

BAB II

KAJIAN LITERATUR

2.1 Bahan Pangan Pokok (Sembako)

Harga pangan adalah nilai tukar yang terbentuk dalam pertukaran komoditas pangan di pasar, yang mencerminkan titik keseimbangan antara pasokan (*supply*) dan permintaan (*demand*) pada waktu dan tempat tertentu. Dalam mekanisme pasar, harga berfungsi sebagai indikator sekaligus sinyal bagi produsen dan konsumen dalam mengalokasikan sumber daya secara efisien [16]. Stabilitas harga pangan menjadi landasan krusial bagi kesejahteraan masyarakat, terutama untuk komoditas sembilan bahan pokok (sembako) yang merupakan pemenuhan hak dan kebutuhan dasar manusia sehari-hari.

Dalam konteks ketahanan pangan di Provinsi Sumatera Utara, tiga komoditas sembako utama yang mendominasi pembentukan harga dan biaya hidup adalah beras, minyak goreng, dan telur. Komoditas ini memiliki bobot kontribusi inflasi yang signifikan karena perannya: beras berkedudukan sebagai sumber karbohidrat utama dengan tingkat konsumsi absolut yang tinggi; minyak goreng merupakan bahan pangan esensial pelengkap yang frekuensi penggunaannya hampir terjadi setiap hari; sedangkan telur ayam ras memegang peran sebagai sumber protein hewani yang paling rasional, terjangkau, dan mudah diakses oleh berbagai lapisan masyarakat [7].

Pembentukan dan pergerakan harga ketiga komoditas sembako ini didasari oleh beberapa karakteristik dasar sebagai berikut:

1. Sifat Permintaan yang Tidak Elastis

Sembako tergolong barang dengan permintaan inelastis, yang menunjukkan bahwa perubahan harga tidak serta-merta mengubah jumlah permintaan secara proporsional. Masyarakat akan tetap membeli komoditas ini meskipun harganya melonjak karena sifatnya sebagai kebutuhan dasar yang tidak dapat ditunda [16]. Kondisi ini sangat membebani kelompok ekonomi menengah ke bawah yang mengalokasikan sebagian besar pendapatannya untuk konsumsi pangan [1][2].

2. Kerentanan Terhadap Iklim dan Cuaca

Sebagai produk agrikultur, ketersediaan pasokan sembako berbanding lurus dengan kondisi alam. Anomali cuaca dapat mengganggu ketersediaan pasokan secara drastis melalui gagal panen (pada padi), penurunan produktivitas ternak (pada ayam petelur), maupun terhambatnya panen kelapa sawit [6][7][26]

3. Kompleksitas Rantai Distribusi

Secara geografis, panjangnya rantai pasok dari sentra produksi ke pasar konsumen menciptakan biaya transportasi dan distribusi tambahan. Gangguan fisik pada infrastruktur logistik (seperti bencana alam) akan memotong jalur pasokan (*supply shock*), yang di pasar akan langsung direspons dengan lonjakan harga yang signifikan [3][7].

4. Dampak Sosial Ekonomi

Ketidakstabilan harga komoditas pokok menciptakan dampak negatif yang luas. Peningkatan harga sembako secara langsung menggerus daya beli riil masyarakat. Pada kondisi ekstrem, hal ini memaksa terjadinya pengorbanan (*trade-off*) pengeluaran di sektor krusial lainnya, seperti kesehatan dan pendidikan, demi mempertahankan kebutuhan makan sehari-hari [2][15].

Berdasarkan uraian di atas, dapat disimpulkan bahwa komoditas sembako—khususnya beras, minyak goreng, dan telur memiliki relasi sebab-akibat yang rentan antara sisi pasokan dan permintaan. Tingkat permintaan yang inelastis (wajib dipenuhi) ini seringkali dihadapkan pada ancaman ketersediaan pasokan yang rapuh akibat guncangan anomali cuaca serta hambatan pada kelancaran rantai distribusi logistik. Ketidakseimbangan dari karakteristik inilah yang kerap kali memicu lonjakan harga mendadak, sehingga dampaknya akan langsung menghancurkan daya beli serta kesejahteraan kelompok ekonomi rentan. Tingginya risiko sosial-ekonomi akibat perpaduan kerentanan pasokan ini menjadi landasan mengapa instrumen prediksi serta pemantauan (*monitoring*) komoditas yang cerdas sangat krusial diimplementasikan. Dengan deteksi pergerakan data harga di masa depan, guncangan pasar akibat anomali distribusi maupun iklim dapat diredam sedini mungkin melalui intervensi pencegahan oleh pemerintah [1][2][6][7][16].

2.2 Inflasi dan Harga Pangan

Inflasi adalah kenaikan harga barang dan jasa secara umum dan terus-menerus dalam periode waktu tertentu. Pengukuran inflasi dilakukan menggunakan Indeks Harga Konsumen (IHK), yakni indikator ekonomi yang mengukur rata-rata perubahan harga dari suatu paket komoditas (keranjang representatif barang dan jasa) yang biasa dikonsumsi oleh rumah tangga. Secara matematis, tingkat inflasi dihitung dengan cara membandingkan

nilai IHK pada periode waktu saat ini dengan nilai IHK pada periode dasar (waktu yang ditetapkan sebagai titik tolak ukur) [18].

Dalam pengklasifikasiannya, Badan Pusat Statistik (BPS) membagi inflasi berdasarkan sumber kerentanannya. Salah satu komponen yang paling krusial adalah inflasi volatile food (komponen bergejolak), yaitu inflasi yang murni didorong oleh fluktuasi harga bahan makanan pokok [19]. Secara umum, inflasi kelompok pangan memiliki karakteristik, faktor pendorong, dan dampak fundamental sebagai berikut:

1. Kontribusi Komoditas Pangan terhadap Inflasi

Komoditas pangan pokok seperti beras, minyak goreng, dan telur memiliki bobot kontribusi yang sangat besar dalam keranjang perhitungan IHK. Perubahan dan fluktuasi harga pada komoditas-komoditas dasar ini memberikan andil yang sangat dominan terhadap pembentukan angka inflasi secara keseluruhan, baik dalam eskalasi jangka pendek maupun jangka panjang [3].

2. Tekanan Inflasi Volatile Food dan Disparitas Regional

Inflasi kelompok bahan makanan (volatile food) bergerak secara sangat tidak stabil dan kerap kali melampaui tingkat rata-rata inflasi umum. Tingkat inflasi pangan ini juga memunculkan fenomena disparitas (kesenjangan) antarwilayah. Wilayah dengan akses infrastruktur logistik yang terbatas akan menghadapi tekanan inflasi yang jauh lebih tinggi dibandingkan kawasan urban, karena adanya beban biaya distribusi tambahan yang menciptakan perbedaan harga antar daerah secara drastis [6][7].

3. Faktor-Faktor Pendorong Fluktuasi Inflasi Pangan

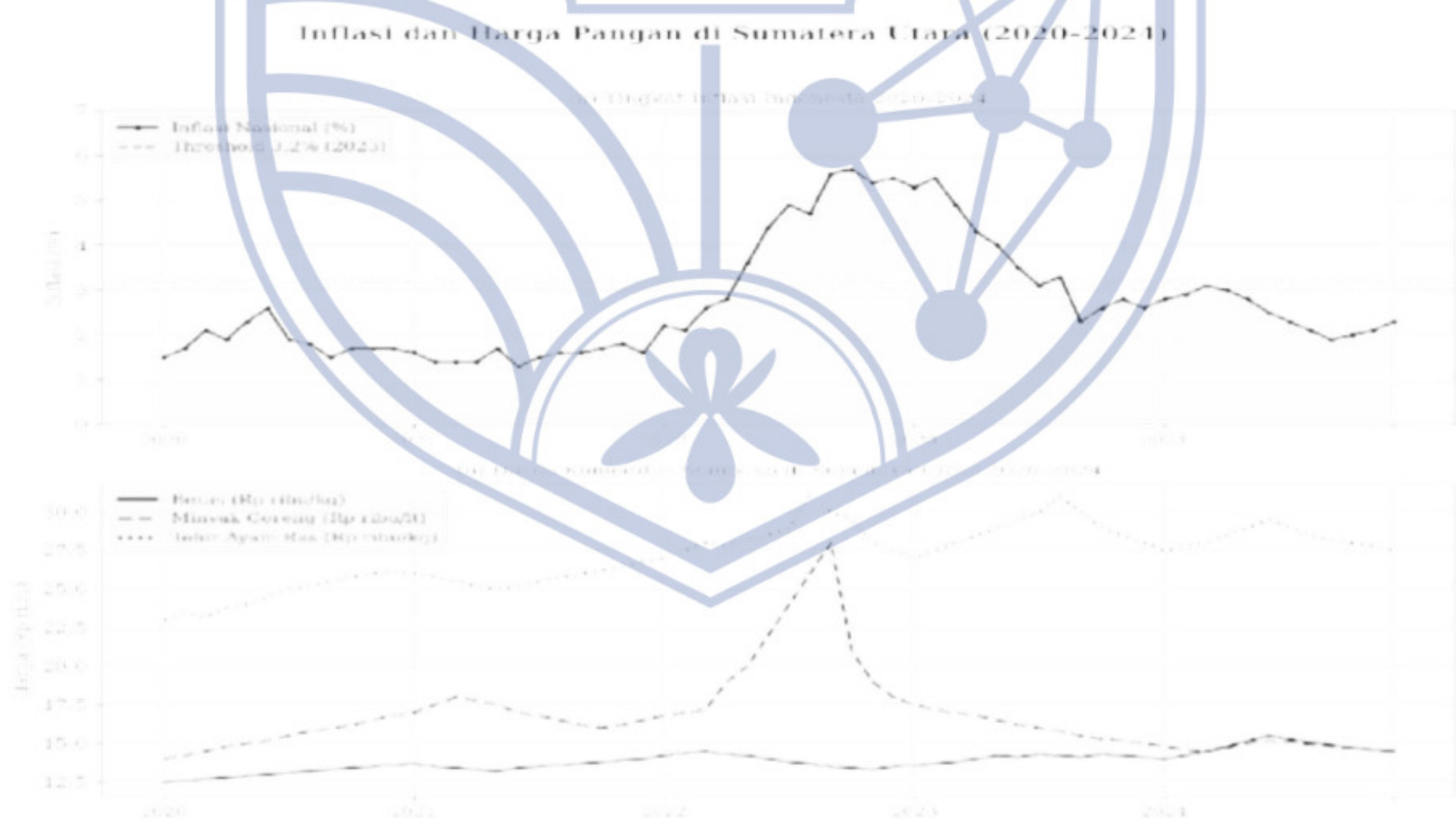
Pergerakan inflasi pangan cenderung mengalami fluktuasi yang ekstrem akibat tingginya kerentanan dari sisi penawaran (supply shock). Faktor-faktor utama yang secara langsung memicu lonjakan harga dan inflasi pangan ini meliputi:

- a. Gangguan pasokan akibat anomali iklim dan cuaca ekstrem yang menurunkan kuantitas serta kualitas hasil panen [26].
- b. Lonjakan biaya sarana produksi pertanian, seperti kenaikan harga pupuk di tingkat petani [20].
- c. Distribusi hasil panen yang tidak merata di berbagai sentra produksi.
- d. Hambatan operasional pada jalur logistik dan transportasi akibat kendala geografis atau infrastruktur [7].

4. Dampak terhadap Daya Beli Masyarakat

Tingginya angka inflasi pangan memberikan tekanan langsung terhadap daya beli riil masyarakat, khususnya bagi kelompok ekonomi menengah ke bawah. Kelompok masyarakat rentan yang mengalokasikan persentase terbesar dari pendapatannya murni untuk pemenuhan konsumsi pangan harian menanggung beban finansial paling berat saat terjadi lonjakan harga. Kenaikan harga bahan pokok utama ini secara otomatis menggerus daya beli, menghambat laju pertumbuhan ekonomi makro, dan menurunkan tingkat kesejahteraan masyarakat secara komprehensif [2][20].

Dinamika ini tercermin pada Gambar 2.1 yang menyajikan dua grafik mengenai kondisi inflasi dan harga pangan pokok di Sumatera Utara selama periode 2020–2024. Grafik (a) menunjukkan laju inflasi nasional yang sempat melampaui ambang batas 3,2% secara signifikan pada 2022–2023, mencapai puncaknya di kisaran 5,7%, sebelum kembali menurun mendekati level normal pada 2024. Sementara itu, Grafik (b) memperlihatkan fluktuasi harga tiga komoditas utama: beras, minyak goreng, dan telur ayam ras. Lonjakan harga minyak goreng yang sangat tajam pada awal 2022 mencerminkan dampak gangguan rantai pasok global, sedangkan harga beras dan telur menunjukkan kecenderungan meningkat secara bertahap sepanjang periode tersebut. Pola fluktuasi historis ini menegaskan perlunya sistem prediksi harga yang andal sebagai instrumen kebijakan stabilisasi pangan.



Gambar 2.1 Grafik Inflasi dan Harga Pangan Pokok di Sumatera Utara (2020–2024)
[7][19]

Berdasarkan pemaparan tersebut, dapat disimpulkan bahwa inflasi pangan khususnya pada kelompok *volatile food* merupakan instrumen penggerak utama laju inflasi secara umum. Pengukuran inflasi melalui perbandingan indeks harga (IHK) antarperiode secara presisi memperlihatkan tingginya tingkat kerentanan harga sembako terhadap berbagai gangguan eksternal, mulai dari anomali cuaca, lonjakan biaya produksi agrikultur, hingga masalah hambatan logistik yang memicu disparitas harga spasial. Hal ini terbukti secara empiris melalui data historis sebelumnya, di mana lonjakan tajam pada komoditas spesifik (seperti minyak goreng) berkontribusi langsung pada titik puncak lonjakan inflasi yang melampaui batas normal pada periode 2022-2023. Mengingat tingginya inflasi pangan secara langsung mengancam daya beli dan kelangsungan hidup kelompok masyarakat kelas menengah ke bawah, maka kemampuan memantau, memprediksi pergerakan harga, dan mengendalikan faktor-faktor pendorong inflasi sangat krusial dilakukan. Langkah ini penting demi merancang kebijakan intervensi pasar yang tanggap dalam menjaga stabilitas perekonomian serta melindungi kesejahteraan masyarakat [6][7][18-20][28].

2.3 Data Mining

Data mining adalah proses mengekstraksi pola, informasi, dan pengetahuan yang berguna dari kumpulan data berskala besar (*big data*) menggunakan pendekatan komputasi, statistik, dan *machine learning*. Tujuan utama dari data mining adalah menemukan pola tersembunyi (*hidden patterns*), hubungan antarvariabel yang kompleks, serta pengetahuan baru yang dapat dijadikan dasar dalam proses pengambilan keputusan analitis [24].

2.3.1 Fungsi Data Mining

Data mining memiliki beberapa fungsi komputasi utama yang dapat diterapkan dalam berbagai domain, termasuk dalam analisis data ekonomi dan pergerakan harga [24]:

1. Deskripsi (Description)

Fungsi deskripsi bertujuan untuk merangkum dan menggambarkan karakteristik data secara keseluruhan. Dalam analisis harga, fungsi ini bekerja untuk mengidentifikasi pola umum, seperti tren pergerakan harga, siklus musiman, dan distribusi harga di berbagai wilayah (misalnya: pola kenaikan harga beras pada masa paceklik atau siklus lonjakan menjelang hari raya).

2. Estimasi (*Estimation*)

Fungsi estimasi digunakan untuk memperkirakan nilai dari suatu variabel kuantitatif yang tidak diketahui berdasarkan korelasi dengan variabel lain yang nilainya telah diketahui. Fungsi ini digunakan untuk memperkirakan nilai suatu variabel target pada kondisi saat ini berdasarkan indikator-indikator pembentuknya yang telah diketahui dan memiliki korelasi terhadap variabel tersebut.

3. Prediksi (*Prediction*)

Fungsi prediksi adalah proses pemanfaatan pola yang diekstraksi dari data historis (masa lalu) untuk memproyeksikan atau meramalkan nilai variabel target di masa depan. Berbeda dengan estimasi yang berfokus pada kondisi saat ini, prediksi murni berorientasi pada masa depan. Fungsi ini digunakan untuk meramalkan nilai suatu variabel target pada rentang waktu ke depan berdasarkan pola data runtut waktu (time-series) dan variabel eksternal yang relevan.

4. Klasifikasi (*Classification*)

Fungsi klasifikasi bertujuan untuk memetakan atau mengkategorikan data ke dalam kelas-kelas diskrit yang telah ditentukan sebelumnya. Dalam pengolahan data harga, klasifikasi digunakan untuk mengelompokkan status kondisi pasar (misalnya kategori: harga stabil, waspada lonjakan, atau krisis) berdasarkan indikator tertentu.

2.3.2 Pengelompokan Data Mining

Berdasarkan pendekatan algoritmanya dalam memproses data, teknik-teknik dalam data mining dapat diklasifikasikan ke dalam beberapa kelompok utama sebagai berikut:

1. Pembelajaran Terarah (*Supervised Learning*)

Pendekatan ini bekerja pada data yang telah memiliki label atau kategori yang jelas sebagai acuan pembelajaran. Model komputasi mempelajari pola dari pasangan data input dan output yang sudah diketahui, kemudian menggunakan pola tersebut untuk mengklasifikasikan atau memprediksi data baru. Pendekatan ini lazim digunakan dalam tugas klasifikasi dan prediksi nilai numerik pada data terstruktur, mencakup algoritma seperti regresi, pohon keputusan, hingga Extreme Gradient Boosting [22][23].

2. Pembelajaran Tak Terarah (*Unsupervised Learning*)

Pendekatan ini bekerja pada data yang tidak memiliki label atau target keluaran. Tujuan utamanya adalah mengeksplorasi struktur, distribusi, atau pola tersembunyi secara mandiri dari dalam data. Teknik clustering dan *association rules* merupakan

bentuk implementasi utama dari pendekatan ini, yang berguna untuk pengelompokan wilayah berdasarkan pola harga maupun identifikasi hubungan [22][24].

3. Pembelajaran Semi-Terarah (*Semi-Supervised Learning*)

Pendekatan ini merupakan kombinasi dari dua pendekatan sebelumnya, dengan model dilatih menggunakan sebagian kecil data berlabel dan sebagian besar data tidak berlabel. Pendekatan ini berguna untuk mengatasi kendala ketika proses pemberian label pada big data membutuhkan sumber daya yang terlalu besar [22][24].

4. Pembelajaran Berbasis Penguatan (*Reinforcement Learning*)

Pendekatan ini menempatkan sebuah agen komputasi yang belajar mengambil keputusan melalui interaksi terus-menerus dengan suatu lingkungan simulasi. Model menerima umpan balik berupa penghargaan (*reward*) saat keputusannya tepat atau hukuman (*penalty*) saat salah. Pendekatan ini banyak digunakan pada otomatisasi sistem kontrol dinamis dan optimasi rantai pasok secara *real-time* [22].

Keempat pendekatan algoritmik di atas memberikan fleksibilitas dalam memilih metode yang paling sesuai dengan karakteristik data. Untuk memberikan gambaran yang lebih ringkas mengenai perbedaan keempat paradigma tersebut, perbandingannya dapat diamati pada Tabel 2.1. Dalam penelitian ini, pendekatan yang dipilih adalah pendekatan terarah (*supervised learning*) karena algoritma XGBoost membutuhkan data yang memiliki pasangan input dan output yang jelas, yaitu data historis harga sebagai input dan harga target di masa depan sebagai output. Ketiga pendekatan lainnya tidak diterapkan karena tidak sesuai dengan karakteristik data dan tujuan penelitian ini.

Tabel 2.1 Perbandingan Pendekatan Data Mining [22]

Pendekatan	Karakteristik	Contoh Algoritma	Butuh Label
Supervised Learning	Data berlabel; model belajar dari input-output yang diketahui	XGBoost, Random Forest, SVM, Regresi	Ya
Unsupervised Learning	Data tanpa label; mencari pola tersembunyi	K-Means, DBSCAN, PCA	Tidak
Semi-Supervised Learning	Sebagian data berlabel, sebagian tidak	Self-training, Label Propagation	Sebagian
Reinforcement Learning	Belajar melalui reward/penalty dari interaksi lingkungan	Q-Learning, Deep RL	Tidak

Proses untuk mendapatkan informasi dari berbagai pendekatan di atas dilakukan melalui serangkaian tahapan yang sistematis. Alur proses data mining secara umum terdiri dari enam tahapan berurutan: Data Mentah → Seleksi Data → *Preprocessing & Cleaning* → Transformasi Data → Data Mining → Pengetahuan/Pola. Tahap seleksi data melibatkan pemilihan subset data yang relevan dari basis data atau *data warehouse*. Selanjutnya dilakukan pembersihan dan normalisasi data untuk menangani nilai yang hilang (*missing values*), diikuti transformasi dan agregasi data agar siap diproses oleh algoritma. Pada tahap inti, algoritma data mining berupa klasifikasi, regresi, atau *cluster* diterapkan untuk mengekstraksi pengetahuan dan pola tersembunyi dari data [24]. Ilustrasi lengkap mengenai tahapan ini dapat dilihat pada Gambar 2.2 berikut.



Gambar 2.2 Proses Data Mining: Dari Data Mentah ke Pengetahuan [24]

Berdasarkan uraian mengenai pendekatan algoritmik dan alur proses di atas, data mining merupakan instrumen analitis tingkat lanjut yang krusial dalam mengurai kompleksitas *big data*. Keempat pendekatan algoritmik yang tersedia memberikan fleksibilitas dalam memilih metode yang paling sesuai dengan karakteristik data dan tujuan analisis yang hendak dicapai. Selain itu, seperti yang diringkas pada gambar alur proses sebelumnya, tahapan yang terstruktur dari hulu (data mentah) ke hilir (pengetahuan) memastikan bahwa kualitas data tetap terjaga sebelum diekstraksi oleh algoritma. Dengan gabungan antara fleksibilitas fungsi operasional (deskripsi, estimasi, klasifikasi, dan prediksi) dan proses pengolahan yang sistematis, data mining terbukti menjadi teknologi fundamental dalam membangun sistem analisis berbasis data ilmiah [22][24][29].

2.3.3 Cross-Industry Standard Process for Data Mining (CRISP-DM)

Cross-Industry Standard Process for Data Mining atau CRISP-DM merupakan standar proses yang paling populer bagi praktisi data science dalam mengelola siklus hidup sebuah proyek data mining. Metodologi ini bukan sekadar urutan kerja teknis melainkan sebuah kerangka kerja untuk memastikan setiap tahapan pengolahan data sejalan dengan

tujuan penyelesaian masalah nyata di lapangan. Penggunaan CRISP-DM dalam penelitian ini menjadi sangat relevan karena alurnya yang fleksibel dan memungkinkan peneliti untuk kembali ke tahapan sebelumnya jika ditemukan ketidakkonsistenan pada data. Efektivitas metodologi ini terlihat dari kemampuannya dalam membantu peneliti fokus pada ekstraksi pola tersembunyi dari sekumpulan data yang sangat besar dan kompleks [24].

Secara garis besar CRISP-DM mencakup enam tahapan utama yang saling berkaitan yaitu pemahaman bisnis/pemahaman masalah, pemahaman data, persiapan data, pemodelan, evaluasi, dan penyebaran model. Keunggulan utama dari standar ini terletak pada sifatnya yang iteratif atau berulang. Hal tersebut memungkinkan penyesuaian berkelanjutan terhadap perubahan data atau kebutuhan analisis yang dinamis. Proses pengembangan model tidak akan langsung melompat ke tahap teknis melainkan harus melewati validasi logika yang kuat sejak awal. Pendekatan terstruktur seperti ini mampu meningkatkan kualitas hasil pada berbagai studi yang memiliki tingkat variabilitas data tinggi [32].

Walaupun istilah asli dalam kerangka kerja ini menggunakan *Business Understanding*, namun tahap pertama ini sangat fleksibel untuk diadaptasi menjadi *Problem Understanding* atau *Domain Understanding* sesuai dengan konteks yang dikerjakan. Fokus utama dalam lingkup pengembangan teknologi adalah solusi masalah melalui pemahaman domain secara mendalam daripada sekadar aspek komersial. Keberhasilan sebuah proyek data mining sangat ditentukan oleh ketepatan dalam mendefinisikan masalah dari sudut pandang pengguna akhir [24]. Pemahaman domain mencakup pengetahuan mendalam tentang ekosistem dan faktor pemicu fluktuasi data. Implementasi model memang harus dibangun di atas pemahaman konteks yang kuat agar hasil yang didapatkan tidak mengalami bias [32].

Kerangka kerja ini tetap menjadi standar emas yang paling banyak diadopsi dalam berbagai proyek meskipun sudah diperkenalkan sejak lama. Literatur terbaru pada tahun 2021 menunjukkan bahwa metodologi ini tetap dominan karena strukturnya sangat adaptif terhadap perkembangan teknologi baru [33]. Alur kerja tersebut terbukti mampu menjembatani kompleksitas teknis dengan kebutuhan solusi praktis di dunia nyata. Fleksibilitas ini juga memungkinkan pengembang untuk lebih fokus pada kualitas data sebelum masuk ke tahap pemodelan. Faktor tersebut sangat krusial karena akurasi model bergantung penuh pada kebersihan data historis yang digunakan [34]. Penggunaan

pendekatan terstruktur ini secara signifikan dapat meminimalkan risiko kegagalan proyek dalam menghasilkan informasi yang akurat.

2.3.4 Prapemrosesan Data (Data Preprocessing)

Prapemrosesan data merupakan tahapan krusial untuk memastikan bahwa informasi yang diolah oleh algoritma memiliki kualitas tinggi. Salah satu masalah utama yang sering ditemukan dalam dataset mentah adalah keberadaan nilai kosong atau *missing value* yang dapat mengganggu performa model prediksi secara signifikan. Untuk mengatasi hal tersebut, teknik pengisian data maju atau yang dikenal sebagai *forward fill* menjadi salah satu solusi teknis yang sangat andal dalam analisis data deret waktu [33]. Prosedur ini bekerja dengan cara mengisi kolom yang kosong menggunakan nilai terakhir yang tersedia tepat sebelum kekosongan tersebut terjadi. Strategi ini sangat efektif untuk menjaga konsistensi urutan kronologis data tanpa merusak tren yang sedang berlangsung.

Selain teknik pengisian maju, terdapat metode interpolasi linear yang menawarkan pendekatan lebih dinamis dalam menangani data hilang pada titik waktu tertentu [35]. Berbeda dengan *forward fill* yang hanya menduplikasi nilai sebelumnya, interpolasi linear bekerja dengan cara mengestimasi nilai kosong berdasarkan tren antara dua titik data aktual yang mengapitnya. Teknik ini menggunakan perhitungan matematis untuk membentuk garis lurus yang menghubungkan nilai sebelum dan sesudah kekosongan sehingga menghasilkan estimasi nilai yang lebih proporsional terhadap pergerakan waktu [36]. Penggunaan teknik ini memberikan perkiraan nilai yang lebih mendekati realitas fluktuasi di lapangan karena mempertimbangkan arah pergerakan harga menuju titik waktu berikutnya.

Penggunaan teknik imputasi ini dilakukan dengan asumsi logis bahwa dalam rentang waktu yang singkat, nilai suatu variabel cenderung memiliki stabilitas pola sebelum terjadi perubahan data baru. Publikasi ilmiah pada tahun 2022 menyebutkan bahwa pemilihan metode penanganan data hilang yang tepat sangat krusial untuk menjaga integritas informasi historis tanpa menghilangkan observasi yang berharga [33]. Hal ini memungkinkan dataset tetap memiliki jumlah baris yang utuh sehingga informasi penting tetap dapat dipertahankan untuk proses pelatihan model pembelajaran mesin.

Hasil kajian terbaru pada tahun 2023 juga mengonfirmasi bahwa metode prapemrosesan ini merupakan standar dasar yang sangat tangguh dalam menjaga stabilitas pola tren lokal sebelum data diproses lebih lanjut oleh algoritma [34]. Melalui penerapan teknik pengisian yang sesuai, integritas dataset dapat dijamin keberlangsungannya

sehingga model machine learning mampu mempelajari pola fluktuasi secara lebih akurat. Hal tersebut menjadi faktor penentu dalam menghasilkan performa prediksi yang stabil dan meminimalkan bias akibat hilangnya data pada titik waktu tertentu sehingga hasil akhirnya menjadi lebih objektif.

2.4 Machine Learning

Machine learning adalah cabang dari kecerdasan buatan (*artificial intelligence*) yang memungkinkan sistem komputer untuk belajar dan meningkatkan kinerjanya dari pengalaman data tanpa diprogram secara eksplisit. Berbeda dengan pemrograman tradisional yang mengikuti instruksi statis, *machine learning* membangun model matematis berdasarkan data sampel (data latih) untuk membuat keputusan atau memprediksi nilai. Secara umum, *machine learning* beroperasi dengan mengidentifikasi pola kompleks yang tersembunyi di dalam data, termasuk hubungan antar variabel input dan variabel target yang sulit diukur melalui analisis statistik konvensional [22].

2.4.1 Supervised Learning

Supervised learning adalah paradigma *machine learning*, dengan melibatkan proses pelatihan model menggunakan dataset berlabel, yaitu data yang sudah memiliki pasangan variabel input (fitur) dan output (*target*) yang pasti [23]. Algoritma mempelajari fungsi pemetaan dari input ke output. Pemodelan fungsi pemetaan supervised learning ini dapat dilihat pada persamaan (1).

Fungsi Pemetaan Supervised Learning:

$$Y = f(X) + \epsilon \dots \dots \dots (1)$$

Keterangan:

Y = Variabel output atau target

X = Matriks variabel input atau fitur

$f(X)$ = Fungsi pemetaan pola yang dipelajari oleh model *machine learning*

ϵ = Error atau tingkat kesalahan acak yang tidak dapat dijelaskan oleh model

Model ensemble dalam supervised learning memiliki kemampuan untuk menangkap hubungan non-linear dan interaksi kompleks antar variabel, sehingga mampu menghasilkan tingkat kesalahan yang jauh lebih rendah dibandingkan model peramalan deret waktu konvensional [10][11][23].

1. Proses Training dan Testing

Proses *supervised learning* melibatkan dua tahap utama: *training* (pelatihan) dan *testing* (pengujian). Pada tahap training, dataset dibagi menjadi dua bagian: data latih (*training set*) dan data uji (*testing set*), umumnya dengan proporsi 70-80% untuk pelatihan dan 20-30% untuk pengujian. Model dilatih menggunakan data latih untuk mengenali pola, kemudian dievaluasi menggunakan data uji untuk mengukur kemampuannya dalam memprediksi data baru. Pembagian ini wajib dilakukan untuk mencegah *overfitting*, yaitu kondisi ketika model menghafal data latih secara sempurna namun gagal memprediksi data baru. Sebaliknya, *underfitting* terjadi ketika model terlalu sederhana sehingga gagal menangkap pola dasar dari data itu sendiri.

2. Cross-Validation

Validasi silang atau *cross-validation* merupakan prosedur evaluasi yang sangat penting untuk menilai performa model secara objektif serta menjamin ketahanan hasil terhadap bias pemisahan data. Dalam penelitian ini, teknik yang digunakan adalah *Time Series Split* yang dirancang khusus untuk menangani dataset dengan ketergantungan waktu agar urutan kronologis tetap terjaga. Berbeda dengan validasi silang konvensional yang membagi data secara acak, metode ini memastikan bahwa proses evaluasi dilakukan dengan cara yang menyerupai kondisi nyata di mana masa lalu digunakan untuk memprediksi masa depan. Publikasi ilmiah pada tahun 2022 menekankan bahwa menjaga integritas urutan waktu selama validasi adalah standar utama dalam menghindari kebocoran informasi dari masa depan ke masa lalu [35].

Prosedur ini bekerja dengan membagi data ke dalam beberapa lipatan secara berurutan. Setiap iterasi baru akan menambahkan data dari periode waktu berikutnya ke dalam set pelatihan dan menggunakan periode setelahnya sebagai set pengujian. Melalui pendekatan ini, model dipaksa untuk terus beradaptasi dengan tren yang berkembang seiring berjalannya waktu. Kajian teknologi pada tahun 2023 menyebutkan bahwa penggunaan validasi deret waktu memberikan gambaran akurasi yang jauh lebih realistis dan akurat jika dibandingkan dengan metode acak yang sering kali menghasilkan performa semu [36]. Implementasi teknik ini menjamin bahwa model yang dihasilkan memiliki kemampuan generalisasi yang baik saat menghadapi fluktuasi data di masa mendatang.

3. Feature Engineering

Feature engineering (rekayasa fitur) merupakan proses ekstraksi, pemilihan, dan transformasi variabel input mentah menjadi representasi data yang lebih informatif,

sehingga algoritma *machine learning* dapat mengidentifikasi pola dengan lebih optimal [23]. Implementasi rekayasa fitur pada pemodelan deret waktu (*time-series*) mencakup beberapa pendekatan, antara lain:

- a. Fitur Temporal: Ekstraksi komponen waktu secara spesifik (seperti indeks hari, bulan, atau kuartal) untuk memodelkan efek musiman dan pola periodik dalam data.
- b. Fitur *Lag* (Jeda Waktu): Pemanfaatan nilai observasi pada periode waktu sebelumnya ($t-n$) untuk memetakan dependensi historis dan autokorelasi terhadap prediksi masa depan.
- c. Fitur Statistik (*Rolling Window*) Penghitungan metrik agregat, seperti *moving average* (rata-rata bergerak) atau standar deviasi pada rentang waktu tertentu guna mereduksi *noise* dan memperhalus tren data.
- d. Fitur Interaksi: Penggabungan operasi matematis dari dua atau lebih variabel independen untuk merepresentasikan hubungan kompleks atau multivariat yang tidak linier.

Keseluruhan proses ini bertujuan untuk meningkatkan kapasitas model dalam menangkap tren jangka panjang, siklus musiman, serta anomali eksternal dengan tingkat presisi yang lebih tinggi [10].

2.4.2 Keunggulan Machine Learning

Pendekatan *machine learning* menawarkan berbagai keunggulan fundamental dibandingkan metode statistik tradisional:

1. Kemampuan Menangani Non-Linearitas

Metode statistik tradisional umumnya mengasumsikan hubungan linear (garis lurus) antara input dan output [23]. Faktanya, pergerakan harga dipengaruhi oleh hubungan non-linear yang kompleks (misalnya, curah hujan yang berlebihan atau terlalu sedikit sama-sama merusak hasil panen dan menaikkan harga). *Machine learning* mampu memetakan pola non-linear ini secara otomatis tanpa memerlukan campur tangan penentuan rumus secara manual [10][11].

2. Integrasi Variabel Eksternal secara Masif

Metode konvensional seringkali hanya mengandalkan riwayat masa lalu dari satu variabel. Sebaliknya, algoritma *machine learning* dirancang untuk mengintegrasikan berbagai jenis variabel eksternal (multivariat) ke dalam satu ruang pemodelan, seperti

penggabungan indikator anomali iklim global dengan data makroekonomi domestik [11].

3. Adaptasi terhadap Perubahan Pola

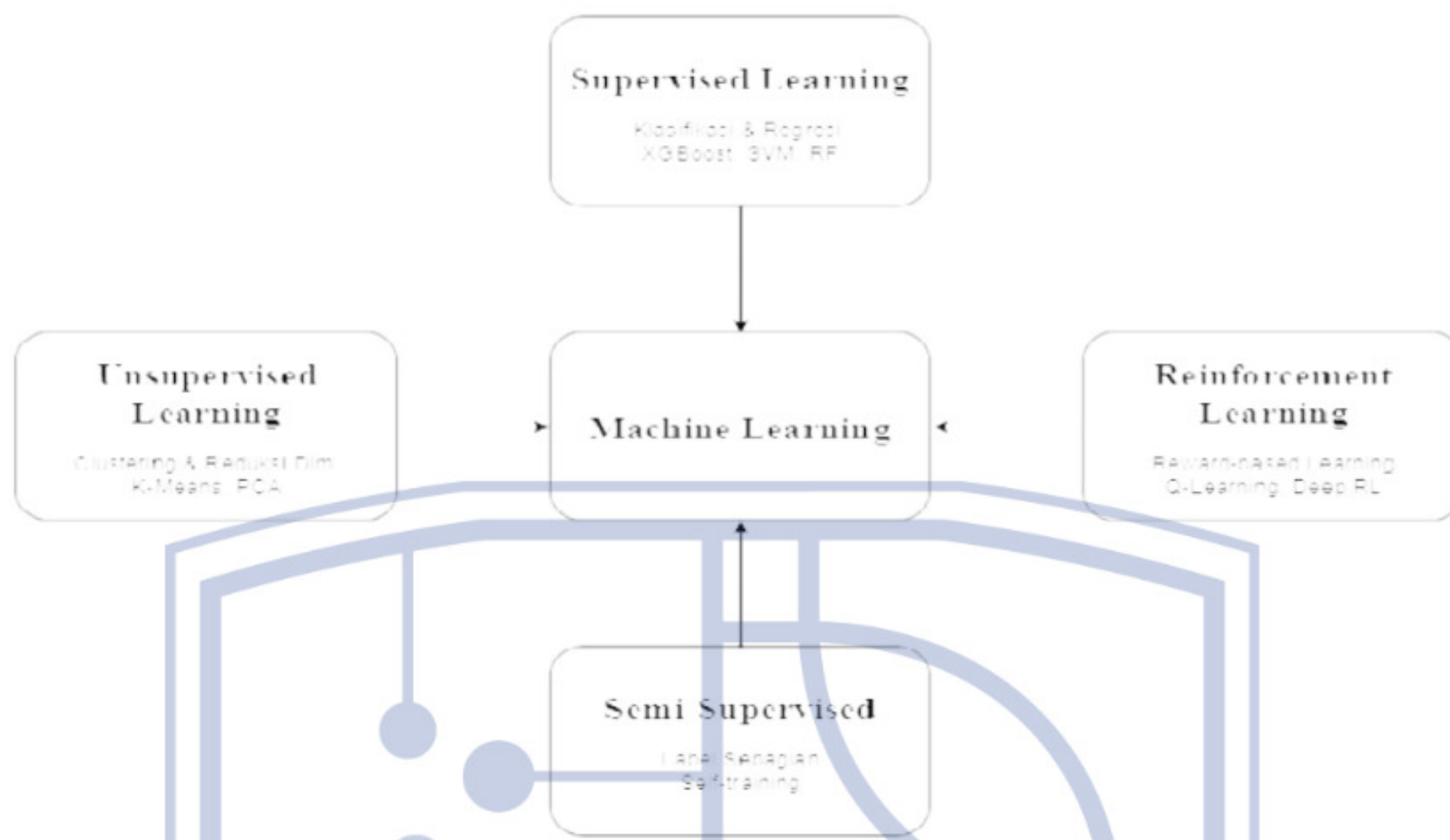
Model *machine learning* dapat dilatih ulang (retrained) secara berkala menggunakan data terbaru. Hal ini memastikan sistem selalu beradaptasi dengan perubahan struktural yang terjadi di lapangan, seperti perubahan kebijakan pemerintah, pergeseran pola konsumsi, maupun perbaikan infrastruktur distribusi [22].

4. Penanganan Data Berdimensi Tinggi

Machine learning memiliki ketahanan tinggi dalam memproses dataset yang memiliki ratusan variabel input (*high-dimensional data*) tanpa mengalami penurunan performa secara drastis [22][23]. Algoritma ini secara otomatis menyeleksi dan memberikan bobot lebih besar pada variabel yang memiliki pengaruh paling kuat terhadap pembentukan harga pasar.

Untuk memahami klasifikasi dari pendekatan ini, Gambar 2.3 di bawah ini menampilkan taksonomi *machine learning* yang secara umum terdiri dari empat paradigma utama. Pertama, *Supervised Learning* yang mencakup tugas klasifikasi dan regresi dengan contoh algoritma *XGBoost*, *SVM*, dan *Random Forest*. Kedua, *Unsupervised Learning* untuk pengelompokan dan reduksi dimensi (*K-Means*, *PCA*). Ketiga, *Semi-Supervised Learning* yang memanfaatkan data berlabel sebagian dengan teknik *self-training*. Keempat, *Reinforcement Learning* berbasis *reward* (*Q-Learning*, *Deep RL*). Dalam konteks penelitian ini, paradigma *supervised learning* dengan algoritma *XGBoost* dipilih karena kemampuannya yang unggul dalam menangani prediksi harga pangan yang bersifat non-linear dan multivariat [22].

Taksonomi Machine Learning: Jenis dan Algoritma Utama



Gambar 2.3 Taksonomi Machine Learning: Jenis dan Algoritma Utama [22]

Berdasarkan pemaparan di atas, dapat disimpulkan bahwa *machine learning* merupakan solusi komputasi tingkat lanjut yang mampu mengatasi keterbatasan analisis statistik konvensional. Seperti yang diilustrasikan pada taksonomi sebelumnya, *machine learning* menawarkan berbagai paradigma untuk berbagai skenario data, di mana paradigma *supervised learning* menjadi pilihan yang paling tepat untuk kebutuhan prediksi berbasis target. Melalui mekanisme pemetaan matematis input-output, teknik validasi silang (*cross-validation*), dan rekayasa fitur (*feature engineering*), algoritma pada paradigma ini mampu mengurai hubungan non-linear dan mengintegrasikan berbagai variabel makroekonomi secara bersamaan. Kemampuannya beradaptasi terhadap data berdimensi tinggi menjadikan *machine learning* sebagai instrumen yang tangguh dan akurat dalam meramalkan pergerakan data yang fluktuatif [10][11][21-23][30].

2.5 Algoritma XGBoost

Extreme Gradient Boosting (XGBoost) adalah implementasi lanjutan dari algoritma *gradient boosting* yang pertama kali diperkenalkan pada tahun 2016 sebagai sistem *boosting* pohon keputusan yang sangat skalabel dan efisien. XGBoost merupakan pustaka *machine learning* terdistribusi yang dirancang untuk bekerja secara adaptif dan portabel, menggunakan algoritma komputasi paralel untuk menyelesaikan permasalahan regresi dan klasifikasi data tabular secara cepat dan akurat [25]. Perlu dibedakan bahwa algoritma *gradient boosting* itu sendiri telah dikembangkan oleh Friedman sejak tahun 1999/2001 melalui kerangka *functional gradient descent*, sedangkan XGBoost merupakan

implementasi skalabel yang mengoptimalkan algoritma tersebut dengan menambahkan komputasi paralel, regularisasi L1/L2 terintegrasi, dan efisiensi memori sehingga jauh lebih cepat dan akurat dibandingkan implementasi gradient boosting sebelumnya [25]. XGBoost dilengkapi dengan mekanisme regularisasi canggih untuk mencegah *overfitting*, menjadikannya sangat sesuai untuk memprediksi data dengan volatilitas tinggi yang dipengaruhi oleh interaksi kompleks antara variabel ekonomi dan faktor eksternal [9][10-12][32].

Secara lebih spesifik, dalam konteks permasalahan regresi seperti prediksi harga komoditas pangan, XGBoost diimplementasikan melalui antarmuka *XGBRegressor*, yaitu implementasi XGBoost yang dirancang khusus untuk tugas regresi (prediksi nilai kontinu). *XGBRegressor* menggunakan fungsi kerugian berbasis kuadrat (*squared error loss*) sebagai fungsi objektif utamanya, serta mendukung seluruh mekanisme regularisasi, optimasi berbasis gradien, dan *early stopping* yang menjadi keunggulan inti XGBoost. Antarmuka ini kompatibel dengan ekosistem *scikit-learn* sehingga memudahkan integrasi dengan berbagai *pipeline* pemrosesan data dan teknik validasi seperti *cross-validation* dan *grid search* dalam proses optimasi model [25].

2.5.1 Konsep Gradient Boosting

Gradient Boosting merupakan teknik *ensemble* yang bekerja dengan membangun model secara sekuensial (berurutan), dengan setiap model baru bertujuan untuk mengoreksi kesalahan (*error*) yang dibuat oleh model sebelumnya [25]. Berbeda dengan teknik *ensemble* lain yang membangun pohon secara paralel, *Gradient Boosting* membangun pohon keputusan satu per satu untuk meminimalkan residual (selisih nilai aktual dan prediksi).

Setiap penambahan model dalam *ensemble* adalah pohon keputusan yang lemah (*weak learner*), umumnya berupa pohon dengan kedalaman terbatas. Kombinasi dari banyak pohon lemah ini secara akumulatif menghasilkan model yang sangat kuat (*strong learner*). Kontribusi masing-masing pohon digabungkan secara aditif untuk menghasilkan prediksi akhir. Secara matematis, prediksi akhir XGBoost pada data ke-*i* dinyatakan sebagai penjumlahan kontribusi dari seluruh *K* pohon keputusan sebagaimana ditunjukkan pada persamaan (2):

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), \quad f_k \in F \dots\dots\dots (2)$$

Keterangan:

f_k = fungsi pohon keputusan ke- k

x_i = vektor fitur sampel ke- i ,

F = ruang semua pohon keputusan yang mungkin (CART).

Fungsi objektif XGBoost meminimalkan dua komponen sekaligus: fungsi kerugian (*loss function*) yang mengukur selisih prediksi terhadap nilai aktual, serta suku regularisasi yang mengontrol kompleksitas model guna mencegah *overfitting* [25]. Fungsi objektif tersebut dinyatakan secara matematis pada persamaan (3):

$$Obj(\theta) = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k) \dots \dots \dots (3)$$

Keterangan

$l(y_i, \hat{y}_i)$ = *loss function* yang mengukur selisih antara nilai aktual y_i $\hat{y}_i$

$\Omega(f_k)$ = suku regularisasi yang mengontrol kompleksitas model.

Suku regularisasi Ω secara eksplisit mengandung tiga komponen penalti sebagaimana dinyatakan pada persamaan (4):

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 + \alpha \sum_{j=1}^T |w_j| \dots \dots \dots (4)$$

Keterangan

T = jumlah daun pada pohon

w_j = bobot prediksi pada daun ke- j

γ = penalti terhadap jumlah daun

λ = parameter regularisasi L2 (*reg_lambda*)

α = parameter regularisasi L1 (*reg_alpha*)

Pada setiap iterasi ke- k , XGBoost menggunakan pendekatan ekspansi Taylor orde kedua untuk mengaproksimasi *loss function*. Gradien orde pertama (g_i) dan Hessian orde kedua (h_i) dihitung sebagaimana ditunjukkan pada persamaan (5) dan (6):

$$g_i = \frac{\partial l(y_i, \hat{y}_i^{(k-1)})}{\partial \hat{y}_i^{(k-1)}} \dots \dots \dots (5)$$

$$h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{(k-1)})}{\partial (\hat{y}_i^{(k-1)})^2} \dots \dots \dots (6)$$

Keterangan

g_i = gradien negatif (pseudo-residual) yang menjadi target pembelajaran pohon ke- k .

h_i = Hessian yang digunakan sebagai bobot pada proses pembelahan (*splitting*).

2.5.2 Parameter Utama XGBoost

Performa XGBoost sangat bergantung pada konfigurasi *hyperparameter*. Parameter utama yang mengendalikan arsitektur pembelajaran algoritma ini meliputi [25]:

1. *learning_rate*: Mengontrol besaran koreksi (*step size*) pada setiap iterasi pohon. Nilai kecil membuat model belajar lebih lambat dan stabil, namun membutuhkan jumlah pohon yang lebih banyak.
2. *max_depth*: Menentukan kedalaman maksimum dari setiap cabang pohon keputusan. Semakin dalam pohon, semakin kompleks interaksi variabel yang ditangkap, namun meningkatkan risiko *overfitting*.
3. *n_estimators*: Menentukan jumlah total pohon keputusan yang akan dibangun di dalam sistem ensemble.
4. *subsample*: Proporsi baris data latih yang diambil secara acak untuk membangun satu pohon. Pengambilan sampel acak ini mencegah model menghafal keseluruhan dataset.
5. *colsample_bytree*: Proporsi jumlah kolom (fitur/variabel) yang dipilih secara acak untuk setiap pohon, memastikan agar setiap pohon mengeksplorasi variabel yang berbeda-beda.
6. *reg_lambda*: Parameter regularisasi L2 yang menekan bobot prediksi agar tidak menjadi terlalu ekstrem.
7. *reg_alpha*: Parameter regularisasi L1 yang mendorong sparsity, yaitu mengeliminasi (memberikan bobot nol pada) fitur-fitur yang dianggap tidak memiliki pengaruh terhadap target.

Untuk memberikan gambaran analitis yang lebih sistematis, skema pemetaan antara atribut hiperparameter, rentang uji lazim optimal (*empirical scope tuning limit values*), beserta deteksi rasional efek komputasi algoritma manakala fungsi hiperparameter melewati batas maksimum stabilitas nilai dapat disimak pada Tabel 2.2 berikut.

Tabel 2.2 Parameter Utama XGBoost Dan Rentang Nilainya [25]

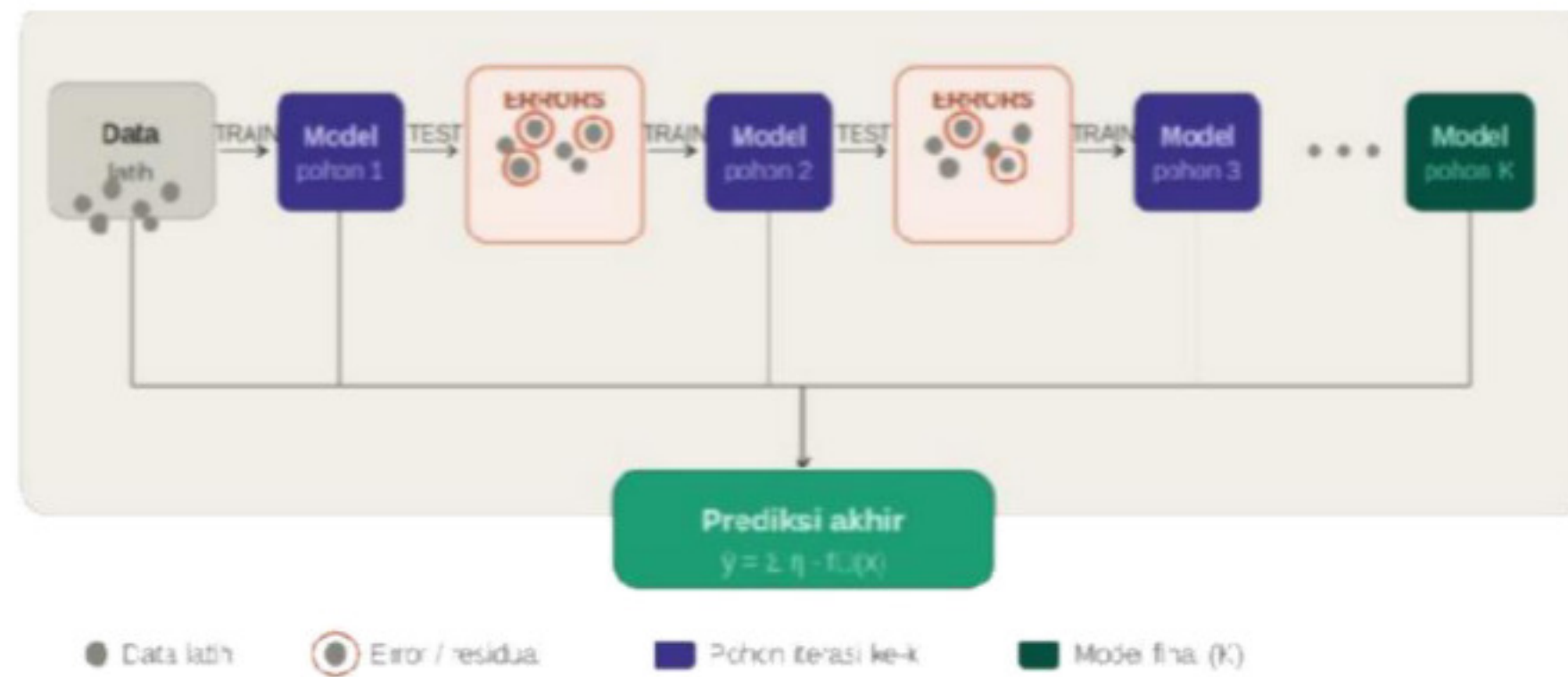
Paramete r	Fungsi	Rentang Nilai	Pengaruh Jika Terlalu Besar
<i>learning_rate</i>	Mengontrol kontribusi tiap pohon terhadap prediksi akhir	0,01 - 0,3	Konvergensi cepat tapi <i>overfitting</i>
<i>max_depth</i>	Kedalaman maksimum tiap pohon keputusan	3 - 7	Model terlalu kompleks, <i>overfitting</i>
<i>n_estimators</i>	Jumlah total pohon dalam	300 - 800	Waktu komputasi

	ensemble		meningkat, overfitting
subsample	Proporsi data latih yang digunakan per pohon	0,7 - 0,9	Diversitas antarpohon berkurang, overfitting
colsample_bytree	Proporsi fitur yang dipilih acak per pohon	0,7 - 0,9	Setiap pohon kurang bervariasi
reg_lambda (L2)	Regularisasi L2 untuk mencegah bobot terlalu besar	0,5 - 1,5	Model terlalu dipenalti, underfitting
reg_alpha (L1)	Regularisasi L1 untuk mendorong sparsity fitur	0 - 0,5	Banyak fitur di-nolkan, underfitting

Berdasarkan tinjauan diatas, pengaturan *hyperparameter* pada arsitektur XGBoost memegang peran fundamental dalam membangun model prediktif yang adaptif terhadap data fluktuatif. Pengelolaan parameter ini bukan sekadar berfungsi untuk mempercepat komputasi dan meminimalkan galat (*error*), melainkan untuk merancang konfigurasi pemodelan yang proporsional. Sebagai contoh, kompleksitas percabangan pohon (*max_depth*) harus selalu dikalibrasi selaras dengan laju pembelajarannya (*learning_rate*). Lebih lanjut, pengintegrasian parameter regularisasi L1 (*alpha*) dan L2 (*lambda*) bertugas memberikan batasan agar model tetap stabil. Konfigurasi komprehensif pada parameter-parameter inilah yang mencegah model dari risiko *overfitting* (hanya menghafal data latih), sehingga sistem mampu menghasilkan generalisasi prediksi yang akurat, presisi, dan tangguh saat dihadapkan pada data pengujian yang baru [10-12][25][31].

2.5.3 Prosedur Algoritma XGBoost

Algoritma XGBoost memiliki serangkaian tahapan prosedural dalam membangun model prediktif berbasis *gradient tree boosting*. Keseluruhan proses ini dieksekusi secara berulang dengan menambahkan pohon keputusan baru secara struktural. Secara lebih spesifik, alur lengkap algoritma XGBoost ditunjukkan pada Gambar 2.4, mulai dari tahap inisialisasi, perhitungan gradien dan Hessian, pembentukan pohon, penentuan bobot daun terbaik, hingga pembaruan hasil prediksi secara iteratif.



Algoritma XGBRegressor — Prosedur Sekuensial Gradient Boosting	
Input	Data latih: $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, jumlah pohon K , learning rate η
Inisialisasi	$\hat{y}_i^{(0)} = \underset{c}{\operatorname{argmin}} \sum_i l(y_i, c) = \bar{y}$ (rata-rata target)
Untuk $k = 1 \dots K$:	
Langkah 1	Hitung gradien: $g_i = \partial l(y_i, \hat{y}_i^{(k-1)}) / \partial \hat{y}_i^{(k-1)}$
Langkah 2	Hitung Hessian: $h_i = \partial^2 l(y_i, \hat{y}_i^{(k-1)}) / \partial (\hat{y}_i^{(k-1)})^2$
Langkah 3	Cari pembelahan terbaik dengan $\text{Gain} = \frac{1}{2} [GL^2 / (HL + \lambda) + GR^2 / (HR + \lambda) - G^2 / (H + \lambda)] - \gamma$
Langkah 4	Hitung bobot optimal setiap daun j : $w_j^* = -G_j / (H_j + \lambda)$
Langkah 5	Update prediksi: $\hat{y}_i^{(k)} = \hat{y}_i^{(k-1)} + \eta \cdot f_k(x_i)$
Output	Prediksi akhir: $\hat{y}_i = \sum_{k=1}^K \eta \cdot f_k(x_i)$

Gambar 2.4 Tahapan Algoritma XGBoost (Gradient Boosting Sekuensial) [25]

Inisialisasi model prediksi awal:

$$\hat{y}_i^{(0)} = \bar{y} = \underset{c}{\operatorname{argmin}} \sum_i l(y_i, c) = \bar{y} \dots \dots \dots (7)$$

Keterangan:

$\hat{y}_i^{(0)}$ = nilai prediksi awal untuk setiap observasi pada iterasi ke-0

$\underset{c}{\operatorname{argmin}}$ = untuk mencari nilai konstanta yang meminimalkan total kerugian

$l(y_i, c)$ = fungsi kerugian antara nilai aktual dan nilai konstanta yang ditetapkan

$\bar{y}$ = nilai rata-rata dari seluruh data target pada variabel dependen

Perhitungan gradien (turunan pertama fungsi loss):

$$g_i = \frac{\partial L}{\partial \hat{y}_i} \dots \dots \dots (8)$$

Keterangan:

g_i = turunan pertama fungsi loss terhadap prediksi $\hat{y}_i$

L = fungsi loss (misalnya MSE untuk regresi)

Perhitungan Hessian (turunan kedua fungsi loss):

$$h_i = \frac{\partial^2 L}{\partial \hat{y}_i^2} \dots\dots\dots (9)$$

Keterangan:

h_i = turunan kedua fungsi loss terhadap prediksi $\hat{k}_i$

Penentuan pembelahan terbaik menggunakan nilai Gain:

$$Gain = \frac{1}{2} \left(\frac{GL^2}{HL + \lambda} + \frac{GR^2}{HR + \lambda} - \frac{(GL + GR)^2}{HL + HR + \lambda} \right) - \gamma \dots\dots\dots (10)$$

Keterangan:

G_L = jumlah gradien pada cabang kiri pohon

G_R = jumlah gradien pada cabang kanan pohon

H_L, H_R = jumlah Hessian pada cabang kiri dan kanan pohon

λ = parameter regularisasi L2

γ = parameter penalti kompleksitas pohon (minimum Gain untuk melakukan pembelahan)

Perhitungan bobot optimal setiap daun pohon:

$$w^* = \frac{-G}{H + \lambda} \dots\dots\dots (11)$$

Keterangan:

w^* = bobot nilai optimal pada setiap daun pohon keputusan

G = total jumlah gradien pada satu daun

H = total jumlah Hessian pada satu daun

Pembaruan prediksi secara aditif (output akhir setelah K iterasi):

$$\hat{y}_t = \hat{y}_{t-1} + \eta \cdot f_t(x) \dots\dots\dots (12)$$

Keterangan:

$\hat{k}_t$ = nilai prediksi setelah iterasi ke- t

η = *learning rate* yang mengontrol kontribusi setiap pohon ($0 < \eta \leq 1$)

$f_t(x)$ = pohon keputusan baru yang ditambahkan pada iterasi ke- t

Prediksi akhir model (Output):

$$\hat{y}_i = \sum_{k=1}^K \eta * f_1(x_1) \dots\dots\dots (13)$$

Keterangan:

$\hat{y}_i$ = nilai prediksi awal untuk setiap observasi pada iterasi ke-1

K = jumlah total pohon atau iterasi yang telah dibangun

η = learning rate yang digunakan untuk mengontrol besaran koreksi model

$f_k(x_i)$ = fungsi prediksi dari pohon keputusan pada iterasi ke-k

Secara sistematis, alur kerja XGBoost pada Gambar 2.4 menunjukkan proses pembelajaran aditif yang meminimalkan residual secara bertahap dari pohon pertama hingga pohon ke-K. Algoritma ini bekerja dengan memanfaatkan nilai gradien dan Hessian sebagai informasi utama untuk menentukan arah perbaikan model serta mencari titik pembelahan terbaik berdasarkan nilai Gain. Melalui perhitungan bobot optimal pada setiap daun yang melibatkan parameter regularisasi, setiap pohon baru akan secara konsisten mengoreksi kesalahan dari tahap sebelumnya. Hasil akhirnya adalah akumulasi seluruh kontribusi pohon yang dikalikan dengan *learning rate* untuk menghasilkan nilai prediksi dengan tingkat akurasi yang paling optimal [25][31].

2.6 Model Early Warning

Model *early warning* (model peringatan dini) merupakan pendekatan analitis yang dirancang untuk mendeteksi secara dini sinyal-sinyal awal dari potensi ketidakstabilan, anomali, atau penyimpangan pada suatu variabel yang dipantau. Model ini berperan sebagai instrumen pencegahan proaktif yang memungkinkan pemangku kepentingan mengambil tindakan sebelum kondisi kritis benar-benar terjadi, bukan hanya mencatat kejadian yang telah berlalu secara reaktif [32].

2.6.1 Definisi dan Tujuan Model Early Warning

Secara konseptual, model *early warning* didefinisikan sebagai suatu mekanisme analitis yang digunakan untuk mendeteksi, menganalisis, dan memberikan indikasi kemungkinan terjadinya kondisi anomali atau kritis dalam rentang waktu tertentu. Berbeda dengan sistem yang bersifat komprehensif, model ini berfokus pada proses pengolahan data dan pembangkitan sinyal peringatan. Fungsi utama dari model *early warning* meliputi:

1. Pemantauan kondisi variabel berdasarkan data historis dan data terkini,
2. Deteksi penyimpangan atau anomali terhadap pola normal yang telah dipelajari,
3. Pembangkitan sinyal peringatan (*warning signal*) ketika ambang batas tertentu terlampaui, serta
4. Penyediaan informasi sebagai dasar pendukung pengambilan keputusan secara lebih dini.

Tujuan utama dari model *early warning* adalah menyediakan informasi dini yang dapat digunakan sebagai dasar dalam proses pengambilan keputusan. Secara umum, fungsi dari model ini meliputi pemantauan kondisi, deteksi anomali, serta penyampaian indikasi risiko secara lebih awal sebelum kondisi tersebut berkembang menjadi lebih serius. Dalam konteks ekonomi dan pangan, model ini banyak digunakan untuk mengantisipasi fluktuasi harga dan potensi gangguan stabilitas pasar [3][32].

2.6.2 Peran Machine Learning dalam Model Early Warning

Perkembangan teknologi machine learning telah memberikan kontribusi signifikan dalam peningkatan kinerja model *early warning*. Pendekatan ini memungkinkan sistem untuk mempelajari pola dari data secara otomatis, termasuk hubungan yang kompleks dan non-linear antar variabel, tanpa memerlukan aturan eksplisit yang ditentukan sebelumnya [22][32].

Dalam literatur, model *early warning* berbasis *machine learning* umumnya dibangun melalui proses pelatihan menggunakan data historis, di mana model belajar mengenali pola dan tren dari variabel yang diamati. Selanjutnya, model tersebut digunakan untuk mengevaluasi data baru guna mendeteksi adanya penyimpangan dari pola yang telah dipelajari. Apabila ditemukan indikasi anomali atau perubahan signifikan, maka model dapat memberikan sinyal sebagai bentuk peringatan dini [10][11].

Keunggulan pendekatan machine learning dalam model *early warning* terletak pada kemampuannya dalam menangani data dalam jumlah besar serta kemampuannya beradaptasi terhadap perubahan pola dari waktu ke waktu. Hal ini menjadikan pendekatan ini banyak digunakan dalam berbagai studi prediksi [10][11][32].

2.7 Metrik Evaluasi

Metrik evaluasi adalah alat ukur kuantitatif yang digunakan untuk memvalidasi dan mengukur kualitas performa model *machine learning* dalam menghasilkan prediksi. Pemilihan metrik yang tepat sangat krusial karena setiap metrik memberikan perspektif matematis yang berbeda mengenai karakteristik tingkat kesalahan model. Dalam permasalahan regresi untuk memprediksi nilai kontinu, salah satu metrik yang paling umum dan banyak digunakan adalah *Mean Absolute Error* (MAE), yang mengukur rata-rata selisih absolut antara nilai prediksi dan nilai aktual secara langsung dan intuitif. Dibandingkan metrik evaluasi lainnya, MAE dinilai lebih mudah diinterpretasikan karena menghasilkan nilai kesalahan dalam satuan yang sama dengan variabel target, sehingga lebih komunikatif dalam berbagai konteks pelaporan dan pengambilan keputusan [8][23].

2.7.1 Konsep Teori Mean Absolute Error (MAE)

Mean Absolute Error (MAE) adalah metrik evaluasi yang mengukur rata-rata selisih absolut antara nilai prediksi model dan nilai aktual, tanpa membedakan arah kesalahan (positif atau negatif). MAE menghitung besaran deviasi secara proporsional dan merata pada seluruh sampel data uji, sehingga menghasilkan nilai kesalahan dalam satuan yang sama dengan variabel target [8][23]. Perhitungan MAE diformulasikan pada persamaan (7) berikut.

Menghitung *Mean Absolute Error* (MAE):

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \dots \dots \dots (14)$$

Keterangan:

n = Jumlah total sampel data pengujian

y_i = Nilai target aktual pada data ke- i

$\hat{y}_i$ = Nilai hasil prediksi pada data ke- i

Semakin kecil nilai MAE (mendekati nol), maka semakin tinggi tingkat akurasi suatu model prediksi karena rata-rata simpangan antara nilai prediksi dan nilai aktual semakin minim. Keunggulan utama dari metrik evaluasi MAE adalah kemampuannya untuk menyajikan nilai kesalahan (*error*) dalam satuan yang sama dengan variabel target. Hal ini membuat hasil evaluasi model menjadi lebih intuitif dan mudah diinterpretasikan secara langsung tanpa memerlukan perhitungan atau konversi lanjutan [8][23].

2.7.2 Perbandingan MAE dengan Metrik Evaluasi Konvensional

Dalam mengevaluasi performa model prediksi terutama pada fluktuasi harga pangan, terdapat beberapa metrik yang umum digunakan yaitu *Mean Absolute Error* (MAE), *Root Mean Square Error* (RMSE), dan koefisien determinasi R^2 . Pemilihan metrik yang tepat sangat krusial karena setiap indikator memiliki karakteristik pengukuran yang berbeda dalam menangkap tingkat akurasi model [8][23]:

1. MAE vs RMSE: Ketahanan terhadap Anomali Data

Perbedaan mendasar antara MAE dan RMSE terletak pada cara keduanya memperlakukan kesalahan prediksi. MAE menghitung rata-rata selisih absolut antara nilai prediksi dan nilai aktual secara proporsional. Sebaliknya, RMSE mengakumulasi kesalahan dengan mekanisme penalti kuadrat sehingga nilai error yang besar akan dikuadratkan terlebih dahulu sebelum dirata-ratakan. Mekanisme penalti kuadrat pada

RMSE menyebabkan metrik ini sangat sensitif terhadap keberadaan pencilan atau *outlier* dalam data [8].

Apabila terdapat nilai prediksi yang menyimpang jauh dari nilai aktual akibat volatilitas harga komoditas yang tidak menentu, RMSE akan menghasilkan nilai yang cenderung membengkak dan dapat mendistorsi gambaran akurasi model secara keseluruhan [10]. MAE dinilai lebih stabil dalam merepresentasikan kesalahan rata-rata karena struktur kalkulasinya bersifat linier dan tidak memberikan beban berlebih pada satu atau dua kesalahan ekstrem yang mungkin muncul dalam dataset [30].

2. MAE vs R^2 : Kepadanan Dimensi Pengukuran

Koefisien determinasi R^2 merupakan indeks statistik yang mengukur proporsi variansi variabel dependen yang dapat dijelaskan oleh model regresi [23]. Meskipun sering digunakan, R^2 bersifat adimensional atau tidak memiliki satuan sehingga hanya mencerminkan tingkat kecocokan relatif model terhadap pola data tanpa memberikan gambaran konkret mengenai besaran deviasi dalam satuan mata uang yang nyata. Hal ini terkadang menyulitkan dalam pengambilan keputusan praktis yang membutuhkan angka kepastian selisih harga [24].

MAE mengatasi keterbatasan tersebut dengan menghasilkan nilai deviasi dalam satuan yang identik dengan variabel target. Jika model memprediksi harga beras, maka nilai MAE yang dihasilkan juga dalam satuan Rupiah per kilogram. Kesepadanan dimensi ini memungkinkan laporan akurasi model diinterpretasikan secara langsung oleh pemangku kepentingan tanpa memerlukan transformasi statistik tambahan [12]. MAE memberikan informasi yang lebih komunikatif dalam konteks ekonomi pangan karena menunjukkan seberapa jauh secara rata-rata prediksi meleset dari harga pasar sebenarnya [16].

3. Urgensi Penggunaan Mean Squared Error (MSE) dalam Pemodelan

Meskipun MAE unggul dalam hal interpretasi hasil akhir, dalam proses pembangunan model *machine learning* seperti XGBoost, penggunaan *Mean Squared Error* (MSE) memiliki peran yang berbeda dan sangat fundamental. Perlu dipahami bahwa dalam bab selanjutnya, MSE tidak hanya dipandang sebagai alat ukur akurasi semata, melainkan sebagai fungsi objektif atau *loss function* utama yang mengarahkan proses pembelajaran model [25].

MSE dipilih sebagai landasan dalam proses optimasi karena sifat matematisnya yang *differentiable* atau dapat diturunkan. Algoritma seperti XGBoost memerlukan nilai

turunan pertama yang disebut gradien dan turunan kedua yang disebut hessian untuk menentukan arah perbaikan pohon keputusan di setiap iterasinya [25][31]. Tanpa fungsi loss seperti MSE, proses minimisasi kesalahan secara iteratif akan sulit dilakukan secara komputasi. Oleh karena itu, MSE berperan sebagai "mesin" yang memandu model untuk belajar dari kesalahan, sementara MAE tetap menjadi tolok ukur utama untuk memvalidasi seberapa baik hasil akhir prediksi tersebut dalam konteks dunia nyata [23].

2.7.3 Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error (MAPE) merupakan metrik evaluasi yang mengukur tingkat kesalahan prediksi model dalam bentuk persentase relatif terhadap nilai aktual. Berbeda dengan MAE yang menghasilkan nilai kesalahan dalam satuan asli variabel target, MAPE mengekspresikan deviasi sebagai proporsi terhadap nilai yang sesungguhnya sehingga bersifat bebas satuan (unit-free). Karakteristik ini menjadikan MAPE sangat berguna ketika diperlukan perbandingan akurasi antar model pada skala atau satuan yang berbeda-beda [22][23][41].

Secara matematis, MAPE diformulasikan sebagai rata-rata dari selisih absolut antara nilai aktual dan nilai prediksi yang dibagi dengan nilai aktual, dikalikan seratus persen untuk memperoleh ekspresi persentase. Perhitungan MAPE diformulasikan pada persamaan (15) berikut.

$$MAPE = \left(\frac{1}{n}\right) \times \sum \left| \frac{(y_i - \hat{y}_i)}{y_i} \right| \times 100\% \dots\dots\dots (15)$$

Keterangan:

n = Jumlah total sampel data pengujian

y_i = Nilai aktual pada data ke- i

$\hat{y}_i$ = Nilai hasil prediksi pada data ke- i

Semakin kecil nilai MAPE, semakin tinggi akurasi model prediksi yang dihasilkan. Nilai MAPE yang mendekati nol menunjukkan bahwa prediksi model hampir sepenuhnya sesuai dengan data aktual. Sebaliknya, nilai MAPE yang besar mengindikasikan deviasi yang signifikan antara hasil prediksi dan nilai nyata di lapangan. Salah satu keunggulan utama MAPE adalah kemampuannya memberikan gambaran akurasi yang mudah dikomunikasikan kepada pemangku kepentingan non-teknis karena dinyatakan dalam satuan persen yang familiar dan intuitif [41].

Dalam literatur dan praktik industri, terdapat standar interpretasi yang umum digunakan untuk menilai kualitas model berdasarkan nilai MAPE yang dihasilkan. Standar

ini pertama kali dipopulerkan oleh Lewis (1982) dan kemudian diadopsi secara luas dalam berbagai publikasi ilmiah bidang peramalan dan analitik data. Kategori standar interpretasi MAPE tersebut disajikan pada Tabel 2.2 berikut.

Tabel 2.3 Standar Interpretasi Nilai MAPE [41]

Nilai MAPE	Kategori Akurasi	Interpretasi
$\text{MAPE} < 10\%$	Sangat Akurat	Model memiliki akurasi prediksi yang sangat tinggi dan layak diterapkan
$10\% \leq \text{MAPE} < 20\%$	Akurat Baik	Model masih dapat diterima untuk aplikasi umum dalam konteks prediksi bisnis
$20\% \leq \text{MAPE} < 50\%$	Akurasi Sedang	Model memerlukan penyesuaian lebih lanjut sebelum dapat diaplikasikan
$\text{MAPE} \geq 50\%$	Akurasi Rendah	Model tidak layak digunakan karena tingkat kesalahan prediksi terlalu besar

Berdasarkan Tabel 2.2, model dengan nilai MAPE di bawah 10% dikategorikan memiliki akurasi yang sangat tinggi dan sangat layak diterapkan dalam aplikasi dunia nyata. Standar ini menjadi acuan yang banyak digunakan dalam penelitian prediksi harga komoditas, prediksi permintaan energi, serta berbagai domain peramalan deret waktu lainnya. Dibandingkan dengan MAE, MAPE memiliki kelebihan dalam hal komparabilitas antar komoditas yang memiliki skala harga berbeda-beda, namun memiliki kelemahan ketika nilai aktual mendekati nol karena dapat menghasilkan nilai MAPE yang sangat besar atau tidak terdefinisi [23][24].

Perlu dicatat bahwa MAPE bersifat asimetris dalam penalti kesalahan: nilai yang under-estimate (prediksi lebih rendah dari aktual) dan over-estimate (prediksi lebih tinggi dari aktual) dengan besaran yang sama akan menghasilkan nilai MAPE yang berbeda. Hal ini perlu dipertimbangkan dalam interpretasi hasil, terutama pada data harga pangan yang memiliki tren kenaikan jangka panjang. Dalam praktik penelitian prediksi harga, MAPE

sering digunakan bersama MAE untuk memberikan gambaran evaluasi yang lebih komprehensif dan saling melengkapi [23].

