

**DETEKSI KEKURANGAN NUTRISI PADA TANAMAN *LETTUCE*
ICEBERG MENGGUNAKAN METODE *AUTOENCODER* DAN
*MULTILAYER PERCEPTRON***

LAPORAN TUGAS AKHIR

Laporan ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Strata 1 (S1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang

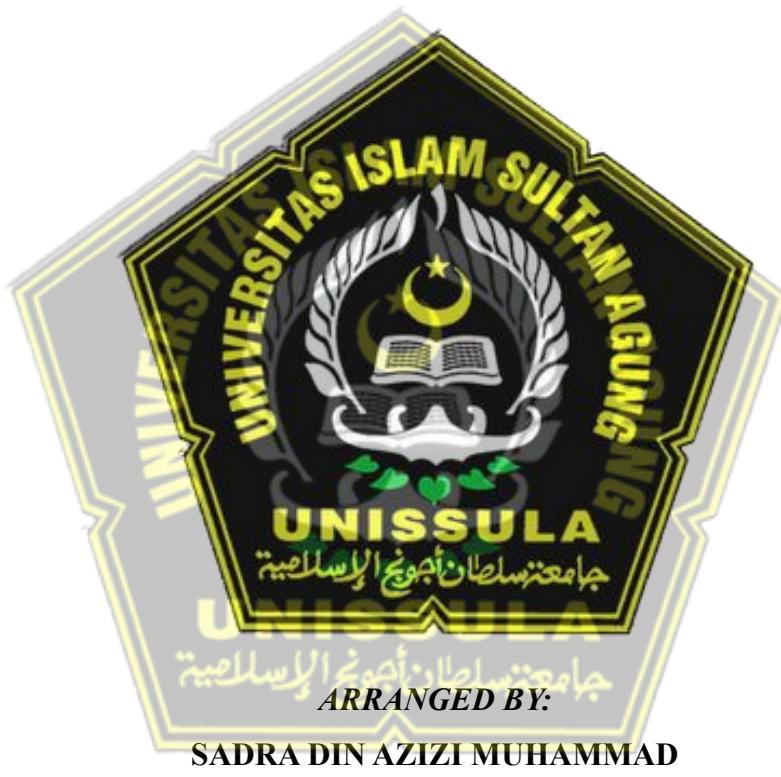


DISUSUN OLEH:
SADRA DIN AZIZI MUHAMMAD
32602100135

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG
2025**

FINAL PROJECT
DETECTION OF NUTRITION DEFICIENCY IN ICEBERG LETTUCE
PLANTS USING AUTOENCODER AND MULTILAYER PERCEPTRON
METHODS

*Proposed to complete the requirement to obtain a bachelor's degree (S-1) at
Informatics Engineering Departement of Industrial Technology
Faculty Sultan Agung Islamic University.*



MAJORING OF INFORMATICS ENGINEERING
INDUSTRIAL TECHNOLOGY FACULTY
SULTAN AGUNG ISLAMIC UNIVERSITY
SEMARANG

2025

LEMBAR PENGESAHAN
TUGAS AKHIR

DETEKSI KEKURANGAN NUTRISI PADA TANAMAN *LETTUCE ICEBERG* MENGGUNAKAN METODE *AUTOENCODER DAN MULTILAYER PERCEPTRON*

SADRA DIN AZIZI MUHAMMAD
NIM. 32602100135

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir
Program Studi Teknik Informatika
Universitas Islam Sultan Agung
Pada tanggal : 23 Oktober 2025..

TIM PENGUJI UJIAN SARJANA:

Dedy Kurniadi, M.Kom

NIK. 210615048

(Ketua Penguji)

Moch.Taufik, ST., MIT

NIK. 210604034

(Anggota Penguji)

Sam Farisa Chaerul Haviana, ST., M.Kom

NIK. 210616051

(Pembimbing)

12/11/2025

12/11/2025

13/11/2025

13 November 2025
Semarang,

Mengetahui,
Kaprodi Teknik Informatika
Universitas Islam Sultan Agung

Moch.Taufik, ST., MIT
NIK. 210604034

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

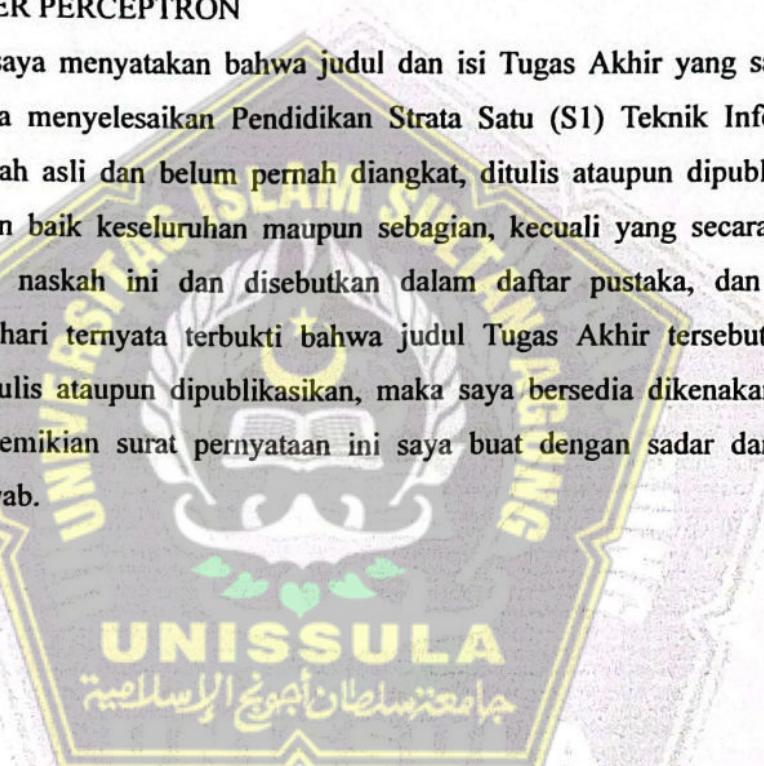
Yang bertanda tangan dibawah ini :

Nama : Sadra Din Azizi Muhammad

NIM : 32602100135

Judul Tugas Akhir : DETEKSI KEKURANGAN NUTRISI PADA TANAMAN LETTUCE ICEBERG MENGGUNAKAN METODE AUTOENCODER DAN MULTILAYER PERCEPTRON

Dengan ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila dikemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.



20 NOVEMBER 2025
Semarang,

Yang Menyatakan,



Sadra Din Azizi Muhammad

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Sadra Din Azizi Muhammad

NIM : 32602100135

Program Studi : Teknik Informatika

Fakultas : Teknologi industri

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul :
Deteksi Kekurangan Nutrisi Pada Tanaman *Lettuce Iceberg* Menggunakan Metode
Autoencoder Dan Multilayer Perceptron

Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila di kemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan agung.

Semarang, 20 November 2025

Yang menyatakan,



Sadra Din Azizi Muhammad

KATA PENGANTAR

Dengan mengucap Syukur alhamdulillah atas kehadiran Allah SWT yang telah memberikan rahmat dan karunianya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul "Deteksi Kekurangan Nutrisi Pada Tanaman *Lettuce Iceberg* Menggunakan Metode *Autoencoder* dan *Multilayer Perceptron*" ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S-1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

Tugas Akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak, materi maupun teknis, oleh karena itu saya selaku penulis mengucapkan terimakasih kepada:

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, S.H., M.H yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Ir. Novi Marlyana, ST., MT., IPU., ASEAN Eng.
3. Dosen pembimbing penulis Bapak Sam Farisa Chaerul Haviana, ST, M.Kom yang telah meluangkan waktu dan memberi ilmu. Serta memberikan banyak nasehat dan saran.
4. Orang tua penulis yang telah mengizinkan untuk menyelesaikan laporan ini,
5. Dan kepada semua pihak yang tidak dapat saya sebutkan satu persatu.

Dengan segala kerendahan hati, penulis menyadari masih terdapat banyak kekurangan dari segi kualitas atau kuantitas dari ilmu pengetahuan dalam penyusunan laporan, sehingga penulis mengharapkan adanya saran dan kritikan yang bersifat membangun demi kesempurnaan laporan ini dan masa mendatang.

Semarang,.....

Sadra Din Azizi Muhammad

DAFTAR ISI

LAPORAN TUGAS AKHIR	i
LEMBAR PENGESAHAN TUGAS AKHIR.....	iii
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH.....	v
KATA PENGANTAR.....	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	x
DAFTAR LAMPIRAN	xi
ABSTRAK	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	4
1.3 Pembatasan Masalah	4
1.4 Tujuan.....	5
1.5 Manfaat	5
1.6 Sistematika penulisan	5
BAB II TINJAUAN PUSTAKAN DAN DASAR TEORI	7
2.1 Tinjauan Pustaka	7
2.2 Dasar Teori	17
2.2.1 Defisiensi Nutrisi	17
2.2.2 Pengolahan Citra (<i>Image Processing</i>).....	18
2.2.3 <i>Autoencoder</i> dalam Ekstraksi Fitur Citra.....	21
2.2.4 Ekstraksi Fitur dengan <i>Autoencoder</i>	24
2.2.5 <i>Multilayer Perceptron</i> (MLP)	24
BAB III METODE PENELITIAN.....	28
3. 1 Metode Penelitian.....	28
3.1.1 Integrasi <i>Autoencoder</i> dan MLP	28
3.1.2 Evaluasi Model.....	33

3. 2	Analisa kebutuhan.....	35
3. 3	Penggunaan Sistem	37
3. 4	Perancangan <i>User Interface</i>	38
3.4.1	Halaman Awal Sistem	38
3.4.2	Tampilan Hasil Sistem	40
BAB IV HASIL PENELITIAN DAN IMPLEMENTASI SISTEM		42
4.1	Hasil Pengumpulan Data.....	42
4.2	Arsitektur Model Autoencoder dan Multilayer Perceptron.....	43
4.3	Implementasi Sistem Deteksi.....	45
4.3.1	<i>Preprocessing</i> Dataset.....	45
4.3.2	Pelatihan Model <i>Autoencoder</i>	46
4.3.3	Pelatihan Model <i>Multilayer Perceptron</i> (MLP).....	47
4.3.4	Evaluasi Model.....	49
4.4	Implementasi Antar Muka Menggunakan <i>Streamlit</i>	53
4.4.1	Fitur Utama Aplikasi <i>Streamlit</i>	53
4.4.2	Cara Akses <i>Streamlit</i>	54
4.4.3	Hasil Implementasi.....	54
4.5	Pembahasan Implementasi	59
BAB V KESIMPULAN DAN SARAN		61
5.1	Kesimpulan	61
5.2	Saran.....	61
DAFTAR PUSTAKA		

DAFTAR GAMBAR

Gambar 2. 1 Illustrasi <i>Image Processing</i> (Taha dkk., 2022).....	20
Gambar 2. 2 Arsitektur <i>Autoencoder</i>	22
Gambar 2. 3 Arsitektur Jaringan Saraf Tiruan MLP (Kusuma dkk., 2023)	25
Gambar 3. 1 Alur Kerja Model	29
Gambar 3. 2 Alur Sistem.....	37
Gambar 3. 3 <i>User Interface desktop</i> sebelum <i>upload</i> gambar	39
Gambar 3. 4 <i>User Interface mobile</i> sebelum <i>upload</i> gambar	39
Gambar 3. 5 <i>User Interface desktop</i> setelah <i>upload</i> gambar	40
Gambar 3. 6 <i>User Interface desktop</i> setelah <i>upload</i> gambar	41
Gambar 4. 1 Arsitektur <i>Autoencoder</i> dan MLP	44
Gambar 4. 2 Hasil <i>Preprocessing</i>	45
Gambar 4. 3 Hasil Ekstraksi <i>Autoencoder</i>	47
Gambar 4. 4 Dataset hasil prediksi menggunakan MLP	49
Gambar 4. 5 <i>Confusion Matrix</i>	51
Gambar 4. 6 Tampilan <i>home</i> pada <i>desktop</i>	54
Gambar 4. 7 Tampilan <i>home</i> pada <i>mobile</i>	55
Gambar 4. 8 Tampilan prediksi pada <i>desktop</i> sebelum <i>capture</i> gambar.....	56
Gambar 4. 9 Tampilan prediksi pada <i>mobile</i> sebelum <i>capture</i> gambar	56
Gambar 4. 10 Tampilan prediksi pada <i>desktop</i> setelah <i>capture</i> gambar	58
Gambar 4. 11 Tampilan prediksi pada <i>mobile</i> setelah <i>capture</i> gambar.....	58

DAFTAR TABEL

Tabel 4. 1 Tabel Sampel Dataset	42
Tabel 4. 2 Tabel <i>Hyperparameter Autoencoder</i>	46
Tabel 4. 3 Tabel <i>Hyperparameter MLP</i>	48
Tabel 4. 4 Tabel Evaluasi Model	52



DAFTAR LAMPIRAN



ABSTRAK

Lettuce iceberg dipilih sebagai objek karena memiliki karakteristik morfologi dan respons visual yang jelas terhadap kekurangan nutrisi tertentu. Selain itu, beberapa pemilik tanaman dengan skala kecil dan menengah, masih mengandalkan teknik identifikasi berbasis pengamatan visual yang bersifat subjektif. Penelitian ini bertujuan untuk merancang sistem deteksi kekurangan nutrisi menggunakan pendekatan *Autoencoder* dengan arsitektur *Encoder* sebagai ekstraksi fitur, *Bottleneck* sebagai kompresi data, *Decoder* sebagai rekonstruksi, dan *Multilayer Perceptron* (MLP) dengan arsitektur *input layer* sebagai pintu masuk data, *hidden layer* sebagai mengolah informasi, dan *output layer* sebagai prediksi akhir. *Sample* dataset bersumber dari *Kaggle* yang sudah dibagi menjadi empat kelas, yaitu Nitrogen, Fosfor, Kalium, dan *Healty*. Proses persiapan pemrosesan dengan mengkonversi warna BGR (*Blue, Green, Red*) menjadi HSV (*Hue, Saturation, Value*), dengan mengubah ukuran menjadi 128*128 piksel, lalu dinormalisasi menjadi skala 0-1, dan dataset *displit* dengan perbandingan 80:20. Proses pelatihan *Autoencoder* berlangsung selama 30 *epoch*, dan MLP selama 25 *epoch*. Performa kinerja MLP menggunakan *confusion matrix*, dan evaluasi model menggunakan metrik Akurasi, Presisi, *Recall*, dan *F1-Score*, di mana model mencapai nilai Akurasi 86%, Presisi 89%, *Recall* 87% dan *F1-Score* 88%. Sistem ini telah diterapkan dalam bentuk aplikasi berbasis web yang ramah pengguna, memudahkan proses pengambilan citra dan menampilkan deteksi kekurangan nutrisi. Kesimpulannya, integrasi *Autoencoder* (*encoder, bottleneck, decoder*), dan MLP (*input layer, hidden layer, output layer*) terbukti efektif untuk melakukan deteksi citra pada tanaman *lettuce iceberg* dengan tingkat akurasi yang tinggi, sehingga dapat menjadi solusi alternatif yang lebih objektif.

Kata Kunci: *Lettuce Iceberg, Image Processing, Autoencoder, Multilayer Perceptron*

ABSTRACT

Iceberg lettuce was chosen as the object because it has morphological characteristics and clear visual responses to certain nutrient deficiencies. In addition, some small and medium scale plant owners still rely on visual observation based identification techniques that are subjective. This study aims to design a nutrient deficiency detection system using an *Autoencoder* approach with an architecture consisting of an *Encoder* for feature extraction, *Bottleneck* for data compression, *Decoder* for reconstruction, and a *Multilayer Perceptron* (MLP) with an architecture consisting of an *input layer* as the data entry point, *hidden layer* for processing information, and *output layer* for final prediction. The sample dataset was obtained from *Kaggle* and divided into four classes: Nitrogen, Phosphorus, Potassium, and *Healty*. The preprocessing stage involved converting the image color from BGR (*Blue, Green, Red*) to HSV (*Hue, Saturation, Value*), resizing to 128*128 pixels, normalizing to a scale of 0-1, and splitting the dataset with an 80:20 ratio. The *Autoencoder* training process was carried out for 30 epoch, and the MLP for 25 epoch. The MLP performance was measured using a *confusion matrix*, and the model evaluation employed Accuracy, Precision, *Recall*, and *F1-Score* metrics, where the model achieved an accuracy of 86%, precision of 89%, *Recall* of 87% and an *F1-Score* of 88%. The system has been implemented in a user friendly web based application, making it easier to capture images and display nutrient deficiency detection result. In conclusion, the integration of *Autoencoder* (*encoder, bottleneck, decoder*) and MLP (*input layer, hidden layer, output layer*) proved effective in detecting images of iceberg lettuce plants with a high level of accuracy, making it a more objective alternative solution.

Keywords: *Lettuce Iceberg, Image Processing, Autoencoder, Multilayer Perceptron*

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pertumbuhan dan produktivitas tanaman sangat dipengaruhi oleh ketersediaan unsur hara yang cukup dan seimbang. Kekurangan unsur hara atau kekurangan nutrisi pada tanaman dapat menyebabkan perubahan fisiologis yang terlihat secara morfologis, khususnya pada daun, seperti klorosis, nekrosis, dan perubahan bentuk. Masalah ini tidak hanya menurunkan hasil panen, tetapi juga membimbulkan kerugian ekonomi yang signifikan bagi petani. Oleh karena itu, deteksi dini terhadap gejala kekurangan nutrisi sangat penting agar dapat segera dilakukan tindakan yang tepat.

Tanaman dapat mengalami kekurangan satu atau lebih unsur hara secara bersamaan. Diagnosis visual merupakan salah satu metode yang umum digunakan oleh petani karena tidak memerlukan alat khusus. Namun, metode ini memiliki kelemahan, yakni subjektivitas tinggi dan rentan terhadap kesalahan identifikasi, terutama jika gejala antar defisiensi serupa.(Armita dkk., 2022)

Tanaman *Lettuce* atau Selada (*Lactuca Sativa var.*), khususnya jenis *Iceberg* (Selada Bokor), merupakan salah satu komoditas hortikultura yang banyak dibudidayakan dengan menggunakan metode hidroponik karena memiliki nilai ekonomis tinggi, waktu panen yang relatif cepat antara 70 hingga 100 hari, serta permintaan pasar yang terus meningkat, terutama di sektor kuliner dan gaya hidup sehat. Namun seperti tanaman pada umumnya, *Lettuce Iceberg* sangat rentan terhadap kekurangan unsur hara seperti Nitrogen (N), fosfor (P), dan Kalium (K), yang dapat menyebabkan gangguan pertumbuhan, penurunan kualitas daun, dan kegagalan panen jika tidak ditangani dengan cepat. (Hong dkk., 2022)

Pemilihan metode hidroponik dalam konteks penelitian ini didasarkan pada karakteristik utama sistem hidroponik yang sangat bergantung pada

keseimbangan nutrisi dalam larutan sebagai sumber utama pertumbuhan tanaman. Tidak seperti metode konvensional menggunakan tanah, sistem hidroponik tidak memiliki media penyimpan hara alami, sehingga komposisi nutrisi harus terus dipantau secara ketat agar tanaman dapat tumbuh optimal. Dalam kondisi ini, bahkan gangguan yang sangat kecil terhadap komposisi unsur hara dalam larutan nutrisi dapat menimbulkan dampak signifikan terhadap Kesehatan dan pertumbuhan tanaman. Oleh karena itu, deteksi dini terhadap ketidakseimbangan atau kekurangan nutrisi menjadi aspek yang krusial, agar tindakan korektif dapat segera dilakukan sebelum kondisi tanaman memburuk.

Dalam penelitian ini, tanaman *Lettuce Iceberg* dipilih sebagai objek karena memiliki karakteristik morfologi dan respons visual yang cukup jelas terhadap kekurangan nutrisi tertentu. Selain itu, *Lettuce Iceberg* merupakan salah satu varietas tanaman hortikultura yang popular dibudidayakan secara hidroponik, baik konsumsi rumah tangga maupun produksi skala komersial.

Namun, kenyataannya di lapangan menunjukkan bahwa sebagian besar pemilik tanaman *Lettuce Iceberg*, khususnya dalam skala kecil dan menengah, masih mengandalkan teknik identifikasi berbasis pengamatan visual secara langsung dan pengalaman pribadi untuk menilai kondisi nutrisi tanaman. Pendekatan ini cenderung bersifat subjektif, tidak konsisten, serta bergantung pada intuisi dan keterampilan individu, yang menyebabkan risiko menimbulkan kesalahan dalam diagnosis maupun penanganan yang dilakukan.

Dalam perkembangan teknologi, pendekatan berbasis kecerdasan buatan (AI), khususnya *Deep Learning*, mulai banyak digunakan dalam bidang bertanian presisi. Penelitian yang dilakukan oleh Oktavia dkk. (2022) dari Universitas Telkom, menunjukkan bahwa klasifikasi defisiensi nutrisi tanaman padi menggunakan CNN dengan arsitektur *ResNet-50* mampu mencapai akurasi tinggi hingga 97,73% dengan citra daun sebagai *input* klasifikasi.(Oktavia dkk., 2022)

Meski CNN memberikan performa tinggi, arsitekturnya tergolong kompleks dan memerlukan sumber daya komputasi besar. Oleh karena itu, alternatif pendekatan yang lebih ringan seperti penggunaan *Autoencoder* untuk ekstraksi fitur dan *Multilayer Perceptron* (MLP) untuk deteksi dapat menjadi solusi efisien. *Autoencoder* adalah jaringan saraf tiruan yang digunakan untuk melakukan reduksi dimensi dan ekstraksi fitur dari data yang dimasukan seperti gambar daun. Dengan memproses gambar *Autoencoder*, sistem dapat mengubah citra daun yang kompleks menjadi representasi fitur numerik yang lebih sederhana, namun tetap menyimpan informasi seperti tekstur, pola, dan perubahan warna akibat kekurangan nutrisi.

Fitur hasil ekstraksi dari *Autoencoder* ini kemudian dijadikan *input* untuk model *Multilayer Perceptron* (MLP), yaitu jaringan saraf *feedforward* yang mampu melakukan deteksi berdasarkan pola-pola *non-linear*. Dengan konfigurasi yang tepat, MLP dapat mengenali jenis-jenis penyakit dari pola visual yang telah diproses sebelumnya oleh *Autoencoder*.

Kombinasi antara *Autoencoder* dan MLP memberikan keunggulan dalam hal efisiensi pemrosesan citra dan akurasi deteksi. *Autoencoder* bertindak sebagai alat penyaring dan peringkas informasi, sementara MLP berperan sebagai pengambilan keputusan berdasarkan informasi tersebut. Oleh karena itu, penelitian ini bertujuan untuk membangun sistem deteksi kekurangan nutrisi pada tanaman menggunakan pendekatan gabungan antara *Autoencoder* dan *Multilayer Perceptron*, dengan data berupa citra daun sebagai kontribusi bagi penerapan teknologi dalam sektor pertanian.

Metode *Autoencoder* dan *Multilayer Perceptron* dipilih karena memiliki arsitektur yang relatif lebih sederhana dan ringan dibandingkan dengan metode lain seperti *Convolutional Neural Network* (CNN). *Autoencoder* mampu melakukan ekstraksi fitur secara otomatis dari citra *input* tanpa perlu desain fitur secara manual, dan cocok diterapkan pada dataset yang terbatas.

Sementara itu, *Multilayer Perceptron* (MLP) dipilih sebagai *classifier* karena efektif dalam mengolah representasi vektor hasil ekstraksi fitur dari *Encoder*. Selain kemudahannya dalam implementasi dan efisiensi dalam pelatihan tanpa memerlukan sumber daya komputasi yang tinggi, MLP juga menawarkan fleksibilitas tinggi dalam desain arsitekturnya. Dengan *Multilayer* yang menunjukkan bahwa model ini dapat terdiri dari lebih dari satu lapisan (*Layer*), sehingga memungkinkan untuk membangun model yang lebih kompleks dan mampu mempelajari hubungan *non-linear* antara fitur dan label target. Fleksibilitas ini sangat berguna dalam konteks klasifikasi citra daun, di mana pola visual yang mewakili kekurangan nutrisi sering kali bersifat halus dan memerlukan pemrosesan hierarkis untuk dipahami secara efektif. Dengan demikian, MLP menjadi pilihan yang tepat sebagai tahap klasifikasi akhir dalam sistem deteksi kekurangan nutrisi berbasis *Autoencoder*, karena mampu menjembatani representasi fitur dengan hasil klasifikasi.

1.2 Perumusan Masalah

Pemilik tanaman *Lettuce Iceberg* masih mengandalkan pengamatan visual dan pengalaman pribadi yang cenderung subjektif dan tidak akurat dalam mengidentifikasi kekurangan nutrisi pada tanaman, sehingga sering kali menyebabkan kesalahan diagnosis dan penanganan yang tidak tepat.

1.3 Pembatasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Penelitian ini hanya membahas deteksi kekurangan nutrisi pada satu jenis tanaman hortikultura, yaitu *Lettuce Iceberg*.
2. Data yang digunakan berupa citra daun tanaman yang sehat dan menunjukkan gejala kekurangan unsur hara tertentu (N, P, K).
3. Sistem tidak membahas penanganan kekurangan nutrisi, melainkan hanya fokus pada proses deteksi gejala dari citra daun.

1.4 Tujuan

Menerapkan metode *image processing* untuk mengolah dan menyiapkan data citra daun dari tanaman *Lettuce Iceberg* sebagai *input* bagi sistem deteksi kekurangan nutrisi.

1.5 Manfaat

Manfaat dari Tugas Akhir ini adalah sebagai berikut:

1. Penelitian ini memberikan solusi praktis bagi pemilik tanaman *Lettuce Iceberg* dalam mendeteksi secara dini kekurangan nutrisi pada tanaman *Lettuce Iceberg*. Dengan adanya sistem deteksi berbasis citra digital, tindakan perbaikan dapat dilakukan secara lebih cepat dan efisien tanpa harus bergantung pada pemeriksaan visual manual yang bersifat subjektif.
2. Penelitian ini memberikan kontribusi dalam pengembangan penerapan teknologi kecerdasan buatan dibidang pertanian, khususnya melalui pemandangan metode *Autoencoder* dan *Multilayer Perceptron*.

1.6 Sistematika penulisan

Untuk mempermudah penulisan tugas akhir ini, penulis membuat suatu sistematika yang terdiri dari:

BAB I: PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang pemilihan judul tugas akhir "DETEKSI KEKURANGAN NUTRISI PADA TANAMAN *LETTUCE ICEBERG* MENGGUNAKAN METODE *AUTOENCODER* DAN *MULTILAYER PERCEPTRON*". Rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II: TINJAUAN PUSTAKA DAN DASAR TEORI

Bab ini bertujuan memuat dasar teori yang berfungsi sebagai sumber dalam memahami permasalahan yang dipilih.

BAB III: METODE PENELITIAN

Bab ini menjelaskan proses tahapan-tahapan penelitian dimulai dari Analisa kebutuhan sistem, kemudian perancangan sistem hingga selesai dibuat.

BAB IV: HASIL PENELITIAN DAN IMPLEMENTASI SISTEM

Bab ini menjelaskan hasil penelitian berupa teks dari gambar tanaman *Lettuce* berdasarkan deteksi kekurangan nutrisi menggunakan *Autoencoder* dan MLP.

BAB V: KESIMPULAN DAN SARAN

Bab ini memuat Kesimpulan dari keseluruhan uraian bab-bab sebelumnya dan saran-saran dari hasil yang diperoleh dan diharapkan dapat bermanfaat dalam penelitian selanjutnya.



BAB II

TINJAUAN PUSTAKAN DAN DASAR TEORI

2.1 Tinjauan Pustaka

Dalam penelitian oleh Taha dkk (2022), pendekatan berbasis pengolahan citra digital dan *Deep Learning* telah diterapkan untuk mendeteksi defisiensi nutrisi pada tanaman *Lettuce*. Penelitian ini menunjukkan bahwa fitur visual seperti perubahan warna, bentuk, dan tekstur daun akibat kekurangan Nitrogen, fosfor, dan Kalium dapat diklasifikasikan secara otomatis menggunakan algoritma seperti *InceptionV3* dan *ResNet18*. Proses deteksi dimulai dari akuisisi citra, segmentasi daun, ekstraksi fitur visual, hingga klasifikasi menggunakan model pembelajaran mesin. Hasil evaluasi menunjukkan akurasi tinggi, membuktikan bahwa teknologi ini dapat menjadi solusi efektif dalam mendiagnosis defisiensi nutrisi secara cepat dan objektif. (Taha dkk., 2022)

Penelitian oleh Vought dkk (2024), memahami efisiensi penggunaan nutrisi dan potensi kehilangan unsur hara utama seperti Nitrogen (N), fosfor (P), dan Kalium (K) dalam sistem hidroponik berbasis *Nutrient Film Technique* (NFT), khususnya untuk tanaman daun seperti selada. Dalam sistem NFT, larutan nutrisi mengalir secara terus-menerus di sekitar akar tanaman, sehingga tanaman sangat bergantung pada komposisi nutrisi dalam larutan tersebut. Selain itu, mengungkap bahwa sistem NFT dirancang untuk efisiensi tinggi, terdapat tingkat kehilangan nutrisi yang cukup signifikan selama masa pertumbuhan tanaman. Mereka mencatat bahwa kehilangan Nitrogen dari larutan nutrisi bisa mencapai 27-40%, sementara kehilangan fosfor berkisar antara 11-35%, tergantung pada fase pertumbuhan dan jenis manajemen larutan yang digunakan. Faktor penyebab kehilangan nutrisi ini meliputi penguapan, presipitasi senyawa dalam bentuk yang tidak dapat diserap tanaman, penumpukan residu pada sistem, serta adsorpsi nutrient pada media tanam atau dinding saluran hidroponik. Penelitian ini menjadi dasar penting dalam mengembangkan sistem monitoring dan manajemen

nutrisi berbasis sensor maupun kecerdasan buatan, untuk memastikan bahwa suplai nutrisi tetap dalam kondisi optimal dan terhindar dari risiko defisiensi. (Vought dkk., 2024)

Penelitian oleh Hong dkk (2022), memberikan kontribusi signifikan terhadap pemahaman tentang pengaruh pemupukan *makronutrien* terhadap pertumbuhan dan kualitas tanaman selada. Dalam studi ini, para peneliti mengevaluasi efek dari berbagai dosis Nitrogen, fosfor, dan Kalium terhadap parameter agronomis dan kualitas hasil panen, termasuk tinggi tanaman, jumlah daun, bobot segar, serta kandungan nutrisi seperti vitamin C. Hasil penelitian menunjukkan bahwa aplikasi Nitrogen hingga dosis tertentu mampu meningkatkan tinggi tanaman dan bobot hasil secara signifikan, tetapi pemberian Nitrogen secara berlebihan justru menurunkan efisiensi pemanfaatan nutrisi oleh tanaman. Sementara itu, pemberian fosfor terbukti meningkatkan diameter batang dan kualitas fisiologis daun, namun aplikasi fosfor yang terlalu tinggi juga menimbulkan penurunan kualitas hasil panen. Kalium, disisi lain diketahui sangat berpengaruh terhadap kualitas tekstur daun dan diameter batang, serta memiliki peran penting dalam menjaga keseimbangan air dan metabolisme tanaman, meskipun dalam jumlah berlebihan dapat menghambat penyerapan unsur hara lain dan menyebabkan penurunan kualitas daun. (Hong dkk., 2022)

Penelitian oleh Lu dkk (2023) ini menyoroti penggunaan pendekatan *machine vision* dalam mengenali gejala kekurangan unsur hara mikro pada tanaman selada (*Lactuca Savita L.*) dengan menerapkan metode *pattern recognition* berbasis citra. Penelitian ini memanfaatkan beberapa jenis fitur visual yang diperoleh dari gambar daun, yaitu fitur warna, tekstur, dan bentuk. Fitur warna diekstraksi melalui *color moments* dan *histogram* warna RGB (*Red, Green, Blue*) untuk menangkap distribusi intensitas warna secara menyeluruh, yang umumnya terdistorsi akibat kekurangan unsur hara seperti Nitrogen, fosfor dan Kalium. Setelah itu, fitur tekstur diolah dengan memanfaatkan *Gray-Level Co-occurrence Matrix* (GLCM) yang menghitung parameter seperti *Angular Second Moment, Entropy, Contrast*,

dan *Correlation* pada berbagai sudut orientasi citra (0° , 45° , 90° , dan 135°), untuk mendeteksi kehalusan atau kekasaran tekstur yang biasanya muncul saat terjadi defisiensi nutrisi. Disisi lain, bentuk dan pola tepi daun juga menjadi pertimbangan penting yang diekstraksi menggunakan metode *Scale-Invariant Feature Transform* (SIFT), yang berguna untuk mendeteksi perubahan struktur fisik daun, seperti kerusakan atau bercak. (Lu dkk., 2023)

Penelitian oleh Xie dkk (2025) melakukan eksplorasi pendekatan multidimensi dalam analisis citra digital untuk mendeteksi kekurangan unsur hara pada tanaman selada secara presisi. Penelitian ini memfokuskan pada penerapan metode *advance pattern recognition* berbasis analisis citra multidimensi, yang menggabungkan tiga teknik yaitu *adaptive threshold segmentation*, *Canny edge detection*, dan *gradient-based region refinement*. Proses dimulai dengan konversi citra kedalam ruang warna HSV (*Hue*, *Saturation*, *Value*) guna memisahkan warna daun latar belakang dengan lebih akurat, karena perubahan warna daun merupakan salah satu indikator paling signifikan dalam defisiensi nutrisi. Setelah itu, dilakukan segmentasi awal menggunakan *threshold* adaptif untuk menangkap area yang mencurigakan secara global. Teknik *Canny edge detection* yang telah dimodifikasi untuk mendeteksi kontur tepi daun dari berbagai arah digunakan untuk meningkatkan ketelitian dalam menentukan batas antar jaringan sehat dan jaringan yang mengalami defisiensi. Langkah akhir menggunakan pendekatan *gradient-guided adaptive segmentation* untuk menyempurnakan hasil segmentasi, terutama untuk memperjelas area yang memiliki variasi tekstur dan perubahan warna halus. (Xie dkk., 2025)

Penelitian oleh Okasha dkk (2022) dijelaskan bahwa budidaya *Iceberg Lettuce* dalam sistem hidroponik dapat dioptimalkan dengan memanfaatkan air irigasi asin (*saline water*) yang telah diproses secara *magnetic*. Studi ini bertujuan untuk meningkatkan produktivitas air dan hasil tanaman melalui pendekatan teknologi yang inovatif, dengan menggunakan dua varietas *Iceberg Lettuce*. Hasil dari penelitian menunjukkan bahwa kombinasi antara

air irigasi yang dimagnetisasi dan pengaturan sudut kemiringan pipa hidroponik mampu memberikan pengaruh signifikan terhadap peningkatan berat segar tanaman (*fresh weight*), efisiensi penggunaan air, serta mutu tanaman secara keseluruhan. Salah satu temuan penting dari studi ini adalah bahwa perlakuan dengan sudut kemiringan pipa 2% menggunakan air yang telah dimagnetisasi mampu meningkatkan hasil tanaman hingga lebih dari 70% dibandingkan dengan kontrol. Penelitian ini memperkuat pentingnya faktor lingkungan teknis dalam pertumbuhan *Iceberg Lettuce* serta memberikan wawasan bahwa pengelolaan kondisi hidroponik secara tepat dapat berkontribusi dalam mendukung performa visual dan fisiologis tanaman, yang juga berkaitan erat dengan proses deteksi kekurangan nutrisi berbasis citra daun. (Okasha dkk., 2022)

Penelitian oleh Vanacore dkk (2024) berfokus pada perbandingan pertumbuhan, respons ekofisiologis, dan komposisi mineral daun dari tanaman *Lettuce Iceberg* yang dibudidayakan dalam dua sistem berbeda, yaitu hidroponik dan akuaponik. Studi ini bertujuan untuk mengetahui sistem mana yang lebih optimal dalam mendukung pertumbuhan dan kualitas daun selada. Berdasarkan hasil pengamatan, tanaman yang ditanam dalam sistem hidroponik menunjukkan kinerja yang lebih unggul dibandingkan dengan yang dibudidayakan dalam sistem akuaponik. Hal ini ditunjukkan dari hasil biomassa daun segar yang lebih tinggi sebesar 26%, kandungan kandungan Kalium dalam daun meningkat sekitar 63%, serta nilai SPAD (*Soil Plant Analysis Development*) dan laju fotosintesis yang lebih baik. Selain itu, sistem hidroponik juga menghasilkan efisiensi penggunaan air yang lebih tinggi, menjadikannya sebagai pilihan sistem budidaya yang lebih efisien dan productif untuk tanaman selada. Temuan ini sangat relevan dalam konteks pengolahan citra daun, karena berbagai parameter fisiologis yang diukur dalam penelitian ini, seperti warna, tekstur, dan tingkat kehijauan daun, secara langsung memengaruhi fitur visual yang dapat digunakan dalam proses ekstraksi ciri dan klasifikasi pada sistem

berbasis pengolahan citra digital untuk deteksi defisiensi nutrisi. (Vanacore dkk., 2024)

Penelitian oleh Bedi & Gole (2021) menjelaskan bahwa *Autoencoder*, khususnya dalam bentuk *Convolutional Autoencoder* (CAE), sangat efektif digunakan untuk proses ekstraksi fitur dalam klasifikasi citra tanaman. Dalam studi ini, peneliti mengembangkan arsitektur hibrida yang menggabungkan CAE untuk tahap ekstraksi fitur dan CNN untuk proses klasifikasinya. *Autoencoder* digunakan untuk mempelajari representasi visual yang lebih dalam dan abstrak dari citra daun tanaman yang terdampak penyakit, seperti variasi warna, tekstur, dan pola morfologis lainnya. CAE dalam sistem ini mampu mereduksi dimensi *input* dan mengekstraksi fitur laten penting dari citra tanaman secara efisien. Menariknya, meskipun hanya memiliki 9.900 parameter, sistem mampu mencapai akurasi lebih dari 98% dalam mengklasifikasikan berbagai jenis penyakit tanaman, yang menunjukkan bahwa fitur-fitur yang dipelajari dari *Autoencoder* sangat informatif. Hal ini menegaskan peran penting *Autoencoder* dalam menyaring informasi visual mentah menjadi representasi yang bermakna. (Bedi & Gole, 2021)

Penelitian oleh Huddar dkk (2024), menunjukkan pendekatan yang lebih kompleks dengan memanfaatkan *Denoising Autoencoder* (DAE) untuk meningkatkan kualitas citra dan sekaligus melakukan ekstraksi fitur. Dalam studi ini, DAE digunakan untuk menghilangkan noise dari citra daun tanaman yang diperoleh dari lapangan. Proses denoising ini membantu model untuk fokus hanya pada fitur penting yang berkaitan dengan gejala penyakit atau kekurangan nutrisi, seperti perubahan warna dan tekstur daun. Setelah citra diperbaiki, berbagai jenis fitur seperti tekstur, warna, dan wavelet diekstraksi dan dikombinasikan untuk membentuk satu set fitur yang kaya dan informatif. Hasil dari ekstraksi ini kemudian diberikan kepada model klasifikasi berbasis SVM. Berdasarkan evaluasi, sistem berhasil mencapai akurasi klasifikasi lebih dari 98%, yang bahkan melebihi metode konvensional yang hanya menggunakan satu jenis fitur atau CNN

saja. Temuan ini menegaskan bahwa *Autoencoder* tidak hanya berguna untuk reduksi dimensi atau pembelajaran representasi, tetapi juga efektif dalam meningkatkan kualitas data sebelum klasifikasi. Oleh karena itu, dalam konteks sistem klasifikasi tanaman berbasis citra seperti *Lettuce Iceberg*, penggunaan *Autoencoder* terbukti menjadi pendekatan yang strategis untuk menghasilkan *input* yang representatif bagi model klasifikasi. (Huddar dkk., 2024)

Penelitian oleh Iatrou dkk (2022), Penelitian yang membahas tentang pemanfaatan *Variational Autoencoder* (VAE), sebuah arsitektur lanjutan dari *Autoencoder* konvensional, yang dirancang untuk menghasilkan representasi laten yang lebih terdistribusi dan bersifat probabilistik, guna mendekripsi dan memprediksi kebutuhan Nitrogen pada tanaman padi. Dalam studi ini, digunakan data citra multispectral dari satelit *Sentinel-2* yang mencakup informasi NDRE (*Normalized Difference Red Edge Index*). VAE dilatih untuk mengekstraksi fitur-fitur penting dari citra tersebut dan membentuk representasi laten yang mampu merepresentasikan kondisi fisiologis tanaman secara efisien. Hasil ekstraksi fitur ini kemudian digunakan untuk memprediksi jumlah Nitrogen yang dibutuhkan oleh tanaman pagi secara lebih akurat jika dibandingkan dengan metode konvensional seperti *Principal Component Analysis* (PCA) atau *Partial Least Square Regression* (PLSR). Penelitian ini menunjukkan bahwa penggunaan VAE mampu meningkatkan akurasi prediksi kebutuhan Nitrogen secara signifikan, yang pada akhirnya dapat digunakan sebagai dasar pengambilan keputusan dalam manajemen pemupukan secara lebih presisi dan berkelanjutan. (Iatrou dkk., 2022)

Penelitian oleh Kolhar dkk (2024), membahas tentang kerangka *Deep Learning* yang efisien dan menggabungkan *Xception* model, *Vision Transformer*, serta model *MLP-Mixer* yang berbasis *Multilayer Perceptron* untuk mengidentifikasi defisiensi Nitrogen (N), fosfor (P), dan Kalium (K) pada tanaman padi menggunakan citra RGB daun. Model-model ini diuji secara langsung untuk mengetahui sejauh mana *MLP-Mixer* mampu

bersaing dengan arsitektur CNN dan *transformer* modern. Hasil penelitian menunjukkan bahwa ketiga model berhasil mencapai tingkat akurasi di atas 92%, dengan *Xception* yang mencapai akurasi sebesar 95,14%, meskipun jumlah parameter pelatihannya hanya sekitar 1,2 juta, jauh lebih sedikit dibandingkan arsitektur *Vision Transformer*. Model *MLP-Mixer* berbasis MLP juga menunjukkan performa yang kompetitif, memberikan bukti bahwa MLP mampu menangani representasi citra dengan baik meskipun tanpa menggunakan lapisan konvolusional tradisional. Studi ini menegaskan bahwa MLP tidak hanya cocok diterapkan didomain citra secara umum, tetapi juga layak digunakan dalam klasifikasi kondisi nutrisi tanaman berdasarkan citra daun. (Kolhar dkk., 2024)

Penelitian oleh Song dkk (2023) membuat sebuah pendekatan berbasis *Multilayer Perceptron* dan *Artificial Neural Network* (MLP-ANN) untuk mengklasifikasikan status nutrisi tanaman berdasarkan fitur warna daun. Penelitian ini menekankan pada pentingnya visual daun, khususnya warna sebagai indikator utama untuk mendeteksi defisiensi unsur hara makro seperti Nitrogen (N), fosfor (P), dan Kalium (K). data yang digunakan dalam penelitian ini berupa citra daun tanaman yang kemudian diolah untuk diambil nilai rata-rata dari *channel* warna RGB (*Red*, *Green*, *Blue*), yang selanjutnya dijadikan sebagai *input vector* fitur ke dalam jaringan MLP. Arsitektur MLP yang digunakan terdiri dari tiga lapisan utama: *input layer*, yang menerima data fitur warna, *hidden layer* yang melakukan proses pembelajaran dengan fungsi aktivasi ReLU untuk memetakan hubungan *non-linear*, serta *output layer* yang terdiri dari beberapa neuron sesuai jumlah kelas dan menggunakan fungsi aktivasi *Softmax* untuk menghasilkan probabilitas klasifikasi. Proses pelatihan dilakukan dengan metode *backpropagation* dan optimasi fungsi *loss* menggunakan algoritma *gradient descent*, yang memungkinkan model mempelajari representasi kompleks dari hubungan warna daun terhadap status nutrisi. Dari segi hasil dan evaluasi, model MLP-ANN menunjukkan performa klasifikasi yang cukup tinggi. Dalam pengujian terhadap dataset yang telah dipisah menjadi data

pelatihan dan data pengujian, sistem berhasil mencapai akurasi sebesar 87,8%. Selain itu, metrik evaluasi lain seperti *precision*, *Recall*, dan *F1-Score* juga menunjukkan nilai yang tinggi dan seimbang, menandakan bahwa model tidak hanya akurat secara umum, tetapi juga konsisten dalam membedakan antar kelas defisiensi yang berbeda. (Song dkk., 2023)

Penelitian oleh Veazie dkk (2024), mengembangkan model klasifikasi berbasis *Machine Learning* untuk menginterpretasikan analisis jaringan daun *Lettuce*, yang terdiri dari 1.950 sampel daun yang diambil dari berbagai eksperimen serta data laboratorium *diagnostic* nasional. Setiap sampel diberi label dalam lima kategori nutrisi (*deficient*, *low*, *sufficient*, *high*, *excessive*) untuk 11 unsur hara utama (termasuk N, P, K, Ca, Mg, dan S), berdasarkan rentang *interpretative* distribusi *Gamma* atau *Weibull* yang telah diturunkan dari data luas tersebut. Model *Machine Learning* yang diuji mencakup *decision tree* (J48), *Random Forest* (RF), *Sequential Minimal Optimization* (SMO) sebagai algoritma SVM, serta *Multilayer Perceptron* (MLP). Kinerja masing-masing model dievaluasi menggunakan dua teknik *cross-validation*: 66% *split* dan 10-fold stratified *cross-validation*. Metode evaluasi yang diukur meliputi *Percentage Correct Classification* (PCC), *Kappa Statistic*, dan *ROC Score*. Hasil akhir menunjukkan bahwa MLP memiliki performa baik, tetapi secara umum *Random Forest* dan J48 memberikan akurasi tertinggi pada *hamper* semua unsur hara yang diuji. Sebagai contoh, untuk beberapa unsur seperti magnesium dan sulfur, *Random Forest* mencapai akurasi hingga 99%, sedangkan MLP lebih rendah konsisten. Meski demikian, MLP tetap mempertahankan performa signifikan dengan nilai PCC dan *Kappa* yang menunjukkan klasifikasi yang jauh diatas acak ($ROC > 0.5$), menandakan model bekerja dengan baik. (Veazie dkk., 2024)

Multilayer Perceptron (MLP) merupakan arsitektur jaringan saraf tiruan yang efektif dalam mengklasifikasikan penyakit tanaman berdasarkan citra daun. Dalam penelitian oleh Naeem dkk (2021), MLP digunakan untuk mengidentifikasi enam jenis daun tanaman obat. Penelitian ini

memanfaatkan kombinasi fitur *multispectral* dan tekstur dari citra daun untuk menghasilkan representasi data yang lebih kaya. Hasil penelitian menunjukkan bahwa MLP mampu mengolah fitur-fitur citra secara efisien dan memberikan akurasi klasifikasi yang tinggi, yaitu antara 98% hingga 99%, menjadikannya sebagai metode yang potensial dalam pengenalan jenis daun berdasarkan citra. (Naeem dkk., 2021)

Penelitian oleh Pacal dkk (2024) memberikan tinjauan menyeluruh terhadap 160 artikel ilmiah yang terbit antara tahun 2020 hingga 2024. Kajian ini bertujuan untuk mengevaluasi perkembangan dan efektivitas berbagai teknik *Deep Learning* dalam mendekripsi serta mengklasifikasi penyakit tanaman melalui citra daun. Fokus utama dalam studi ini adalah pemanfaatan metode *Autoencoder* untuk ekstraksi fitur dan *Multilayer Perceptron* (MLP) sebagai model klasifikasi. Dari hasil analisis yang dilakukan, ditemukan kombinasi antara *Autoencoder* dan MLP sering digunakan dalam berbagai penelitian karena kemampuannya dalam menyaring fitur penting dari citra daun dan menerapkannya secara akurat dalam proses klasifikasi penyakit. Secara keseluruhan, akurasi yang dicapai dalam studi-studi tersebut berkisar antara 92% hingga 99%, menunjukkan bahwa pendekatan ini cukup untuk digunakan dalam sistem diagnosis otomatis berbasis citra. (Pacal dkk., 2024)

Penelitian oleh Al-Safaar & Al-Yaseen (2023), membahas tentang pendekatan hibrida berbasis *Deep Learning* dengan mengintegrasikan *Autoencoder* (AE) dan *Multilayer Perceptron* (MLP) untuk membangun sistem deteksi yang efisien dan akurat. Studi ini menyoroti keunggulan penggunaan *Autoencoder* sebagai metode *unsupervised feature extraction*, di mana *Autoencoder* dirancang untuk mempelajari representasi laten dari data masukan secara otomatis dengan cara melakukan kompresi dan rekonstruksi. Proses *encoding* yang dilakukan oleh *Autoencoder* bertujuan untuk mereduksi kompleksitas data asli dengan tetap mempertahankan fitur penting yang paling representatif. Hasil dari *encoding* ini kemudian digunakan sebagai *input* bagi MLP, yang berperan sebagai pengklasifikasi

utama dalam arsitektur ini. MLP sebagai model *supervised learning* dilatih untuk mengenali pola-pola dari representasi fitur yang telah dipadatkan oleh *Autoencoder*. Dalam eksperimen penelitian ini, pendekatan AE-MLP diterapkan pada sistem deteksi intrusi jaringan, namun prinsip dan arsitekturnya sangat relevan untuk diterapkan pada bidang pengolahan citra, khususnya dalam klasifikasi berbasis gambar seperti mendekripsi defisiensi nutrisi pada daun tanaman. (Al-Safaar & Al-Yaseen, 2023)

Penelitian oleh Boukhlifah & Chibani (2024) memperkenalkan arsitektur jaringan saraf konvolusional (CNN) yang dirancang secara ringan dengan memanfaatkan pendekatan *Convolutional Autoencoder* (CAE) untuk melakukan identifikasi penyakit pada daun tanaman tomat. Pendekatan ini dirancang untuk menghasilkan sistem klasifikasi yang efisien namun tetap memiliki akurasi tinggi. Dalam metode ini, CAE dimanfaatkan sebagai tahap awal untuk mengekstraksi fitur-fitur penting dari citra daun tomat yang terinfeksi. Hasil ekstraksi tersebut kemudian dijadikan *input* untuk proses klasifikasi menggunakan CNN, yang bertugas menentukan jenis penyakit berdasarkan pola visual yang telah diolah. Model ini dievaluasi menggunakan dataset *PlantVillage* yang mencakup 10 kelas penyakit daun tomat, dan berhasil mencapai tingkat akurasi sebesar 99,13%. Akurasi yang tinggi ini menunjukkan bahwa kombinasi CAE dan CNN sangat efektif dalam menangani klasifikasi citra. (Boukhlifa & Chibani, 2024)

Penelitian oleh Sikati & Nouaze (2023) mengusulkan sistem klasifikasi defisiensi nutrisi berbasis *real-time* untuk tanaman selada hidroponik. Model ini memanfaatkan YOLOv8 *Nano*, varian dari arsitektur YOLO yang dioptimalkan untuk kecepatan dan efisiensi. Penelitian ini merespons kebutuhan akan sistem klasifikasi yang dapat diimplementasikan langsung di lapangan tanpa perangkat keras yang mahal. Dataset citra daun selada yang menunjukkan defisiensi N, P, dan K digunakan dalam pelatihan model. Hasilnya, model ini tidak hanya mampu mencapai akurasi lebih dari 85%, tetapi juga dapat memproses setiap citra dalam waktu inferensi kurang dari 170 milidetik, yang menunjukkan potensi besar untuk penerapan

langsung di sistem *monitoring* pertanian berbasis *Internet of Things* (IoT) atau perangkat mobile. (Sikati & Nouaze, 2023)

Penelitian oleh Adianggiali dkk (2023) mengkaji pendekatan klasifikasi defisiensi nutrisi pada tanaman selada hidroponik dengan memanfaatkan arsitektur *Deep Learning* ringan, yaitu *MobileNetV2*. Penelitian ini dilatarbelakangi oleh pentingnya deteksi dini terhadap kekurangan nutrisi pada tanaman hidroponik, yang kerap menunjukkan gejala visual pada daun. Metode yang digunakan melibatkan pengumpulan dataset citra daun selada dari *platform* publik seperti Kaggle dan roboflow, yang kemudian diproses melalui tahapan *Preprocessing*, augmentasi, serta pelatihan model menggunakan *platform google colab*. Model *MobileNetV2* dipilih karena keunggulannya dalam efisiensi komputasi tanpa mengorbankan akurasi. Hasil evaluasi menunjukkan bahwa model ini mampu mencapai tingkat akurasi sebesar 88%, membuktikan bahwa arsitektur *lightweight CNN* dapat diterapkan secara praktis dalam mendekripsi kekurangan nutrisi secara visual pada sistem pertanian modern berbasis hidroponik. (Adianggiali dkk., 2023)

2.2 Dasar Teori

2.2.1 Defisiensi Nutrisi

Defisiensi nutrisi merupakan kondisi di mana tanaman tidak memperoleh jumlah unsur hara esensial yang mencukupi untuk menjalankan fungsi fisiologis secara optimal. Dalam budidaya hidroponik, kondisi ini sangat rentan terjadi mengingat seluruh kebutuhan nutrisi tanaman disuplai secara penuh melalui laurtan nutrisi. Ketidak seimbangan dalam komposisi larutan, baik karena kekurangan maupun kelebihan unsur tertentu, dapat menyebabkan gejala fisiologis yang khas dan menurunkan kualitas serta produktivitas tanaman.

Menurut Djidonou & Leskovar (2019), tanaman selada yang dibudidayakan secara hidroponik menunjukkan gejala visual yang spesifik terhadap kekurangan beberapa unsur hara makro dan mikro. Misalnya,

kekurangan Nitrogen (N) menyebabkan klorosis atau perubahan warna daun menjadi kuning dimulai dari daun-daun tua. Fosfor (P) yang rendah ditunjukkan oleh daun yang berwarna hijau gelap dengan potensi pigmentasi keunguan, sedangkan defisiensi Kalium (K) menyebabkan bercak nekrotik atau kematian jaringan pada daun. (Djidonou & Leskovar, 2019)

Seiring dengan kemajuan teknologi, pendekatan konvensional berbasis pengamatan visual yang bersifat subjektif dan memerlukan keahlian khusus dinilai kurang efisien untuk mendeteksi defisiensi nutrisi secara akurat dan cepat. Oleh karena itu, metode berbasis citra digital dan kecerdasan buatan mulai dikembangkan sebagai solusi modern untuk mengidentifikasi kekurangan unsur hara pada tanaman. Menurut Pandey dkk (2003) menjelaskan bahwa penggunaan teknologi *hyperspectral imaging* (HSI) yang dikombinasikan dengan algoritma *Deep Learning* dapat secara signifikan meningkatkan akurasi deteksi gejala defisiensi pada daun tanaman *Lettuce*. Setiap jenis kekurangan nutrisi menunjukkan pola *spectral* unik yang dapat dianalisis secara otomatis menggunakan model pembelajaran mesin, sehingga memungkinkan sistem mendeteksi defisiensi dengan tingkat presisi tinggi bahkan dalam kondisi pencahayaan yang bervariasi. (Pandey dkk., 2023)

Penerapan pendekatan ini tidak hanya mengurangi ketergantungan pada pengamatan subjektif, tetapi juga mempercepat proses diagnosis dan pengambilan keputusan budidaya tanaman, terutama dalam sistem hidroponik yang sensitif terhadap perubahan komposisi nutrisi. Dengan dukuran data visual yang objektif dan akurat, teknologi ini membuka peluang bagi pengembangan sistem pemantauan nutrisi tanaman secara *real-time*, yang sangat relevan untuk meningkatkan efisiensi dan produktivitas di sektor pertanian modern.

2.2.2 Pengolahan Citra (*Image Processing*)

Pengolahan citra digital telah menjadi teknologi yang sangat penting dalam bidang pertanian, terutama dalam deteksi kekurangan nutrisi. Dengan menggunakan teknik pengolahan citra, kita dapat menganalisis gambar daun

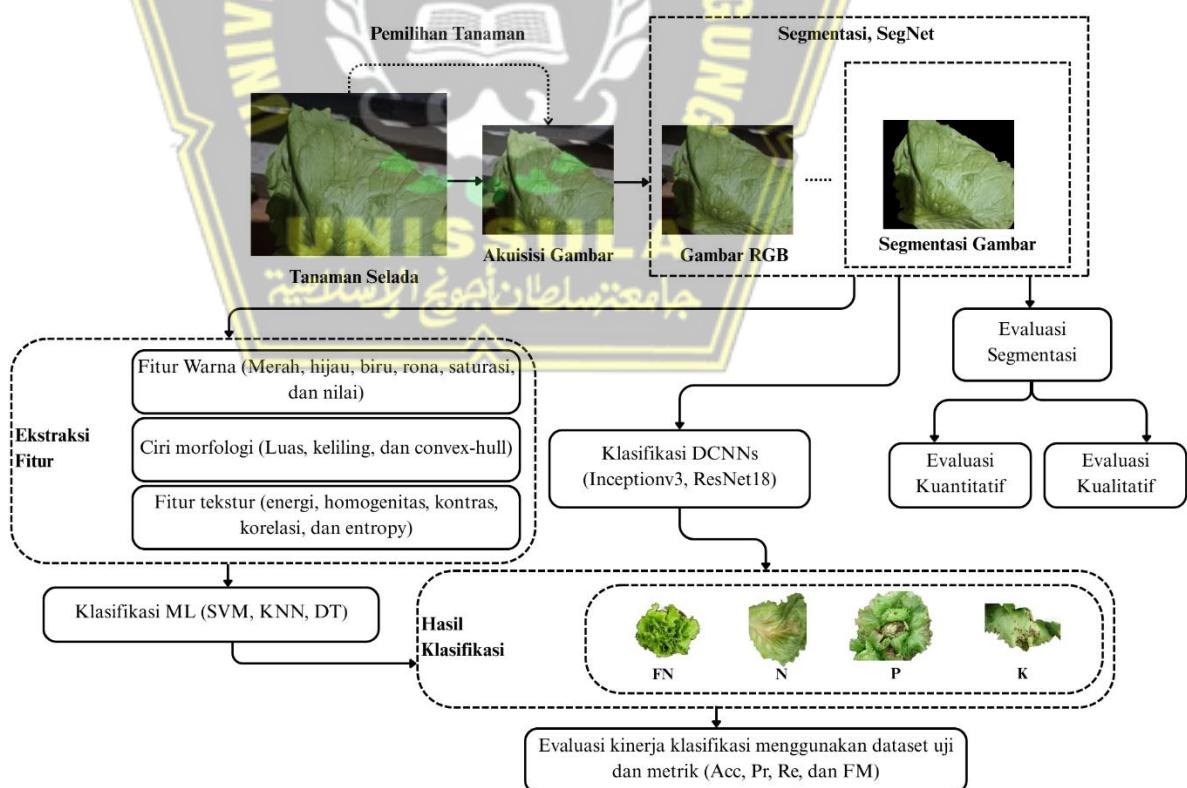
tanaman untuk mengidentifikasi gejala penyakit seperti perubahan warna, tekstur, atau bentuk. Proses ini melibatkan beberapa tahapan yang sangat penting, yaitu:



1. Pengambilan gambar daun menggunakan kamera digital atau perangkat lain.
2. Pembersihan dan penyesuaian gambar untuk memisahkan objek daun dari latar belakang.
3. Identifikasi ciri-ciri khas dari gambar yang relevan untuk deteksi kekurangan nutrisi.
4. Penentuan kategori penyakit berdasarkan ciri-ciri yang telah dideteksi.

Dengan demikian, pengolahan citra digital telah menjadi teknologi yang sangat penting dalam bidang pertanian, terutama dalam mendeteksi penyakit tanaman. Dengan menggunakan teknik ini, kita dapat meningkatkan efisiensi dan akurasi dalam mendeteksi penyakit, sehingga dapat mengurangi kerugian yang disebabkan oleh penyakit tanaman.

Untuk memperjelas proses dan tahapan pengolahan citra yang diterapkan dalam deteksi kekurangan nutrisi pada tanaman, berikut ditampilkan ilustrasi alur *image processing* dari penelitian yang relevan.



Gambar 2. 1 Illustrasi *Image Processing* (Taha dkk., 2022)

Pada gambar 2. 1 menunjukkan ilustrasi alur *image processing* untuk mendeteksi kekurangan nutrisi pada tanaman selada yang ditulis oleh Taha dkk (2022). Proses dimulai dari pemilihan tanaman dan pengambilan gambar, dilanjutkan dengan tahap segmentasi menggunakan model *Deep Learning* seperti *SegNet* untuk memisahkan area daun dari latar belakang gambar.

Setelah proses segmentasi, dilakukan ekstraksi fitur dari citra daun, yang mencakup:

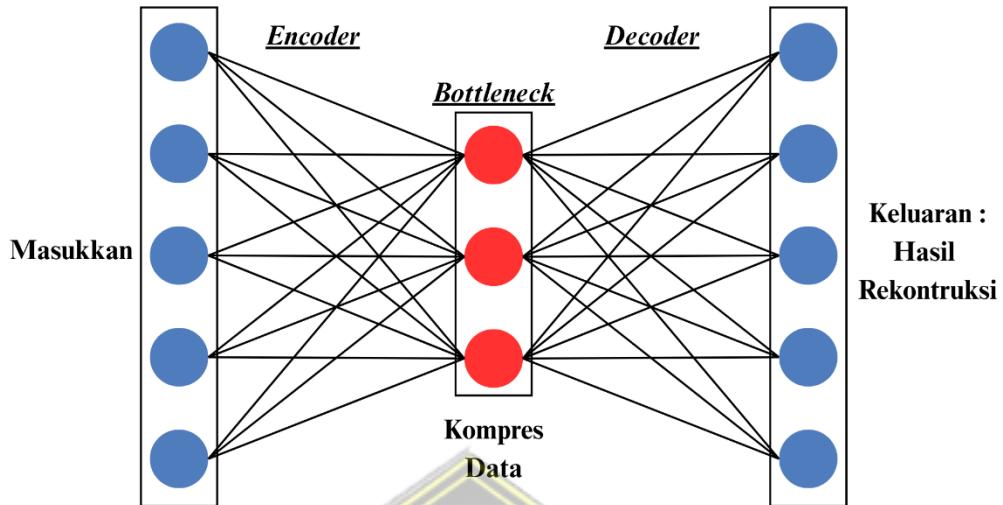
1. Fitur warna (seperti *red*, *green* *blue*, *hue*, *saturation*, dan *value*),
2. Fitur morfologis (seperti area, perimeter, *convex hull*), dan
3. Fitur tekstur (seperti *energy*, *homogeneity*, *contrast*, *correlation*, dan *entropy*).

Fitur-fitur tersebut kemudian digunakan sebagai *input* untuk model klasifikasi, baik *Machine Learning* tradisional seperti SVM, KNN, dan *Decision Tree*, maupun *Deep Learning Convolutional Neural Networks* (DCNNs) seperti *InceptionV3* dan *ResNet18*.

Hasil dari proses klasifikasi digunakan untuk menentukan jenis kekurangan nutrisi, seperti FN (*Full Nutrient*), (*Nitrogen Deficiency*), P (*Phosphorus Deficiency*), dan K (*Potassium Deficiency*). Terakhir, dilakukan evaluasi kinerja klasifikasi menggunakan metrik umum seperti akurasi, presisi, *Recall*, dan *F1-Score*.

2.2.3 *Autoencoder* dalam Ekstraksi Fitur Citra

Autoencoder adalah jenis jaringan saraf tiruan yang digunakan untuk belajar representasi data dalam bentuk dimensi yang lebih rendah secara tidak terawasi.



Gambar 2. 2 Arsitektur *Autoencoder*

Pada gambar 2. 2 menjelaskan tentang struktur dasar *Autoencoder* terdiri dari tiga komponen utama:

1. *Encoder*: Mengubah *input* menjadi representasi laten berdimensi lebih rendah.
2. *Bottleneck*: lapisan tersempit yang memaksa jaringan untuk belajar representasi yang paling penting dari data.
3. *Decoder*: merekonstruksi data asli dari representasi laten.

Dalam konteks deteksi penyakit tanaman, *Autoencoder* digunakan untuk mengekstraksi fitur penting dari citra daun secara otomatis tanpa label. Fitur-fitur ini kemudian digunakan sebagai *input* untuk tahap klasifikasi menggunakan MLP.(naz & Malik, 2023)

Rumus *Autoencoder* terdiri dari tiga bagian utama:

1. *Encoder*:

Mengubah *input* x menjadi representasi laten z melalui fungsi aktivasi:

$$z = f_{enc}(x) = o_1(W_e x + b_e) \quad (1)$$

Penjelasan rumus:

- a. x : *input* asli (citra daun)
- b. $f_{enc}(x)$: fungsi *Encoder* yang mengubah *input* x menjadi representasi laten z .

- c. W_e : bobot pada *layer encoder*.
- d. b_e : bias pada *layer encoder*.
- e. o_1 : fungsi aktivasi *encoder* menggunakan ReLU.
- f. z : representasi fitur laten hasil kompresi dari *input*

Fungsi rumus:

Rumus ini digunakan untuk menyandikan data *input* ke dalam bentuk vektor fitur yang lebih ringkas (laten). Tujuannya adalah menangkap informasi penting dari citra secara efisien, sambil membuang *noise* atau informasi yang kurang relevan.

2. Bottleneck:

Merupakan representasi padat hasil dari *Encoder*:

$$z \in \mathbb{R}^n \quad (2)$$

Penjelasan rumus:

- a. z : vektor representasi laten dari *input*.
- b. \mathbb{R}^n : ruang berdimensi n (*real number*), artinya vektor z memiliki n dimensi, lebih kecil dimensi *input*.

Fungsi rumus:

Bottleneck merupakan lapisan ter sempit pada *Autoencoder*. Di sini, representasi *input* dikompresi maksimal, berfungsi sebagai inti dari data. Informasi dari sinilah yang akan direkonstruksi oleh *Decoder* dan digunakan untuk ekstraksi fitur lebih lanjut.

3. *Decoder*:

$$x = f_{dec}(z) = o_2(W_d z + b_d) \quad (3)$$

Penjelasan rumus:

- a. z : representasi dari *Encoder*.
- b. $f_{dec}(z)$: fungsi *Decoder* untuk mengubah z kembali ke bentuk menyerupai x .
- c. W_d : bobot pada *Encoder*.
- d. b_d : bias pada *Decoder*.
- e. o_2 : fungsi aktivasi pada *Encoder*, sering menggunakan sigmoid atau ReLU.

- f. x : *output* rekonstruksi, yaitu hasil akhir dari *Decoder* yang mencoba merekonstruksi kembali *input* asli.

Fungsi rumus:

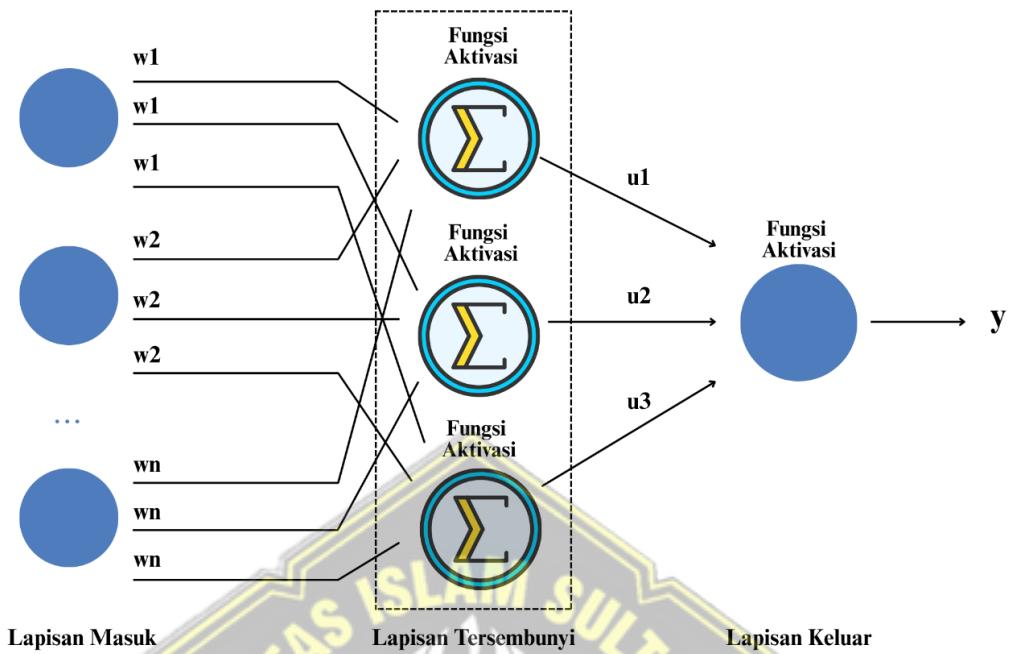
Decoder digunakan untuk membangun kembali *input* dari representasi laten. Jika hasil rekonstruksi sangat mirip dengan *input*, itu menunjukkan bahwa fitur penting telah berhasil ditangkap oleh *Encoder*.

2.2.4 Ekstraksi Fitur dengan *Autoencoder*

Autoencoder tidak mengekstraksi fitur secara eksplisit seperti warna, bentuk, atau tekstur secara langsung seperti metode konvensional pada pengolahan citra. Namun, *Autoencoder* mampu membentuk representasi numerik dalam ruang laten yang secara tidak langsung menangkap pola-pola penting dan struktur mendalam dari gambar. Representasi ini menyimpan informasi penting yang dibutuhkan oleh sistem untuk memahami perbedaan antar kelas gambar, meskipun tidak dalam bentuk fitur yang secara kasat mata dapat diidentifikasi oleh manusia.

2.2.5 *Multilayer Perceptron (MLP)*

Multilayer Perceptron (MLP) adalah jenis jaringan saraf tiruan yang terdiri dari tiga atau lebih lapisan, seperti *input layer*, *hidden layer*, dan *output layer*. Setiap *neuron* dalam MLP menggunakan fungsi aktivitas *non-linear*, memungkinkan jaringan untuk memodelkan hubungan kompleks antara *input* dan *output*. Dalam deteksi penyakit tanaman, MLP digunakan untuk mengklasifikasikan citra daun berdasarkan fitur yang diekstraksi oleh *Autoencoder*. (Kusuma dkk., 2023)



Gambar 2. 3 Arsitektur Jaringan Saraf Tiruan MLP (Kusuma dkk., 2023)

Jaringan pada gambar 2. 3 tersusun atas tiga komponen utama, yaitu:

1. *Input Layer*
 - a. Fungsi: menjadi titik masuk data ke dalam jaringan saraf.
 - b. Komponen: tiap bulatan biru mewakili satu *neuron (node) input* yang menerima fitur dari data mentah. Misalnya, jika kita memproses citra daun, maka tiap *neuron* bisa mewakili nilai piksel, warna, tekstur, atau fitur citra lainnya.
2. *Hidden Layer*
 - a. Fungsi: menangkap hubungan kompleks antara *input* dan *output* melalui proses pembobotan dan fungsi aktivasi.
 - b. Komponen:
 - 1) Setiap neuron menerima *input* dari seluruh *neuron* di *layer* sebelumnya (*input layer*).
 - 2) Hasil *input layer* tersebut kemudian dilewatkan ke fungsi aktivasi untuk menghasilkan *output non-linear*.

3. Output Layer

- a. Fungsi: memberikan hasil akhir dari jaringan.
- b. Komponen:
 - 1) *Neuron* di *output layer* menerima hasil dari seluruh *neuron* di *hidden layer*.
 - 2) Prosesnya mirip seperti di *hidden layer*, yaitu penjumlahan bobot dan aktivasi.
 - 3) Hasil akhirnya adalah y , yang bisa berupa klasifikasi atau nilai prediksi.

Dalam arsitektur aringan *Multilayer Perceptron* (MLP), fungsi aktivasi (*activation function*) memegang peranan penting dalam memungkinkan jaringan untuk belajar dan memodelkan hubungan *non-linear* antar data. Fungsi aktivasi disisipkan disetiap *neuron* pada *hidden layer* maupun *output layer* untuk menentukan apakah suatu *neuron* akan diaktifkan atau tidak berdasarkan nilai *input* yang diterimanya.

1. ReLU (*Rectified Linear Unit*)

Pada penelitian ini, digunakan fungsi aktivasi ReLU dibagian *input layer* dan *hidden layer*. Fungsi ini bekerja dengan cara meneruskan nilai positif sebagaimana adanya dan mengubah semua nilai negatif menjadi nol. Secara matematis dituliskan sebagai:

$$f(x) = \max(0, x) \quad (4)$$

Penjelasan Rumus:

- a. $f(x)$: nilai keluaran dari fungsi aktivasi ReLU terhadap *input* x .
- b. x : nilai *input* dari *neuron* sebelumnya (hasil perhitungan bobot dan bias).
- c. $\max(0, x)$: fungsi maksimum yang akan memilih nilai terbesar antara 0 dan x .

ReLU digunakan karena:

- a. Menghindari masalah *vanishing gradient*
- b. Mempercepat konvergensi pelatihan model
- c. Sederhana dan efisien secara komputasi

2. *Softmax*

Pada *output layer*, digunakan fungsi aktivasi *Softmax*, yang berfungsi untuk mengubah *output* jaringan menjadi distribusi probabilitas terhadap kelas-kelas target. Fungsi ini memastikan bahwa total *output* bernilai 1 dan masing-masing *output* merepresentasikan kemungkinan *input* termasuk ke dalam masing-masing kelas. Rumus fungsi *Softmax* adalah:

$$y_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (5)$$

Penjelasan rumus:

- a. y_i : nilai *output* (probabilitas) kelas ke- i setelah fungsi *Softmax* diterapkan.
- b. z_i : skor dari *neuron* ke- i *output layer*.
- c. e : bilangan eksponensial (sekitar 2.718).
- d. K : jumlah total kelas dalam masalah klasifikasi.
- e. $\sum_{j=1}^K e^{z_j}$: penjumlahan seluruh nilai eksponensial dari setiap skor kelas sebagai normalisasi.

Softmax digunakan karena:

Penggunaan fungsi *Softmax* sangat cocok untuk kelas *multiclass classification*, seperti dalam penelitian ini yang mengklasifikasikan daun ke dalam empat kategori: Nitrogen, *Phosphor*, Kalium, dan *Healty*.

BAB III

METODE PENELITIAN

3.1 Metode Penelitian

3.1.1 Integrasi *Autoencoder* dan MLP

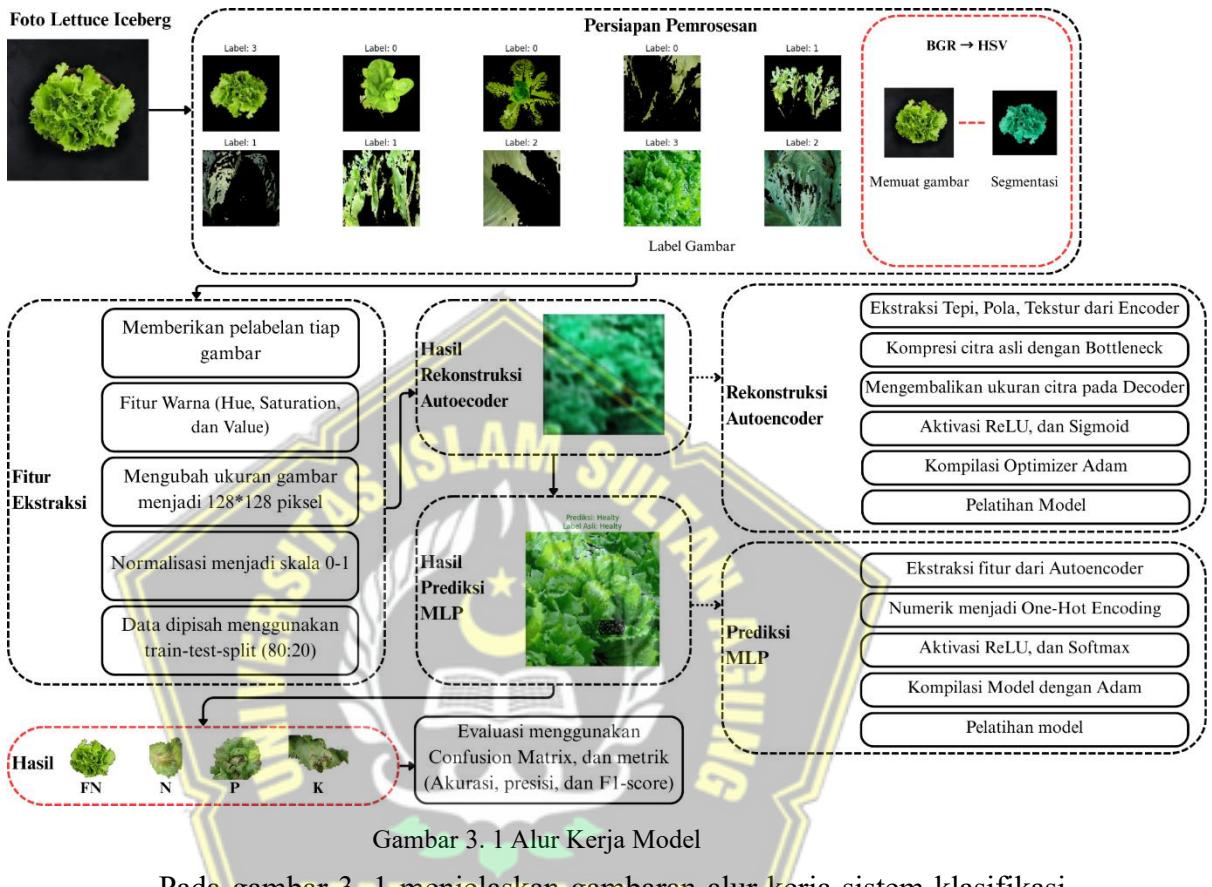
Pada tahap ini dilakukan integrasi antara model *Autoencoder* dan *Multilayer Perceptron* (MLP) sebagai solusi dalam mendeteksi kekurangan nutrisi tanaman berdasarkan citra daun. Pendekatan ini memungkinkan sistem melakukan ekstraksi fitur penting secara otomatis dari citra *input* dan mengklasifikannya ke dalam kategori defisiensi nutrisi.

Pendekatan gabungan antara *Autoencoder* dan *Multilayer Perceptron* (MLP) merupakan solusi yang menjanjikan untuk menggabungkan mekanisme ekstraksi fitur dan kemampuan klasifikasi dalam satu sistem yang ringan dan efektif.

Peratama, *Autoencoder* bertugas mereduksi dimensi dan mengekstrak fitur penting dari citra daun dengan cara mengompresi informasi visual ke dalam representasi laten. Salah satu contoh implementasi sukses adalah model *PDSE-Lite* oleh Bedi, dkk (2023), yang memanfaatkan *Convolutional Autoencoder* untuk mendeteksi tingkat keparahan penyakit pada tanaman secara efisien(Bedi dkk., 2023)

Selain itu, dikembangkan juga sebuah pendekatan berbasis *Deep Learning* yang memanfaatkan jaringan *Autoencoder* konvolusional untuk mendeteksi penyakit pada tanaman. Penelitian ini secara khusus menyoroti kemampuan *Autoencoder* dalam melakukan ekstraksi fitur penting dari citra daun tanaman secara otomatis, tanpa perlu proses segmentasi atau pemrosesan manual yang kompleks. Proses ekstraksi ini dilakukan melalui kompresi informasi visual ke dalam representasi yang lebih padat dan bermakna menggunakan jaringan *Encoder*, yang kemudian diolah untuk menghasilkan informasi yang dapat dikenali oleh sistem. (Natarajan dkk., 2024)

Setelah fitur diekstraksi melalui arsitektur *Autoencoder*, langkah selanjutnya dalam penelitian ini adalah melakukan deteksi terhadap jenis penyakit yang teridentifikasi dalam citra, seperti *Workflow* dibawah ini:



Gambar 3. 1 Alur Kerja Model

Pada gambar 3. 1 menjelaskan gambaran alur kerja sistem klasifikasi defisiensi nutrisi pada tanaman *Lettuce Iceberg* berbasis pengolahan citra digital yang terdiri dari beberapa tahap utama, yaitu: pengambilan data (*input* citra), pemrosesan awal (*preprocessing*), ekstraksi fitur dengan *Autoencoder*, klasifikasi dengan MLP, dan evaluasi model.

1. Pengambilan Data (Foto *Lettuce Iceberg*)

Proses dimulai dengan pengumpulan citra tanaman *lettuce iceberg* yang diambil dari dataset. Gambar-gambar ini menjadi *input* awal sistem.

2. *Preprocessing*

Tahap ini merupakan proses penting dalam mempersiapkan data gambar sebagai *input* model:

a. Pemberian Label

Setiap gambar diberi label numerik yang merepresentasikan kondisi nutrisi daun, seperti 0 untuk defisiensi Nitrogen, 1 untuk *Phosphor*, 2 untuk Kalium, dan 3 untuk daun sehat

b. Segmentasi Warna (HSV):

Gambar awal dikonveri dari format RGB (*Red, Green, Blue*) ke ruang warna HSV (*Hue, Saturation, Value*). Model HSV digunakan karena dapat memisahkan informasi warna (*Hue*) dari kecerahan (*Value*), sehingga mempermudah dalam melakukan filter objek berdasarkan warna tertentu. Dengan menentapkan *threshold* pada *channel hue*, bagian daun dapat disegmentasi dari latar belakang yang tidak relevan.

c. Pengubahan Ukuran Gambar (*Resize*)

Gambar hasil segmentasi kemudian diubah ukurannya menjadi 128*128 piksel agar seragam yang dapat mengurangi kompleksitas komputasi dan menyamaratakan ukuran dalam *input*.

d. Normalisasi Piksel

Gambar yang telah diresized kemudian dinormalisasi ke rentang [0, 1] dengan membagi setiap nilai piksel RGB dengan 255 yang bertujuan untuk mempercepat proses pelatihan dan menghindari dominasi nilai piksel besar dalam perhitungan bobot jaringan.

e. Pembagian Dataset

Dataset gambar dan label kemudian dibagi menjadi data latih dan data uji menggunakan metode *train-test split* dengan

rasio 80:20, yang di mana 80% data digunakan untuk pelatihan model, dan 20% untuk pengujian.

3. Ekstraksi fitur dengan *Autoencoder*

Pada tahap ini, fitur-fitur penting dari gambar diekstrak, seperti pada fitur warna (merah, hijau, dan biru) digunakan karena perubahan warna daun merupakan indikator utama dari defisiensi nutrisi. Gambar hasil *preprocessing* dioleh menggunakan arsitektur *Autoencoder*, yang bertugas mengekstrak dan merepresentasikan fitur penting dalam bentuk *encoded* vektor berdimensi lebih kecil. *Autoencoder* juga digunakan untuk melakukan rekonstruksi citra untuk memastikan bahwa fitur yang diambil cukup merepresentasikan informasi penting dari gambar.

4. Klasifikasi dengan MLP

Tahap seetelah *Autoencoder* Adalah melakukan klasifikasi menggunakan model MLP. Berikut Adalah penjelasannya:

a. Pembuatan *Encoder* dari *Autoencoder*

Encoder dari *Autoencoder* yang sudah dilatih lalu dipisahkan sehingga hanya digunakan untuk menghasilkan representasi terkompresi atau *encoded* vektor dari setiap gambar daun.

b. Ekstraksi dari Flatten Fitur

Proses ekstraksi dilakukan dari *encoder* dan diratakan hasilnya agar dapat diolah model MLP yang menerima *input* berupa vektor satu dimensi.

c. *One-Hot Encoding* Label

Label yang berawal dari bentuk angka diubah menjadi format *one-hot encoding*, untuk merepresentasikan setiap kelas sebagai vektor biner yang sesuai dengan jumlah kelas yang ada.

d. Aktivasi ReLU dan *Softmax*

Pada proses ini, MLP didefinisikan menjadi tiga lapisan utama, yaitu lapisan *dense* pertama yang berjumlah 256 *neuron*

dengan aktivasi ReLU untuk memproses semua fitur dengan dropout sebanyak 30% untuk mencegah *overfitting*, lapisan kedua berjumlah 126 *neuron* dengan aktivasi ReLU dan *dropout* 30%, dan lapisan ketiga dengan 4 *neuron* yang menggunakan aktivasi *Softmax* untuk menghasilkan probabilitas prediksi bagi masing-masing kelas.

e. Kompilasi Model

Model ini dikompilasi dengan menggunakan *optimizer Adam*, fungsi *loss categorical crossentropy* yang sesuai untuk klasifikasi *multiclass* dengan label *one-hot* dan metrik evaluasi akurasi.

f. Pelatihan Model

Proses pelatihan dilakukan sebanyak 25 *epoch*, dengan menggunakan data pengujian digunakan untuk melihat performa model di setiap *epoch*.

5. Evaluasi Model

Tahap akhir adalah evaluasi performa dari model klasifikasi. Dua jenis evaluasi yang digunakan adalah:

- a. *Confusion Matrix*, yang menampilkan penyebaran prediksi benar dan salah antar kelas. Matriks ini memberikan gambaran detail tentang berapa banyak gambar tiap kelas yang diklasifikasikan dengan benar atau salah.
- b. Grafik Evaluasi Metrik (Akurasi, Presisi, *Recall*, dan *F1-Score*) yang menunjukkan performa numerik model:
 - 1) Akurasi sebagai mengukur seberapa banyak prediksi yang benar dari keseluruhan data.
 - 2) Presisi sebagai menggambarkan seberapa akurat model dalam memprediksi suatu kelas.
 - 3) *Recall* sebagai menggambarkan seberapa baik model dalam menemukan semua contoh dari satu kelas tertentu.
 - 4) *F1-Score* sebagai perhitungan antara presisi dan *Recall*.

3.1.2 Evaluasi Model

Evaluasi terhadap performa model dilakukan untuk mengukur seberapa baik sistem dalam mengklasifikasikan kondisi daun tanaman *lettuce iceberg* berdasarkan jenis kekurangan nutrisinya. Dalam penelitian ini, proses evaluasi dilakukan menggunakan pendekatan, yaitu metrik akurasi, presisi, dan *F1-Score*.

Akurasi digunakan untuk mengukur persentase prediksi yang benar dari keseluruhan data yang diuji, yang dihitung dengan rumus sebagai berikut:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

Di mana TP (*True Positive*) adalah jumlah prediksi positif yang benar, TN (*True Negative*) adalah jumlah prediksi negatif yang benar, FP (*False Positive*) adalah jumlah prediksi positif yang salah, dan FN (*False Negative*) adalah jumlah prediksi yang salah.

Selain akurasi, *F1-Score* juga digunakan untuk memberikan penilaian performa model yang lebih seimbang, khususnya pada dataset yang memiliki distribusi tidak merata antar kelas. *F1-Score* mempertimbangkan keseimbangan antara *precision* dan *Recall*, dan dihitung menggunakan rumus berikut:

$$F1-score = \frac{2 \cdot (Precision \cdot Recall)}{Precision + Recall} \quad (7)$$

Dengan:

1.

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

2.

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

Evaluasi dilakukan menggunakan data validasi yang tidak dilibatkan dalam proses pelatihan, sehingga hasil evaluasi ini memberikan Gambaran umum mengenai kemampuan generalisasi model terhadap data baru. Metri-metrik tersebut

diimplementasikan menggunakan *library scikit-learn* untuk mendapatkan nilai akurasi dan *F1-Score* dari hasil prediksi model terhadap data uji.

Selain metrik akurasi, presisi, dan *F1-Score*, evaluasi performa model juga dilengkapi dengan *Confusion Matrix* untuk memberikan gambaran lebih detail mengenai hasil prediksi model terhadap masing-masing kelas.

Confusion Matrix Adalah tabel yang menunjukkan jumlah prediksi benar dan salah yang dilakukan oleh model untuk setiap kelas. Baris pada tabel mewakili kelas sebenarnya (*Actual*), sedangkan kolom mewakili kelas prediksi (*Predicted*). Dengan menggunakan *Confusion Matrix*, kita dapat mengidentifikasi kelas mana yang paling sering diprediksi salah, serta mengevaluasi sejauh mana model membedakan antar kelas.

Tabel 3. 1 Contoh Tabel *Confusion Matrix*

Actual\Predicted	N	P	K	H
N	9	1	3	0
P	0	5	5	0
K	0	0	14	0
H	0	0	0	7

Tabel 3. 1 adalah contoh tabel *Confusion Matrix* yang dihasilkan menggunakan fungsi *confusion_matrix* dari *library scikit-learn*, yang menerima label *actual* dan label prediksi sebagai *input*.

Dari tabel *Confusion Matrix* pada 3.1, dapat dilihat bahwa model berhasil mengklasifikasikan sebagian besar sampel dengan benar, terutama pada kelas Kalium (K) dan *Healty* (H) yang menunjukkan nilai prediksi sempurna sebanyak 14 dan 7 kali secara berurutan. Namun, terdapat beberapa kesalahan klasifikasi pada kelas Nitrogen (N) dan *Phosphor* (P), di mana model memprediksi 1 sampel kelas Nitrogen sebagai *Phosphor*, serta 5 sampel kelas *Phosphor* salah diklasifikasikan sebagai Kalium. Dari hasil ini menunjukkan bahwa meskipun performa model cukup baik secara umum, masih ada tantangan dalam membedakan fitur visual yang serupa antara kelas tertentu.

3.2 Analisa kebutuhan

Pada tahap ini penulis menganalisa apa saja kebutuhan selama pembangunan sistem, seperti Bahasa pemrograman, *software*, *library*, dan *Framework* yang digunakan. Berikut adalah apa saja yang digunakan dalam pembangunan sistem:

1. Bahasa Pemrograman

a. *Python*

Bahasa pemrograman utama yang digunakan dalam pengembangan sistem, karena dukungan ekosistem *library* yang sangat kuat dalam bidang *Machine Learning*, pengolahan citra, dan pengembangan web berbasis data.

2. *Software*

a. *Google colab*

Digunakan sebagai *platform cloud* untuk menulis dan menjalankan kode *python*, terutama dalam pelatihan model *Deep Learning* dan *Preprocessing* gambar.

b. *Google Drive*

Berfungsi sebagai penyimpanan dataset citra dan model hasil pelatihan yang diakses langsung dari Colab.

c. *Draw.io*

digunakan untuk membuat diagram alur sistem, arsitektur jaringan, serta visualisasi perancangan aplikasi.

d. *Visual Studio Code*

Digunakan untuk menulis dan menyusun *File* program *Streamlit* yang kemudian dijalankan secara lokal.

e. *Figma*

Dimanfaatkan untuk membuat desain tampilan antarmuka pengguna aplikasi sebelum diimplementasikan di *Streamlit*.

f. *Canva*

Digunakan untuk membuat sebuah desain arsitektur dan alur sistem model.

g. *Streamlit Cloud*

Digunakan untuk mendeploy sistem yang sudah dibuat untuk diakses secara umum.

3. *Library dan Framework*

a. *TensorFlow & Keras*

TensorFlow adalah *platform open-source* yang popular untuk mengembangkan *Machine Learning* dan *Deep Learning*. Dalam penelitian ini, *TensorFlow* digunakan sebagai kerangka utama untuk membangun arsitektur *Autoencoder* dan *Multilayer Perceptron* (MLP). *Keras*, yang merupakan API dari *TensorFlow*, memberikan antarmuka yang lebih sederhana dan intuitif dalam membangun dan melatih model.

b. *NumPy*

NumPy merupakan *library fundamental* untuk komputasi numerik dalam *Python*. Dalam konteks penelitian ini, *NumPy* digunakan untuk melakukan sebagai operasi terhadap *array* atau *tensor*, seperti mengubah bentuk *array*, menggabungkan data, serta menyusun dan memanipulasi fitur yang diekstraksi dari *Autoencoder*.

c. *OpenCV*

OpenCV digunakan untuk membaca dan memproses citra digital. Dalam penelitian ini, *OpenCV* dimanfaatkan untuk membaca dataset citra daun *lettuce*, mengubah ukuran gambar, dan melakukan transformasi gambar dasar seperti konversi warna.

d. *Matplotlib*

Matplotlib adalah *library visualisasi* yang banyak digunakan dalam *Python*. *Library* ini digunakan untuk menampilkan hasil visualisasi selama proses *Preprocessing* dan evaluasi, seperti menampilkan *batch* gambar hasil *augmentasi*, dan hasil prediksi model klasifikasi.

e. *Scikit-learn (sklearn)*

Scikit-learn adalah *library* yang menyediakan berbagai *tools* untuk *Machine Learning* dan analisis data. Dalam penelitian ini, *scikit-learn* digunakan untuk evaluasi model klasifikasi menggunakan metrik seperti akurasi, presisi, *Recall*, *F1-Score*, *Confusion Matrix*, serta *classification report*.

f. *PIL (Python Imaging Library)*

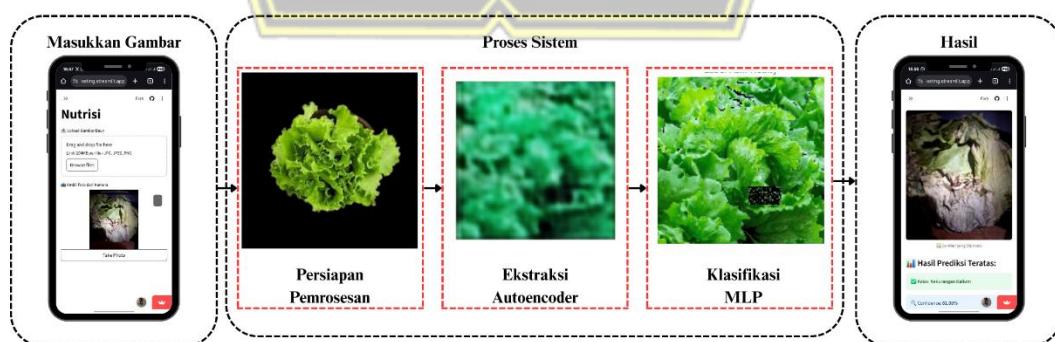
PIL digunakan untuk memproses gambar dalam antarmuka pengguna. Dalam konteks *Streamlit* yang dibangun, *PIL* digunakan untuk membuka *File* gambar yang diunggah pengguna agar bisa ditampilkan dan diproses lebih lanjut oleh model klasifikasi.

g. *Streamlit*

Digunakan untuk membangun antarmuka pengguna berbasis web agar pengguna dapat mengunggah gambar dan melihat hasil deteksi secara langsung.

3.3 Penggunaan Sistem

Pada tahap ini dilakukan Analisa untuk menentukan alur kerja penggunaan sistem restorasi citra yang akan dilakukan oleh *user* dalam bentuk *Flowchart*. Alur kerja sistem dapat dilihat dibawah ini:



Gambar 3. 2 Alur Sistem

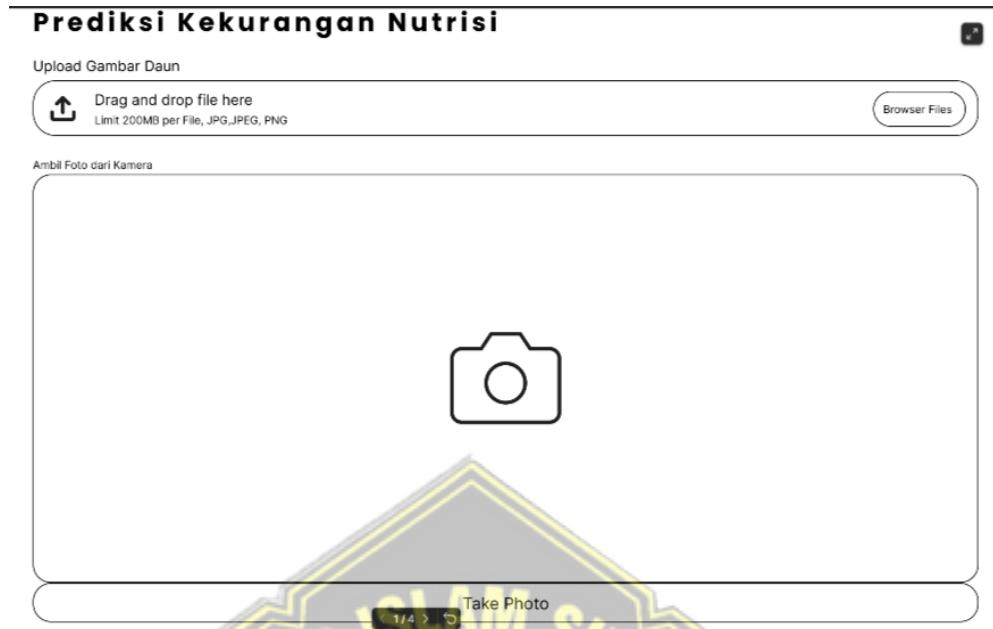
Pada gambar 3. 2 adalah alur sistem deteksi kekurangan nutrisi saat digunakan oleh *user*. Pada gambar 3. 2 memiliki beberapa tahapan, sebagai berikut:

1. *User* membuka aplikasi kemudian akan muncul tampilan awal *website*.
2. *User* mengunggah gambar bisa dengan tiga cara, yaitu menggunakan *Drag and drop File here*, *Browser File* atau *Take Photo* untuk memasukkan citra yang ingin dideteksi.
3. Selanjutnya sistem akan memproses citra daun tersebut dengan melihat ciri-ciri warna, bercak, dan sebagainya untuk menentukan daun memiliki kekurangan nutrisi atau tidak.
4. Terakhir, *user* akan mendapatkan *input* dari sistem berupa citra daun yang sudah dideteksi dari citra daun yang *user* masukkan ke sistem.

3.4 Perancangan *User Interface*

3.4.1 Halaman Awal Sistem

Gambar berikut merupakan *wireframe* atau rancangan awal antarmuka pengguna (*user interface*) dari sistem deteksi kekurangan nutrisi daun *lettuce iceberg*. Desain ini menggambarkan tampilan halaman awal yang dirancang untuk memudahkan pengguna dalam mengunggah citra daun sebelum dilakukan proses klasifikasi lebih lanjut oleh sistem.



Gambar 3. 3 *User Interface* desktop sebelum *upload* gambar



Gambar 3. 4 *User Interface* mobile sebelum *upload* gambar

Pada gambar 3. 3 dan 3. 4 menampilkan rancangan antarmuka pengguna (*user interface*) pada halaman awal sistem deteksi kekurangan nutrisi. Halaman ini merupakan titik awal interaksi pengguna dengan sistem, di mana pengguna diarahkan untuk mengunggah citra daun yang ingin diperiksa tingkat kecukupan nutrisinya. Desain antar muka dibuat sederhana dan intuitif agar dapat digunakan dengan mudah oleh siapapun.

Pada bagian tengah halaman, terhadap area unggah (*upload area*) dengan instruksi "Drag and drop File here" bagi pengguna yang ingin langsung menyeret gambar ke area tersebut atau tombol "Browser Files" yang dapat diklik untuk memilih *File* gambar secara manual dari perangkat pengguna.

3.4.2 Tampilan Hasil Sistem

Gambar berikut merupakan wireframe atau rancangan awal antarmuka pengguna (*user interface*) dari sistem deteksi kekurangan nutrisi daun *lettuce iceberg*. Desain ini menggambarkan tampilan halaman awal yang dirancang apabila gambar berhasil diupload.

Hasil Prediksi Teratas:

Kelas: **Sehat**

Confidence: 90%

Persentase Tiap Kelas:

Kekurangan Nitrogen: 5%

Kekurangan Fosfor: 5%

Kekurangan Kalium: 5%

Sehat: 85%

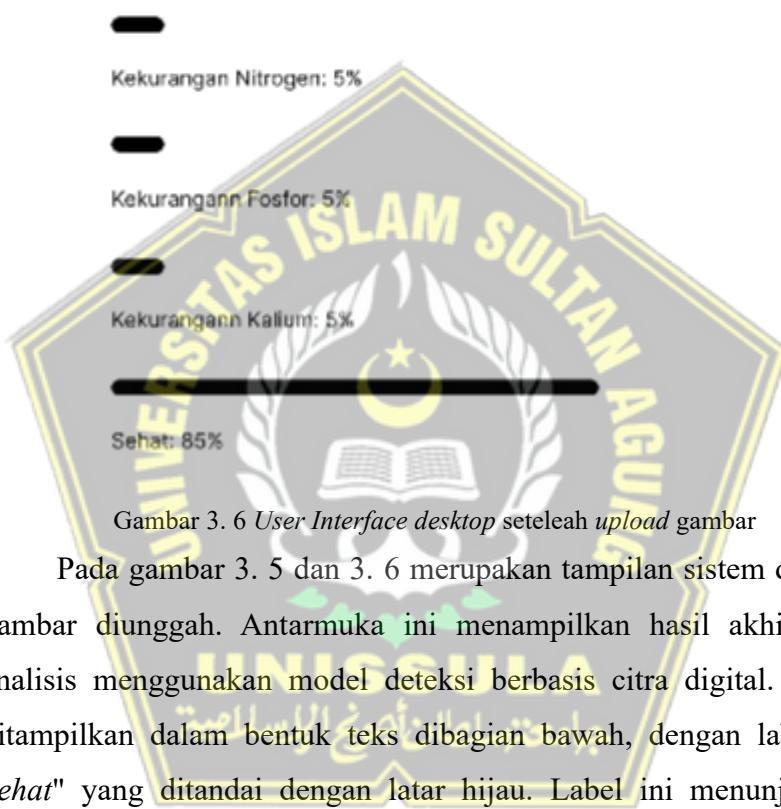
Gambar 3. 5 *User Interface* desktop setelah *upload* gambar

Hasil Prediksi Teratas:

Kelas: **Sehat**

Confidence: 90%

Persentase Tiap Kelas:



Gambar 3. 6 *User Interface desktop* setelah *upload* gambar

Pada gambar 3. 5 dan 3. 6 merupakan tampilan sistem deteksi setelah gambar diunggah. Antarmuka ini menampilkan hasil akhir dari proses analisis menggunakan model deteksi berbasis citra digital. Hasil deteksi ditampilkan dalam bentuk teks dibagian bawah, dengan label "Kelas: **Sehat**" yang ditandai dengan latar hijau. Label ini menunjukkan bahwa sistem mengenali gambar daun yang diunggah sebagai daun dengan kondisi sehat.

Antar muka ini dirancang dengan pendekatan minimalis dan fokus pada fungsionalitas, untuk memudahkan pengguna dalam memahami hasil tanpa kerumitan teknis.

BAB IV

HASIL PENELITIAN DAN IMPLEMENTASI SISTEM

4.1 Hasil Pengumpulan Data

Berikut ini menunjukkan tabel sampel citra daun tanaman lettuce iceberg yang digunakan dalam penelitian, lengkap dengan deskripsi masing-masing kategori kondisi nutrisi. Setiap kategori mewakili satu jenis tanaman, mulai dari tanaman sehat hingga tanaman yang mengalami defisiensi nutrisi seperti Nitrogen (N), Fosfor (P), dan Kalium (K). dataset citra daun ini diperoleh dari Kaggle, yang menyediakan data visual tanaman lettuce iceberg dengan berbagai tingkat Kesehatan.

Tabel 4. 1 Tabel Sampel Dataset

Citra	Deskripsi
	Tanaman yang memiliki nutrisi yang terpenuhi, terlihat dari warna daun yang hijau merata dan pertumbuhan yang baik.
	Tanaman yang mengalami defisiensi Nitrogen, ditandai dengan warna daun yang menguning secara menyeluruh, terutama pada tua.
	Tanaman yang mengalami kekurangan fosfor, terlihat dari warna daun yang berubah menjadi ungu atau kemerahan pada bagian bawah.
	Tanaman yang mengalami defisiensi Kalium, ditunjukkan oleh tepi daun yang mengering atau berwarna coklat serta bentuk daun yang tidak normal.

Pada tabel 4. 1 proses pengumpulan data dalam penelitian ini dilakukan dengan mengumpulkan citra daun tanaman *lettuce* yang diklasifikasikan kedalam empat kategori, yaitu daun sehat (*Healty*), kekurangan Nitrogen (*Nitrogen Deficiency*), kekurangan fosfor (*Phosphorus Deficiency*), dan kekurangan Kalium (*Potassium Deficiency*). Dataset diperoleh dari sumber digital dan secara langsung, kemudian disusun dalam direktori terpisah berdasarkan masing-masing kelas untuk memudahkan proses pelabelan dan pelatihan model.

Adapun deskripsi dari masing-masing kategori adalah sebagai berikut:

1. Daun Sehat (*Healty*)

Ditandai dengan warna hijau segar merata di seluruh permukaan daun, tidak terdapat bercak atau perubahan warna, serta memiliki tekstur yang normal tanpa kerusakan fisik.

2. Kekurangan Nitrogen (*Nitrogen Deficiency*)

Ditandai dengan warna yang menguning, terutama pada daun bagian bawah, pertumbuhan daun melambat, dan ukuran daun cenderung lebih kecil dari normal.

3. Kekurangan Fosfor (*Phosphorus Deficiency*)

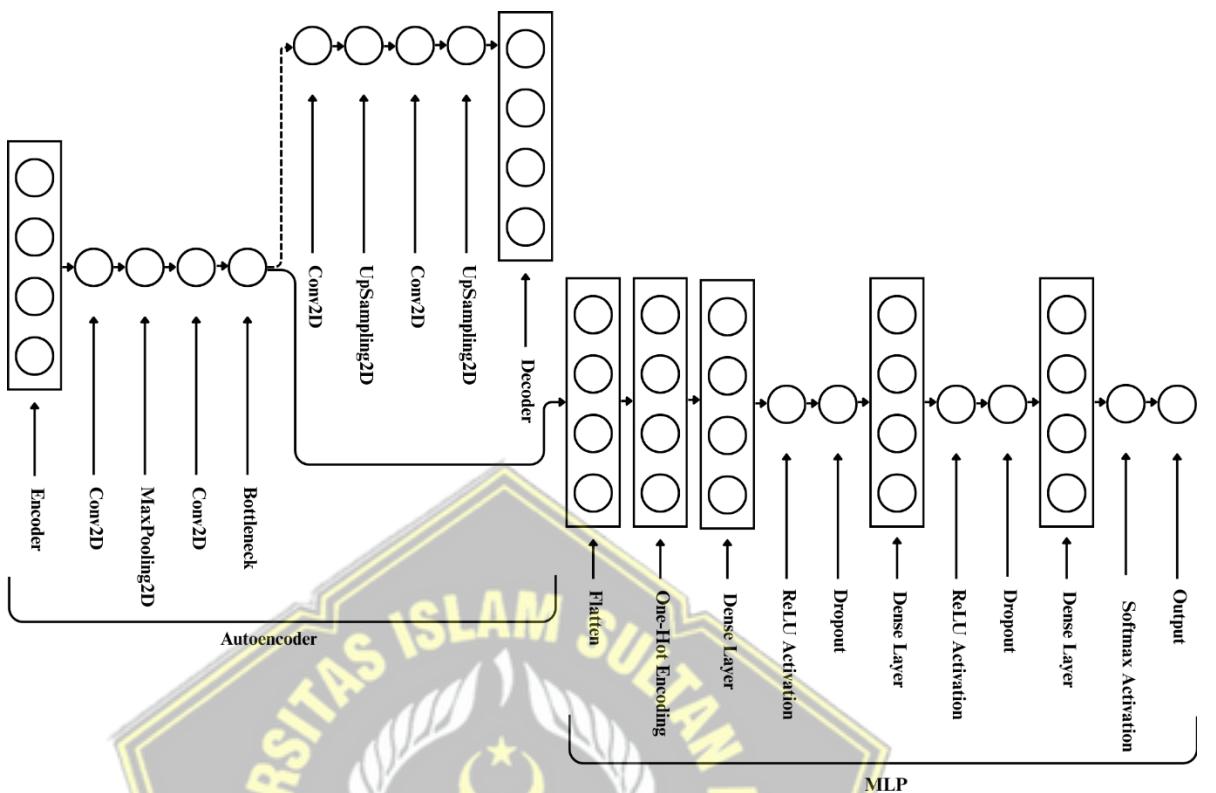
Ditandai dengan warna daun yang hijau gelap, biasanya terlihat pada daun tua, serta pertumbuhan akar yang lambat.

4. Kekurangan Kalium (*Potassium Deficiency*)

Ditandai ujung dan tepi daun yang mengering atau berwarna coklat, serta daun menggulung atau terlihat keriting.

4.2 Arsitektur Model Autoencoder dan Multilayer Perceptron

Arsitektur sistem yang dibuat dengan menggabungkan dua model, yaitu *Autoencoder* dan *Multilayer Perceptron* (MLP) untuk mendeteksi kekurangan nutrisi pada daun *lettuce iceberg*. *Autoencoder* berfungsi sebagai ekstraktor fitur laten melalui proses *encoding*, sedangkan MLP digunakan sebagai klasifikator untuk memprediksi kondisi nutrisi berdasarkan fitur yang telah diekstraksi.



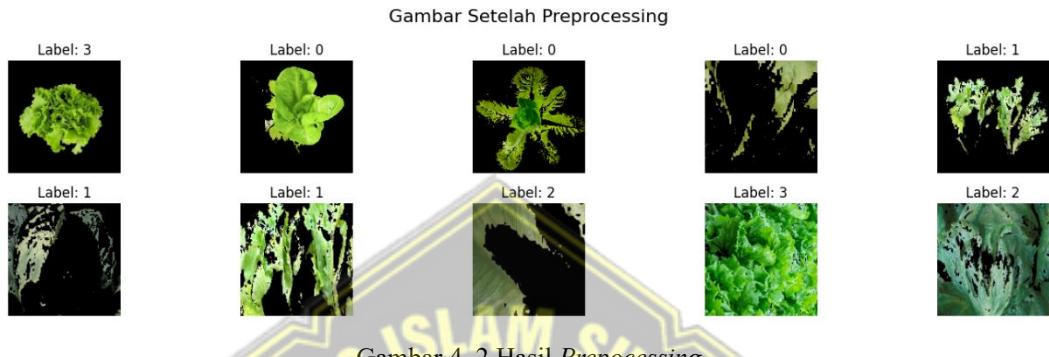
Gambar 4. 1 Arsitektur *Autoencoder* dan *MLP*

Pada gambar 4. 1 menampilkan arsitektur dari model *Autoencoder* dan *Multilayer Perceptron* (MLP), proses dimulai dari lapisan *encoder* yang melakukan ekstraksi fitur menggunakan *Conv2D* dan *MaxPooling2D*, kemudian hasil kompresi pada *bottleneck* *flatten* untuk menghasilkan representasi fitur yang lebih sederhana. *Decoder* berfungsi untuk proses rekonstruksi sebagai bahan evaluasi dari kualitas ekstraksi fitur, namun tidak digunakan dalam tahap klasifikasi. Representasi fitur selanjutnya diproses oleh MLP yang terdiri dari beberapa *Dense Layer* dengan aktivasi ReLU dan *Dropout* untuk mencegah *overfitting*, serta lapisan *Softmax* pada *output* untuk menghasilkan daun ke dalam empat kategori nutrisi.

4.3 Implementasi Sistem Deteksi

4.3.1 Preprocessing Dataset

Gambar berikut menampilkan hasil proses *preprocessing* pada dataset citra daun *lettuce iceberg* yang melewati *labeling*, segmentasi, *resize*, normalisasi, dan *train-test split*.



Gambar 4.2 Hasil Preprocessing

Pada gambar 4. 2 menampilkan hasil proses *preprocessing*, diperoleh citra daun *lettuce iceberg* yang sudah melalui serangkaian tahapan pengolahan awal. Tahap pertama adalah pemberian label pada setiap citra sesuai kategori masing-masing, yaitu Nitrogen, *Phosphor*, Kalium, dan *Healty*. Sehingga setiap data sudah memiliki identitas kelas yang jelas. Selanjutnya dilakukan tahap segmentasi, di mana citra dikonversi dari warna BGR (*Blue, Green, Red*) menjadi HSV (*Hue, Saturation, Value*). Model HSV digunakan karena lebih efektif untuk memisahkan warna objek dari latar belakang, khususnya dalam mendekripsi warna hijau daun. Rentang nilai warna hijau ditentukan menggunakan *lower bound* dan *upper bound* dalam format HSV, sehingga sistem dapat membuat *mask* yang mempertahankan bagian daun dan menghilangkan seluruh area latar belakang.

Citra hasil segmentasi kemudian diubah menjadi ukuran 128*128 piksel agar seluruh data memiliki resolusi yang sama rata dan sesuai kebutuhan model arsitektur model. Setelah itu, setiap nilai piksel pada citra dinormalisasi dari skala 0-255 menjadi 0-1 dengan cara membagi nilai piksel dengan 255.0, yang bertujuan untuk mempercepat konvergensi saat pelatihan model serta menjaga stabilitas perhitungan.

Dataset yang telah diproses kemudian dibagi menjadi dua bagian menggunakan metode *train-test split* dengan perbandingan 80% untuk data pelatihan, dan 20% untuk data pengujian. Proses pemisahan dilakukan secara *stratified* untuk memastikan proporsi kelas pada data pelatihan dan pengujian tetap seimbang.

4.3.2 Pelatihan Model *Autoencoder*

Tabel berikut menyajikan konfigurasi *hyperparameter* yang digunakan pada model *Autoencoder*. *Hyperparameter* ini ditetapkan untuk mengoptimalkan proses pelatihan, termasuk pengaturan laju pembelajaran, fungsi aktivasi, jumlah *epoch*, dan mekanisme *stopping* untuk mencegah *overfitting*.

Tabel 4. 2 Tabel *Hyperparameter Autoencoder*

No.	Hyperparameter	Value
1.	<i>Learning Rate</i>	0.001
2.	<i>Optimizer</i>	Adam
3.	<i>Loss</i>	<i>Binary Crossentropy</i>
4.	<i>Activation</i>	ReLU
5.	<i>Output</i>	<i>Sigmoid</i>
6.	<i>Train Epoch</i>	30
7.	<i>Batch Size</i>	32
8.	<i>Callback</i>	<i>EarlyStopping</i>
9.	<i>Monitor</i>	<i>Val_loss</i>
10.	<i>Patience</i>	5
11.	<i>Restore Best Weights</i>	<i>True</i>

Pada tabel 4. 2 menampilkan tabel *hyperparameter* dari model *Autoencoder*, model tersebut dirancang dengan *learning rate* sebesar 0.001 menggunakan *optimizer Adam*, yang dipilih karena kemampuannya dalam menyesuaikan laju pembelajaran secara adaptif terhadap setiap parameter sehingga proses konvergensi menjadi lebih cepat dan stabil. Fungsi *loss* yang digunakan adalah *binary crossentropy*, yang sesuai untuk rekonstruksi citra dengan nilai piksel yang telah dinormalisasi pada rentang 0 hingga 1.

Fungsi aktivasi ReLU diterapkan pada *bottleneck* untuk mempercepat proses pembelajaran dengan mencegah masalah *vanishing gradient*, dan aktivasi *Sigmoid* digunakan pada *output* untuk menghasilkan rekonstruksi nilai piksel dalam skala yang diinginkan. Pelatihan dilakukan selama 30 *epoch* dengan *batch size* 32, proses ini dapat berhenti lebih awal karena penggunaan *callback EarlyStopping* yang memantau nilai *val_loss*. Jika tidak terjadi peningkatan performa selama 5 *epoch* berturut-turut, pelatihan akan dihentikan secara otomatis, dan bobot terbaik sebelum terjadinya *overfitting* akan dipulihkan untuk memastikan model memiliki performa optimal pada data validasi.



Gambar 4.3 Hasil Ekstraksi *Autoencoder*

Pada gambar 4.3 menampilkan perbandingan antara citra asli dan pada baris pertama dan citra hasil rekonstruksi pada baris kedua. Pada hasil rekonstruksi terlihat bahwa struktur utama daun, pola warna hijau, dan bentuk umum objek masih terjaga, meskipun beberapa detail halus seperti tekstur dan tepian daun mengalami penyederhanaan. Hal ini merupakan indikasi bahwa *Autoencoder* berhasil menangkap fitur visual yang relevan, seperti gradasi warna, perbedaan intensitas cahaya, dan bentuk objek utama.

4.3.3 Pelatihan Model *Multilayer Perceptron* (MLP)

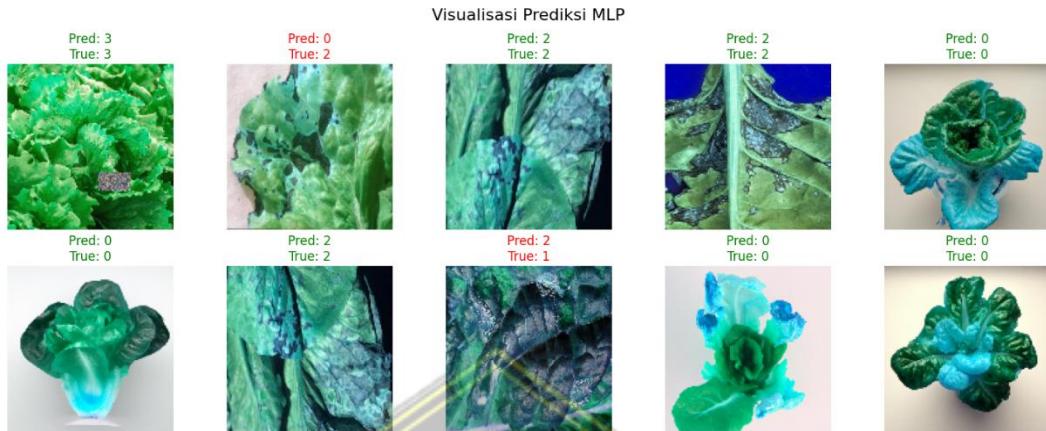
Tabel berikut menyajikan konfigurasi *hyperparameter* yang digunakan pada model *Multilayer Perceptron* (MLP). *Hyperparameter* ini dirancang untuk menfukung proses klasifikasi berdasarkan fitur yang telah diekstraksi dari *Autoencoder*, dengan pengaturan mencakup laju pembelajaran, fungsi aktivasi, jumlah *epoch* pelatihan, serta penggunaan *callback EarlyStopping* untuk mengoptimalkan performa model dan mencegah *overfitting*.

Tabel 4. 3 Tabel *Hyperparameter* MLP

No.	<i>Hyperparameter</i>	<i>Value</i>
1.	<i>Learning Rate</i>	0.001
2.	<i>Optimizer</i>	Adam
3.	<i>Loss</i>	<i>Categorical Crossentropy</i>
4.	<i>Metrics</i>	<i>Accuracy</i>
5.	<i>Activation</i>	ReLU
6.	<i>Output</i>	<i>Softmax</i>
7.	<i>Train Epoch</i>	25
8.	<i>Batch Size</i>	32
9.	<i>Validation Data</i>	<i>x_test_enc, y_test_cat</i>
10.	<i>Callback</i>	<i>EarlyStopping</i>
11.	<i>Monitor</i>	<i>Val_loss</i>
12.	<i>Patience</i>	5
13.	<i>Restore Best Weights</i>	<i>True</i>

Pada tabel 4. 3 menampilkan tabel *hyperparameter* dari model MLP, model tersebut menggunakan *learning rate* sebesar 0.001 dengan *optimizer Adam* yang dapat menyesuaikan laju pembelajaran secara adaptif untuk mempercepat proses konvergensi. Fungsi *loss* yang digunakan adalah *categorical crossentropy*, sesuai dengan permasalahan klasifikasi multi-kelas, serta *metrics* yang dipantau adalah *accuracy* untuk mengukur tingkat ketepatan prediksi. Fungsi aktivasi ReLU diterapkan pada *hidden layer*, sedangkan fungsi aktivasi *Softmax* digunakan pada lapisan *output layer* untuk menghasilkan distribusi probabilitas pada masing-masing kelas. Pelatihan dilakukan dengan maksimal 25 *epoch* dan *batch size* 32, dengan proses validasi menggunakan data (*X_test_enc, y_test_cat*) untuk memantau kinerja model selama pelatihan. Selain itu, digunakan *callback EarlyStopping* yang memantau *val_loss* untuk menghentikan pelatihan lebih awal apabila tidak terjadi penurunan nilai kerugian validasi selama 5 *epoch* berturut-turut, serta *restore best weights* diaktifkan agar model

menggunakan bobot terbaik sebelum overfitting terjadi, sehingga hasil pelatihan lebih optimal pada data uji.



Gambar 4. 4 Dataset hasil prediksi menggunakan MLP

Pada gambar 4. 4 menunjukkan hasil visualisasi prediksi model MLP terhadap beberapa citra daun *lettuce iceberg*. Setiap citra menampilkan dua informasi penting, yaitu *pred* (hasil prediksi model) dan *true* (label sebenarnya). Angka-angka tersebut merepresentasikan kelas yang telah didefinisikan sebelumnya, seperti kategori defisiensi Nitrogen, *Phosphor*, Kalium, atau *Healty*. Label prediksi ditampilkan dalam warna hijau apabila model memberikan hasil yang sesuai dengan label sebenarnya, sedangkan warna merah menandakan prediksi yang salah. Dari visualisasi ini terlihat bahwa model mampu mengklasifikasikan sebagian besar citra dengan benar, namun pada beberapa kasus masih terjadi kesalahan prediksi, yang kemungkinan disebabkan oleh kemiripan pola antar kelas atau kompleksitas tekstur daun.

4.3.4 Evaluasi Model

Evaluasi dilakukan menggunakan metrik akurasi, presisi, dan *F1-Score*. Akurasi mengukur jumlah prediksi yang tepat terhadap keseluruhan data uji, sedangkan *F1-Score* digunakan untuk mengevaluasi keseimbangan antara presisi dan *Recall*, khususnya pada data yang memiliki distribusi kelas tidak seimbang. Hasil evaluasi menunjukkan bahwa sistem memiliki performa yang baik dengan akurasi mencapai 86%, Presisi mencapai 89%, *Recall* mencapai 87%, dan *F1-Score* mencapai 88% Evaluasi visual

terhadap citra validasi juga menunjukkan bahwa sebagian besar prediksi sudah sesuai dengan label aslinya.

Evaluasi model dilakukan dengan menggunakan empat metrik utama, yaitu akurasi, presisi, *recall*, dan *F1-Score*, untuk menilai kinerja model secara menyeluruh. Akurasi digunakan untuk mengukur seberapa besar proporsi prediksi yang benar dibandingkan dengan total data uji yang tersedia. Meskipun akurasi memberikan gambaran umum performa model, metrik ini bisa kurang representatif jika data memiliki distribusi kelas yang tidak seimbang.

Untuk melengkapi pengukuran, digunakan model presisi yang mengukur sejauh mana model mampu menghindari kesalahan dalam klasifikasi positif. Presisi didefinisikan sebagai rasio antara jumlah prediksi positif yang benar terhadap seluruh prediksi positif yang dihasilkan oleh model. Dengan kata lain, presisi menunjukkan seberapa tepat model dalam memutuskan bahwa suatu sampel termasuk dalam kelas tertentu. Presisi menjadi sangat penting terutama ketika konsekuensi dari prediksi salah positif cukup besar.

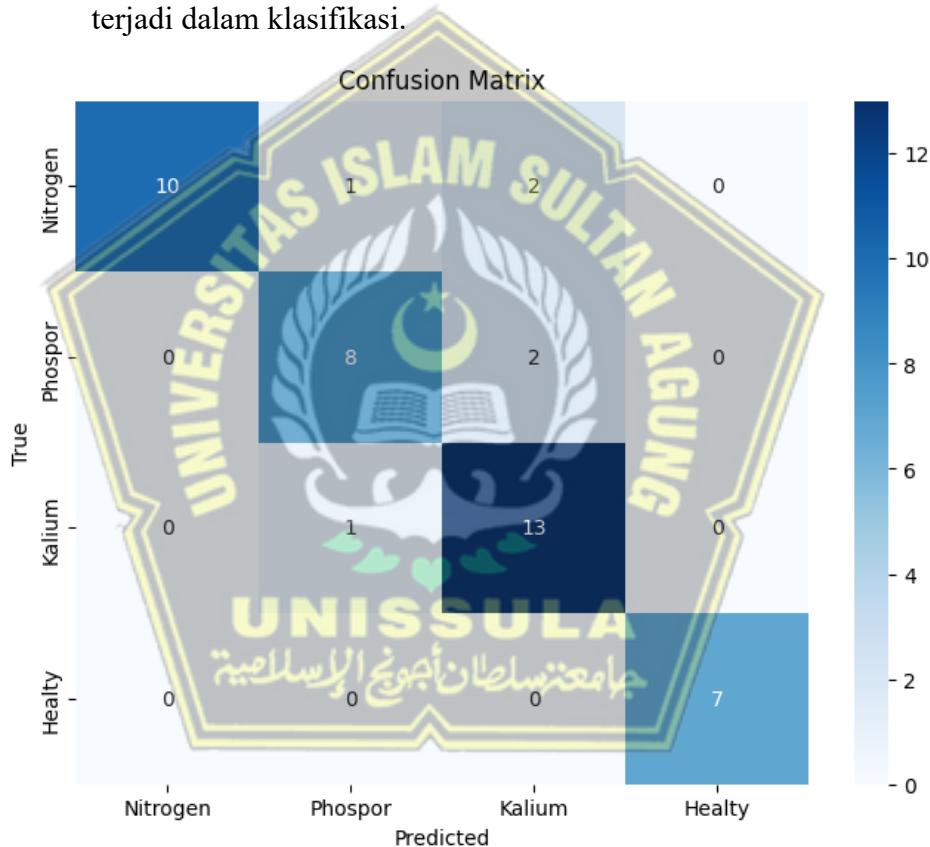
Selanjutnya, metrik *F1-Score* digunakan sebagai ukuran gabungan antara presisi dan *Recall* untuk memberikan evaluasi yang seimbang. *F1-Score* sangat berguna dalam kasus di mana data tidak seimbang, karena mempertimbangkan baik prediksi yang benar maupun yang tidak terdeteksi. Nilai *F1-Score* yang tinggi menandakan bahwa model tidak hanya akurat, tetapi juga konsisten dalam mengidentifikasi kelas target tanpa banyak salah klasifikasi.

Hasil evaluasi menunjukkan bahwa sistem memiliki performa yang baik, dengan capaian akurasi sebesar 86%, presisi sebesar 89%, *Recall* sebesar 87% dan *F1-Score* sebesar 88%. Selain evaluasi berbasis angka, juga dilakukan evaluasi visual terhadap hasil prediksi dengan membandingkannya terhadap label asli pada sejumlah citra validasi. Hasilnya menunjukkan bahwa sebagian besar prediksi telah sesuai, yang menandakan bahwa sistem ini dapat digunakan secara andal untuk

membantu identifikasi kekurangan nutrisi pada tanaman berdasarkan citra daun.

1. Visualisasi Evaluasi Model

Untuk memberikan gambaran visual mengenai performa klasifikasi model terhadap data uji, berikut ditampilkan *Confusion Matrix* hasil prediksi model MLP terhadap empat kelas kondisi daun. Visualisasi ini membantu dalam mengidentifikasi sejauh mana model berhasil mengenali setiap kelas secara benar maupun kesalahan yang terjadi dalam klasifikasi.



Gambar 4. 5 *Confusion Matrix*

Pada gambar 4. 5 menampilkan sebuah *Confusion Matrix* yang ditunjukkan di atas menggambarkan performa klasifikasi model MLP dalam mengklasifikasikan empat kelas kondisi daun, yaitu Nitrogen, Phosphorus, Potassium, dan Healthy. Setiap baris mewakili label sebenarnya (*True Label*), sedangkan setiap kolom menunjukkan label yang diprediksi oleh model.

Dari matriks diatas, dapat dilihat bahwa:

1. Kelas Nitrogen memiliki total 13 data uji, dengan 10 data diklasifikasikan dengan benar, 1 data salah diklasifikasikan sebagai *Phosphor*, dan 2 data salah diklasifikasikan sebagai Kalium.
 2. Kelas *Phosphor*, dari 10 data uji, hanya 8 data yang diklasifikasikan dengan benar, sementara 2 lainnya diklasifikasikan sebagai Kalium.
 3. Kelas Kalium, dari 14 data uji, ada 13 data yang diklasifikasikan dengan benar, dan 1 data diklasifikasikan sebagai *Phosphor*.
 4. Kelas *Healthy* memiliki performa yang tinggi dengan 7 data berhasil diprediksi dengan benar.
2. Visualisasi Kinerja Model

Selain menggunakan *Confusion Matrix*, evaluasi performa model juga disajikan dalam bentuk grafik batang untuk memudahkan analisis secara visual terhadap masing-masing metrik evaluasi. Grafik ini memperlihatkan nilai dari empat metrik utama, yaitu Akurasi, Presisi, *Recall*, dan *F1-Score* yang mencerminkan kinerja keseluruhan model dalam mengklasifikasikan kondisi daun dengan benar.

Tabel 4. 4 Tabel Evaluasi Model

Akurasi	Presisi	Recall	F1-Score
0.86	0.89	0.87	0.88

Pada tabel 4.2 menampilkan evaluasi kinerja dari model MLP dengan menampilkan metrik utama yaitu, Akurasi, Presisi, *Recall*, dan *F1-Score* yang masing-masing diukur berdasarkan performa model klasifikasi terhadap data uji. Dari grafik data tersebut, dapat dilihat bahwa nilai akurasi model mencapai 0.86, nilai presisi mencapai 0.89, nilai *Recall* mencapai 0.87, dan nilai *F1-Score* mencapai 0.88.

Secara keseluruhan, hasil evaluasi ini menunjukkan bahwa model MLP yang dikembangkan mampu melakukan klasifikasi dengan performa yang cukup baik dan stabil, serta menunjukkan

klasifikasi dengan performa yang cukup baik dan stabil, serta menunjukkan generalisasi yang layak terhadap data uji yang belum pernah dilihat sebelumnya. Keempat metrik ini hampir sejajar secara visual juga menandakan tidak adanya ketimpangan ekstrem antara kemampuan model dalam memprediksi dan mengenali pola data secara menyeluruh.

4.4 Implementasi Antar Muka Menggunakan *Streamlit*

4.4.1 Fitur Utama Aplikasi *Streamlit*

Aplikasi antarmuka yang dibangun menggunakan *Streamlit* dilengkapi dengan sejumlah fitur utama yang dirancang untuk mempermudah pengguna dalam melakukan deteksi kekurangan nutrisi tanaman. Adapun fitur-fitur tersebut adalah sebagai berikut:

1. *Upload* dan *capture* gambar daun langsung dari perangkat.

Pengguna dapat dengan mudah memilih dan mengunggah citra daun dari perangkat lokal mereka, baik melalui *desktop* maupun *mobile*. *Streamlit* secara otomatis akan membaca *file* gambar tersebut dan menampilkannya di layar.

2. Tampilan visual gambar yang diunggah.

Setelah gambar diunggah, sistem akan menampilkan pratinjau gambar tersebut secara langsung di halaman antarmuka. Fitur ini penting untuk memastikan bahwa gambar yang dipilih sudah benar sebelum diproses lebih lanjut.

3. Hasil klasifikasi jenis kekurangan nutrisi yang ditampilkan secara *real-time*.

Setelah gambar diproses oleh model *Autoencoder* dan *MLP* di *backend*, hasil deteksi akan ditampilkan secara langsung dalam bentuk teks informatif. Hasil ini mencakup jenis kekurangan nutrisi yang teridentifikasi pada daun berdasarkan citra yang diunggah.

4.4.2 Cara Akses *Streamlit*

Agar sistem ini dapat digunakan secara fleksibel, termasuk melalui perangkat *smartphone*, disediakan dua cara akses utama terhadap aplikasi *Streamlit* yang berjalan secara lokal. Pertama, pengguna dapat menjalankan aplikasi pada jaringan lokal menggunakan perintah "*Streamlit run app.py*". setelah dijalankan, aplikasi akan tersedia pada alamat IP lokal dari *desktop* tempat aplikasi dijalankan. *mobile* yang berada dalam jaringan *Wi-Fi* yang sama dapat mengakses aplikasi ini melalui *browser* dengan mengetikkan alamat IP lokal tersebut, diikuti oleh *port default Streamlit (8501)*.

Kedua, untuk penggunaan yang lebih praktis dan dapat diakses dari mana saja tanpa harus berada pada jaringan yang sama, aplikasi ini sudah *deploy* kedalam *Streamlit Cloud*. Untuk cara akses nya bisa menggunakan "<https://detection-lettuce-iceberg.streamlit.app>"

4.4.3 Hasil Implementasi

1. Tampilan *Home*

Halaman utama sistem ini menampilkan judul aplikasi serta penjelasan singkat mengenai tujuan penelitian. Sistem ini dirancang untuk mengidentifikasi kekurangan nutrisi pada daun *lettuce iceberg* dengan memanfaatkan metode *Autoencoder* dan *Multilayer Perceptron (MLP)*.



Gambar 4. 6 Tampilan *home* pada *desktop*

Sistem Deteksi Kekurangan Nutrisi Daun Lettuce Iceberg

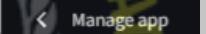
Aplikasi ini menggunakan Deep Learning (Autoencoder + MLP) untuk mendeteksi kekurangan nutrisi pada daun lettuce iceberg.

Kategori Deteksi:

- Kekurangan Nitrogen
- Kekurangan Fosfor
- Kekurangan Kalium
- Kondisi Sehat

Fitur:

- Upload gambar daun
- Deteksi langsung dari kamera
- Menampilkan persentase tian.



Gambar 4. 7 Tampilan *home* pada *mobile*

Pada gambar 4. 6 dan 4. 7 menampilkan kategori deteksi yang mencakup kekurangan unsur hara seperti Nitrogen, Fosfor, Kalium, serta kondisi Sehat. Selain itu, sistem juga dilengkapi dengan fitur untuk mengunggah gambar daun, melakukan deteksi langsung melalui kamera, serta menampilkan persentase kemungkinan dari setiap hasil klasifikasi.

2. Hasil Implementasi sebelum di *Upload*

Sebelum proses *upload* gambar dilakukan, antarmuka awal dari sistem yang dibangun akan menampilkan halaman utama dengan menampilkan sederhana dan informatif. Pada tampilan ini, pengguna diberikan petunjuk untuk mengunggah gambar daun yang akan dianalisis. Hal ini dirancang agar pengguna dapat dengan mudah memahami langkah awal dalam penggunaan sistem. Adapun tampilan antarmuka sebelum *upload* dapat dilihat pada gambar berikut:



Gambar 4. 8 Tampilan prediksi pada *desktop* sebelum *capture* gambar



Gambar 4. 9 Tampilan prediksi pada *mobile* sebelum *capture* gambar

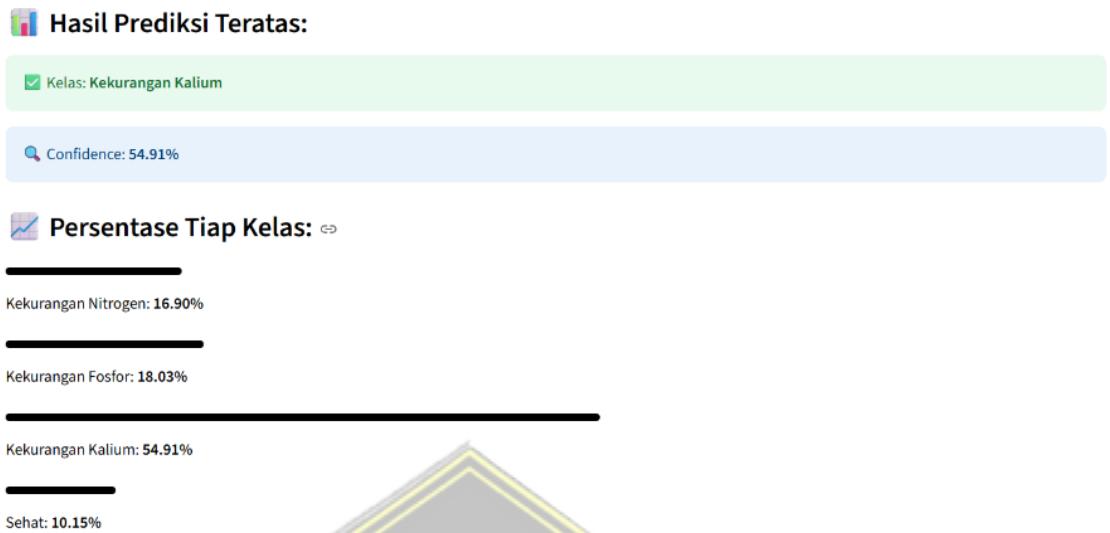
Pada gambar 4. 8 dan 4. 9 menampilkan tampilan *Streamlit* sebelum *upload* gambar dari aplikasi deteksi kekurangan nutrisi.

Tampilan ini menunjukkan halaman utama aplikasi yang memfasilitasi pengguna untuk mengunggah gambar daun atau *capture* gambar yang akan dianalisis. Fitur utama pada halaman ini adalah komponen *File uploader*, di mana pengguna dapat melakukan "drag and drop", memilih *File* gambar secara manual menggunakan tombol *Browse Files*, atau men-*capture* secara langsung. Sistem ini mendukung format *File* gambar seperti JPG, PNG, dan JPEG, dengan batas ukuran maksimum 200MB per gambar, memungkinkan fleksibilitas dalam proses *input* data.

Secara keseluruhan, tampilan ini tidak hanya mencerminkan aspek fungsional dari sistem, tetapi juga menunjukkan kesiapan aplikasi dalam mendukung deteksi secara *real-time*, berbasis *web*, tanpa harus menginstal perangkat lunak tambahan. Antarmuka yang ramah pengguna ini menjadi bagian penting dari keberhasilan implementasi model deteksi kekurangan nutrisi berbasis *Deep Learning* yang telah dikembangkan sebelumnya menggunakan *Autoencoder* dan *MLP*.

3. Hasil Implementasi setelah di *Upload*

Setelah pengguna berhasil mengunggah gambar daun, sistem akan secara otomatis memproses citra yang diberikan, melakukan ekstraksi fitur, dan mengklasifikasikan jenis defisiensi nutrisi yang terdeteksi menggunakan model *MLP*. Hasil prediksi ditampilkan dalam bentuk label klasifikasi, yang menunjukkan kategori kekurangan nutrisi pada daun seperti Nitrogen, Fosfor, Kalium, atau kondisi sehat. Tampilan antarmuka setelah proses *upload* ditunjukkan pada gambar berikut:



Gambar 4. 10 Tampilan prediksi pada desktop setelah capture gambar



Gambar 4. 11 Tampilan prediksi pada mobile setelah capture gambar

Pada Gambar 4. 10 dan 4. 11 menampilkan hasil akhir dari antarmuka pengguna aplikasi berbasis *Streamlit* yang dikembangkan untuk mendeteksi kekurangan nutrisi pada daun *lettuce iceberg*. Setelah pengguna mengunggah gambar daun melalui tampilan awal aplikasi, sistem secara otomatis akan memproses citra tersebut menggunakan *Flowchart* model yang telah dibangun, yang terdiri dari proses *Preprocessing*, ekstraksi fitur menggunakan *Autoencoder*, dan klasifikasi menggunakan *Multilayer Perceptron* (MLP).

Berdasarkan hasil prediksi gambar, sistem mendeteksi bahwa kondisi tanaman yang dianalisis memiliki kemungkinan tertinggi mengalami kekurangan kalium dengan tingkat kepercayaan sebesar 54,91%. Selain itu, persentase untuk kelas lainnya menunjukkan bahwa tanaman ini memiliki kemungkinan 16,90% mengalami kekurangan nitrogen, 18,03% mengalami kekurangan fosfor, dan 10,15% dalam kondisi sehat.

4.5 Pembahasan Implementasi

Implementasi sistem deteksi kekurangan nutrisi pada daun *lettuce iceberg* menggunakan kombinasi *Autoencoder* dan *Multilayer Perceptron* (MLP) telah berhasil direalisasikan secara menyeluruh. Sistem ini dirancang untuk memproses citra daun secara otomatis mulai dari tahap *Preprocessing*, ekstraksi fitur, hingga menghasilkan prediksi jenis kekurangan nutrisi yang dialami oleh tanaman. Proses klasifikasi terbukti berjalan dengan baik, di mana model *Autoencoder* mampu menangkap fitur penting dari citra daun secara efisien, sementara model MLP dapat mengklasifikasikan jenis defisiensi nutrisi dengan akurasi yang memuaskan.

Sistem yang dibangun menunjukkan performa yang cukup baik, dengan tingkat akurasi mencapai 86%, presisi sebesar 89%, *Recall* sebesar 87% dan *F1-Score* sebesar 88%, yang menunjukkan bahwa model mampu memprediksi jenis kekurangan nutrisi dengan ketepata yang tinggi dan konsistensi hasil yang baik. Nilai metrik tersebut mengindikasikan kemampuan generalisasi model terhadap data uji, sehingga dapat diandalkan

dalam melakukan klasifikasi citra daun yang belum pernah dilihat sebelumnya.

Dari sisi antarmuka, pengguna *Streamlit* sebagai *platform* sebagai aplikasi berbasis web memberikan nilai tambah yang signifikan. Sistem telah berhasil diintegrasikan ke dalam antarmuka pengguna berbasis *Streamlit*, memungkinkan pengguna untuk mengunggah gambar daun dan langsung menerima hasil klasifikasi secara *real-time* melalui browser, baik di perangkat *desktop* maupun *mobile*, tanpa memerlukan instalasi perangkat lunak tambahan. Antarmuka yang dibangun bersifat ringan, interaktif, dan ramah pengguna dari berbagai kalangan.

Secara keseluruhan, sistem ini telah berhasil memenuhi tujuan pengembangan, yaitu menyediakan alat bantu berbasis kecerdasan buatan yang mempu mendeteksi kekurangan nutrisi pada tanaman secara otomatis, akurat, dan mudah diakses. Dengan adanya sistem ini, diharapkan pengguna seperti petani, peneliti, maupun praktisi pertanian dapat lebih cepat dalam mengambil tindakan yang tepat untuk mengatasi defisiensi nutrisi pada tanaman.

Sebagai catatan dan arahan pengembangan sistem ke depan, disarankan agar dilakukan perluasan dan peningkatan variasi dataset, terutama dari segi pencahayaan, sudut pengambilan gambar, serta kondisi daun yang lebih beragam. Hal ini penting agar model mampu melakukan *generalization* yang lebih baik terhadap kondisi nyata di lapangan. Selain itu, penerapan Teknik data *augmentation* seperti *rotasi*, *flipping*, dan *zoom* juga dapat digunakan untuk meningkatkan keragaman data pelatihan tanpa harus melakukan pengumpulan data baru secara besar-besaran. Dengan langkah tersebut, diharapkan performa sistem dapat ditingkatkan dan lebih adaptif dalam menghadapi variasi data yang lebih kompleks di dunia nyata.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Sistem yang dibangun menunjukkan performa yang cukup baik, dengan tingkat akurasi mencapai 86%, presisi mencapai 89%, *Recall* 87% dan *F1-Score* mencapai 88%, yang berarti model mampu memprediksi jenis kekurangan nutrisi dengan tingkat ketepatan yang tinggi. Hasil tersebut diperoleh dari proses pelatihan menggunakan 147 citra daun dan pengujian sebanyak 36 citra daun, yang masing-masing mewakili kondisi dari kelasnya masing-masing. Selain itu, sistem telah berhasil diintegrasikan ke dalam antarmuka pengguna berbasis *Streamlit*, sehingga memungkinkan pengguna untuk mengunggah gambar daun dan langsung menerima hasil klasifikasi melalui *browser*, baik di perangkat *desktop* maupun *mobile*.

5.2 Saran

Untuk meningkatkan performa sistem kedepannya, disarankan agar dilakukan perluasan dan peningkatan variasi dataset, terutama dari segi pencahayaan, sudut pengambilan gambar, dan kondisi daun yang lebih beragam. Hal ini penting agar model dapat melakukan generalisasi dengan lebih baik pada kondisi di dunia nyata. Teknik augmentasi data seperti rotasi, *flipping*, dan *zoom* dapat diterapkan untuk meningkatkan keragaman data pelatihan tanpa harus mengumpulkan data baru dalam jumlah besar.

DAFTAR PUSTAKA

- ADIANGGIALI, A., IRAWATI, I. D., HADIYOSO, S., & LATIP, R. (2023). Classification of Nutrient Deficiencies Based on Leaf Image in Hydroponic Lettuce using MobileNet Architecture. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 11(4), 958. <https://doi.org/10.26760/elkomika.v11i4.958>
- Al-Safaar, D., & Al-Yaseen, W. L. (2023). Hybrid AE-MLP: Hybrid Deep Learning Model Based on Autoencoder and Multilayer Perceptron Model for Intrusion Detection System. *International Journal of Intelligent Engineering and Systems*, 16(2), 35–49. <https://doi.org/10.22266/ijies2023.0430.04>
- Armita, D., Wahdaniyah, W., Hafsan, H., & Al Amanah, H. (2022). Diagnosis Visual Masalah Unsur Hara Esensial Pada Berbagai Jenis Tanaman. *Teknosains: Media Informasi Sains dan Teknologi*, 16(1), 139–150. <https://doi.org/10.24252/teknosains.v16i1.28639>
- Bedi, P., & Gole, P. (2021). Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network. *Artificial Intelligence in Agriculture*, 5, 90–101. <https://doi.org/10.1016/j.aiia.2021.05.002>
- Boukhlifa, G., & Chibani, Y. (2024). A Lightweight CNN Design Based on Convolutional Autoencoder for Tomato Disease Identification. *Proceedings - 8th IEEE International Conference on Image and Signal Processing and their Applications, ISPA 2024, April 2024*. <https://doi.org/10.1109/ISPA59904.2024.10536761>
- Djidonou, D., & Leskovar, D. I. (2019). Seasonal changes in growth, nitrogen nutrition, and yield of hydroponic lettuce. *HortScience*, 54(1), 76–85. <https://doi.org/10.21273/HORTSCI13567-18>
- Hong, J., Xu, F., Chen, G., Huang, X., Wang, S., Du, L., & Ding, G. (2022). Evaluation of the Effects of Nitrogen, Phosphorus, and Potassium Applications on the Growth, Yield, and Quality of Lettuce (*Lactuca sativa* L.). *Agronomy*, 12(10). <https://doi.org/10.3390/agronomy12102477>

- Huddar, S., Prabhushetty, K., Jakati, J., Havaldar, R., & Sirdeshpande, N. (2024). Deep autoencoder based image enhancement approach with hybrid feature extraction for plant disease detection using supervised classification. *International Journal of Electrical and Computer Engineering*, 14(4), 3971–3985. <https://doi.org/10.11591/ijece.v14i4.pp3971-3985>
- Iatrou, M., Karydas, C., Tseni, X., & Mourelatos, S. (2022). Representation Learning with a Variational Autoencoder for Predicting Nitrogen Requirement in Rice. *Remote Sensing*, 14(23). <https://doi.org/10.3390/rs14235978>
- Kolhar, S., Jagtap, J., & Shastri, R. (2024). Deep Neural Networks for Classifying Nutrient Deficiencies in Rice Plants Using Leaf Images. *International Journal of Computing and Digital Systems*, 16(1), 305–314. <https://doi.org/10.12785/ijcds/160124>
- Kusuma, J., Rubianto, Rosnelly, R., Hartono, & Hayadi, B. H. (2023). Klasifikasi Penyakit Daun Pada Tanaman Jagung Menggunakan Algoritma Support Vector Machine, K-Nearest Neighbors dan Multilayer Perceptron. *Journal of Applied Computer Science and Technology*, 4(1), 1–6. <https://doi.org/10.52158/jacost.v4i1.484>
- Lu, J., Peng, K., Wang, Q., & Sun, C. (2023). Lettuce Plant Trace-Element-Deficiency Symptom Identification via Machine Vision Methods. *Agriculture (Switzerland)*, 13(8). <https://doi.org/10.3390/agriculture13081614>
- Naeem, S., Ali, A., Chesneau, C., Tahir, M. H., Jamal, F., Sherwani, R. A. K., & Hassan, M. U. (2021). The classification of medicinal plant leaves based on multispectral and texture feature using machine learning approach. *Agronomy*, 11(2). <https://doi.org/10.3390/agronomy11020263>
- naz, U., & Malik, M. M. (2023). A Comprehensive Review of Plant Disease Detection Using Deep Learning. *University of Wah Journal of Computer Science*, 5, 1–12.
- Okasha, A. M., Eldib, E. M., Elmetwalli, A. H., Farooque, A. A., Yaseen, Z. M., & Elsayed, S. (2022). Maximization of Water Productivity and Yield of Two

- Iceberg Lettuce Cultivars in Hydroponic Farming System Using Magnetically Treated Saline Water. *Agriculture (Switzerland)*, 12(1), 1–18. <https://doi.org/10.3390/agriculture12010101>
- Oktavia, D. A. P., Rizal, S., & Pratiwi, N. K. C. (2022). Klasifikasi Gejala Defisiensi Nutrisi Pada Tanaman Padi Menggunakan CNN Dengan Arsitektur Resnet-50. *e-Proceeding of Engineering*, 8(6), 3171–3175.
- Pacal, I., Kunduracioglu, I., Alma, M. H., Deveci, M., Kadry, S., Nedoma, J., Slany, V., & Martinek, R. (2024). A systematic review of deep learning techniques for plant diseases. *Artificial Intelligence Review*, 57(11). <https://doi.org/10.1007/s10462-024-10944-7>
- Pandey, P., Veazie, P., Whipker, B., & Young, S. (2023). Predicting foliar nutrient concentrations and nutrient deficiencies of hydroponic lettuce using hyperspectral imaging. *Biosystems Engineering*, 230, 458–469. <https://doi.org/10.1016/j.biosystemseng.2023.05.005>
- Sikati, J., & Nouaze, J. C. (2023). YOLO-NPK: A Lightweight Deep Network for Lettuce Nutrient Deficiency Classification Based on Improved YOLOv8 Nano †. *Engineering Proceedings*, 58(1), 0–7. <https://doi.org/10.3390/ecsa-10-16256>
- Song, Y., Meng, X., Li, Y., Liu, Z., & Zhang, H. (2023). An Integrative Approach for Mineral Nutrient Quantification in Dioscorea Leaves: Uniting Image Processing and Machine Learning. *Traitement du Signal*, 40(3), 1153–1161. <https://doi.org/10.18280/ts.400331>
- Taha, M. F., Abdalla, A., Elmasry, G., Gouda, M., Zhou, L., Zhao, N., Liang, N., Niu, Z., Hassanein, A., Al-Rejaie, S., He, Y., & Qiu, Z. (2022). Using Deep Convolutional Neural Network for Image-Based Diagnosis of Nutrient Deficiencies in Plants Grown in Aquaponics. *Chemosensors*, 10(2), 1–23. <https://doi.org/10.3390/chemosensors10020045>
- Vanacore, L., El-Nakhel, C., Modarelli, G. C., Roushanel, Y., Pannico, A., Langellotti, A. L., Masi, P., Cirillo, C., & De Pascale, S. (2024). Growth, Ecophysiological Responses, and Leaf Mineral Composition of Lettuce and Curly Endive in Hydroponic and Aquaponic Systems. *Plants*, 13(20), 1–14.

<https://doi.org/10.3390/plants13202852>

- Veazie, P., Chen, H., Hicks, K., Holley, J., Eylands, N., Mattson, N., Boldt, J., Brewer, D., Lopez, R., & Whipker, B. (2024). Developing supervised machine learning algorithms to classify lettuce foliar tissue samples into interpretation zones for 11 plant essential nutrients. *Urban Agriculture and Regional Food Systems*, 9(1), 1–11. <https://doi.org/10.1002/uar2.70002>
- Vought, K., Bayabil, H. K., Pompeo, J., Crawford, D., Zhang, Y., Correll, M., & Martin-Ryals, A. (2024). Dynamics of micro and macronutrients in a hydroponic nutrient film technique system under lettuce cultivation. *Helijon*, 10(11), e32316. <https://doi.org/10.1016/j.heliyon.2024.e32316>
- Xie, J., Lv, S., Zhang, X., Song, W., Liu, X., & Lu, Y. (2025). Exploring Nutrient Deficiencies in Lettuce Crops: Utilizing Advanced Multidimensional Image Analysis for Precision Diagnosis. *Sensors*, 25(7). <https://doi.org/10.3390/s25071957>

