel dari citra yang dapat ditangkap oleh sistem penglihatan. Kecerahan pada sebuah titik (piksel) di dalam citra merupakan intensitas rata-rata dari suatu area yang melingkupinya.

2. *Contrast*

Kontras menyatakan sebaran terang dan gelap dalam sebuah citra. Pada citra yang baik, komposisi gelap dan terang tersebar secara merata.

3. Kontur

Adalah keadaan yang ditimbulkan oleh perubahan intensitas pada piksel-piksel yang bertetangga. Karena adanya perubahan intensitas inilah mata mampu mendeteksi tepi-tepi objek di dalam citra.

4. Warna

Sebagai persepsi yang ditangkap sistem visual terhadap panjang gelombang cahaya yang dipantulkan objek.

5. Bentuk

Bentuk adalah properti intrinsik dari objek 3 dimensi, dengan pengertian bahwa bentuk merupakan properti intrinsik utama untuk sistem penglihatan manusia.

6. Tekstur

Tekstur dicirikan sebagai distribusi spasial dari derajat keabuan di dalam sekumpulan pixel. Tekstur adalah karakteristik yang dimiliki oleh satu daerah yang cukup besar. Informasi tekstur dapat digunakan untuk membedakan sifat-sifat permukaan suatu benda dalam citra yang berhubungan dengan kasar dan halus, juga sifat-sifat spesifik dari kekasaran dan kehalusan permukaan tadi, yang sama sekali terlepas dari warna permukaan tersebut.

2.4. *MobileNets*

MobileNets merupakan serangkaian arsitektur yang dirancang khusus untuk tugas-tugas penglihatan komputer dan pemrosesan citra pada perangkat bergerak, seperti ponsel cerdas dan perangkat lainnya. MobileNets dikembangkan oleh Google dan menjadi salah satu arsitektur yang sangat populer dalam pemrosesan citra di perangkat dengan sumber daya terbatas.

Berikut adalah beberapa fitur utama dari MobileNets:

1. Ringan dan Efisien: Salah satu keunggulan utama MobileNets adalah efisiensinya dalam hal penggunaan sumber daya komputasi dan memori. Ini dirancang khusus untuk perangkat bergerak yang memiliki batasan daya pemrosesan dan memori.
2. *Arsitektur Convolutional Neural Network (CNN)*: MobileNets memanfaatkan arsitektur CNN untuk mengekstraksi fitur dari citra, yang merupakan pendekatan yang efektif dalam penglihatan komputer.
3. *Depthwise Separable Convolution*: Salah satu inovasi utama dalam MobileNets adalah penggunaan operasi konvolusi yang disebut *depthwise separable convolution*. Ini memungkinkan pengurangan besar dalam jumlah parameter yang harus dilatih, yang mengarah pada model yang lebih kecil dan lebih cepat.

4. *Configurable*: MobileNets memiliki beberapa versi yang dapat dikonfigurasi sesuai dengan tingkat kebutuhan komputasi dan kualitas hasil akhir. Anda dapat memilih versi yang sesuai dengan perangkat dan aplikasi Anda.
5. Pemrosesan Citra *Real-Time*: Meskipun efisien, MobileNets masih mampu melakukan pemrosesan citra secara *real-time* dalam berbagai tugas, seperti deteksi objek, pengenalan wajah, dan banyak lagi.
6. Dukungan untuk *Transfer Learning*: Anda dapat menggunakan MobileNets sebagai dasar untuk transfer learning. Artinya, Anda dapat melatih kembali bagian akhir model dengan dataset khusus Anda sendiri untuk tugas tertentu tanpa perlu melatih seluruh model dari awal.
7. Penggunaan Umum: MobileNets dapat digunakan dalam berbagai aplikasi, termasuk pengenalan gambar, deteksi objek, analisis video, pengenalan wajah, *augmented reality*, dan lainnya.



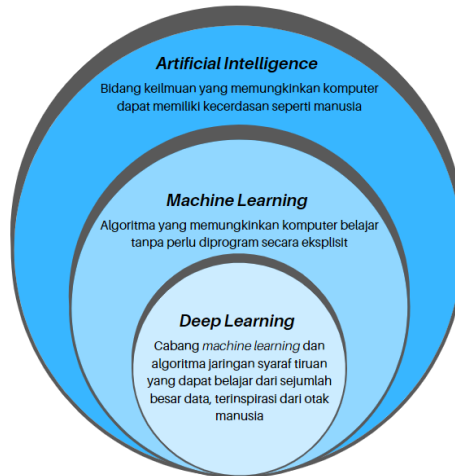
Gambar 2.6 Model MobileNets

MobileNets telah menjadi solusi yang sangat populer dalam pengembangan aplikasi berbasis citra di perangkat bergerak karena kinerjanya yang baik, arsitektur ini terus berkembang dan digunakan dalam berbagai aplikasi dan perangkat.

2.5. Machine Learning

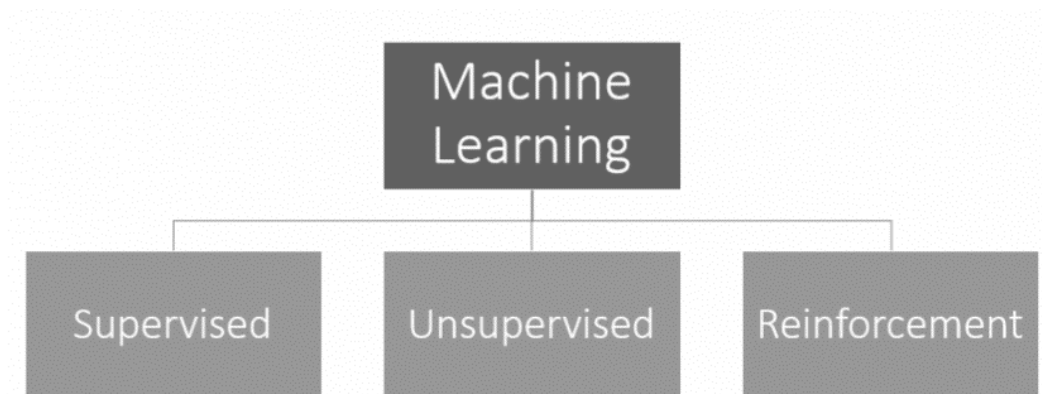
Machine learning adalah jenis dari kecerdasan buatan yang terdapat perancangan dan pengembangan algoritma yang dapat membuat komputer untuk mengembangkan perilaku berdasarkan data yang ada. Sistem pembelajaran *machine learning* ini dapat memanfaatkan contoh data data untuk menangkap ciri yang diperlukan dari probabilitas yang

mendasarinya. Algoritma machine learning mencari pola tertentu dari setiap kumpulan data yang menentukan kekhasan masing-masing untuk kemudian menyimpulkan sebuah aturan. Selanjutnya, aturan ini dapat digunakan untuk melakukan identifikasi dan prediksi bagi data baru yang relevan dengan model yang kita miliki.



Gambar 2.7 *Machine Learning* merupakan bagian dari *Artificial Intellegent* (sumber: www.datasains.co.id)

Penelitian terkini mengungkapkan bahwa machine learning ini terbagi atas 3 kategori yaitu *Supervised learning*, *Unsupervised learning* dan *Reinforcement learning* (Somvanshi et al., 2016).



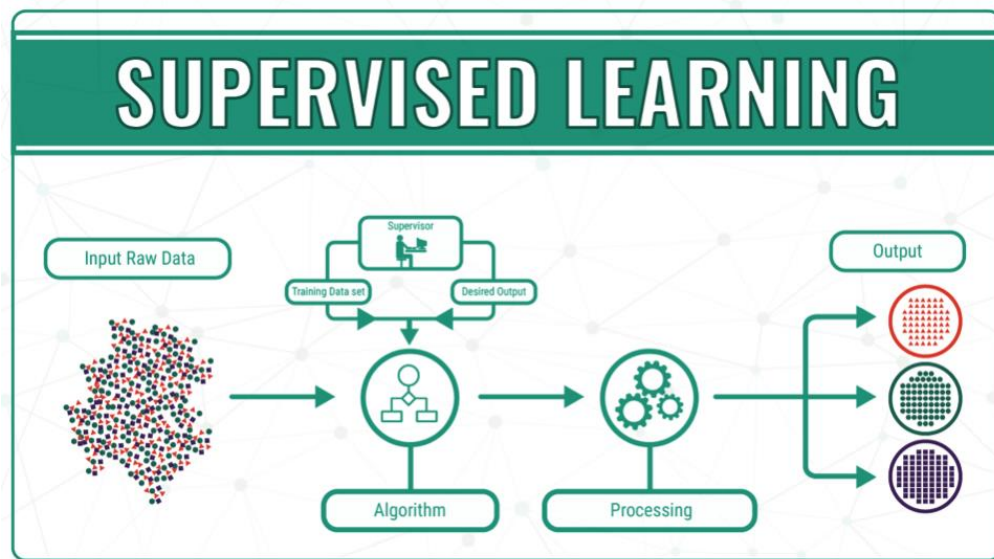
Gambar 2.8 Kategori *Machine Learning* (sumber: www.towardsdatascience.com)

Berdasarkan teknik pembelajarannya, dapat dibedakan *supervised learning* menggunakan dataset (*data training*) yang sudah berlabel, sedangkan *unsupervised learning* menarik kesimpulan berdasarkan dataset.

Supervised learning adalah suatu pendekatan dalam pembuatan AI. Disebut “*supervised*” karena dalam pendekatan ini, *machine learning* dilatih untuk mengenali pola antara input data dan label output. Tidak hanya itu, *machine learning* juga dilatih untuk mengidentifikasi hubungan yang mendasari koneksi input data dengan label output. Jika diibaratkan, *supervised learning* adalah pembelajaran di sekolah. Saat belajar di sekolah, Anda mempelajari mata pelajaran yang sudah ditentukan. *Supervised learning* pun demikian, *machine learning* dilatih untuk mempelajari data yang telah diberi label khusus.

Sedangkan *unsupervised learning* adalah suatu teknik yang digunakan machine learning dalam pembuatan *artificial intelligence*. Dalam pendekatan ini, Anda tidak perlu melatih algoritma komputer untuk mengenali pola penyusun AI. Model dirancang untuk bisa “belajar mandiri” dalam mengumpulkan informasi, termasuk mengenali data yang tidak berlabel. Disebut “*unsupervised*” karena model pada pendekatan ini tidak perlu dilatih. Perbedaan yang paling mendasar terletak pada bagaimana algoritma komputer bekerja dalam masing-masing pendekatan. Pada *supervised learning*, algoritma dilatih terlebih dulu baru bisa bekerja. Sedangkan algoritma komputer *unsupervised learning* telah dirancang untuk bisa langsung bekerja walaupun tanpa dilatih terlebih dulu.

Supervised Learning menggunakan *data training* untuk membuat *machine learning* model dan model ini akan digunakan untuk diuji pada data test. *Unsupervised Learning* tidak menggunakan data training dan hanya tergantung pada data test sehingga kita tidak bisa melakukan evaluasi terhadap model.



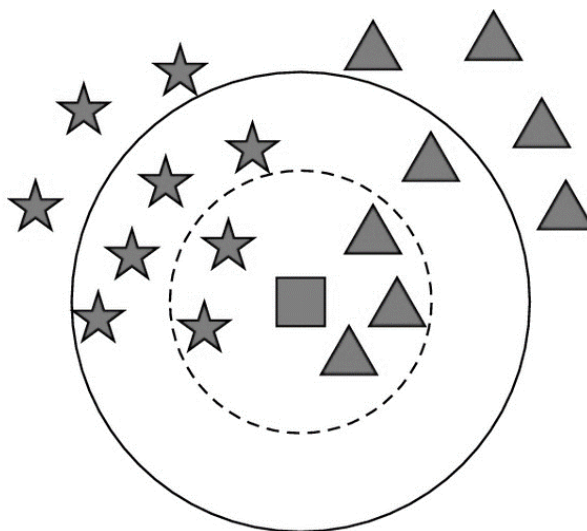
Gambar 2.9 Konsep Algoritma *Supervised Learning*
(sumber: www.cherrythecoder.wordpress.com)

Reinforcement learning merupakan kategori algoritma *machine learning* yang bisa membuat agen *software* dan mesin bekerja secara otomatis untuk menentukan perilaku yang ideal sehingga dapat memaksimalkan kinerja algoritmanya.

Dalam *reinforcement learning*, agen belajar secara otomatis menggunakan umpan balik tanpa data berlabel, tidak seperti metode pembelajaran *supervised learning*. Karena tidak ada data berlabel, maka agen terikat untuk belajar dari pengalamannya saja. *Reinforcement learning* memecahkan jenis masalah tertentu dimana pengambilan keputusan berurutan, dan tujuannya adalah jangka panjang, seperti bermain *game*, robotika, dan sebagainya.

2.6. *K-Nearest Neighbors (KNN)*

K-Nearest Neighbors adalah salah satu algoritma dalam pembelajaran *supervised learning* yang umumnya digunakan pada proses klasifikasi maupun regresi (Zhang, 2016).



Gambar 2.10 Konsep *K-Nearest Neighbors*
(sumber: www.researchgate.net)

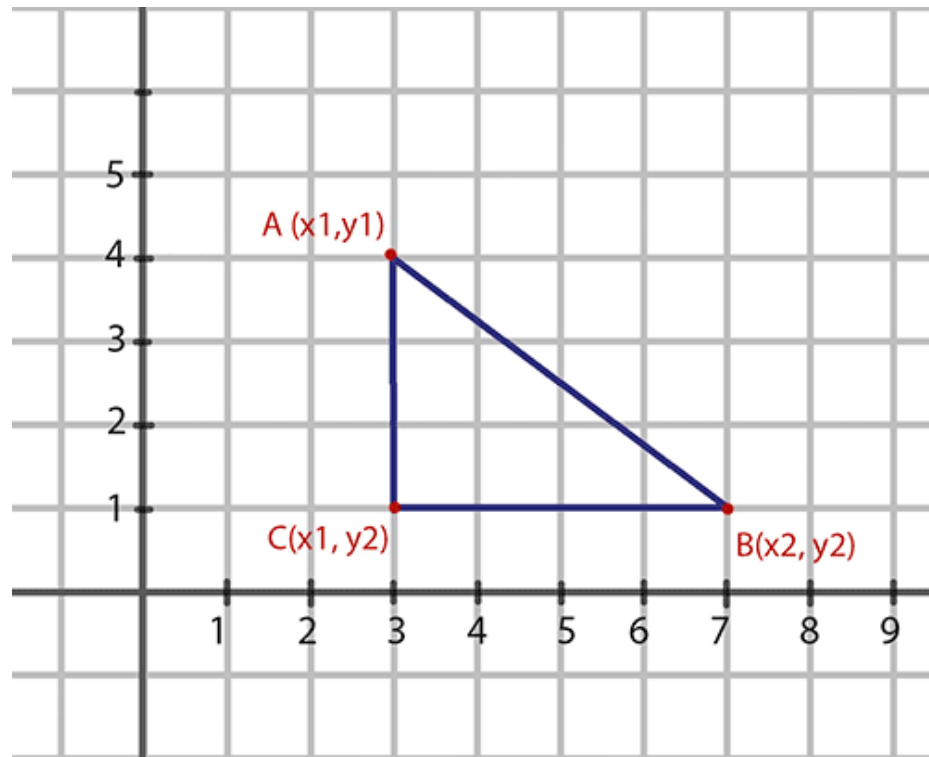
K-Nearest Neighbors bersifat bersifat *non-parametric* dan *lazy learning*. Metode yang bersifat *non-parametric* memiliki makna bahwa metode tersebut tidak membuat asumsi apapun tentang distribusi data yang mendasarinya. Dengan kata lain, tidak ada jumlah parameter atau estimasi parameter yang tetap dalam model, terlepas data tersebut berukuran kecil ataupun besar.

Algoritma *non-parametric* seperti KNN menggunakan sejumlah parameter yang fleksibel, dan jumlah parameter seringkali bertambah seiring data yang semakin banyak. Algoritma *non-parametric* secara komputasi lebih lambat, tetapi membuat lebih sedikit asumsi tentang data.

Pengklasifikasian KNN digunakan untuk mengklasifikasikan pengamatan yang tidak berlabel dengan menetapkan ke kelas dari contoh berlabel yang paling mirip. Klasifikasi data pada KNN ditentukan melalui perhitungan jarak antara test data yang diklasifikasi dengan *training* data. Perhitungan jarak ini dihitung menggunakan rumus *Euclidean Distance*.

Euclidean Distance merupakan perhitungan jarak dari 2 buah titik dalam *euclidean space*. *Euclidean space* diperkenalkan oleh Euclid, seorang matematikawan yang berasal dari Yunani sekitar tahun 300 B.C.E. yang digunakan untuk mempelajari hubungan antara sudut

dan *Euclidean Distance* ini berkaitan dengan Teorema Pythagoras dan biasanya diterapkan pada 1, 2 dan 3 dimensi. *Euclidean Distance* biasa diterapkan untuk membantu proses klasifikasi. Contoh algoritma yang menggunakan jarak ini adalah *K-Nearest Neighbors*. *Euclidean Distance* ini juga bisa diterapkan untuk algoritma lainnya.



Gambar 2.11 Ilustrasi Pengukuran Jarak *Euclidean*
(sumber: www.kitainformatika.com)

Langkah-langkah kerja dalam algoritma KNN adalah sebagai berikut (Fan et al., 2021):

1. Inisialisasi nilai K.

K merupakan jumlah parameter atau tetangga yang akan digunakan untuk mengklasifikasikan sebuah data. Misalkan, jika $K = 3$, maka akan ada 3 *training data* terdekat dari *test data* yang akan dibandingkan. Tidak ada rumus khusus untuk menentukan nilai optimal K. Namun semakin rendah nilai K, maka kemungkinan noise pada hasil akan semakin tinggi, sehingga dapat memunculkan efek *outlier*, dan semakin tinggi nilai K, maka proses komputasi akan semakin kompleks.

2. Perhitungan jarak antara *test data* dengan *training data*.

Untuk menghitung jarak antara *test data* dengan seluruh *training data*, digunakan rumus *Euclidean Distance*.

$$d(x, y) = \sqrt{\sum (x_i - y_i)^2} \quad (1)$$

3. *Finding the neighbors*

Setelah melakukan perhitungan jarak antara *test data* dengan semua *training data*, langkah selanjutnya adalah menemukan anggota-anggota *training data* yang paling dekat (bertetangga) dengan *test data* dengan jumlah anggota sebanyak nilai K.

4. *Classification*

Setelah mendapatkan tetangga-tetangga yang berdekatan dengan *test data*, langkah selanjutnya ialah melakukan *voting*. Proses *voting* merupakan penghitungan label / kelas data yang terbanyak dari data-data yang bertetangga dengan *test data*. Hasil dari *voting* ini akan klasifikasi kelas / label dari *test data*.

2.7. *Javascript*

Javascript merupakan bahasa *script*, yaitu bahasa yang tidak memerlukan kompiler untuk dapat menjalankannya, tetapi cukup dengan interpreter. Tidak perlu ada proses kompilasi terlebih dahulu agar program dapat dijalankan. *Browser web* Netscape Navigator dan Internet Explorer adalah salah satu contoh dari salah banyak interpreter, karena kedua *browser* ini telah dilengkapi dengan Interpreter Javascript. Tetapi tidak semua *browser web* dapat menjadi interpreter javascript karena belum tentu browser tersebut dilengkapi dengan interpreter javascript (Yani & Saputra, 2018).

JavaScript adalah bahasa pemrograman yang digunakan *developer* untuk membuat halaman web yang interaktif. Dari menyegarkan umpan media sosial hingga menampilkan animasi dan peta interaktif, fungsi JavaScript dapat meningkatkan pengalaman pengguna situs *web*. Sebagai bahasa skrip sisi klien, JavaScript adalah salah satu teknologi inti dari *World Wide Web*.

JavaScript sudah populer di kalangan developer untuk membuat situs dengan konten website yang bersifat dinamis, konten dinamis artinya bisa bergerak dan berubah dalam layar tanpa harus melakukan *reload* pada halaman *website*

Contohnya yaitu fitur gambar animasi, *slideshow* foto, pengisian *poling*, dan masih banyak lagi. Pada umumnya, JS berkolaborasi dengan HTML dan CSS. HTML berguna untuk membuat struktur website, sedangkan CSS untuk merancang *style* halaman agar lebih menarik, kemudian JS menambah elemen interaktif.

Awalnya JavaScript menjadi bahasa pemrograman yang hanya bekerja pada sisi klien atau pengolahan kode berjalan pada *browser*. Lalu, Google membuat JS semakin populer dengan AJAX untuk fitur *suggestion* kolom pencarian, sehingga terdapat fitur baru yang menampilkan website tanpa proses *reload*. Seiring berkembangnya zaman, JS bisa bekerja dari sisi *server* dengan memanfaatkan *platform framework* JavaScript seperti React.js, Node.js, dan lain-lain. Dengan adanya framework, para developer mulai tertarik menggunakan JS untuk mengembangkan produk-produk canggih mulai dari aplikasi *mobile* hingga *game online*.

2.8. Framework CodeIgniter 3

CodeIgniter adalah framework pemrograman PHP yg handal dan aman. CI3 adalah standar terbaru dari *framework* CodeIgniter, dikembangkan dari versi sebelumnya, dengan menggunakan PHP 7 sebagai basis pengembangan *framework*-nya. CodeIgniter memiliki pendekatan yang sangat baik, sehingga pemrogram bisa secara bertahap mengerti dan tahu bagaimana bekerja dengan menggunakan *framework*.

Codeigniter 3 ini merupakan salah satu Framework PHP kuat dan tahan lama yang sangat populer dengan menggunakan Konsep MVC (*Model View Controller*) dan sering digunakan oleh *developer* dan komunitas di seluruh penjuru dunia. Dengan menggunakan *framework* ini pula kita tidak akan terlalu kesulitan mencari tutorial yang membahas framework ini karena *developer* Codeigniter 3 sudah menyediakan dokumentasi secara lengkap.

2.9. Confusion Matrix

Confusion Matrix merupakan salah satu tools analitik prediktif yang menampilkan dan membandingkan nilai aktual atau nilai sebenarnya dengan nilai hasil prediksi model yang dapat digunakan untuk menghasilkan metrik evaluasi seperti *accuracy*, *precision*, dan *recall*.

Nilai *accuracy* didapatkan dari jumlah data bernilai positif yang diprediksi positif dan data bernilai negatif yang diprediksi negatif dibagi dengan jumlah seluruh data di dalam dataset. *Precision* adalah peluang kasus yang diprediksi positif yang pada kenyataannya termasuk kasus kategori positif. *Recall / Sensitivity* adalah peluang kasus dengan kategori positif yang dengan tepat diprediksi positif. Ilustrasi tabel *confusion matrix* dapat dilihat pada Gambar 2.12 berikut.

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

Gambar 2.12 Ilustrasi *Confusion Matrix*
(sumber: www.researchgate.net)

BAB 3

ANALISIS DAN PERANCANGAN

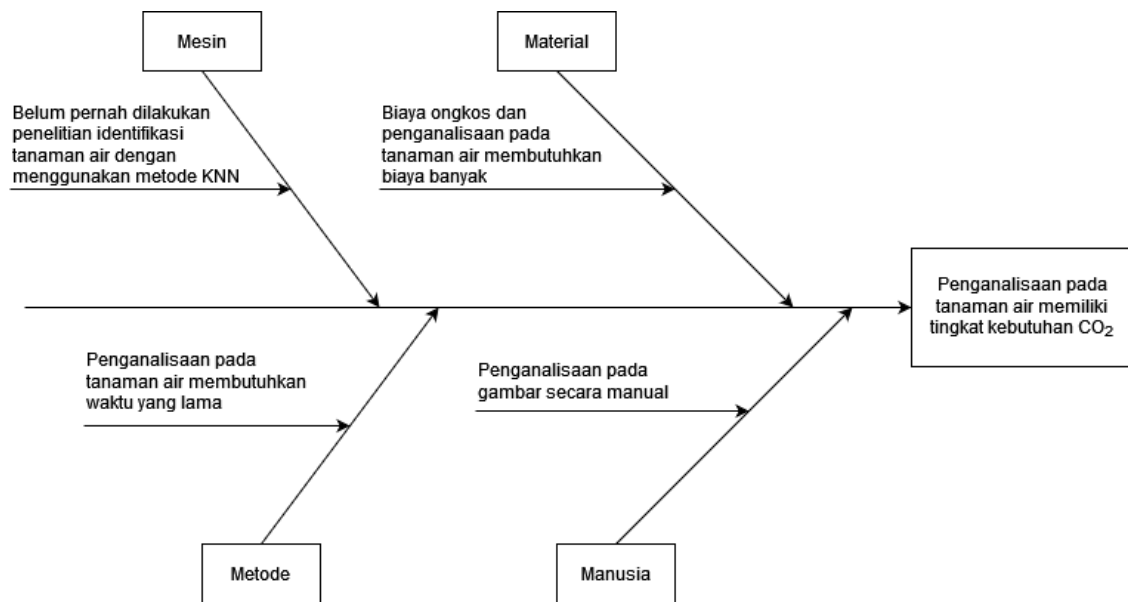
3.1. *Analisis Sistem*

Analisis sistem adalah suatu dasar dalam perancangan sistem dimana permasalahan dari suatu sistem akan dibuat menjadi bagian-bagian sederhana. Analisis sistem bertujuan untuk dapat lebih mudah mengetahui dan memperbaiki permasalahan yang terjadi sehingga kinerja sistem dapat menjadi lebih efisien. Ada beberapa tahapan analisis sistem yaitu analisis masalah dan analisis kebutuhan.

3.1.1. Analisis Masalah

Analisis masalah adalah analisis dapat mengidentifikasi suatu kasus berdasarkan penyebab yang ada dalam proses pembuatan sistem tersebut dan alternatif pemecahan menurut masalah tersebut. Analisis masalah dibutuhkan agar dapat meminimalisir kesalahan pada perancangan sistem yang dibangun sehingga sistem dapat berjalan dengan baik. Dalam analisis sistem tersebut akan dilakukan analisa terhadap proses pembuatan sistem agar bisa bekerja dengan baik dan juga dapat menghasilkan tingkat persentase kemungkinan yang mendekati keakuratan dari hasil metode tersebut dan melakukan analisa terhadap aplikasi berbasis *web* yang akan dirancang demi terwujudnya kemudahan bagi pengguna tersebut.

Dalam memudahkan pengidentifikasikan masalah akan digunakan media berupa diagram tulang ikan (diagram Ishikawa / *fishbone diagram*) yang akan menggambarkan dan mengidentifikasi hubungan antar sebab dan akibat (*cause and effect*) dari analisis permasalahan yang akan dibahas pada penelitian ini. Berikut diagram Ishikawa pada analisis permasalahan tersebut.



Gambar 3.1 Diagram Ishikawa

3.1.2. Analisis Kebutuhan (*Requirement Analysis*)

Untuk menyediakan aplikasi yang dapat memberikan informasi lengkap, maka dibutuhkan komponen penunjang pada aplikasi tersebut. Analisis kebutuhan akan menganalisa apa saja kebutuhan yang akan dijadikan sebagai syarat agar terwujudnya sistem yang berfungsi secara baik. Berikut analisis kebutuhan berdasarkan kategori sebagai berikut.

1. Kebutuhan Fungsional (*Functional Requirement*)

Kebutuhan fungsional adalah kebutuhan yang berhubungan dengan fungsi yang harus dimiliki pada setiap perangkat lunak. Berikut kebutuhan fungsional dari sistem tersebut sebagai berikut.

- a. Sistem menampilkan informasi seputar tanaman air pada aquascape.
- b. Sistem dapat melakukan perhitungan dengan mengimplementasikan metode *K-Nearest Neighbors* (KNN).
- c. Sistem mampu memberikan tingkat kebutuhan CO₂ berdasarkan gambar tanaman air telah diunggah sebelumnya kepada pengguna tersebut.

2. Kebutuhan Non-Fungsional (*Non-Functional Requirement*)

Kebutuhan non-fungsional adalah batasan layanan terkait dengan kemampuan yang diberikan melalui aplikasi. Berikut ini adalah kebutuhan non-fungsional pada aplikasi tersebut sebagai berikut.

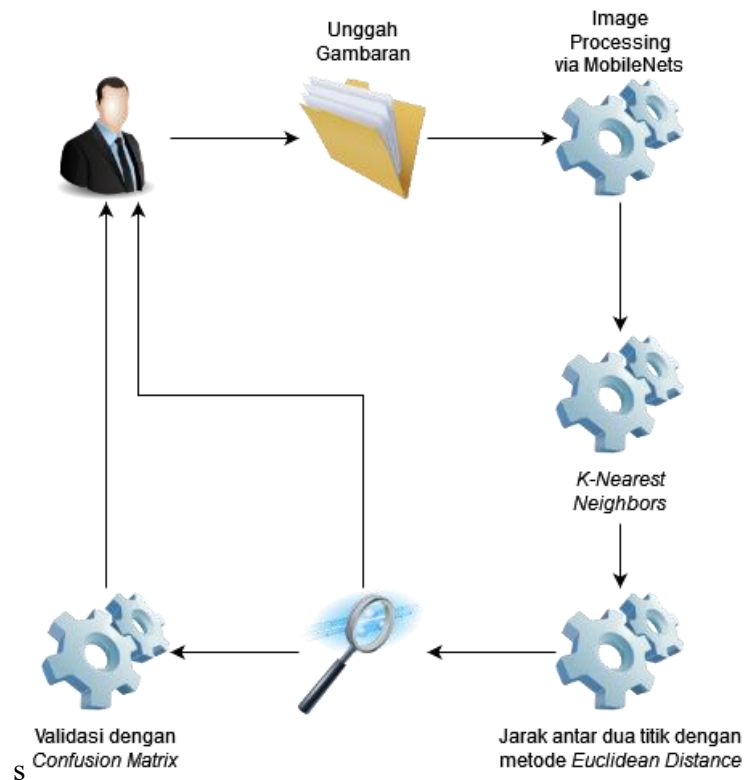
- a. Aplikasi memiliki antarmuka sederhana yang mudah dipahami oleh pengguna tersebut.
- b. Aplikasi tidak memerlukan biaya tambahan dan dapat bekerja dengan baik.
- c. Aplikasi mudah diakses di mana saja dan kapan saja.

3.2. *Pemodelan Sistem*

Pemodelan sistem digunakan untuk menggambarkan semua peran dan kondisi dari setiap pengguna terhadap sistem yang dibuat. Pemodelan sistem dibangun menggunakan *Unified Modeling Language* (UML) untuk mendesain dan melakukan perancangan sistem tersebut. Pemodelan pada penelitian ini menggunakan *general architecture*, *use case diagram*, *activity diagram*, dan *sequence diagram*.

3.2.1. General Architecture

General architecture adalah rancangan penting sebagai gambaran alur, proses dan interaksi antar komponen dalam suatu sistem.



Gambar 3.2 *General Architecture*

3.2.1.1. Unggah Gambar

Proses mengunggah gambar merupakan tahapan awal pada sistem ini, yaitu mengunggah salah satu citra gambar yang digunakan sebagai input pada sistem. Pada penelitian ini kumpulan citra gambar terbagi dua bagian, yaitu citra yang berfungsi untuk data *testing* dan citra yang digunakan untuk data *training* yang terdiri dari 45 gambar tersebut.

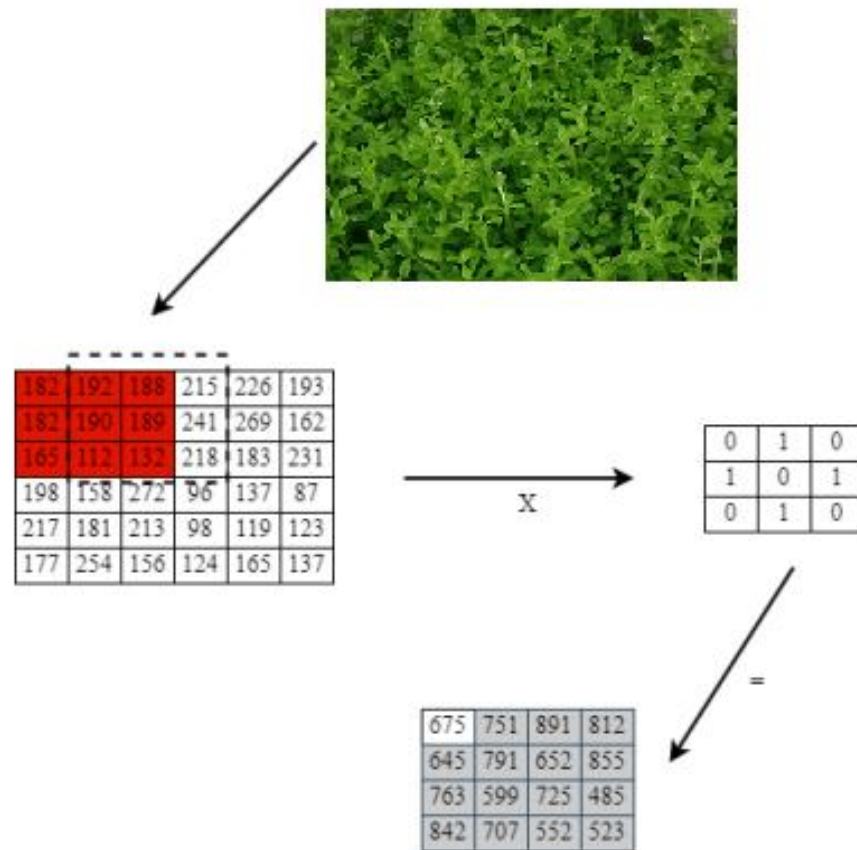


Gambar 3.3 Unggah Gambar

3.2.1.2. Ekstraksi Fitur

Dalam ekstraksi fitur menggunakan arsitektur MobileNets. Berikut adalah langkah-langkah yang umumnya terjadi dalam ekstraksi fitur menggunakan MobileNets:

1. Pengolahan Citra Input: Citra input dimasukkan ke dalam jaringan MobileNet.
2. Lapisan Konvolusi Awal: Lapisan konvolusi awal dari MobileNet bertujuan untuk mengekstrak fitur-fitur sederhana seperti tepi, sudut, dan tekstur dari citra. Lapisan ini adalah bagian dari proses konvolusi yang berulang-ulang dan menghasilkan fitur-fitur yang lebih halus.
3. Blok *Depthwise Separable Convolution*: Ini merupakan komponen kunci dalam MobileNets. Blok ini terdiri dari dua tahap: konvolusi *depthwise* dan konvolusi *pointwise*. Konvolusi *depthwise* memproses setiap saluran pada citra secara terpisah, sedangkan konvolusi *pointwise* menggabungkan hasil dari konvolusi *depthwise* untuk menghasilkan representasi fitur yang lebih kaya. Proses konvolusi ini dapat dijabarkan pada ilustrasi berikut:



Gambar 3.4 Proses Perhitungan *Convolutional Layer Filter*

Gambar 3.4 di atas menunjukkan ilustrasi perhitungan konvolusi dengan filter berukuran 3x3 dan menggunakan stride 1. *Stride* adalah berapa banyak penggeseran filter pada input matrix yang berjumlah satu.

Gambar 3.5 Posisi Filter saat Proses Konvolusi.

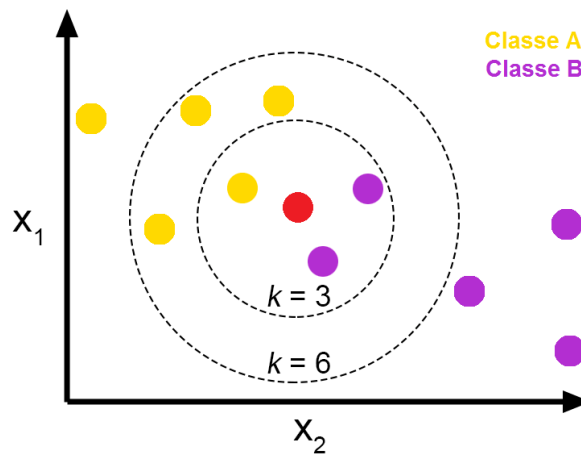
Gambar 3.5 menunjukkan proses perhitungan dot product pada layer konvolusional ketika proses konvolusi dengan filter 3x3 dimulai dari posisi paling kiri. Berikut adalah perhitungan dari *dot product*.

1. $(182 \times 0) + (192 \times 1) + (188 \times 0) + (182 \times 1) + (190 \times 0) + (189 \times 1) + (165 \times 0) + (112 \times 1) + (132 \times 0) = 675$
2. $(192 \times 0) + (188 \times 1) + (215 \times 0) + (190 \times 1) + (189 \times 0) + (241 \times 1) + (112 \times 0) + (132 \times 1) + (218 \times 0) = 751$
3. $(188 \times 0) + (215 \times 1) + (226 \times 0) + (189 \times 1) + (241 \times 0) + (269 \times 1) + (132 \times 0) + (218 \times 1) + (183 \times 0) = 891$
4. $(215 \times 0) + (226 \times 1) + (193 \times 0) + (241 \times 1) + (269 \times 0) + (162 \times 1) + (218 \times 0) + (183 \times 1) + (231 \times 0) = 812$
5. $(182 \times 0) + (190 \times 1) + (189 \times 0) + (165 \times 1) + (112 \times 0) + (132 \times 1) + (198 \times 0) + (158 \times 1) + (272 \times 0) = 645$
6. $(190 \times 0) + (189 \times 1) + (241 \times 0) + (112 \times 1) + (132 \times 0) + (218 \times 1) + (158 \times 0) + (272 \times 1) + (96 \times 0) = 791$
7. $(189 \times 0) + (241 \times 1) + (269 \times 0) + (132 \times 1) + (218 \times 0) + (183 \times 1) + (272 \times 0) + (96 \times 1) + (137 \times 0) = 652$

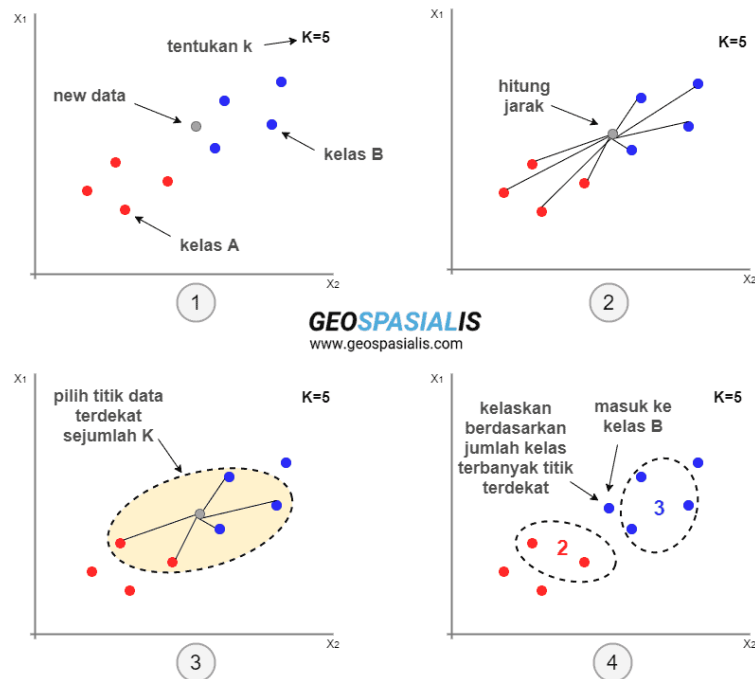
8. $(241 \times 0) + (269 \times 1) + (162 \times 0) + (218 \times 1) + (183 \times 0) + (231 \times 1) + (96 \times 0) + (137 \times 1) + (87 \times 0) = 855$
9. $(165 \times 0) + (112 \times 1) + (132 \times 0) + (198 \times 1) + (158 \times 0) + (272 \times 1) + (217 \times 0) + (181 \times 1) + (213 \times 0) = 763$
10. $(112 \times 0) + (132 \times 1) + (218 \times 0) + (158 \times 1) + (272 \times 0) + (96 \times 1) + (181 \times 0) + (213 \times 1) + (98 \times 0) = 599$
11. $(132 \times 0) + (218 \times 1) + (183 \times 0) + (272 \times 1) + (96 \times 0) + (137 \times 1) + (213 \times 0) + (98 \times 1) + (119 \times 0) = 725$
12. $(218 \times 0) + (183 \times 1) + (231 \times 0) + (96 \times 1) + (137 \times 0) + (87 \times 1) + (98 \times 0) + (119 \times 1) + (123 \times 0) = 485$
13. $(198 \times 0) + (158 \times 1) + (272 \times 0) + (217 \times 1) + (181 \times 0) + (213 \times 1) + (177 \times 0) + (254 \times 1) + (156 \times 0) = 842$
14. $(158 \times 0) + (272 \times 1) + (96 \times 0) + (181 \times 1) + (213 \times 0) + (98 \times 1) + (254 \times 0) + (156 \times 1) + (124 \times 0) = 707$
15. $(272 \times 0) + (96 \times 1) + (137 \times 0) + (213 \times 1) + (98 \times 0) + (119 \times 1) + (156 \times 0) + (124 \times 1) + (165 \times 0) = 552$
16. $(96 \times 0) + (137 \times 1) + (87 \times 0) + (98 \times 1) + (119 \times 0) + (123 \times 1) + (124 \times 0) + (165 \times 1) + (137 \times 0) = 523$
4. Lapisan *Pooling*: Beberapa versi MobileNets dapat mencakup lapisan *pooling* seperti *Global Average Pooling* (GAP) yang menggantikan lapisan *fully connected* tradisional. GAP menghasilkan representasi fitur akhir dengan mengambil rata-rata dari seluruh saluran di lapisan sebelumnya.
5. Keluaran Fitur: Hasil dari lapisan-lapisan sebelumnya adalah representasi fitur yang lebih abstrak dari citra input. Representasi ini dapat digunakan untuk berbagai tugas seperti klasifikasi objek, deteksi objek, atau tugas-tugas pemrosesan citra lainnya.

3.2.1.3. *K-Nearest Neighbors*

Sesudah melewati tahapan-tahapan sebelumnya, citra yang tadinya sudah diolah akan diproses dengan *training* data menggunakan metode *K-Nearest Neighbors*.



Gambar 3.6 *K-Nearest Neighbors*



Gambar 3.7 Ilustrasi *K-Nearest Neighbors* (www.geospasialis.com)

Setelah proses metode tersebut selesai, sistem akan menampilkan tingkat kebutuhan CO₂ pada tanaman air yang tadi diunggah oleh pengguna.

3.2.1.4. Euclidean Distance

Euclidean distance biasa digunakan untuk klasifikasi data pada *K-Nearest Neighbors* seperti yang dapat dilihat pada ilustrasi dibawah berikut:

Misal gambar yang akan di testing merupakan Gambar X dengan Fitur (2.8, 3.3, 1.6) dengan kelas yang belum diketahui dan $K = 10$ dan dengan data training:

Gambar 1: Fitur (2.5, 3.0, 1.8) - Tinggi

Gambar 2: Fitur (2.7, 2.8, 1.6) - Tinggi

Gambar 3: Fitur (1.5, 1.8, 2.2) - Sedang

Gambar 4: Fitur (1.9, 2.3, 2.0) - Sedang

Gambar 5: Fitur (3.0, 3.5, 1.5) - Sedang

Gambar 6: Fitur (2.2, 2.9, 1.7) - Sedang

Gambar 7: Fitur (1.8, 2.0, 2.1) - Rendah

Gambar 8: Fitur (2.0, 2.5, 2.3) - Rendah

Gambar 9: Fitur (3.2, 3.8, 1.4) - Rendah

Gambar 10: Fitur (2.4, 3.1, 1.9) - Rendah

Menghitung jarak Euclidean antara gambar X dan setiap gambar dari data testing:

- Jarak antara Citra X dan Citra 1: $\sqrt{((2.8-2.5)^2 + (3.3-3.0)^2 + (1.6-1.8)^2)} = \sqrt{(0.09 + 0.09 + 0.04)} = \sqrt{0.22}$
- Jarak antara Citra X dan Citra 2: $\sqrt{((2.8-2.7)^2 + (3.3-2.8)^2 + (1.6-1.6)^2)} = \sqrt{(0.01 + 0.25 + 0.00)} = \sqrt{0.26}$
- Jarak antara Citra X dan Citra 3: $\sqrt{((2.8-1.5)^2 + (3.3-1.8)^2 + (1.6-2.2)^2)} = \sqrt{(2.53 + 2.13 + 0.36)} = \sqrt{5.02}$
- Jarak antara Citra X dan Citra 4: $\sqrt{((2.8-1.9)^2 + (3.3-2.3)^2 + (1.6-2.0)^2)} = \sqrt{(0.81 + 1.00 + 0.16)} = \sqrt{1.97}$
- Jarak antara Citra X dan Citra 5: $\sqrt{((2.8-3.0)^2 + (3.3-3.5)^2 + (1.6-1.5)^2)} = \sqrt{(0.04 + 0.04 + 0.01)} = \sqrt{0.09}$
- Jarak antara Citra X dan Citra 6: $\sqrt{((2.8-2.2)^2 + (3.3-2.9)^2 + (1.6-1.7)^2)} = \sqrt{(0.36 + 0.16 + 0.01)} = \sqrt{0.53}$
- Jarak antara Citra X dan Citra 7: $\sqrt{((2.8-1.8)^2 + (3.3-2.0)^2 + (1.6-2.1)^2)} = \sqrt{(1.00 + 2.89 + 0.25)} = \sqrt{4.14}$
- Jarak antara Citra X dan Citra 8: $\sqrt{((2.8-2.0)^2 + (3.3-2.5)^2 + (1.6-2.3)^2)} = \sqrt{(0.64 + 0.64 + 0.49)} = \sqrt{1.77}$

- Jarak antara Citra X dan Citra 9: $\sqrt{((2.8-3.2)^2 + (3.3-3.8)^2 + (1.6-1.4)^2)} = \sqrt{(0.16 + 0.25 + 0.04)} = \sqrt{0.45}$
- Jarak antara Citra X dan Citra 10: $\sqrt{((2.8-2.4)^2 + (3.3-3.1)^2 + (1.6-1.9)^2)} = \sqrt{(0.16 + 0.04 + 0.09)} = \sqrt{0.29}$

Pilih 10 tetangga terdekat dengan jarak Euclidean terpendek.

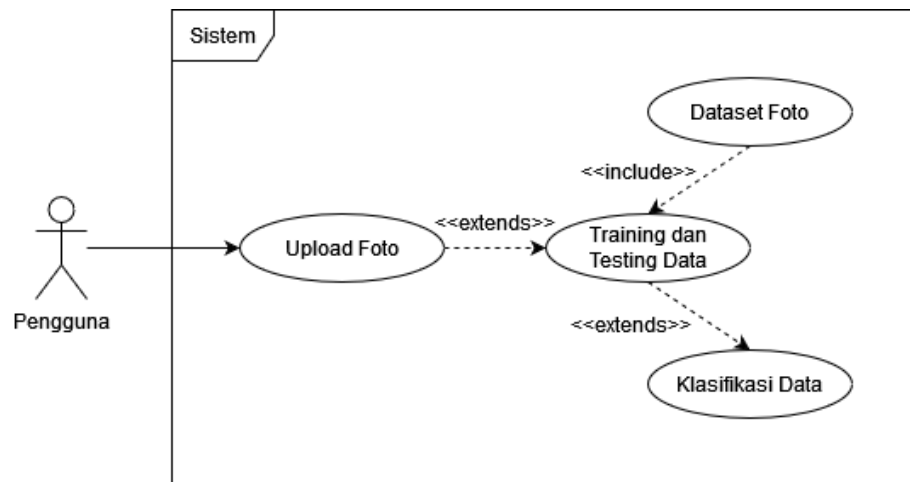
- Citra 5: Jarak = $\sqrt{0.09}$
- Citra 10: Jarak = $\sqrt{0.29}$
- Citra 1: Jarak = $\sqrt{0.22}$
- Citra 2: Jarak = $\sqrt{0.26}$
- Citra 9: Jarak = $\sqrt{0.45}$
- Citra 4: Jarak = $\sqrt{1.97}$
- Citra 8: Jarak = $\sqrt{1.77}$
- Citra 6: Jarak = $\sqrt{0.53}$
- Citra 7: Jarak = $\sqrt{4.14}$
- Citra 3: Jarak = $\sqrt{5.02}$

Ada 3 citra tinggi, 2 citra sedang, dan 5 citra rendah. Karena mayoritas kelas adalah rendah dengan 5 tetangga terdekat, maka gambar X akan diklasifikasikan sebagai kelas rendah.

Lalu digunakan confusion matrix untuk memvalidasi hasil, seperti: mengukur akurasi dari metode yang digunakan, evaluasi sensitivitas, spesifitas dan menentukan skor presisi.

3.2.2. Use Case Diagram

Use case diagram bertujuan untuk pengguna mengetahui layanan yang tersedia pada sistem dimana *use case diagram* dapat menjelaskan hubungan yang terjadi antar pengguna terhadap sistem dalam bentuk diagram secara ringkas.

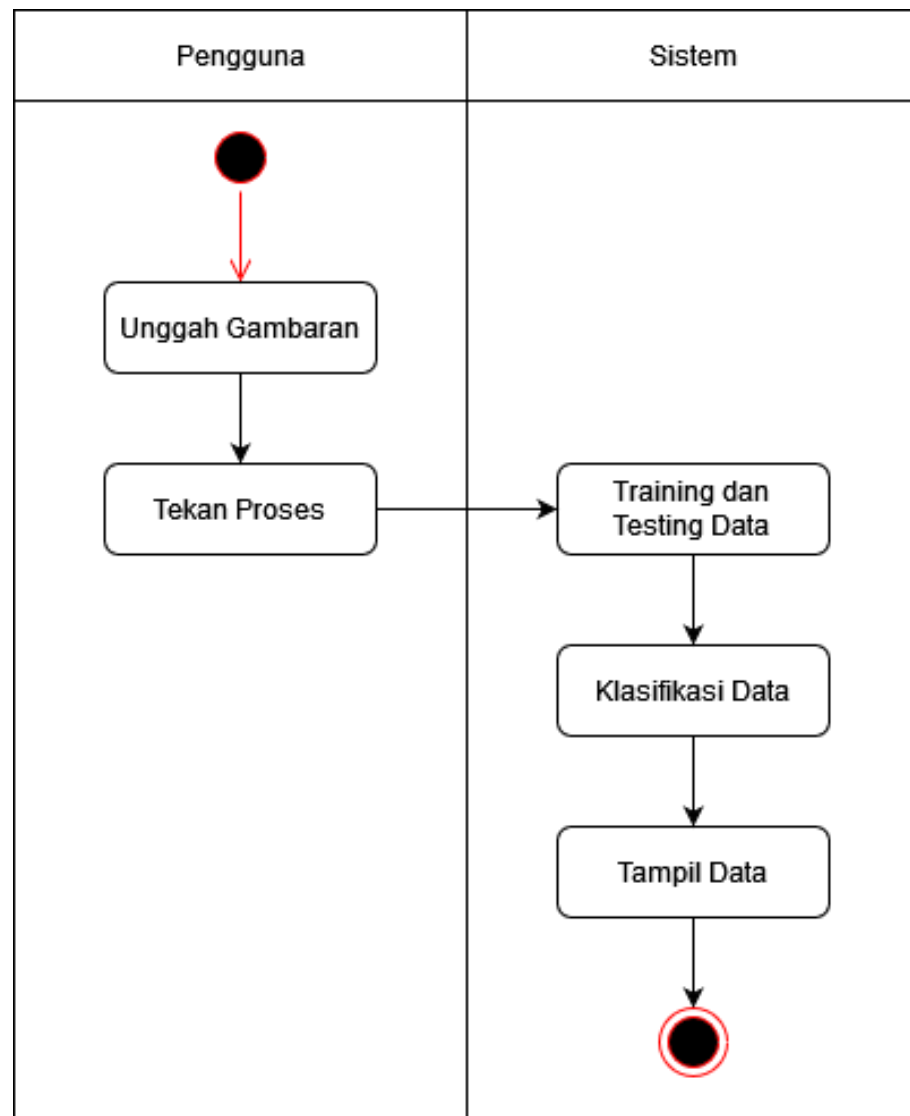


Gambar 3.8 *Use Case Diagram*

Berdasarkan pada Gambar 3.8. dimulai dengan pengguna mengunggah foto / gambaran. Kemudian proses *training data* dengan metode *K-Nearest Neighbors* dan menghasilkan tingkat kebutuhan CO₂ pada tanaman air tersebut.

3.2.3. Activity Diagram

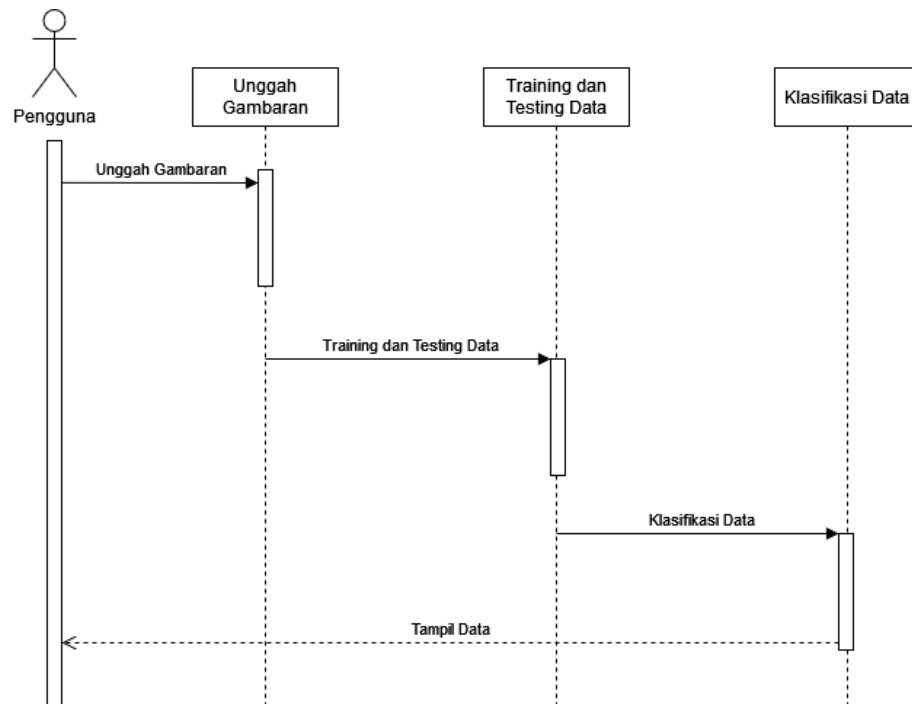
Activity diagram adalah diagram yang mempresentasikan proses dan urutan aktivitas sistem. Komponen pada *activity diagram* dihubungkan dengan tanda panah, tanda panah tersebut menunjukkan urutan aktivitas sistem dari awal sampai akhir. *Activity diagram* dapat ditinjau pada Gambar 3.9.



Gambar 3.9 *Activity Diagram*

3.2.4. Sequence Diagram

Sequence diagram adalah diagram interaksi yang menjelaskan secara detail sebuah operasi berlangsung antar objek. Selain itu, *sequence diagram* berfokus pada rangkaian waktu dan urutan interaksi yang dapat dilihat secara visual. *Sequence diagram* dapat ditinjau pada Gambar 3.10.



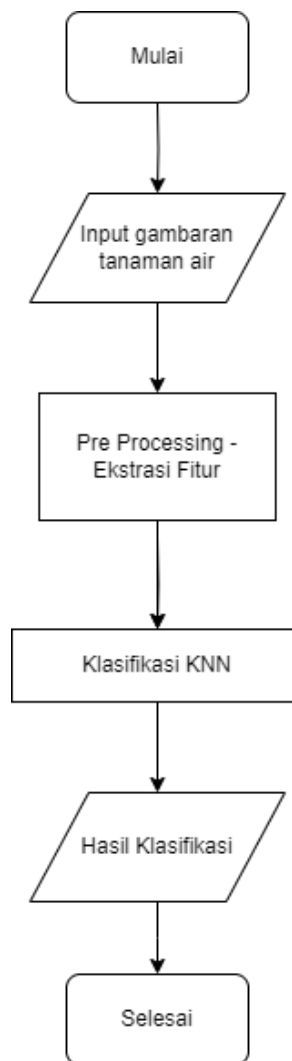
Gambar 3.10 *Sequence Diagram*

3.3. *Flowchart*

Flowchart atau diagram alir adalah sebuah representasi simbolik yang menggambarkan setiap langkah yang akan dilakukan dalam suatu proses.

3.3.1. **Flowchart Klasifikasi dengan K-Nearest Neighbors**

Flowchart pada Gambar 3.11 mendeskripsikan proses bagaimana *testing data diproses* pada KNN Model. Bagian awal dari proses ini adalah melakukan inputan dari citra / gambar *aquatic plant* yang akan diklasifikasikan. Setelah melalui proses pengolahan citra (*image processing*), maka selanjutnya proses klasifikasi akan berjalan dengan KNN Model. Hasil klasifikasi akan digolongkan ke dalam 3 jenis, yaitu tumbuhan dengan kebutuhan karbon dioksida rendah (*low*), sedang (*medium*), dan tinggi (*high*).



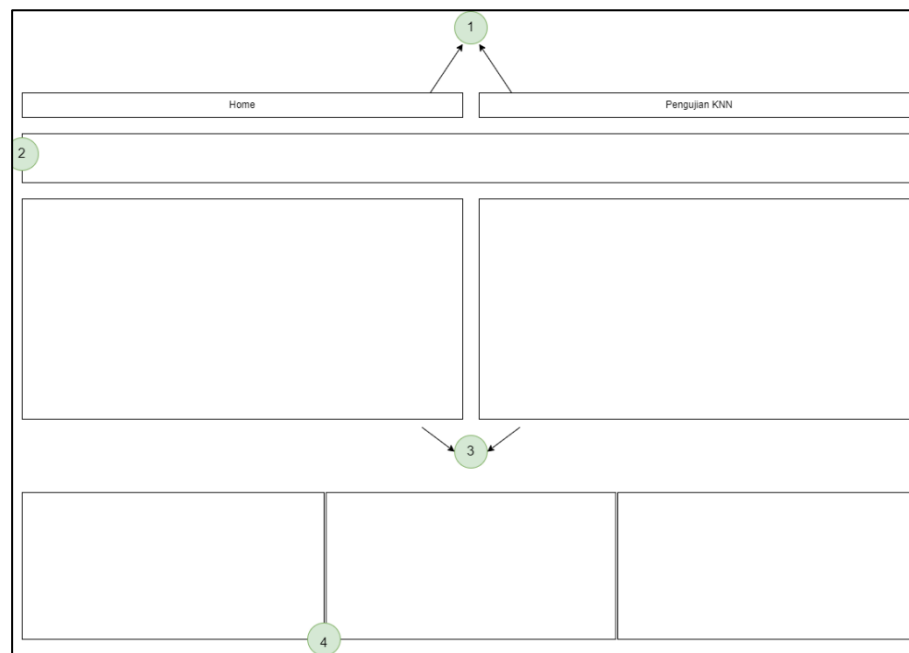
Gambar 3.11 Flowchart Klasifikasi dengan *K-Nearest Neighbors*

3.4. Perancangan Antarmuka (Interface)

Perancangan Antarmuka (*Interface*) merupakan bagian yang sangat penting dilakukan sebelum membuat suatu sistem. Perancangan Antarmuka (*Interface*) memberikan gambaran tentang tampilan sistem serta komponen-komponen sistem di dalamnya. Dengan adanya perancangan antarmuka (*interface*), maka akan mempermudah user dalam menggunakan sistem.

3.4.1. Rancangan Halaman Depan

Halaman depan atau yang biasa disebut dengan *homepage* merupakan halaman pertama yang akan dilihat oleh pengunjung web saat mengunjungi url suatu web, dengan kata lain *homepage* adalah halaman *default* yang digunakan untuk menarik perhatian pengunjung dan memberitahukan pengunjung tentang apa pada *web* tersebut. Tampilan *homepage* dapat dilihat pada Gambar 3.12.



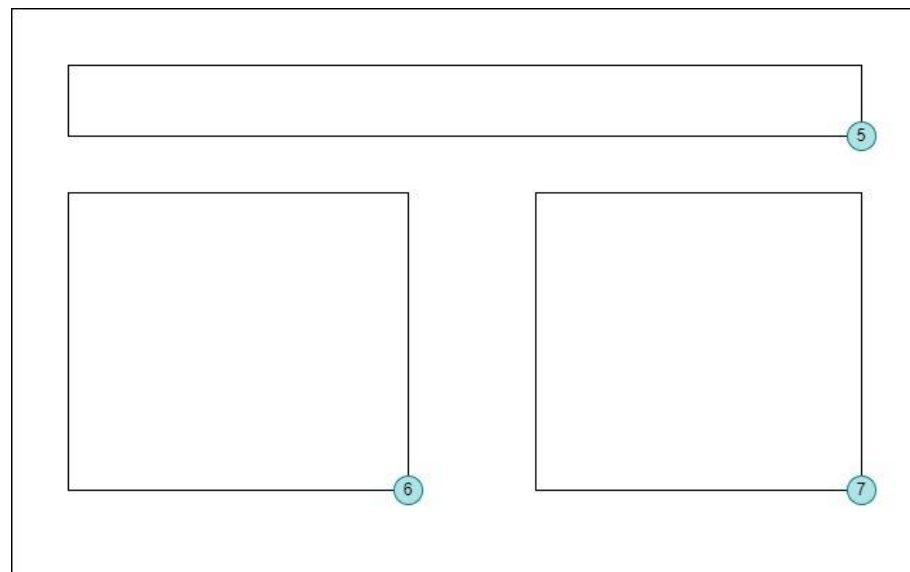
Gambar 3.12 Halaman Depan

Keterangan:

1. *Navigation Area* : memuat tautan navigasi untuk berpindah halaman
2. *Label* : berisikan judul dari web tersebut
3. *Input File* : berisikan informasi tentang aquascape
4. *Label* : berisi tentang informasi tanaman aquascape

3.4.2. Rancangan Halaman Klasifikasi Gambar

Halaman klasifikasi gambar merupakan tampilan bertujuan untuk men-*training* data kemudian melakukan proses klasifikasi data dengan model *K-Nearest Neighbors*. Setelah proses model tersebut, sistem akan menampilkan tingkat kebutuhan CO₂ dapat ditinjau pada Gambar 3.13.



Gambar 3.13 Halaman Klasifikasi Gambar

Keterangan:

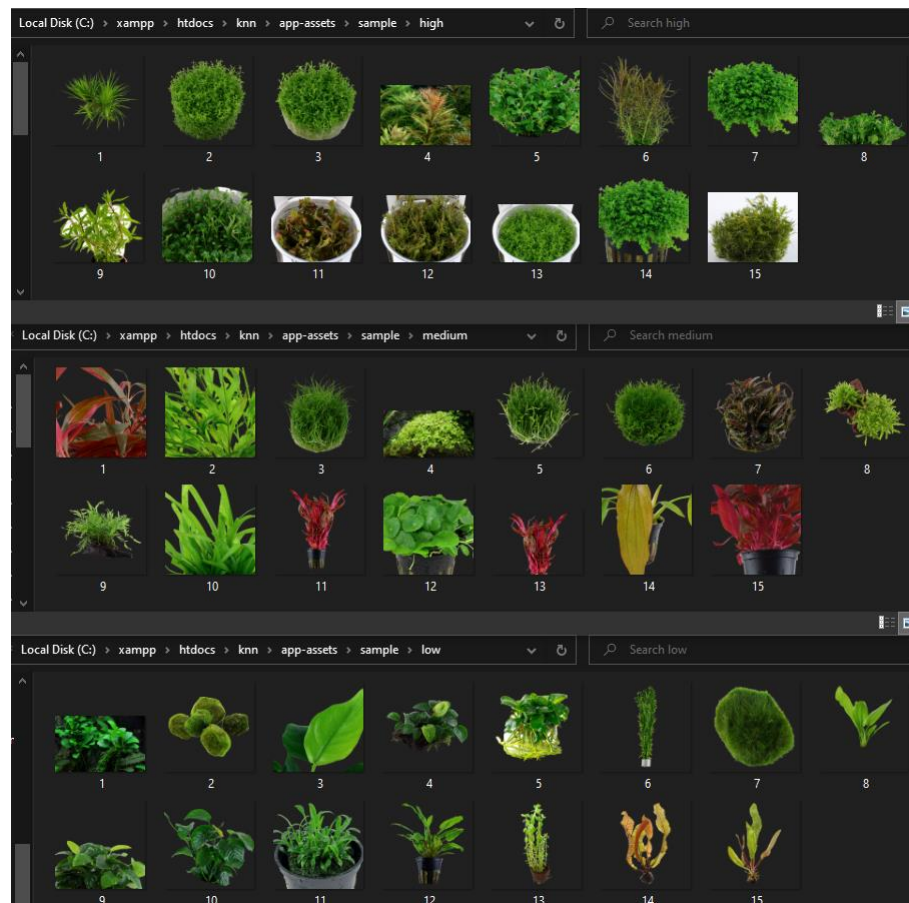
- 5. *Label* : berisikan judul aplikasi
- 6. *Input File* : untuk mengunggah gambar.
- 7. *Label* : berisikan hasil tampil dari model KNN

BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. *Implementasi Sistem*

Implementasi sistem merupakan sebuah tahapan dimana pengembang akan membangun sistem sesuai dengan rancangan yang telah dibuat sebelumnya. Pada tahapan ini, sistem dibuat kemudian akan diuji dengan validitasnya apakah sesuai dengan hasil yang diharapkan. Data-data yang akan diuji berjumlah 10 dan data yang akan di-*training* berjumlah 45 dapat ditinjau pada Gambar 4.1.

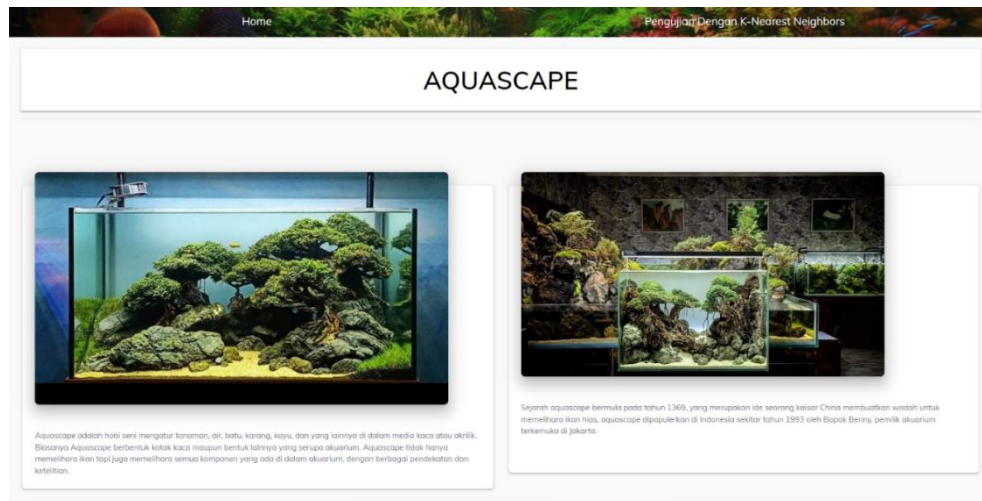


Gambar 4.1 *Training Data*

Berikut ini adalah implementasi sistem identifikasi kebutuhan CO₂ tanaman air pada aquaspace berdasarkan citra gambar.

4.1.1. Antarmuka Halaman Depan

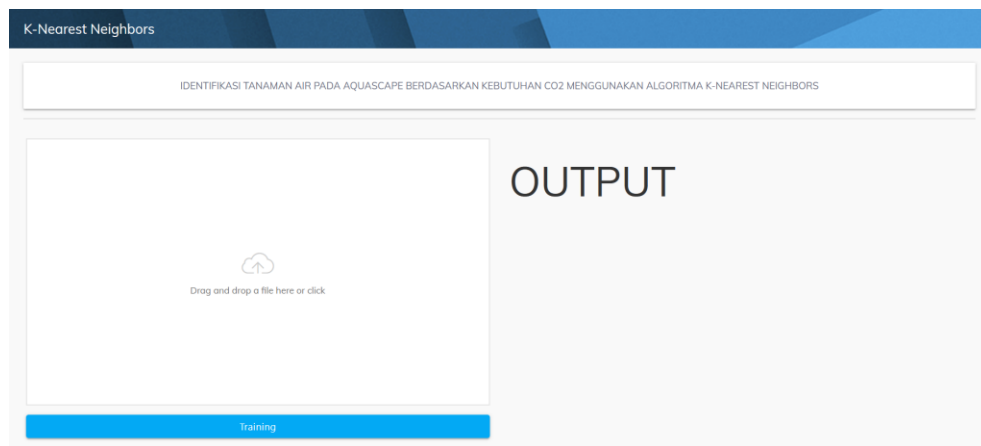
Halaman depan atau yang biasa disebut dengan *homepage* merupakan halaman pertama yang akan dilihat oleh pengunjung web, di halaman ini terdapat *navigation bar* di bagian atas, dan berisi tentang informasi seputar aquascape dapat ditinjau pada Gambar 4.2.



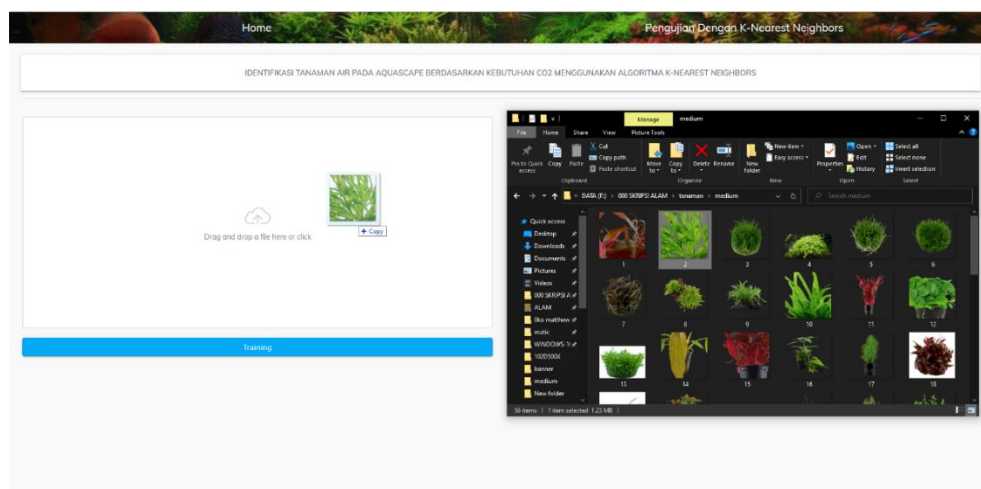
Gambar 4.2 Halaman Depan

4.1.2. Antarmuka Halaman Klasifikasi Gambar

Halaman klasifikasi gambar merupakan tampilan bertujuan untuk *men-training* data kemudian melakukan proses klasifikasi data dengan model *K-Nearest Neighbors*. Setelah proses model tersebut, sistem akan menampilkan tingkat kebutuhan CO₂ dapat ditinjau pada Gambar 4.3 dan Gambar 4.4.

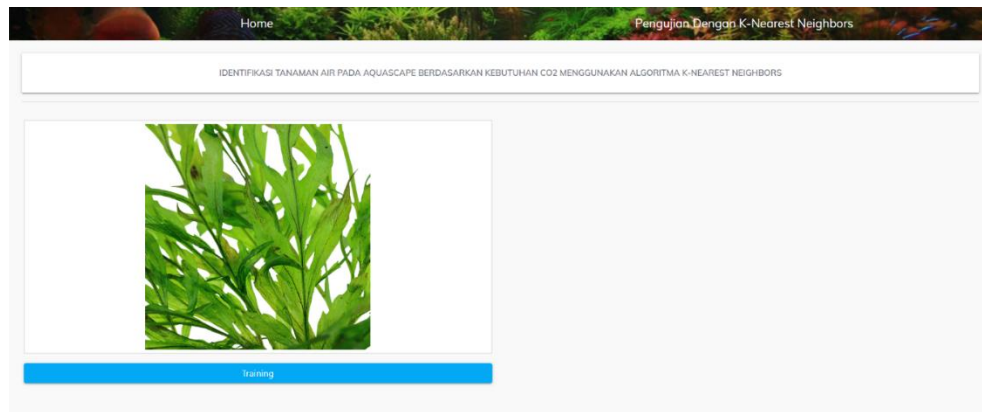


Gambar 4.3 Halaman Klasifikasi Gambar



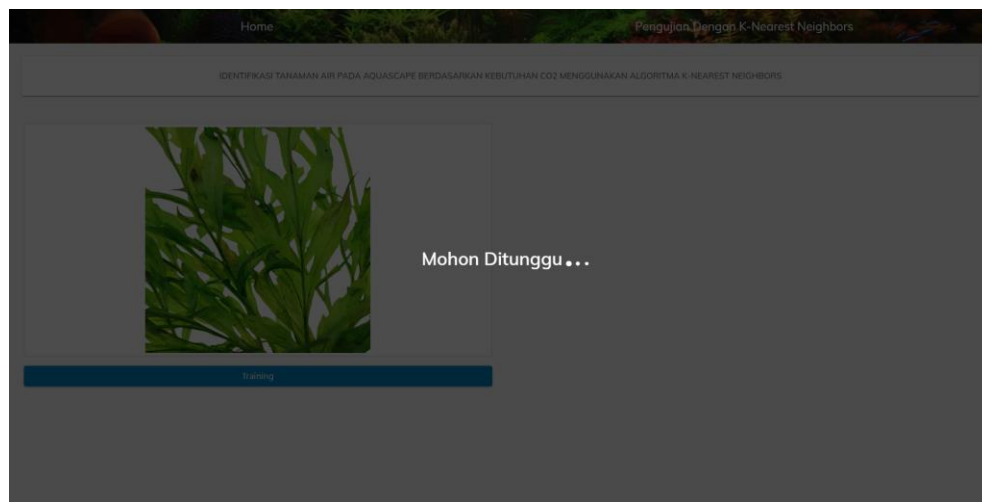
Gambar 4.4 *Drag and Drop* Gambar pada Halaman Klasifikasi Gambar

Untuk memasukkan data gambar yang akan diuji pada halaman web ni, kita dapat mengklik kotak di bagian kiri ataupun kita dapat melakukan *drag and drop* seperti Gambar 4.4.



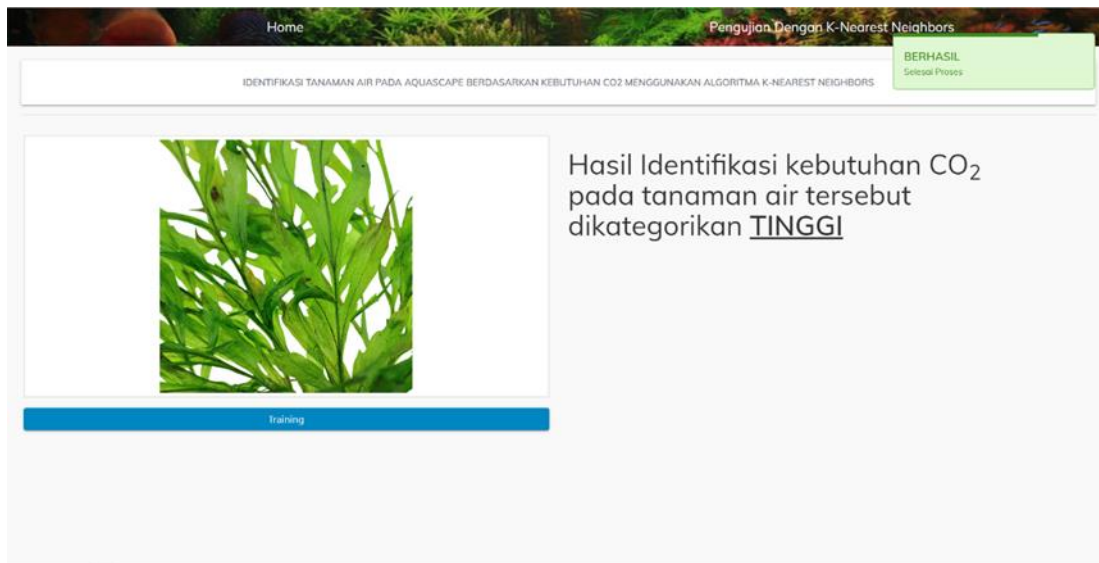
Gambar 4.5 Tampilan Setelah Dilakukan *Drag and Drop*

Setelah data yang berupa gambar diinput, maka gambar yang tadinya dimasukkan akan muncul di kotak sebelah kiri seperti pada Gambar 4.5 di atas.



Gambar 4.6 Data Sedang Diproses

Dan saat data selesai diproses, maka akan muncul notifikasi yang bertuliskan berhasil di bagian atas dan keterangan berupa kategori kebutuhan CO_2 dari tanaman aquascape seperti yang ada pada Gambar 4.7.



Gambar 4.7 Data Selesai Diproses

4.2. *Pengujian Sistem*

Proses pengujian yang akan dilakukan pada penelitian ini diantaranya adalah pengujian *black box*, pengujian validitas antar hasil yang diharapkan dengan sistem tersebut.

4.2.1. *Pengujian Black Box*

Pengujian black box adalah pengujian yang dilakukan untuk menemukan kesalahan atau cacat pada sistem tersebut. Pengujian ini digunakan untuk menilai fungsionalitas sistem apakah berjalan dengan baik atau tidak. Berikut tabel pengujian *black box* pada sistem identifikasi kebutuhan CO₂ tanaman air pada aquaspace berdasarkan citra gambar menggunakan *K-Nearest Neighbors* dapat ditinjau pada Tabel 4.1.

Tabel 4.1 Pengujian *Black Box*

No.	<i>Input</i>	<i>Proses</i>	<i>Output</i>	Hasil
1	Unggah <i>file</i> dengan cara “ <i>drag and drop</i> ”	Mengambil <i>file</i> dan muncul pada kotak tersebut.	Tampil foto pada kotak tersebut.	Berhasil

No.	Input	Proses	Output	Hasil
2	Unggah <i>file</i> dengan cara “ <i>click</i> ”	Mengambil <i>file</i> dan muncul pada kotak tersebut.	Tampil foto pada kotak tersebut.	Bperhasil
3	Tekan tombol <i>training</i>	Melakukan proses <i>training</i> data untuk menghasilkan hasil klasifikasi pada <i>output</i> tersebut.	Tampil <i>output</i> .	Berhasil

4.2.2. Pengujian Validitas

Pengujian validitas dengan membandingkan hasil aktual dengan hasil pada sistem tersebut. Adapun hasil evaluasi sistem dapat ditinjau pada Tabel 4.2.

Tabel 4.2 Pengujian Validitas dengan nilai $k = 2$

Percobaan	Hasil Aktual	Hasil Sistem	Hasil Pengujian
1	Tinggi	Tinggi	Sesuai
2	Tinggi	Tinggi	Sesuai
3	Sedang	Rendah	Tidak Sesuai
4	Sedang	Rendah	Tidak Sesuai
5	Sedang	Rendah	Tidak Sesuai
6	Sedang	Rendah	Tidak Sesuai
7	Rendah	Rendah	Sesuai
8	Rendah	Rendah	Sesuai
9	Rendah	Rendah	Sesuai
10	Rendah	Rendah	Sesuai

Tabel 4.3 Pengujian Validitas dengan nilai $k = 4$

Percobaan	Hasil Aktual	Hasil Sistem	Hasil Pengujian
-----------	--------------	--------------	-----------------

1	Tinggi	Tinggi	Sesuai
2	Tinggi	Tinggi	Sesuai
3	Sedang	Rendah	Tidak Sesuai
4	Sedang	Rendah	Tidak Sesuai
5	Sedang	Rendah	Tidak Sesuai
6	Sedang	Sedang	Sesuai
7	Rendah	Rendah	Sesuai
8	Rendah	Rendah	Sesuai
9	Rendah	Rendah	Sesuai
10	Rendah	Rendah	Sesuai

Tabel 4.4 Pengujian Validitas dengan nilai $k = 6$

Percobaan	Hasil Aktual	Hasil Sistem	Hasil Pengujian
1	Tinggi	Tinggi	Sesuai
2	Tinggi	Tinggi	Sesuai
3	Sedang	Rendah	Tidak Sesuai
4	Sedang	Rendah	Tidak Sesuai
5	Sedang	Rendah	Tidak Sesuai
6	Sedang	Sedang	Sesuai
7	Rendah	Rendah	Sesuai
8	Rendah	Rendah	Sesuai
9	Rendah	Rendah	Sesuai
10	Rendah	Rendah	Sesuai

Tabel 4.5 Pengujian Validitas dengan nilai $k = 8$

Percobaan	Hasil Aktual	Hasil Sistem	Hasil Pengujian
1	Tinggi	Tinggi	Sesuai
2	Tinggi	Tinggi	Sesuai

3	Sedang	Rendah	Tidak Sesuai
4	Sedang	Sedang	Sesuai
5	Sedang	Rendah	Tidak Sesuai
6	Sedang	Sedang	Sesuai
7	Rendah	Rendah	Sesuai
8	Rendah	Rendah	Sesuai
9	Rendah	Rendah	Sesuai
10	Rendah	Rendah	Sesuai

Tabel 4.6 Pengujian Validitas dengan nilai $k = 10$

Percobaan	Hasil Aktual	Hasil Sistem	Hasil Pengujian
1	Tinggi	Tinggi	Sesuai
2	Tinggi	Tinggi	Sesuai
3	Sedang	Sedang	Sesuai
4	Sedang	Sedang	Sesuai
5	Sedang	Rendah	Tidak Sesuai
6	Sedang	Sedang	Sesuai
7	Rendah	Rendah	Sesuai
8	Rendah	Rendah	Sesuai
9	Rendah	Rendah	Sesuai
10	Rendah	Rendah	Sesuai

Tabel 4.7 Pengujian Validitas dengan nilai $k = 12$

Percobaan	Hasil Aktual	Hasil Sistem	Hasil Pengujian
1	Tinggi	Tinggi	Sesuai
2	Tinggi	Rendah	Tidak Sesuai
3	Sedang	Rendah	Tidak Sesuai
4	Sedang	Tinggi	Tidak Sesuai

5	Sedang	Rendah	Tidak Sesuai
6	Sedang	Sedang	Sesuai
7	Rendah	Rendah	Sesuai
8	Rendah	Rendah	Sesuai
9	Rendah	Rendah	Sesuai
10	Rendah	Rendah	Sesuai

Setelah melakukan pengujian data berdasarkan pada ke-enam tabel tersebut dengan nilai k berbeda masing-masing dan nilai $k = 10$ sebagai nilai terbaik, maka selanjutnya melakukan pengujian evaluasi performa menggunakan *confusion matrix* untuk mengukur performa berupa sensitif, spesifik, presisi, dan akurasi. Hasil klasifikasi menggunakan metode *K-Nearest Neighbors* tersebut menghasilkan perhitungan yang telah diklasifikasikan dapat dilihat pada Tabel 4.8.

Tabel 4.8 Tabel *Confusion Matrix*

Kelas Sebenarnya	Kelas Prediksi			
		Rendah	Sedang	Tinggi
	Rendah	4	0	0
	Sedang	1	3	0
	Tinggi	0	0	2

$$TP = 4 + 3 + 2 = 9$$

$$Accuracy = \frac{9}{10} \times 100\% = 90\%$$

Tabel 4.9 Tabel Presisi

	Rendah	Sedang	Tinggi
TP	4	3	2
FP	0	1	0
Precision	$4 / (4 + 0) = 1$	$3 / (3 + 1) = 0.75$	$2 / (2 + 0) = 1$

$$Precision = \frac{1 + 0.75 + 1}{3} \times 100\% = 91.167\%$$

Tabel 4.10 Tabel Recall

	Rendah	Sedang	Tinggi
TP	4	3	2
FP	1	0	0
Precision	$4 / (4 + 1) = 0.8$	$3 / (3 + 0) = 1$	$2 / (2 + 0) = 1$

$$Recall = \frac{0.8 + 1 + 1}{3} \times 100\% = 93.333\%$$

Jadi, pada pengujian *confusion matrix* yang diperoleh menggunakan 10 data menghasilkan akurasi sebesar 90%, presisi sebesar 91.167%, serta *recall* sebesar 93.333%.

BAB 5

KESIMPULAN DAN SARAN

5.1. *Kesimpulan*

Berdasarkan hasil pengujian pada sistem identifikasi kebutuhan CO₂ tanaman air berdasarkan citra gambar menggunakan metode *K-Nearest Neighbors* yang telah dilakukan, diperoleh beberapa kesimpulan sebagai berikut.

1. Sistem identifikasi CO₂ pada tanaman air telah berhasil dibangun dengan menggunakan metode *K-Nearest Neighbors*.
2. Data yang digunakan sebagai data latih berjumlah 45, dan data diambil dari website www.tropica.com
3. Hasil perbandingan uji sistem dengan 10 data sampel dapat menunjukkan bahwa akurasi sebesar 90%, presisi sebesar 91.167%, serta sensitif sebesar 93.333%.

5.2. *Saran*

Berdasarkan hasil penelitian, adapun beberapa saran yang diberikan penulis terkait pengembangan dan perbaikan untuk ke depannya adalah sebagai berikut.

1. Metode yang digunakan untuk menghitung jarak antara *test data* dan *training data* pada sistem tersebut sebaiknya dikembangkan dengan penambahan metode lainnya agar menambah keakuratan dari hasil yang didapatkan.
2. Dataset terkait tanaman air dapat ditambahkan agar hasil klasifikasi lebih akurat.
3. Sistem diimplementasikan tidak hanya berbasis *web*, namun juga diimplementasikan dalam bentuk *desktop app* agar dapat mempermudah jangkauan oleh pengguna tersebut.

DAFTAR PUSTAKA

- Ernawati, S., & Wati, R. (2018). Penerapan Algoritma K-Nearest Neighbors Pada Analisis Sentimen Review Agen Travel. *Jurnal Khatulistiwa Informatika*, VI(1), 64–69. <https://www.trustpilot.com/categories/tr>
- Gazali, W., Soeparno, H., & Ohliati, J. (2012). Penerapan Metode Konvolusi dalam Pengolahan Citra Digital. *Jurnal Mat Stat*, 12(2), 103–113.
- Hariyatno, H., Isanawikrama, I., Wimpertiwi, D., & Jhony Kurniawan, Y. (2018). Membaca Peluang Merakit “Uang” dari Hobi Aquascape. *Jurnal Pengabdian Dan Kewirausahaan*, 2(2), 117–125. <http://journal.ubm.ac.id/>
- Herfina, H. (2013). Pengenalan Pola Bentuk Bunga Menggunakan Principle Component Analysis dan KNN. *Seminar Nasional Teknologi Informasi Dan Multimedia*, 7, 25–30.
- Khamdani, M. K., Hidayat, N., & Dewi, R. K. (2021). Implementasi Metode K-Nearest Neighbor Untuk Mendiagnosis Penyakit Tanaman Bawang Merah. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 5(1), 11–16. <http://j-ptiik.ub.ac.id>
- Legia, A. (2021). Sejuta Manfaat Tanaman Air untuk Aquascape. In *Badan Riset dan Sumber Daya Manusia Kelautan dan Perikanan*. <http://pengaduan.kkp.go.id>
- Munantri, N. Z., Sofyan, H., & Yanu, M. (2019). Aplikasi Pengolahan Citra Digital untuk Identifikasi Umur Pohon. *TELEMATIKA*, 16(2), 97–104.
- Ni'mah, F. S., Sutojo, T., & Setiadi, D. R. I. M. (2018). Identifikasi Tumbuhan Obat Herbal Berdasarkan Citra Daun Menggunakan Algoritma Gray Level Co-occurrence Matrix dan K-Nearest Neighbors. *Jurnal Teknologi Dan Sistem Komputer*, 6(2), 51–56. <https://doi.org/10.14710/jtsiskom.6.2.2018.51-56>
- Nuryadi, N., Haryati, H., & Indrayana, L. (2021). Rancang Bangun Sistem Kendali Keseimbangan Kebutuhan Tanaman Air dalam Aquascape. *Konferensi Nasional Ilmu Komputer (KONIK)*, 5.

- Putri, A. R. (2016). Pengolahan Citra dengan Menggunakan Web Cam pada Kendaraan Bergerak di Jalan Raya. *JUPI (Jurnal Ilmiah Pendidikan Informatika)*, 1(1), 1–6.
- Raharjo, S., Kurniawan, E., & Nurcahya, E. D. (2018). Sistem Otomatisasi Fotosintesis Buatan pada Aquascape Berbasis Android. *Penerbitan Artikel Ilmiah Mahasiswa Universitas Muhammadiyah Ponorogo*, 2(1), 39–48.
- Siska, M., Naufaldi, I., & Taslim, R. (2020). Desain Alat Pemberi Pupuk Cair Aquascape Otomatis Menggunakan Kansei Engineering dan Kano. *Seminar Nasional Teknologi Informasi, Komunikasi Dan Industri (SNTIKI) 12*, 511–517.
- Somvanshi, M., Chavan, P., Tambade, S., & Shinde, S. V. (2016, February 21). A review of machine learning techniques using decision tree and support vector machine. *Proceedings - 2nd International Conference on Computing, Communication, Control and Automation, ICCUBEA*. <https://doi.org/10.1109/ICCUBEA.2016.7860040>
- Sukmawati, T., Fitrihidajati, H., & Indah, N. K. (2015). Penyerapan Karbon Dioksida pada Tanaman Hutan Kota di Surabaya. *Jurnal Mahasiswa Universitas Negeri Surabaya*, 4(1), 108–111. <http://ejournal.unesa.ac.id/index.php/lenterabio>
- Widjaja, T. (2013). *Aquascape: Pesona Taman dalam Akuarium*. Agromedia Pustaka. <https://opac.perpusnas.go.id/DetailOpac.aspx?id=866518>
- Yani, A., & Saputra, B. (2018). Rancang Bangun Sistem Informasi Evaluasi Siswa dan Kehadiran Guru Berbasis Web (Studi Kasus di SMK Nusa Putra Kota Tangerang). *JURNAL PETIR*, 11(2), 107–124.
- Zhang, Z. (2016). Introduction to machine learning: K-nearest neighbors. In *Annals of Translational Medicine* (Vol. 4, Issue 11). AME Publishing Company. <https://doi.org/10.21037/atm.2016.03.37>

CURRICULUM VITAE

Data Pribadi

Nama Lengkap : Ferouz Syahalam
Tempat / Tanggal Lahir : Padang, 15 Mei 2000
Jenis Kelamin : Laki – Laki
Agama : Islam
Kebangsaan : Indonesia
Alamat : Jl. Pasar I Setiabudi, Komp. SBE E2, Medan
Telepon : +62 812 6071 0178
Email : syahalamferouz@gmail.com



Pendidikan Formal

- 2017 – 2023 S1 Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara
- 2015 – 2017 SMA Plus Al-Azhar Medan
- 2012 – 2015 SMP Panca Budi Medan
- 2009 – 2012 SDN 112143 Kec. Rantau Utara
- 2006 – 2009 SDN Percobaan Sei Petani Medan

Pengalaman Kerja

- 2020 Praktik Kerja Lapangan di Balai Pelaksana Pemilihan Jasa Konstruksi (BP2JK) Kota Medan.

Kemampuan

Bahasa : Indonesia, Inggris
Bahasa Pemrograman : PHP.
Lainnya : Ms.Office, Video Editing , Graphics Design.