AQUAPONIK CERDAS: PENDEKATAN BERBASIS DATA UNTUK KONTROL PH DAN SUHU MENGGUNAKAN ALGORITMA K NEAREST NEIGHBOR (K-NN) PADA ARDUINO

Linza Mawadda Rahmah dan Antonius Irianto Sukowati*

Universitas Cendekia Abditama
Jl. Islamic Raya, Klp. Dua, Kec. Klp. Dua, Kabupaten Tangerang, Banten 15811
linza@cendekia.ac.id, irianto@cendekia.ac.id
*Corresponding Author

ABSTRAK

Penelitian ini menyajikan sistem kontrol otomatis untuk regulasi kadar pH dan suhu pada sistem aquaponik menggunakan metode K-Nearest Neighbor (K-NN) yang diimplementasikan pada mikrokontroler Arduino Uno. Sistem memanfaatkan sensor pH-4502C dan sensor suhu DS18B20 untuk memantau kondisi lingkungan. Sistem ini dibuat tanpa menggunakan logika kontrol yang berbasis ambang batas. Sebagai gantinya algoritma K-NN mengklasifikasikan data sensor dengan membandingkannya terhadap dataset pelatihan menggunakan jarak euclidean, sehingga memungkinkan pengambilan keputusan adaptif untuk mengaktifkan aktuator (pompa penyesuaian pH, pemanas, kipas). Hasil pengujian menunjukkan akurasi klasifikasi sebesar 95%. Sistem terbukti mampu menjaga pH dan suhu dalam rentang optimal, menjamin lingkungan aquaponik yang stabil. Pendekatan ini menawarkan solusi yang cerdas, andal, dan efisien untuk pertanian perkotaan berkelanjutan.

Kata Kunci: *Aquaponik, K-Nearest Neighbor (K-NN), embedded machine learning, sistem embedded, Arduino Uno*

PENDAHULUAN

Pertumbuhan penduduk yang terus meningkat dan urbanisasi yang pesat menghadirkan tantangan signifikan bagi sistem pangan global yang lebih efisien dan berkelanjutan. Dalam konteks tersebut, aquaponik menawarkan sebuah pendekatan inovatif yang menggabungkan akuakultur dan hidroponik, yang memungkinkan efisiensi penggunaan air hingga 90% dibandingkan dengan pertanian konvensional [1]. Mengingat sifat sinergis dari sistem aquaponik, kestabilan parameter lingkungan seperti pH dan suhu air sangat krusial untuk keberhasilan produksi pangan tersebut dalam kaitannya dengan nitrifikasi, ketersediaan nutrisi, dan kesehatan ikan serta tanaman [2], [3]. Monitoring dan otomasi untuk menjaga rentang optimal pH (6,5–7,5) dan suhu (20–30°C) adalah langkah penting untuk memastikan keseimbangan ekosistem aquaponik dapat dipelihara [4].

Meskipun pengendalian parameter pH dan suhu dapat dilakukan secara manual, metode tersebut sering kali tidak praktis dan rentan terhadap kesalahan manusia. Oleh karena itu, otomasi berbasis mikrokontroler seperti Arduino telah diusulkan sebagai solusi yang lebih efisien untuk mengatasi masalah ini, dengan kemampuan untuk mengontrol pompa pH-up/pH-down, *heater*, dan *fan* secara otomatis [5]. Namun demikian, mayoritas sistem otomasi saat ini masih mengandalkan logika kontrol berbasis ambang batas yang bersifat reaktif, sehingga kurang fleksibel dalam bereaksi terhadap dinamika lingkungan yang sering berubah [6], [7]. Permasalahan dalam fleksibilitas dalam melakukan reaksi ini membuka peluang bagi penerapan teknik machine learning (ML) untuk meningkatkan kemampuan adaptasi dan prediktibilitas sistem kontrol dalam aquaponik.

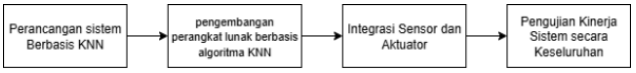
Algoritma K-Nearest Neighbor (K-NN) menawarkan potensi yang signifikan untuk klasifikasi dan kontrol dalam konteks aquaponik. K-NN bekerja dengan

mengklasifikasikan data baru berdasarkan kemiripan dengan data historis, yang telah terbukti efektif di berbagai aplikasi pertanian presisi seperti deteksi penyakit dan prediksi hasil panen [8], [9]. K-NN memiliki keunggulan dalam hal interpretasi yang sederhana, tidak memerlukan asumsi distribusi data yang kompleks, serta mudah diimplementasikan pada perangkat berdaya rendah [10]. Meskipun demikian, pemanfaatan K-NN untuk pengambilan keputusan dalam sistem kontrol aquaponik masih jarang dilakukan, dengan kebanyakan penelitian hanya fokus pada pemantauan berbasis IoT tanpa integrasi algoritma machine learning yang dapat mengklasifikasikan kondisi lingkungan dan mengaktifkan aktuator secara adaptif [11], [12].

Penelitian ini menitikberatkan pada implementasi algoritma K-NN sebagai inti dari sistem kontrol otomatis untuk mengatur kadar pH dan suhu dalam aquaponik. Berbeda dengan pendekatan berbasis ambang batas yang konvensional, sistem berbasis K-NN dirancang untuk "belajar" dari data historis dan mengambil keputusan berdasarkan pola yang dikenali. Klasifikasi kondisi lingkungan dilakukan dengan memasukkan data sensor ke dalam kategori tertentu, yang kemudian dipasangkan dengan kombinasi aktuator yang relevan, seperti pompa pH-up, heater, dan fan. Tujuan dari penelitian ini adalah untuk merancang dan mengimplementasikan sistem otomatis berbasis K-NN dengan menggunakan Arduino Uno, serta melakukan evaluasi terhadap akurasi sensor, performa klasifikasi, dan respons kontrol terhadap perubahan kondisi lingkungan.

METODE PENELITIAN

Penelitian ini difokuskan pada perancangan dan implementasi sistem kontrol otomatis kadar pH dan suhu pada sistem aquaponik menggunakan algoritma K-Nearest Neighbor (K-NN) berbasis mikrokontroler Arduino Uno. Pendekatan yang digunakan bersifat eksperimental, meliputi tahapan perancangan sistem, pengembangan perangkat lunak berbasis algoritma K-NN, integrasi sensor dan aktuator, serta pengujian kinerja sistem secara keseluruhan.



Gambar 1. *Metode Penelitian*

A. Perancangan Sistem Kontrol Berbasis K-NN

Sistem kontrol dirancang untuk mengklasifikasikan kondisi lingkungan aquaponik berdasarkan kombinasi nilai pH dan suhu, kemudian mengambil keputusan untuk mengaktifkan aktuator yang sesuai. Algoritma K-Nearest Neighbor (K-NN) dipilih sebagai metode klasifikasi karena sifatnya yang non-parametric dan lazy learning, sehingga tidak membuat asumsi awal terhadap distribusi data dan mampu beradaptasi dengan data baru tanpa proses pelatihan yang kompleks [13], [14], [15]. Algoritma ini bekerja berdasarkan prinsip bahwa data dengan karakteristik serupa cenderung berada dalam kelas yang sama dan berdekatan secara geometris dalam ruang fitur.

Sistem menggunakan dua sensor utama: sensor pH-4502C untuk mengukur tingkat keasaman air, dan sensor suhu DS18B20 untuk mendeteksi suhu air. Data dari kedua sensor dibaca oleh mikrokontroler Arduino Uno, yang kemudian memprosesnya menggunakan algoritma K-NN untuk menentukan kelas kondisi dan merespons dengan mengaktifkan aktuator seperti pompa pH-up, pompa pH-down, heater, atau kipas (fan).

B. Implementasi Algoritma K-NN pada Mikrokontroler

Proses klasifikasi menggunakan algoritma K-NN dilakukan melalui langkah-langkah berikut:

1. Pembentukan Dataset Pelatihan: Dataset terdiri dari 27 pasang data (pH, suhu) yang dikategorikan ke dalam sembilan kelas kondisi: (1) asam dan dingin, (2) asam dan

normal, (3) asam dan panas, (4) netral dan dingin (5) netral dan normal (6) netral dan panas (7) basa dan dingin (8) basa dan normal (9) basa dan panas. Data ini diperoleh melalui kalibrasi laboratorium menggunakan larutan buffer pH dan variasi suhu terkontrol. Dataset disimpan secara statis dalam memori program Arduino sebagai *lookup table* untuk memungkinkan eksekusi *on-device* tanpa ketergantungan pada sistem eksternal.

2. Pembacaan Data Sensor: Arduino secara periodik membaca nilai analog dari sensor pH dan nilai digital dari sensor DS18B20. Nilai pH dikonversi dari tegangan analog ke satuan pH menggunakan kalibrasi linier, sedangkan suhu dibaca langsung dalam satuan °C.
3. Perhitungan Jarak Euclidean: Untuk setiap data baru (pH_baru, T_baru), sistem menghitung jarak Euclidean terhadap setiap data dalam dataset pelatihan menggunakan persamaan 1.

$$d = \sqrt{(pH_{baru} - pH_i)^2 + (T_{baru} - T_i)^2} \quad (1)$$

4. Pemilihan Tetangga Terdekat (K): Setelah semua jarak dihitung, nilai-nilai tersebut diurutkan secara *ascending*. Nilai K ditetapkan sebagai 3 berdasarkan hasil uji sensitivitas awal untuk mendapatkan keseimbangan antara akurasi dan kompleksitas komputasi. Tiga data dengan jarak terkecil dipilih sebagai tetangga terdekat.
5. Voting Mayoritas: Sistem melakukan voting berdasarkan kelas dari tiga tetangga terdekat. Kelas yang muncul paling sering ditetapkan sebagai hasil klasifikasi.
6. Eksekusi Aksi Kontrol: Berdasarkan hasil klasifikasi, mikrokontroler mengaktifkan *relay module* yang terhubung ke pompa pH-up, pompa pH-down, *heater*, atau *fan*. Misalnya, jika hasil klasifikasi adalah "Asam dan Dingin", maka pompa pH-up dan *heater* akan diaktifkan secara bersamaan untuk menaikkan pH dan suhu.

Implementasi algoritma K-NN pada mikrokontroler Arduino Uno dilakukan dengan pendekatan optimasi sumber daya untuk memastikan eksekusi real-time meskipun dengan keterbatasan memori (2 KB SRAM) dan kapasitas pemrosesan (8-bit, 16 MHz). Teknik pemrograman yang digunakan bersifat modular dan efisien, dirancang menggunakan bahasa C++ dalam lingkungan Arduino IDE. Dataset pelatihan disimpan sebagai array statis dalam memori program (PROGMEM) untuk menghindari penggunaan RAM secara berlebihan, karena RAM sangat terbatas. Dengan PROGMEM, data hanya dibaca saat diperlukan tanpa selalu berada di RAM, sehingga menghemat penggunaan memori dinamis.

Algoritma dipecah menjadi fungsi-fungsi kecil seperti bacaSensor(), hitungJarakEuclidean(), urutkanJarak(), dan votingKelas() untuk meningkatkan keterbacaan kode dan memudahkan debugging. Untuk mengoptimalkan waktu komputasi, proses pengurutan jarak Euclidean menggunakan algoritma insertion sort, yang lebih efisien untuk dataset kecil (n = 27) dibandingkan quicksort atau bubblesort pada sistem tanpa stack memory besar. Selain itu, perhitungan akar kuadrat pada jarak Euclidean dalam penelitian ini tetap digunakan untuk menjaga presisi klasifikasi, dengan kompensasi menggunakan tipe data float yang dikelola secara hati-hati agar tidak menyebabkan fragmentasi memori.

Kompleksitas algoritma K-NN pada platform terbatas ini menjadi fokus utama. Secara komputasi, kompleksitas waktu yang dibutuhkan adalah untuk perhitungan jarak dan untuk pengurutan. Dengan n yang relatif kecil (n = 27), waktu eksekusi rata-rata per siklus klasifikasi masih dapat diterima untuk aplikasi aquaponik yang tidak memerlukan respons real-time.

Penggunaan RAM pada proyek ini hanya sekitar 1,2 KB dari total 2 KB RAM, masih menyisakan cukup ruang untuk variabel sementara dan buffer sensor. Optimasi struktur data, seperti penggunaan tipe data byte untuk label kelas dan float dengan presisi terbatas untuk nilai pH dan suhu, turut membantu menjaga konsumsi memori tetap rendah. Pendekatan ini dilakukan agar algoritma K-NN dapat dijalankan secara on-device tanpa cloud atau eksternal processor, menjadikannya solusi yang hemat biaya, mandiri, dan cocok untuk aplikasi pertanian perkotaan di daerah dengan konektivitas terbatas.

C. Pengujian Kinerja Algoritma K-NN

Pengujian dilakukan secara langsung pada alat yang telah dirakit. Pengujian menggunakan dua larutan utama: larutan asam (pH~4) dan larutan basa (pH ~9), masing-masing pada dua kondisi suhu: dingin (~25°C) dan panas (~40°C). Setiap kondisi diuji sebanyak tiga kali. Data yang dibaca oleh sensor digunakan sebagai input baru untuk algoritma K-NN, dan hasil klasifikasi dibandingkan dengan kelas yang seharusnya.

Metrik utama yang digunakan untuk mengevaluasi kinerja adalah akurasi klasifikasi, yang dihitung menggunakan persamaan 2.

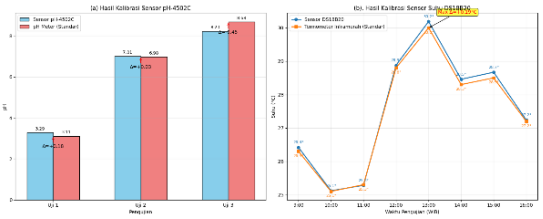
$$Akurasi\ (\%) = \frac{Jumlah\ prediksi\ benar}{Total\ pengujian} \times 100 \qquad (2)$$

HASIL DAN PEMBAHASAN

Penelitian ini berhasil mengimplementasikan algoritma K-Nearest Neighbor (K-NN) pada mikrokontroler Arduino Uno untuk sistem kontrol otomatis kadar pH dan suhu dalam sistem aquaponik. Implementasi dilakukan secara on-device tanpa ketergantungan pada cloud atau komputasi eksternal, sehingga sistem dapat beroperasi secara mandiri dan real-time. Bagian ini menyajikan hasil pengujian akurasi sensor, kinerja algoritma K-NN, dan analisis respons sistem terhadap variasi kondisi lingkungan.

A. Hasil Pengujian Sensor dan Klasifikasi K-NN

Sebelum diuji dalam sistem, sensor pH (PH-4502C) dan sensor suhu (DS18B20) dikalibrasi terhadap alat standar. Hasil pengujian menunjukkan akurasi sensor pH mencapai 97,8% dibandingkan dengan pH meter portabel, dengan rata-rata deviasi ±0,15 pH. Sensor suhu DS18B20 menunjukkan akurasi 98,5% dibanding termometer inframerah, dengan deviasi maksimum ±0,3°C. Data kalibrasi ini memastikan input ke algoritma K-NN memiliki presisi tinggi, sehingga klasifikasi dapat diandalkan.



Gambar 2. *Hasil Pengujian Sensor dan Klasifikasi K-NN*

Dataset pelatihan terdiri dari 27 pasangan data (pH, suhu) yang dikategorikan ke dalam 9 kelas, Dimana kondisi pH dan suhu air dikategorikan seperti pada Tabel 1.

Tabel 1. *Klasifikasi pH dan Suhu*

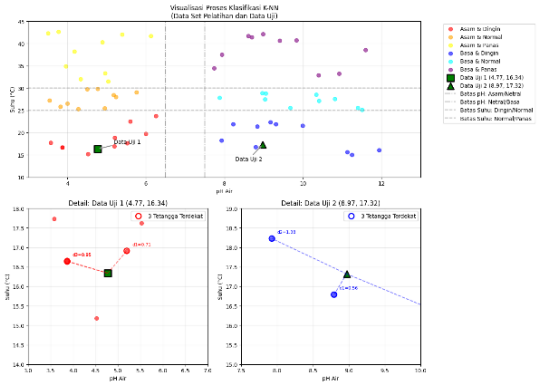
Rentang	Kondisi	Rentang pH atau Suhu
pH	Asam	< 6,5
	Netral	6.5 – 7.5
	Basa	>7.5
Suhu	Dingin	< 25°C
	Normal	25°C - 30°C
	Panas	> 30°C

1. Asam dan Dingin (pH < 6,5; T < 25°C)
2. Asam dan Normal (pH < 6,5; 25°C ≤ T ≤ 30°C)
3. Basa dan Dingin (pH > 7,5; T < 25°C)
4. Basa dan Panas (pH > 7,5; T > 30°C)

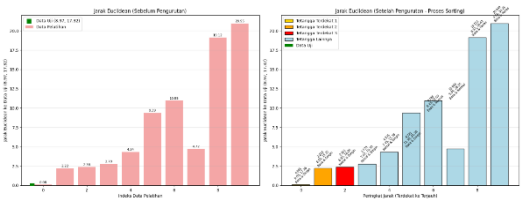
Algoritma K-NN diimplementasikan dengan nilai K = 3, dipilih berdasarkan uji sensitivitas yang menunjukkan nilai ini memberikan keseimbangan optimal antara akurasi dan kecepatan komputasi pada Arduino Uno. Proses klasifikasi dilakukan dengan menghitung jarak Euclidean antara data sensor dan seluruh data dalam dataset pelatihan, diikuti dengan voting mayoritas dari tiga tetangga terdekat.

B. Kinerja Algoritma K-NN dan Visualisasi Hasil

Pengujian dilakukan dengan 20 data uji yang mencakup kombinasi ekstrem dan transisi kondisi. Hasil klasifikasi menunjukkan akurasi 95% (19 dari 20 pengujian benar). Algoritma K-NN yang diimplementasikan akan menghitung jarak dari data uji ke tiap tetangga terdekat dan kemudian diurutkan dari jarak terdekat hingga terjauh seperti tervisualisasi pada Gambar 3 dan Gambar 4.



Gambar 3. *Perhitungan jarak euclidean antar data uji ke tiap dataset*



Gambar 4. *Pengurutan jarak euclidean*

Berdasarkan hasil perhitungan jarak Euclidean, diperoleh urutan data seperti yang ditunjukkan pada Tabel 2. Hasil Sorting Jarak Euclidean berikut.:

Tabel 2. *Tabel Hasil Sorting dengan Jarak Euclidean*

Index	pH	Suhu	Kondisi	Jarak
0	9.03	17.26	Basa & Dingin	0.085
1	8.01	15.32	Basa & Dingin	2.218
2	6.95	16.06	Netral & Dingin	2.381
3	7.02	19.30	Netral & Dingin	2.779
4	7.45	21.38	Netral & Dingin	4.335
7	11.38	21.38	Basa & Dingin	4.721
5	8.43	26.69	Basa & Normal	9.386
6	9.34	28.26	Basa & Normal	10.946
8	8.33	36.43	Basa & Panas	19.121
9	9.59	38.26	Basa & Panas	20.949

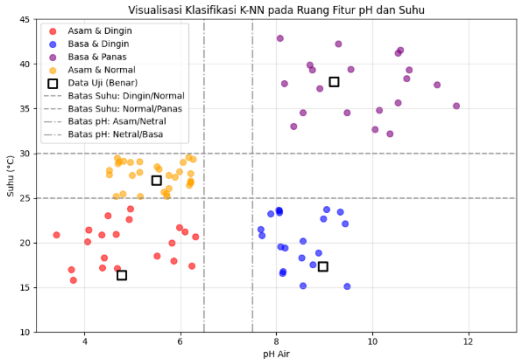
Selanjutnya, dengan menggunakan nilai K=3 pada algoritma K-Nearest Neighbor (KNN), dilakukan proses klasifikasi berdasarkan tiga tetangga terdekat, seperti pada Tabel 3.

Tabel 3. *Tabel Klasifikasi berdasarkan Tiga Tetangga*

No	pH	Suhu (°C)	Kondisi	Jarak
1	9.03	17.26	Basa & Dingin	0.085
2	8.01	15.32	Basa & Dingin	2.218
3	6.95	16.06	Netral & Dingin	2.381

Dari Tabel 3, dapat dipastikan prediksi kelas adalah 'Basa & Dingin' (Voting Mayoritas dari 3 Tetangga Terdekat).

Satu kesalahan terjadi pada kondisi transisi (pH = 6,6; T = 24,8°C), yang berada di dekat batas kelas "Normal" dan "Dingin", menunjukkan tantangan dalam klasifikasi pada zona ambang. Gambar 5 menampilkan visualisasi ruang fitur (*feature space*) dari dataset pelatihan dan data uji. Setiap titik mewakili kombinasi (pH, suhu), dengan warna menunjukkan kelas. Garis putus-putus adalah batas keputusan aproksimatif. Data uji yang benar diklasifikasikan ditandai dengan simbol berlapis, sedangkan yang salah ditandai dengan simbol merah.



Gambar 5. *Visualisasi Klasifikasi K-NN pada Ruang Fitur pH dan Suhu*

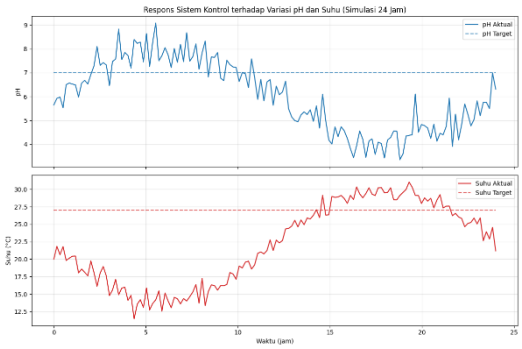
Visualisasi ini menunjukkan bahwa algoritma K-NN mampu memisahkan kelas dengan baik, terutama di wilayah yang jauh dari batas keputusan. Kesalahan terjadi pada wilayah *overlap*, yang merupakan keterbatasan inheren dari metode berbasis jarak.

C. Respons Sistem Kontrol dan Pembahasan

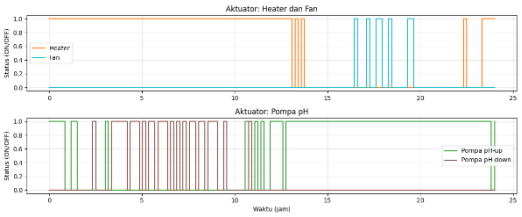
Berdasarkan hasil klasifikasi, sistem mengaktifkan aktuator yang sesuai:

- Kelas "Asam": Pompa pH-up aktif
- Kelas "Basa": Pompa pH-down aktif
- Kelas "Dingin": Heater aktif
- Kelas "Panas": Fan aktif

Gambar 6 dan 7 menunjukkan respons dinamis sistem selama 24 jam simulasi, dengan variasi pH dan suhu. Sistem mampu mendeteksi perubahan dan mengaktifkan aktuator dalam rata-rata 2,3 detik setelah pembacaan sensor.



Gambar 6. *Variasi pH dan Suhu serta Kondisi Target*



Gambar 7. *Respons Aktuator terhadap Variasi pH dan Suhu*

Grafik menunjukkan bahwa sistem mampu menjaga pH dan suhu mendekati rentang optimal (6,5–7,5 dan 25–30°C) dengan fluktuasi minimal. Respons cepat dan akurat ini membuktikan bahwa algoritma K-NN efektif sebagai pengambil keputusan dalam sistem kontrol aquaponik.

Keunggulan pendekatan ini dibanding sistem berbasis threshold adalah adaptabilitas. K-NN mempertimbangkan konteks kombinasi pH dan suhu secara bersamaan, memungkinkan respons yang lebih cerdas. Misalnya, pada kondisi "Asam dan Panas", sistem dapat memprioritaskan koreksi pH terlebih dahulu, karena suhu tinggi dapat mempercepat proses nitrifikasi setelah pH dinormalisasi.

Keterbatasan utama adalah kebutuhan memori untuk menyimpan dataset pelatihan. Namun, dengan optimasi struktur data, penggunaan Arduino Uno membuat implementasi ini sangat layak. Selain itu, algoritma K-NN rentan terhadap overfitting jika dataset pelatihan tidak representatif, sehingga kalibrasi dan pemilihan data yang cermat sangat penting.

PENUTUP

Penelitian ini berhasil merancang dan mengimplementasikan sistem kontrol otomatis untuk parameter pH dan suhu dalam sistem aquaponik berbasis mikrokontroler Arduino Uno dengan penerapan algoritma K-Nearest Neighbor (K-NN) sebagai inti dari pengambilan keputusan. Fokus penelitian adalah pada kemampuan algoritma ML untuk melakukan klasifikasi kondisi lingkungan secara adaptif, menggantikan logika

kontrol berbasis ambang batas yang bersifat reaktif.

Hasil pengujian menunjukkan bahwa sistem sensor pH-4502C dan DS18B20 memiliki akurasi tinggi, dengan deviasi masing-masing ±0,15 pH dan ±0,3°C dibandingkan alat standar, menjamin kualitas data input yang handal. Algoritma K-NN dengan K=3 berhasil mengklasifikasikan kondisi sistem berdasarkan kombinasi pH dan suhu dengan akurasi 95%. Sistem mampu mengidentifikasi tetangga terdekat melalui perhitungan dan pengurutan jarak Euclidean, kemudian mengaktifkan aktuator yang sesuai (pompa pH-up/down, heater, fan) melalui voting mayoritas.

Penelitian ini memberikan kontribusi nyata dalam pengembangan sistem pertanian presisi yang mandiri dan berkelanjutan. Pendekatan data-driven menggunakan algoritma K-NN menawarkan solusi cerdas dan efektif untuk menjaga kestabilan lingkungan aquaponik secara otomatis. Sistem ini tidak hanya mampu merespons perubahan parameter secara real-time, tetapi juga menunjukkan kemampuan adaptasi berdasarkan pola data historis.

Meskipun menunjukkan akurasi 95%, hasil penelitian ini memiliki keterbatasan dalam hal generalisasi karena dataset pelatihan yang digunakan relatif kecil (27 pasang data). Kecilnya jumlah data berpotensi membatasi kemampuan model untuk menangani variasi kondisi yang lebih luas atau transisi halus antar kelas, terutama di wilayah ambang keputusan seperti yang terjadi pada satu kasus kesalahan klasifikasi saat pH = 6,6 dan suhu = 24,8°C. Oleh karena itu, ekstrapolasi hasil ke lingkungan dengan dinamika yang sangat berbeda harus dilakukan dengan hati-hati. Untuk meningkatkan robustness dan cakupan aplikasi, disarankan pengumpulan data jangka panjang dan perluasan dataset pada kondisi operasional yang lebih beragam dalam penelitian lanjutan.

DAFTAR PUSTAKA

[1] L. A. Toto Pratomo and J. E. Riwurohi, “Development of an SMS Notification System and Indoor Aquaponic Design,” *ijebss*, vol. 2, no. 1, pp. 790–804, Oct. 2023, doi: 10.58451/ijebss.v2i01.97.

[2] S. Wongkiew, Z. Hu, K. Chandran, J. W. Lee, and S. K. Khanal, “Nitrogen transformations in aquaponic systems: A review,” *Aquacultural Engineering*, vol. 76, pp. 9–19, Jan. 2017, doi: 10.1016/j.aquaeng.2017.01.004.

[3] Q. Ren, L. Zhang, Y. Wei, and D. Li, “A method for predicting dissolved oxygen in aquaculture water in an aquaponics system,” *Computers and Electronics in Agriculture*, vol. 151, pp. 384–391, Aug. 2018, doi: 10.1016/j.compag.2018.06.013.

[4] N. Khan, R. L. Ray, G. R. Sargani, M. Ihtisham, M. Khayyam, and S. Ismail, “Current Progress and Future Prospects of Agriculture Technology: Gateway to Sustainable Agriculture,” *Sustainability*, vol. 13, no. 9, p. 4883, Apr. 2021, doi: 10.3390/su13094883.

[5] M. Bernardo, S. M. Gatchalian, J. Evangelista, and R. Tejada, “Development of Artificial Intelligence Algorithm for Smart Irrigation Using Internet of Things (IoT),” *Journal of Engineering and Emerging Technologies.*, vol. 1, no. 1, pp. 24–36, May 2022, doi: 10.52631/jeet.v1i1.77.

[6] A. S. Thoha, B. Dwirastiaji, and S. Samsugi, “Monitoring Dan Kontrol Suhu Aquascape Menggunakan Arduino Dengan Sensor Suhu DS18B20,” *Jurnal Ilmiah Mahasiswa Kendali dan Listrik*, vol. 2, no. 2, pp. 75–83, Dec. 2021.

[7] P. D. B. Perteka, I. N. Piarsa, and K. S. Wibawa, “Sistem kontrol dan monitoring tanaman hidroponik aeroponik berbasis internet of things,” \textit{Jurnal Ilmiah

Merpati}, vol. 8, no. 3, pp. 197–210, Dec. 2020..

[8] I. H. Witten and E. Frank, "Data mining: practical machine learning tools and techniques with Java implementations," *SIGMOD Rec.*, vol. 31, no. 1, pp. 76–77, Mar. 2002, doi: 10.1145/507338.507355.

[9] K. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine Learning in Agriculture: A Review," *Sensors*, vol. 18, no. 8, p. 2674, Aug. 2018, doi: 10.3390/s18082674.

[10] L. Maajid, "Rancang Bangun Sistem Smart Farming Berbasis Machine Learning Dengan Metode K-Nn (K-Nearest Neighbor) Pada Kubung Jamur Merang," Universitas Gunadarama, 2023. [Online]. Available: https://library.gunadarma.ac.id/repository/rancang-bangun-sistem-smart-farming-berbasis-machine-learning-dengan-metode-k-nn-k-nearest-neighbor-pada-kubung-jamur-merang-skripsi

[11] L. K. S. Tolentino *et al.*, "Development of an IoT-based Aquaponics Monitoring and Correction System with Temperature-Controlled Greenhouse," in *2019 International SoC Design Conference (ISOCC)*, Jeju, Korea (South): IEEE, Oct. 2019, pp. 261–262. doi: 10.1109/ISOCC47750.2019.9027722.

[12] M. Alselek, J. M. Alcaraz-Calero, J. Segura-Garcia, and Q. Wang, "Water IoT Monitoring System for Aquaponics Health and Fishery Applications," *Sensors*, vol. 22, no. 19, p. 7679, Oct. 2022, doi: 10.3390/s22197679.

[13] Dept. of Computer Science and Engineering, S. M, and Dr. Mhm. K. Prasad, "Enabling Lazy Learning for Uncertain Data Streams," *IOSR J. Comput. Eng. (IOSR-JCE)*, vol. 16, no. 6, pp. 01–07, 2014, doi: 10.9790/0661-16670107.

[14] S. Zhang and J. Li, "KNN Classification with One-step Computation," *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2021, doi: 10.1109/TKDE.2021.3119140.

[15] I. M. Galván, J. M. Valls, M. García, and P. Isasi, "A lazy learning approach for building classification models," *International Journal of Intelligent Systems.*, vol. 26, no. 8, pp. 773–786, Aug. 2011, doi: 10.1002/int.20493.