KLASIFIKASI KEBUTUHAN NUTRISI TANAMAN MELON PADA SISTEM HIDROPONIK BERBASIS *INTERNET OF THINGS* MENGGUNAKAN *SUPPORT VECTOR MACHINE*

SKRIPSI

Oleh :
LAILY SABRINA HAPSARI
NIM. 220605110113

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025

KLASIFIKASI KEBUTUHAN NUTRISI TANAMAN MELON PADA SISTEM HIDROPONIK BERBASIS *INTERNET OF THINGS* MENGGUNAKAN *SUPPORT VECTOR MACHINE*

SKRIPSI

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh :
LAILY SABRINA HAPSARI
NIM. 220605110113

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025

HALAMAN PERSETUJUAN

KLASIFIKASI KEBUTUHAN NUTRISI TANAMAN MELON PADA SISTEM HIDROPONIK BERBASIS *INTERNET OF THINGS* MENGGUNAKAN *SUPPORT VECTOR MACHINE*

SKRIPSI

Oleh :
LAILY SABRINA HAPSARI
NIM. 220605110113

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 20 November 2025

Pembimbing I,	Pembimbing II,
Shoffin Nahwa Utama, M.T.	Nurizal Dwi Priandani, M.Kom.
NIP. 19860703 202012 1 003	NIP. 19920830 202203 1 001

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Supriyono, M.Kom.
NIP. 19841010 201903 1 012

HALAMAN PENGESAHAN

KLASIFIKASI KEBUTUHAN NUTRISI TANAMAN MELON PADA SISTEM HIDROPONIK BERBASIS *INTERNET OF THINGS* MENGGUNAKAN *SUPPORT VECTOR MACHINE*

SKRIPSI

Oleh :
LAILY SABRINA HAPSARI
NIM. 220605110113

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 17 Desember 2025

Susunan Dewan Penguji

Ketua Penguji	: Johan Ericka Wahyu Prakasa, M.Kom NIP. 19831213 201903 1 004	( )
Anggota Penguji I	: Ajib Hanani, M.T NIP. 19840731 202321 1 013	( )
Anggota Penguji II	: Shoffin Nahwa Utama, M.T NIP. 19860703 202012 1 003	( )
Anggota Penguji III	: Nurizal Dwi Priandani, M.Kom. NIP. 19920830 202203 1 001	( )

Mengetahui dan Mengesahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Supriyono, M.Kom.
NIP. 19841010 201903 1 012

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama	: LAILY SABRINA HAPSARI
NIM	: 220605110113
Fakultas / Program Studi	: Sains dan Teknologi / Teknik Informatika
Judul Skripsi	: Klasifikasi Kebutuhan Nutrisi Tanaman Melon Pada Sistem Hidroponik Berbasis *Internet of Things* Menggunakan *Support Vector Machine*

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 20 November 2025
Yang membuat pernyataan,

Laily Sabrina Hapsari

MOTTO

... Sesungguhnya beserta kesulitan ada kemudahan ...

HALAMAN PERSEMBAHAN

Dengan penuh rasa syukur *Alhamdulillah* kepada Allah SWT atas segala rahmat, hidayah, kemudahan, dan kekuatan yang telah diberikan-Nya, sehingga skripsi ini dapat terselesaikan dengan baik. Karya ini penulis persembahkan kepada:

Bunda dan Ayah,

Atas segala dukungan, doa, dan kasih sayang tak terhitung yang telah diberikan.

Teman, Sahabat, dan Orang Terdekat,

Atas dukungan, motivasi, dan semangat yang tidak pernah absen di setiap langkah perjalanan hingga di titik ini.

Dosen,

Atas ilmu, bimbingan, dan arahannya.

Keluarga besar Infinity Teknik Informatika Angkatan 2022,

Berjuang bersama, saling mendukung dari awal hingga akhir.

Dan untuk diri sendiri,

Terimakasih.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh.

Syukur Alhamdulillah senantiasa penulis panjatkan kepada Allah SWT karena atas rahmat dan hidayah-Nya, sehingga penulis dapat tuntas menulis skripsi yang berjudul "Klasifikasi Kebutuhan Nutrisi Tanaman Melon Pada Sistem Hidroponik Berbasis *Internet of Things* Menggunakan *Support Vector Machine*" dengan baik dan lancar.

Penulisan skripsi ini tidak terlepas dari bimbingan, doa, dan dukungan dari berbagai pihak. Oleh sebab itu, penulis menyampaikan terimakasih kepada :

1. Prof. Dr. Hj. Ilfi Nur Diana, M.Si., CAHRM., CRMP selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang beserta seluruh jajaran pimpinan universitas.
2. Dr. Agus Mulyono, M.Kes selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang beserta jajarannya.
3. Supriyono, M.Kom., selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Shoffin Nahwa Utama, M.T., selaku dosen pembimbing I yang senantiasa mendukung, membimbing, dan memberikan masukan dengan sabar selama penulisan.
5. Nurizal Dwi Priandani, M.Kom., selaku dosen pembimbing II yang telah memberikan masukan dan bimbingan untuk penulis dalam menyelesaikan skripsi ini.

6. Johan Ericka Wahyu Prakasa, M.Kom., selaku dosen penguji I dan Ajib Hanani, M.T., selaku dosen penguji II yang telah memberikan saran, masukan, kritik, dan diskusi yang membangun sehingga penulis dapat menyelesaikan skripsi dengan baik.
7. Dosen, laboran, dan jajaran staf Program Studi Teknik Informatika yang telah memberikan ilmu, pengetahuan, dan dukungan selama penulis menjalani studi.
8. Kedua orangtua dan kedua adik tercinta, Muhammad Agus Sahbana, S.T, M.T. dan Rita Ayu Bulan Trisna, S.H. serta Alya Nugrahanti dan Muhammad Fauzan Azhiima untuk segala dukungan dan doa yang diberikan selama penulis menyelesaikan studi ini.
9. Teman terdekat, Alhubul Austad Ramadhan dan Achmad Furqon Rachmadie yang saling mendukung dan menguatkan serta mendorong penulis untuk terus semangat dan berjuang bersama.
10. Teman-teman tim penelitian hidroponik yang saling membantu, berbagi ide, dan bertukar pendapat selama proses penelitian berlangsung.
11. Keluarga besar "*Infinity*" Teknik Informatika Angkatan 2022 yang saling membantu, mengingatkan, berbagi ilmu, dan berbagi pengalaman yang tak terlupakan bersama dari awal studi hingga akhir.
12. Seluruh pengurus HIMATIF "*Encoder*" periode 2023 dan 2024 yang telah menjadi rumah kedua penulis di kampus dan menyalurkan kontribusi selama menjalani masa studi.

13. Seluruh pengurus *Google Developer Group on Campus* Universitas Islam Negeri Maulana Malik Ibrahim Malang yang sudah berbagi pengalaman, ilmu, dan bantuan selama masa studi penulis.
14. Seluruh pengurus ONTAKI yang telah memberikan tempat untuk penulis bisa bereksplorasi dan meningkatkan skill di bidang robotika dan otomasi.
15. Sultan Arif Hidayatullah, seseorang yang telah memberikan motivasi dan dukungan baik secara moril maupun emosional terbesar untuk penulis selama penyusunan skripsi.
16. Teman-teman KKM 25 "*Sahitya Nawasena*" yang telah menjadi keluarga, sahabat, sekaligus rekan selama berkegiatan dan mengabdi di desa.
17. Rekan *Wedding Organizer* Mantu Bareng khususnya kepada Handika Ventura selaku owner yang telah memberikan doa, dukungan, motivasi, kesempatan, dan masukan kepada penulis.
18. Seluruh pihak yang tidak dapat penulis sebutkan satu per satu atas bantuannya kepada penulis selama masa studi berlangsung.
19. Diri sendiri yang telah berjuang, bertahan, dan berdiri tegap hingga saat ini menyelesaikan apa yang telah dimulai dengan baik.

Penulis menyadari bahwa karya ini masih memiliki ruang untuk pengembangan. Oleh karena itu, penulis membuka diri terhadap kritik, saran, maupun masukan konstruktif demi peningkatan kualitas dan penyempurnaan penelitian ke depan. Penulis meyakini bahwa proses belajar tidak pernah berhenti, dan setiap masukan merupakan langkah penting dalam perjalanan akademik dan pengembangan ilmu pengetahuan. Besar harapannya, skripsi ini dapat memberikan

manfaat bagi pembaca, peneliti berikutnya, serta pihak yang berkepentingan. Semoga penelitian ini turut berkontribusi pada perkembangan ilmu pengetahuan dan tetap relevan untuk digunakan di masa mendatang.

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Malang, 20 November 2025

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Hapsari, Laily Sabrina. 2025. **Klasifikasi Kebutuhan Nutrisi Tanaman Melon Pada Sistem Hidroponik Berbasis *Internet of Things* Menggunakan *Support Vector Machine*.** Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Shoffin Nahwa Utama, M.T. (II) Nurizal Dwi Priandani, M.Kom.

Kata Kunci: hidroponik, *Internet of Things* (IoT), klasifikasi nutrisi, *Support Vector Machine* (SVM)

Budidaya melon secara hidroponik membutuhkan pengelolaan nutrisi yang presisi karena perubahan pH, TDS, dan suhu air sangat memengaruhi penyerapan hara. Kendala yang muncul pada pertanian hidroponik adalah proses pemantauan kondisi nutrisi yang masih dilakukan secara manual, sehingga berisiko menimbulkan keterlambatan dalam mendeteksi keadaan nutrisi yang kurang, cukup, atau berlebih. Penelitian ini bertujuan mengembangkan sistem pemantauan berbasis *Internet of Things* (IoT) yang terintegrasi dengan algoritma *Support Vector Machine* (SVM) untuk mengklasifikasikan kebutuhan nutrisi secara otomatis. Proses penelitian meliputi akuisisi data sensor secara *real-time*, pembersihan data, normalisasi Min-Max, penyeimbangan kelas menggunakan *Random Under Sampling*, dan reduksi dimensi dengan *Principal Component Analysis* sebelum dilakukan klasifikasi menggunakan SVM kernel linear. Hasil pengujian menunjukkan bahwa model SVM memperoleh akurasi 94,63% pada rasio 80:20, 94,51% pada rasio 70:30, dan 94,83% pada rasio 60:40, yang menandakan performa model cukup stabil terhadap variasi data latih dan uji. Pengujian kalibrasi sensor menghasilkan nilai *Mean Absolute Error* sebesar 0,335 untuk pH, 134,692 ppm untuk TDS, dan 1,275°C untuk suhu air, yang menunjukkan tingkat penyimpangan yang masih dapat diterima. Secara keseluruhan, sistem IoT dan model SVM ini mampu memberikan klasifikasi nutrisi secara akurat dan efisien sehingga mendukung optimalisasi pengelolaan larutan nutrisi pada budidaya melon hidroponik.

ABSTRACT

Hapsari, Laily Sabrina. 2025. ***Classification of Nutrient Requirements for Melon Plants in a Hydroponic System Based on the Internet of Things Using Support Vector Machine.*** Undergraduate Thesis. Informatics Engineering Study Program, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang. Supervisor: (I) Shoffin Nahwa Utama, M.T., (II) Nurizal Dwi Priandani, M.Kom.

Keywords: hydroponics, Internet of Things (IoT), nutrient classification, Support Vector Machine (SVM)

Hydroponic melon cultivation requires precise nutrient management because fluctuations in pH, TDS, and water temperature significantly affect nutrient absorption. A major challenge in hydroponic agriculture is that nutrient condition monitoring is still performed manually, which increases the risk of delayed detection of nutrient states such as deficient, adequate, or excessive. This study aims to develop an Internet of Things (IoT)-based monitoring system integrated with the Support Vector Machine (SVM) algorithm to automatically classify nutrient requirements. The research process includes real-time sensor data acquisition, data cleaning, Min-Max normalization, class balancing using Random Under Sampling, and dimensionality reduction with Principal Component Analysis before performing classification using a linear kernel SVM. The results show that the SVM model achieved accuracies of 94.63% with an 80:20 ratio, 94.51% with a 70:30 ratio, and 94.83% with a 60:40 ratio, indicating stable performance across different train-test distributions. Sensor calibration tests produced Mean Absolute Error values of 0.335 for pH, 134.692 ppm for TDS, and 1.275°C for water temperature, demonstrating that the sensors operate within acceptable deviation levels. Overall, the developed IoT system and SVM model are capable of providing accurate and efficient nutrient classification, thereby supporting the optimization of nutrient solution management in hydroponic melon cultivation.

مستخلص البحث

ەافساري، ليلي صبرينا. 2025. تصنيف احتياجات المغذيات لنبات الشمام في نظام الزراعة المائية المعتمد على إنترنت الأشياء باستخدام آلة المتجه الداعم. البحث الجامعي. قسم الهندسة المعلوماتية، كلية العلوم والتكنولوجيا بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: صفين نحو أوتامى، الماجستير؛ المشرف الثاني: نور ريزال دوي قرياندانى، الماجستير.

الكلمات الرئيسية: زراعة مائية، إنترنت أشياء (IoT)، تصنيف مغذيات، آلة متجه داعم (SVM)

زراعة الشمام بالطريقة المائية تتطلب إدارة دقيقة للعناصر الغذائية لأن تغير الرقم الهيدروجيني، وTDS، ودرجة حرارة الماء يؤثر بشكل كبير على امتصاص المغذيات. من العقبات التي تواجه الزراعة المائية هي عملية مراقبة حالة العناصر الغذائية التي تتم يدويًا حتى الآن، مما يزيد من خطر التأخر في اكتشاف حالة نقص أو كفاية أو زيادة العناصر الغذائية. يهدف هذا البحث إلى تطوير نظام مراقبة قائم على الإنترنت للأشياء (IoT) متكامل مع خوارزمية آلة المتجه الداعم (SVM) لتصنيف احتياجات العناصر الغذائية تلقائيًا. تشمل عملية البحث الحصول على بيانات المستشعرات في الوقت الفعلي، وتنظيف البيانات، والتطبيع باستخدام *Min-Max*، وموازنة الفئات باستخدام اختيار عينة عشوائية (*Random Under Sampling*)، وتقليل الأبعاد باستخدام تحليل المركبات الرئيسية قبل التصنيف باستخدام SVM بنواة خطية .أظهرت نتائج الاختبار أن نموذج SVM حقق دقة بنسبة 94,63% عند نسبة 80:20، و94,51% عند نسبة 70:30، و94,83% عند نسبة 60:40، ما يشير إلى أن أداء النموذج مستقر إلى حد كبير تجاه تنوع بيانات التدريب والاختبار. أسفرت اختبارات معايرة المستشعر عن قيمة متوسط الخطأ المطلق بمقدار 0,335 لـ أس هيدروجيني pH. ، و134,692 جزء في المليون لTDS ، و1,275 درجة مئوية لدرجة حرارة الماء، مما يشير إلى مستوى انحراف لا يزال مقبولاً. بشكل عام، فإن نظام إنترنت الأشياء ونموذج SVM قادران على تقديم تصنيف دقيق وفعال للعناصر الغذائية بما يدعم تحسين إدارة المحاليل المغذية في زراعة الشمام بالنظام المائي.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi dan komunikasi saat ini telah membawa perubahan signifikan dalam berbagai bidang, termasuk sektor pertanian. Salah satu inovasi yang banyak dikembangkan adalah pertanian berbasis *Internet of Things* (IoT) (Kurniawan & Prasetyo, 2022). IoT memungkinkan perangkat sensor digunakan untuk memantau kondisi lingkungan secara *real time*. Sistem ini menjadi solusi potensial bagi pertanian modern, khususnya pada metode hidroponik yang bergantung pada kestabilan nutrisi dan kondisi lingkungan (Nugroho & Fathurrahman, 2021). Dengan demikian, pemanfaatan IoT telah terbukti mampu meningkatkan efisiensi dalam pengelolaan budidaya pertanian (Santoso & Firdaus, 2023).

Tanaman melon merupakan salah satu komoditas hortikultura dengan nilai ekonomi tinggi di Indonesia. Permintaan pasar terhadap buah melon terus meningkat seiring dengan tingginya konsumsi masyarakat (Prasetyo & Sari, 2020). Namun, budidaya melon secara hidroponik menghadapi tantangan dalam menjaga keseimbangan pH larutan, *Total Dissolved Solid* (TDS), suhu air, dan umur tanaman. Ketidakseimbangan faktor-faktor tersebut dapat menyebabkan tanaman mengalami defisiensi atau kelebihan nutrisi (Syahputra et al., 2022). Dampaknya adalah penurunan kualitas maupun kuantitas hasil panen yang merugikan petani (Dewi & Lestari, 2021).

Pengelolaan nutrisi yang tepat menjadi kunci dalam keberhasilan sistem hidroponik. Faktor pH dan TDS sangat menentukan ketersediaan unsur hara yang dapat diserap oleh tanaman (Wahyudi & Putra, 2023). Sementara itu, suhu air dan umur tanaman turut memengaruhi tingkat kebutuhan nutrisi (Arifin & Kusuma, 2020). Tanpa adanya sistem pengelolaan berbasis data, petani sering menghadapi kesulitan dalam menentukan dosis nutrisi yang sesuai (Wijayanti & Susanti, 2021). Alat manual yang digunakan petani hidroponik untuk mengukur nutrisi tidak menunjukkan secara langsung nutrisi yang terlarut pada sistem hidroponik sudah tergolong cukup atau tidak. Petani perlu menghitung terlebih dahulu data yang didapat dari alat manual untuk menentukan nutrisi tanaman sudah cukup atau belum.

Salah satu pendekatan yang dapat digunakan untuk mengatasi masalah ini adalah dengan memanfaatkan algoritma *Support Vector Machine* (SVM). Algoritma ini dikenal memiliki performa tinggi dalam menangani masalah klasifikasi dengan jumlah fitur terbatas (Anindita & Saputra, 2021). Dengan memanfaatkan data hasil pembacaan sensor IoT, SVM dapat digunakan untuk mengklasifikasikan kondisi kebutuhan nutrisi tanaman melon hidroponik (Rachman et al., 2021). Klasifikasi ini dapat dibagi ke dalam kategori kurang, cukup, atau berlebih sesuai dengan kondisi aktual (Wahyuni & Pratama, 2020).

Urgensi penelitian ini terletak pada integrasi antara IoT dan machine learning dalam mendukung pertanian presisi. Sistem klasifikasi nutrisi berbasis SVM dapat membantu petani lebih mudah mengambil keputusan terkait pengaturan nutrisi (Nugroho & Fathurrahman, 2021). Proses budidaya melon hidroponik pun

menjadi lebih efisien, produktif, dan berkelanjutan (Santoso & Firdaus, 2023). Hal ini sekaligus menjawab tantangan modernisasi pertanian dengan memanfaatkan teknologi sebagai bagian dari solusi (Syahputra et al., 2022).

Dalam perspektif Islam, manusia diberi amanah sebagai khalifah di bumi untuk menjaga keseimbangan alam. Hal ini ditegaskan dalam firman Allah SWT pada Surah *Al-An'am* ayat 141, yang berbunyi:

وَهُوَ الَّذِي أَنشَأَ جَنَّاتٍ مَّعْرُوشَاتٍ وَغَيْرَ مَعْرُوشَاتٍ وَالنَّخْلَ وَالزَّرْعَ مُخْتَلِفًا أُكُلُهُ وَالزَّيْتُونَ وَالرُّمَّانَ مُتَشَابِهًا وَغَيْرَ مُتَشَابِهٍ ۚ كُلُوا مِن ثَمَرِهِ إِذَا أَثْمَرَ وَآتُوا حَقَّهُ يَوْمَ حَصَادِهِ ۖ وَلَا تُسْرِفُوا ۚ إِنَّهُ لَا يُحِبُّ الْمُسْرِفِينَ

"Dan Dialah yang menjadikan kebun-kebun yang berjunjung dan yang tidak berjunjung, pohon kurma, tanam-tanaman yang bermacam-macam rasanya, zaitun, dan delima yang serupa dan yang tidak serupa. Makanlah dari buahnya bila dia berbuah, dan tunaikanlah haknya di hari memetik hasilnya, dan janganlah kamu berlebih-lebihan. Sesungguhnya Allah tidak menyukai orang yang berlebih-lebihan." (QS. Al-An'am [6]:141).

Menurut penafsiran para ulama, ayat ini menegaskan bahwa manusia diperintahkan untuk mengelola hasil pertanian dengan penuh tanggung jawab (Rahmawati, 2021). Ayat tersebut juga menekankan pentingnya keberlanjutan dan larangan berlebih-lebihan dalam setiap aktivitas pertanian (Hidayat, 2020; Yusuf, 2019).

Selain landasan Al-Qur'an, penelitian ini sejalan dengan ajaran Rasulullah SAW yang menekankan pentingnya menjaga kelestarian alam dan memanfaatkannya secara bijak. Rasulullah SAW dalam haditsnya bersabda:

"Tidaklah seorang muslim menanam pohon atau menabur benih, lalu sebagian darinya dimakan oleh burung, manusia, atau binatang, melainkan itu menjadi sedekah baginya." [HR. Bukhari No. 2320].

Hadits ini menunjukkan bahwa setiap usaha manusia dalam bercocok tanam memiliki nilai keberkahan (Rahmawati, 2021). Selain itu, hadits ini juga menegaskan pentingnya pemanfaatan teknologi untuk meningkatkan produktivitas pertanian (Hidayat, 2020). Pesan moralnya adalah bahwa keberlanjutan lingkungan harus tetap dijaga meskipun teknologi diterapkan (Yusuf, 2019).

Beberapa penelitian sebelumnya telah membahas penggunaan algoritma machine learning dalam bidang pertanian. Penelitian Wahyuni dan Pratama (2020) menunjukkan bahwa SVM memiliki performa tinggi dalam mengklasifikasikan data pertanian. Selanjutnya, penelitian Anindita dan Saputra (2021) menegaskan efektivitas SVM pada dataset dengan jumlah fitur terbatas. Namun demikian, sebagian besar penelitian terdahulu lebih berfokus pada komoditas lain, bukan pada melon hidroponik (Rachman et al., 2021). Hal ini menunjukkan adanya peluang penelitian lebih lanjut (Nugroho & Fathurrahman, 2021). Penelitian ini hadir untuk mengisi kesenjangan tersebut dengan menitikberatkan pada klasifikasi kebutuhan nutrisi melon hidroponik berbasis IoT (Santoso & Firdaus, 2023).

Selain memperkuat basis akademis, penelitian ini juga memiliki manfaat praktis bagi masyarakat dan pelaku usaha tani. Sistem klasifikasi nutrisi yang dihasilkan dapat membantu petani dalam mengurangi kesalahan pemberian nutrisi (Wahyudi & Putra, 2023). Efisiensi penggunaan pupuk dapat tercapai sehingga lebih ramah lingkungan (Dewi & Lestari, 2021). Dampak lanjutannya adalah peningkatan kualitas dan kuantitas hasil panen, yang pada akhirnya dapat mendukung kesejahteraan petani (Syahputra et al., 2022).

Lebih jauh, penelitian ini diharapkan memberikan kontribusi nyata dalam pengembangan sistem pertanian presisi berbasis teknologi. Integrasi sensor IoT dan algoritma *machine learning* diharapkan menghasilkan model klasifikasi yang mampu memberikan rekomendasi berbasis data secara akurat (Arifin & Kusuma, 2020). Kontribusi utama penelitian ini adalah menghadirkan solusi inovatif dalam pengelolaan nutrisi tanaman hidroponik, khususnya melon (Prasetyo & Sari, 2020). Dengan demikian, penelitian ini mendukung efisiensi sumber daya, meningkatkan produktivitas, dan memperkuat ketahanan pangan nasional (Wijayanti & Susanti, 2021).

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah bagaimana mengembangkan sistem *Internet of Things* untuk mengklasifikasikan kebutuhan nutrisi tanaman melon hidroponik menggunakan *Support Vector Machine* yang secara otomatis menunjukkan klasifikasi nutrisi termasuk kurang, cukup, atau berlebih?

1.3 Batasan Masalah

Agar penelitian ini lebih terarah dan tidak melebar dari fokus utama, maka batasan masalah dalam penelitian ini ditentukan sebagai berikut:

1. Parameter yang digunakan sebagai fitur klasifikasi terbatas pada pH larutan, suhu air, TDS, dan umur tanaman.
2. Data diperoleh dari sensor IoT yang dipasang pada sistem hidroponik dengan kalibrasi pengukuran alat manual.

3. Klasifikasi kebutuhan nutrisi tanaman hanya dibatasi pada tiga kategori, yaitu *kurang, cukup,* dan *berlebih*.
4. Penelitian ini hanya difokuskan pada proses klasifikasi kebutuhan nutrisi tanaman, tanpa membahas lebih lanjut dampak hasil klasifikasi terhadap pertumbuhan tanaman.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk mengembangkan sistem *Internet of Things* yang mampu mengklasifikasikan kebutuhan nutrisi tanaman melon hidroponik menggunakan algoritma *Support Vector Machine*, sehingga secara otomatis dapat menunjukkan kategori nutrisi kurang, cukup, atau berlebih.

1.5 Manfaat Penelitian

Hasil penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Bagi peneliti, penelitian ini menjadi sarana pengembangan pengetahuan dan keterampilan dalam penerapan teknologi IoT dan algoritma *machine learning*, khususnya *Support Vector Machine*, dalam bidang pertanian presisi.
2. Bagi petani atau praktisi pertanian, sistem klasifikasi nutrisi tanaman melon hidroponik yang dihasilkan dapat membantu dalam pengambilan keputusan pemupukan secara lebih tepat, efisien, dan berbasis data.
3. Bagi akademisi dan peneliti selanjutnya, penelitian ini dapat menjadi referensi dan dasar pengembangan lebih lanjut pada penelitian di bidang

pertanian cerdas, klasifikasi data sensor, maupun penerapan metode machine learning lainnya.

4. Bagi masyarakat luas, penelitian ini dapat mendukung ketersediaan produk hortikultura berkualitas tinggi melalui budidaya yang lebih ramah lingkungan dan efisien, sehingga sejalan dengan prinsip pertanian berkelanjutan.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Penelitian oleh Sulaiman et al. (2024) mengembangkan model *hybrid ensemble machine learning*, melibatkan *Random Forest,* SVM*, dan* KNN untuk memprediksi konsentrasi fosfor dalam larutan hidroponik menggunakan data spek-troskopi. Hasil menunjukkan bahwa SVM tunggal mencapai akurasi 99,6%, sedangkan kombinasi SVM/KNN via *stacking* mencapai 99,73%. Studi ini relevan karena menunjukkan potensi SVM dalam klasifikasi nutrisi berbasis data spektral, meskipun fokusnya bukan pada melon dan fitur yang digunakan berbeda.

Islam et al. (2024) menggunakan teknik *image processing* dan *SVM* untuk mengidentifikasi gejala stres lingkungan pada bibit cabai di *greenhouse (plant factory)*. Dengan metode ini, klasifikasi *stress symptoms* mencapai akurasi tinggi dan menunjukkan kecocokan SVM untuk aplikasi agronomi berbasis citra. Meskipun bukan pada hidroponik melon, pendekatan ini menguatkan ragam penerapan *SVM* dalam sistem pertanian terkendali.

Muftashiva, Munadi, & Haryanto (2024) dalam penelitiannya mengembangkan sistem *smart plant monitoring* pada budidaya melon hidroponik berbasis Internet of Things menggunakan metode *dutch bucket*. Sistem ini dirancang untuk memantau kondisi lingkungan tanaman, seperti kelembapan, suhu, serta nutrisi secara real-time, sehingga petani dapat mengetahui status tanaman tanpa harus melakukan pengecekan manual. Hasil penelitian tersebut menunjukkan bahwa penerapan IoT mampu meningkatkan efisiensi pemantauan tanaman melon

hidroponik dengan memberikan data yang lebih cepat, akurat, dan mudah diakses melalui perangkat digital.

Tabel 2.1 Penelitian terkait

No	Judul Penelitian	Hasil	Persamaan	Perbedaan
1	*Hybrid Ensemble Machine Learning Models for Nutrient Solution Phosphorus Estimation in Hydroponics* (Sulaiman et al., 2024)	SVM akurasi 99,6%; *stacking* hingga 99,73%	Penggunaan SVM untuk klasifikasi nutrisi	Bukan melon, basis data spektral
2	*Application of SVM for Stress Symptom Classification in Chili Seedlings under Plant Factory Conditions* (Islam et al., 2024)	Akurasi tinggi dalam identifikasi gejala stres	Penerapan SVM dalam kondisi terkontrol	Respons terhadap stres, bukan nutrisi
3	*Sistem Smart Plant Monitoring Pada Hidroponik Melon Berbasis Internet Of Things* (Muftashiva, Munadi, & Haryanto, 2024)	Implementasi sistem *smart plant monitoring* berbasis *Internet of Things*	Pemanfaatan *Internet of Things* pada budidaya melon hidroponik	Menggunakan metode *dutch bucket*. Sedangkan penelitian ini menggunakan SVM.
4	*Support Vector Machine untuk Prediksi Pola Pertumbuhan Selada Hidroponik* (Rahmadi et al., 2025)	Akurasi ±80,5%	Penggunaan SVM di sistem hidroponik NFT	Fokus pada pertumbuhan, bukan nutrisi
5	*Pemantauan Nutrisi Hidroponik Menggunakan Sensor pH/TDS dan Logika Fuzzy* (Julianti & Kurniawan, 2024)	Sistem akurasi tinggi dalam kendali nutrisi	Pemantauan nutrisi pH/TDS berbasis IoT	Metode fuzzy, bukan klasifikasi SVM
Usulan Penelitian				
6	Klasifikasi Kebutuhan Nutrisi Tanaman Melon Hidroponik Berbasis *Internet of Things* Menggunakan *Support Vector Machine*	-	Klasifikasi kebutuhan nutrisi tanaman melon hidroponik berbasis data sensor IoT yang di integrasikan di *greenhouse* hidroponik dengan menggunakan metode *Support Vector Machine* (SVM).	

2.2 Klasifikasi

Menurut Kamus Besar Bahasa Indonesia (KBBI), klasifikasi adalah penyusunan secara sistematis dalam kelompok atau golongan berdasarkan kaidah atau standar tertentu (KBBI, 2024). Secara umum, klasifikasi merupakan proses mengelompokkan data atau objek berdasarkan kesamaan ciri atau atribut yang

dimiliki. Dalam ranah *machine learning*, klasifikasi termasuk ke dalam metode *supervised learning* yang menggunakan data berlabel untuk mempelajari pola, kemudian diaplikasikan pada data baru guna menentukan kelas yang sesuai (IBM, 2024). Berbagai algoritma digunakan dalam klasifikasi, seperti *Decision Tree*, *Naïve Bayes*, *Artificial Neural Network*, hingga *Support Vector Machine* (SVM). SVM menjadi salah satu metode yang banyak digunakan karena kemampuannya membangun *hyperplane* optimal untuk memisahkan data dalam kelas yang berbeda, bahkan pada data berdimensi tinggi (Firdausi, 2024). Dengan adanya klasifikasi, proses analisis data yang sebelumnya dilakukan secara manual dapat ditingkatkan akurasinya serta dipercepat melalui dukungan komputasi.

2.3 Nutrisi Hidroponik

Dalam budidaya melon hidroponik, kebutuhan nutrisi berbeda pada setiap fase pertumbuhan, sehingga diperlukan pengaturan parameter pH, suhu air, dan konsentrasi larutan secara tepat. Pada fase vegetatif (0–3 minggu setelah tanam (MST)), pH larutan umumnya dipertahankan pada kisaran 5,8–6,3 untuk memaksimalkan penyerapan unsur hara makro dan mikro (Grozine, 2023). Suhu air nutrisi yang ideal berada pada kisaran 20–27 °C, karena rentang ini dapat menjaga ketersediaan oksigen terlarut dan mencegah stres akar (Grozine, 2023). Konsentrasi larutan nutrisi pada fase ini diberikan secara bertahap, mulai dari sekitar 800 ppm pada minggu pertama dan meningkat menjadi 1000 ppm pada minggu ketiga (Tappi & Tandibayang, 2025).

Memasuki fase generatif (berbunga hingga berbuah), kebutuhan nutrisi tanaman meningkat signifikan. pH larutan tetap dijaga stabil pada kisaran 5,8–6,4

(Grozine, 2023), sementara suhu air dapat dipertahankan pada 20–27 °C, meskipun beberapa penelitian melaporkan tanaman melon masih dapat tumbuh baik hingga suhu air larutan 30 °C apabila sistem aerasi dan sirkulasi terkontrol (Kawasaki et al., 2020). Pada fase ini, konsentrasi nutrisi larutan dinaikkan ke level 1500–1700 ppm, yang terbukti mendukung pembentukan bunga dan pembesaran buah secara optimal (Tappi & Tandibayang, 2025; Furoidah, 2018; Darmawan, Dinarto, & Widodo, 2024).

Tabel 2.2 *Ground Truth* Kebutuhan Nutrisi Tanaman

Fase Pertumbuhan	Parameter	Kurang	Cukup (Optimal)	Berlebih	Sumber
Vegetatif (0–3 MST)	**pH**	< 5,8	5,8 – 6,4	> 6,4	Grozine, 2023
	Suhu Air	< 20 °C	20 – 27 °C	> 27 °C	Grozine, 2023
	PPM	< 800 ppm	800 – 1000 ppm	> 1000 ppm	Tappi & Tandibayang, 2025
Generatif (≥4 MST, berbunga – berbuah)	**pH**	< 5,8	5,8 – 6,4	> 6,4	Grozine, 2023
	Suhu Air	< 20 °C	20 – 27 °C	> 27 °C	Kawasaki et al., 2020
	PPM	< 1500 ppm	1500 – 2000 ppm	> 2000 ppm	Tappi & Tandibayang, 2025; Furoidah, 2018; Darmawan et al., 2024

2.4 *Cleaning*

Tahap data *cleaning* merupakan bagian penting dalam *preprocessing* yang bertujuan untuk meningkatkan kualitas data sebelum masuk ke proses analisis atau pelatihan model. Data yang dikumpulkan dari sensor sering kali mengandung permasalahan seperti *missing values*, data duplikat, inkonsistensi, atau *noise* akibat gangguan teknis. Apabila tidak ditangani dengan baik, kondisi ini dapat

menyebabkan bias dalam hasil analisis dan menurunkan performa model klasifikasi yang dibangun.

Menurut García, Luengo, & Herrera (2016), kualitas data memiliki pengaruh yang lebih besar terhadap akurasi model dibandingkan dengan kompleksitas algoritma yang digunakan. Oleh karena itu, proses *cleaning* menjadi tahap yang tidak dapat diabaikan. Teknik yang umum digunakan antara lain adalah penghapusan atau imputasi nilai hilang, deteksi dan penanganan *outliers*, serta standarisasi format data. Abid, Farooqi, & Khan (2021) juga menekankan bahwa pembersihan data adalah langkah preventif yang memastikan hasil analisis lebih reliabel dan mudah direplikasi.

Dalam konteks penelitian ini, *cleaning* dilakukan untuk memastikan data hasil pembacaan sensor pH, suhu air, dan TDS/PPM benar-benar merepresentasikan kondisi aktual tanaman. Data yang terdeteksi *error* atau tidak logis, seperti nilai di luar batas biologis tanaman melon, akan diidentifikasi dan diperbaiki menggunakan teknik imputasi sederhana atau dihapus bila tidak relevan. Dengan demikian, dataset yang diperoleh menjadi lebih konsisten dan siap digunakan dalam tahap klasifikasi selanjutnya.

2.5 Normalisasi

Tahap normalisasi merupakan salah satu proses penting dalam *preprocessing* yang bertujuan untuk menyeragamkan skala data sehingga setiap atribut memiliki kontribusi yang seimbang dalam proses analisis maupun pelatihan model. Hal ini menjadi krusial karena data sensor pH, suhu air, dan TDS/PPM berada pada skala yang berbeda-beda. Misalnya, pH berkisar antara 0–14, suhu air

biasanya 15–35 °C, sementara TDS dapat mencapai ribuan ppm. Jika tidak dinormalisasi, atribut dengan skala besar (seperti TDS) akan mendominasi proses perhitungan jarak atau fungsi kernel dalam algoritma *machine learning*, termasuk *Support Vector Machine* yang digunakan dalam penelitian ini.

Menurut Han, Pei, & Kamber (2012), normalisasi data dapat meningkatkan kinerja algoritma dengan mempercepat konvergensi dan mengurangi bias akibat perbedaan skala antar fitur. Teknik normalisasi yang umum digunakan meliputi *Min-Max Normalization*, *Z-score Standardization*, dan *Decimal Scaling*. Pada penelitian ini, dipilih metode *Min-Max Normalization* karena dapat mengubah setiap atribut ke dalam rentang [0,1], sehingga lebih mudah diinterpretasikan dan sesuai untuk algoritma SVM yang sensitif terhadap jarak antar data (Patro & Sahu, 2015).

Dengan adanya normalisasi, data sensor yang beragam skala dapat diproses secara konsisten sehingga hasil klasifikasi kebutuhan nutrisi tanaman menjadi lebih akurat. Proses ini juga memastikan bahwa sistem IoT yang dibangun mampu memberikan rekomendasi yang proporsional berdasarkan setiap parameter lingkungan yang diukur.

2.6 *Balancing*

Data *balancing* adalah proses penyeimbangan distribusi kelas dalam dataset agar model klasifikasi tidak bias terhadap kelas mayoritas. Ketidakseimbangan kelas (*imbalanced dataset*) sering menyebabkan model menghasilkan prediksi yang tampak akurat secara keseluruhan, tetapi gagal mengenali kelas minoritas dengan baik. Hal ini penting diperhatikan karena dalam penelitian berbasis klasifikasi,

kualitas model tidak hanya ditentukan oleh performa pada kelas mayoritas, melainkan juga oleh kemampuannya dalam mengidentifikasi kelas dengan jumlah data yang lebih sedikit (Gao et al., 2025; Chen et al., 2024).

Balancing memungkinkan setiap kelas berkontribusi secara lebih proporsional dalam proses pembelajaran. Studi terbaru menegaskan bahwa teknik *balancing*, baik melalui pengurangan data pada kelas mayoritas maupun penambahan data pada kelas minoritas, dapat membantu mengurangi bias model dan meningkatkan keandalan hasil klasifikasi pada data multikelas (Chen et al., 2024). Lebih jauh, penelitian empiris menunjukkan bahwa meskipun balancing dapat menurunkan jumlah informasi pada kelas tertentu, langkah ini tetap bermanfaat karena memperbaiki sensitivitas model terhadap kelas minoritas yang jumlahnya lebih sedikit (Yang et al., 2024).

Dalam konteks penelitian ini, balancing diterapkan pada data hasil sensor yang dilabeli kondisi nutrisi tanaman (*kurang, cukup, berlebih*) menggunakan metode *Equalized Class Reconstruction* (ECR). Tujuannya adalah memastikan setiap kategori memiliki representasi data yang proporsional dan maksimum sehingga model *Support Vector Machine* dapat belajar secara lebih adil, tidak hanya fokus pada kelas mayoritas, tetapi juga mampu mendeteksi kelas minoritas dengan lebih baik.

2.7 *Principal Component Analysis* (PCA)

Principal Component Analysis (PCA) adalah metode reduksi dimensi yang digunakan untuk menyederhanakan dataset dengan banyak variabel menjadi sejumlah kecil komponen utama yang tetap mampu merepresentasikan sebagian

besar variasi data (Jolliffe & Cadima, 2016). Teknik ini bekerja dengan mengubah kumpulan variabel awal menjadi kombinasi linier baru yang tidak saling berkorelasi, yang disebut sebagai *principal components*. Komponen pertama menyimpan variasi data terbesar, diikuti komponen berikutnya yang menyimpan variasi terbesar kedua, dan seterusnya.

Dalam penelitian klasifikasi berbasis *machine learning*, PCA bermanfaat untuk meningkatkan efisiensi komputasi, mengurangi *noise*, dan mencegah *overfitting* dengan cara menghilangkan korelasi antarfitur (Sharma & Paliwal, 2021). PCA juga sering digunakan dalam pengolahan data sensor pertanian, misalnya untuk mengekstraksi pola utama dari parameter lingkungan sehingga hasil klasifikasi menjadi lebih akurat dan stabil (Zhang et al., 2022).

Secara matematis, PCA menggunakan dekomposisi matriks kovarian atau *singular value decomposition (SVD)* untuk menghasilkan vektor eigen dan nilai eigen. Jika X adalah matriks data dengan mean 0, maka kovarian dihitung dengan (Krishana, & Dr. Vinod Kumar, 2024):

$$C = \frac{1}{n-1} X^T X \tag{2.1}$$

Keterangan:
C : matriks kovarian
n : jumlah sampel data
X : matriks data yang sudah dikurangi mean
X^{T} : transpose dari matriks X

Selanjutnya dilakukan dekomposisi nilai eigen:

$$Cvi = \lambda ivi \tag{2.2}$$

Keterangan:
C : matriks kovarian
vi : eigenvector ke-i
λi : eigenvalue ke-i

dengan vi adalah *eigenvector* yang menjadi arah komponen utama dan λi adalah *eigenvalue* yang merepresentasikan besarnya variansi yang dijelaskan oleh komponen tersebut.

Dengan demikian, PCA relevan digunakan pada penelitian ini untuk membantu analisis data sensor hidroponik (pH, TDS, suhu air) dengan cara menyoroti dimensi paling berpengaruh sebelum masuk ke tahap klasifikasi menggunakan SVM.

2.8 *Support Vector Machine* (SVM)

Support Vector Machine (SVM) adalah algoritma *supervised learning* yang dirancang untuk menyelesaikan masalah klasifikasi dan regresi dengan cara menentukan *hyperplane* pemisah yang optimal di antara kelas data (Cervantes et al., 2020). Keunggulan SVM terletak pada prinsip *maximum margin*, yaitu memilih batas pemisah dengan jarak terluas antara dua kelas, sehingga model mampu melakukan generalisasi dengan baik meskipun ukuran dataset terbatas (Amaya-Tejera et al., 2024).

Dalam penelitian ini, SVM digunakan untuk mengklasifikasikan kebutuhan nutrisi tanaman melon hidroponik ke dalam tiga kategori, yaitu kurang, cukup, dan berlebih. Input klasifikasi berasal dari data sensor pH, TDS, dan suhu air. Pemilihan SVM relevan karena metode ini telah terbukti unggul dalam penelitian terdahulu terkait pertanian presisi. Misalnya, Islam et al. (2024) menggunakan SVM untuk mengklasifikasikan gejala stres pada bibit cabai dengan hasil akurasi tinggi, sementara Sulaiman et al. (2024) menerapkan SVM dalam estimasi nutrisi

hidroponik berbasis *machine learning* dengan tingkat keberhasilan yang signifikan. Hal ini menunjukkan bahwa SVM efektif diterapkan pada kasus pertanian berbasis data, termasuk pada penelitian ini.

Secara umum, fungsi keputusan dalam SVM dapat dituliskan sebagai (Shuzhanfan, 2018):

$$f(x) = w \cdot x + b \tag{2.3}$$

Keterangan:
$f(x)$: nilai fungsi keputusan
w : vektor bobot
x : vektor input fitur
b : bias

Tujuan optimisasi SVM adalah memaksimalkan margin dengan fungsi objektif:

$$min \frac{1}{2} \| w \|^2 \tag{2.4}$$

Keterangan:
w : vektor bobot
$|w||^2$: norm kuadrat bobot

dengan syarat,

$$yi(w \cdot xi + b) \geq 1 \tag{2.5}$$

Keterangan:
yi : label kelas
xi : vektor fitur sampel ke-i
w : bobot
b : bias

Untuk data yang tidak sepenuhnya dapat dipisahkan secara linear, digunakan variabel slack ξi dengan parameter regularisasi C:

$$\min \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{n} \xi i \tag{2.6}$$

Keterangan:
w : bobot
C : parameter regularisasi
ξi : slack variable / kesalahan yang diizinkan
n : jumlah sampel

Formulasi ini memungkinkan SVM tetap bekerja meskipun terdapat kesalahan klasifikasi kecil pada data latih.

2.9 Kernel Linear

Dalam *Support Vector Machine* (SVM), *kernel* berfungsi untuk mengubah data input ke dalam ruang fitur berdimensi lebih tinggi agar data dapat dipisahkan dengan *hyperplane*. Salah satu bentuk kernel yang paling sederhana adalah *kernel linear*, yang bekerja tanpa transformasi non-linear sehingga fungsi pemisah tetap berupa garis atau bidang datar (Harrington, 2020). Fungsi kernel linear dapat dituliskan sebagai (Cortes, C., & Vapnik, V, 1995):

$$K(xi, xj) = xi \cdot xj \tag{2.7}$$

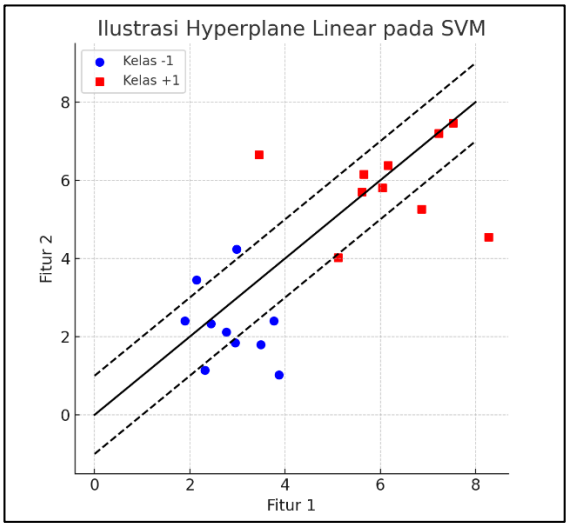
Keterangan:
$K(xi, xj)$: nilai kernel
xi : vektor fitur sampel pertama
xj : vektor fitur sampel kedua

Penggunaan *kernel linear* sangat sesuai pada dataset dengan jumlah fitur yang terbatas dan ketika hubungan antarvariabel cenderung linier. Dalam penelitian ini, fitur yang digunakan meliputi pH, TDS, suhu air, dan umur tanaman, yang secara logis memiliki korelasi linier terhadap klasifikasi status nutrisi (kurang, cukup, berlebih). Keunggulan kernel linear antara lain perhitungan yang sederhana, waktu komputasi lebih cepat, serta hasil yang lebih mudah diinterpretasikan

dibandingkan kernel non-linear seperti RBF atau polynomial (Cervantes et al., 2020).

Selain itu, penelitian terdahulu juga menunjukkan bahwa kernel linear dapat memberikan performa yang cukup baik pada bidang pertanian berbasis IoT ketika jumlah fitur relatif sedikit. Misalnya, SVM dengan kernel linear telah digunakan pada klasifikasi pertumbuhan tanaman hidroponik dengan akurasi kompetitif dibandingkan kernel kompleks lainnya (Rahmadi et al., 2025). Hal ini menjadi dasar bahwa kernel linear tepat digunakan dalam penelitian ini sebagai pendekatan klasifikasi kebutuhan nutrisi melon hidroponik berbasis IoT.

Pada gambar 2.1, menunjukkan ilustrasi sederhana penerapan *Support Vector Machine* (SVM) dengan *kernel linear* dalam memisahkan dua kelas data. Titik biru dan merah merepresentasikan dua kelas berbeda, sedangkan garis hitam merupakan *hyperplane* pemisah yang dibentuk oleh SVM. Garis putus-putus menandai batas margin yang mengapit *hyperplane*, di mana titik data terdekat disebut *support vector*. Prinsip *maximum margin* ini memungkinkan SVM untuk meminimalkan kesalahan klasifikasi sekaligus meningkatkan kemampuan generalisasi pada data baru (Cervantes et al., 2020; Amaya-Tejera et al., 2024).



Gambar 2.1 Ilustrasi Hyperplane Linear

2.10 *Confusion Matrix*

Confusion matrix merupakan metode evaluasi yang banyak digunakan dalam mengukur kinerja model klasifikasi. Matriks ini menyajikan perbandingan antara hasil prediksi model dengan data aktual yang diketahui, sehingga dapat menggambarkan tingkat akurasi maupun kesalahan klasifikasi (Hossin & Sulaiman, 2015). Struktur confusion matrix terdiri dari empat komponen utama, yaitu *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)*, yang masing-masing menjelaskan posisi hasil prediksi terhadap kenyataan (Chicco & Jurman, 2020). Pada tabel 2.3 berikut menunjukkan bentuk umum confusion matrix :

Tabel 2.3 Confusion Matrix

	Prediksi Positif	**Prediksi Negatif**
Aktual (Fakta) Positif	*True Positive* (TP)	*False Negative* (FN)
Aktual (Fakta) Negatif	*False Positive* (FP)	*True Negative* (TN)

True Positive (TP) adalah situasi hasil dari data aktual dan hasil prediksi keduanya benar. *True Negative* (TN) adalah kondisi di mana data aktual tidak benar,

tetapi hasil prediksi benar. *False Positive* (FP) terjadi ketika hasil prediksi salah, sementara data aktualnya benar. *False Negative* (FN) adalah keadaan ketika hasil prediksi maupun data aktual keduanya salah. Dalam evaluasi berbasis label akan mengevaluasi setiap label secara terpisah, yang pada dasarnya mengubah pengklasifikasi multi-label menjadi pengklasifikasi biner untuk setiap label. Pendekatan ini menghasilkan empat kemungkinan hasil prediksi: *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN). Metrik yang digunakan didefinisikan sebagai berikut (Swaminathan, Sathyanarayanan & Tantri, B Roopashri, 2024) :

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (2.8)$$

$$\text{Presisi} = \frac{TP}{TP+FP} \times 100\% \quad (2.9)$$

$$\text{Recall} = \frac{TP}{TP+FN} \times 100\% \quad (2.10)$$

$$\text{F1 Score} = 2 \times \frac{Precision \times Recall}{Precision+Recall} \times 100\% \quad (2.11)$$

Keterangan:
TP : *true positive*
TN : *true negative*
FP : *false positive*
FN : *false negative*

Evaluasi ini penting karena setiap metrik memberikan sudut pandang yang berbeda. *Accuracy* baik digunakan ketika distribusi kelas seimbang, namun pada data tidak seimbang, *precision*, *recall*, dan F1-*score* lebih dapat menggambarkan performa model secara adil (Sokolova & Lapalme, 2009). Oleh karena itu,

kombinasi dari metrik-metrik ini akan digunakan dalam penelitian untuk menilai kinerja model klasifikasi kebutuhan nutrisi tanaman melon hidroponik.

2.11 Pengujian Kalibrasi Sensor

Pengujian kalibrasi sensor merupakan tahap awal yang penting untuk memastikan keandalan data yang digunakan dalam penelitian. Sensor yang tidak terkalibrasi dengan baik dapat menghasilkan data yang bias atau menyimpang dari kondisi sebenarnya, sehingga berdampak langsung pada akurasi analisis maupun performa model klasifikasi yang dibangun. Menurut Vasylenko et al. (2020), proses kalibrasi dilakukan untuk menyesuaikan pembacaan sensor terhadap standar acuan sehingga kesalahan sistematis dapat diminimalisasi. Hal serupa ditegaskan oleh Roriz et al. (2021), bahwa kalibrasi bukan hanya meningkatkan akurasi, tetapi juga menjamin konsistensi pengukuran dari waktu ke waktu.

Kalibrasi sensor dilakukan dengan cara membandingkan hasil pembacaan sensor terhadap alat ukur manual yang dijadikan referensi. Setiap parameter sensor (pH, suhu air, dan TDS) diukur secara bersamaan menggunakan sensor dan alat referensi pada beberapa titik pengukuran. Selisih antara hasil sensor dan referensi dihitung sebagai error, kemudian dirata-ratakan untuk mengetahui tingkat akurasi sensor.

Metrik yang digunakan dalam penelitian ini adalah *Mean Absolute Error* (MAE) karena lebih sederhana dan mudah diinterpretasikan. MAE menunjukkan rata-rata besar deviasi pembacaan sensor terhadap nilai referensi. Rumus MAE ditunjukkan pada persamaan berikut (Willmott, C. J., & Matsuura, K, 2005):

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |yi^{sen} - yi^{ref}| \quad (2.12)$$

Keterangan:
N : jumlah sampel
yi^{sen} : nilai referensi
yi^{ref} : nilai hasil sensor

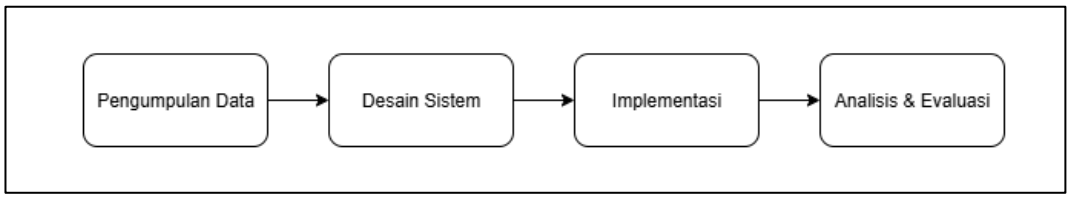
Sensor dianggap layak digunakan apabila nilai MAE berada dalam batas toleransi, misalnya pH ≤ 0,2; suhu air ≤ 0,5 °C; dan TDS ≤ 50–100 ppm (Willmott & Matsuura, 2005).

BAB III

DESAIN DAN IMPLEMENTASI

3.1 Prosedur Penelitian

Penelitian ini dilakukan melalui beberapa tahapan yang dirancang secara sistematis agar tujuan penelitian dapat tercapai dengan baik. Prosedur penelitian berfungsi sebagai gambaran umum mengenai alur kegiatan, mulai dari perancangan sistem hingga tahap analisis dan evaluasi hasil. Secara lebih jelas, prosedur penelitian ini ditunjukkan pada Gambar 3.1.



Gambar 3.1. Tahapan Prosedur Penelitian

Tahap pertama adalah pengumpulan data, yang dilakukan setelah sistem siap digunakan. Peneliti menggunakan sensor yang telah dipasang untuk mencatat data secara periodik sesuai dengan kebutuhan penelitian. Data ini menjadi dasar untuk analisis selanjutnya.

Tahap kedua adalah desain sistem, di mana peneliti merancang kebutuhan perangkat keras dan perangkat lunak yang digunakan. Rancangan ini mencakup pemilihan sensor IoT untuk mengukur parameter penting tanaman melon hidroponik seperti pH, suhu air, TDS, serta penentuan arsitektur sistem yang menghubungkan sensor dengan basis data.

Selanjutnya, tahap implementasi dilakukan dengan membangun model klasifikasi menggunakan algoritma *Support Vector Machine* (SVM) sesuai

rancangan yang telah dibuat. Model ini dilatih menggunakan data hasil preprocessing untuk menghasilkan sistem klasifikasi kebutuhan nutrisi tanaman melon hidroponik.

Tahap terakhir adalah analisis dan evaluasi, di mana peneliti menilai kinerja model dengan menggunakan metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score* dan pengujian kalibrasi sensor pada sistem IoT. Hasil analisis metrik evaluasi digunakan untuk mengetahui sejauh mana efektivitas model dalam melakukan klasifikasi, sekaligus memberikan gambaran mengenai perbaikan yang dapat dilakukan pada penelitian berikutnya. Sedangkan pengujian kalibrasi sensor digunakan untuk menguji seberapa akurat sensor yang digunakan untuk pembacaan parameter.

3.1.1 Pengumpulan Data

Tahap pengumpulan data merupakan langkah awal yang sangat penting dalam penelitian ini. Peneliti akan mengumpulkan data yang digunakan bersumber dari hasil pengukuran parameter tanaman melon hidroponik dengan sistem IoT. Parameter yang dicatat meliputi pH larutan nutrisi, nilai TDS (*Total Dissolved Solids*), suhu air sistem hidroponik, serta umur tanaman. Rincian fitur yang digunakan dalam penelitian ini ditunjukkan pada Tabel 3.1.

Tabel 3.1 Fitur penelitian

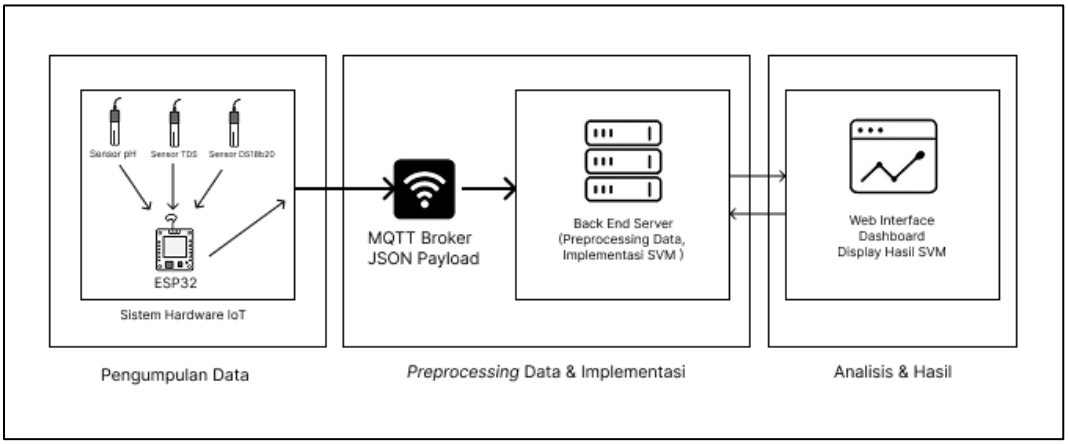
No	Fitur	Satuan	Keterangan
1	pH	-	Menunjukkan tingkat keasaman atau kebasaan larutan nutrisi hidroponik.
2	TDS	ppm	Mengukur jumlah padatan terlarut yang merepresentasikan konsentrasi nutrisi.
3	Suhu Air	°C	Menunjukkan kondisi air sistem hidroponik yang mempengaruhi penyerapan nutrisi tanaman.
4	Umur Tanaman	Hari (HST)	Menggambarkan fase pertumbuhan tanaman sejak ditanam.

Proses pengumpulan data dilakukan dengan memanfaatkan sensor yang terhubung pada sistem mikrokontroler. Sensor pH digunakan untuk mengukur tingkat keasaman larutan, sensor TDS untuk mengetahui konsentrasi nutrisi terlarut, dan sensor DS18b20 untuk mencatat suhu air pada sistem hidroponik. Sementara itu, fitur umur tanaman hanya berperan sebagai *context feature* yang digunakan untuk mengetahui fase pertumbuhan, bukan indikator nutrisi. Umur tanaman tidak memiliki hubungan langsung terhadap perubahan pH, TDS, maupun suhu, sehingga tidak digunakan sebagai fitur prediktif dalam proses labeling ataupun pemodelan. Data yang diperoleh dari sensor kemudian dikirimkan secara otomatis ke server atau penyimpanan berbasis *cloud* melalui modul IoT.

Pencatatan data dilakukan secara periodik setiap 1 jam sekali selama masa penelitian berlangsung. Interval pencatatan ini dipilih untuk memperoleh variasi data yang cukup serta mampu merepresentasikan perubahan kondisi larutan nutrisi secara *real-time*.

3.1.2 Desain Sistem

Peneliti merancang sistem monitoring dan klasifikasi nutrisi melon hidroponik yang terdiri dari tiga subsistem utama: (1) subsistem perangkat keras (sensor & mikrokontroler), (2) subsistem komunikasi dan penyimpanan (protokol MQTT dan server/database), serta (3) subsistem perangkat lunak untuk pengolahan dan klasifikasi (*backend* untuk *preprocessing* & SVM dan antarmuka web untuk visualisasi). Desain sistem bertujuan memastikan data sensor (pH, TDS, DS18b20) dikirim secara periodik, tersimpan rapi, dan tersedia untuk proses *preprocessing* serta inferensi model SVM.



Gambar 3.2 Desain Sistem

Desain sistem yang digunakan dalam penelitian ini terdiri atas perangkat keras dan perangkat lunak yang saling terintegrasi. Pada sisi perangkat keras, peneliti menggunakan tiga jenis sensor, yaitu sensor pH untuk mengukur tingkat keasaman larutan nutrisi, sensor TDS untuk mengukur konsentrasi total padatan terlarut, serta sensor DS18b20 yang digunakan untuk memantau kondisi air pada sistem hidroponik. Ketiga sensor ini dihubungkan dengan mikrokontroler ESP32 yang berfungsi sebagai pengendali utama sekaligus pengirim data ke *server*.

Pada sisi perangkat lunak, peneliti menerapkan protokol komunikasi MQTT dengan broker *Mosquitto* sebagai penghubung antara perangkat IoT dan server. Data dari mikrokontroler dikirim dalam format JSON ke broker, kemudian diteruskan ke server *backend* yang bertugas sebagai *subscriber*. *Backend* menangani proses *preprocessing* data, seperti *cleaning*, normalisasi, *balancing*, dan reduksi dimensi dengan PCA, sebelum data diproses menggunakan algoritma *Support Vector Machine* (SVM) untuk klasifikasi kebutuhan nutrisi. Hasil klasifikasi kemudian disimpan pada database server lokal, dan ditampilkan melalui

dashboard web untuk memudahkan pengguna dalam memantau kondisi tanaman secara *real-time.*

Secara umum, alur kerja sistem dimulai dari sensor yang membaca parameter larutan, dilanjutkan dengan pengiriman data melalui mikrokontroler ke broker MQTT. Data tersebut diterima server, diproses melalui *pipeline preprocessing*, diklasifikasikan menggunakan SVM, lalu hasilnya ditampilkan di antarmuka web sebagai media pemantauan dan pengambilan keputusan.

Tabel 3.2 Spesifikasi *Hardware*

Komponen	Spesifikasi / Fungsi
Sensor pH	Mengukur tingkat keasaman larutan nutrisi hidroponik (range 0–14, akurasi ±0,1 pH).
Sensor TDS	Mengukur konsentrasi total padatan terlarut (ppm) sebagai indikator kadar nutrisi.
Sensor DS18b20	Digunakan untuk mengukur kondisi air pada system hidroponik (suhu air larutan nutrisi).
Mikrokontroler	ESP32, membaca data sensor, melakukan validasi awal, dan mengirim data ke server.
Catu daya	*Power supply* untuk mendukung operasi mikrokontroler dan sensor.

Tabel 3.1 menjelaskan komponen perangkat keras yang digunakan dalam penelitian ini. Sensor pH, TDS, dan DS menjadi perangkat utama yang berfungsi membaca parameter penting pada larutan nutrisi hidroponik. Data hasil pembacaan sensor tersebut diolah oleh mikrokontroler ESP32 yang kemudian mengirimkannya ke *server* melalui protokol komunikasi MQTT. Perangkat keras ini dipilih karena mampu memberikan pengukuran yang cukup akurat dan mudah diintegrasikan dalam sistem berbasis IoT.

Tabel 3.2 Spesifikasi *Software*

Komponen	Spesifikasi / Fungsi
Protokol komunikasi	MQTT (*Message Queuing Telemetry Transport*), digunakan untuk pertukaran data IoT.
Broker	*Mosquitto* MQTT broker sebagai pusat distribusi data sensor.
Server Backend	REST API / Subscriber MQTT, menangani preprocessing data, PCA, dan inferensi model SVM.
Database	PostgreSQL untuk penyimpanan data sensor dan hasil klasifikasi.

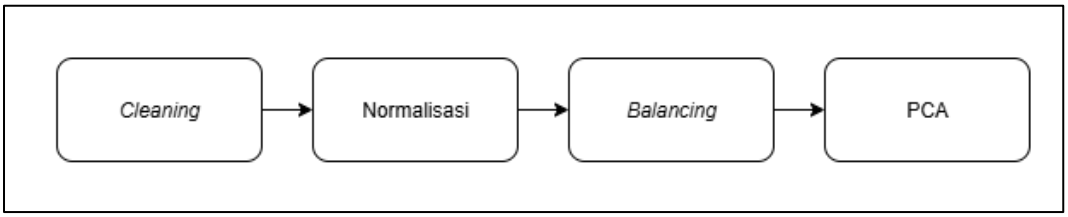
Tabel 3.2 Lanjutan

Komponen	Spesifikasi / Fungsi
Algoritma	*Support Vector Machine* (SVM) dengan *kernel linear* untuk klasifikasi kebutuhan nutrisi.
Antarmuka Web	*Dashboard* berbasis web untuk menampilkan grafik, histori data, dan hasil klasifikasi.

Sementara itu, Tabel 3.2 menunjukkan perangkat lunak dan infrastruktur yang digunakan. Broker MQTT *Mosquitto* menjadi penghubung komunikasi antar perangkat IoT dengan server backend. *Backend* bertugas menangani *preprocessing* data, melakukan reduksi dimensi dengan PCA, dan menjalankan algoritma klasifikasi *Support Vector Machine* (SVM). Data hasil klasifikasi disimpan dalam *database* server lokal, kemudian divisualisasikan melalui *dashboard* web yang memudahkan pengguna dalam melakukan *monitoring* kebutuhan nutrisi tanaman melon hidroponik secara *real-time*.

3.1.3 *Preprocessing* Data

Pada tahap ini, peneliti akan melakukan *preprocessing* data terlebih dahulu agar data siap digunakan dalam pengembangan model. Gambar 3.3 memperlihatkan tahapan-tahapan *preprocessing* yang dilakukan dalam penelitian ini, meliputi cleaning, normalisasi, balancing, dan PCA.



Gambar 3.3 Tahapan *preprocessing* data

Proses *preprocessing* ini sangat penting karena berfungsi untuk menghasilkan data yang bersih dan terstruktur, sehingga dapat mendukung proses pelatihan

maupun pengujian model secara optimal. Mengingat data merupakan aspek yang sangat krusial, tahapan *preprocessing* berpengaruh langsung terhadap kualitas hasil klasifikasi dan evaluasi performa model.

3.1.3.1 *Cleaning*

Tahap cleaning merupakan langkah awal dalam proses *preprocessing* yang bertujuan untuk memastikan kualitas data yang akan digunakan dalam penelitian. Data hasil pencatatan sensor sering kali mengandung nilai yang tidak valid, hilang (*missing values*), atau *outlier* yang dapat memengaruhi kinerja algoritma klasifikasi. Oleh karena itu, tahap ini akan dilakukan peneliti untuk mengidentifikasi dan memperbaiki ketidaksesuaian tersebut.

Peneliti melakukan proses *cleaning* dengan cara mengecek konsistensi data dari sensor pH, TDS, suhu air. Jika ditemukan data yang tidak tercatat atau berada jauh di luar batas normal, data tersebut akan ditangani dengan dua cara: mengganti nilai yang hilang menggunakan metode interpolasi linear yang diterapkan pada tahap *balancing* atau menghapus data yang benar-benar tidak dapat digunakan.

Tahapan cleaning ini penting karena kualitas data sangat menentukan performa model SVM yang digunakan dalam klasifikasi kebutuhan nutrisi tanaman melon hidroponik. Data yang bersih akan mengurangi risiko bias, meningkatkan akurasi prediksi, serta mendukung hasil evaluasi model yang lebih andal.

3.1.3.2 Normalisasi

Tahap normalisasi dilakukan untuk menyamakan skala antar fitur agar tidak ada parameter yang terlalu mendominasi dalam proses klasifikasi. Hal ini penting

karena data yang diperoleh memiliki satuan yang berbeda, seperti pH dalam rentang 0–14, TDS dalam satuan ppm, dan suhu air dalam °C. Jika tidak dilakukan normalisasi, algoritma SVM dapat memberikan bobot yang lebih besar pada fitur dengan nilai skala lebih tinggi, sehingga memengaruhi akurasi hasil klasifikasi.

Peneliti menggunakan metode *Min-Max Normalization*, yaitu metode yang mengubah nilai data ke dalam rentang tertentu, biasanya [0,1]. Rumus *Min-Max Normalization* dapat dituliskan sebagai berikut (Patel, H., & Prajapati, P, 2018):

$$X' = \frac{X - Xmin}{Xmax - Xmin} \tag{3.1}$$

Keterangan:
X : nilai asli data
X' : nilai hasil normalisasi
$Xmin$: nilai minimum pada fitur
$Xmax$: nilai maksimum pada fitur

Sebagai contoh, jika pada parameter suhu air tercatat nilai minimum 20°C dan maksimum 35°C, maka data suhu air 25°C akan dinormalisasi menjadi:

$$X' = \frac{25 - 20}{35 - 20} = \frac{5}{15} = 0.33$$

Dengan adanya normalisasi ini, seluruh fitur berada dalam skala yang sebanding, sehingga mendukung proses klasifikasi SVM agar bekerja lebih optimal.

3.1.3.3 *Balancing*

Tahap balancing pada penelitian ini menggunakan pendekatan rekonstruksi kelas berbasis *Equalized Class Reconstruction* (ECR). Metode ini dikembangkan untuk mengatasi ketidakseimbangan kelas tanpa harus membuang data pada kelas

mayoritas, sekaligus menghindari hilangnya informasi penting yang sebelumnya terjadi pada proses undersampling. Ketidakseimbangan kelas (*class imbalance*) muncul ketika distribusi jumlah sampel pada masing-masing kelas berbeda secara signifikan, misalnya kelas "cukup" memiliki jumlah data jauh lebih besar dibandingkan kelas "kurang" dan "berlebih". Kondisi ini berpotensi menyebabkan model SVM bias terhadap kelas yang dominan dan mengurangi akurasi pada kelas minoritas.

Dalam pendekatan ECR, penyeimbangan dilakukan bukan dengan memangkas data kelas mayoritas, tetapi dengan melakukan proses rekonstruksi sintetis (*synthetic reconstruction*) pada kelas yang memiliki jumlah data lebih sedikit. Proses rekonstruksi dilakukan dengan mengkombinasikan dua titik data secara linier untuk menghasilkan sampel baru yang masih mempertahankan karakteristik pola sensor. Rumus pembentukan sampel sintetis dapat dituliskan sebagai berikut (Bishop, 2006)

$$xsynthetic = \alpha \cdot xi + (1 - \alpha) \cdot xj \tag{3.2}$$

Keterangan :

xi	: data asli pertama dalam kelas yang sama
xj	: data asli kedua yang dipilih secara acak dalam kelas yang sama
$x$synthetic	: data sintetis hasil rekonstruksi
α	: koefisien penggabungan (nilai antara 0 dan 1).

Dengan menggunakan rumus tersebut, proses rekonstruksi dilakukan hingga jumlah data pada setiap kelas mencapai jumlah target, yaitu kelas dengan anggota terbanyak. Setelah proses ini selesai, dataset menjadi seimbang, dan model klasifikasi, dalam hal ini SVM, dapat belajar secara proporsional pada setiap kelas tanpa bias terhadap kelas mayoritas.

3.1.3.4 *Principal Component Analysis* (PCA)

Peneliti menggunakan metode *Principal Component Analysis* (PCA) untuk mereduksi dimensi data sebelum dilakukan proses klasifikasi. PCA dipilih karena mampu menyederhanakan representasi data tanpa banyak kehilangan informasi penting, sehingga model klasifikasi dapat bekerja lebih efisien. Proses PCA dalam penelitian ini dilakukan melalui beberapa langkah utama sebagai berikut.

Langkah awal yang dilakukan peneliti adalah standarisasi nilai dari seluruh fitur yang digunakan, yaitu pH, TDS, suhu air. Hal ini penting karena masing-masing fitur memiliki rentang nilai yang berbeda. Sebagai contoh, nilai TDS dapat mencapai ribuan ppm, sementara pH hanya berada pada rentang 0–14. Jika tidak dilakukan standarisasi, fitur dengan skala lebih besar akan lebih dominan dalam perhitungan. Standarisasi dilakukan dengan metode *min-max normalization* seperti yang telah dijelaskan pada poin 3.1.3.2, sehingga setiap fitur memiliki rata-rata 0 dan standar deviasi 1. Dengan standarisasi, semua fitur berada pada skala yang sama, sehingga memiliki kontribusi yang seimbang dalam analisis PCA.

Setelah standarisasi, peneliti membentuk matriks kovarians untuk melihat hubungan antar fitur. Matriks kovarians ini menunjukkan bagaimana dua fitur saling berkorelasi. Nilai kovarians positif menunjukkan bahwa dua fitur cenderung naik bersama, nilai negatif menunjukkan hubungan berlawanan, sementara nilai mendekati nol menunjukkan hampir tidak ada hubungan. Pada penelitian ini, dengan tiga fitur (pH, TDS, suhu air), matriks kovarians yang dihasilkan berbentuk 3x3, dengan diagonal utama berisi nilai varians masing-masing fitur, sedangkan elemen lainnya berisi kovarians antar fitur.

Tabel 3.4 Contoh Matriks Kovarians Data Sensor

Fitur	pH	TDS	Suhu
pH	Var(pH)	Cov(pH,TDS)	Cov(pH,Suhu)
TDS	Cov(TDS,pH)	Var(TDS)	Cov(TDS,Suhu)
Suhu	Cov(Suhu,pH)	Cov(Suhu,TDS)	Var(Suhu)

Langkah berikutnya adalah menghitung nilai eigen (*eigenvalue*) dan vektor eigen (*eigenvector*). Vektor eigen berfungsi sebagai arah atau sumbu baru pada data, sedangkan nilai eigen menunjukkan seberapa besar variasi data yang dapat dijelaskan oleh masing-masing vektor. Misalnya, jika *eigenvector* pertama banyak dipengaruhi oleh kombinasi TDS dan pH, maka arah ini menjadi sumbu baru yang paling penting. Semakin besar nilai eigen, semakin besar pula kontribusi vektor tersebut dalam menjelaskan variasi data.

Nilai eigen yang dihasilkan kemudian diurutkan dari yang terbesar hingga terkecil. Komponen dengan nilai eigen terbesar dipilih sebagai *principal component* karena mampu menjelaskan variasi data paling banyak. Jika peneliti ingin mereduksi data dari empat dimensi menjadi dua dimensi, maka dua *eigenvector* dengan nilai eigen terbesar yang dipilih. Proses transformasi dilakukan dengan mengalikan data yang telah distandarisasi dengan *eigenvector* terpilih untuk menghasilkan *principal component* (Jolliffe, I. T., & Cadima, J, 2016).

$$PC = Xstandar \cdot V \tag{3.3}$$

Keterangan:
PC : data baru hasil reduksi
$Xstandar$: data asli yang sudah distandarisasi
V : matriks *eigenvector* terpilih.

Hasil akhir dari proses PCA adalah data baru dalam bentuk *principal component* yang lebih ringkas namun tetap menyimpan variasi penting dari data

asli. Data hasil transformasi inilah yang digunakan peneliti sebagai masukan pada algoritma SVM untuk melakukan klasifikasi kebutuhan nutrisi tanaman melon hidroponik.

3.1.4 Implementasi *Support Vector Machine* (SVM)

Pada tahap implementasi, peneliti melatih model *Support Vector Machine* (SVM) dengan data yang telah melalui tahap praproses, termasuk normalisasi, reduksi dimensi dengan Principal Component Analysis (PCA), serta penyeimbangan data. Data hasil PCA kemudian digunakan untuk membentuk matriks kernel, yang merepresentasikan hubungan antar sampel berdasarkan kombinasi fitur penelitian, yaitu pH, TDS, suhu air.

Setelah kernel terbentuk, disusun matriks *Hessian* sebagai bagian dari proses optimisasi SVM. Optimisasi ini dilakukan menggunakan algoritma *Sequential Minimal Optimization* (SMO), yaitu metode efisien untuk memperbarui parameter *Lagrange multiplier* secara iteratif. Parameter ini menentukan sampel mana yang berperan sebagai *support vector*, sehingga model dapat membentuk *hyperplane* optimal untuk memisahkan kelas.

Nilai bias dan vektor bobot kemudian dihitung untuk membentuk fungsi keputusan SVM. Proses pelatihan dilakukan berulang hingga parameter model mencapai kondisi stabil. Model akhir terdiri atas *support vector*, vektor bobot, dan bias yang digunakan untuk mengklasifikasikan data baru.

Dalam penelitian ini, label kelas diperoleh dari tabel acuan nutrisi hidroponik yang telah dibahas pada Bab II poin 2.3 nutrisi hidroponik. Kategori “kurang”, “cukup”, dan “berlebih” ditetapkan berdasarkan rentang optimal pH,

suhu air, dan TDS pada fase vegetatif maupun berbunga tanaman melon. Hasil klasifikasi dari model SVM nantinya secara langsung merepresentasikan kondisi nutrisi tanaman sesuai baseline yang telah ditentukan.

3.2 Skenario Pengujian

Pengujian pada penelitian ini dilakukan untuk mengevaluasi kinerja model klasifikasi kebutuhan nutrisi tanaman melon hidroponik berbasis SVM dengan kernel linear. Skenario pengujian dirancang agar dapat menunjukkan sejauh mana model mampu mengenali pola dari data sensor dan mengklasifikasikan kondisi nutrisi tanaman ke dalam kategori tertentu (cukup, kurang, atau berlebih).

Pada tahap ini, peneliti membagi dataset hasil pengukuran pH, TDS, dan suhu air menjadi dua bagian, yaitu data latih (*training set*) dan data uji (*testing set*). Perbandingan rasio data latih dan uji dilakukan dalam beberapa skenario untuk melihat stabilitas model terhadap variasi pembagian data. Model kemudian dilatih menggunakan data latih dan diuji menggunakan data uji yang belum pernah dikenali sebelumnya.

Selain pembagian data, peneliti juga melakukan proses *hyperparameter tuning* untuk mencari parameter terbaik yang dapat meningkatkan kinerja SVM. Pada kernel linear, parameter utama yang diuji adalah nilai *regularization parameter* (C). Parameter ini berfungsi untuk mengontrol keseimbangan antara margin pemisah yang maksimal dan kesalahan klasifikasi pada data latih. Nilai C yang terlalu kecil dapat menghasilkan margin yang lebar tetapi rawan salah klasifikasi, sementara nilai C yang terlalu besar dapat membuat model terlalu kaku (*overfitting*). Oleh karena itu, beberapa nilai C akan diuji dengan metode pencarian

sistematis (*grid search*) serta validasi silang (*cross-validation*) untuk menentukan parameter terbaik.

Tabel 3.5 Skenario Pengujian

Skenario	Data Latih	Data Uji	Hyperparameter C	Tujuan
1	80%	20%	0.1, 1, 10, 100	Melihat performa model pada rasio umum yang banyak digunakan dalam penelitian.
2	70%	30%	0.1, 1, 10, 100	Menguji kinerja model dengan data uji yang lebih besar.
3	60%	40%	0.1, 1, 10, 100	Mengevaluasi ketahanan model saat data latih relatif lebih sedikit.

Hasil prediksi dari model akan dibandingkan dengan label aktual, kemudian dievaluasi menggunakan *confusion matrix* dihitung *accuracy*, *precision*, *recall*, dan *F1-score.*

Pengujian kedua dalam penelitian ini adalah pengujian kalibrasi sensor menggunakan rumus *Mean Absolute Error* (MAE) yang telah dijelaskan pada bab 2 di poin 2.11. Pengujian kalibrasi sensor dilakukan untuk memastikan bahwa data yang diperoleh dari sensor yang ada di sistem *Internet of Things* memiliki tingkat akurasi yang dapat dipertanggungjawabkan. Skenario pengujian dirancang dengan cara membandingkan hasil pembacaan sensor dengan alat ukur manual yang digunakan sebagai referensi. Parameter yang diuji meliputi pH larutan, suhu air, dan TDS (*Total Dissolved Solids*) karena ketiganya merupakan indikator utama dalam sistem hidroponik.

Pengujian dilakukan harian yang mewakili variasi kondisi nyata pada sistem hidroponik melon, baik pada fase vegetatif maupun berbunga. Setiap pengukuran dilakukan secara bersamaan antara sensor dan alat referensi untuk memastikan keseragaman kondisi uji. Data hasil pembacaan sensor dicatat, kemudian

dibandingkan dengan hasil pengukuran manual. Selisih antara keduanya dihitung sebagai nilai *error*.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pengumpulan Data

Pengumpulan data dilakukan selama periode 8 Agustus 2025 hingga 9 Oktober 2025 melalui sistem *Internet of Things* (IoT) yang telah dirancang pada penelitian ini. Data dikumpulkan secara otomatis dari sensor pH**,** TDS**,** dan DS18B20 (suhu air) yang terintegrasi dengan mikrokontroler ESP32. Seluruh hasil pembacaan sensor dikirimkan secara periodik ke server melalui protokol MQTT dan disimpan dalam basis data untuk keperluan analisis. Dataset yang diperoleh menjadi dasar pengembangan model klasifikasi kebutuhan nutrisi tanaman melon hidroponik berbasis algoritma *Support Vector Machine* (SVM)**.**

Secara keseluruhan, sistem berhasil merekam 1.498 baris data mentah selama masa pengamatan. Dataset tersebut terdiri atas empat fitur utama, yaitu pH larutan**,** TDS (*Total Dissolved Solids*)**,** suhu air, dan umur tanaman (Hari Setelah Tanam). Keempat fitur ini mencerminkan kondisi aktual larutan nutrisi pada dua fase pertumbuhan tanaman melon, yaitu fase vegetatif dan fase generatif. Data yang diperoleh mencakup berbagai variasi kondisi lingkungan yang terjadi selama penelitian berlangsung.

Periode pertama pengambilan data dilakukan pada 8 Agustus hingga 29 Agustus 2025, dengan total 528 baris data. Pada rentang waktu ini, sistem masih berada pada tahap penyesuaian awal dan proses kalibrasi sensor, sehingga ditemukan 238 nilai kosong (N/A) yang sebagian besar terjadi pada tanggal 11 hingga 20 Agustus 2025. Nilai kosong tersebut disebabkan oleh *error* dari

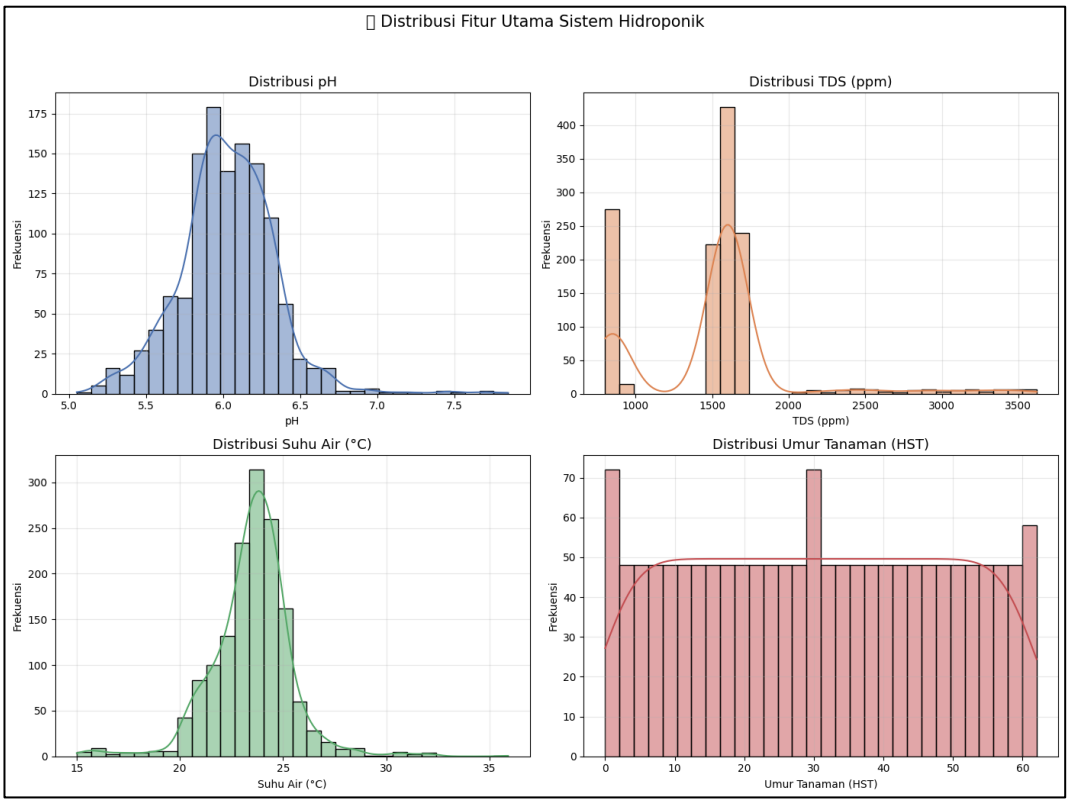
pembacaan sensor dan ketidakstabilan sistem pada fase uji coba awal. Meskipun demikian, data yang berhasil terekam tetap memberikan gambaran awal mengenai pola fluktuasi parameter larutan nutrisi selama fase vegetatif tanaman.

Periode kedua berlangsung pada 30 Agustus hingga 9 Oktober 2025, dengan total 720 baris data yang berhasil terekam secara lengkap dan stabil. Seluruh data pada periode ini bersifat numerik tanpa adanya nilai kosong, dengan rentang nilai TDS berada antara 1500 hingga 1700 ppm. Rentang tersebut menunjukkan bahwa larutan nutrisi berada pada kondisi normal dan sesuai dengan kisaran optimal bagi tanaman melon pada fase generatif awal. Selain itu, kestabilan data pada periode ini menunjukkan bahwa sistem IoT telah berfungsi dengan baik dalam mengirim dan menyimpan hasil pembacaan sensor secara konsisten.

Namun pada pembacaan di tanggal 6 Oktober hingga 9 Oktober 2025, dengan total 81 baris data. Pada rentang waktu ini, terjadi peningkatan nilai TDS hingga mencapai ≥2000 ppm, yang menunjukkan kondisi larutan nutrisi dalam keadaan “jebol” atau melebihi batas optimal. Fenomena ini menandakan adanya akumulasi padatan terlarut pada larutan nutrisi, yang berpotensi menyebabkan ketidakseimbangan ketersediaan unsur hara bagi tanaman. Meskipun demikian, data dari periode ini tetap dipertahankan dalam dataset karena menggambarkan kondisi ekstrem yang relevan untuk proses klasifikasi kebutuhan nutrisi tanaman.

Selanjutnya pada pengumpulan data, analisis distribusi dilakukan untuk memahami karakteristik penyebaran nilai pada masing-masing fitur hasil pembacaan sensor, meliputi pH, TDS, suhu air, dan umur tanaman. Tujuan analisis ini adalah untuk melihat pola kecenderungan data, kestabilan sensor, serta potensi

adanya nilai ekstrem (*outlier*) sebelum dilakukan tahap *preprocessing*. Visualisasi distribusi tiap fitur disajikan pada Gambar 4.1.



Gambar 4.1. Visualisasi Distribusi per Fitur

Berdasarkan Gambar 4.1, dapat diamati bahwa fitur pH memiliki distribusi yang mendekati normal dengan rata-rata 6,03 dan kisaran nilai antara 5,05 hingga 7,85. Sebagian besar nilai pH berada di sekitar rentang 5,8–6,4, yang menunjukkan kestabilan larutan pada kondisi ideal untuk pertumbuhan tanaman melon hidroponik. Nilai pH di bawah 5,5 dan di atas 7,0 muncul dalam jumlah kecil, mengindikasikan adanya fluktuasi sementara akibat proses penyesuaian nutrisi atau perubahan suhu lingkungan.

Fitur TDS (*Total Dissolved Solids*) menunjukkan dua puncak utama (*bimodal distribution*) dengan nilai rata-rata sebesar 1.510 ppm dan kisaran 800

hingga 3.619 ppm. Puncak pertama muncul di sekitar 1.000 ppm, yang menggambarkan kondisi pada fase vegetatif, sedangkan puncak kedua berada di kisaran 1.500–1.700 ppm yang merupakan fase generatif. Nilai di atas 2.000 ppm tergolong tinggi dan menandakan kondisi larutan "jebol" atau kelebihan nutrisi, sebagaimana terjadi pada periode 6–9 Oktober 2025. Pola distribusi ini menunjukkan bahwa sistem berhasil merekam perubahan konsentrasi nutrisi secara dinamis antar fase pertumbuhan.

Sementara itu, fitur suhu air (*water_temp_c)* memiliki distribusi yang relatif normal dengan rata-rata 23,5°C dan kisaran 15,0 hingga 35,9°C. Sebagian besar nilai berada di kisaran 22–25°C, yang termasuk rentang optimal untuk penyerapan unsur hara oleh akar tanaman. Nilai ekstrem di bawah 20°C dan di atas 30°C hanya muncul dalam jumlah kecil, kemungkinan besar disebabkan oleh variasi kondisi lingkungan eksternal seperti intensitas cahaya atau sirkulasi udara di sekitar sistem hidroponik.

Fitur terakhir, yaitu umur tanaman (Hari Setelah Tanam/HST), memiliki distribusi yang hampir merata dari 0 hingga 62 hari. Sebaran yang relatif seragam menunjukkan bahwa data pengukuran mencakup keseluruhan siklus pertumbuhan tanaman, mulai dari fase awal penanaman hingga menjelang panen. Hal ini penting karena memberikan variasi data yang cukup untuk pelatihan model klasifikasi, sehingga sistem dapat mengenali kebutuhan nutrisi pada berbagai tahap pertumbuhan.

Secara keseluruhan, hasil analisis distribusi fitur menunjukkan bahwa data yang dikumpulkan memiliki keragaman dan representasi yang baik terhadap

kondisi aktual sistem hidroponik. Pola distribusi pH dan suhu cenderung normal dan stabil, sedangkan TDS menunjukkan variasi yang lebih besar antar fase pertumbuhan. Variasi inilah yang menjadi dasar bagi sistem klasifikasi dalam mengidentifikasi kondisi nutrisi tanaman melon secara akurat.

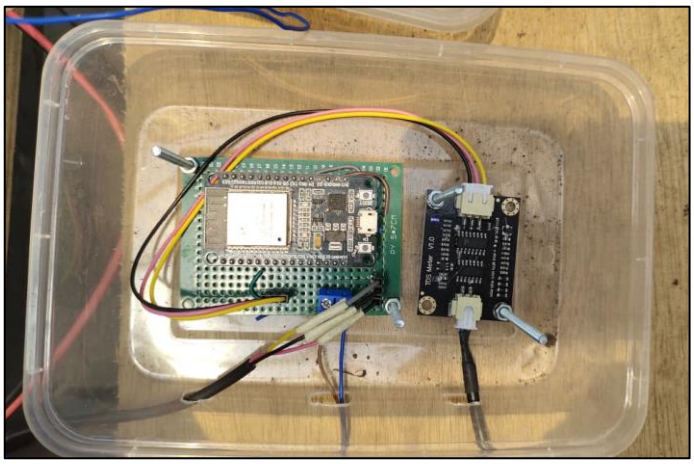
4.2 Sistem *Internet of Things* (IoT)

Sistem *Internet of Things* (IoT) yang telah direalisasikan pada penelitian ini berfungsi sebagai platform utama pengumpulan dan analisis data nutrisi tanaman melon hidroponik secara otomatis dan *real-time*. Implementasi sistem ini terdiri atas tiga lapisan utama, yaitu lapisan pengumpulan data, lapisan pemrosesan data, dan lapisan analisis hasil. Masing-masing lapisan memiliki peran spesifik dalam mendukung alur kerja sistem mulai dari akuisisi data sensor hingga penampilan hasil klasifikasi pada *dashboard* web.

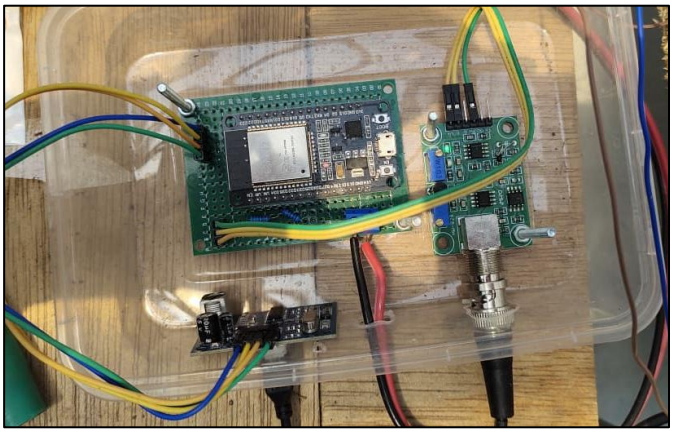
Pada lapisan pertama yang dimana merupakan bagian yang bertanggung jawab terhadap proses akuisisi data dari lingkungan hidroponik. Pada penelitian ini, perangkat yang digunakan terdiri atas mikrokontroler ESP32 yang terhubung dengan sensor pH, sensor TDS, dan sensor DS18B20. Sistem ini dirancang untuk membaca parameter kualitas larutan nutrisi secara berkala, kemudian mengirimkan data dalam format JSON *Payload* melalui jaringan Wi-Fi menggunakan protokol MQTT menuju server.

Hasil implementasi fisik dari perangkat ini ditunjukkan pada Gambar 4.2 dan Gambar 4.3, yaitu prototype sistem IoT yang digunakan selama proses pengujian. Rangkaian ini terdiri atas papan ESP32, sensor-sensor yang terhubung

melalui konektor digital dan analog, serta catu daya yang memungkinkan sistem beroperasi secara mandiri di lapangan.



Gambar 4.2 Rangkaian *Hardware* Sensor TDS & DS18b20

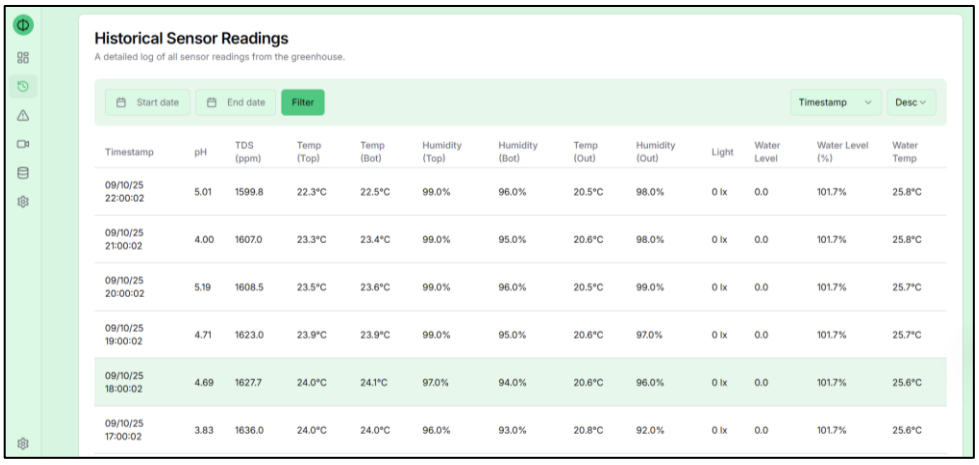


Gambar 4.3 Rangkaian *Hardware* Sensor pH

Prototype sistem tersebut kemudian dipasang di dalam greenhouse percobaan untuk melakukan pengukuran langsung pada larutan nutrisi hidroponik melon. Posisi alat diatur agar sensor pH dan TDS terendam dalam bak nutrisi, sedangkan sensor DS18B20 diletakkan pada aliran air untuk mengukur suhu aktual larutan. Implementasi penempatan alat di lapangan ditunjukkan pada Gambar 4.4 berikut.

Gambar 4.4 Posisi Penempatan *Hardware* di *Greenhouse*

Dari hasil implementasi lapangan tersebut, sistem mampu mengirimkan data sensor secara periodik dengan interval pembacaan yang stabil, serta mempertahankan koneksi MQTT selama proses pengamatan berlangsung. Pada Gambar 4.5, menunjukkan bahwa perangkat keras telah berfungsi sesuai rancangan dan mampu merekam kondisi nutrisi secara kontinu.



Historical Sensor Readings

A detailed log of all sensor readings from the greenhouse.

Start date | End date | Filter | Timestamp | Desc

Timestamp	pH	TDS (ppm)	Temp (Top)	Temp (Bot)	Humidity (Top)	Humidity (Bot)	Temp (Out)	Humidity (Out)	Light	Water Level	Water Level (%)	Water Temp
09/10/25 22:00:02	5.01	1599.8	22.3°C	22.5°C	99.0%	96.0%	20.5°C	98.0%	0 lx	0.0	101.7%	25.8°C
09/10/25 21:00:02	4.00	1607.0	23.3°C	23.4°C	99.0%	95.0%	20.6°C	98.0%	0 lx	0.0	101.7%	25.8°C
09/10/25 20:00:02	5.19	1608.5	23.5°C	23.6°C	99.0%	96.0%	20.5°C	99.0%	0 lx	0.0	101.7%	25.7°C
09/10/25 19:00:02	4.71	1623.0	23.9°C	23.9°C	99.0%	95.0%	20.6°C	97.0%	0 lx	0.0	101.7%	25.7°C
09/10/25 18:00:02	4.69	1627.7	24.0°C	24.1°C	97.0%	94.0%	20.6°C	96.0%	0 lx	0.0	101.7%	25.6°C
09/10/25 17:00:02	3.83	1636.0	24.0°C	24.0°C	96.0%	93.0%	20.8°C	92.0%	0 lx	0.0	101.7%	25.6°C

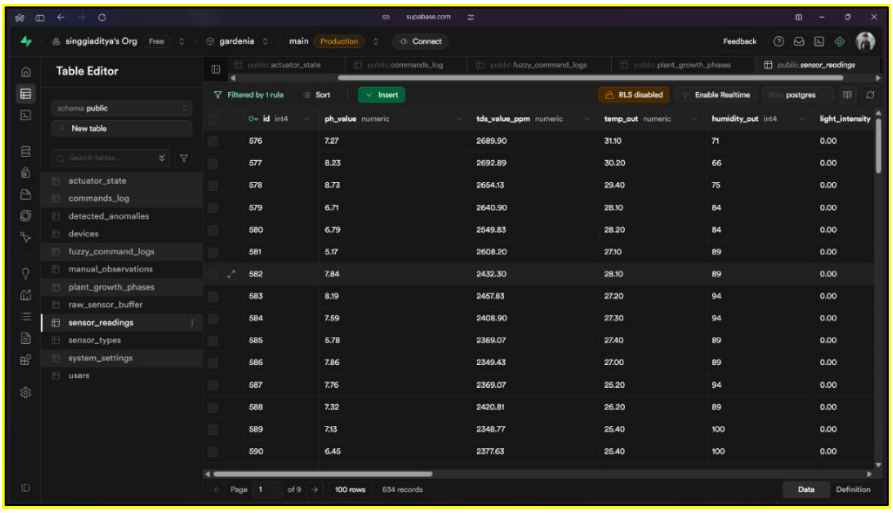
Gambar 4.5 Riwayat Pembacaan Sensor

Lapisan kedua berfungsi sebagai pusat pengelolaan dan analisis data yang dikirim dari perangkat IoT. Pada penelitian ini, digunakan server *backend* yang

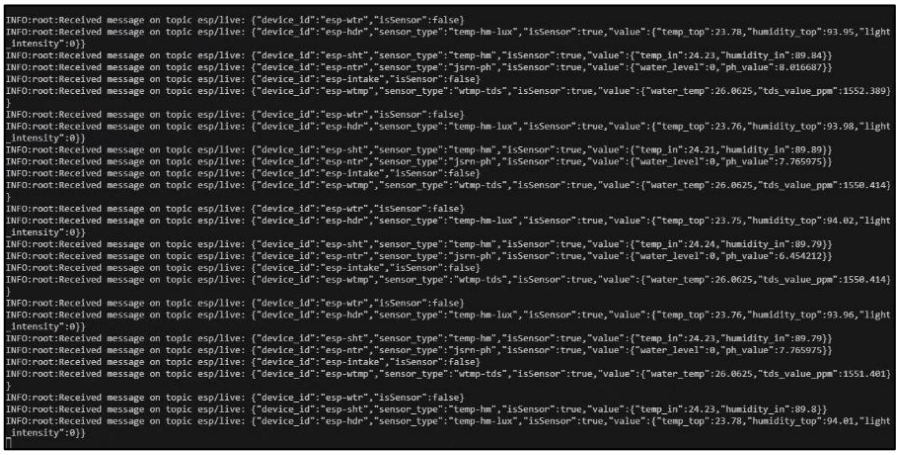
menjalankan proses penerimaan data, penyimpanan dalam basis data, serta *preprocessing* untuk keperluan klasifikasi menggunakan algoritma *Support Vector Machine* (SVM).

Data yang diterima dari broker MQTT secara otomatis disimpan dalam *database* server untuk kemudian melalui tahapan *labeling, cleaning, normalisasi, balancing,* dan *reduksi dimensi (PCA)*. Setelah data siap, model SVM digunakan untuk mengklasifikasikan kondisi nutrisi tanaman ke dalam tiga kategori, yaitu *kurang*, *cukup*, dan *berlebih*.

Hasil implementasi lapisan pemrosesan data ditunjukkan pada Gambar 4.6 dan Gambar 4.7 berikut, yang memperlihatkan tampilan server dan struktur basis data yang digunakan. Sistem backend ini berjalan pada lingkungan lokal yang telah dikonfigurasi agar dapat menerima data dari perangkat IoT secara langsung melalui protokol MQTT.



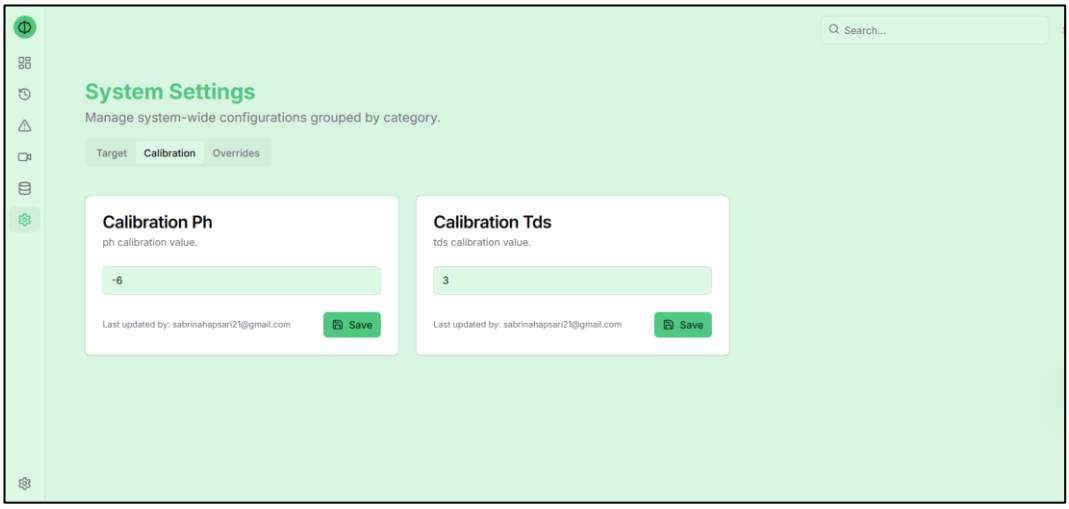
Gambar 4.6 *Interface Database*



Gambar 4.7 JSON *Payload* MQTT

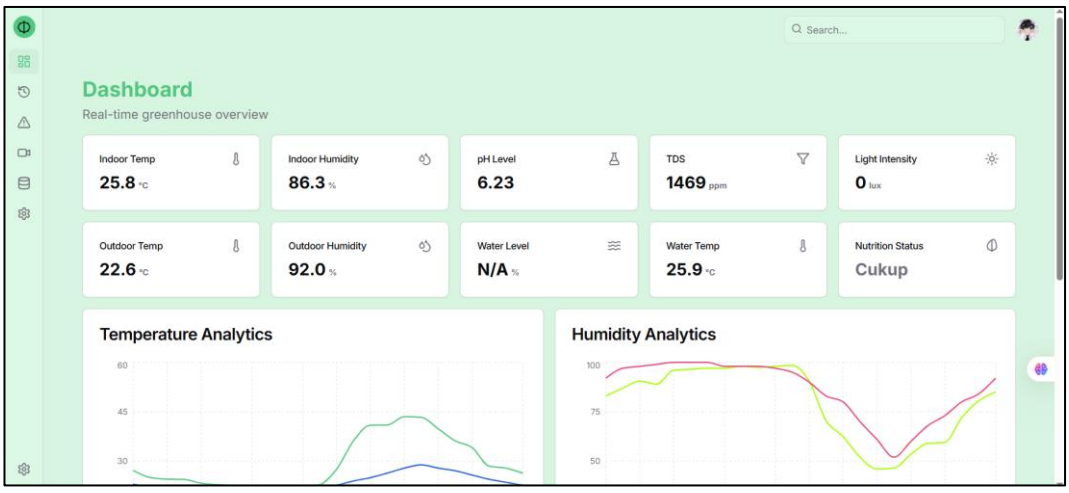
Lapisan terakhir merupakan bagian antarmuka pengguna (*user interface*) yang berfungsi untuk menampilkan hasil klasifikasi serta informasi sensor secara visual. Antarmuka ini dikembangkan dalam bentuk *dashboard* web interaktif, yang menampilkan data pH, TDS, suhu air, serta hasil klasifikasi nutrisi tanaman dalam bentuk indikator status.

Pada gambar 4.8, menunjukkan bahwa d*ashboard* ini juga dilengkapi dengan laman kalibrasi sensor, yang digunakan untuk menyesuaikan hasil pembacaan sensor terhadap alat ukur referensi. Fitur ini memastikan bahwa sistem tetap akurat dalam membaca nilai-nilai parameter lingkungan selama pengoperasian.



Gambar 4.8 Laman Kalibrasi Sensor

Implementasi antarmuka web ditunjukkan pada Gambar 4.9, yang memperlihatkan tampilan dashboard utama dan halaman kalibrasi sensor. Melalui dashboard ini, pengguna dapat memantau kondisi larutan nutrisi secara langsung, melihat riwayat data, serta mengidentifikasi perubahan status nutrisi berdasarkan hasil klasifikasi SVM.



Gambar 4.9 *Dashboard* Klasifikasi Nutrisi

Secara keseluruhan, hasil implementasi sistem *Internet of Things* (IoT) yang telah dikembangkan menunjukkan bahwa ketiga lapisan mulai dari pengumpulan

data, pemrosesan server, hingga visualisasi hasil klasifikasi telah berfungsi dengan baik dan saling terintegrasi. Sistem dapat melakukan pemantauan kualitas larutan nutrisi secara *real-time*, menyimpan data secara otomatis, dan menampilkan hasil analisis dengan akurat melalui *dashboard* web.

4.3 *Preprocessing* Data

4.3.1 *Labeling* Data

Tahap awal sebelum proses *preprocessing* adalah pelabelan data (labeling)**,** yang bertujuan untuk memberikan kategori kelas terhadap setiap sampel data hasil pembacaan sensor. Kategori ini digunakan sebagai acuan dalam proses pelatihan model klasifikasi kebutuhan nutrisi tanaman melon hidroponik berbasis *Support Vector Machine (SVM)*.

Pelabelan dilakukan berdasarkan pendekatan berbasis aturan (*rule-based classification*), dengan mempertimbangkan tiga parameter utama hasil pengukuran sensor, yaitu pH air (*ph*)**,** *Total Dissolved Solids* (*tds_ppm*)**,** dan suhu air (*water_temp_c*)**.** Masing-masing parameter memiliki rentang nilai ideal yang telah ditetapkan berdasarkan acuan kebutuhan nutrisi tanaman hidroponik sebagaimana dijelaskan pada Tabel 2.2 di Bab II. Nilai pH ideal berkisar antara 5,8–6,4, sedangkan nilai TDS optimal berada pada rentang 800–1700 ppm tergantung fase pertumbuhan tanaman. Adapun suhu air ideal berkisar pada 20–25°C, karena rentang tersebut mendukung penyerapan nutrisi secara optimal.

Proses pelabelan dilakukan dengan menerapkan aturan logis yang membandingkan nilai setiap parameter terhadap rentang ideal tersebut. Kriteria pelabelan ditetapkan sebagai berikut:

1. Apabila satu atau lebih parameter berada di bawah batas bawah rentang ideal, maka data diklasifikasikan sebagai “Kurang” (label 0), yang menunjukkan bahwa larutan nutrisi belum mencapai konsentrasi optimal.
2. Apabila seluruh parameter berada dalam rentang ideal, maka data diklasifikasikan sebagai “Cukup” (label 1), yang menandakan kondisi nutrisi berada pada tingkat optimal untuk pertumbuhan tanaman.
3. Apabila satu atau lebih parameter melebihi batas atas rentang ideal, maka data diklasifikasikan sebagai “Berlebih” (label 2), yang mengindikasikan larutan terlalu pekat atau suhu air terlalu tinggi.

Secara implementatif, proses *labeling* dilakukan menggunakan fungsi pemetaan berbasis kondisi logis (*conditional mapping*) pada *pseudocode* 4.1 berikut:

Pseudocode 4.1

```
Algoritma Pelabelan_Otomatis_Nutrisi
Input  : Dataset dengan kolom pH, TDS (ppm), dan suhu air (°C)
Output : Kolom nutrisi_label dengan nilai (0 = Kurang, 1 =
Cukup, 2 = Berlebih)
Mulai
    1. Konversi kolom pH, TDS, dan suhu air menjadi numerik.
    2. Untuk setiap baris data pada dataset:
        a. Ambil nilai ph, tds, dan suhu.
        b. Jika salah satu bernilai kosong (NaN), maka:
              Label ← NaN
        c. Jika 5.8 ≤ ph ≤ 6.4 dan
             (800 ≤ tds ≤ 1000 atau 1500 ≤ tds ≤ 1700) dan
             20 ≤ suhu ≤ 25 maka:
              Label ← 1   // Cukup
        d. Jika ph < 5.8 atau tds < 800 atau suhu < 20 maka:
              Label ← 0   // Kurang
        e. Jika ph > 6.4 atau tds > 1700 atau suhu > 25 maka:
              Label ← 2   // Berlebih
        f. Jika tidak memenuhi semua kondisi di atas:
              Label ← NaN
    3. Simpan label hasil ke kolom baru “nutrisi_label”.
    4. Hitung jumlah distribusi tiap label (0, 1, 2).
    5. Visualisasikan distribusi label dalam bentuk grafik
batang.
Selesai
```

Hasil dari proses labeling menunjukkan bahwa data terbagi ke dalam tiga kategori utama, yaitu kurang (0), cukup (1), dan berlebih (2), serta sejumlah data yang tidak memiliki label akibat nilai kosong atau *missing value*. Distribusi label awal ditunjukkan pada Tabel 4.1 berikut:

Tabel 4.1 Distribusi Label

Kelas	**Kategori**	**Jumlah Data**	**Persentase (%)**
0	Kurang	220	14.7
1	Cukup	870	58.2
2	Berlebih	131	8.7
-	Tidak Terlabel (NaN)	277	18.5
Total	**–**	**1.498**	**100.0**

Berdasarkan Tabel 4.1, dapat diketahui bahwa sebagian besar data berada pada kategori cukup dengan persentase sekitar 58,2% dari total dataset. Hal ini menunjukkan bahwa kondisi larutan nutrisi selama masa pengamatan berada pada kisaran ideal yang mendukung pertumbuhan tanaman. Sementara itu, kategori kurang dan berlebih memiliki proporsi yang lebih kecil, masing-masing sebesar 14,7% dan 8,7%, yang menunjukkan variasi kondisi nutrisi di luar batas optimal. Adapun 18,5% data tidak memiliki label karena adanya nilai kosong (NaN) pada parameter pH atau TDS, yang kemudian ditangani pada tahap *cleaning* selanjutnya.

Pendekatan *labeling* ini dinilai efektif karena mampu merepresentasikan kondisi aktual sistem nutrisi hidroponik secara menyeluruh dengan mempertimbangkan beberapa faktor lingkungan sekaligus. Selain itu, hasil distribusi label yang tidak seimbang juga memberikan dasar bagi penerapan metode *balancing* pada tahap *preprocessing* berikutnya agar model klasifikasi dapat belajar secara proporsional terhadap seluruh kelas.

4.3.2 *Cleaning*

Setelah proses labeling selesai dilakukan, tahap berikutnya adalah pembersihan data **(***data cleaning***)**. Tahap ini bertujuan untuk memastikan bahwa dataset yang digunakan dalam proses pelatihan model berada dalam kondisi bersih, konsisten, dan bebas dari kesalahan pencatatan. Proses *cleaning* dilakukan melalui beberapa langkah utama, yaitu menghapus data duplikat, mengatasi data ekstrem dan *noise* (*outlier*), serta menstandarkan penamaan kolom agar sesuai dengan kebutuhan proses komputasi.

Langkah pertama adalah pemeriksaan dan penghapusan data duplikat. Berdasarkan hasil pemeriksaan, tidak ditemukan adanya baris data yang identik atau berulang sehingga jumlah duplikat yang dihapus adalah 0 baris. Langkah ini penting dilakukan untuk mencegah terjadinya bias pada distribusi data dan menjaga integritas hasil analisis.

Langkah kedua adalah penanganan data ekstrem dan *noise* (*outlier*)**,** khususnya pada fitur pH**,** TDS, suhu air dan label nutrisi. Sebelum dilakukan pembersihan, jumlah nilai ekstrem tercatat sebagai berikut:

Tabel 4.2 *Outlier / Noise* Dataset

Kolom	Jumlah Data Ekstrem
ph	272
tds_ppm	238
water_temp_c	0
nutrisi_label	277

Data tersebut umumnya terjadi akibat gangguan pembacaan sensor atau keterlambatan pengiriman data pada periode tertentu. Untuk mengatasinya, digunakan metode interpolasi linier yang diterapkan pada tahap *balancing* untuk

menjaga keseimbangan data pada fitur pH, TDS, suhu air (*water_temp_c*). Metode ini menghitung nilai pengganti berdasarkan antar pasangan sampel acak pada kelas yang sama, sehingga pola perubahan data tetap terjaga secara alami tanpa menimbulkan distorsi signifikan. Sedangkan data *outlier* dan *noise* di *drop* seluruhnya untuk menjaga integritas data sebelum diproses pada tahap selanjutnya.

Langkah ketiga adalah standarisasi penamaan kolom agar seragam dan mudah diakses pada tahap komputasi berikutnya. Penamaan kolom disesuaikan menjadi huruf kecil tanpa spasi, seperti ph, *tds_ppm*, dan *water_temp_c*, untuk menjaga konsistensi sintaksis dalam implementasi kode program.

Hasil akhir dari tahap *cleaning* menunjukkan bahwa seluruh data ekstrem dan noise telah berhasil ditangani. Setelah seluruh proses pembersihan selesai dilakukan, jumlah data yang tersisa adalah 1.221 baris dari total 1.498 data mentah. Dengan demikian, dataset yang digunakan untuk tahap *preprocessing* berikutnya telah bebas dari data ekstrem dan *noise* serta siap untuk dilakukan proses normalisasi data.

4.3.3 Normalisasi

Pada tahap normalisasi, tiga fitur utama yang dinormalisasi adalah pH**,** TDS (*Total Dissolved Solids*), dan suhu air. Proses normalisasi dilakukan menggunakan *library scikit-learn* melalui fungsi *MinMaxScaler()* pada *pseudocode* 4.2 berikut:

Pseudocode 4.2

```
Algoritma Normalisasi_Data_MinMax
Input  : Dataset dengan kolom ph, tds_ppm, water_temp_c
Output : Dataset dengan nilai fitur ternormalisasi pada rentang
[0, 1]
Mulai
    1. Tentukan daftar kolom fitur yang akan dinormalisasi:
```

```
          fitur ← [ph, tds_ppm, water_temp_c]
    2. Buat salinan dataset untuk menyimpan hasil normalisasi.
    3. Untuk setiap fitur pada daftar fitur:
          a. Hitung nilai minimum (min) dan maksimum (max) fitur.
          b. Lakukan normalisasi setiap nilai menggunakan rumus:

             nilai_normalisasi = (nilai_asli - min) / (max - min)
    4. Simpan hasil normalisasi ke dataset baru (df_scaled).
    5. Simpan parameter min dan max untuk tiap fitur agar dapat
digunakan pada proses prediksi berikutnya.
    6. Tampilkan nilai minimum dan maksimum tiap fitur sebelum
proses normalisasi.
    7. Tampilkan 5 data pertama hasil normalisasi.
Selesai.
```

Hasil proses normalisasi menunjukkan bahwa seluruh fitur berhasil ditransformasikan ke dalam rentang nilai 0–1. Nilai minimum dan maksimum setiap fitur sebelum proses *scaling* ditunjukkan pada Tabel 4.3 berikut.

Tabel 4.3 Nilai Min-Max per Fitur

Fitur	Nilai Minimum	Nilai Maksimum
Ph	5.05	7.85
TDS (ppm)	800.50	3,619.20
Suhu Air (°C)	15.00	30.80

Setelah proses normalisasi, nilai setiap fitur diubah secara proporsional sesuai rentang skala baru. Lima data pertama hasil normalisasi ditunjukkan pada Tabel 4.4 berikut.

Tabel 4.4 Data Hasil Normalisasi

Timestamp	pH	TDS (scaled)	Suhu Air (scaled)	Label
08/08/2025 00:00	0.2929	0.0053	0.6139	1
08/08/2025 01:00	0.3464	0.0203	0.3987	1
08/08/2025 02:00	0.3857	0.0214	0.6203	1
08/08/2025 03:00	0.4107	0.0149	0.5696	1
08/08/2025 04:00	0.6143	0.0259	0.9810	2

Dari hasil tersebut dapat disimpulkan bahwa seluruh nilai telah berada dalam skala yang seragam, tanpa mengubah pola relatif antar nilai asli. Proses ini memastikan bahwa tidak ada fitur yang memiliki pengaruh dominan hanya karena

perbedaan satuan atau rentang nilai. Dengan demikian, dataset hasil normalisasi siap digunakan pada tahap *balancing* dan pelatihan model klasifikasi.

4.3.4 *Balancing*

Pada tahap ini dilakukan proses *Equalized Class Reconstruction* untuk menghasilkan distribusi data yang seimbang pada seluruh kelas label nutrisi. Pertama, dataset hasil normalisasi disalin ke variabel `df_bal`, kemudian dihitung distribusi awal jumlah sampel per kelas. Nilai jumlah sampel terbesar digunakan sebagai *target* agar seluruh kelas memiliki jumlah sampel identik setelah balancing.

Selanjutnya, dataset dipisah berdasarkan label, lalu diproses secara individual. Untuk kelas yang jumlah datanya di bawah target, dilakukan rekonstruksi data sintetis melalui interpolasi linier antar pasangan sampel acak pada kelas tersebut. Formula interpolasi digunakan untuk menghasilkan sampel baru yang secara statistik berada di tengah ruang distribusi kelas, sehingga tetap merepresentasikan pola sensorik aslinya. Berikut pseudocode yang menjelaskan proses *Equalized Class Reconstruction* (ECR) dan *balancing* :

Pseudocode 4.3

```
Algoritma Balancing_Data_ECR
```

```
Input  : Dataset hasil normalisasi (df_scaled) berisi fitur [ph,
tds_ppm, water_temp_c] dan label nutrisi_label
Output : Dataset seimbang (df_re) dengan jumlah sampel identik
pada tiap kelas
```

```
Mulai
    1. df_bal ← salin dataset hasil normalisasi
    2. Hitung distribusi kelas awal pada df_bal
    3. Tentukan target ← jumlah sampel terbanyak (kelas
mayoritas)
    4. Buat list kosong df_final
    5. Untuk setiap label dalam nutrisi_label lakukan:
         a. group ← subset data dengan label tersebut
         b. n ← jumlah data dalam group
         c. Jika n < target:
                needed ← target - n
```

```
                Pilih pasangan indeks acak idx1 dan idx2
sebanyak needed
                Buat data sintetis:
                    synth ← 0.5 * group[idx1] + 0.5 *group[idx2]
            Tambahkan label ke setiap baris synthetic
                df_new ← gabungkan group + synthetic
            Jika n > target:
                df_new ← sample acak group sebanyak target
        d. Tambahkan df_new ke df_final
    6. df_re ← gabungkan semua elemen dalam df_final
    7. Lakukan shuffle pada df_re untuk menghapus pola
pengurutan label
    8. Kembalikan df_re
Selesai
```

Perbandingan jumlah data sebelum dan sesudah dilakukan *balancing* ditunjukkan pada Tabel 4.5 berikut.

Tabel 4.5 Perbandingan Data Sebelum dan Setelah *Balancing*

Kelas	**Kategori**	**Jumlah Data Sebelum**	**Jumlah Data Sesudah**	**Perubahan**
0	Kurang	220	870	+ 650
1	Cukup	870	870	–
2	Berlebih	131	870	+ 739
Total	–	**1.221**	2.610	**–**

Tabel di atas menunjukkan bahwa seluruh kelas telah disamakan jumlahnya menjadi 870 sampel per kelas, sehingga total data akhir yang digunakan untuk pelatihan model klasifikasi adalah 2.610 sampel. Penambahan data (*over-sampling*) terjadi pada kelas (*Kurang*) dan kelas *Berlebih*, sedangkan kelas *Cukup* tetap dipertahankan sebagai acuan keseimbangan.

Setelah seluruh kelas selesai diproses, hasilnya digabungkan ke dalam satu dataset baru dan dilakukan proses *shuffle* untuk memastikan tidak ada pengelompokan data berdasarkan label. Dataset akhir ini kemudian digunakan sebagai masukan pada tahap PCA dan pelatihan model SVM.

4.3.5 *Principal Component Analysis* (PCA)

Proses PCA dilakukan terhadap data hasil *balancing* menggunakan tiga fitur utama: *ph, tds_ppm*, dan *water_temp_c*. Jumlah komponen yang digunakan adalah dua komponen utama (PC1 dan PC2), dengan implementasi menggunakan pustaka *scikit-learn* pada *pseudocode* 4.4 berikut:

Pseudocode 4.4

```
Algoritma Reduksi_Dimensi_PCA
Input   : Dataset hasil balancing dengan fitur (ph, tds_ppm, water_temp_c) dan label nutrisi_label
Output : Dataset hasil reduksi dimensi dengan dua komponen utama (PC1, PC2)
Mulai
    1. Pisahkan data menjadi:
X ← [ph, tds_ppm, water_temp_c]
y ← [nutrisi_label]
    2. Tentukan jumlah komponen utama yang akan digunakan:
         jumlah_komponen ← 2
    3. Buat objek PCA dengan jumlah_komponen.
    4. Terapkan PCA pada data fitur:
         X_pca ← hasil transformasi PCA terhadap X
    5. Simpan hasil transformasi ke dataset baru (df_pca) dengan kolom:
         [PC1, PC2, nutrisi_label]
    6. Hitung proporsi variansi yang dijelaskan oleh setiap komponen utama:
         a. PC1_var ← proporsi variansi komponen 1
         b. PC2_var ← proporsi variansi komponen 2
    7. Hitung total variansi yang dijelaskan oleh seluruh komponen.
    8. Simpan nilai *component loadings* untuk melihat kontribusi setiap fitur terhadap komponen utama.
    9. Tampilkan hasil proporsi variansi dan *component loadings*.
    10. Tampilkan lima data pertama hasil reduksi PCA.
Selesai
```

Hasil analisis menunjukkan bahwa dua komponen utama yang terbentuk mampu menjelaskan 87,85% dari total variansi data, dengan rincian proporsi variansi untuk masing-masing komponen seperti ditunjukkan pada Tabel 4.6 berikut.

Tabel 4.6 Proporsi Variansi Komponen

Komponen	Proporsi Variansi	Keterangan
PC1	0.5567	Menjelaskan variansi terbesar, dominan pada fitur TDS
PC2	0.3218	Menjelaskan variansi tambahan, dominan pada fitur pH dan suhu air
Total	0.8785	**87,85% total variansi data terjelaskan**

Berdasarkan nilai *loading components* yang dihasilkan, terlihat bahwa komponen utama pertama (PC1) memiliki kontribusi terbesar dari fitur TDS (0.945227), sedangkan komponen kedua (PC2) dipengaruhi kuat oleh fitur pH (0.884727) dan suhu air (0.346572). Matriks *component loadings* dapat dilihat pada Tabel 4.7 berikut.

Tabel 4.7 Matriks *Component Loadings*

Komponen	pH	TDS	Suhu Air
PC1	0.251480	**0.945227**	0.208095
PC2	**0.884727**	-0.311683	**0.346572**

Dari hasil tersebut dapat disimpulkan bahwa TDS merupakan fitur dengan pengaruh paling dominan terhadap variasi data secara keseluruhan, sedangkan pH dan suhu air memberikan kontribusi tambahan terhadap komponen kedua. Dengan total varian terjelaskan sebesar 87,85%, dua komponen utama ini sudah cukup representatif untuk menggambarkan karakteristik data sensor secara keseluruhan.

Selain itu, hasil transformasi PCA juga digunakan untuk visualisasi distribusi data antar kelas nutrisi, yang menunjukkan bahwa masing-masing kelas (kurang, cukup, dan berlebih) memiliki kecenderungan pola yang berbeda pada ruang dua dimensi PC1–PC2. Hal ini menunjukkan bahwa data hasil *balancing* memiliki separabilitas yang baik dan layak digunakan untuk tahap pelatihan model klasifikasi pada subbab berikutnya.

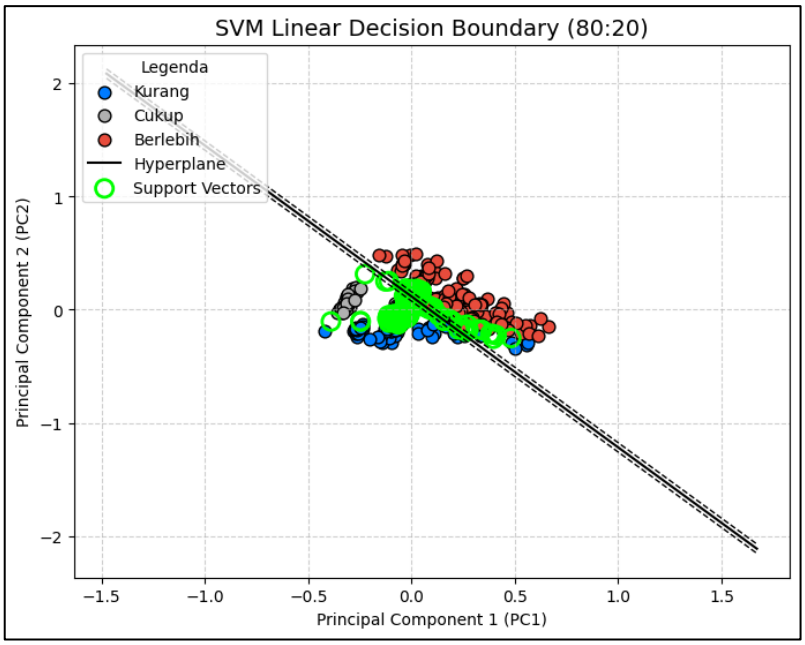
4.4 Implementasi *Support Vector Machine* (SVM)

Implementasi model klasifikasi dilakukan menggunakan algoritma *Support Vector Machine* (SVM) dengan kernel linear, yang secara bawaan menerapkan pendekatan *One-vs-Rest* (OvR) untuk menangani kasus multi-kelas. Pada pendekatan ini, SVM membentuk tiga *hyperplane* secara internal yang masing-masing memisahkan satu kelas terhadap dua kelas lainnya (*Berlebih vs lainnya*, *Cukup vs lainnya*, dan *Kurang vs lainnya*). Setiap titik data kemudian dievaluasi terhadap ketiga *hyperplane* tersebut, dan kelas dengan nilai *decision function* tertinggi ditetapkan sebagai hasil prediksi akhir. Mekanisme ini memungkinkan model linear untuk memetakan data hasil reduksi *Principal Component Analysis* (PCA) ke dalam bidang dua dimensi (PC1–PC2) dengan margin optimal antar kelas.

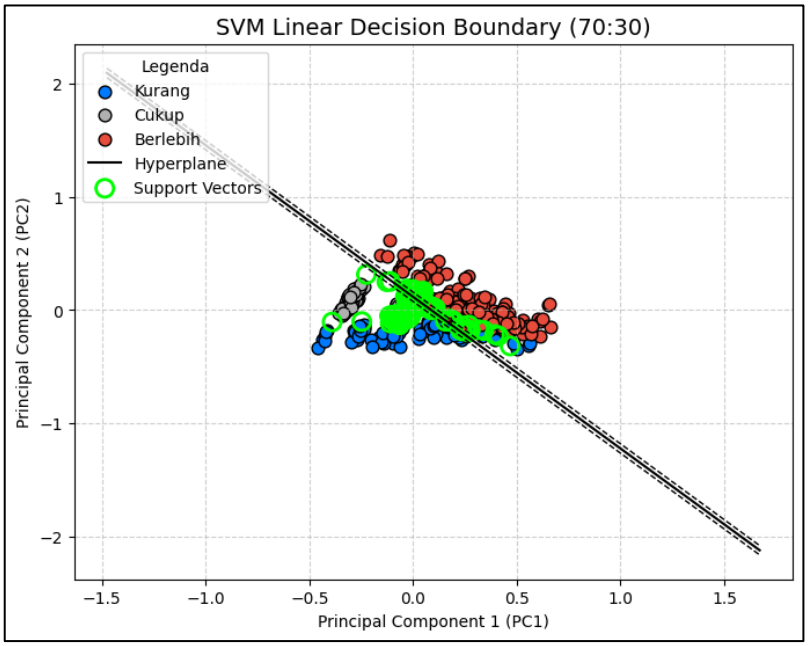
Dalam penelitian ini, tiga model SVM linear dibangun dengan perbedaan proporsi data latih dan uji, yaitu 80:20, 70:30, dan 60:40 seperti yang sudah dijelaskan pada Tabel 3.5 Skenario Pengujian. Setiap model menjalani proses pelatihan dan tuning parameter menggunakan metode *Grid Search* dengan *5-Fold Cross Validation* untuk memperoleh nilai parameter optimal, khususnya pada variabel regularisasi (*C*). Model yang diperoleh dari setiap konfigurasi dievaluasi terhadap data uji untuk mengukur akurasi dan stabilitas klasifikasi terhadap variasi proporsi data pelatihan.

Visualisasi hasil pelatihan masing-masing model ditunjukkan pada Gambar 4.10, 4.11, dan 4.12, yang menggambarkan posisi *hyperplane* dan margin pemisah antar kelas pada bidang PC1–PC2. Garis hitam merepresentasikan batas pemisah

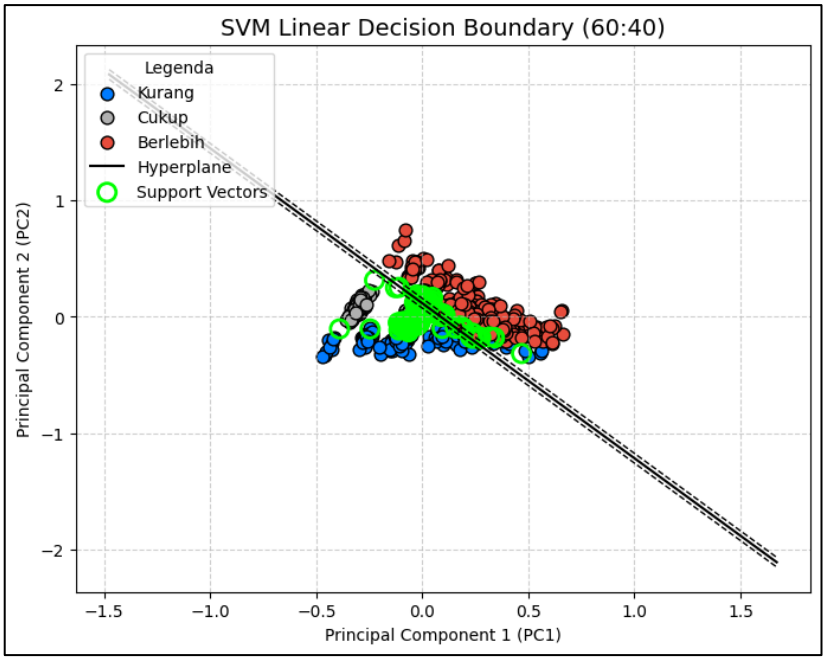
linear yang dihasilkan SVM, sedangkan titik berwarna menunjukkan distribusi data hasil PCA. Titik berwarna hijau dengan garis tepi menandakan *support vectors* yang digunakan model untuk menentukan posisi margin optimal. Pola sebaran menunjukkan bahwa setiap model menghasilkan *decision boundary* yang konsisten, dengan sedikit variasi pada distribusi margin akibat perubahan proporsi data latih dan uji.



Gambar 4.10 Visualisasi *Hyperplane* Data *Train-Test* 80:20



Gambar 4.11 Visualisasi *Hyperplane* Data *Train-Test* 70:30



Gambar 4.12 Visualisasi *Hyperplane* Data *Train-Test* 60:40

Hasil tersebut menunjukkan bahwa pendekatan linear SVM dengan skema OvR mampu mengklasifikasikan data nutrisi tanaman menjadi tiga kategori secara efektif. Meskipun terdapat 61umpeng tindih margin antar kelas, terutama antara

kelas *Cukup* dan *Kurang*, struktur pemisahan tetap terbentuk jelas pada ruang dua dimensi hasil PCA. Secara keseluruhan, implementasi ketiga model menunjukkan performa yang stabil dan adaptif terhadap variasi ukuran data pelatihan.

4.5 Pengujian Model

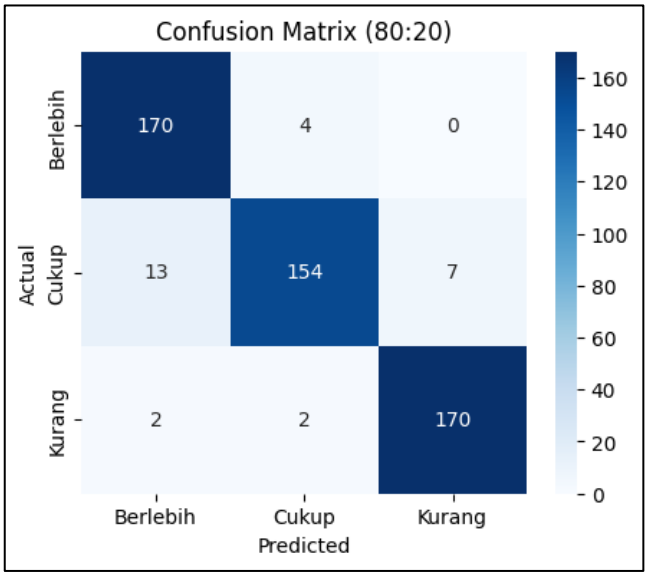
Pengujian model dilakukan terhadap tiga variasi rasio pembagian data latih dan uji, yaitu 80:20, 70:30, dan 60:40. Setiap model menjalani proses pelatihan dengan *Grid Search Cross Validation* untuk menemukan parameter terbaik. Masing-masing konfigurasi pembagian data latih dan uji juga melalui pengujian c*onfusion matrix* untuk menghitung *accuracy, precision, recall,* dan *F1-score*.

Pada konfigurasi pertama, sebanyak 80% data digunakan untuk pelatihan dan 20% untuk pengujian. Model memperoleh parameter terbaik pada C = 100 dengan kernel linear, serta skor *cross-validation* sebesar 0,9478. Hasil pengujian pada data uji menunjukkan akurasi sebesar 94,64%.

Tabel 4.8 Pengujian *Hyperparameter Tuning* Model SVM Rasio 80:20

Parameter Terbaik	**Akurasi (%)**	***Cross-Validation Score***	**Jumlah *Support Vector***
C = 100, Linear	94,64	0.9478	340

Visualisasi *Confusion Matrix* dan kinerja model secara per kelas ditunjukkan pada gambar dan tabel berikut.



Gambar 4.13 *Confusion Matrix* Model SVM Rasio 80:20

Tabel 4.9 Metrik Evaluasi Model SVM Rasio 80:20

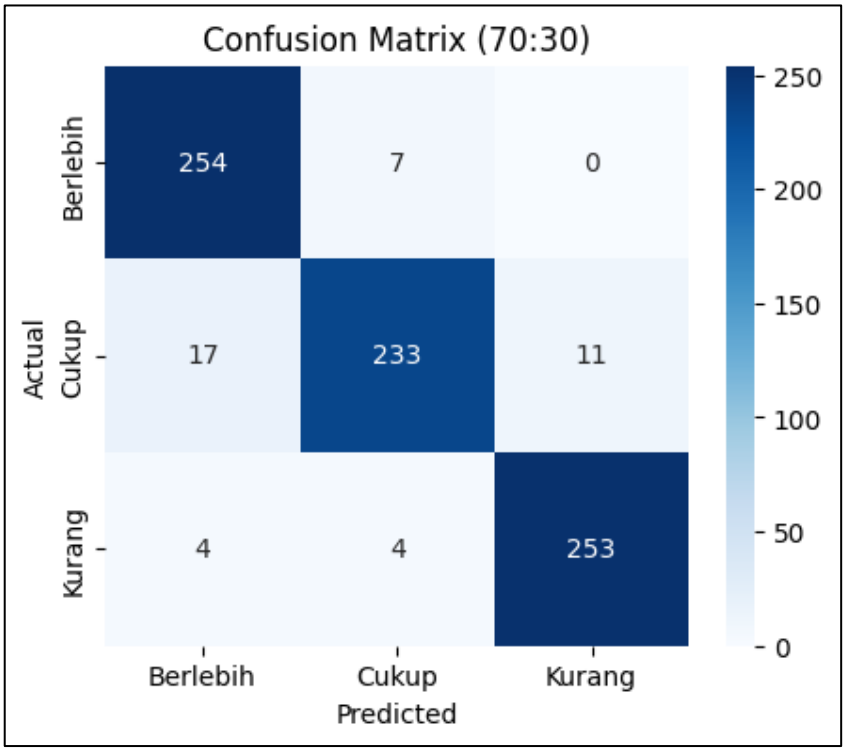
Kelas	***Precision***	***Recall***	***F1-Score***	**Jumlah Data**
Berlebih	0.92	0.98	0.95	174
Cukup	0.96	0.89	0.92	174
Kurang	0.96	0.98	0.97	174
Rata-rata	0.95	0.95	0.95	—

Hasil menunjukkan bahwa model mampu membedakan ketiga kelas dengan baik. Nilai *precision* tertinggi diperoleh pada kelas Berlebih, sedangkan penurunan kecil pada kelas Kurang menandakan sebagian data nutrisi rendah masih memiliki kemiripan nilai dengan kelas Cukup.

Pada konfigurasi kedua, model menggunakan 70% data untuk pelatihan dan 30% untuk pengujian. Parameter terbaik diperoleh dengan C = 100 dan kernel linear, dengan skor *cross-validation* sebesar 0.9486. Hasil pengujian menunjukkan akurasi 94,51%, yang relatif stabil dibandingkan rasio sebelumnya.

Tabel 4.10 Pengujian *Hyperparameter Tuning* Model SVM Rasio 70:30

Parameter Terbaik	**Akurasi (%)**	***Cross-Validation* Score**	**Jumlah *Support Vector***
C = 100, Linear	94,51%	0.9486	305



Gambar 4.14 *Confusion Matrix* Model SVM Rasio 70:30

Tabel 4.11 Metrik Evaluasi Model SVM Rasio 70:30

Kelas	***Precision***	***Recall***	***F1-Score***	**Jumlah Data**
Berlebih	0.92	0.97	0.95	261
Cukup	0.95	0.89	0.92	261
Kurang	0.96	0.97	0.96	261
Rata-rata	0.95	0.95	0.94	—

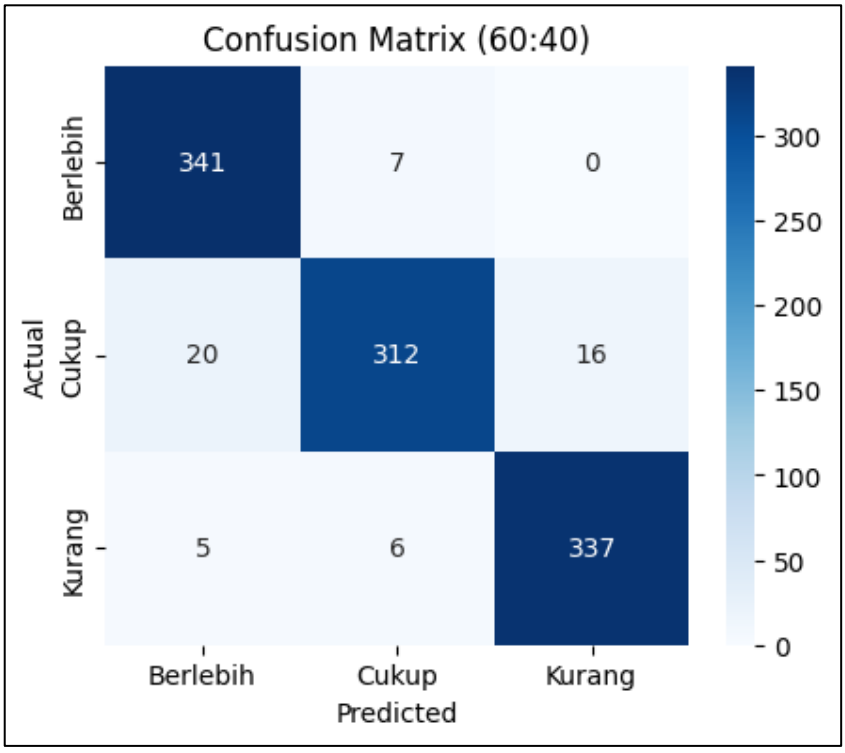
Model tetap menunjukkan konsistensi performa antar kelas, dengan nilai *recall* tertinggi terdapat pada kelas Berlebih dan Kurang (0.97), menandakan model lebih sensitif terhadap kondisi nutrisi tinggi dan kurang akurasi keseluruhan 94,51%, menunjukkan kemampuan model mempertahankan generalisasi meskipun proporsi data uji meningkat.

Pada konfigurasi ketiga, proporsi data latih dikurangi menjadi 60%, sementara 40% digunakan sebagai data uji. Parameter terbaik diperoleh pada C =

100 dengan kernel linear, menghasilkan skor *cross-validation* 0.9464 dan akurasi 94,83% pada data uji.

Tabel 4.12 Pengujian *Hyperparameter Tuning* Model SVM Rasio 60:40

Parameter Terbaik	**Akurasi (%)**	***Cross-Validation Score***	**Jumlah *Support Vector***
C = 100, Linear	94.83	0.9464	269



Gambar 4.15 *Confusion Matrix* Model SVM Rasio 60:40

Tabel 4.13 Metrik Evaluasi Model SVM Rasio 60:40

Kelas	***Precision***	***Recall***	***F1-Score***	**Jumlah Data**
Berlebih	0.93	0.98	0.96	348
Cukup	0.96	0.90	0.93	348
Kurang	0.95	0.97	0.96	348
Rata-rata	0.95	0.95	0.95	—

Meskipun jumlah data latih lebih sedikit, model masih mampu menjaga akurasi dan keseimbangan metrik antar kelas. Nilai *precision* dan *recall* yang relatif merata menunjukkan bahwa model linear tetap *robust* terhadap variasi ukuran data pelatihan.

Secara keseluruhan, seluruh model menunjukkan performa yang stabil dengan rata-rata akurasi di atas 90%. Variasi nilai parameter *C* hanya memengaruhi jumlah *support vector* yang digunakan untuk membentuk margin optimal. Rangkuman hasil pengujian ditunjukkan pada tabel berikut.

Tabel 4.14 Rangkuman Hasil Pengujian Model SVM

Rasio *Train-Test*	**Parameter Terbaik**	**Akurasi (%)**	***Cross-Validation Score***	**Jumlah *Support Vector***
80:20	C = 100, Linear	94,63	0.9478	340
70:30	C = 100, Linear	94,51	0.9486	305
60:40	C = 100, Linear	94,83	0.9464	269

Berdasarkan hasil pengujian pada ketiga variasi rasio data, dapat disimpulkan bahwa model SVM dengan kernel linear menunjukkan performa yang stabil dan konsisten di seluruh skenario. Perbedaan rasio data pelatihan dan pengujian tidak memberikan perubahan signifikan terhadap nilai akurasi maupun metrik evaluasi lainnya. Nilai akurasi tertinggi diperoleh pada rasio 60:40 sebesar 94,83%, diikuti oleh dua rasio lainnya yang mencapai 94,51% pada rasio 70:30 dan 94,63% pada rasio 80:20. Perbedaan kecil sekitar satu hingga dua persen menunjukkan bahwa model memiliki kemampuan generalisasi yang baik terhadap data yang belum pernah dilihat sebelumnya.

Dari sisi parameter, nilai *C* yang seragam pada masing-masing rasio menunjukkan bahwa model memerlukan margin yang lebih sempit untuk meminimalkan kesalahan klasifikasi ketika data pelatihan lebih sedikit. Selain itu, jumlah *support vector* cenderung menurun seiring berkurangnya proporsi data pelatihan, dari 340 pada rasio 80:20 menjadi 269 pada rasio 60:40, menandakan bahwa kompleksitas model berkurang sejalan dengan ukuran data latih.

Secara keseluruhan, hasil pengujian ini mengindikasikan bahwa SVM linear dengan mekanisme *One-vs-Rest* (OvR) efektif dalam memisahkan tiga kelas nutrisi (*Kurang*, *Cukup*, dan *Berlebih*) pada ruang dua dimensi hasil transformasi PCA. Model tidak hanya menunjukkan ketepatan yang tinggi, tetapi juga konsistensi performa antar skenario, sehingga dapat diandalkan untuk diimplementasikan dalam sistem pemantauan nutrisi hidroponik berbasis IoT secara *real-time*.

4.6 Pengujian Kalibrasi Sensor

Pengujian dan kalibrasi sensor dilakukan untuk mengevaluasi akurasi serta stabilitas pembacaan sensor pH, TDS, dan suhu air pada sistem hidroponik terhadap data pembanding yang diperoleh secara manual menggunakan alat ukur laboratorium. Tujuan utama dari tahap ini adalah memastikan kesesuaian hasil pembacaan sensor otomatis dengan alat ukur acuan sebelum sistem digunakan untuk pemantauan kualitas air secara berkelanjutan.

Pengukuran manual dilakukan menggunakan alat ukur digital multifungsi TDS/pH/EC/*Salinity meter* seperti ditunjukkan pada Gambar 4.15. Alat ini digunakan sebagai pembanding (*ground truth*) terhadap pembacaan sensor otomatis yang terpasang pada sistem IoT. Perangkat tersebut mampu mengukur beberapa parameter kualitas air secara bersamaan, meliputi pH, *Total Dissolved Solids* (TDS) dalam satuan ppm, konduktivitas listrik (EC), serta suhu air. Proses pengukuran dilakukan dengan cara mencelupkan *probe* alat ke dalam larutan nutrisi, menunggu hingga nilai pembacaan stabil, kemudian mencatat hasilnya secara manual.

Gambar 4.16 Alat Pengukuran Manual

Setiap pengukuran dilakukan pada waktu yang relatif bersamaan dengan sistem otomatis untuk memastikan kesesuaian waktu (*timestamp*) antara kedua sumber data. Alat ini dipilih karena memiliki tingkat kepraktisan tinggi, kecepatan respon cepat, serta akurasi yang cukup baik untuk kebutuhan kalibrasi lapangan pada sistem hidroponik skala kecil hingga menengah.

Pada tahap kalibrasi awal, perbedaan antara pembacaan sensor dan alat manual masih berada pada rentang yang relatif kecil, dengan selisih pH sekitar 2 poin dan rasio pembacaan TDS sekitar 1:1,2 terhadap alat ukur manual. Namun seiring waktu penggunaan, sensor mengalami degradasi kinerja yang cukup signifikan akibat paparan nutrisi berkonsentrasi tinggi**,** penumpukan residu pada *probe*, serta perubahan suhu lingkungan di dalam *greenhouse*. Pada periode akhir pengujian, selisih pembacaan pH meningkat menjadi sekitar 5–6 poin, sementara rasio pembacaan TDS membesar hingga sekitar 1:3, menandakan adanya penurunan sensitivitas sensor terhadap larutan nutrisi.

Proses pengujian dilakukan sepanjang periode pengambilan data otomatis dan data manual, yaitu dari 8 Agustus 2025 hingga 9 Oktober 2025. Data yang

diperoleh terdiri atas 1498 data otomatis dan 63 data pembacaan manual. Proses perhitungan MAE dijelaskan pada *pseudocode* 4.5 berikut:

Pseudocode 4.5

```
Algoritma Perhitungan_MAE_Sensor_vs_Manual
Input : Dataset pembacaan sensor otomatis (AUTO), Dataset pembacaan manual (MANUAL)
Output : Nilai MAE untuk pH, TDS, dan suhu air
Mulai
  1. Baca file data AUTO dan MANUAL.
  2. Konversi kolom waktu pada data AUTO menjadi format datetime.
  3. Gabungkan kolom tanggal dan waktu pada data MANUAL menjadi satu kolom datetime.
  4. Hapus baris yang memiliki waktu tidak valid (NaT).
  5. Urutkan kedua dataset berdasarkan waktu masing-masing.
  6. Cocokkan data MANUAL dengan data AUTO menggunakan pencarian waktu terdekat (nearest match)
      dengan toleransi selisih maksimum 1 jam.
  7. Hapus baris hasil pencocokan yang tidak memiliki pasangan nilai sensor.
  8. Hitung nilai MAE untuk setiap parameter:
      a. MAE_pH ← rata-rata(|pH_manual – pH_sensor|)
      b. MAE_TDS ← rata-rata(|TDS_manual – TDS_sensor|)
      c. MAE_Suhu ← rata-rata(|Suhu_manual – Suhu_sensor|)
  9. Tampilkan nilai MAE pH, MAE TDS, dan MAE suhu air.
  10. (Opsional) Visualisasikan perbandingan manual vs sensor dalam grafik garis.
Selesai
```

Setelah proses sinkronisasi waktu dilakukan, diperoleh 52 data pasangan yang memiliki *timestamp* berdekatan dan dapat dibandingkan secara langsung. Contoh hasil penggabungan kedua data tersebut disajikan pada Tabel 4.15.

Tabel 4.15 Data Teratas Hasil Penggabungan Manual dan Sensor Otomatis

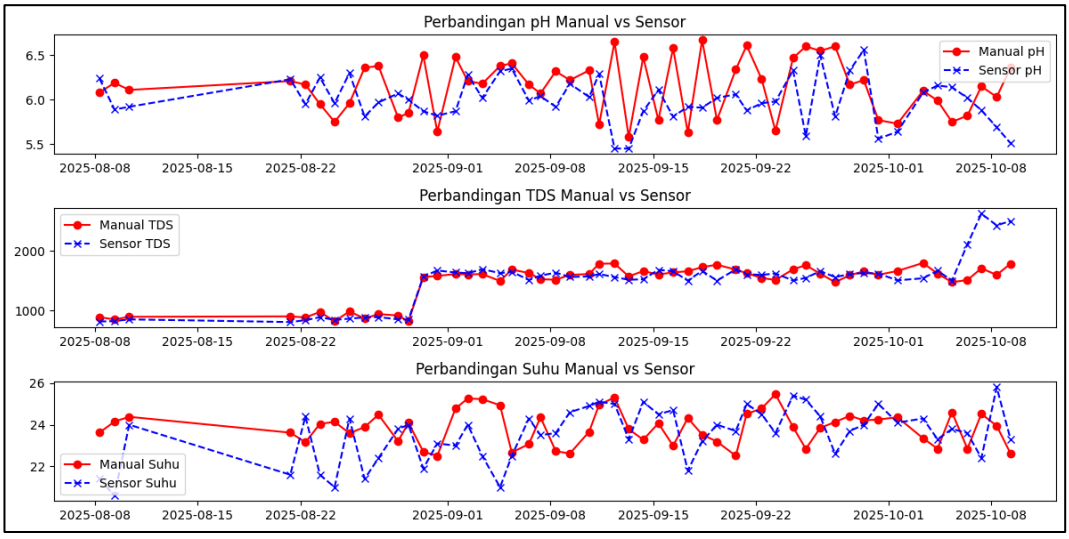
Tanggal	pH Manual	pH Sensor	TDS Manual (ppm)	TDS Sensor (ppm)	Suhu Manual (°C)	Suhu Sensor (°C)
08-08-2025	6.08	6.24	883.8	811.3	23.62	21.4
09-08-2025	6.19	5.89	846.7	817.6	24.15	20.6
10-08-2025	6.11	5.92	890.9	847.2	24.38	24.0
21-08-2025	6.21	6.23	898.0	802.0	23.62	21.6
22-08-2025	6.17	5.95	880.4	834.2	23.16	24.4

Perhitungan *Mean Absolute Error* (MAE) pada penelitian ini dilakukan menggunakan fungsi *mean_absolute_error* yang disediakan oleh modul *sklearn.metrics*. Implementasi rumus MAE ada pada blok kode berikut:

```
mae_ph = mean_absolute_error(df_merge['pH_Manual'],
df_merge['ph'])
mae_tds = mean_absolute_error(df_merge['TDS_Manual'],
df_merge['tds_ppm'])
mae_suhu = mean_absolute_error(df_merge['Suhu_Air_Manual'],
df_merge['water_temp_c'])
```

Fungsi tersebut secara internal menerapkan rumus MAE, yaitu rata-rata nilai absolut dari selisih antara nilai referensi dan nilai hasil pembacaan sensor dari masing-masing fitur.

Visualisasi perbandingan antara hasil pembacaan manual dan otomatis ditunjukkan pada Gambar 4.16 berikut.



Gambar 4.17 Perbandingan Pembacaan Manual dan Sensor

Dari hasil grafik tersebut terlihat bahwa pola perubahan nilai sensor otomatis secara umum mengikuti tren pembacaan manual, meskipun terdapat deviasi pada beberapa titik. Penyimpangan ini semakin terlihat pada akhir periode pengamatan, khususnya pada sensor pH dan TDS yang menunjukkan penurunan akurasi akibat faktor lingkungan dan kondisi fisik sensor yang menurun.

Untuk mengukur tingkat kesalahan rata-rata antara hasil pengukuran manual dan otomatis, digunakan metrik *Mean Absolute Error* (MAE). Nilai MAE masing-masing parameter ditunjukkan pada Tabel 4.16.

Tabel 4.16 Nilai MAE per Paramater

Parameter	**Nilai MAE**	**Satuan**
pH	0.335	–
TDS	134.692	ppm
Suhu Air	1.275	°C

Berdasarkan hasil tersebut, dapat disimpulkan bahwa sensor suhu air menunjukkan performa paling stabil dengan kesalahan rata-rata kecil, sedangkan sensor pH dan TDS menunjukkan variasi lebih besar terutama akibat penurunan sensitivitas sensor dan faktor kalibrasi lapangan. Meskipun demikian, hasil pengujian ini menunjukkan bahwa sistem masih dapat berfungsi dengan baik untuk pemantauan relatif (*relative monitoring*) kualitas air nutrisi dalam sistem hidroponik.

4.7 Analisis dan Pembahasan

Hasil penelitian menunjukkan bahwa sistem klasifikasi nutrisi tanaman hidroponik berbasis *Support Vector Machine* (SVM) dengan data hasil pengukuran sensor pH, TDS, dan suhu air berhasil diimplementasikan secara menyeluruh mulai dari tahap akuisisi data, *preprocessing*, hingga evaluasi model. Tahapan

preprocessing meliputi pembersihan data (*cleaning*), normalisasi dengan metode *Min-Max Scaling*, penyeimbangan kelas menggunakan teknik *Random Under Sampling* (RUS), serta reduksi dimensi menggunakan *Principal Component Analysis* (PCA).

Pada tahap reduksi dimensi, dua komponen utama (PC1 dan PC2) dari hasil PCA mampu menjelaskan 87,85% total variansi data, yang menunjukkan bahwa sebagian besar informasi penting dari ketiga fitur sensor tetap terwakili dalam ruang dua dimensi. PCA pada tahap ini digunakan untuk menyederhanakan representasi data agar proses klasifikasi oleh model SVM lebih efisien secara komputasi, tanpa mengubah struktur atau karakteristik dasar dari data sensor.

Model SVM yang digunakan dalam penelitian ini menerapkan kernel *linear*, yang dipilih karena kesederhanaan perhitungannya, waktu komputasi yang lebih cepat, serta hasil yang mudah diinterpretasikan dibandingkan kernel *non-linear* seperti *Radial Basis Function (RBF)* atau *polynomial* (Cervantes et al., 2020). Dengan pendekatan *One-vs-Rest* (OvR)**,** model membangun tiga *hyperplane* yang masing-masing memisahkan satu kelas terhadap dua kelas lainnya, yaitu *"Kurang vs lainnya"*, *"Cukup vs lainnya"*, dan *"Berlebih vs lainnya"*.

Hasil pengujian menunjukkan bahwa rasio pembagian data pada masing-masing menghasilkan akurasi yang cukup seragam pada akurasi 94%. Hal ini menandakan bahwa masing-masing model memiliki stabilitas performa terhadap variasi jumlah data pelatihan. Visualisasi *decision boundary* pada ruang dua dimensi hasil PCA menunjukkan bahwa model mampu memisahkan ketiga kelas dengan margin yang cukup jelas. Titik-titik *support vector* yang berada di tepi

setiap kelas menandakan terbentuknya *hyperplane* optimal dengan margin maksimum. Adanya tumpang tindih antar kelas *Cukup* dengan *Kurang* maupun *Berlebih* diperkirakan disebabkan oleh nilai pH dan TDS yang berada di batas rentang optimal nutrisi.

Model *Support Vector Machine* (SVM) yang digunakan dalam sistem IoT pada *interface* web merupakan hasil pelatihan dengan rasio pembagian data 60:40, karena konfigurasi tersebut menghasilkan akurasi tertinggi sebesar 94,83% dibandingkan dua skenario lainnya. Oleh sebab itu, model dengan parameter terbaik C = 100 dan kernel linear dipilih sebagai model final dan diintegrasikan pada sistem web berbasis *Internet of Things* (IoT) untuk klasifikasi kondisi nutrisi tanaman melon hidroponik.

Model klasifikasi nutrisi diterapkan dalam bentuk *pipeline* terintegrasi yang disimpan dalam berkas .pkl menggunakan pustaka *joblib*. *Pipeline* ini memuat tiga komponen utama, yaitu *MinMaxScaler* untuk proses normalisasi data, *Principal Component Analysis* (PCA) untuk reduksi dimensi, serta *Support Vector Machine* (SVM) sebagai algoritma klasifikasi utama. Ketiga komponen tersebut bekerja secara berurutan untuk memastikan proses *preprocessing* dan inferensi berlangsung otomatis dan konsisten di sisi *server* web.

Data masukan berasal dari hasil pembacaan sensor pH (*ph*), kadar zat terlarut total (*tds_ppm*), dan suhu air (*water_temp_c*) yang dikirim secara real-time dari perangkat IoT. Setelah pipeline berhasil dimuat dari direktori *saved_pipeline/svm_pca_pipeline.pkl*, sistem melakukan serangkaian proses mulai

dari normalisasi, transformasi PCA, hingga klasifikasi akhir oleh model SVM. Alur inferensi tersebut dijelaskan melalui *pseudocode* 4.6 berikut:

Pseudocode 4.6

```
Algoritma Pipeline Klasifikasi Nutrisi Air
Input: Data sensor (ph, tds_ppm, water_temp_c)
Output: Label klasifikasi nutrisi air {Kurang, Cukup, Berlebih}
Mulai
   1. Muat pipeline model terlatih yang telah disimpan dalam
      file "svm_pca_pipeline.pkl".
   2. Terima data sensor dari perangkat IoT berupa nilai pH, TDS
      (ppm), dan suhu air (°C).
   3. Lakukan tahap praproses data dengan langkah-langkah
      berikut:
   4. Normalisasi setiap fitur menggunakan MinMaxScaler agar
      seluruh parameter berada pada rentang nilai yang seragam.
   5. Transformasikan data hasil normalisasi menggunakan
      Principal Component Analysis (PCA) untuk mereduksi dimensi
      dan mengoptimalkan representasi fitur.
   6. Jalankan model Support Vector Machine (SVM) yang telah
      terintegrasi dalam pipeline untuk menghasilkan hasil
      klasifikasi kelas nutrisi air.
Selesai.
```

Pada tahap pengujian, *pipeline* menerima input data dari sensor IoT berupa nilai pH, TDS, dan suhu air. Data tersebut kemudian melewati tahapan normalisasi dan transformasi PCA sebelum akhirnya diprediksi oleh model SVM. Hasil klasifikasi berupa label “Kurang”, “Cukup”, atau “Berlebih” yang muncul setelah melewati tahapan pengujian ditampilkan pada *dashboard* web di bagian *nutrition status* dan dapat di monitor secara *real-time* berdasarkan data sensor yang didapat.

Secara keseluruhan, hasil ini menunjukkan bahwa SVM dengan kernel *linear* dapat menjadi pendekatan yang efektif untuk klasifikasi kondisi nutrisi tanaman hidroponik berbasis data sensor IoT. Pendekatan ini juga memudahkan interpretasi hasil model pada tahap visualisasi dan evaluasi sistem, sehingga sesuai untuk diterapkan dalam sistem *monitoring* berbasis data *real-time*.

Selain dari sisi teknis, hasil penelitian ini juga dapat ditinjau dari perspektif integrasi Islam dan sains, yang menempatkan teknologi sebagai sarana untuk mewujudkan kemaslahatan dan menjaga keseimbangan dalam pengelolaan sumber daya alam. Hal ini sejalan dengan firman Allah dalam QS. *Al-An'am* ayat 141 yang berbunyi:

وَهُوَ الَّذِي أَنشَأَ جَنَّاتٍ مَّعْرُوشَاتٍ وَغَيْرَ مَعْرُوشَاتٍ وَالنَّخْلَ وَالزَّرْعَ مُخْتَلِفًا أُكُلُهُ وَالزَّيْتُونَ وَالرُّمَّانَ مُتَشَابِهًا وَغَيْرَ مُتَشَابِهٍ ۚ كُلُوا مِن ثَمَرِهِ إِذَا أَثْمَرَ وَآتُوا حَقَّهُ يَوْمَ حَصَادِهِ ۖ وَلَا تُسْرِفُوا ۚ إِنَّهُ لَا يُحِبُّ الْمُسْرِفِينَ

"Dan Dialah yang menjadikan kebun-kebun yang berjunjung dan yang tidak berjunjung, pohon kurma, tanam-tanaman yang bermacam-macam rasanya, zaitun, dan delima yang serupa dan yang tidak serupa. Makanlah dari buahnya bila dia berbuah, dan tunaikanlah haknya di hari memetik hasilnya, dan janganlah kamu berlebih-lebihan. Sesungguhnya Allah tidak menyukai orang yang berlebih-lebihan." (QS. *Al-An'am* [6]:141).

Dimana ayat tersebut menegaskan agar manusia tidak berlebih-lebihan dan tetap memperhatikan prinsip keberlanjutan dalam bertani dan mengelola hasil bumi. Landasan kedua yang dimana Rasulullah juga bersabda dalam HR. Bukhari No. 2320,

"Tidaklah seorang muslim menanam pohon atau menabur benih, lalu sebagian darinya dimakan oleh burung, manusia, atau binatang, melainkan itu menjadi sedekah baginya." [HR. Bukhari No. 2320].

Hadits tersebut menjelaskan bahwa kedua landasan tersebut menunjukkan bahwa penggunaan teknologi IoT dan kecerdasan buatan dalam bidang pertanian bukan hanya bagian dari inovasi sains, tetapi juga wujud nyata dari yang amal ilmiah mencerminkan tanggung jawab manusia sebagai *khalifah* di bumi untuk menjaga keberlanjutan lingkungan. Penelitian ini tidak hanya berkontribusi pada

peningkatan efisiensi dan akurasi pemantauan nutrisi tanaman, tetapi juga mencerminkan penerapan nilai-nilai Islam dalam upaya mengoptimalkan potensi ilmu pengetahuan untuk kemaslahatan umat dan kelestarian alam.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan seluruh rangkaian penelitian yang telah dilaksanakan, dapat disimpulkan bahwa sistem *Internet of Things* (IoT) yang dikembangkan berhasil memenuhi tujuan penelitian, yakni mengklasifikasikan kebutuhan nutrisi tanaman melon hidroponik secara otomatis menggunakan algoritma *Support Vector Machine* (SVM). Sistem mampu melakukan akuisisi data pH, TDS, dan suhu air, kemudian mengolahnya melalui serangkaian tahapan meliputi pembersihan data, normalisasi *Min-Max*, penyeimbangan kelas, dan reduksi dimensi dengan PCA sebelum dilakukan klasifikasi ke dalam kategori Kurang, Cukup, dan Berlebih. Model SVM kernel linear yang diimplementasikan menunjukkan performa yang konsisten dengan akurasi 94,63% pada rasio 80:20, 94,51% pada 70:30, dan 94,83% pada 60:40, menandakan bahwa model mampu memisahkan kelas dengan efektif meskipun variasi rasio pelatihan-pengujian berubah. Selain itu, pengujian kalibrasi sensor menghasilkan nilai MAE sebesar 0,335 untuk pH, 134,692 ppm untuk TDS, dan 1,275°C untuk suhu air, menunjukkan tingkat penyimpangan sensor yang masih berada pada batas wajar sehingga layak digunakan dalam operasional sistem. Integrasi sistem dengan *server* dan *dashboard* web juga memungkinkan pemantauan nutrisi secara *real-time*, memberikan dukungan keputusan yang cepat dan presisi dalam pengelolaan larutan nutrisi. Secara keseluruhan, sistem IoT yang dibangun dan model SVM yang diterapkan mampu

menyediakan solusi pemantauan dan klasifikasi nutrisi yang otomatis, akurat, efisien, dan sesuai untuk mengoptimalkan manajemen budidaya melon hidroponik.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan oleh peneliti, saran yang dapat dijadikan sebagai pengembangan penelitian selanjutnya antara lain.

1. Pengembangan Model:

   Penelitian selanjutnya disarankan untuk menambahkan algoritma pembanding seperti *Random Forest*, *K-Nearest Neighbor* (KNN), atau *Neural Network* guna memperoleh pembanding performa yang lebih komprehensif terhadap SVM.

2. Perluasan Dataset:

   Dataset sensor sebaiknya diperluas dengan waktu pengambilan data yang lebih panjang dan variasi kondisi lingkungan yang lebih beragam agar model memiliki kemampuan generalisasi yang lebih baik.

3. Kalibrasi Sensor Berkala:

   Berdasarkan hasil pengujian kalibrasi, diketahui bahwa akurasi sensor mengalami pergeseran nilai seiring waktu, khususnya pada sensor pH dan TDS. Oleh karena itu, diperlukan proses kalibrasi berkala untuk menjaga keakuratan data dan stabilitas sistem.

4. Integrasi Sistem Kontrol Otomatis:

   Sistem dapat dikembangkan menjadi *closed-loop system* dengan menambahkan aktuator otomatis yang mampu mengatur dosis nutrisi

berdasarkan hasil klasifikasi, sehingga proses monitoring dan pengendalian dapat berlangsung secara mandiri.

DAFTAR PUSTAKA

Amaya-Tejera, N., et al. (2024). A distance-based kernel for classification via Support Vector Machines. Frontiers in Artificial Intelligence. https://doi.org/10.3389/frai.2024.1287875

Anindita, D., & Saputra, A. (2021). Penerapan algoritma *Support Vector Machine* dalam klasifikasi data pertanian. *Jurnal Teknologi Informasi dan Ilmu Komputer, 8*(2), 145–153. https://doi.org/10.25126/jtiik.202182345

Arifin, M., & Kusuma, R. (2020). Analisis kebutuhan nutrisi tanaman hidroponik berbasis data lingkungan. *Jurnal Agritech, 40*(3), 201–209. https://doi.org/10.22146/agritech.56789

Badan Pengembangan dan Pembinaan Bahasa. (2024). Kamus Besar Bahasa Indonesia (KBBI). Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia. https://kbbi.kemdikbud.go.id

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Cafaro, M., & Pulimeno, C. (2019). Supervised Learning: Classification. Dalam Encyclopedia of Bioinformatics and Computational Biology (Vol. 1, hlm. 342–349). Elsevier. https://doi.org/10.1016/B978-0-12-809633-8.20332-4

Cervantes, J., et al. (2020). A comprehensive survey on support vector machine: applications, trends, and challenges. Neurocomputing, 408, 189–215. https://doi.org/10.1016/j.neucom.2020.03.014

Chen, W., Yang, K., Yu, Z., Shi, Y., & Chen, C. L. P. (2024). A survey on imbalanced learning: Latest research, applications and future directions. Artificial Intelligence Review, 57, 137. https://doi.org/10.1007/s10462-024-10759-6

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics, 21(1), 6. https://doi.org/10.1186/s12864-019-6413-7

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297. https://doi.org/10.1007/BF00994018

Darmawan, A. A., Dinarto, W., & Widodo, D. A. (2024). Pengaruh formulasi nutrisi AB Mix (racikan vs pabrikan) pada pertumbuhan dan hasil tanaman melon hidroponik sistem irigasi tetes. Savana Cendana: Jurnal Pertanian

Konservasi Lahan Kering, 9(3), 78–82. https://doi.org/10.32938/sc.v9i03.2340

Dewi, R., & Lestari, S. (2021). Pengaruh manajemen nutrisi terhadap kualitas hasil panen melon hidroponik. *Jurnal Agrosains, 12*(1), 55–64. https://doi.org/10.31289/agrosains.v12i1.2345

Furoidah, N. (2018). Efektivitas nutrisi AB Mix terhadap hasil dua varietas melon. Agritrop: Jurnal Ilmu-Ilmu Pertanian, 16(1), 186–196. https://jurnal.unmuhjember.ac.id/index.php/AGRITROP/article/view/1562

Gao, X., Xie, D., Zhang, Y., Wang, Z., He, C., Yin, H., & Zhang, W. (2025). A comprehensive survey on imbalanced data learning. arXiv preprint. https://arxiv.org/abs/2502.08960

Hidayat, A. (2020). Pertanian berkelanjutan dalam perspektif Al-Qur'an. *Jurnal Studi Islam, 15*(2), 220–231. https://doi.org/10.21009/jsi.152.07

Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. International Journal of Data Mining & Knowledge Management Process (IJDKP), 5(2), 1–11. https://doi.org/10.5121/ijdkp.2015.5201

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374(2065), 20150202. https://doi.org/10.1098/rsta.2015.0202

Kawasaki, Y., Komatsu, H., & Shirai, T. (2020). Effects of nutrient solution temperature on growth and yield of netted melon in hydroponics. Agronomy, 10(6), 898–908. https://doi.org/10.3390/agronomy10060898

Kurniawan, A., & Prasetyo, Y. (2022). Pemanfaatan *Internet of Things* untuk sistem monitoring pertanian modern. *Jurnal Teknologi Pertanian, 13*(1), 33–41. https://doi.org/10.34128/jtp.v13i1.982

Krishana, & Dr. Vinod Kumar. (2024). Covariance Matrix Analysis Through Eigenvalues and Eigenvectors: Insights into Multivariate Data Structures. *Revista Electronica De Veterinaria*, *25*(2), 698-702. https://doi.org/10.69980/redvet.v25i2.1495

Muftashiva, T. D., Munadi, M., & Haryanto, I. (2024). Sistem smart plant monitoring pada hidroponik melon berbasis Internet of Things. Jurnal Teknik Mesin, 12(3), 51–56. Retrieved from https://ejournal3.undip.ac.id/index.php/jtm/article/view/45973

Nugroho, A., & Fathurrahman, M. (2021). Integrasi sensor IoT dalam sistem hidroponik cerdas. *Jurnal Sistem Informasi, 17*(2), 189–197. https://doi.org/10.24089/jsi.v17i2.403

Patel, H., & Prajapati, P. (2018). Study and analysis of various normalization techniques for data preprocessing. *International Journal of Engineering Research and Technology (IJERT), 7*(5), 33–35. https://doi.org/10.17577/IJERTV7IS050052

Pennsylvania State University. (n.d.). *Lesson 7: Principal Components Analysis (STAT 508)*. Retrieved from https://online.stat.psu.edu/stat508/Lesson07.html

Prasetyo, D., & Sari, M. (2020). Analisis peluang pasar komoditas melon di Indonesia. *Jurnal Ekonomi Pertanian, 8*(4), 76–85. https://doi.org/10.33512/jep.v8i4.3376

Rachman, T., Utami, F., & Hidayah, N. (2021). Implementasi *Support Vector Machine* untuk klasifikasi kualitas hasil pertanian. *Jurnal Ilmu Komputer dan Informatika, 9*(1), 66–74. https://doi.org/10.33387/jik.v9i1.3052

Rahmadi, A., Sari, N., & Yuliani, D. (2025). Support Vector Machine untuk Prediksi Pola Pertumbuhan Selada Hidroponik. Jurnal Teknologi Pertanian, 12(1), 45–53.

Rahmawati, I. (2021). Ayat-ayat pertanian dalam Al-Qur'an dan implikasinya terhadap pembangunan berkelanjutan. *Jurnal Al-Tafsir, 6*(1), 55–67. https://doi.org/10.15575/jat.v6i1.1234

Roriz, R., Carvalho, H., & Ribeiro, J. (2021). Sensor calibration: A review. Measurement, 174, 108998. https://doi.org/10.1016/j.measurement.2020.108998

Santoso, H., & Firdaus, R. (2023). Efisiensi pertanian presisi dengan integrasi teknologi IoT. *Jurnal Informatika Pertanian, 5*(1), 1–10. https://doi.org/10.47134/jip.v5i1.1764

Sharma, A., & Paliwal, K. K. (2021). Principal Component Analysis and its Applications in Machine Learning. International Journal of Engineering Research & Technology, 10(6), 354–360. https://doi.org/10.17577/IJERTV10IS060207

Shuzhanfan. (2018). Understanding the mathematics behind Support Vector Machines. Retrieved from https://shuzhanfan.github.io/2018/05/understanding-mathematics-behind-support-vector-machines/

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Stanford NLP. (n.d.). Soft margin classification. In Introduction to Information Retrieval. Stanford University. Retrieved from https://nlp.stanford.edu/IR-book/html/htmledition/soft-margin-classification-1.html

Sulaiman, S., et al. (2024). Hybrid Ensemble Machine Learning Models for Nutrient Solution Estimation in Hydroponics. Computers and Electronics in Agriculture, 215, 108426. https://doi.org/10.1016/j.compag.2023.108426

Swaminathan, Sathyanarayanan & Tantri, B Roopashri. (2024). Confusion Matrix-Based Performance Evaluation Metrics. African Journal of Biomedical Research. 27. 4023-4031. 10.53555/AJBR.v27i4S.4345.

Syahputra, A., Wibowo, L., & Maulana, H. (2022). Kendala dan solusi pengelolaan nutrisi dalam hidroponik. *Jurnal Pertanian Modern, 14*(2), 101–111. https://doi.org/10.25077/jpm.14.2.101-111.2022

Tappi, A., & Tandibayang, C. (2025). Optimasi pemberian nutrisi AB Mix pada fase pertumbuhan melon hidroponik. Jurnal Agroteknologi, 16(1), 55–63.

Vasylenko, T., Shatokhin, A., & Lysenko, O. (2020). The role of sensor calibration in ensuring measurement accuracy. Journal of Physics: Conference Series, 1553, 012002. https://doi.org/10.1088/1742-6596/1553/1/012002

Visual Studio Magazine. (2020). *Data prep for machine learning: Normalization*. Retrieved from https://visualstudiomagazine.com/articles/2020/08/04/ml-data-prep-normalization.aspx

Wahyudi, B., & Putra, I. (2023). Peran pH dan TDS dalam penyerapan nutrisi hidroponik. *Jurnal Sains Pertanian Indonesia, 11*(2), 132–139. https://doi.org/10.25077/jspi.11.2.132-139.2023

Wahyuni, A., & Pratama, R. (2020). Klasifikasi kebutuhan nutrisi tanaman berbasis *Support Vector Machine*. *Jurnal Teknologi Informasi, 6*(2), 75–83. https://doi.org/10.30872/jti.v6i2.1745

Wijayanti, S., & Susanti, D. (2021). Kesulitan petani dalam pengelolaan nutrisi hidroponik. *Jurnal Agroteknologi, 15*(1), 45–53. https://doi.org/10.25077/agroteknologi.15.1.45-53.2021

Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate Research, 30(1), 79–82. https://doi.org/10.3354/cr030079

Yusuf, M. (2019). Islam dan pembangunan pertanian berkelanjutan. *Jurnal Ilmu Syariah, 10*(2), 144–155. https://doi.org/10.21043/jis.v10i2.5076

Zhang, L., Li, H., & Wang, J. (2022). Application of PCA in agricultural data analysis for crop monitoring. Computers and Electronics in Agriculture, 194, 106780. https://doi.org/10.1016/j.compag.2022.106780

LAMPIRAN

Lampiran 1. Kode Program Pelabelan Otomatis (*Rule-Based Labeling*)

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Pastikan kolom numerik
df['ph'] = pd.to_numeric(df['ph'], errors='coerce')
df['tds_ppm'] = pd.to_numeric(df['tds_ppm'], errors='coerce')
df['water_temp_c'] = pd.to_numeric(df['water_temp_c'],
errors='coerce')

def label_nutrisi(row):
    ph = row['ph']
    tds = row['tds_ppm']
    temp = row['water_temp_c']

    if pd.isna(ph) or pd.isna(tds) or pd.isna(temp):
        return np.nan

    # --- RULE: Cukup ---
    if (5.8 <= ph <= 6.4) and ((800 <= tds <= 1000) or (1500 <=
tds <= 1700)) and (20 <= temp <= 27):
        return 1  # Cukup

    # --- RULE: Kurang ---
    elif (ph < 5.8) or (tds < 800) or (temp < 20):
        return 0  # Kurang

    # --- RULE: Berlebih ---
    elif (ph > 6.4) or (tds > 1700) or (temp > 27):
        return 2  # Berlebih

    else:
        return np.nan

# Terapkan fungsi
df['nutrisi_label'] = df.apply(label_nutrisi, axis=1)

# Distribusi hasil labeling
print("📊 Distribusi Label Nutrisi:")
print(df['nutrisi_label'].value_counts(dropna=False).sort_index())
print("\n0 = Kurang | 1 = Cukup | 2 = Berlebih")

# Visualisasi hasil labeling
plt.figure(figsize=(6,4))
sns.countplot(x='nutrisi_label', data=df, palette='coolwarm')
plt.title("Distribusi Label Nutrisi (Rule-Based)")
plt.xlabel("Label (0=Kurang, 1=Cukup, 2=Berlebih)")
plt.ylabel("Jumlah Data")
plt.grid(alpha=0.3)
```

```
plt.show()
```

Lampiran 2. Kode Program Normalisasi Data (*Min–Max Scaling*)

```
from sklearn.preprocessing import MinMaxScaler
import joblib # Import joblib

print("⚙️ Proses normalisasi dimulai...\n")

# Pilih kolom fitur yang mau dinormalisasi
fitur = ['ph', 'tds_ppm', 'water_temp_c']

# Buat scaler dan fit_transform ke data
scaler = MinMaxScaler()
df_scaled = df.copy()
df_scaled[fitur] = scaler.fit_transform(df_scaled[fitur])

# Simpan parameter min-max biar bisa dipakai di prediksi nanti
minmax_params = {
    'min': scaler.data_min_,
    'max': scaler.data_max_,
    'range': scaler.data_range_
}

scaler_filename = "fitted_scaler.pkl"
joblib.dump(scaler, scaler_filename)

print("✅ Normalisasi selesai!")
print("Nilai minimum & maksimum tiap fitur sebelum scaling:")
for i, f in enumerate(fitur):
    print(f"  {f}  →  min: {minmax_params['min'][i]:.3f} | max:
{minmax_params['max'][i]:.3f}")

print("\nCek 5 data pertama hasil normalisasi:")
df_scaled.head()
```

Lampiran 3. Kode Program *Balancing* Data (*Equalized Class Reconstruction*)

```
import numpy as np

print("🔄 Melakukan FINAL RE-ENGINEER + BALANCING...\n")

df_bal = df_scaled.copy()

print("Jumlah awal per kelas (df_bal):")
print(df_bal['nutrisi_label'].value_counts().sort_index())

fitur_sensor = ['ph', 'tds_ppm', 'water_temp_c']

# TARGET = mayoritas (870)
target = df_bal['nutrisi_label'].value_counts().max()
print("\n🎯 Target jumlah tiap kelas:", target)

df_final = []
```

```
for label, group in df_bal.groupby('nutrisi_label'):

    group = group.copy().reset_index(drop=True)
    n = len(group)

    # --- Jika kurang dari target → buat data synthetic ---
    if n < target:
        needed = target - n

        # ambil pasangan titik untuk interpolasi synthetic
        idx1 = np.random.randint(0, n, needed)
        idx2 = np.random.randint(0, n, needed)

        synth =
group[fitur_sensor].iloc[idx1].reset_index(drop=True)*0.5 + \
                group[fitur_sensor].iloc[idx2].reset_index(drop=Tr
ue)*0.5

        synth['nutrisi_label'] = label

        df_new = pd.concat([group, synth], axis=0)

    # --- Jika lebih dari target → sampling acak ---
    else:
        df_new = group.sample(target, random_state=42)

    df_final.append(df_new)

# Gabungkan & shuffle
df_re = pd.concat(df_final).sample(frac=1,
random_state=42).reset_index(drop=True)

print("\n Jumlah akhir per kelas:")
print(df_re['nutrisi_label'].value_counts().sort_index())
```

Lampiran 4. Kode Program Reduksi Dimensi Menggunakan PCA

```
from sklearn.decomposition import PCA
import pandas as pd
import joblib # Import joblib to save the fitted PCA object

print(" Proses PCA dimulai...\n")

# Pisahkan fitur dan label
X = df_balanced[['ph', 'tds_ppm', 'water_temp_c']]
y = df_balanced['nutrisi_label']

# Buat objek PCA dengan 2 komponen
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X)

# *** SIMPAN OBJEK PCA YANG SUDAH DI-FIT ***
pca_filename = "fitted_pca.pkl"
joblib.dump(pca, pca_filename)
```

```
print(f"✅ Objek PCA yang sudah di-fit disimpan di ->
{pca_filename}")
# ***************************************

# Konversi ke DataFrame baru
df_pca = pd.DataFrame(data=X_pca, columns=['PC1', 'PC2'])
df_pca['nutrisi_label'] = y.values

# Tampilkan hasil varian
print("🔷 Proporsi Variansi Tiap Komponen:")
for i, ratio in enumerate(pca.explained_variance_ratio_):
    print(f"  PC{i+1}: {ratio:.4f}")

print(f"\n🔷 Total variansi yang terjelaskan oleh 2 komponen:
{sum(pca.explained_variance_ratio_):.4f}")

# Cek data hasil PCA
df_pca.head()

print("\n☑️ PCA Component Loadings:")
loadings = pd.DataFrame(pca.components_, columns=X.columns,
index=['PC1', 'PC2'])
print(loadings)
```

Lampiran 5. Model Pipeline untuk Penerapan Klasifikasi di Web

```
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import MinMaxScaler
from sklearn.decomposition import PCA
from sklearn.svm import SVC
import joblib
import os
import pandas as pd # Import pandas

# ====================================
# MEMBUAT PIPELINE MODEL
# ====================================

# 1. Muat objek scaler yang sudah di-fit pada data normalisasi
scaler_filename = "fitted_scaler.pkl"
if not os.path.exists(scaler_filename):
    print(f"❌ File Scaler yang sudah di-fit tidak ditemukan di:
{scaler_filename}")
    print("❗ Harap jalankan kembali cell Random Undersampling
untuk membuat file ini.")
    # Anda perlu menjalankan kembali cell Random Undersampling
untuk membuat file ini.
else:
    scaler_loaded = joblib.load(scaler_filename)
    print(f"✅ Objek Scaler yang sudah di-fit berhasil dimuat
dari -> {scaler_filename}")

    # 2. Muat objek PCA yang sudah di-fit
```

```
    #    Objek ini sudah di-fit pada data yang *undersampled* di
cell PCA (ZtlsbXejE3wb)
    pca_filename = "fitted_pca.pkl"
    if not os.path.exists(pca_filename):
        print(f"❌ File PCA yang sudah di-fit tidak ditemukan di:
{pca_filename}")
        print("❗ Harap jalankan kembali cell PCA untuk membuat
file ini.")
        # Anda perlu menjalankan kembali cell PCA untuk membuat
file ini.
    else:
        pca_reloaded = joblib.load(pca_filename)
        print(f"✅ Objek PCA yang sudah di-fit berhasil dimuat
dari -> {pca_filename}")

        # 3. Muat model SVM terbaik (dari split 80:20)
        model_path = "saved_models/SVM_linear_80_20.pkl"
        if not os.path.exists(model_path):
             print(f"❌ File model SVM tidak ditemukan di:
{model_path}")
             print("❗ Harap jalankan kembali cell Training SVM
untuk membuat file ini.")
        else:
            svm_model_loaded = joblib.load(model_path)
            print(f"✅ Model SVM ({model_path}) berhasil
dimuat.")

            # 4. Buat Pipeline
            #    Gunakan scaler_loaded yang dimuat dari file
            pipeline = Pipeline([
                ('scaler', scaler_loaded), # Gunakan scaler yang
dimuat
                ('pca', pca_reloaded),
                ('svm', svm_model_loaded)
            ])

            print("\n🎉 Pipeline model berhasil dibuat!")

            # ====================================
            # SIMPAN PIPELINE MENGGUNAKAN JOBLIB
            # ====================================

            # Pastikan folder penyimpanan tersedia
            save_dir = "saved_pipeline"
            os.makedirs(save_dir, exist_ok=True)

            pipeline_filename = os.path.join(save_dir,
"svm_pca_pipeline.pkl")
            joblib.dump(pipeline, pipeline_filename)

            print(f"\n✅ Pipeline model disimpan di ->
{pipeline_filename}")
```

Lampiran 6. Implementasi Model Klasifikasi Pada Sistem Web

```
import joblib
import pandas as pd
import os

pipeline_filename = os.path.join("saved_pipeline",
"svm_pca_pipeline.pkl")

# Pastikan file pipeline ada
if not os.path.exists(pipeline_filename):
    print(f"❌ File pipeline tidak ditemukan di:
{pipeline_filename}")
else:
    loaded_pipeline = joblib.load(pipeline_filename)
    print(f"✅ Pipeline model berhasil dimuat dari ->
{pipeline_filename}")

    new_data = pd.DataFrame({
        'ph': [6.7],
        'tds_ppm': [2000],
        'water_temp_c': [24]
    })

    print("\nInput data baru:")
    display(new_data)

    # --- Langkah Debugging: Lihat hasil Scaling dan PCA ---
    # Ambil scaler dan pca dari pipeline
    scaler_step = loaded_pipeline.named_steps['scaler']
    pca_step = loaded_pipeline.named_steps['pca']

    # Scaling
    scaled_data = scaler_step.transform(new_data)
    scaled_df = pd.DataFrame(scaled_data, columns=['ph_scaled',
'tds_ppm_scaled', 'water_temp_c_scaled'])
    print("\nData setelah Scaling:")
    display(scaled_df)

    # PCA Transformation
    pca_data = pca_step.transform(scaled_data)
    pca_df = pd.DataFrame(pca_data, columns=['PC1', 'PC2'])
    print("\nData setelah PCA:")
    display(pca_df)
    # --- Akhir Langkah Debugging ---

    prediction = loaded_pipeline.predict(new_data)

    # Mapping hasil klasifikasi ke label yang mudah dibaca
    label_map = {0.0: 'Kurang', 1.0: 'Cukup', 2.0: 'Berlebih'}
    predicted_label = label_map.get(prediction[0], "Label tidak
dikenali")

    print(f"\nHasil Klasifikasi Label Nutrisi: {prediction}")
```