

BAB II TINJAUAN PUSTAKA

A. Penelitian Terkait

Berikut penelitian-penelitian terdahulu yang berkaitan dengan penerapan metode *Extreme Gradient Boosting* (XGBoost), *Artificial Neural Network* (ANN), serta pengembangan sistem prediksi berbasis web.

Tabel 1 Penelitian Terkait

No	Nama (Tahun)	Judul
(1)	(2)	(3)
1	R. Illaihiyah, G. Abdurrahman, Daryanto (2026)	Judul: Implementasi Algoritma <i>XGBoost Classifier</i> untuk Klasifikasi Kerumunan Jemaah Haji dan Umrah Metode: XGBoost Hasil Penelitian: Berdasarkan penelitian ini, Algoritma <i>Extreme Gradient Boosting</i> (XGBoost) menunjukkan kinerja klasifikasi yang sangat tinggi dalam memprediksi kepadatan kerumunan jemaah, dengan akurasi 98,7% serta presisi, recall, dan f1-score pada rentang 98–99% setelah dioptimasi menggunakan <i>Randomized Search CV</i> dan pembobotan kelas.
2	S. Ruscikasani, R. Roro, N. Oktalivia et al. (2025)	Judul: Prediksi Pembelian <i>E-Commerce</i> menggunakan XGBoost Berbasis Perilaku Sesi Pengguna Metode: XGBoost Hasil Penelitian: Penelitian ini menunjukkan bahwa algoritma <i>Extreme Gradient Boosting</i> (XGBoost) mampu memprediksi keputusan pembelian pada platform <i>e-commerce</i> dengan akurasi 0,785 dan ROC-AUC 0,804, serta precision, recall, dan F1-score yang menunjukkan

No	Nama (Tahun)	Judul
(1)	(2)	(3)
		kemampuan model dalam membedakan sesi berpotensi pembelian.
3	G. Cepi, F. Umbara, H. Ashaury (2025)	Judul: Prediksi Pembatalan Reservasi Hotel menggunakan Metode Xgboost Metode: XGBoost Hasil Penelitian: Penelitian ini menunjukkan bahwa algoritma <i>Extreme Gradient Boosting</i> (XGBoost) mampu memprediksi pembatalan reservasi hotel dengan akurasi 94,15% serta unggul dalam presisi, recall, dan F1-score. Faktor utama yang memengaruhi pembatalan adalah riwayat pembatalan, perubahan reservasi, dan permintaan khusus, dengan potensi pengembangan melalui teknik <i>oversampling</i> dan studi kasus yang lebih luas.
4	A. Saputra, B. Sari, C. Rozikin (2024)	Judul: Penerapan Algoritma <i>Extreme Gradient Boosting</i> (XGBoost) untuk Analisis Risiko Kredit Metode: XGBoost Hasil Penelitian: Berdasarkan penelitian ini, hasil evaluasi menunjukkan bahwa algoritma <i>Extreme Gradient Boosting</i> (XGBoost) efektif untuk analisis kelayakan kredit dan mengalami peningkatan performa setelah penerapan SMOTE, dengan skenario terbaik (90% data latih dan 10% data uji) menghasilkan <i>accuracy</i> 0,83 dan AUC 0,918.
5	W. Kurniawan, C. Indahyanti (2024)	Judul: Prediksi Angka Harapan Hidup Penduduk menggunakan Metode XGBoost Metode: XGBoost Hasil Penelitian:

No	Nama (Tahun)	Judul
(1)	(2)	(3)
		Penelitian ini menghasilkan model <i>Extreme Gradient Boosting</i> (XGBoost Regressor) yang mampu memprediksi angka harapan hidup dengan akurasi 96,8% dan MAE sebesar 0,97, menunjukkan efektivitasnya sebagai alat estimasi untuk mendukung perencanaan kebijakan berbasis data di kawasan Asia.
6	H. Fadil, E. Rouza, D. Suherna et al. (2026)	Judul: Implementasi <i>Artificial Neural Network</i> (ANN) untuk Memprediksi Cuaca Harian berdasarkan Data Suhu dan Kelembapan Metode: ANN Hasil Penelitian: Penelitian ini menunjukkan bahwa <i>Artificial Neural Network</i> (ANN) mampu memprediksi kondisi cuaca harian berbasis suhu dan kelembapan dengan akurasi 89% serta kemampuan generalisasi yang baik dalam menangkap pola non-linear. Model ini juga berhasil diimplementasikan dalam aplikasi web sebagai sistem prediksi yang praktis, dengan potensi pengembangan melalui penambahan variabel meteorologi dan penggunaan arsitektur <i>deep learning</i> yang lebih kompleks.
7	S. Anugrahaini, R. Fatchan, T. Wiyatno (2025)	Judul: Model Prediksi Tingkat Stres di Kalangan Siswa dengan Pendekatan Algoritma <i>Artificial Neural Network</i> Metode: ANN Hasil Penelitian: Penelitian ini menunjukkan bahwa algoritma <i>Neural Network</i> mampu memprediksi kelulusan mahasiswa dengan presisi 72,32%, recall 82,53%, dan akurasi 73,21%, sehingga

No	Nama (Tahun)	Judul
(1)	(2)	(3)
		cukup baik dalam mengidentifikasi mahasiswa yang berpotensi lulus. Untuk meningkatkan performa, diperlukan peningkatan kualitas data, optimasi <i>hyperparameter</i> , serta penerapan validasi silang dan evaluasi berkala.
8	R. Pratama, R. Herdiana, R. Hamonangan et al. (2024)	Judul: Analisis Prediksi Kelulusan Mahasiswa menggunakan Metode <i>Artificial Neural Network</i> Metode: ANN Hasil Penelitian: Penelitian ini menunjukkan bahwa algoritma <i>Neural Network</i> mampu memprediksi kelulusan mahasiswa dengan presisi 72,32%, recall 82,53%, dan akurasi 73,21%, sehingga cukup andal dalam mengidentifikasi mahasiswa yang berpotensi lulus.
9	J. Veronika, Andri (2022)	Judul: Penerapan Metode Algoritma <i>Neural Network</i> untuk Memprediksi Penjualan Bahan Bakar Minyak Metode: ANN Hasil Penelitian: Penelitian ini menggunakan 4.324 data transaksi dari PT Dempo Migas / SPBU 24.31550 dan menghasilkan akurasi tertinggi pada produk Biosolar sebesar 99,83% (skema 80:20) melalui pengolahan data mining menggunakan RapidMiner dengan atribut Nozzel, nomor penjualan, jumlah penjualan, bulan, tahun, dan bahan bakar sebagai label.
10	S. Prasetyo, T. Efendi, Muqorobin (2024)	Judul: Implementasi Sistem Prediksi Curah Hujan dengan Penerapan Jaringan Syaraf Tiruan Berbasis Website

No	Nama (Tahun)	Judul
(1)	(2)	(3)
		Metode: ANN Hasil Penelitian: Penelitian ini membangun sistem prediksi curah hujan bernama RainCast ID menggunakan <i>Neural Network</i> metode <i>Backpropagation</i> dengan arsitektur 4-4-1 berdasarkan data historis cuaca Indonesia tahun 2023–Mei 2024. Hasil pengujian menunjukkan bahwa peningkatan epoch dan <i>learning rate</i> (0,5) mampu meningkatkan akurasi dari 54,34% (epoch 20) menjadi 87,43% (epoch 500), dan sistem diimplementasikan menggunakan Flask berbasis Python.
11	A. Pangestu, S. Mujiono (2025)	Judul: Implementasi Algoritma XGBoost untuk Prediksi Status Gizi Balita Berbasis Website Metode: XGBoost Hasil Penelitian: Penelitian ini berhasil mengembangkan sistem prediksi status gizi balita menggunakan <i>Extreme Gradient Boosting</i> (XGBoost) dengan akurasi 97,17% serta presisi, recall, dan F1-score sekitar 97% setelah optimasi <i>Grid Search</i>, berdasarkan 5.169 data berkualitas hasil data cleaning. Sistem berbasis web ini berpotensi menjadi alat skrining awal yang efisien untuk mendukung program penurunan stunting.
12	S. Grace, D. Gunawan (2025)	Judul: Perbandingan CNN, Resnet50, dan <i>Vision Transformer</i> Untuk Klasifikasi Kanker Payudara Berbasis Web Metode: CNN, Resnet50, dan <i>Vision Transformer</i>

No	Nama (Tahun)	Judul
(1)	(2)	(3)
		Hasil Penelitian: Berdasarkan evaluasi tiga model <i>deep learning</i>, ResNet50 menunjukkan performa terbaik dan paling konsisten dalam deteksi kanker payudara dari citra mammografi, diikuti SimpleCNN, sementara Vision Transformer (ViT) memiliki performa terendah akibat keterbatasan data pelatihan. Ketiga model berhasil diimplementasikan dalam aplikasi berbasis Flask.
13	W. Su, F. Jiang et al. (2023)	Judul: <i>An XGBoost - Based Knowledge Tracing Model</i> Metode: XGBoost Hasil Penelitian: Penelitian ini menunjukkan bahwa algoritma <i>Extreme Gradient Boosting</i> (XGBoost) dapat diterapkan secara efektif pada model <i>knowledge tracing</i> dengan nilai AUC terbaik mencapai 0,9855 saat menggunakan multi-fitur. Dibandingkan model <i>deep learning</i> sebelumnya, XGBoost juga lebih efisien dalam waktu pelatihan serta lebih mudah diimplementasikan pada platform pendidikan daring.
14	S. Arya, Anju, N. Ramli (2024)	Judul: <i>Predicting the stress level of students using Supervised Machine Learning and Artificial Neural Network (ANN)</i> Metode: ANN Hasil Penelitian: Penelitian ini menggabungkan berbagai model <i>machine learning</i> dan <i>deep learning</i> seperti <i>Support Vector Machine</i>, <i>Naive Bayes</i>, dan <i>Extreme Gradient Boosting</i> untuk mengidentifikasi faktor penyebab stres pada

No	Nama (Tahun)	Judul
(1)	(2)	(3)
		mahasiswa, dengan akurasi tertinggi diperoleh <i>Naïve Bayes</i> sebesar 90% dan terendah SVM sebesar 85,45%.

Berdasarkan kajian terhadap penelitian-penelitian terdahulu, diketahui bahwa metode *Extreme Gradient Boosting* (XGBoost) dan *Artificial Neural Network* (ANN) telah banyak diterapkan dalam berbagai bidang seperti klasifikasi, analisis risiko, dan prediksi berbasis web. Namun, sebagian besar penelitian tersebut hanya berfokus pada penerapan satu metode tanpa melakukan perbandingan performa secara langsung pada permasalahan yang sama. Selain itu, penelitian komparatif yang ada umumnya dilakukan pada bidang kesehatan, pendidikan, atau klasifikasi citra, sehingga masih terbatas penelitian yang membandingkan kedua metode tersebut dalam konteks prediksi produksi pertanian berbasis variabel iklim.

Selain itu, belum banyak penelitian yang mengintegrasikan hasil komparasi algoritma ke dalam sistem prediksi berbasis web sebagai bentuk implementasi nyata untuk mendukung pengambilan keputusan. Oleh karena itu, penelitian ini bertujuan untuk membandingkan performa metode XGBoost dan ANN dalam memprediksi produksi padi berdasarkan variabel iklim, serta mengimplementasikan metode dengan performa terbaik ke dalam sistem prediksi berbasis web guna mendukung perencanaan produksi dan pengambilan kebijakan berbasis data. Diagram *mind map* dijelaskan pada Gambar 1.



Gambar 1 MindMap Penelitian

B. Landasan Teori

Landasan teori pada penelitian “Studi Komparasi Performa *Algoritma Extreme Gradient Boosting (XGBoost)* dan *Artificial Neural Network (ANN)* pada Sistem Prediksi Produksi Padi Berbasis Web” yaitu:

1. Produksi Padi

Padi (*Oryza sativa L.*) merupakan tanaman pangan pokok bagi sebagian besar penduduk di Asia, termasuk Indonesia. Sebagai komoditas strategis, padi menjadi tulang punggung ketahanan pangan nasional yang menyumbang kebutuhan kalori terbesar masyarakat Indonesia. Menurut Badan Pusat Statistik, produksi padi di Kabupaten Cilacap pada tahun 2024 sebanyak 651,14 ribu ton gabah kering giling (GKG), dan mengalami penurunan 69,78 ribu ton (15,71 persen) dibandingkan dengan produksi tahun 2023. Fluktuasi produksi dari tahun ke tahun akibat pengaruh iklim dan kondisi lahan [30].

Produksi padi dipengaruhi oleh dua kelompok faktor utama, yaitu teknis dan sosial-ekonomi. Faktor teknis meliputi varietas padi yang digunakan, kualitas benih, teknik budidaya yang tepat, serta efisiensi pemupukan. Sementara itu, sosial-ekonomi mencakup kondisi tanah, ketersediaan air irigasi, ketidakpastian musim tanam, serta kondisi iklim [1]. Di antara faktor-faktor tersebut, kondisi iklim menjadi variabel yang paling sulit dikendalikan namun memberikan pengaruh yang dominan terhadap fluktuasi produksi padi.

2. Iklim

Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) mendefinisikan iklim sebagai pola rata-rata cuaca dalam jangka waktu panjang (biasanya 30 tahun atau lebih) yang terjadi di suatu wilayah. Iklim berbeda dengan cuaca, karena cuaca bersifat harian dan dapat berubah dalam waktu singkat, sedangkan iklim menunjukkan kecenderungan jangka panjang [31]. Secara lebih spesifik, iklim merupakan sintesis atau rangkuman kejadian cuaca dalam periode waktu yang panjang, yang kemudian dapat digunakan untuk mengidentifikasi karakteristik dan pola cuaca suatu wilayah [32].

Indonesia memiliki iklim tropis yang dicirikan oleh curah hujan tinggi, suhu udara relatif stabil sepanjang tahun, kelembaban relatif tinggi, dan lama penyinaran matahari yang relatif konstan [33]. Curah hujan merupakan sumber air utama bagi pertanian padi dengan kebutuhan optimal berkisar 200-300 mm per bulan selama musim tanam, sementara suhu optimal untuk pertumbuhan padi berkisar antara 25-30°C dengan suhu ekstrem di atas 35°C atau di bawah 15°C dapat menyebabkan gangguan pada proses reproduksi tanaman[3][34]. Kelembaban relatif yang optimal berada pada kisaran 70-80%. Kelembaban terlalu tinggi dapat meningkatkan risiko penyakit jamur, sedangkan lama penyinaran matahari yang cukup (6-8 jam per hari) sangat diperlukan untuk proses fotosintesis dan pengisian bulir padi [35][36].

Pola iklim Indonesia sangat dipengaruhi oleh berbagai fenomena atmosfer dan oseanik seperti *El Nino-Southern Oscillation* (ENSO) yang berperan penting dalam variabilitas iklim tahunan. Indeks ENSO menjadi indikator penting dalam memprediksi variabilitas iklim yang berpengaruh terhadap produktivitas pertanian di Indonesia. Pada fase *El Niño* terjadi kecenderungan kemarau panjang yang berdampak pada penurunan produksi padi, sedangkan pada fase *La Niña* terjadi peningkatan curah hujan yang berpotensi menimbulkan banjir dan gagal panen. [37]. Kelima variabel iklim tersebut—lama penyinaran matahari, curah hujan, suhu udara, kelembaban relatif, dan Indeks ENSO—memiliki korelasi dengan produksi padi dan menjadi variabel input utama dalam penelitian ini.

3. *Machine Learning*

Machine Learning (pembelajaran mesin) merupakan cabang dari kecerdasan buatan (*Artificial Intelligence*) yang memungkinkan sistem komputer memiliki kemampuan untuk belajar secara otomatis dari data dan pengalaman tanpa harus diprogram secara eksplisit untuk setiap tugas tertentu [22]. Konsep ini pertama kali diperkenalkan oleh Arthur Samuel pada tahun 1959 yang mendefinisikan *machine learning* sebagai bidang studi yang memberikan kemampuan komputer untuk belajar tanpa diprogram secara eksplisit. Melalui proses pembelajaran dari data, *machine learning* dapat mengidentifikasi pola dan hubungan antarvariabel untuk menghasilkan prediksi atau keputusan [12].

Secara umum, *machine learning* terdiri atas beberapa kategori, salah satunya adalah *supervised learning* yang merupakan metode pembelajaran yang menggunakan data yang telah memiliki variabel input dan variabel target. Model mempelajari hubungan antara variabel input dan variabel target berdasarkan data historis, kemudian menggunakan pola yang telah dipelajari untuk menghasilkan prediksi terhadap data baru [38].

Permasalahan dalam *supervised learning* dapat diselesaikan menggunakan berbagai algoritma, seperti *Linear Regression*, *Support Vector Regression* (SVR), *Decision Tree*, *Random Forest*, *XGBoost*, *AdaBoost*, *CatBoost*, dan *Artificial Neural Network* (ANN). Setiap algoritma memiliki mekanisme pembelajaran dan karakteristik yang berbeda sehingga pemilihannya perlu disesuaikan dengan karakteristik data serta tujuan penelitian [25]. Dalam penelitian ini, *machine learning* digunakan untuk mempelajari hubungan antara variabel iklim sebagai variabel input dan jumlah produksi padi sebagai variabel target.

4. *Metode Boosting*

Boosting merupakan salah satu metode yang menggabungkan beberapa model sederhana (*weak learner*) untuk membentuk model prediksi yang lebih

kuat dan akurat. Pada metode *boosting*, model dibangun secara bertahap dan berurutan (*sequential*). Setiap model baru berfokus untuk memperbaiki kesalahan yang dihasilkan oleh model sebelumnya sehingga kemampuan prediksi dapat meningkat secara bertahap. Hasil dari seluruh model kemudian digabungkan untuk menghasilkan prediksi akhir [39].

Beberapa algoritma yang menggunakan metode *boosting* antara lain AdaBoost, CatBoost, LightGBM dan XGBoost.

a. *Adaptive Boosting* (AdaBoost)

Adaptive Boosting (AdaBoost) merupakan salah satu algoritma *boosting* yang membangun beberapa model secara bertahap. Pada setiap proses pembelajaran, AdaBoost memberikan bobot yang lebih besar pada data yang sebelumnya mengalami kesalahan prediksi. Dengan demikian, model berikutnya akan memberikan perhatian lebih terhadap data yang sulit diprediksi. Hasil prediksi dari seluruh model kemudian digabungkan untuk menghasilkan prediksi akhir. AdaBoost memiliki mekanisme yang relatif sederhana, tetapi dapat sensitif terhadap data yang mengandung *noise* atau *outlier* [40].

b. *Categorical Boosting* (CatBoost)

CatBoost (*Categorical Boosting*) merupakan algoritma *boosting* yang dirancang untuk menangani data kategorikal secara efektif. CatBoost menggunakan mekanisme *ordered boosting* untuk membantu mengurangi risiko *overfitting* serta dapat mengolah variabel kategorikal tanpa memerlukan proses *encoding* secara manual. Karakteristik tersebut menjadikan CatBoost sesuai digunakan pada dataset yang memiliki banyak variabel kategorikal. Selain itu, CatBoost juga dapat digunakan untuk menyelesaikan permasalahan regresi maupun klasifikasi [41].

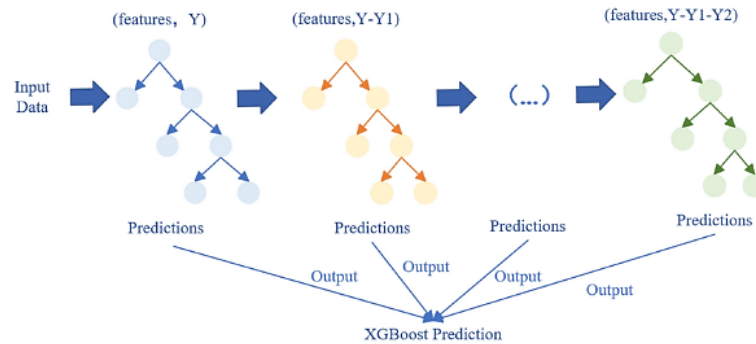
c. *Light Gradient Boosting Machine* (LightGBM)

Light Gradient Boosting Machine (LightGBM) merupakan algoritma *gradient boosting* yang dirancang untuk menghasilkan proses pelatihan yang cepat dan efisien. LightGBM menggunakan strategi pertumbuhan pohon secara *leaf-wise*, yaitu memilih daun dengan potensi penurunan kesalahan terbesar untuk dikembangkan terlebih dahulu. Strategi tersebut memungkinkan LightGBM menghasilkan proses pembelajaran yang efisien, terutama pada dataset berukuran besar dan memiliki jumlah fitur yang banyak. Namun, pada data yang relatif kecil, strategi *leaf-wise* dapat meningkatkan risiko *overfitting* apabila parameter model tidak diatur dengan baik [42].

d. *Extreme Gradient Boosting* (XGBoost)

Extreme Gradient Boosting (XGBoost) adalah algoritma *machine learning* berbasis ensemble yang dikembangkan oleh Tianqi Chen dan Carlos Guestrin pada tahun 2016 [11][43]. XGBoost bekerja dengan

membangun model prediksi secara bertahap dan berurutan (*sequential*), yang setiap pohon keputusan (*decision tree*) baru dibangun untuk memperbaiki kesalahan yang dibuat oleh pohon-pohon sebelumnya [20][44]. Proses ini dilakukan secara bertahap hingga diperoleh model dengan tingkat kesalahan yang semakin kecil.



Gambar 2 Algoritma XGBoost

Berdasarkan Gambar 2, proses pembelajaran XGBoost dilakukan secara bertahap melalui pembentukan sejumlah *decision tree*. Model awal menghasilkan prediksi, kemudian kesalahan prediksi digunakan sebagai dasar untuk membangun *decision tree* berikutnya. Setiap *decision Tree* baru berfungsi untuk memperbaiki kesalahan dari model sebelumnya. Hasil prediksi dari seluruh *decision tree* kemudian digabungkan untuk menghasilkan prediksi akhir [18].

XGBoost menerapkan fitur regularisasi L1 (*Lasso*) dan L2 (*Ridge*) yang berfungsi untuk mengurangi risiko *overfitting*, sehingga model menjadi lebih stabil ketika digunakan pada data baru [11][19]. Selain itu, XGBoost mendukung komputasi paralel sehingga proses pelatihan dapat dilakukan untuk bekerja lebih cepat dan efisien [45][46]. XGBoost juga memiliki kemampuan yang baik dalam mengolah data numerik berbentuk tabel (*tabular data*) dan dapat digunakan untuk menyelesaikan permasalahan regresi maupun klasifikasi. Keunggulan tersebut menjadikan XGBoost banyak digunakan dalam berbagai studi kasus prediksi, salah satunya pada bidang pertanian [39][47].

Peneliti memilih XGBoost karena data yang digunakan berupa data numerik berbentuk tabel (*tabular data*) dengan jumlah data yang relatif terbatas serta memiliki hubungan yang kompleks dan nonlinier antara variabel iklim dan produksi padi. Dibandingkan AdaBoost, XGBoost memiliki mekanisme optimasi dan regularisasi yang lebih lengkap. Sementara itu, LightGBM memiliki keunggulan dalam efisiensi proses pelatihan, terutama pada dataset berukuran besar, sedangkan CatBoost memiliki kemampuan yang baik dalam menangani data kategorikal. Karakteristik XGBoost yang mampu

mengolah data numerik, memodelkan hubungan nonlinier, serta menerapkan regularisasi dinilai sesuai dengan karakteristik data dan tujuan penelitian. Oleh karena itu, XGBoost digunakan sebagai algoritma *boosting* untuk memprediksi produksi padi.

Secara sistematis, persamaan prediksi XGBoost adalah sebagai berikut:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i)$$

Keterangan:

$\hat{y}_i$ = nilai prediksi

x_i = data masukan

f_k = *decision tree* ke- k

k = jumlah *decision Tree*

Berdasarkan persamaan di atas, setiap *decision Tree* ditambahkan secara bertahap untuk memperbaiki kesalahan prediksi dari *tree* sebelumnya.

Dalam proses pelatihan, XGBoost menggunakan fungsi objektif yang terdiri atas fungsi kesalahan (*loss function*) dan fungsi regularisasi. Persamaannya adalah sebagai berikut:

$$\text{Obj} = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k)$$

Keterangan:

Obj = fungsi objektif

$l(y_i, \hat{y}_i)$ = fungsi kesalahan

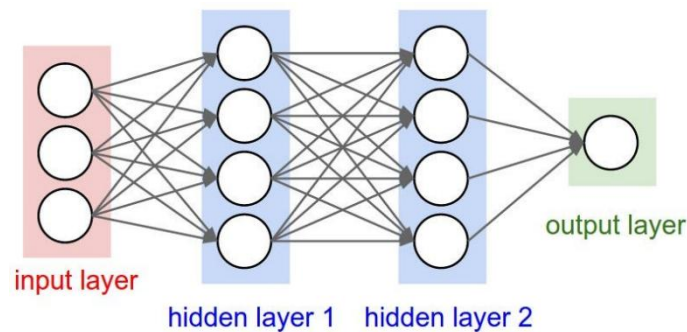
$\Omega(f_k)$ = fungsi regularisasi

n = jumlah data

k = jumlah *decision tree*

Berdasarkan persamaan tersebut, XGBoost tidak hanya berfokus pada minimisasi kesalahan prediksi, tetapi juga mengurangi risiko *overfitting*.

5. Artificial Neural Network (ANN)



Gambar 3 Algoritma ANN

Artificial Neural Network (ANN) atau Jaringan Syaraf Tiruan merupakan model komputasi yang terinspirasi dari struktur dan fungsi jaringan saraf biologis pada otak manusia [24]. Konsep dasar ANN pertama kali diperkenalkan oleh McCulloch dan Pitts (1943) melalui model *neuron* formal, dan kemudian berkembang pesat dengan diperkenalkannya algoritma *backpropagation* yang memungkinkan pelatihan jaringan berlapis. ANN terdiri dari unit-unit pemrosesan yang disebut *neuron*, yang terdiri dari lapisan input yang menerima data masukan, satu atau lebih lapisan tersembunyi (*hidden layer*) yang melakukan transformasi non-linear terhadap data, dan lapisan output yang menghasilkan prediksi akhir [21]. Proses pembelajaran ANN dilakukan melalui algoritma *backpropagation* yang memperbarui bobot-bobot koneksi antar *neuron* secara berulang untuk meminimalkan *error*, hingga mencapai model dengan tingkat kesalahan yang lebih kecil [22].

Dalam proses pembelajaran, ANN menyesuaikan bobot koneksi antar neuron berdasarkan nilai kesalahan (*error*) yang dihasilkan selama pelatihan [26][48]. Struktur berlapis memungkinkan model menangkap pola tersembunyi dalam data sehingga memungkinkan model untuk membuat prediksi akurat [49]. Selain itu, arsitektur ANN bersifat fleksibel karena dapat disesuaikan berdasarkan jumlah *neuron*, jumlah *hidden layer*, dan parameter pelatihan, sehingga dapat dioptimalkan sesuai karakteristik data yang digunakan [50].

Secara matematis, proses pada ANN dimulai dengan perhitungan nilai masukan pada setiap *neuron*. Nilai tersebut diperoleh dari penjumlahan hasil perkalian antara data masukan dan bobot, kemudian ditambahkan dengan nilai bias. Persamaan perhitungan *neuron* ditunjukkan sebagai berikut:

$$z = \sum_{i=1}^n w_i x_i + b$$

Keterangan:

z = nilai masukan *neuron*

x_i = nilai input

w_i = bobot

b = bias

Nilai z kemudian diproses menggunakan fungsi aktivasi untuk menghasilkan keluaran *neuron*. Pada penelitian ini, fungsi aktivasi yang digunakan adalah *Rectified Linear Unit* (ReLU), yang dirumuskan sebagai berikut:

$$f(z) = \max(0, z)$$

Keterangan:

$f(z)$ = keluaran *neuron*

z = nilai masukan *neuron*

Selanjutnya, proses *backpropagation* digunakan untuk memperbarui bobot berdasarkan nilai kesalahan yang dihasilkan selama pelatihan. Persamaan pembaruan bobot dapat dituliskan sebagai berikut:

$$w_{\text{baru}} = w_{\text{lama}} - \eta \frac{\partial E}{\partial w}$$

Keterangan:

w_{baru} = bobot setelah diperbarui

w_{lama} = bobot sebelumnya

η = *learning rate*

E = nilai kesalahan

6. Bahasa Pemrograman Python

Python merupakan bahasa pemrograman yang diciptakan oleh Guido van Rossum pada tahun 1991 [14]. Python merupakan bahasa pemrograman tingkat tinggi (*high-level programming language*) yang bersifat interpretatif, mudah dipelajari, dan memiliki sintaks yang jelas serta ringkas. Python bersifat terbuka (*open-source*) sehingga dapat digunakan dan dikembangkan secara bebas oleh berbagai kalangan. Python banyak digunakan dalam pengembangan aplikasi, analisis data, serta pemodelan *machine learning* [51].

Dalam penelitian ini, Python digunakan sebagai bahasa pemrograman utama dalam proses pengolahan data variabel iklim, pembangunan model XGBoost dan ANN, serta evaluasi performa model menggunakan metrik seperti R^2 , MAE, dan RMSE. Selain itu, Python juga berperan dalam proses integrasi model ke dalam sistem berbasis web melalui *framework* tertentu, sehingga hasil prediksi dapat diakses oleh pengguna secara lebih mudah [47].

7. Parameter Evaluasi (R^2 , MAE, RMSE)

Parameter perbandingan berdasarkan penelitian terdahulu dapat dilihat pada Tabel 2 berikut [16][26][44].

Tabel 2 Parameter Perbandingan

No	Parameter	Keterangan	Output
(1)	(2)	(3)	(4)
1.	<i>R-Squared</i> (R^2)	Mengukur seberapa besar variasi data produksi padi dapat dijelaskan oleh model	Skor (0-1)
2.	<i>Mean Absolute Error</i> (MAE)	Mengukur rata-rata selisih absolut antara nilai prediksi dan nilai aktual	Rata-rata (ton)

No	Parameter	Keterangan	Output
(1)	(2)	(3)	(4)
3.	<i>Root Mean Squared Error</i> (RMSE)	Mengukur akar dari rata-rata kuadrat kesalahan prediksi	Rata-rata (ton)

Metrik evaluasi yang telah ditentukan pada tabel tersebut menjadi dasar dalam pengukuran kinerja model regresi. Untuk memberikan gambaran yang lebih jelas, berikut penjelasan dari masing-masing parameter.

a. *R-Squared* (R^2)

R-Squared atau koefisien determinasi merupakan metrik yang digunakan untuk mengukur seberapa besar variasi pada data aktual dapat dijelaskan oleh model prediksi. Nilai R^2 berada pada rentang 0 sampai 1. Semakin mendekati 1, maka semakin besar kemampuan model dalam menjelaskan hubungan antara variabel input (variabel iklim) dan variabel output (produksi padi). Sebaliknya, nilai yang mendekati 0 menunjukkan bahwa model kurang mampu menjelaskan variasi data. Secara matematis, R^2 dapat dirumuskan sebagai berikut:

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

Keterangan:

y_i = nilai aktual

$\hat{y}_i$ = nilai prediksi

$\bar{y}$ = rata-rata nilai aktual

b. *Mean Absolute Error* (MAE)

Mean Absolute Error (MAE) digunakan untuk mengukur rata-rata selisih absolut antara nilai prediksi dan nilai aktual. MAE memberikan gambaran besarnya kesalahan prediksi secara umum tanpa memperhatikan arah kesalahan (positif atau negatif). Nilai MAE yang lebih kecil menunjukkan bahwa hasil prediksi model semakin mendekati nilai sebenarnya. Rumus MAE adalah sebagai berikut:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Keterangan:

n = jumlah data

y_i = nilai aktual

$\hat{y}_i$ = nilai prediksi

c. *Root Mean Squared Error (RMSE)*

Root Mean Squared Error (RMSE) merupakan metrik yang mengukur akar dari rata-rata kuadrat selisih antara nilai prediksi dan nilai aktual. RMSE memberikan penalti yang lebih besar terhadap kesalahan yang bernilai besar karena proses pengkuadratan. Nilai RMSE yang lebih kecil menunjukkan performa model yang lebih baik. Rumus RMSE adalah sebagai berikut:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

Keterangan:

n = jumlah data

y_i = nilai aktual

$\hat{y}_i$ = nilai prediksi

8. *Random Split*

Random Split merupakan metode pembagian data yang dilakukan secara acak menjadi data latih (*training*) dan data uji (*testing*). Data *training* digunakan untuk membentuk model, sedangkan data *testing* digunakan untuk menguji model. Tujuan dari pembagian data ini agar model dapat dilatih menggunakan sebagian data dan kemudian dievaluasi menggunakan data yang berbeda sehingga performa model dapat diukur secara lebih objektif. Dengan metode ini, setiap data memiliki peluang yang sama untuk terpilih sebagai data latih maupun data uji, sehingga dapat mengurangi potensi bias dalam proses pembentukan model. Oleh karena itu, *random split* banyak digunakan dalam penelitian *machine learning* untuk menguji kemampuan model dalam mengenali pola data dan menghasilkan prediksi yang baik pada data yang belum pernah dipelajari sebelumnya [18][19][51].

Pada penelitian ini, data dibagi menggunakan 80:20, yaitu 80% data digunakan sebagai data latih dan 20% data digunakan sebagai data uji. Rasio tersebut dipilih karena memberikan proporsi data yang lebih besar untuk proses pelatihan, sekaligus menyediakan data yang cukup untuk pengujian. Dengan demikian, rasio 80:20 diharapkan dapat memberikan keseimbangan antara jumlah data pelatihan dan pengujian serta mendukung proses pembentukan model yang lebih stabil pada algoritma XGBoost dan ANN [20].

9. *Metode Quantile*

Quantile merupakan metode pengelompokan data berdasarkan distribusi nilai pada suatu dataset. Pembagian dilakukan berdasarkan nilai persentil tertentu sehingga batas antar kelompok ditentukan oleh sebaran data yang ada, bukan oleh nilai yang ditetapkan secara manual. Penggunaan metode *quantile*

memungkinkan setiap kategori memiliki proporsi data yang lebih seimbang dan representatif terhadap karakteristik dataset. Dalam penelitian ini, metode *quantile* digunakan untuk mengelompokkan data produksi padi ke dalam tiga kategori, yaitu rendah, sedang, dan tinggi berdasarkan distribusi data historis produksi padi. Pengelompokan tersebut bertujuan untuk memudahkan pengguna dalam memahami hasil prediksi produksi padi yang ditampilkan pada sistem serta sebagai dasar dalam pemberian rekomendasi yang sesuai dengan kategori produksi yang dihasilkan [52][53].

10. Sistem Berbasis Web

Sistem berbasis web merupakan aplikasi yang dapat diakses melalui jaringan internet atau intranet menggunakan peramban (browser) tanpa memerlukan instalasi khusus pada perangkat pengguna. Arsitektur sistem berbasis web mengikuti model *client-server*. *Client* (browser pengguna) berperan dalam mengirimkan permintaan (*request*) ke server melalui protokol HTTP/HTTPS, sedangkan server bertugas memproses permintaan tersebut dan mengirimkan respon kembali kepada *client*. Keunggulan sistem berbasis web terletak pada kemudahan akses, fleksibilitas penggunaan, serta kemampuannya untuk diakses dari berbagai perangkat seperti komputer maupun *smartphone* [27].

Dalam pengembangan sistem berbasis web, diperlukan teknologi *backend* untuk mengatur proses pengolahan data dan komunikasi antara pengguna dengan server. Teknologi *backend* tersebut dapat diimplementasikan menggunakan *framework* berbasis Python yang mendukung integrasi dengan model *machine learning* sehingga sistem prediksi dapat berjalan dengan baik.

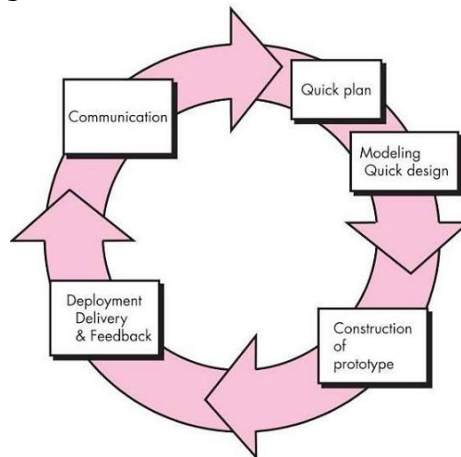
11. *Framework* FastAPI

FastAPI merupakan *framework backend* berbasis Python yang digunakan untuk membangun *Application Programming Interface* (API) secara cepat dan efisien. *Framework* ini banyak digunakan dalam pengembangan sistem berbasis *machine learning* karena mampu menangani proses pertukaran data antara model prediksi dan antarmuka web dengan baik. Selain itu, FastAPI mampu melakukan validasi data secara otomatis sehingga dapat membantu menjaga konsistensi dan integritas data sebelum diproses oleh model prediksi [54].

Pada penelitian ini, FastAPI digunakan sebagai penghubung antara sistem berbasis web dengan model prediksi *machine learning*. Ketika pengguna memasukkan data variabel iklim melalui antarmuka web, sistem akan mengirimkan permintaan (*request*) ke server menggunakan format JSON. Selanjutnya, FastAPI menerima data tersebut dan meneruskannya ke proses prediksi yang telah dibangun menggunakan Python. Setelah model menghasilkan nilai prediksi produksi padi, FastAPI mengirimkan hasil tersebut kembali ke antarmuka web dalam bentuk respons sehingga dapat ditampilkan

kepada pengguna. Dengan mekanisme tersebut, proses integrasi antara sistem web dan model *machine learning* dapat berjalan secara terstruktur dan efisien.

12. Metode *Prototyping*



Gambar 4 Metode *Prototyping*

Metode pengembangan sistem *prototyping* merupakan salah satu model pengembangan perangkat lunak yang berfokus pada pembuatan versi awal (*prototype*) dari sistem untuk menggambarkan kebutuhan dan fungsionalitas yang diinginkan pengguna. *Prototype* yang dibuat bersifat sementara dan digunakan sebagai media komunikasi antara pengembang dan pengguna agar kebutuhan sistem dapat dipahami secara lebih jelas sebelum sistem akhir dikembangkan. Melalui pendekatan ini, pengguna dapat memberikan umpan balik secara langsung terhadap tampilan maupun fungsi sistem sehingga risiko kesalahan dalam perancangan dapat diminimalkan [55].

Tahapan dalam metode *prototyping* umumnya meliputi proses komunikasi untuk mengetahui kebutuhan, perancangan awal (*quick design*), pembuatan *prototype*, evaluasi oleh pengguna, perbaikan berdasarkan umpan balik, dan pengembangan sistem akhir. Metode ini sesuai diterapkan pada sistem yang memerlukan keterlibatan aktif pengguna, seperti sistem berbasis web, karena memungkinkan proses pengembangan dilakukan secara bertahap melalui evaluasi dan penyempurnaan hingga sistem memenuhi kebutuhan yang diharapkan [56].

13. *Black Box Testing*

Black box testing merupakan metode pengujian perangkat lunak yang berfokus pada pengujian fungsi sistem tanpa melihat atau menganalisis struktur kode program di dalamnya. Pengujian ini dilakukan dengan cara memberikan inputan ke dalam sistem, kemudian mengamati apakah output yang dihasilkan telah sesuai dengan kebutuhan yang telah dirancang sebelumnya. Pada pendekatan ini, penguji tidak perlu mengetahui proses internal sistem,

melainkan hanya memastikan bahwa sistem memberikan respons yang benar berdasarkan input yang diberikan [24][57].

Pada penelitian ini, metode *black box testing* digunakan untuk memastikan bahwa seluruh proses pada sistem prediksi produksi padi berbasis web berjalan dengan benar sesuai alur yang telah dirancang. Pengujian dilakukan dengan memberikan berbagai variasi input parameter seperti tahun, bulan tanam, serta variabel iklim, kemudian diamati apakah sistem mampu memproses data tersebut tanpa *error* dan menghasilkan output prediksi yang sesuai dengan logika model yang telah diintegrasikan. Melalui pengujian ini, dapat dipastikan bahwa fungsi sistem bekerja sebagaimana mestinya dan tidak terjadi kesalahan pada proses maupun hasil yang ditampilkan kepada pengguna.

14. Kolmogorov–Smirnov

Uji *Kolmogorov–Smirnov* merupakan salah satu metode statistik yang digunakan untuk menguji apakah suatu data mengikuti distribusi normal. Pengujian dilakukan dengan membandingkan distribusi data yang diamati dengan distribusi normal. Hasil pengujian dapat dilihat berdasarkan nilai signifikansi (Sig.). Apabila nilai signifikansi lebih besar dari 0,05, data dinyatakan berdistribusi normal. Sebaliknya, apabila nilai signifikansi kurang dari atau sama dengan 0,05, data dinyatakan tidak berdistribusi normal [58]. Dalam penelitian ini, uji *Kolmogorov–Smirnov* digunakan untuk menguji normalitas data selisih nilai *Absolute Error* (AE) antara model XGBoost dan ANN sebelum dilakukan pengujian statistik komparatif.

15. Wilcoxon Signed-Rank Test

Wilcoxon Signed-Rank Test merupakan uji statistik nonparametrik yang digunakan untuk membandingkan dua kelompok data yang saling berpasangan apabila data selisih tidak berdistribusi normal. Pengujian dilakukan berdasarkan perbedaan nilai pada setiap pasangan data untuk mengetahui apakah terdapat perbedaan yang signifikan antara kedua kelompok. Dalam penelitian ini, *Wilcoxon Signed-Rank Test* digunakan untuk membandingkan nilai *Absolute Error* (AE) model XGBoost dan ANN pada periode pengamatan yang sama. Pengambilan keputusan dilakukan berdasarkan nilai signifikansi (*Asymp. Sig. (2-tailed)*). Apabila nilai signifikansi kurang dari 0,05, maka terdapat perbedaan yang signifikan antara nilai AE kedua model. Sebaliknya, apabila nilai signifikansi lebih besar dari atau sama dengan 0,05, maka tidak terdapat perbedaan yang signifikan [59].