

Sistem Pemantauan Kualitas Air Ikan Nila Media Bioflok Menggunakan Algoritma LSTM dan SVM

T. A Nugroho¹, Dina Angela², Yoel Wijaya Kristianto Hadikusumo³

^{1,2} Teknik Elektro, Institut Teknologi Harapan Bangsa, Jl. Dipati Ukur No. 80-84, Bandung 40132, Indonesia

³ Teknik Komputer, Institut Teknologi Harapan Bangsa, Jl. Dipati Ukur No. 80-84, Bandung 40132, Indonesia

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima Redaksi: 01-08-2025

Revisi Akhir: 10 September 2025

Diterbitkan Online: 10 September 2025

KATA KUNCI

Biofloc,

SVM,

LSTM,

Classification,

Prediction

KORESPONDENSI

E-mail: tunggul@ithb.ac.id

ABSTRACT

Tilapia farming using the biofloc system often encounters problems maintaining water quality stability, particularly due to fluctuations in temperature and pH that negatively impact fish health. To address this issue, this study aims to develop an Internet of Things (IoT)-based water quality monitoring system integrated with machine learning prediction and classification methods. The system was implemented using the MCU ESP8266, Atmega328, DS18B20 (temperature sensor), and PH-4502C (pH sensor). Historical sensor data were processed using the Long Short-Term Memory (LSTM) algorithm to predict values of water quality parameters. The prediction results were then classified using Support Vector Machine (SVM) into three categories: 'Water Too Acidic,' 'Good Water Quality,' and 'Water Too Alkaline.' The classification output was automatically sent to farmers via the Telegram application. The testing results demonstrated that the LSTM model achieved an R-squared value of 0.8871 with very low errors (MAE: 0.0115, MSE: 0.0019, RMSE: 0.0436), while the SVM classification model achieved an accuracy of 99.86%. These findings suggest that the developed system can support farmers in making quick and accurate decisions, thereby reducing fish mortality risks and improving time and labor efficiency. Future research is recommended to expand the system by integrating more water quality parameters and testing it in real pond environments for long-term performance evaluation.

1. PENDAHULUAN

Ikan nila (*Oreochromis niloticus*), spesies asli sungai Nil, telah menjadi komoditas budidaya global yang sangat penting, termasuk di Indonesia. Popularitasnya didorong oleh adaptabilitas yang tinggi, pertumbuhan yang cepat, dan nilai gizi yang baik. Salah satu inovasi dalam budidayanya adalah teknologi bioflok yang secara efisien memanfaatkan mikroorganisme untuk mengubah limbah organik seperti sisa makanan dan kotoran ikan menjadi sumber protein tambahan [1]. Meskipun demikian, penerapan metode ini menghadapi tantangan besar bagi peternak ikan nila itu sendiri, terutama dalam menjaga stabilitas kualitas air yang sering terganggu oleh perubahan temperature dan iklim yang dinamis [2]. Fluktuasi kondisi air ini dapat memicu perkembangan penyakit dan parasite

sehingga berisiko menurunkan kesehatan ikan dan mengancam keberhasilan panen [3], [4].

Fluktuasi iklim dan cuaca yang signifikan berubah menjadi tantangan utama dalam budidaya sistem bioflok karena berisiko menyebabkan destabilitas parameter yang krusial bagi kelangsungan hidup ikan dan mikroorganisme [3]. Parameter esensial, seperti tingkat keasaman (pH) sebesar (6,5-8,5) dan temperatur air (25-30°C), memainkan peran penting dalam menjaga keseimbangan pada kolam bioflok [5]. Pemantauan yang efisien yang diterapkan secara otomatis oleh sistem dapat membantu efisiensi pembudidayaan bagi peternak. Kesehatan ekosistem kolam bioflok dapat terjaga dengan optimal [6],[7], [8]. Sejumlah penelitian terdahulu telah berfokus pada pengembangan sistem monitoring kualitas air untuk budidaya ikan nila dalam sistem bioflok. Beberapa studi terdahulu, seperti yang dilakukan oleh mahasiswa Institut Teknologi Purwokerto menggunakan NodeMCU 8266 melalui aplikasi MyCayenne,

berhasil membangun sistem pemantauan untuk parameter dasar, seperti pH, temperatur, dan kelembaban [9], [10]. Pada tingkat yang lebih lanjut, penelitian di Universitas Brawijaya tidak hanya melakukan monitoring, tetapi juga mengintegrasikan sistem rekomendasi cerdas. Dengan memanfaatkan sensor temperature (DS18B20), pH (PH-4502c), dan turbidity (SEN-0189) yang dihubungkan ke mikrokontroler Arduino Mega 2560, penelitian tersebut juga menerapkan algoritma *random forest* untuk menganalisis data sensor dan menghasilkan rekomendasi pengelolaan kualitas air secara otomatis [11]. Secara kolektif, studi-studi ini menunjukkan adanya evolusi dari sistem monitoring sederhana menuju sistem pendukung keputusan yang lebih cerdas untuk membantu pembudidaya ikan.

Pengelolaan kualitas air yang buruk dapat memengaruhi kesehatan ikan dan produktivitas pada budidaya [12]. Ketepatan dan akurasi dalam kestabilan kualitas air adalah hal utama dalam budidaya ikan nila pada media bioflok. Oleh sebab itu, menjaga kualitas air secara otomatis oleh sistem sangat penting untuk memastikan pertumbuhan yang optimal dan mendapatkan hasil panen yang baik [2].

Dari permasalahan yang sudah dijabarkan di atas, terdapat dua solusi yang diusulkan dalam penelitian ini untuk mengembangkan sebuah sistem pemantauan kualitas air pada budidaya ikan nila media bioflok. Pertama, mengimplementasikan *hybrid* algoritma dengan menggabungkan model LSTM-SVM untuk memprediksi kualitas air kolam dengan mengklasifikasikan hasil dari model ke dalam 3 kategori air, yakni “Air Terlalu Asam”, “Kualitas Air Baik”, dan “Air Terlalu Basa”. Kedua, sistem dapat memberikan saran kepada peternak ikan untuk tindakan selanjutnya apabila kualitas air kolam bioflok tidak optimal atau tidak memenuhi standar kolam bioflok. Sistem notifikasi ini dikirimkan ke pengguna melalui aplikasi Telegram.

Penelitian ini bertujuan untuk merancang dan membangun sebuah sistem cerdas untuk memprediksi dan merekomendasikan tindakan terhadap kualitas air pada kolam bioflok. Sistem ini direalisasikan menggunakan arsitektur perangkat keras berbasis *internet of things* (IoT) yang terintegrasi dengan *microcontroller unit* (MCU) ATmega328 dan ESP8266, serta serangkaian sensor untuk akuisisi data. Data historis yang terkumpul kemudian diolah dan dimodelkan menggunakan metode *long short-term memory* (LSTM) untuk memprediksi kualitas air kolam [13] kemudian menggunakan *support support vector machine* untuk mengklasifikasikan kondisi air ke dalam tiga kategori. Hasil akhir dari prediksi kualitas air ini kemudian akan diklasifikasi dan diimplementasikan sebagai sistem peringatan dini yang secara otomatis mengirimkan notifikasi melalui aplikasi Telegram kepada peternak sehingga memungkinkan intervensi cepat untuk mencegah kerusakan fatal pada ekosistem kolam.

Hasil akhir pada penelitian ini diharapkan dapat membantu peternak dalam melakukan tindakan pencegahan awal sebelum kondisi kualitas air kolam bioflok memburuk dan dapat menyebabkan kematian. Manfaat lainnya bagi peternak ialah menghemat tenaga dan waktu peternak selama proses operasional kolam bioflok. Peternak tidak harus memantau kolam secara berkala, sistem yang akan memantau kolam secara terus-menerus.

2. TINJAUAN PUSTAKA

2.1. Model Algoritma *Long Short-Term Memory* (LSTM)

Pemilihan algoritma LSTM pada penelitian ini karena kemampuannya mengatasi keterbatasan dari arsitektur *recurrent neural network* (RNN) konvensional, struktur informasi yang adaptif dan unit memori jangka panjang, serta derivasi formal dari dasar-dasar *signal processing*. Pengembangan LSTM ini menambah stabilitas dan kemampuan dalam menangani keterikatan jangka panjang yang menjadi kelemahan RNN konvensional [14], [15]. Tabel I menunjukkan hasil kajian keunggulan dan kekurangan dari beberapa aspek dalam penggunaan model LSTM yang dilakukan dalam penelitian ini.

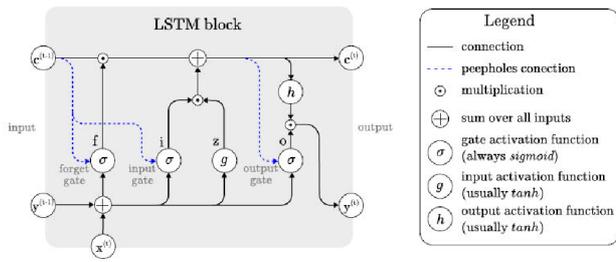
Pelatihan model menggunakan algoritma LSTM, jaringan yang dibuat untuk menyelesaikan permasalahan ketergantungan jangka panjang pada data yang berurutan secara *real-time*. LSTM juga mampu mengatasi masalah *vanishing gradient* yang sering muncul selama proses pelatihan RNN standar. Sebuah *cell* memori dalam jaringan LSTM dapat menyimpan data masuk dan keluar untuk periode yang panjang. Proses penyimpanan dan pengeluaran data ini diatur oleh tiga gerbang, yaitu *forget gate*, *input gate*, dan *output gate*. *Forget gate* menentukan informasi mana dari *cell state* sebelumnya (*previous cell state*) yang akan disimpan dan informasi mana yang akan dihilangkan dan tidak diperlukan lagi. *Input gate* menetapkan data mana yang perlu masuk ke dalam *cell state*, sementara *output gate* menentukan dan mengatur hasil keluaran. Gambar 1 merupakan gambar arsitektur LSTM.

2.2. Model Algoritma *Support Vector Machine* (SVM)

Pemilihan algoritma SVM pada penelitian ini karena kemampuan model dalam membaca data dan mengaplikasikannya secara *iterative*. Pengembangan model ini menyesuaikan dengan *hyperparameter* yang ada pada model hingga diperoleh model yang optimal untuk data yang diberikan. Proses pemilihan *hyperparameter* pada model bertujuan mendapatkan *hyperlane* maksimal serta margin dan matrik yang optimal untuk data secara efektif [16].

Alasan utama penggunaan algoritma SVM pada penelitian ini dibandingkan dengan algoritma klasifikasi dan regresi lainnya adalah model SVM bersifat sederhana, namun mampu bekerja dengan maksimal dengan kemampuan generalisasi yang tinggi. Kelebihan algoritma ini mencakup kemampuan dalam menangani data *nonlinear*. Model sederhana, tetapi bekerja dengan maksimal dengan kinerja yang baik pada dimensi yang tinggi dan fleksibilitas algoritma dalam ketersediaan implementasi yang luas [17].

Support vector machine (SVM) merupakan algoritma *supervised learning* yang dibagi menjadi dua jenis utama berdasarkan tugasnya, yaitu *support vector classification* (SVC) untuk mengklasifikasi dan *support vector regression* (SVR) untuk prediksi nilai kontinu [16]. Keunggulan fundamental SVM terletak pada kemampuannya untuk menangani data *nonlinear* melalui mekanisme *kernel*. Dengan mekanisme *kernel*, model SVM dapat me-



Gambar 1 Arsitektur LSTM [18]

Tabel I
Kelebihan dan Kekurangan Algoritma LSTM [14], [15]

Aspek	Kelebihan	Kekurangan
Memori	Unggul dalam mengingat pola masa lalu yang kompleks.	Waktu pelatihan lama, membutuhkan sumber daya komputasi yang baik.
Kemampuan Mempelajari Pola	Efektif untuk pola yang tidak linear dan dapat mempelajari hubungan rumit antar fitur.	Kurang efektif untuk data besar/kecil, membutuhkan sekuens untuk mempelajari pola dengan efektif.
Fleksibilitas	Mudah dikonfigurasi untuk menerima banyak fitur sekaligus.	Sulit untuk menginterpretasikan prediksi, dengan banyak parameter yang tidak mudah dipahami.
Kompleksitas Tuning	Bisa menangani berbagai hyperparameter dan fleksibel dalam konfigurasi.	Sulit menemukan arsitektur optimal karena banyaknya <i>hyperparameter</i> yang perlu diatur.
Penanganan Data	Dirancang untuk menangani data berurutan (sekuensial).	Rentan terhadap <i>overfitting</i> , karena dapat mengingat noise dan pola pada data pelatihan.

metakan data dalam ruang berdimensi tinggi untuk menemukan *hyperlane* pemisah yang optimal. Beberapa jenis *kernel* yang sering digunakan adalah *linear*, *polynomial*, dan *radial basis function* (RBF). Kinerja pada model dipengaruhi pada optimasi *hyperparameter* kunci, yaitu *C* yang mengontrol toleransi seberapa jauh kesalahan model dalam mengklasifikasi dan *gamma* yang mengidentifikasi pengaruh satu titik data pada *kernel* RBF [17].

Dalam penelitian ini, jenis SVM yang akan diimplementasikan adalah *support vector classification* (SVC) karena penelitian ini perlu mengklasifikasikan kualitas air pada tiga kategori yang berbeda. Kajian mengenai kelebihan dan kekurangan pada algoritma SVM yang dilakukan dalam penelitian ini dapat dilihat pada Tabel II.

2.3. Riset Terkait

Penelitian ini mengambil inspirasi dari beberapa riset yang terkait dari topik penelitian yang akan didalami. Penelitian dengan judul “Implementasi Algoritma *Naive Bayes* pada Sistem *Monitoring* Kualitas Air Kolam Lele Berbasis IoT”. Hasil pada penelitian ini menghasilkan akurasi algoritma *Naive Bayes* sebesar 70,95% dan secara signifikan menurunkan tingkat kematian ikan sebesar 40% dibandingkan sistem manual atau tanpa menggunakan metode bioflok [7].

Penelitian kedua adalah penelitian yang dilakukan oleh Jason Wijaya dkk. dengan mengembangkan sistem *monitoring* dan rekomendasi kualitas air kolam budidaya bibit ikan nila berdasarkan parameter kekeruhan, temperatur, dan pH dengan memanfaatkan algoritma *random forest*. Peneliti berhasil menghasilkan sistem dengan tingkat akurasi yang tinggi sebesar 94,44% untuk model *random forest*. Luaran dari penelitian ini

Tabel II
Kelebihan dan Kekurangan Metode SVM [17]

Aspek	Kelebihan	Kekurangan
Kemampuan Mengatasi Data	Mampu mengatasi data tidak linear dengan kernel dan margin soft.	Memerlukan penyesuaian parameter kernel untuk hasil optimal.
Klasifikasi Linear & Nonlinear	Dapat menangani keduanya dengan pemilihan kernel yang sesuai.	Pemilihan kernel yang salah dapat menurunkan performa model.
Kemampuan Generasi	Model cenderung lebih general dengan batasan margin yang besar.	Dapat menyebabkan <i>overfitting</i> jika parameter tidak diatur dengan benar.
Implementasi dalam Berbagai Bidang	Dapat digunakan dalam berbagai bidang sesuai kebutuhan.	<i>Tuning hyperparameter</i> harus hati-hati untuk hasil terbaik.
Bekerja pada Dimensi Tinggi	Efektif pada dimensi tinggi.	Perhitungan dan pelatihan menjadi berat jika dataset besar.

mencakup data dari hasil pengukuran air oleh sensor dan data rekomendasi yang dianalisis oleh algoritma *random forest* [11].

3. METODOLOGI

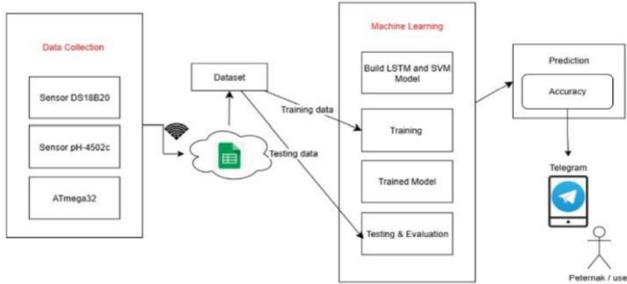
3.1. Identifikasi Masalah

Tantangan utama dalam budidaya ikan nila metode bioflok adalah perubahan cuaca dinamis yang memengaruhi parameter fisikokimia pada air. Ketidakstabilan suhu dan pH dapat mengganggu keseimbangan ekosistem, termasuk kadar oksigen terlarut yang esensial bagi ikan. Selain itu, metode pemantauan konvensional yang mengandalkan pengecekan manual kurang efisien, terutama untuk skala besar atau lokasi terpencil. Dengan sistem ini, pengukuran tidak hanya jauh lebih efisien dalam hal waktu, tetapi juga lebih akurat dan presisi dengan sistem yang digunakan. Sistem dapat mengurangi risiko kesalahan dalam pengukuran yang berpotensi fatal bagi organisme kolam.

3.2. Sistem Yang Diusulkan

Pada penelitian ini dirancang sistem yang dapat merekomendasikan tindakan yang dilakukan peternak terkait kualitas air kolam bioflok berdasarkan hasil analisis model prediksi dan klasifikasi. Sistem dirancang untuk mengklasifikasikan kualitas air kolam bioflok berdasarkan tiga kategori. Arsitektur sistem yang diusulkan dapat dilihat pada Gambar 2. Sistem ini dirancang untuk memantau kualitas air pada kolam ikan nila secara *real-time* dengan memanfaatkan dua buah sensor dan dilengkapi dengan penerapan algoritma *machine learning*.

Sistem memanfaatkan aplikasi Telegram sebagai sarana notifikasi yang efektif dan efisien. Dengan mengintegrasikan sensor kualitas air ke dalam sistem, data terkini mengenai parameter-parameter air seperti pH, temperatur, dan tingkat kekeruhan air dapat dipantau secara *real-time*. Ketika nilai parameter melebihi ambang batas yang telah ditetapkan, *bot* Telegram secara otomatis mengirimkan notifikasi kepada pengguna. Data yang sudah diolah diunggah ke *platform*



Gambar 2 Arsitektur sistem yang diusulkan

Telegram. Mekanisme pengiriman data dirancang untuk menghindari *spam* kepada pengguna, namun dapat memastikan peternak selalu mendapatkan informasi terkini mengenai kualitas air kolam.

3.3. Arsitektur Sistem

Penelitian ini dibagi ke dalam dua tahap, yaitu tahap akuisisi data dan tahap pengolahan data menggunakan model algoritma LSTM dan SVM. Pada tahap akuisisi data, data diambil dari kolam bioflok menggunakan sensor. Data yang sudah diambil sensor kemudian diolah oleh mikrokontroler ATmega328 dan selanjutnya diinputkan ke ESP8266 untuk dikirim ke *cloud platform*, *Google sheets*, untuk disimpan dan diproses lebih lanjut.

Proses *training* model dilakukan untuk melatih model agar mengenali pola dari data yang digunakan untuk memprediksi dan mengklasifikasi kualitas air kolam bioflok pada tingkat tertentu. Sementara itu, proses *testing* digunakan untuk menilai kinerja dari kedua model pada data yang belum pernah dipakai oleh model sebelumnya, dengan tipe dan jenis data yang sama.

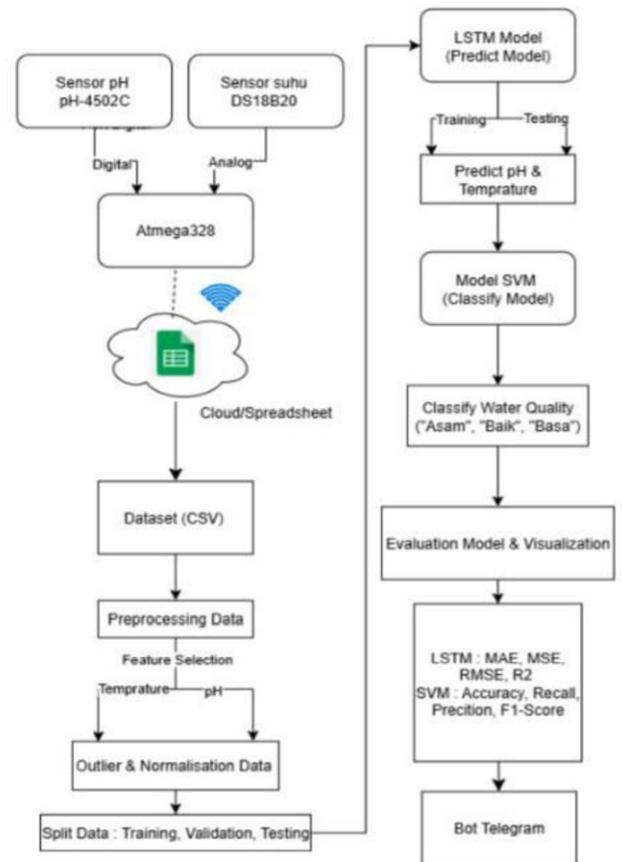
Pada tahap terakhir, model diintegrasikan pada sistem kemudian model yang sudah dilatih dan dievaluasi dengan *tuning* parameter disimpan pada perangkat lokal. *Bot* telegram mengakses model dan memanfaatkan model untuk memberikan prediksi atau keputusan langsung berdasarkan data IoT. Performa model dipantau secara berkala dan diperbarui jika diperlukan. Keseluruhan proses dari sistem dapat dilihat pada Gambar 3.

Perbedaan nilai data yang signifikan yang diakibatkan oleh parameter atau kelas dalam *dataset* dapat menyebabkan *overfitting* maupun hasil yang tidak optimal pada model. Oleh sebab itu, *dataset* dengan nilai dan keadaan yang tidak seimbang maupun seimbang, dilakukan proses normalisasi data dengan menerapkan *MinMaxScaler* pada proses *preprocessing* [19].

Penerapan *MinMaxScaler* sangat penting dalam proses *preprocessing* data, terutama ketika menggunakan model berbasis *neural network*, seperti LSTM maupun algoritma lain yang sensitif terhadap skala fitur, seperti SVM. *MinMaxScaler*

bekerja dengan cara mengubah nilai setiap fitur ke dalam rentang 0 hingga 1. Tanpa normalisasi, seperti *MinMaxScaler*, model dapat mengalami kesulitan dalam menemukan pola yang optimal, bahkan bisa gagal belajar jika ada perbedaan skala yang signifikan antar fitur ke dalam rentang [0, 1], seperti pada persamaan (1).

$$x' = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{1}$$



Gambar 3 Alur kerja sistem yang diusulkan

3.4. Perancangan Sistem

3.4.1. Perancangan Perangkat Keras

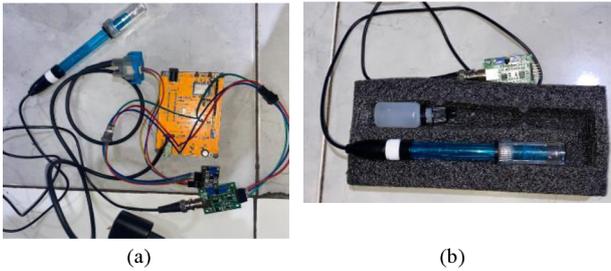
Dalam perancangan perangkat keras untuk implementasi sistem monitoring kualitas air kolam ikan nila media bioflok dibutuhkan mikrokontroler yang baik untuk digunakan secara berkala. Pemilihan mikrokontroler ATmega328 dan ESP8266 (Gambar 4a) pada sistem ini mempertimbangkan banyaknya sensor analog yang digunakan pada sistem, kestabilan, dan keandalan mikrokontroler, serta kompatibilitas dengan perangkat lain, juga diperlukannya tegangan berdaya rendah dan penambahan modul Wi-fi pada mikrokontroler untuk keperluan pengiriman data sensor ke *cloud*. Pemilihan sensor DS18B20 (Gambar 4b) sebagai sensor temperatur dan sensor pH-4502c sebagai sensor keasaman pada air dilakukan dengan mempertimbangkan keunggulan masing-masing dibandingkan dengan sensor sejenis lainnya yang ada di pasaran.

3.4.2. Perancangan Perangkat Lunak

Penelitian ini melibatkan aplikasi yang sudah ada untuk mendukung sistem berfungsi dengan baik, yaitu Telegram. Gambar 5 menggambarkan *flowchart* dari proses perancangan perangkat lunak, mulai dari proses pengambilan data oleh sensor sampai data diolah dan dapat ditampilkan kepada pengguna.

1) Googlesheets

Tahap pertama adalah pengumpulan dan pengiriman data dari sensor ke Googlesheets. Pada tahap ini perancangan perangkat lunak dimulai dengan mengumpulkan pembacaan data sensor. Setiap data yang masuk secara otomatis ditambahkan ke baris baru di Googlesheets



Gambar 4 (a) ATmega32 dan (b) sensor pH

dengan mencatat informasi waktu (stempel waktu), nilai pH, dan temperatur dari kolam.

2) Google *appscript*

Tahap kedua adalah Google *appscript* yang diprogram untuk memberikan notifikasi otomatis ke Telegram. Ketika ada data terbaru yang diambil oleh sensor, Google *appscript* akan memberi tahu *bot* Telegram yang telah dikonfigurasi sebelumnya. Selain itu, fungsi lain pada *appscript* adalah mengembangkan Googlesheets supaya terintegrasi dengan Telegram dan mengembangkan *interface* pada Telegram.

3) Telegram

Tahap ketiga adalah mengintegrasikan aplikasi Telegram yang memanfaatkan *bot* Telegram sebagai sistem kontrol sederhana pada sistem. *Bot* Telegram diimplementasikan sebagai *interface* yang berfungsi sebagai perantara antar pengguna dan hasil analisis model *machine learning*.

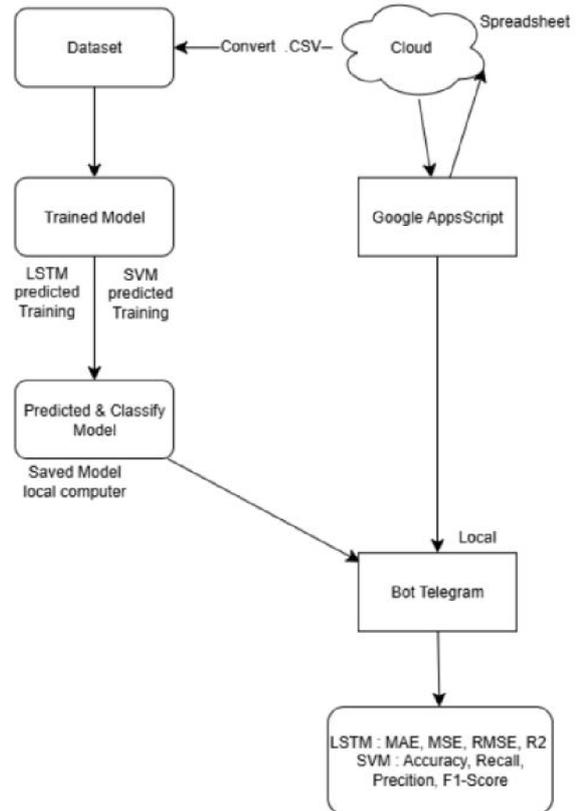
3.4.3. Alur Sistem *Machine Learning*

Data yang sudah dikumpulkan oleh sensor kemudian digunakan pada model untuk proses *training* dan *testing* [20]. Dalam merancang sistem dengan tujuan implementasi *machine learning* yang dibangun secara terstruktur dan runtun dengan mengintegrasikan dua model algoritma, LSTM dan SVM, dibutuhkan kejelasan tugas antar kedua model dalam menggabungkan model dalam kesatuan sistem.

Model LSTM memiliki peran utama dalam memodelkan pola temporal dan hubungan jangka panjang dalam data sekuensial air sehingga mampu mengidentifikasi tren dan fluktuasi yang sulit dideteksi secara langsung [20]. Hal ini sangat penting dalam prediksi kualitas air karena parameter-parameter tersebut dipengaruhi oleh fluktuasi musiman, cuaca, dan proses alami lainnya yang berlangsung secara berurutan. Penggunaan LSTM dalam sistem prediksi memastikan bahwa data historis digunakan secara optimal untuk memperkirakan keadaan masa depan secara

akurat dan mengurangi risiko kesalahan prediksi akibat hilangnya konteks temporal yang kritis [21].

Sementara itu, model SVM sebagai model klasifikasi yang mampu melakukan pengoptimalan *hyperplane* dengan margin terbesar untuk memisahkan data secara efektif. SVM terkenal karena kemampuannya menangani data dengan dimensi tinggi dan ketidaklinieran melalui *kernel* trick yang membuatnya sangat cocok untuk mengklasifikasi dan memperkirakan kualitas air yang memiliki pola *nonlinier* kompleks. Melalui proses pelatihan dan pengujian *tuning* parameter dalam memilih *kernel* yang tepat dalam penggunaan sistem, model SVM memungkinkan penyesuaian dan *tuning* prediksi dengan memperkuat



Gambar 5 *Flowchart* perancangan perangkat lunak

kemampuan generalisasi model, terutama ketika menghadapi data yang *noise* atau tidak lengkap [20], [22].

Melalui pendekatan ini, kelebihan masing-masing model dimanfaatkan untuk mendapatkan hasil prediksi yang optimal. Dengan demikian, kedua model yang diimplementasikan meningkatkan fungsionalitas untuk dua tugas yang berbeda pada model [21], [22].

3.4.4. Kebutuhan Informasi

Seluruh data yang telah diambil oleh perangkat yang diletakkan di kolam biofolk ikan nila secara periodik ditransmisikan pada *platform* berbasis *cloud*, yaitu Googlesheets. Sistem ini mengimplementasikan sebuah mekanisme notifikasi otomatis sebagai *interface* bagi peternak. Informasi dari Googlesheets dikirimkan secara langsung kepada pengguna melalui aplikasi Telegram.

3.5. Teknik Pengujian

Penelitian ini menerapkan metode pengujian *blackbox* yang melibatkan pengujian fungsionalitas suatu sistem tanpa harus mengetahui rincian implementasinya. Fungsi utama dalam pengujian ini adalah memverifikasi dan mengevaluasi bahwa sistem bekerja sesuai dengan yang diharapkan. Penerapan pengujian ini pada model *machine learning* berarti menguji model tanpa perlu mengetahui detail-detail implementasi dari model.

3.6. Skenario Pengujian

Skenario adalah langkah-langkah yang diterapkan untuk menguji sistem yang telah dibangun dengan tujuan mencapai hasil yang diinginkan. Langkah-langkah dari skenario pengujian dapat dilihat pada Gambar 6.

- 1) Pengambilan data dengan sensor
Tahap awal dalam pengujian model system, yakni pengambilan data dari sensor dan penyimpanan di *cloud Google sheets*. Pada tahap ini data yang diambil oleh sensor masih berupa data mentah kemudian format data diubah menjadi .csv yang nantinya akan diolah oleh model.
- 2) Pengolahan data (*preprocessing*)
Tahap selanjutnya adalah pengolahan data, seperti yang ditunjukkan pada Gambar 7, sebelum masuk proses pengujian model algoritma.
- 3) Pelatihan model LSTM dan SVM
Pada tahap ini model LSTM dan SVM dilatih menggunakan data *training* yang telah diproses untuk menemukan performa terbaik. Proses ini melibatkan serangkaian uji coba *tuning* parameter dengan berbagai kombinasi *hyperparameter*, seperti jumlah unit pada *layer* LSTM, jumlah *epoch*, jumlah *batch size*, dan nilai *dropout*, jenis *kernel* pada SVM, dan parameter *gamma* dan nilai *C* yang diuji secara sistematis. Tujuan utama pada tahap ini adalah pelatihan *iterative* untuk mencari dan menemukan konfigurasi parameter paling optimal yang akan digunakan oleh kedua model.
- 4) Evaluasi model LSTM dan SVM
Model dievaluasi menggunakan metrik kuantitatif yang sesuai, seperti MAE, MSE, RMSE, dan *R-squared*, untuk model prediksi LSTM dan *accuracy*, *precision*, *recall*, dan *F1-Score* untuk model klasifikasi SVM.

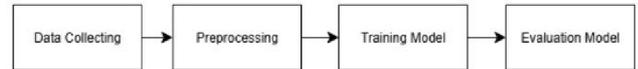
4. HASIL DAN PEMBAHASAN

4.1. Analisis Hasil Pengujian

4.1.1. Pengujian model LSTM dan SVM

Pengujian pada model *machine learning* bertujuan untuk menemukan kombinasi nilai parameter yang paling optimal agar model dapat mempelajari pola data dengan lebih baik. Proses ini penting karena setiap *dataset* memiliki karakteristik yang berbeda sehingga perlu mengatur parameter, seperti jumlah *layer*, *learning rate*, *batch size* pada model LSTM dan parameter *kernel*, nilai *C*, dan *gamma* pada model SVM.

Grid search adalah teknik optimasi *hyperparameter* yang memilih nilai satu per satu pada ruang *hyperparameter* dan mencobanya pada algoritma yang terfokus. *Grid search* dapat menjadi metode *tuning* yang berusaha untuk menghitung nilai ideal dari *hyperparameter*. Ini adalah pencarian komprehensif yang dilakukan pada nilai parameter tertentu dari suatu model. Model juga dikenal sebagai estimator [23].

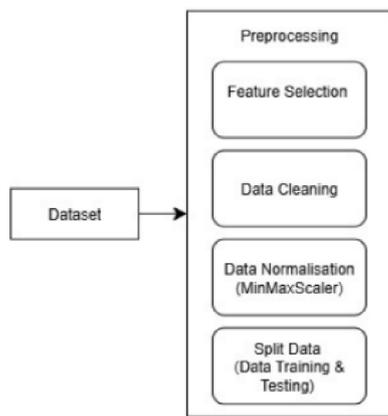


Gambar 6 Skenario pengujian sistem

Penggunaan *early stopping* dalam konteks *neural networks* yang tidak dilatih (*untrained*) untuk menyelesaikan masalah invers linier yang bersifat *ill-posed*. Penelitian ini secara teoretis membuktikan bahwa prinsip diskrepansi (*discrepancy principle*) dapat berfungsi sebagai metode *early stopping* yang efektif untuk menghentikan iterasi pelatihan sebelum terjadinya *overfitting*. Hasil numerik juga menunjukkan bahwa *early stopping* sangat penting dalam praktik, terutama ketika arsitektur jaringan yang digunakan sangat *over-parameterized*, karena iterasi yang berlebihan dapat menyebabkan solusi menyimpang dari solusi sebenarnya. Studi ini memberikan landasan teoretis pertama untuk penggunaan *early stopping* dalam metode berbasis *deep image prior* [24].

Berbagai kombinasi pengaturan parameter diuji untuk model LSTM dalam 256 percobaan berbeda. Setiap pengujian menggunakan konfigurasi jumlah unit yang berbeda pada dua *layer* LSTM, mulai dari 64–512 *unit layer* pertama dan 32–256 *unit layer* kedua. Selain itu, nilai *dropout* yang digunakan bervariasi antara 0,2, 0,3, 0,4 hingga 0,5 untuk mengurangi risiko *overfitting*. Jumlah *epoch* pelatihan juga diuji dari 50 dan 100, sedangkan *batch size* ditingkatkan dari 32 hingga 64 untuk melihat pengaruhnya terhadap stabilitas dan kecepatan pelatihan. Melalui proses *tuning* ini, dapat dievaluasi kombinasi parameter mana yang memberikan performa terbaik pada model untuk prediksi kualitas air.

Sama halnya dengan LSTM, berbagai kombinasi pengaturan diuji pada model SVM dalam 16 percobaan berbeda. Setiap pengujian menggunakan *kernel* yang berbeda, yaitu RBF (*radial basis function*) dan “linear” untuk mengeksplorasi pengaruh tipe fungsi *kernel* terhadap performa klasifikasi. Nilai parameter *C* yang berperan sebagai regularisasi divariasikan antara 0,1, 0,5, 1,0 hingga 2,0 untuk mengendalikan *trade-off* antara *margin* maksimal dan kesalahan klasifikasi. Parameter *gamma* juga diuji dengan pengaturan ‘*scale*’ dan ‘*auto*’ yang mempengaruhi seberapa jauh pengaruh satu data *training* terhadap lainnya. Dengan melakukan *tuning* pada kombinasi parameter-parameter ini, diharapkan dapat ditemukan konfigurasi SVM yang paling optimal dalam memisahkan kelas-kelas kualitas air berdasarkan fitur yang tersedia.



Gambar 7 Skema *preprocessing* sistem

4.1.2. Pengujian Performansi Model LSTM

Tuning parameter pada model LSTM dilakukan dengan pengujian sebanyak 256 iterasi yang komprehensif dan kompleks untuk mengidentifikasi konfigurasi arsitektur dengan performa paling optimal. Kinerja model terhadap 256 percobaan dievaluasi secara kuantitatif dengan serangkaian metrik evaluasi, yaitu MAE (*mean absolute error*), MSE (*mean squared error*), RMSE (*root mean squared error*), dan *R-squared*. Hasil evaluasi menunjukkan variasi performa yang signifikan. Beberapa konfigurasi menghasilkan model yang belum optimal yang ditandai dengan nilai *error* yang tinggi dan *R-squared* negatif. Namun, pada beberapa percobaan lain ditemukan bahwa model dengan arsitektur yang lebih kompleks dilatih dengan 100 *epoch* dan 64 *batch size* secara konsisten menunjukkan performa yang baik, yaitu nilai *error* yang kecil dan *R-squared* yang mendekati atau melebihi 0,8.

Tabel III menunjukkan nilai evaluasi metrik dari nilai kombinasi parameter *tuning* yang sudah dilakukan. Pada iterasi ke-94 terdapat nilai maksimum dari percobaan yang sudah dilakukan dengan nilai *R-squared* sebesar 0,8871. Nilai tersebut menunjukkan bahwa kombinasi parameter yang dicoba memiliki nilai yang optimal. Hal ini menandakan bahwa model sangat baik dalam memprediksi nilai di masa depan. Tabel III merupakan tabel evaluasi dari kombinasi nilai parameter yang paling optimal (<0,8).

4.1.3. Pengujian Performansi Model SVM

Tuning parameter pada model SVM dengan menguji sebanyak 16 iterasi yang komprehensif dan kompleks dilakukan untuk mengidentifikasi konfigurasi arsitektur dengan performa paling optimal. Nilai *recall*, *accuracy*, *precision*, dan *F1-Score* semuanya berada di atas 0,9986 pada setiap pengujian. Hal ini menandakan bahwa model mampu mengidentifikasi kelas dengan sangat baik, baik dari sisi sensitivitas (*recall*) maupun ketepatan (*precision*). *F1-score* yang juga sangat tinggi menunjukkan keseimbangan antara *recall* dan *precision* sehingga model tidak hanya akurat, tetapi juga stabil dalam mengenali masing-masing kelas. Konsistensi nilai evaluasi yang hampir sempurna ini mengindikasikan bahwa model SVM sangat efektif dalam memetakan data fitur ke kelas target pada *dataset* yang digunakan. Hasil evaluasi dari pengujian *tuning* parameter model SVM ditampilkan pada Tabel IV.

Tabel III

Evaluasi Metrik Optimal Model LSTM

Pengujian ke-	MAE	MSE	RMSE	<i>R-Squared</i>
94	0,0115	0,0019	0,0436	0,8871
116	0,0139	0,0023	0,0480	0,8746
56	0,0129	0,0022	0,0465	0,8683
164	0,0123	0,0022	0,0471	0,8604
252	0,0106	0,0014	0,0379	0,8590
185	0,0114	0,002	0,0442	0,8559
248	0,0125	0,0017	0,0411	0,8565
190	0,012	0,0015	0,039	0,8461
53	0,012	0,0017	0,0409	0,8470
87	0,0127	0,0016	0,0404	0,8430
95	0,0146	0,0023	0,0476	0,8423
Rata-rata	0,0124	0,0476	0,0433	0,8582
Nilai Maks	0,0146	0,0023	0,048	0,8871
Nilai Min	0,0106	0,0014	0,0379	0,8423

4.1.4. Analisis Hasil *Tuning* Parameter LSTM

Tabel V adalah parameter pada model LSTM yang akan digunakan. Parameter yang akan digunakan tersebut memiliki konfigurasi arsitektur yang dioptimalkan untuk menangkap pola data *time series*. Model ini terdiri dari dua *layer* LSTM dengan jumlah unit yang besar, yaitu 128 unit pada *layer* pertama dan 64 unit pada *layer* kedua, yang mampu mempelajari representasi fitur yang lebih mendalam. Nilai *dropout* sebesar 0,2 diterapkan untuk mengurangi risiko *overfitting* dengan cara mengacak sebagian neuron selama proses pelatihan. Proses *training* dilakukan selama 50 *epoch* dengan *batch size* sebesar 64 yang memungkinkan model belajar secara efisien dari jumlah data yang besar dalam setiap iterasi. *Optimizer* yang digunakan adalah “Adam” dengan *learning rate* 0,001 yang dikenal karena kemampuannya mempercepat konvergensi dan menjaga kestabilan pembaruan bobot.

Hasil evaluasi model LSTM pada *dataset training* yang diuji pada model menunjukkan nilai *error* yang relatif rendah, yaitu *mean absolute error* (MAE) sebesar 0,0115, *mean squared error* (MSE) sebesar 0,0019, dan *root mean squared error* (RMSE) sebesar 0,0436 dengan indikator yang paling krusial, yaitu nilai *R-squared* sebesar 0,8871. Dapat disimpulkan bahwa model LSTM berhasil mencapai performa prediksi yang sangat baik dengan nilai *R-squared* sebesar 0,8871 atau sebesar 88,71%.

4.1.5. Analisis Hasil *Tuning* Parameter SVM

Berdasarkan Tabel VI, dapat dijelaskan bahwa model SVM menggunakan *kernel* RBF (*radial basis function*) yang dikenal efektif untuk menangani data dengan pola nonlinear. Nilai parameter *C* yang digunakan adalah 1,0. Nilai ini berfungsi sebagai regularisasi untuk mengontrol *trade-off* antara *margin* maksimal dalam kesalahan klasifikasi. Parameter *gamma* diatur ke ‘scale’ yang secara otomatis menyesuaikan pengaruh tiap data *training* terhadap bentuk *decision boundary*. Model ini menggunakan dua fitur input utama, yaitu “Water Temp” dan “pH” untuk memprediksi kualitas air. Target label dikategorikan menjadi tiga kelas, yaitu “Air Terlalu Asam”, “Kualitas Air Baik”, dan “Air Terlalu Basa”, berdasarkan rentang nilai pH yang telah ditentukan melalui proses *binning*, yaitu pH < 6,4 untuk asam, 6,4 ≤ pH ≤ 8,4 untuk kualitas baik, dan pH > 8,4 untuk basa.

Model SVM yang sudah diuji dan dievaluasi menunjukkan tingkat akurasi yang sangat tinggi, yaitu sebesar 0,9986 atau 99,86%. Hasil ini dicapai dengan menggunakan konfigurasi standar yang kuat, yaitu *kernel radial basis function* (RBF) de-

Tabel IV
Evaluasi Metrik Optimal Model SVM

Pengujian ke-	Recall	Accuracy	Precision	F-1Score
6	0,9986	0,9986	0,9986	0,9986
1	0,9970	0,9970	0,9970	0,9970
2	0,9970	0,9970	0,9970	0,9970
3	0,9970	0,9970	0,9970	0,9970
8	0,9940	0,9940	0,9940	0,9940
Rata-rata	0,99672	0,99672	0,99672	0,99672
Nilai Maks	0,9986	0,9986	0,9986	0,9986
Nilai Min	0,9940	0,9940	0,9940	0,9940

ngan parameter regulasi *C* sebesar 1,0 dan *gamma* pada mode “scale”. Berdasarkan masukan yang akurat dari model LSTM dan definisi kelas yang jelas, model SVM berhasil menunjukkan kinerja klasifikasi yang sangat baik untuk mengklasifikasi kualitas air kolam bioflok ikan nila.

Gambar 8 merupakan visualisasi dari skor *loss* pada model prediksi selama proses *training* model LSTM (kurva garis biru). Penurunan yang konsisten ini mengidentifikasi model belajar dengan baik dari data *training*. Kurva *loss validation* (garis kuning) menunjukkan tren penurunan yang sejalan dengan *training loss*. Jarak yang konsisten antara kedua kurva menunjukkan tidak adanya *overfitting* yang signifikan. Penurunan kurva antara *training* dan *validation loss* menunjukkan arsitektur LSTM yang tepat untuk *datasets* yang digunakan. Tidak adanya *gap* yang relatif besar menunjukkan model tergeneralisasi dengan baik.

4.1.6. Hasil dan Analisis Model Prediksi LSTM

Gambar 9 menunjukkan grafik dari data aktual dan prediksi temperatur air pada model LSTM. Secara umum prediksi temperatur (garis kuning) mengikuti tren aktual (garis biru) dengan cukup baik. Model mampu menangkap pola fluktuasi temperatur sepanjang *time steps*. Kenaikan dan penurunan temperatur yang terjadi disebabkan oleh pola data aktual yang berubah secara drastis, bukan karena akibat *training* data. Dapat disimpulkan bahwa model LSTM berhasil mempelajari pola dari perubahan temperatur air dengan mempelajari tren historis pada data. Hal ini merupakan bukti bahwa model bekerja dengan baik dan tidak *overfitting*.

Gambar 10 menunjukkan grafik dari data aktual dan prediksi pH air pada model LSTM. Prediksi pH (garis kuning) hampir sempurna dengan tumpang tindih dengan nilai *actual* (garis biru). Perbedaan antara nilai prediksi dan aktual pH sangat kecil hampir di semua *time steps*. Model dengan konsisten memprediksi nilai pH pada air. Dapat disimpulkan bahwa model LSTM berhasil mempelajari pola dari perubahan pH air. Nilai pH pada *dataset* memiliki nilai yang periodik dan konsisten dibanding dengan ni-

Tabel V
Parameter Model LSTM

No.	Parameter	Nilai
1	LSTM Layer 1	128

No.	Parameter	Nilai
2	LSTM Layer 2	64
3	Dropout	0,5
4	Epoch	50
5	Optimizer	Adam
6	Batch_size	64
7	Scaler	Minmaxscaler

Tabel VI
Parameter Model SVM

No.	Parameter	Nilai
1	kernel	RBF (<i>radial basis function</i>)
2	C	1,0
3	Gamma	scale
4	Input Features	“Water Temp”, “pH”
5	Target Labels	{“Air Terlalu Asam”, “Kualitas Air Baik”, “Air Terlalu Basa”}
6	Target Bins	Inf [6,4, 8,4]

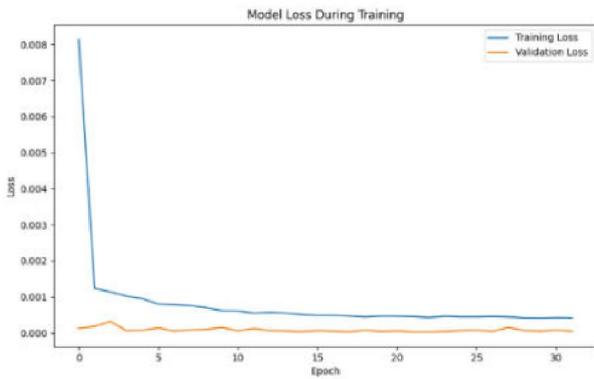
lai temperature pada *dataset*. Data pH memiliki lebih sedikit *noise* dibanding data temperatur.

4.1.7. Hasil dan Analisis Model Prediksi SVM

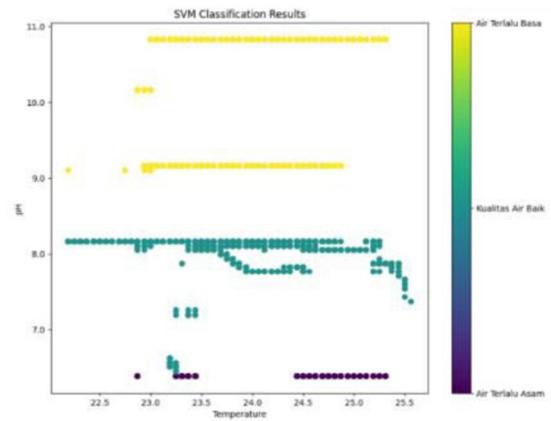
Pada Gambar 11 ditampilkan hasil dari *confussion matrix* model SVM. Semua data untuk kategori “Air Terlalu Asam” dan “Air Terlalu Basa” diprediksi dengan benar, tanpa kesalahan prediksi ke kelas lain. Terdapat 1.938 data kategori “Air Terlalu Asam” yang diprediksi sebagai kategori tersebut dengan benar; 8.214 data kategori “Kualitas Air Baik” yang diprediksi dengan benar dan 42 data kategori “Kualitas Air Baik” yang diprediksi sebagai kategori “Air Terlalu Asam”; dan 3.621 data yang diprediksi sebagai “Air Terlalu Basa” dengan benar. Hasil visualisasi data pada *confussion matrix* menunjukkan bahwa sebagian besar data diklasifikasikan dengan benar. Hanya ada sedikit kesalahan pada kelas “Kualitas Air Baik” sebanyak 42 data yang salah diklasifikasikan sebagai “Air Terlalu Asam”.

Gambar 12 merupakan grafik dari *scatter plot* pada model SVM. Pada model ini terdapat 3 jenis kategori kualitas air yang dibedakan dengan 3 warna pada grafik, yaitu ungu, hijau dan kuning. Terdapat sumbu *y* yang mewakili fitur persebaran nilai pH dengan batasan 6,0 sampai 11,0 dan sumbu *x* yang mewakili persebaran nilai temperatur dengan batasan 23,5°C sampai 29,5°C. Interpretasi “Air Terlalu Asam” (ungu) berada di area pH rendah, di bawah 6,4. Titik-titik ini tersebar di berbagai temperatur, tetapi selalu berada pada pH rendah. Pada “Kualitas Air Baik” (hijau), titik-titiknya berada di rentang pH sekitar 6,4 hingga 8,4. Pada “Air Terlalu Basa” (kuning), titik-titiknya berada di pH tinggi, di atas 8,4. Titik-titik ini tersebar di berbagai temperatur, tetapi selalu ada pada pH tinggi.

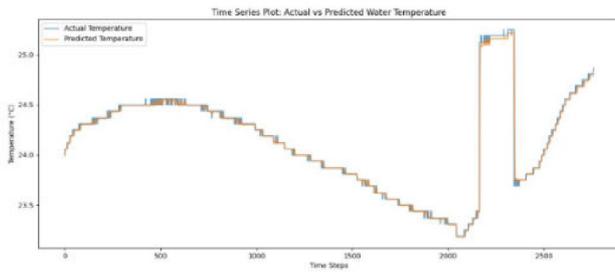
Scatter plot menunjukkan bahwa model SVM ini mampu mengklasifikasikan kualitas air dengan sangat baik berdasarkan nilai pH dan temperatur. Visualisasi ini menampilkan distribusi data nilai pH dan temperatur. Jika ada titik yang berada pada area



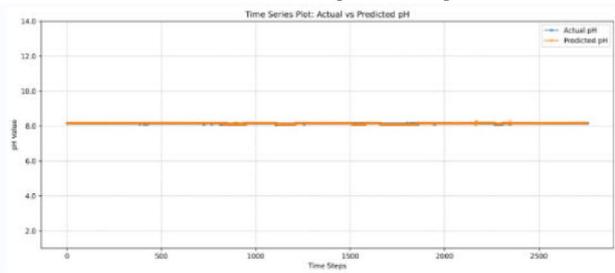
Gambar 8 Grafik nilai *loss over time training* model LSTM



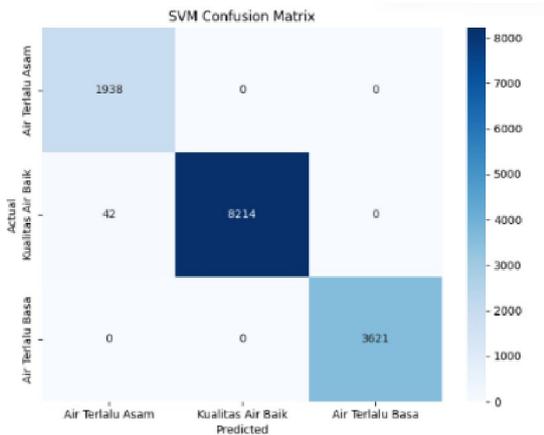
Gambar 12 *Scatter plots* model SVM



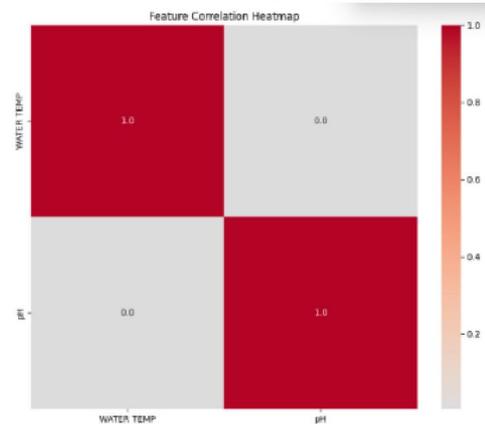
Gambar 9 Grafik LSTM prediksi temperatur



Gambar 10 Grafik LSTM prediksi pH



Gambar 11 *Confusion matrix* model SVM



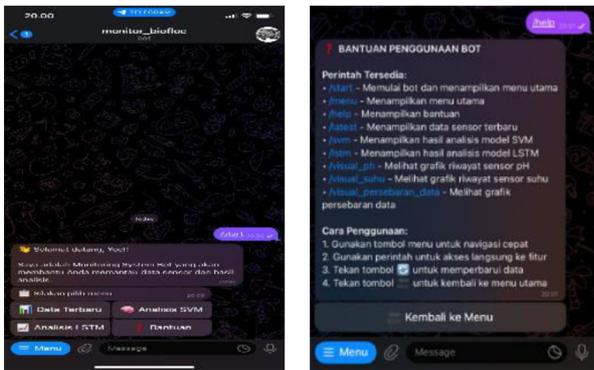
Gambar 13 *Feature correlation heatmap*

transisi antar warna, menandakan bahwa data dengan pH mendekati ambang batas kategori.

Berdasarkan gambar *heatmap correlation* pada Gambar 13, terlihat bahwa nilai korelasi antara parameter temperatur ("Water Temp") dan pH adalah 0,0. Hal ini menunjukkan bahwa tidak terdapat hubungan linier antara temperatur air dan nilai pH pada *dataset* yang digunakan. Dengan kata lain, perubahan temperatur air tidak berpengaruh secara langsung terhadap perubahan nilai pH dan sebaliknya. Korelasi yang sangat rendah (mendekati nol) ini menandakan kedua parameter tersebut tidak saling berkorelasi dalam konteks data yang dianalisis. Oleh karena itu, model prediksi yang dikembangkan ini berfokus pada parameter pH dan temperatur air saja, dengan memisahkan prediksi pada dua parameter yang terpisah.

4.1.8. Hasil dan Analisis Pengujian *Bot* Telegram

Hasil pengujian membuktikan bahwa *bot* Telegram ini sebagai *user interface* (UI) pada sistem yang sangat praktis bagi peternak



Gambar 14 Pengujian bot Telegram

ikan nila. Gambar 14 memperlihatkan contoh tampilan bot Telegram yang dibuat. Sistem peringatan dini ini dapat memberikan peternak kesempatan untuk melakukan tindakan antisipasi terjadinya hal yang lebih buruk.

5. SIMPULAN DAN SARAN

Berdasarkan pengujian yang sudah dilakukan, kesimpulan dari penelitian ini mencakup beberapa aspek pencapaian dalam merancang dan mengimplementasikan sistem *monitoring* kualitas air kolam bioflok dengan menggunakan metoda *machine learning* LSTM (*long short-term memory*) dan SVM (*support vector machine*). Penelitian ini berhasil membangun model algoritma LSTM untuk melakukan prediksi nilai parameter kualitas air kolam bioflok serta mengimplementasikan algoritma SVM dalam melakukan klasifikasi nilai parameter kualitas air ke dalam tiga kategori, yaitu “Air Terlalu Asam”, “Kualitas Air Baik”, dan “Air Terlalu Basa. Dari hasil pengujian, hasil model LSTM berhasil diimplementasikan dengan performa optimal yang menghasilkan nilai *R-squared* sebesar 0,8871, MAE 0,0115, MSE 0,0019, dan RMSE 0,0436. Sementara itu, pada model SVM berhasil mengklasifikasi performa sangat tinggi pada semua percobaan dan menghasilkan akurasi sebesar 99,86%.

Penelitian ini berhasil membangun sistem untuk merekomendasikan kualitas air kolam ikan bioflok yang tepat kepada peternak untuk menjaga kestabilan kualitas air supaya tetap optimal melalui bot Telegram. Berdasarkan hasil klasifikasi oleh model SVM, sistem akan mengirimkan pesan mengenai kualitas air kolam.

DAFTAR PUSTAKA

[1] Y. Cahyanti and I. Awalina, “Studi literatur: pengaruh suhu terhadap ikan nila (*Oreochromis niloticus*),” *J. Ilm. Pendidik. Sains dan Terap.*, vol. 2, no. 4, pp. 224–235, 2022, doi: 10.36312/pjipst.v2i4.110.

[2] D. H. Marisda and Anisa, “Penerapan teknologi bioflok budidaya ikan nila untuk pemanfaatan pekarangan rumah nonproduktif,” *Sewagati J. Pengabd. Kpd. Masy.*, vol. 3, no. 3, pp. 79–84, 2019.

[3] T. Simangunsong and M. K. Anam, “Penerapan terkini teknologi bioflok dalam budidaya ikan nila: sebuah tinjauan,” *Glob. Sci. J. Res. Dev.*, vol. 3, no. 1, pp. 41–48, 2022.

[4] Y. Hikmayani, M. Yulisti, and H. Hikmah, “Evaluasi kebijakan peningkatan produksi perikanan budidaya,” *J. Kebijak. Sos. Ekon. Kelaut. dan Perikan.*, vol. 2, no. 2, p. 85, 2012, doi: 10.15578/jksekp.v2i2.9277.

[5] A. Irhamna, A. Pratama, R. P. Fhonna, and M. Ula, “Sistem

monitoring air pada kolam budidaya ikan lele berbasis Internet of Things,” *Sisfo J. Ilm. Sist. Inf.*, vol. 8, no. 1, p. 65, 2024, doi: 10.29103/sisfo.v8i1.18089.

[6] M. M. Islam, M. A. Kashem, S. A. Alyami, and M. A. Moni, “Monitoring water quality metrics of ponds with IoT sensors and machine learning to predict fish species survival,” *Microprocess. Microsyst.*, vol. 102, no. December 2022, p. 104930, 2023, doi: 10.1016/j.micpro.2023.104930.

[7] F. Hidayat, A. Harijanto, and B. Supriadi, “Rancang bangun alat ukur sistem monitoring pH dan suhu kolam ikan lele berbasis IoT dengan ESP8266,” *J. Kumbaran Fis.*, vol. 5, no. 2, pp. 77–84, 2022, doi: 10.33369/jkf.5.2.77-84.

[8] N. Nasution, M. I. Nasution, and S. Irawati, “Rancang bangun sistem smart kolam catfish berbasis mikrokontroler Arduino Uno,” *EINSTEIN J. Has. Penelit. Bid. Fis.*, vol. 10, no. 3, pp. 8–13, Oct. 2022, doi: 10.24114/einstein.v10i3.39509.

[9] N. F. N. Azizah, H. Pujiharsono, and M. A. Afandi, “Sistem pengendali suhu dan kadar pH pada kolam ikan lele berbasis IoT pada Desa Kutaringin Kabupaten Banjarnegara,” *J. Ris. Sains Dan Teknol.*, vol. 6, no. 1, p. 65, 2022, doi: 10.30595/jrst.v6i1.11693.

[10] N. Nursobah, S. Salmon, S. Lailiyah, and S. W. Sari, “Prototype sistem telemetri suhu dan pH air kolam budidaya ikan air tawar (ikan nila) berbasis Internet of Things (IoT),” *Sebatik*, vol. 26, no. 2, pp. 788–797, 2022, doi: 10.46984/sebatik.v26i2.2053.

[11] J. Wijaya, D. Syauqy, and R. Primananda, “Sistem monitoring dan rekomendasi kualitas air budidaya bibit ikan nila menggunakan parameter kekeruhan, suhu, dan pH dengan algoritma random forest,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 1, pp. 2548–2964, 2017.

[12] E. Suhartono *et al.*, “Penerapan Internet of Things (IoT) dalam pengendalian suhu, kelembaban dan pH air kolam pada mini EduFarm,” *J. Inf. Syst. Res.*, vol. 4, no. 4, pp. 1237–1244, 2023, doi: 10.47065/josh.v4i4.3723.

[13] Z. Hu *et al.*, “A water quality prediction method based on the deep LSTM network considering correlation in smart mariculture,” *Sensors (Switzerland)*, vol. 19, no. 6, 2019, doi: 10.3390/s19061420.

[14] A. Sherstinsky, “Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network,” *Phys. D Nonlinear Phenom.*, vol. 404, p. 132306, 2020, doi: 10.1016/j.physd.2019.132306.

[15] R. Yang *et al.*, “CNN-LSTM deep learning architecture for computer vision-based modal frequency detection,” *Mech. Syst. Signal Process.*, vol. 144, p. 106885, 2020, doi: 10.1016/j.ymsp.2020.106885.

[16] C. Lange, “Basics of Machine Learning,” in *Practical Machine Learning with R*, 2024, pp. 5–12. doi: 10.1201/9781003367147-2.

[17] J. Luts, F. Ojeda, R. Van de Plas, B. De Moor, S. Van Huffel, and J. A. K. Suykens, “A tutorial on support vector machine-based methods for classification problems in chemometrics,” *Anal. Chim. Acta*, vol. 665, no. 2, doi: 10.1016/j.aca.2010.03.030.

[18] I. R. Putri, “Pemahaman Konteks dalam Pemrosesan Teks dengan Menggunakan Algoritma LSTM,” Dec. 2023.

[19] I. R. Putri, “Pemahaman Konteks dalam Pemrosesan Teks dengan Menggunakan Algoritma LSTM,” no. December, 2023.

[20] T. Ağır, “Daily global solar radiation prediction with hybrid LSTM-SVM: the case of Nusaybin,” *Arab. J. Sci. Eng.*, 2025, doi: 10.1007/s13369-025-10322-7.

- [21] S. Zhou, C. Song, J. Zhang, W. Chang, W. Hou, and L. Yang, "A hybrid prediction framework for water quality with integrated W-ARIMA-GRU and LightGBM methods," *Water (Switzerland)*, vol. 14, no. 9, 2022, doi: 10.3390/w14091322.
- [22] J. Sha, X. Li, M. Zhang, and Z. L. Wang, "Comparison of forecasting models for real-time monitoring of water quality parameters based on hybrid deep learning neural networks," *Water (Switzerland)*, vol. 13, no. 11, 2021, doi: 10.3390/w13111547.
- [23] N. Buslim, I. L. Rahmatullah, B. A. Setyawan, and A. Alamsyah, "Comparing bitcoin's prediction model using GRU, RNN, and LSTM by hyperparameter optimization grid search and random search," in *9th International Conference on Cyber and IT Service Management 2021 (CITSM 2021)*, 2021, doi: 10.1109/CITSM52892.2021.9588947.
- [24] T. Jahn and B. Jin, "Early stopping of untrained convolutional neural networks," *SIAM J. Imaging Sci.*, vol. 17, no. 4, pp. 2331–2361, 2024, doi: 10.1137/24M1636617.

BIODATA PENULIS



T. A Nugroho

Mendapatkan gelar sarjana dan magister dari Teknik Elektro ITB. Bidang penelitian yang ditekuni: *physics informed neural networks*, *embedded system* dan IoT.



Dina Angela

Mendapatkan gelar sarjana dari Teknik Elektro Universitas Kristen Maranatha dan magister dari Teknik Elektro ITB. Saat ini aktif sebagai dosen di Prodi Teknik Elektro ITHB. Bidang penelitian yang ditekuni: telekomunikasi, antenna & propagasi, dan

IoT.



Yoel Wijaya Kristianto Hadikusumo

Menyelesaikan program Sarjana Strata 1 di Program Studi Teknik Komputer Institut Teknologi Harapan Bangsa pada 2025. Minat penelitian pada robotika, IoT, dan *machine learning*.