BAB II
KAJIAN LITERATUR

2.1 Prediksi penjualan

Prediksi penjualan (*sales forecasting*) merupakan proses memperkirakan nilai atau volume penjualan pada periode mendatang dengan memanfaatkan pola historis dan informasi relevan untuk mendukung keputusan bisnis yang lebih terukur. Di industri, prediksi berfungsi sebagai dasar penetapan target, perencanaan operasional, dan fluktuasi permintaan yang tidak menentu dalam sistem pengendalian kinerja berimplikasi langsung pada peningkatan biaya operasional dan penurunan kualitas layanan. Ketika estimasi permintaan meleset, perusahaan berisiko mengalami *overstock* atau *stockout*, sehingga akurasi prediksi berdampak pada efisiensi dan profitabilitas [7,13].

Secara teknis, prediksi penjualan dapat dipandang sebagai pemodelan deret waktu untuk menangkap tren, musiman, dan fluktuasi acak, atau sebagai masalah regresi ketika variabel pendukung dimasukkan untuk mempelajari hubungan yang lebih kompleks. Akurasi prediksi sangat dipengaruhi oleh kualitas data, karena *missing value,* inkonsistensi pencatatan, dan *outlier* dapat mengaburkan pola yang seharusnya dipelajari model. Oleh sebab itu, performa model perlu dilakukan validasi melalui *backtesting* dan metrik evaluasi seperti *MAE, RMSE,* atau R^2 agar model tidak *overfitting* , tetapi juga stabil ketika digunakan pada data baru [14]. Dalam implementasinya, pemilihan metode prediksi penjualan umumnya bersifat komparatif karena karakter data yang musiman dan non-linear, sehingga model statistik, *machine learning,* dan *deep learning* sering dibandingkan atau dikombinasikan untuk mencapai keseimbangan antara akurasi dan *robustness* [4,14].

2.1.1 Prediksi Penjualan pada Ritel

Dalam sektor ritel, prediksi penjualan berperan sebagai landasan pengambilan keputusan strategis operasional mulai dari perencanaan persediaan, penataan produk, hingga sinkronisasi pasokan pada tingkat toko maupun item. Ritel sebagai mata rantai akhir distribusi menghimpun beragam produk untuk memenuhi kebutuhan konsumen, sehingga perubahan permintaan di sisi konsumen cenderung cepat tercermin pada dinamika penjualan [15,16] Konsekuensinya, ketidakakuratan prediksi sering berdampak pada masalah operasional, mulai dari kehilangan penjualan ketika stok tidak tersedia hingga pemborosan biaya ketika terjadi kelebihan persediaan. Dinamika ini semakin menonjol pada periode

pascapandemi karena kebijakan dan respons selama COVID-19 mendorong pemiliki ritel memperluas kanal penjualan ke online serta memanfaatkan analitik untuk memperbaiki prediksi penjualan dan menanggapi ketegangan rantai pasok [16]. Sejalan dengan itu, UNCTAD melaporkan bahwa peningkatan aktivitas belanja online berlanjut hingga 2021, rata-rata proporsi pengguna internet yang berbelanja secara daring naik dari 53% pada tahun 2019 menjadi 60% pada tahun 2020 hingga 2021, dan nilai penjualan ritel daring pada sejumlah negara besar meningkat sekitar $2 triliun pada tahun 2019 menjadi sekitar $2,9 triliun pada tahun 2021 [3]. Perubahan kanal dan perilaku belanja tersebut mengindikasikan bahwa permintaan ritel semakin dipengaruhi faktor digital, sehingga kebutuhan prediksi yang adaptif menjadi semakin penting.

Secara teknis, data penjualan ritel umumnya memuat tren dan musiman, namun juga menunjukkan pola non-linier akibat interaksi berbagai pendorong penjualan. Studi terdahulu menegaskan bahwa model berbasis jaringan saraf memiliki keunggulan dalam mengidentifikasi tren non-linier yang dinamis, pola musiman, serta keterkaitan di antara pola-pola tersebut, sekaligus menekankan pentingnya memasukkan faktor kontekstual seperti harga, promosi, *customer footfall,* hingga cuaca sebagai informasi pendukung, karena variabel-variabel tersebut membantu model merepresentasikan kondisi permintaan secara lebih realistis dan meningkatkan akurasi prediksi [17]. Dengan demikian, prediksi penjualan ritel umumnya lebih kuat apabila diposisikan sebagai masalah multivariat, yakni dengan menggabungkan histori penjualan dan variabel pemengaruh, tidak hanya mengandalkan deret historis semata, terutama ketika struktur permintaan turut dipengaruhi oleh pergeseran kanal serta perilaku belanja pascapandemi.

2.1.2 Jenis-Jenis Prediksi Penjualan

Jenis prediksi penjualan dapat diklasifikasikan berdasarkan horizon peramalan, resolusi temporal data, tingkat agregasi, jumlah variabel masukan, serta pendekatan pemodelan. Klasifikasi ini penting karena jenis prediksi yang digunakan akan menentukan aspek yang harus ditangkap oleh model, seperti tren jangka panjang, pola musiman, atau lonjakan permintaan, bagaimana data dipersiapkan, serta bagaimana hasil dievaluasi agar selaras dengan kebutuhan pengambilan keputusan. Studi terdahulu menekankan bahwa prediksi penjualan pada umumnya memanfaatkan data historis dan dievaluasi menggunakan metrik kesalahan seperti *MAE* dan *RMSE*, serta ukuran kecocokan seperti R^2, guna memastikan perbandingan antar-model bersifat objektif dan dapat dipertanggungjawabkan [8,14]. Adapun jenis-jenis prediksi penjualan tersebut adalah sebagai berikut [5,12,17,18]:

1. Berdasarkan Horizon Peramalan

Prediksi jangka pendek berorientasi pada keputusan operasional yang memerlukan respons cepat terhadap perubahan permintaan sedangkan prediksi jangka menengah dan panjang lebih difokuskan pada perencanaan kapasitas target penjualan serta mitigasi risiko seiring meningkatnya ketidakpastian pada horizon waktu yang lebih jauh. Pada horizon yang panjang akurasi menurun karena keterbatasan informasi historis sehingga model yang hanya mengandalkan data masa lalu lebih rentan terhadap error sementara model dengan prediktor eksogen cenderung lebih stabil. Dalam perbandingan pendekatan *XGBoost* kurang efektif untuk peramalan deret waktu jangka panjang karena tidak memiliki memori sekuensial dan memerlukan rekayasa fitur yang memadai sedangkan *LSTM* lebih unggul dalam menangkap dependensi jangka pendek maupun panjang secara alami. Untuk mengatasi tantangan peramalan pada horizon panjang dengan pola kompleks, pendekatan *hybrid* yang dibangun secara terstruktur melalui diagnosis dan koreksi residu lebih unggul dibandingkan penggabungan model secara *arbitrer* karena mampu mengidentifikasi dan menangkap pola-pola yang sebelumnya terlewatkan oleh model utama.

2. Berdasarkan Resolusi Temporal Data

Resolusi temporal membedakan prediksi pada skala jam, harian, mingguan, bulanan, hingga tahunan yang memengaruhi bentuk pola dominan dalam data serta jenis fitur yang relevan. Pada skala harian, variasi jangka pendek sering muncul sehingga fitur kalender seperti hari dalam minggu, bulan, kuartal, dan indikator hari libur menjadi penting, sedangkan pada skala bulanan, musiman tahunan dan tren jangka panjang lebih menonjol sehingga pemodelan musiman menjadi lebih eksplisit. Dalam praktik analitik, data transaksi dibentuk menjadi deret waktu kontinu melalui standarisasi tanggal, agregasi nilai penjualan sesuai resolusi target, serta penanganan nilai hilang untuk menjaga konsistensi runtun waktu sebelum pemodelan. Ketika organisasi membutuhkan prediksi pada beberapa resolusi sekaligus, tantangan utamanya adalah koherensi lintas resolusi agar prediksi yang lebih rinci tetap selaras dengan prediksi yang lebih agregat, dan kerangka temporal hierarkis yang menekankan rekonsiliasi prediksi untuk menjaga konsistensi tersebut sehingga keputusan operasional dan strategis tidak saling bertentangan.

3. Berdasarkan Tingkat Agregasi

Tingkat agregasi membedakan prediksi pada level yang sangat rinci seperti *Stock Keeping Unit* (*SKU*), level kategori, level toko, hingga total penjualan, di mana setiap

level memiliki karakteristik *noise* dan stabilitas yang berbeda. Prediksi pada level rinci cenderung menunjukkan variabilitas yang lebih tinggi karena dipengaruhi oleh variasi lokal, seperti perubahan promosi pada item tertentu, heterogenitas perilaku antar toko, serta fluktuasi acak dalam permintaan, sehingga pemodelan di level ini memerlukan strategi yang lebih ketat agar model tidak sekadar mengikuti *noise.* Sebaliknya, agregasi cenderung menghaluskan pola sehingga dapat meningkatkan kemudahan prediksi, meskipun berisiko menghilangkan informasi penting yang bersifat spesifik pada item atau kategori tertentu yang diperlukan untuk pengambilan keputusan pengelolaan stok di tingkat operasional. Studi terdahulu menunjukkan bahwa akurasi prediksi pada berbagai tingkat agregasi tidak konsisten antar model dan skenario, sehingga pemilihan metode dan fitur perlu mempertimbangkan level agregasi yang ditargetkan tanpa mengasumsikan keunggulan suatu model pada semua level agregasi.

4. Berdasarkan Jumlah Variabel Masukkan

   Prediksi univariat mengandalkan histori penjualan sebagai sumber informasi utama, sedangkan prediksi multivariat mengintegrasikan histori penjualan dengan variabel pemengaruh sehingga model mampu menangkap perubahan permintaan yang tidak sepenuhnya tercermin dalam pola historis. Pada sektor ritel, variabel pemengaruh tersebut dapat mencakup faktor lingkungan seperti curah hujan dan temperatur, faktor demografis, serta kejadian tertentu seperti hari libur, hari gajian, dan peristiwa sosial. Integrasi variabel-variabel ini menjadi bagian dari *pipeline* peramalan guna meningkatkan kemampuan model dalam merepresentasikan kondisi permintaan secara lebih realistis. Pada level *SKU*, perluasan variabel masukan juga kerap mencakup informasi promosi intra kategori dan antar kategori. Namun, konsekuensinya adalah meningkatnya dimensi kandidat variabel yang dapat mencapai skala sangat besar, sehingga berisiko menimbulkan overfitting dan menyulitkan estimasi model. Untuk menjaga keseimbangan antara kelengkapan informasi dan stabilitas estimasi, pendekatan seleksi variabel bertahap berbasis *lasso* diterapkan. Pendekatan ini memungkinkan pemilihan prediktor dilakukan secara otomatis dan lebih terkontrol ketika sumber informasi diperluas.

5. Berdasarkan Pendekatan Pemodelan

   Pendekatan pemodelan dalam prediksi penjualan umumnya mencakup statistik deret waktu, *machine learning*, *deep learning*, serta *hybrid*, yang masing-masing didasari asumsi dan mekanisme pembelajaran berbeda. Model statistik seperti *SARIMA* unggul pada pola linear dengan musiman stabil, namun performanya menurun ketika data

mengandung pergeseran struktural dan fluktuasi non-linier sehingga diperlukan pendekatan yang lebih adaptif. *Machine learning* seperti *XGBoost* efektif pada data terstruktur, tetapi untuk konteks deret waktu memerlukan rekayasa fitur seperti *lag* agar ketergantungan historis tetap tercermin, sementara *deep learning* seperti *LSTM* lebih mampu mempelajari dependensi sekuensial jangka panjang dan pola kompleks secara dinamis. Untuk kebutuhan yang menuntut penangkapan pola lokal dan jangka panjang sekaligus, model *hybrid* seperti *CNN-LSTM* dirancang untuk mengekstraksi pola lokal melalui mekanisme penyaringan dan memodelkan dependensi jangka panjang melalui *LSTM*, serta dapat diperkuat dengan integrasi variabel eksternal pada dataset multivariat. Studi terbaru menekankan bahwa *hybrid* yang paling kuat adalah yang dibangun melalui diagnosis dan koreksi residu, karena pendekatan ini secara eksplisit menjelaskan pola yang terlewatkan model utama dan alasan penambahan model kedua. Evaluasi tetap menjadi tahap kunci di seluruh pendekatan, dengan *MAE, RMSE*, dan R^2 sebagai dasar objektif untuk membandingkan performa dan memilih konfigurasi model yang paling dapat dipertanggungjawabkan.

2.2 Faktor-Faktor yang Mempengaruhi Penjualan Ritel

Penjualan ritel merupakan hasil dari interaksi yang kompleks antara berbagai faktor internal yang dapat dikendalikan oleh perusahaan, dan faktor eksternal yang berada di luar kendali langsung. Dalam peramalan penjualan ritel, pemahaman yang mendalam terhadap faktor-faktor ini sangat penting karena dapat memengaruhi pola permintaan dan menjadi elemen utama dalam pengembangan model prediksi yang akurat. Berdasarkan kajian literatur, beberapa faktor utama yang paling berpengaruh antara lain adalah kondisi cuaca, promosi, hari besar, preferensi konsumen, serta indikator ekonomi makro [19,20]. Faktor cuaca sendiri dapat mencakup beberapa indikator, seperti suhu udara, curah hujan, kelembapan, kecepatan angin, dan kondisi cuaca ekstrem. Pada sektor ritel global seperti *Walmart*, sebagian faktor eksternal tersebut dapat direpresentasikan melalui variabel yang tersedia dalam dataset, seperti *Temperature, IsHoliday*, dan *MarkDown1-5*, sehingga dapat digunakan sebagai masukan dalam model prediksi penjualan berbasis machine learning [21]. Sementara itu, pada konteks domestik di Indonesia, studi empiris menunjukkan bahwa faktor musiman, promosi digital, dan perilaku konsumen lokal memainkan peranan signifikan dalam mempengaruhi penjualan [21,22]. Dengan demikian, penggabungan antara faktor-faktor internal dan eksternal secara sistematis dalam model peramalan menjadi landasan penting bagi pengambilan keputusan strategis di sektor ritel.

2.2.1 Pengaruh Cuaca Terhadap Penjualan

Cuaca merupakan kondisi atmosfer pada suatu wilayah dan waktu tertentu yang dibentuk oleh beberapa unsur, seperti suhu udara, kelembapan udara, curah hujan, tekanan udara, serta kecepatan angin. Unsur-unsur tersebut dapat memengaruhi aktivitas manusia dan perilaku konsumsi dalam tingkat yang berbeda-beda. Namun, ketersediaan data cuaca pada setiap penelitian sering kali bergantung pada karakteristik dataset yang digunakan. Pada *Walmart Sales Forecast Dataset*, faktor cuaca direpresentasikan melalui variabel *Temperature* yang mengacu pada rata-rata suhu udara wilayah tempat toko beroperasi (*average temperature in the region*). Oleh karena itu, pembahasan faktor cuaca dalam penelitian ini difokuskan pada aspek suhu udara sebagai representasi kondisi cuaca yang tersedia pada dataset dan dianalisis pengaruhnya terhadap penjualan ritel.

Keterkaitan antara cuaca dan volume transaksi toko fisik dijelaskan melalui teori perilaku konsumen, di mana tingkat kenyamanan termal memengaruhi keputusan kunjungan serta pilihan produk pembeli [23]. Perubahan suhu udara terbukti mendorong pergeseran permintaan pada kelompok barang musiman, seperti pakaian khusus dan produk minuman. Berdasarkan pengujian empiris pada data transaksi Walmart di Amerika Serikat, keterikatan positif ditemukan antara kenaikan temperatur harian dan peningkatan volume penjualan kategori barang tertentu [4]. Penurunan jumlah kunjungan toko umumnya tercatat saat kondisi cuaca buruk, sedangkan peningkatan aktivitas belanja lebih sering teramati pada kondisi cuaca yang ramah.

Beberapa studi menggabungkan variabel cuaca sebagai fitur tambahan dalam model prediksi penjualan berbasis *machine learning.* Hasil kajian menunjukkan bahwa dengan menyertakan data cuaca, akurasi model prediksi meningkat secara signifikan, terutama dalam rentang waktu jangka pendek [24]. Di Indonesia, penelitian lokal juga mulai mengaitkan data cuaca harian dengan penjualan produk makanan dan minuman, di mana ditemukan adanya asosiasi kuat antara kondisi hujan dan peningkatan pembelian kopi di kedai ritel [22]. Dengan demikian, data cuaca dapat menjadi informasi pendukung dalam prediksi penjualan, tetapi penggunaannya perlu disesuaikan dengan jenis variabel cuaca yang tersedia pada dataset.

2.2.2 Pengaruh Hari Besar dan Hari Libur

Hari besar keagamaan, nasional, maupun event komersial secara konsisten menjadi pemicu lonjakan penjualan ritel. Pada pembahasan ini, hari besar dipahami sebagai momentum perayaan atau peringatan tertentu yang mendorong peningkatan konsumsi,

sedangkan hari libur dipahami sebagai periode libur yang memengaruhi intensitas kunjungan, waktu pembelian, dan akumulasi penjualan. Penelitian terdahulu menunjukkan bahwa periode menjelang hari besar seperti Natal, Tahun Baru, Ramadan, dan Idulfitri seringkali disertai dengan peningkatan signifikan dalam pembelian produk kebutuhan pokok, pakaian, dan hadiah [20]. Pola ini berlaku secara global, termasuk di Amerika dan Indonesia, dimana konsumen cenderung melakukan pembelian dalam jumlah besar sebagai bentuk persiapan atau perayaan. Studi berbasis data penjualan *Walmart* menunjukkan adanya lonjakan volume transaksi selama pekan-pekan menjelang liburan nasional [21] .

Dalam pengembangan model prediksi penjualan, pengaruh hari besar dan hari libur perlu diperhitungkan sebagai bagian dari pola musiman yang memengaruhi permintaan. Pengaruh tersebut dapat direpresentasikan melalui variabel khusus, seperti indikator biner untuk hari besar, penanda periode libur, atau parameter waktu yang menunjukkan kedekatan suatu minggu terhadap momen perayaan dan libur tertentu. Pendekatan ini menjadi relevan terutama pada data penjualan mingguan, karena dampak hari besar dan hari libur umumnya tidak hanya muncul pada satu hari tertentu, tetapi tercermin dalam akumulasi penjualan selama satu minggu. Penelitian di Indonesia juga menemukan pola pembelian unik selama Ramadan, di mana konsumen cenderung membeli produk makanan ringan dan sembako secara bersamaan [25]. Dengan mempertimbangkan pola tersebut, strategi logistik, pengelolaan persediaan, dan perencanaan promosi dapat disusun secara lebih antisipatif untuk mengurangi risiko kekosongan stok dan merespons lonjakan permintaan secara lebih tepat.

2.2.3 Pengaruh Aktivitas Promosi terhadap Penjualan

Promosi merupakan salah satu strategi utama yang digunakan oleh pelaku ritel untuk meningkatkan penjualan dalam jangka pendek maupun panjang. Studi sebelumnya menyimpulkan bahwa pemberian diskon, *bundling* produk, atau kampanye pemasaran digital dapat mendorong peningkatan volume transaksi yang signifikan [4]. Pada pasar global, program seperti *Black Friday* di Amerika Serikat menunjukkan bahwa promosi besar-besaran mampu menghasilkan lonjakan penjualan yang ekstrem. Penelitian dengan pendekatan *machine learning* juga menunjukkan bahwa periode promosi perlu diindentifikasi secara eksplisit dalam model prediksi penjualan untuk meningkatkan akurasi hasil [10].

Di pasar domestik, promosi yang disesuaikan dengan Kalender hari besar terbukti efektif menarik konsumen. Penelitian pada kedai makanan di Indonesia menemukan bahwa

strategi promosi digital melalui media sosial secara signifikan meningkatkan volume penjualan harian [26].Selain itu, kombinasi antara promosi dan hari besar seperti Ramadan menghasilkan dampak ganda terhadap kenaikan penjualan. Dengan memasukkan variable promosi ke dalam sistem peramalan penjualan, Perusahaan dapat merancang kampanye yang lebih tepat sasaran dan mengoptimalkan pengelolaan persediaan.

2.3 *Machine Learning* untuk Prediksi

Machine learning untuk prediksi merupakan pendekatan komputasional yang menggunakan data historis untuk mempelajari pola, hubungan, dan kecenderungan antardata sehingga model dapat digunakan untuk memperkirakan nilai pada periode mendatang. Pada *forecasting,* pendekatan ini menjadi penting karena data prediksi umumnya tidak hanya memuat pola linier sederhana, tetapi juga mengandung fluktuasi, tren, *seasonality,* serta pengaruh variabel lain yang saling berinteraksi. Kemampuan *machine learning* dalam mengenali pola secara otomatis menjadikannya salah satu pendekatan yang umum digunakan untuk mendukung proses prediksi pada berbagai bidang termasuk penjualan, permintaan, energi, dan data deret waktu lainnya [27].

Secara umum, *machine learning* untuk prediksi bekerja dengan menjadikan data historis sebagai dasar pembelajaran model, kemudian menggunakan pola yang telah dipelajari untuk menghasilkan prediksi pada data baru. Proses ini biasanya melibatkan pembersihan data, transformasi, pembentukan fitur, serta pembagian data ke dalam data pelatihan dan data pengujian agar model tidak sekadar memahami pola dari data masa lalu, tetapi juga dievaluasi berdasarkan kemampuannya melakukan generalisasi. Dalam perkembangannya pendekatan *machine learning* untuk prediksi mencakup berbagai metode, mulai dari algoritma *machine learning* klasik hingga *deep learning* yang masing-masing memiliki karakteristik berbeda dalam menangkap pola data dan menghasilkan prediksi [24].

2.3.1 *Machine Learning*

Machine Learning merupakan disiplin dalam kecerdasan buatan yang memungkinkan sistem komputer mempelajari pola dari data berukuran besar dan kompleks tanpa harus diprogram secara eksplisit untuk setiap aturan keputusan. Pendekatan ini membangun model berbasis data melalui perangkat matematis dan statistika sehingga sistem dapat menjalankan tugas intelektual secara lebih mandiri, terutama ketika hubungan antar-variabel tidak mudah dirumuskan sebagai aturan tetap. Fokus utama *Machine Learning* adalah menurunkan

representasi atau fungsi pemetaan dari data sehingga model mampu melakukan prediksi atau pengambilan keputusan pada data baru. Dalam praktiknya, *Machine Learning* juga berkaitan erat dengan ketidakpastian karena model bekerja pada data yang memiliki variasi, *noise*, dan dinamika yang tidak sepenuhnya deterministik. Oleh karena itu, kualitas hasil *Machine Learning* sangat dipengaruhi oleh ketepatan penyusunan data pelatihan, kesesuaian tujuan pemodelan, serta proses evaluasi untuk memastikan kemampuan generalisasi. Secara umum, kerangka *Machine Learning* dipahami melalui cara model belajar dari data, yang lazim dikelompokkan menjadi beberapa paradigma pembelajaran utama [28].

Secara konseptual, paradigma pembelajaran *Machine Learning* dibedakan berdasarkan ketersediaan label, bentuk umpan balik, dan cara sistem memperbaiki perilakunya selama proses belajar. Klasifikasi ini berfungsi untuk memberikan pemahaman yang lebih sistematis mengenai karakteristik masing-masing jenis *Machine Learning*, sebagaimana diuraikan berikut [28,29]:

1. *Supervised Learning*

   Supervised Learning merupakan pembelajaran *Machine Learning* yang mempelajari fungsi pemetaan dari input ke output menggunakan data latih berlabel yang sudah diketahui nilai targetnya. Melalui data berlabel, model diarahkan untuk mencapai tujuan tertentu yang didefinisikan sejak awal, misalnya meminimalkan kesalahan prediksi antara keluaran model dan label aktual. Karena targetnya tersedia, proses pelatihan berfokus pada optimasi parameter model agar prediksi pada data baru tetap akurat dan konsisten. Konsekuensinya, *supervised learning* menuntut ketersediaan label yang memadai serta representatif agar model tidak hanya mengingat data latih, tetapi benar-benar menangkap pola yang relevan.

2. *Unsupervised Learning*

   Unsupervised Learning merupakan pembelajaran *Machine Learning* yang mempelajari data tanpa label dengan tujuan utamanya menemukan struktur, keterkaitan, atau pola tersembunyi yang muncul secara alami dari sebaran data. Pendekatan ini menekankan analisis berbasis data (*data-driven*) karena tidak ada target eksplisit yang menjadi acuan langsung dalam pelatihan. *Unsupervised learning* sering dikaitkan dengan pembentukan kelompok (*clustering*) atau identifikasi representasi yang lebih ringkas untuk membantu memahami karakteristik data. Karena tidak memiliki jawaban benar berupa label, keberhasilan karakteristik *unsupervised learning* diukur melalui kualitas struktur yang dihasilkan dan kegunaannya dalam interpretasi atau tahap analisis lanjutan.

3. *Reinforcement Learning*

Reinforcement Learning merupakan pembelajaran *Machine Learning* yang melakukan pengambilan keputusan berurutan, ketika agen berinteraksi dengan lingkungan dan menerima umpan balik berupa *reward* untuk memperbaiki perilakunya. Sumber informasi utama bagi agen adalah *reward function,* kemudian agen membangun dasar pengambilan keputusan melalui pendekatan berbasis nilai atau langsung mempelajari kebijakan tindakan. Inti ini dari *Reinforcement Learning* adalah belajar kebijakan yang makin efektif dari waktu ke waktu berdasarkan pengalaman interaksi, bukan dari pasangan input-output berlabel seperti pada *supervised learning.* Dengan demikian, *Reinforcement Learning* menekankan *trade-off* antara eksplorasi tindakan baru dan eksploitasi pengetahuan yang sudah dipelajari agar agen mencapai performa yang optimal pada tujuan yang ditetapkan.

Pada akhirnya, pemilihan jenis *Machine Learning* ditentukan oleh ketersediaan label, tujuan analisis, dan bentuk keputusan yang ingin dihasilkan. Kualitas hasil tidak hanya ditentukan oleh algoritma, tetapi juga oleh kesiapan data dan ketepatan proses evaluasi agar model tidak bias ketika melakukan pelatihan dan pengujian. Studi terdahulu menunjukkan bahwa penggunaan parameter bawaan dapat menghasilkan performa yang kurang optimal dan berpotensi memicu *overfitting*, sehingga pengaturan parameter menjadi langkah yang wajar dalam praktik *Machine Learning*. Dengan demikian, pembahasan *Machine Learning* yang komprehensif perlu menempatkan paradigma pembelajaran, proses evaluasi, serta pengendalian kompleksitas model sebagai satu kesatuan metodologis [11].

2.3.2 *Deep Learning*

Deep Learning merupakan bagian dari machine learning yang berkembang dari jaringan saraf tiruan menuju struktur berlapis (*multi-layer*) sehingga mampu membentuk representasi data secara bertingkat. Studi terdahulu menjelaskan bahwa *Deep Learning* tersusun dari banyak lapisan jaringan saraf dan menyediakan abstraksi tingkat tinggi untuk pemodelan data, sehingga proses belajar tidak hanya bergantung pada fitur yang dirancang manual, tetapi juga pada representasi yang dipelajari model dari data. Konsep berlapis ini membuat setiap lapisan dapat mengekstraksi pola yang semakin kompleks dari keluaran lapisan sebelumnya. Dengan kapasitas tersebut, *Deep Learning* dikenal efektif untuk memodelkan hubungan data yang kompleks, namun umumnya menuntut sumber daya

komputasi yang lebih besar karena jumlah parameter dan proses pelatihannya lebih intensif dibanding pendekatan yang lebih sederhana [28].

Secara umum, *Deep Learning* dapat dipahami melalui beragam arsitektur jaringan saraf yang memiliki pendekatan berbeda dalam mengekstraksi fitur serta memodelkan ketergantungan informasi. Klasifikasi arsitektur ini berfungsi untuk menjelaskan alasan mengapa suatu model lebih sesuai dengan karakteristik pola tertentu dibandingkan arsitektur lainnya, sebagaimana diuraikan berikut ini[28,30,31,32]:

1. *Convolutional Neural Network* (*CNN*)

   Convolutional Neural Network merupakan arsitektur *Deep Learning* yang menekankan ekstrasi fitur otomatis melalui pemrosesan berlapis. Studi terdahulu menegaskan bahwa *Convolutional Neural Network* meminimalkan fungsi *loss* dengan menyebarkan gradien kesalahan ke belakang (*backpropagation*) untuk memperbaiki bobot jaringan secara adaptif. *Convolutional Neural Network* juga memanfaatkan mekanisme filter dan pemindaian pola lokal sehingga efektif menangkap korelasi lokal dan pola yang relatif stabil terhadap perubahan skala. Karakter ini membuat *Convolutional Neural Network* sering dipahami sebagai komponen ekstrasi fitur tingkat tinggi yang membantu model memperoleh representasi yang lebih informatif dari data mentah.

2. *Recurrent Neural Network* (*RNN*)

   Recurrent Neural Network merupakan arsitektur *Deep Learning* yang dirancang untuk data sekuensial dengan cara membawa informasi dari waktu sebelumnya melalui mekanisme rekuren. Keunggulan utamanya terletak pada kemampuan menangkap dependensi temporal, karena keluaran pada satu langkah waktu dipengaruhi oleh keadaan (*state*) dari langkah sebelumnya. Namun, studi terdahulu juga mencatat bahwa *Recurrent Neural Network* memiliki keterbatasan pada dependensi jangka panjang, terutama karena isu stabilitas gradien saat pelatihan, sehingga performanya dapat menurun ketika ketergantungan informasi antar waktu semakin panjang. Oleh karena itu, *Recurrent Neural Network* sering diposisikan sebagai fondasi awal untuk model sekuensial sebelum dikembangkan menjadi varian yang lebih mampu menangani dependensi panjang.

3. *Long Short-Term Memory* (*LSTM*)

   Long Short-Term Memory merupakan arsitektur *Deep Learning* yang dipandang sebagai varian *Recurrent Neural Network* yang dirancang khusus untuk memperkuat kemampuan memori pada data sekuensial. Studi terdahulu menegaskan bahwa *Long*

Short-Term Memory memiliki sistem memori bawaan sehingga mampu mempertahankan informasi penting dari langkah waktu sebelumnya, yang menjadi pembeda utama dibanding *Recurrent Neural Network* standar. Dari sisi arsitektur, *Long Short-Term Memory* memperkenalkan *memory cell* dan mekanisme gerbang (*input, forget, output*) yang mengatur informasi mana yang disimpan, diperbarui, atau dibuang. Mekanisme ini juga dikaitkan dengan kemampuan *Long Short-Term Memory* untuk mengurangi kendala *vanishing gradient* dan menjaga stabilitas pembelajaran ketika dependensi jangka panjang perlu dipelajari.

4. *Bidirectional LSTM* (*BiLSTM*)

   Bidirectional LSTM merupakan arsitektur *Deep Learning* hasil dari pengembangan *Long Short-Term Memory* yang memproses sekuens dari dua arah, yaitu maju dan mundur, sehingga konteks dapat ditangkap dari kedua sisi. Studi terdahulu menyatakan bahwa arsitektur ini menggabungkan lapisan *Long Short-Term Memory forward* dan *backward*, lalu mengombinasikan keluarannya untuk menghasilkan representasi yang lebih kaya secara kontekstual. Konsep dua arah ini membantu mengurangi ketergantungan pada satu titik waktu saja karena informasi masa lalu dan masa depan dalam sekuens dapat digunakan bersamaan. Dengan mekanisme tersebut, *Bidirectional LSTM* sering dijelaskan sebagai pendekatan yang lebih efektif ketika konteks sekuens perlu dipahami secara menyeluruh.

Secara ringkas, *Deep Learning* menempatkan arsitektur berlapis sebagai mekanisme utama untuk membentuk representasi data yang semakin abstrak. *Recurrent Neural Network* menekankan pemodelan sekuens melalui keterkaitan antar waktu, tetapi dapat terbatas pada dependensi panjang. *Long Short-Term Memory* memperkuat aspek memori melalui *memory cell* dan mekanisme gerbang, lalu variannya *Bidirectional LSTM* memperluas kapasitas representasi serta konteks sekuens. Oleh karena itu, pemilihan arsitektur *Deep Learning* sebaiknya didasarkan pada karakter pola yang ingin ditangkap dan kebutuhan konteks dalam data.

2.3.3 Algoritma *Machine Learning* untuk Prediksi

Algoritma *machine learning* yang digunakan untuk prediksi memiliki karakteristik yang berbeda dalam menangani bentuk data, kompleksitas data, kebutuhan interpretasi, sensitivitas terhadapp *hyperparameter,* dan efisiensi komputasi. Oleh karena itu, keandalan suatu algoritma sangat bergantung pada konteks data dan tujuan prediksi, sehingga tidak ada

pendekatan yang selalu paling unggul dalam segala kondisi. Sehingga pemilihan algoritma perlu mempertimbangkan karakteristik data karena Sebagian algoritma lebih efektif pada data tabular hasil *feature engineering,* sementara algoritma lainnya lebih adaptif terhadap pola tren, musiman, atau hubungan non-linier yang dipengaruhi oleh faktor eksternal [24]. Berdasarkan keragaman karakteristik tersebut, berikut akan diuraikan berbagai jenis algoritma *machine learning* yang umum digunakan untuk keperluan prediksi [33-35]:

1. *Random Forest*

   Random forest merupakan algoritma *ensemble* berbasis pohon keputusan yang bekerja dengan membangun banyak pohon dari subset data yang berbeda, kemudian menggabungkan keluaran seluruh pohon untuk menghasilkan satu prediksi akhir. Pada pembahasan regresi, nilai prediksi akhir diperoleh melalui mekanisme rata-rata dari seluruh prediksi pohon yang telah dibangun. Pendekatan *ensemble,* memberikan *random forest* tingkat stabilitas yang lebih tinggi dibandingkan dengan *decision tree* tunggal, karena keputusan akhir tidak lagi bergantung pada struktur pohon tertentu, melainkan merupakan hasil agregasi dari berbagai pohon yang dilatih pada sample data yang berbeda. Keunggulan tersebut menjadikan *random forest* banyak digunakan dalam pemodelan prediktif, terutama pada data tabular, mengingat algoritma ini mampu menangkap hubungan non-linier, mampu mengakomodasi interaksi antarfitur, serta relative tahan terhadap keberadaan *noise* dalam data.

2. *Decision Tree*

   Decision Tree merupakan algoritma yang membentuk aturan keputusan secara bertingkat melalui struktur pohon. Setiap simpul pada pohon mewakili proses pemisahan data berdasarkan fitur tertentu, sedangkan cabang-cabangnya menunjukkan hasil keputusan hingga mencapai node terminal. Keunggulan utama *decision tree* terletak pada interpretabilitasnya, karena proses prediksi dapat dijelaskan melalui jalur keputusan yang jelas. Namun, performa model sangat dipengaruhi oleh hyperparameter seperti *max_depth, min_samples_split, min_samples_leaf*, dan *criterion*. Jika pohon tumbuh terlalu dalam, model cenderung mengalami *overfitting*, sedangkan jika terlalu dangkal, model berisiko gagal menangkap pola penting pada data. Oleh karena itu, pengendalian kompleksitas dan proses *tuning* menjadi aspek penting dalam penggunaan *decision tree* untuk prediksi.

3. *Support Vector Regression* (*SVR*)

   Support Vector Regression merupakan pengembangan dari *Support Vector Machine* yang dirancang khusus untuk menangani tugas regresi, yaitu memprediksi variabel

kontinu. Model ini membangun suatu fungsi regresi dengan prinsip utama menjaga kesalahan prediksi tetap berada dalam batas toleransi tertentu, sembari mempertahankan kemampuan generalisasi yang baik terhadap data baru. Keunggulan utama *SVR* terletak pada fleksibilitasnya dalam memodelkan hubungan non-linier melalui mekanisme fungsi kernel, sehingga algoritma ini menjadi relevan ketika pola data tidak dapat dijelaskan secara memadai oleh pendekatan linier sederhana. Performa *SVR* sangat ditentukan oleh pemilihan jenis kernel serta optimalisasi *hyperparameter*, antara lain parameter *C*, *epsilon*, dan parameter spesifik dari kernel yang digunakan. Oleh karena itu, proses *hyperparameter tuning* menjadi tahapan yang krusial guna memperoleh model dengan tingkat akurasi dan stabilitas yang tinggi. Berbagai studi terbaru menunjukkan bahwa integrasi *SVR* dengan algoritma optimasi, seperti *Particle Swarm Optimization* (*PSO*), dapat meningkatkan kapabilitas model dalam menangkap pola data yang kompleks secara lebih efektif.

4. *eXtreme Gradient Boosting* (*XGBoost*)

   XGBoost merupakan algoritma berbasis pohon keputusan dengan pendekatan *boosting* yang dikembangkan untuk menghasilkan prediksi berakurasi tinggi serta efisiensi komputasi yang baik. Secara konseptual, algoritma ini membangun model secara sekuensial dengan setiap pohon baru yang dirancang untuk mengoreksi residual atau kesalahan dari model sebelumnya. Keunggulan tersebut menjadikan *XGBoost* sebagai salah satu algoritma yang umum diimplementasikan pada data tabular, mengingat kemampuannya dalam menangkap hubungan non-linier serta mengakomodasi sejumlah besar fitur hasil rekayasa data. Meskipun demikian, *XGBoost* memiliki struktur *hyperparameter* yang kompleks meliputi, *n_estimator*, *max_depth*, *min_child_weight*, *gamma*, *subsample*, *colsample_bytree*, *alpha*, *lambda*, serta *learning_rate.* Oleh karena itu, proses *hyperparameter tuning* menjadi langkah yang krusial guna memastikan model mencapai performa yang optimal sesuai dengan karakteristik data yang digunakan.

5. *Prophet*

   Prophet merupakan model peramalan deret waktu yang memodelkan data sebagai gabungan komponen tren, musiman, efek hari libur atau kejadian khusu, serta komponen residual. Salah satu keunggulan utama model ini terletak pada kemampuannya dalam mengakomodasi perubahan tren melalui mekanisme *changepoints,* sehingga tetap adaptif terhadap dinamika pola data dari waktu ke waktu. selaras dengan hal tersebut, *prophet* merepresentasikan komponen musiman secara

eksplisit, sehingga pola periodic seperti mingguan, bulanan, atau tahunan dapat dipelajari dan diinterpretasikan dengan lebih jelas. Pendekatan ini juga mendukung penambahan variabel eksternal atau *regressor* guna memperkaya informasi prediktif, meskipun penerapannya memerlukan ketersediaan nilai variabel pada periode historis maupun horizon peramalan. Karakteristik tersebut menjadikan *prophet* sangat relevan untuk data yang memiliki pola temporal berbasis waktu yang kuat, tren yang berubah secara bertahap, sserta kebutuhan interpretasi model yang tinggi. Meskipun demikian, performa *prophet* berpotensi kurang optimal apabila data lebih didominasi oleh pola acak atau hubungan nonlinier kompleks yang sulit didekomposisi ke dalam komponen temporal utama

2.4 Model *Hybrid Machine Learning*

Model *hybrid machine learning* merupakan suatu pendekatan pemodelan yang menggabungkan dua atau lebih metode dalam satu kerangka analisis untuk meningkatkan kinerja prediksi. Pendekatan ini muncul dari pemahaman bahwa setiap model memiliki karakteristik dan kemampuan yang berbeda dalam mempelajari pola. Setiap model memiliki kapasitas yang berbeda dalam memahami data, dengan sebagain model mampu menangkap struktur linier dan hubungan yang stabil, sedangkan model lainnya lebih unggul dalam mengenali pola non-linier, interaksi antarfitur yang kompleks, serta dependensi yang bersifat berurutan [32, 36].Sehingga, *hybrid machine learning* tidak sekadar penggabungan beberapa algoritma, melainkan suatu strategi untuk mengintegrasikan kelebihan masing-masing pendekatan guna meminimalkan keterbatasan model tunggal serta memperluas cakupan representasi pola data [29,31].Dalam perkembangannya, pendekatan *hybrid* diposisikan sebagai bentuk intergasi yang fleksibel, mengingat kemampuannya dalam mengombinasikan model *machine learning, deep learning,* maupun model statistik sesuai dengan karakteristik struktur data dan tujuan analisis [30]. Dengan karakteristik tersebut, *hybrid machine learning* menjadi pendekatan yang signifikan dalam pengembangan sistem cerdas *modern* karena mampu menyeimbangkan kapasitas pembelajaran, keluasan representasi pola, serta potensi peningkatan kinerja model.

2.4.1 *Hybrid* Model

Hybrid model dipahami sebagai rancangan pemodelan yang menempatkan beberapa komponen model secara saling melengkapi dalam satu sistem prediksi. Pembentukan

struktur tersebut dilatarbelakangi oleh keterbatasan model tunggal dalam mengakomodasi keseluruhan karakteristik data secara optimal, sehingga penggabungan model dilakukan melalui berbagai pola, seperti menjalankan beberapa model secara paralel lalu menggabungkan keluarannya, memanfaatkan keluaran model dasar sebagai masukan bagi model lanjutan, maupun menempatkan model kedua untuk mengoreksi kesalahan dari model sebelumnya [28,36].Secara metodologis, struktur ini menunjukkan bahwa hybrid model bukan sekadar kumpulan algoritma, melainkan desain pemodelan yang menyusun peran setiap model secara fungsional. Dengan demikian, masing-masing komponen dalam hybrid model memiliki kontribusi yang spesifik, baik dalam mengekstraksi pola, mengoreksi galat, meningkatkan generalisasi, maupun memperkuat hasil akhir [30]. Dalam ranah prediksi, pendekatan ini dipandang relevan karena data sering kali memuat pola tren, musiman, nonlinier, dan residual secara bersamaan, sehingga integrasi model dapat mendorong hasil yang lebih stabil dan adaptif [30,37].

2.4.2 Jenis-Jenis *Hybrid* Model

Jenis-jenis *hybrid machine learning* pada dasarnya dibedakan berdasarkan mekanisme integrasi yang digunakan untuk menggabungkan model-model penyusunnya ke dalam satu kerangka prediksi. Perbedaan antarjenis *hybrid* tidak ditentukan oleh banyaknya model yang digunakan, melainkan oleh pola keterhubungan antarmodel, pembagian peran pada setiap komponen, serta mekanisme pembentukan prediksi akhir. Berdasarkan pola yang digunakan dalam literatur, integrasi tersebut umumnya dilakukan melalui penggabungan langsung keluaran prediksi, penyusunan keluaran model dasar sebagai masukan bagi model lanjutan, atau pemodelan bertahap yang menempatkan residual sebagai target pembelajaran lanjutan setelah struktur utama data ditangkap oleh model tahap awal, sebagaimana diuraikan berikut[27,36,37,38,39,40]:

1. Integrasi Paralel

   Pada logika ini, beberapa model dibangun secara paralel sehingga setiap model menghasilkan prediksi dari sudut pandang yang berbeda. Prediksi akhir kemudian dibentuk dengan menggabungkan seluruh keluaran tersebut agar hasilnya lebih stabil dan tidak terlalu bergantung pada satu model tunggal. Literatur menjelaskan bahwa penggabungan dapat dilakukan melalui rata-rata sederhana ketika kontribusi tiap model diperlakukan setara, atau melalui rata-rata berbobot ketika kontribusi model disesuaikan berdasarkan reliabilitasnya. Sebagai bentuk penggabungan yang paling langsung, rata-rata sederhana digunakan ketika semua model dianggap memiliki

tingkat keandalan yang relatif sebanding. Dalam skema ini, prediksi akhir dihitung sebagai nilai tengah dari seluruh prediksi model sehingga setiap model memiliki pengaruh yang sama terhadap hasil akhir. Secara matematis digunakan sebagai berikut

$$\hat{y} = \frac{1}{M}\sum_{j=1}^{M} \hat{y}_j \quad \text{.....(1)}$$

$\hat{y}$: prediksi akhir hasil penggabungan.

M : jumlah model yang digabungkan.

$\hat{y}_j$: prediksi yang dihasilkan model ke-j.

Ketika terdapat perbedaan kualitas antarmodel, rata-rata berbobot memberikan cara yang lebih terarah karena kontribusi setiap model dapat diatur melalui bobot. Dengan demikian, model yang lebih andal dapat diberi porsi pengaruh lebih besar, sementara model lain tetap dipertahankan untuk menjaga diversitas prediksi. Secara matematis digunakan sebagai berikut:

$$\hat{y} = \frac{\sum_{j=1}^{m} w_j \hat{y}_j}{\sum_{j=1}^{m} w_j} \quad \text{.....(2)}$$

$\hat{y}$: prediksi akhir hasil penggabungan.

m: jumlah model yang digabungkan.

$\hat{y}_j$: prediksi model ke-j.

w_j: bobot kontribusi model ke-j. Semakin besar w_j, semakin besar pengaruh model ke-j terhadap $\hat{y}$.

2. *Stacking* dengan meta-model

Jika penggabungan rata-rata masih terlalu kaku karena memperlakukan hubungan antarprediksi secara sederhana, maka stacking menawarkan cara integrasi yang lebih adaptif. Dalam stacking, prediksi dari beberapa model dasar tidak langsung digabungkan, melainkan dijadikan masukan bagi meta-model yang mempelajari kombinasi terbaik berdasarkan data. Literatur menekankan bahwa mekanisme ini membantu menangkap keterkaitan antarprediksi dan pola gabungan yang tidak tertangkap oleh penjumlahan berbobot biasa. Dalam stacking, keluaran beberapa

model dasar diperlakukan sebagai fitur tingkat-atas yang kemudian diproses oleh fungsi penggabungan $g(\cdot)$. Fungsi ini dipelajari dari data sehingga penggabungan tidak bersifat tetap, melainkan menyesuaikan pola hubungan antarprediksi yang muncul pada proses pelatihan. Secara matematis digunakan sebagai berikut:

$$\hat{y}_t = g\left(\hat{y}_t^{(1)}, \hat{y}_t^{(2)} \dots \hat{y}_t^{(n)}\right) \quad \text{.....(3)}$$

$\hat{y}_t$: prediksi akhir pada indeks atau waktu t.

M : jumlah base learner (model dasar).

$\hat{y}_t^{(m)}$: prediksi base learner ke-m pada t.

$g(\cdot)$: meta-model yang mempelajari fungsi penggabungan prediksi base learner menjadi satu prediksi final.

3. *Decomposition-Based Residual*

Pada beberapa kondisi, peningkatan kinerja prediksi dinilai lebih efektif dicapai melalui pembagian peran model secara bertahap daripada melalui penggabungan keluaran pada satu tahap. Pada pendekatan ini, baseline model terlebih dahulu digunakan untuk menangkap struktur utama deret waktu, sedangkan residual diperlakukan sebagai komponen yang masih memuat informasi yang belum dapat dijelaskan secara penuh oleh model awal. Dalam formulasi *Prophet*, deret waktu direpresentasikan ke dalam komponen *tren, seasonality, holiday,* dan residu, sehingga residual telah ditempatkan sebagai bagian formal dari struktur model . Studi terdahulu menunjukkan bahwa komponen trend, seasonal, dan residual dapat dipisahkan serta dimodelkan secara mandiri untuk menyederhanakan proses pembelajaran. Selain itu, residual dari model utama dinyatakan masih dapat mengandung pola nonlinier yang relevan untuk dipelajari pada tahap lanjutan. Proses dekomposisi sebelum pelatihan model lanjutan juga ditunjukkan dapat menurunkan kompleksitas deret waktu sehingga pola selain *seasonality* dapat dipelajari secara lebih terarah. Langkah awal dalam pendekatan ini dilakukan dengan mendefinisikan residual sebagai selisih antara nilai aktual dan prediksi baseline model. *Residual* tersebut merepresentasikan bagian kesalahan yang belum dijelaskan oleh model tahap pertama, sehingga dijadikan target pembelajaran pada tahap berikutnya dan diformulasikan secara matematis sebagai berikut

$$r_t = y_t - \hat{y}_t^{(1)} \quad \text{.....(4)}$$

y_t : nilai aktual pada indeks atau waktu t.

$\hat{y}_t^{(1)}$: prediksi model tahap pertama (baseline) pada t.

r_t : residual, yaitu bagian kesalahan yang belum dijelaskan oleh model tahap pertama.

Setelah residual diperoleh, pemodelan lanjutan dilakukan untuk menangkap pola yang tidak berhasil direpresentasikan oleh baseline model. *Prophet* digunakan untuk memodelkan tren jangka panjang, *seasonality*, dan *holiday*, sedangkan *LSTM* digunakan untuk mempelajari dependensi nonlinier pada residual yang belum dapat dijelaskan oleh *Prophet*. Pada tahap berikutnya, *XGBoost* dapat digunakan untuk memperhalus koreksi residual agar sisa kesalahan yang masih tertinggal dapat dimodelkan secara lebih adaptif . Dengan demikian, prediksi akhir dibentuk melalui penjumlahan antara prediksi *baseline* dan estimasi residual hasil model tahap lanjutan. Secara matematis, mekanisme tersebut dinyatakan sebagai berikut:

$$\hat{y}_t = \hat{y}_t^{(1)} + \hat{r}_t \quad (5)$$

$\hat{y}_t$: prediksi akhir pada t.

$\hat{y}_t^{(1)}$: prediksi baseline pada t.

$\hat{r}_t$: prediksi residual yang dihasilkan model tahap kedua.

Setelah memahami mekanisme integrasi, model-model *hybrid* dapat diklasifikasikan berdasarkan susunan komponen serta keluarga algoritma yang dikombinasikan. Klasifikasi ini memiliki relevansi akademis karena istilah *hybrid* bersifat luas, sementara kualitas rancangan *hybrid* ditentukan oleh ketepatan pembagian peran antar-komponen dan konsistensi mekanisme integrasinya.

2.4.3 Keunggulan *Hybrid* Model

Hybrid model memiliki nilai lebih karena memungkinkan beberapa mekanisme pembelajaran bekerja secara saling melengkapi dalam satu sistem. Melalui pendekatan ini, proses pemodelan tidak hanya bertumpu pada satu cara belajar, tetapi memanfaatkan kemampuan sistem dalam merepresentasikan pola data yang kompleks. Dengan landasan tersebut, keunggulan *hybrid* model dapat diuraikan sebagai berikut[27,28,30,40]:

1. Memanfaatkan keunggulan komplementer antar-model

   Hybrid model memungkinkan beberapa model dengan karakteristik yang berbeda digunakan secara bersama, sehingga kelemahan model tunggal dapat dikurangi melalui kontribusi model lain yang lebih sesuai terhadap pola tertentu.

2. Meningkatkan potensi akurasi model

   Penggabungan beberapa pendekatan pembelajaran dalam satu kerangka pemodelan memberikan peluang yang lebih besar untuk menghasilkan prediksi yang lebih akurat dibandingkan penggunaan model tunggal. Pada integrasi residual, peningkatan akurasi juga didukung oleh proses koreksi bertahap terhadap sisa model sebelumnya.

3. Mampu menangani pola data yang kompleks

   Hybrid model lebih adaptif dalam menghadapi data yang memuat pola linear dan non-linier secara bersamaan, dependensi temporal, serta hubungan yang tidak mudah dijelaskan oleh satu pendekatan saja.

4. Meningkatkan stabilitas dan kemampuan generalisasi

   Integrasi beberapa model mengurangi ketergantungan sistem pada satu mekanisme pembelajaran, sehingga hasil yang diperoleh cenderung lebih stabil dan memiliki kemampuan generalisasi yang lebih baik terhadap variasi data.

5. Memberikan fleksibilitas dalam perancangan sistem

   Hybrid model dapat dibangun melalui berbagai pola integrasi, seperti agregasi, *stacking*, dan residual, sehingga struktur model dapat disesuaikan dengan karakter data serta tujuan analisis yang ingin dicapai.

2.4.4 Metode *Hyperparameter Tuning*

Hyperparameter tuning merupakan proses sistematis untuk menentukan kombinasi parameter eksternal model yang tidak dipelajari langsung dari data, tetapi ditetapkan sebelum proses pelatihan dimulai. *Hyperparameter* berbeda dari parameter model karena nilainya tidak diperbarui melalui optimisasi gradien, melainkan ditentukan melalui strategi pencarian tertentu untuk memperoleh kinerja terbaik. Secara konseptual, *hyperparameter* tuning bertujuan menyeimbangkan akurasi dan generalisasi model, karena konfigurasi yang tidak tepat dapat menyebabkan *underfitting* maupun *overfitting*. Studi terdahulu menegaskan bahwa pemilihan *hyperparameter* yang optimal berpengaruh signifikan terhadap peningkatan performa model prediktif, baik pada algoritma berbasis pohon maupun jaringan saraf dalam[11,33].

Secara umum, strategi *hyperparameter tuning* dapat dibedakan berdasarkan mekanisme eksplorasi ruang parameter dan cara evaluasinya. Beberapa pendekatan yang banyak digunakan dalam literatur dan praktik penelitian dinyatakan sebagai berikut [11,33]:

1. *Random Search*

   Random search merupakan metode pencarian hyperparameter dengan cara memilih kombinasi nilai parameter secara acak dari ruang pencarian yang telah ditentukan. Pendekatan ini tidak mengevaluasi seluruh kemungkinan kombinasi, melainkan mengambil sejumlah sampel acak untuk diuji kinerjanya. Keunggulan *random search* terletak pada efisiensinya ketika ruang parameter sangat luas, karena tidak semua kombinasi harus dihitung. Studi yang membandingkan *grid search* dan *random search* menunjukkan bahwa *random search* dapat memberikan performa yang kompetitif dengan waktu komputasi yang lebih efisien pada beberapa kasus. Secara metodologis, pendekatan ini efektif ketika tidak diketahui secara pasti parameter mana yang paling berpengaruh, sehingga eksplorasi acak memungkinkan penemuan konfigurasi yang tidak terduga.

2. *Grid Search*

   Grid search merupakan metode pencarian terstruktur yang mengevaluasi seluruh kombinasi hyperparameter berdasarkan kisi (*grid*) nilai yang telah ditentukan sebelumnya. Setiap kombinasi diuji dan dievaluasi menggunakan metrik kinerja tertentu, kemudian dipilih konfigurasi dengan performa terbaik. Keunggulan utama *grid search* adalah sifatnya yang sistematis dan mudah direproduksi karena seluruh ruang kombinasi yang ditentukan benar-benar dievaluasi. Namun, metode ini dapat menjadi sangat mahal secara komputasi ketika jumlah parameter dan variasinya besar. Studi terdahulu menegaskan bahwa grid *search* memberikan hasil yang konsisten karena mengevaluasi seluruh kombinasi kandidat, tetapi efisiensinya menurun seiring bertambahnya dimensi ruang parameter.

3. *Bayesian Optimization*

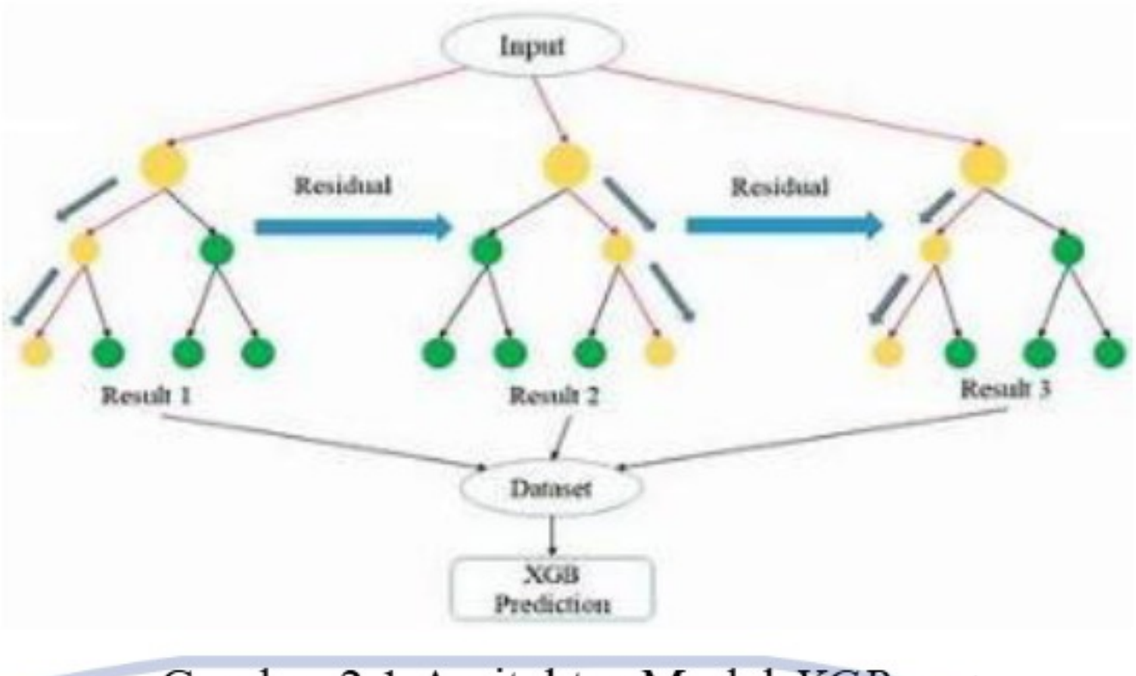
   Bayesian Optimization merupakan pendekatan yang lebih adaptif karena menggunakan model probabilistik untuk memperkirakan fungsi objektif dan menentukan kombinasi *hyperparameter* berikutnya secara strategis. Berbeda dari *grid* dan *random search* yang bersifat eksploratif langsung, *Bayesian Optimization* membangun model *surrogate* untuk memperkirakan hubungan antara konfigurasi *hyperparameter* dan performa model. Berdasarkan estimasi tersebut, algoritma memilih kombinasi yang paling menjanjikan untuk dievaluasi selanjutnya. Studi

mengenai optimisasi *hyperparameter* pada model *deep learning* menunjukkan bahwa *Bayesian Optimization* mampu meningkatkan efisiensi pencarian dengan jumlah evaluasi yang lebih sedikit dibanding pendekatan eksploratif konvensional . Dengan demikian, metode ini relevan ketika biaya pelatihan model relatif tinggi dan diperlukan strategi pencarian yang lebih terarah.

Dengan demikian, *Hyperparameter tuning* merupakan komponen krusial dalam pengembangan model pembelajaran mesin karena secara langsung memengaruhi keseimbangan antara kompleksitas dan generalisasi model. Pemilihan metode pencarian perlu disesuaikan dengan karakteristik ruang parameter, kapasitas komputasi, serta kompleksitas model. Oleh karena itu, pemahaman mendalam terhadap mekanisme *hyperparameter tuning* menjadi fondasi utama dalam memperoleh konfigurasi model yang optimal serta dapat dipertanggungjawabkan secara ilmiah.

2.5 *eXtreme Gradient Boost (XGBoost)*

eXtreme gradient boost merupakan pendekatan dengan metode *ensemble* dalam keluarga *gradient boosting decision tree* yang membangun model secara bertahap melalui penambahan pohon keputusan untuk memperbaiki kesalahan prediksi pada iterasi sebelumnya. Secara konseptual, setiap pohon baru tidak ditujukan untuk menggantikan pohon yang lama, melainkan berperan sebagai komponen koreksi sehingga model semakin mendekati target secara progresif. Arsitektur *XGBoost* dirancang untuk menangkap hubungan non-linear dan interaksi fitur yang kompleks, sekaligus menyediakan mekanisme regularisasi guna meningkatkan kompleksitas sehingga model tidak mudah *overfitting* ketika jumlah pohon meningkat [31,41].Secara matematis, tujuan ini direalisasikan dalam tiga komponen utama, yang diuraikan sebagai berikut [32,41]:



Gambar 2.1 Arsitektur Model *XGBoost*

1. Persamaan Prediksi Adaptif

   Persamaan ini menunjukkan bahwa prediksi dalam XGBoost dibangun secara aditif melalui akumulasi kontribusi dari setiap pohon keputusan. Dengan demikian, keluaran akhir model merepresentasikan penjumlahan sistematis seluruh pohon, di mana representasi matematis dari mekanisme prediksi adaptif tersebut dapat dinyatakan sebagai berikut:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \quad \text{.....(6)}$$

   Pada persamaan ini, $x_i$adalah vektor fitur sampel ke-i, $\hat{y}_i$adalah prediksi, K menyatakan jumlah pohon, dan $f_k(\cdot)$ adalah fungsi pohon ke-k yang memetakan $x_i$ke skor pada daun. Pendekatan aditif pada *XGBoost* menjelaskan bahwa setiap pohon berkontribusi terhadap prediksi akhir. Proses ini menghasilkan model yang dibangun melalui koreksi bertahap, serta konsisten sesuai dengan prinsip *boosting* yang memperbaiki kesalahan secara iteratif.

2. Fungsi Objektif

   Persamaan ini menyatakan bahwa pelatihan *XGBoost* meminimalkan dua komponen sekaligus. Komponen pertama adalah *loss* untuk menekan kesalahan prediksi. Komponen kedua adalah regularisasi untuk menahan kompleksitas model. Dengan persamaan sebagai berikut:

$$Obj = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \quad \text{.....(7)}$$

   Pada persamaan ini, n merepresentasikan jumlah sampel, $y_i$nilai aktual, $\hat{y}_i$prediksi, $L(y_i, \hat{y}_i)$adalah *loss* yang mengukur kesalahan prediksi, sedangkan $\Omega(f_k)$merupakan penalti kompleksitas pohon. Secara metodologis, bentuk ini menegaskan prinsip dasar

optimisasi model tidak hanya didorong untuk menekan error melalui $\sum L(\cdot)$, tetapi juga ditahan agar tidak menjadi terlalu kompleks melalui $\sum \Omega(\cdot)$. Dengan demikian, solusi optimal dicapai melalu keseimbangan terukur antara akurasi dan generalisasi, sebagaimana ditekankan dalam fungsi objektif yang memuat *loss term* dan *regularization term.*

3. Regularisasi Kompleksitas Pohon

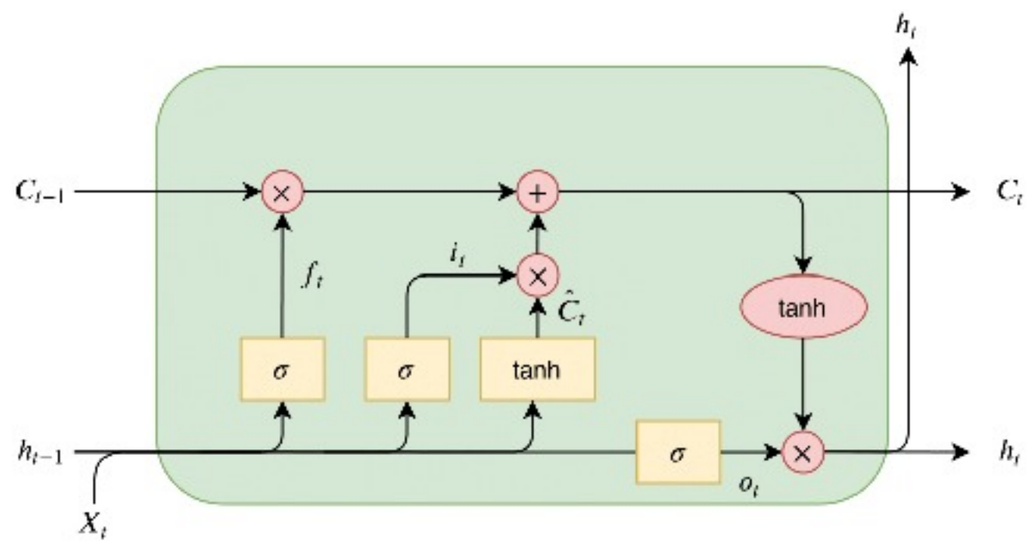
Persamaan ini menyatakan bahwa bentuk penalti kompleksitas yang diberikan pada struktur pohon. Penalti dapat mencegah pohon tumbuh terlalu kompleks melalui jumlah daun. Penalti juga menahan bobot daun agar tidak ekstrem. Dengan persamaan sebagai berikut:

$$\Omega(f) = \gamma N + \frac{1}{2}\lambda \sum_{j=1}^{N} w_j^2 \quad \text{.....(8)}$$

Pada persamaan ini, N merepresentasikan jumlah daun, w_j adalah bobot atau keluaran pada daun ke-*j,* kemudian γ berfungsi mengendalikan kecenderungan pohon untuk menambah daun (penalti sktruktur), dan λ menahan bobot daun agar tidak ekstrem (penalti skala), dari sudut pandang metodologis, interpretasi ini mengaskan bahwa γ mendorong model untuk melakukan pemecahan *node* bila memberikan manfaat yang signifikan, sedangkan λ mengurangi sensitivitas terhadap variasi kecil pada data dengan menekan bobot yang terlalu besar. Kombinasi keduanya membentuk mekanisme formal yang menjaga stabilitas *XGBoost* meskipun jumlah pohon dan kapasitas model meningkat.

2.6 *Long Short-Term Memory* (*LSTM*)

Long Short-Term Memory merupakan varian dari *recurrent neural network* yang dirancang untuk memodelkan data berurutan dengan mempertahankan informasi yang menjangkau lintas langkah waktu. Tantangan klasik pada *recurrent neural network* standar adalah *vanishing* maupun *exploding gradient* pada urutan panjang, sehingga sinyal pembelajaran dari langkah yang jauh dapat melemah dan konteks jangka panjang sulit dipertahankan . *LSTM* mengatasi hal tersebut dengan memperkenalkan *cell state* sebagai jalur memori utama serta mekanisme *gates* yang mengatur informasi yang dilupakan, ditambahkan, dan diekspos sebagai keluaran[31, 33]. Uraian selanjutnya disajikan melalui tahapan-tahapan yang memperlihatkan alur kerja *LSTM* secara konsisten, sebagai berikut[31,33,41]:



Gambar 2.2 Arsitektur Model *LSTM*

1. *Forget Gate*

Forget gate merupakan tahapan yang menentukan bagian mana dari memori sebelumnya yang tetap relevan untuk dipertahankan pada langkah waktu selanjutnya. Tahapan ini memiliki peran penting karena memori lama tidak senantiasa bermanfaat, sehingga harus disaring secara selektif. Secara matematis, proses seleksi diwujudkan melalui fungsi *sigmoid* yang menghasilkan koefisien retensi pada rentang 0 hingga 1, yang dituliskan sebagai berikut:

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right) \quad (9)$$

Persamaan ini merepresentasikan mekanisme pembentukan koefisien retensi memori untuk setiap komponen *cell state.* Vector $[h_{t-1}, x_z]$ berfungsi sebagai representasi gabungan yang merangkum informasi historis melalui h_{t-1} dan sinyal baru x_t, sehingga keputusan mempertahankan atau melupakan informasi ditentukan oleh kombinasi kondisi masa lalu dan masukan terkini. Transformasi linear $W_f[h_{t-1}, x_t] + b_f$ menghasilkan skor untuk setiap dimensi memori, kemudian fungsi *sigmoid* $\sigma(\cdot)$ memetakan skor tersebut ke rentang 0 hingga 1 dan berperan sebagai katup pengendali. Setiap elemen $f_t{}^{(j)}$ menunjukkan tingkat retensi memori pada dimensi ke- j, dengan nilai mendekati 1 menandakan retensi kuat dan nilai mendekati 0 menandakan kecenderungan eliminasi. Bias b_f turut memengaruhi kecenderungan awal gerbang, sehingga dapat memperkuat pilihan untuk menahan atau melupakan informasi sebelum proses pembelajaran berlangsung. Mekanisme pengendalian ini menegaskan bahwa *LSTM* menggunakan gerbang untuk mengatur informasi yang dipertahankan maupun diubah sehingga menghasilkan stabilitas yang lebih baik dibandingkan *RNN* konvensional.

2. *Input Gate*

Input Gate merupakan tahapan yang mengatur informasi baru yang akan ditambahkan ke memori, sehingga pembaruan tidak terjadi secara bebas melainkan terkendali. Tahap ini terdiri dari dua komponen dengan *sigmoid* yang menetukan banyak informasi baru yang akan ditulis, sedangkan *tanh* membentuk kandidat informasi baru yang stabil untuk dimasukkan ke memori. Dengan pemisahan ini *LSTM* dapat memasukkan informasi baru secaar terukur tanpa mengganggu stabilitas memori, yang secara matematis dituliskan sebagai berikut:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad \text{.....(10)}$$

$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad \text{.....(11)}$$

Kedua persamaan ini memisahkan tahapan penulisan memori menjadi dua keputusan yang saling melengkapi. Aspek awal berkaitan dengan sejauh mana informasi baru diizinkan masuk, sedangkan aspek berikutnya berhubungan dengan bentuk atau informasi yang akan ditulis. Pada tahap awal, i_t melalui fungsi *sigmoid* menghasilkan koefisien bernilai 0 hingga 1 yang merepresentasikan tingkat izin penulisan ke memori. Dengan demikian, meskipun terdapat sinyal baru pada x_t, sinyal yang masuk tidak secara otomatis memperbarui memori, melainkan harus melewati control yang diberikan oleh i_t. Selanjutnya $\tilde{c}_t$, dibentuk menggunakan $tanh(\cdot)$ sehingga nilainya berada pada rentang -1 hingga 1. Pembatasan rentang diperlukan karena variabel yang dianalisis merupakan konten yang berpotensi masuk ke memori utama. Jika nilainya tidak dikendalikan, pembaruan memori dapat menjadi terlalu besar dan menyebabkan representasi tidak stabil. Dengan demikian, i_t berperan dalam mengendalikan besarnya pembaruan, sedangkan $\tilde{c}_t$ mengatur arah atau isi pembaruan.

3. *Memory Update Equation*

   Memory update equation merupakan tahapan pembaruan *cell state* menggabungkan dua sumber informasi: memori lama yang dipertahankan oleh gerbang lupa dan informasi baru yang dipilih oleh gerbang input. Hasil penggabungan ini menjadi memori utama terbaru yang membawa konteks ke langkah waktu berikutnya. Secara konseptual, bentuk pembaruan yang bersifat aditif dan terkontrol ini berperan penting untuk menjaga aliran informasi tetap stabil pada sekuens panjang, yang secara matematis dituliskan sebagai berikut:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{c}_t \quad \text{..... (12)}$$

Persamaan pembaruan *cell state* merupakan inti dari *LSTM* karena mendefinisikan bagaimana memori utama diperbarui secara terkontrol melalui penggabungan aditif dua komponen, yaitu memori lama yang disaring oleh gerbang lupa ($f_t \odot C_{t-1}$) dan informasi baru yang dipilih oleh gerbang input ($i_t \odot \tilde{C}_t$). Operator perkalian elemen ($\odot$) memungkinkan kontrol terjadi secara independen per dimensi memori, sehingga setiap komponen dapat dipertahankan atau diubah secara selektif. Suku pertama, $f_t \odot C_{t-1}$, merepresentasikan memori lama yang hanya diteruskan jika relevan, sedangkan suku kedua, $i_t \odot \tilde{C}_t$, menambahkan informasi baru yang telah disaring. Bentuk aditif ini memungkinkan *LSTM* melakukan pembaruan bertahap tanpa harus menghapus dan menulis ulang memori secara keseluruhan pada setiap langkah. Dalam pembelajaran sekuensial, jalur memori yang stabil ini berperan dalam menekan masalah ketidakstabilan gradien pada *RNN* konvensional melalui mekanisme kontrol gerbang yang terstruktur. Sebagai ilustrasi, jika $f_t \approx 1$ dan $i_t \approx 0$, maka $C_t \approx C_{t-1}$ (memori dipertahankan); sebaliknya, jika $f_t \approx 0$, pembaruan didominasi oleh informasi baru.

4. *Output Gate*

Output gate merupakan tahapan yang menentukan bagian dari memori yang akan ditampilkan sebagai *hidden state,* sebagai representasi keluaran yang digunakan untuk langkah waktu berikutnya atau untuk menghasilkan prediksi akhir. Tahapan *output gate* menjadikan keluaran tetap selektif, mengingat tidak semua konten memori relevan untuk ditampilkan pada setiap waktu. Secara matematis, *sigmoid* memilih porsi informasi yang diekspos, kemudian *tanh* menstabilkan nilai memori sebelum dibentuk menjadi keluaran, yang dituliskan sebagai berikut:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad \text{.....(13)}$$

$$h_t = o_t \odot tanh\,(C_t) \quad \text{.....(14)}$$

Kedua persamaan ini mengatur transformasi memori C_t menjadi representasi keluaran h_t. Kemudian diawali dengan pembangkitan koefisien o_t melalui fungsi *sigmoid*, yang menghasilkan nilai kontinu antara 0 dan 1. Koefisien tersebut merepresentasikan tingkat eksposur, yaitu seberapa besar isi memori dapat ditampilkan sebagai keluaran pada langkah waktu t. Pengendalian ini diperlukan karena memori C_t sering menyimpan informasi yang lebih lengkap daripada yang perlu diekspos pada setiap langkah. Sebelum dikeluarkan, nilai memori dibatasi terlebih dahulu oleh fungsi $\tanh(C_t)$ agar keluaran tidak berada pada rentang ekstrem.

Perkalian elemen $o_t \odot \tanh(C_t)$ kemudian menjadikan proses penampilan keluaran bersifat selektif per dimensi. Dengan mekanisme ini, *LSTM* memisahkan fungsi penyimpanan (C_t) dan fungsi ekspresi keluaran (h_t), sehingga informasi dapat disimpan tanpa harus selalu diekspos. Pemisahan fungsi ini merupakan bentuk pengendalian informasi melalui gerbang yang berkontribusi terhadap kemampuan *LSTM* dalam mempertahankan dependensi jangka panjang secara lebih stabil.

2.7 *Prophet*

Prophet merupakan model *decomposable time series* yang dikembangkan oleh *facebook* untuk memodelkan data deret waktu melalui pemisahan komponen tren, musiman, dan efek fenomena khusus. Secara konseptual, model ini dibangun dalam kerangka *additive* model, sehingga nilai observasi tidak diperlakukan sebagai satu pola tunggal, melainkan sebagai gabungan beberapa sumber variasi yang dibentuk secara terpisah lalu dijumlahkan kembali menjadi prediksi akhir[42]. Pendekatan ini penting karena dalam banyak deret waktu, termasuk data penjualan ritel, perubahan data sering kali tidak hanya disebabkan oleh satu mekanisme, melainkan oleh interaksi antara tren jangka panjang, pola musiman, dan kejadian khusus yang terjadi pada periode tertentu. Literatur terdahulu menunjukkan bahwa *prophet* memiliki kemampuan untuk menangani perubahan pola yang bersifat non-linier, mendeteksi titik perubahan tren, serta tetap dapat diterapkan pada data yang mengandung *missing values*, langkah waktu yang tidak sepenuhnya teratur, dan *outliers*, tanpa menuntut prapemrosesan yang terlalu kompleks[42]. Karakteristik ini membuat *prophe*t relevan untuk data ritel, karena data penjualan harian umumnya dipengaruhi oleh variasi musiman, promosi, hari besar, perubahan permintaan, serta ketidakstabilan observasi yang dapat menyebabkan pola data menjadi tidak seragam.

Secara arsitektural, *prophet* disusun secara modular agar setiap sumber variasi dalam deret waktu dapat dimodelkan menurut fungsinya masing-masing. Komponen tren digunakan untuk merepresentasikan arah perubahan umum data dalam jangka panjang, sehingga model dapat membedakan antara pertumbuhan, penurunan, atau pergeseran level data dari waktu ke waktu[8,42]. Komponen musiman kemudian digunakan untuk membaca pola berulang yang muncul secara periodik, baik dalam skala harian, mingguan, maupun tahunan, sehingga fluktuasi berulang tidak tercampur dengan komponen tren[35]. Di samping itu, efek hari libur atau kejadian khusus diperlakukan sebagai komponen tersendiri agar pengaruh dari peristiwa yang hanya muncul pada tanggal tertentu tidak disalahartikan sebagai pola musiman reguler. Fleksibilitas *prophet* juga diperkuat oleh kemampuannya

mendeteksi *changepoints* secara otomatis, mengakomodasi *external regressors*, serta membentuk interval prediksi untuk merepresentasikan ketidakpastian hasil ramalan[8,35]. Dengan struktur seperti ini, model tidak hanya menghasilkan prediksi, tetapi juga menyediakan kerangka interpretasi yang jelas terhadap bagaimana suatu pola deret waktu dibentuk.

Secara matematis, *prophet* dinyatakan dengan formulasi sebagai berikut[8]:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \quad \text{.....(15)}$$

$y(t)$: nilai observasi pada waktu t

$g(t)$: komponen tren non-periodik

$s(t)$: komponen musiman atau pola periodik

$h(t)$: pengaruh hari libur, fenomena khusus, atau efek eksternal tertentu

ε_t : nilai error atau residu

Formulasi tersebut menunjukkan bahwa prediksi pada *prophet* dibentuk sebagai penjumlahan eksplisit dari beberapa komponen utama. Dengan demikian, perubahan pada data tidak dipahami hanya sebagai satu nilai ramalan akhir, tetapi sebagai hasil akumulasi kontribusi tren, musiman, efek kejadian khusus, dan variasi acak yang tersisa. Struktur matematis ini menjadi dasar mengapa *prophet* relatif mudah diinterpretasikan dalam analisis deret waktu.

Rincian tiap komponen dapat dijelaskan sebagai berikut[8,42]:

1. *Trend* $g(t)$

   Trend digunakan untuk menangkap arah perubahan jangka panjang. Dalam Prophet, tren dapat dibentuk dengan fungsi *piecewise linear* maupun pertumbuhan logistik, sehingga laju perubahan tidak diasumsikan tetap sepanjang waktu. Perubahan kemiringan tren ditangkap melalui *changepoints*, yaitu titik ketika struktur data bergeser dan model perlu menyesuaikan kemiringan tren secara otomatis.

2. *Seasonality* $s(t)$

   Musiman digunakan untuk memodelkan pola berulang yang muncul secara sistematis. Pada implementasinya, komponen ini umumnya direpresentasikan dengan deret Fourier agar pola periodik yang kompleks tetap dapat ditangkap secara fleksibel, baik untuk musiman harian, mingguan, maupun tahunan

3. *Holiday* $h(t)$

   Holiday digunakan untuk mengakomodasi efek hari raya, *special days*, atau *event* tertentu seperti promo. Komponen ini penting karena lonjakan atau penurunan permintaan pada tanggal tertentu sering kali tidak dapat dijelaskan hanya oleh tren dan musiman regular.

4. *Error term* ε_t

   Error term merepresentasikan residu atau *noise*, yaitu variasi acak yang belum dapat dijelaskan oleh ketiga komponen utama model.

2.8 Evaluasi Model Prediksi

Evaluasi model merupakan tahapan krusial dalam pengembangan model regresi dan peramalan *(forecasting)* karena berfungsi untuk mengukur sejauh mana hasil prediksi model mendekati nilai aktual. Metrik evaluasi memberikan dasar objektif untuk membandingkan kinerja antar model serta menilai reliabilitas model dalam merepresentasikan pola data historis dan memproyeksikan nilai di masa depan. Dalam pendekatan regresi dan *forecasting*, metrik evaluasi umumnya berfokus pada besarnya kesalahan prediksi *(error-based metrics)* serta kemampuan model dalam menjelaskan variasi data *(goodness-of-fit metrics)*[31,43].

Pada penelitian ini digunakan empat metrik evaluasi, yaitu *Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Symmetric Mean Absolute Percentage Error (sMAPE),* dan *Koefisien Determinasi* (R^2). Keempat metrik ini saling melengkapi karena masing-masing memiliki karakteristik sensitivitas dan interpretasi yang berbeda terhadap kesalahan prediksi, sebagai berikut[44,45]:

1. *Mean Absolute Error* (*MAE*)

   Mean Absolute Error (*MAE*) merupakan metrik yang digunakan untuk mengukur rata-rata nilai absolut selisih antara nilai actual dan nilai prediksi. MAE memberikan Gambaran langsung mengenai besarnya kesalahan prediksi dalam satuan yang sama dengan data asli sehingga mudah diinterpretasikan. Secara matematis, MAE dirumuskan sebagai berikut:

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t| \quad \text{.....(16)}$$

y_t : nilai aktual pada pengamatan ke-t

$\hat{y}_t$: nilai hasil prediksi pada pengamatan ke-t

n : jumlah total data pengamatan

Komponen $| y_t - \hat{y}_t |$ merepresentasikan besar kesalahan absolut pada setiap titik observasi. MAE memperlakukan seluruh kesalahan secara linear tanpa memberikan penalti tambahan pada kesalahan besar. Oleh karena itu, MAE dianggap stabil dan robust terhadap outlier dibandingkan metrik berbasis kuadrat. Dalam *forecasting* penjualan dan regresi, MAE sering digunakan untuk menilai performa model secara umum karena interpretasinya yang sederhana dan konsisten.

2. *Root Mean Squared Error* (*RMSE*)

Root Mean Squared Error (*RMSE*) merupakan akar dari rata-rata kuadrat kesalahan prediksi. RMSE memberikan penalty yang lebih besar terhadap kesalahan yang bernilai besar sehingga lebih sensitif terhadap outlier dibandingkan MAE.

Secara matematis, *RMSE* dinyatakan sebagai berikut:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \quad \text{.....(17)}$$

y_t : nilai aktual pada pengamatan ke-t

$\hat{y}_t$: nilai hasil prediksi pada pengamatan ke-t

n : jumlah total data pengamatan

Komponen $(y_t - \hat{y}_t)^2$ menunjukkan kesalahan kuadrat yang memperbesar kontribusi error dengan deviasi besar. Karakteristik ini membuat *RMSE* sangat efektif untuk mengevaluasi model yang diharapkan mampu meminimalkan kesalahan ekstrem, seperti pada peramalan penjualan ritel dan energi. Nilai *RMSE* yang lebih kecil mengindikasikan kinerja model yang lebih baik dan prediksi yang lebih mendekati nilai aktual.

3. *Symmetric Mean Absolute Percentage Error* (*sMAPE)*

Symmetric Mean Absolute Percentage Error (*sMAPE)* merupakan pengembangan dari metrik *Mean Absolute Percentage Error* (*MAPE*) yang bertujuan mengatasi masalah asimetri dan ketidakstabilan ketika nilai actual mendekati nol. *sMAPE* menormalisasi kesalahan absolut dengan rata-rata nilai aktual dan prediksi sehingga menghasilkan ukuran kesalahan dalam bentuk persentase yang lebih seimbang .

Secara matematis, *sMAPE* dinyatakan sebagai berikut:

$$sMAPE = \frac{100\%}{n}\sum_{t=1}^{n}\frac{|y_t - \hat{y}_t|}{\frac{1}{2}(|y_t| + |\hat{y}_t|)} \quad \text{.....(18)}$$

y_t : nilai aktual pada pengamatan ke-t

$\hat{y}_t$: nilai hasil prediksi pada pengamatan ke-t

n : jumlah total data pengamatan

Pembilang $|y_t - \hat{y}_t|$ menunjukkan besar kesalahan absolut, sedangkan penyebut $\frac{1}{2}(|y_t| + |\hat{y}_t|)$ berfungsi sebagai faktor normalisasi simetris. *sMAPE* banyak digunakan pada studi forecasting karena memberikan interpretasi relatif dalam persentase dan memungkinkan perbandingan performa antar dataset dengan skala berbeda. Dalam peramalan penjualan dan energi, *sMAPE* dinilai lebih representatif dibandingkan *MAPE* konvensional.

4. Koefisien Determinasi (R^2)

Koefisien Determinasi (R^2) merupakan metrik yang mengukur kemampuan model regresi dalam menjelaskan variasi data aktual. R^2 menunjukkan proporsi variansi variabel dependen yang dapat dijelaskan oleh variabel independent atau model prediksi.

Secara matematis, R^2 dirumuskan sebagai:

$$R^2 = 1 - \frac{\Sigma_{t=1}^{n}(y_t - \hat{y}_t)^2}{\Sigma_{t=1}^{n}(y_t - \bar{y})^2} \quad \text{.....(19)}$$

y_t : nilai aktual pada pengamatan ke-t

$\hat{y}_t$: nilai hasil prediksi pada pengamatan ke-t

$\bar{y}$: rata-rata nilai aktual pengamatan

n : jumlah total data pengamatan

Pembilang merepresentasikan jumlah kuadrat error model (*residual sum of squares*), sedangkan penyebut menunjukkan total variasi data aktual (*total sum of squares*). Nilai R^2 berada pada rentang 0 hingga 1, di mana nilai mendekati 1 menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam menjelaskan variasi data . Dalam studi regresi dan *forecasting* berbasis *machine learning,* R^2 sering digunakan sebagai indikator kelayakan model secara keseluruhan .

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

Cross industry standard process for data mining (*CRISP-DM*) merupakan kerangka kerja terstruktur yang digunakan untuk mengelola proyek *data mining, data science,* dan *machine learning* secara sistematis. Kerangka ini menekankan pentingnya kolaborasi antara

tujuan organisasi dan aktivitas analitik, sehingga pengembangan model tidak terlepas dari kebutuhan pemangku kepentingan serta konteks penerapannya. Selain memberikan struktur yang terarah, *CRISP-DM* juga bersifat iteratif, yang memungkinkan perbaikan berulang apabila hasil yang diperoleh belum memenuhi indikator keberhasilan yang telah ditetapkan [12].

CRISP-DM terdiri atas enam tahapan utama yang membentuk alur kerja proyek secara menyeluruh. Keenam tahapan tersebut bersifat iteratif dan dapat diulang sesuai kebutuhan pengembangan serta temuan selama proses analisis, dengan rincian sebagai berikut[9,12,46]:

1. *Business Understanding*

   Tahap *Business Understanding* bertujuan merumuskan tujuan proyek secara jelas dengan menerjemahkan kebutuhan organisasi ke dalam permasalahan analitik yang terdefinisi secara terukur. Pada tahap ini ditetapkan ruang lingkup, batasan, asumsi, risiko, serta indikator keberhasilan yang akan digunakan sebagai acuan evaluasi. Kejelasan perumusan tujuan menjadi krusial karena akan menentukan arah keseluruhan proses analitik, termasuk jenis data yang diperlukan, pendekatan yang dipilih, dan bentuk keluaran yang diharapkan.

2. *Data Understanding*

   Tahap *Data Understanding* berfokus pada pengenalan karakteristik data melalui proses pengumpulan, deskripsi, eksplorasi, dan pemeriksaan kualitas. Aktivitas yang dilakukan mencakup identifikasi struktur data, analisis distribusi, pendeteksian nilai hilang, *outlier*, serta inkonsistensi format yang berpotensi memengaruhi proses analitik. Pada data yang memiliki dimensi waktu, perhatian juga diberikan pada kontinuitas dan kelengkapan deret waktu. Pemahaman yang komprehensif terhadap data diperlukan untuk memastikan bahwa data yang digunakan benar-benar merepresentasikan permasalahan yang dikaji serta untuk mencegah bias dalam tahap pemodelan dan evaluasi.

3. *Data Preparation*

   Tahap *Data Preparation* merupakan proses penyusunan dataset akhir yang siap digunakan pada fase pemodelan. Kegiatan pada tahap ini meliputi pembersihan data, transformasi, pengkodean variabel kategorikal, normalisasi atau standarisasi, serta rekayasa dan seleksi fitur. Literatur menegaskan bahwa *preprocessing* memiliki peran strategis dalam meningkatkan kualitas representasi data, sehingga berdampak langsung pada stabilitas dan performa model yang dihasilkan. Oleh karena itu, tahap ini tidak hanya bersifat teknis, tetapi juga menentukan kualitas informasi yang akan

dipelajari oleh model. Dalam praktiknya, fase ini dapat mengalami penyesuaian ulang apabila hasil pemodelan menunjukkan kebutuhan perbaikan pada struktur atau fitur data.

4. *Modeling*

Tahap *Modeling* mencakup pemilihan metode analitik, pelatihan model, serta penyetelan parameter untuk memperoleh hasil yang optimal sesuai tujuan proyek. Pemilihan teknik pemodelan dilakukan dengan mempertimbangkan karakteristik data dan kebutuhan penggunaan, termasuk aspek interpretabilitas dan keterbatasan implementasi. Proses ini umumnya dilakukan secara iteratif melalui pengujian beberapa alternatif pendekatan dan perbandingan performa secara sistematis. Hubungan antara tahap pemodelan dan persiapan data bersifat dinamis, karena penyesuaian model sering kali memerlukan perubahan pada fitur atau struktur data yang digunakan.

5. *Evaluation*

Tahap *Evaluation* bertujuan menilai apakah model yang dikembangkan telah memenuhi indikator keberhasilan yang ditetapkan pada tahap awal. Evaluasi dilakukan dengan menggunakan metrik yang sesuai serta dengan memeriksa konsistensi performa pada kondisi data yang berbeda. Studi terdahulu juga menunjukkan pentingnya prosedur evaluasi yang robust untuk meminimalkan bias estimasi performa dan memastikan bahwa hasil yang diperoleh dapat digeneralisasikan secara memadai . Apabila hasil evaluasi belum memenuhi kriteria, proses dapat dikembalikan ke tahap sebelumnya untuk dilakukan perbaikan pada data maupun strategi pemodelan.

6. *Deployment*

Tahap *Deployment* merupakan proses penerapan hasil analitik agar dapat dimanfaatkan secara praktis. Penerapan ini dapat berupa integrasi ke dalam sistem, penyajian dalam bentuk laporan atau *dashboard*, maupun penyediaan rekomendasi yang mendukung pengambilan keputusan. Pada tahap ini juga diperhatikan aspek kesiapan implementasi, termasuk kejelasan dokumentasi dan kemudahan interpretasi hasil. Studi terdahulu menegaskan bahwa keberhasilan proyek analitik diukur dari sejauh mana hasil dapat digunakan secara efektif dan berkelanjutan, tidak hanya dari capaian performa model.

2.10 Penelitian Terdahulu

Kajian terhadap penelitian terdahulu dilakukan untuk memetakan pendekatan, algoritma, *metrik* evaluasi, serta temuan yang telah dihasilkan oleh studi-studi sebelumnya sebagai landasan komparatif dalam memposisikan kontribusi penelitian ini secara ilmiah, yang dirangkum pada Tabel 2.1.

Tabel 2.1 Tabel Penelitian Terdahulu

No	Referensi	Judul	Algoritma	Metriks Evaluasi	Kontribusi	Hasil penelitian	Relevansi
1	[11]	*Hyperparameter Optimization Using Grid Search and Random Search to Improve the Performance of Prediction Models with Decision Trees*	*Decision Tree with tuning random search and grid search*	*Accuracy, Precision, recall, F1 score.*	Perbandingan dua optimasi *hyperparameter* untuk peningkatan performa klasifikasi kanker payudara	Temuan menunjukkan bahwa *grid search* memberikan solusi parametrik paling komprehensi, sedangkan *random search* unggul dalam capaian akurasi puncak (97,37%) dengan waktu komputasi yang lebih singkat.	Relevansinya terletak pada landasan metodologis untuk *hyperparameter tuning*. Meskipun domain penelitiannya merupakan klasifikasi medis, studi ini mendukung bahwa *tuning* sistematis perlu dilakukan sebelum membandingkan performa model akhir.
2	[33]	Optimisasi *Hyperparameter BiLSTM* Menggunakan *Bayesian*	*BiLSTM with Bayesian Optimization*	*MAPE*	menunjukkan *Bayesian Optimization* efektif mencari	Hasil evaluasi mengindikasikan bahwa *MAPE* merupakan metrik	Relevansinya terletak pada sensitivitas tinggi dari model turunan *LSTM*

		Optimization untuk Prediksi Harga Saham			*hyperparameter BiLSTM* dengan error rendah	yang paling sesuai untuk menganalisis ketiga saham *blue-chip* di Indonesia.	terhadap *hyperparameter,* sehingga temuan ini dapat dijadikan pembanding dalam menjelaskan pentingnya proses *tuning* pada model *deep learning.*
3	[24]	*Deep Learning based Seasonality and Trend Detection in Sales Forecasting*	*Hybrid CNN-LSTM, CNN, LSTM*	*RMSE, MAE, MAPE*	Pendekatan *Hybrid Deep Learning* memadukan kemampuan ekstraksi fitur dari *CNN* serta pemodelan ketergantungan temporal melalui *LSTM* guna mengidentifikasi pola tren dan musiman.	Hasil menunjukkan bahwa model *hybrid CNN-LSTM* unggul dengan *MAE* 1219,79% dibanding model tunggal *CNN MAE* 2137,89% atau *LSTM MAE* 1720,23%	Relevansinya terletak pada kesamaan domain penelitian yang membahas peramalan penjualan serta menunjukkan bahwa model *hybrid* lebih unggul dibandingkan model tunggal dalam menangani pola penjualan yang kompleks.
4	[47]	*Carbon Emission Prediction And Reduction Analysis Of Wastewater Treatment Plants Based On Hybrid*	*SVR, ANN ,GDBT, AdaBoost, DTR, MLR, GWO-SVR-ANN*	R^2, *RMSE, MAE, MAPE*	Menyusun *Framework Hybrid* prediksi emisi karbon IPAL yang menggabungkan seleksi fitur, model gabungan,	Hasil penelitian menunjukkan bahwa model *Hybrid GWO-SVR-ANN* memiliki performa yang kuat dengan nilai *MAPE* 0.49%, R^2 0.9926%	Relevansinya terletak pada konsep *hybrid learning* dan optimasi parameter. Meskipun domain penelitiannya tidak

		Machine Learning Models			dan optimasi metaheuristik, lalu menerjemahkan prediksi ke skenario pengurangan.	dibanding model 6 model Tunggal.	membahas penjualan, akan tetapi studi ini dijadikan penguat argument bahwa peningkatan akurasi dan generalisasi dapat dicapai melalui penggabungan model.
5	[8]	*A Hybrid LSTM-Prophet Model for Sales Forecasting and Inventory Optimization in E-Commerce Time Series*	*Prophet, LSTM and tuning with weighted fusion, Bayesian optimization*	*MAPE, MAE, R^2 ,RMSE*	Suatu kerangka *LSTM-Prophet end-to-end* diusulkan untuk meningkatkan akurasi prediksi penjualan sekaligus mendukung pengendalian inventori dinamis berbasis ketidakpastian prediksi.	*MAE* dilaporkan turun menjadi 8.7%, atau 29.3% lebih rendah daripada *Prophet* tunggal dan 17.1% lebih rendah daripada *LSTM* tunggal. Selain itu, *inventory turnover* meningkat 18.4% dan *stock-out rate* ditekan menjadi 3.2%.	Relevansi penelitian ini ditunjukkan secara langsung pada kombinasi *LSTM dan Prophet* dalam ranah *sales forecasting*. Studi ini memiliki keterkaitan erat dengan arah penelitian yang dilakukan, terutama dari sisi integrasi model serta pemanfaatan pola tren-musiman dan non-linieritas secara simultan.

6	[48]	*Time Series-Based Demand Forecasting: A Comparative Analysis of Holt-Winters, SARIMA, and Prophet Models on Retail Inventory Data*	*Holt-Winters, SARIMA, Prophet*	*MAE,RMSE, MAPE*	Perbandingan tiga model deret waktu klasik dilakukan pada data inventori ritel untuk menilai kemampuan masing-masing model dalam menangkap tren dan musiman.	Penelitian ini menujukkan bahwa model *Holt-Winters* menghasilkan error terendah dan meunjukkan performa terbaik dibanding *SARIMA* dan *Prophet* pada data uji.	Relevansinya ditunjukkan sebagai studi pembanding bagi *Prophet* dalam peramalan permintaan ritel. Penelitian ini digunakan untuk memperkuat analisis mengenai kelebihan dan keterbatasan *Prophet* ketika dibandingkan dengan model deret waktu lainnya.
7	[5]	*Retail Commodity Sales Prediction: A Prophet-LightGBM Combined Machine Learning Approach*	*Prophet, LightGBM, SVR, LSTM,ARIMA, Prophet-LightGBM*	*MAE,MSE,R^2*	Suatu model *hybrid Prophet-LightGBM* diusulkan, di mana *Prophet* digunakan untuk mendekomposisi komponen tren, musiman, dan efek hari libur, yang selanjutnya dijadikan sebagai fitur masukan dalam pemodelan	Penelitian ini menunjukkan bahwa model *hybrid Prophet-LightGBM* memperoleh nilai *MAE* 0,0621%, *MSE* 0,0071% dan R^2 0,84%. Dengan hasil tersebut menunjukkan bahwa performanya lebih unggul dibanidngkan *SVR, LSTM, ARIMA,* maupun *Prophet* tunggal	Relevansinya ditunjukkan pada kesamaan domain, yaitu prediksi penjualan ritel, serta pendekatan menggunakan *Prophet* yang dimanfaatkan sebagai pengekstrak pola waktu dan model *machine learning* digunakan untuk peningkatan akurasi.

					menggunakan *LightGBM*.		
8	[49]	*Forecasting of Sales Based on Long Short Term Memory Algorithm with Hyperparameter*	*LSTM*	*MSE, RMSE*	Optimasi *hyperparameter* pada model *LSTM* untuk prediksi penjualan 60 hari ke depan dilakukan melalui serangkaian pengujian terhadap komposisi data, jumlah *hidden layer*, nilai *dropout*, ukuran *batch*, dan jumlah *epoch*.	Penelitian ini menunjukkan bahwa konfigurasi terbaik diperoleh pada *batch size* 30, *epoch* 150, 3 *hidden layer*, dan 3 *dropout*, dengan *RMSE training* 0.0855% dan *RMSE testing* 0.0846%.	Relevansi penelitian ini ditunjukkan pada penggunaan *LSTM* untuk prediksi penjualan serta penegasan bahwa *tuning hyperparameter* berpengaruh langsung terhadap akurasi model.
9	[50]	*Seasonal electric vehicle forecasting model based on machine learning and deep learning techniques*	*Neural Network, LSTM, GRU, ANFIs*	*RMSE, MAE ,MAPE ,R^2*	Mengintegrasikan dekomposisi musiman dan pembelajaran deret waktu untuk meningkatkan akurasi prediksi beban pengisian *EV* sehingga membantu penjadwalan	Penelitian menunjukkan *GRU* memberikan prediksi beban pengisian *EV* yang lebih akurat daripada *LSTM* dengan akurasi 99,3% (musim dingin), 97,7% (musim semi), dan 98,3% (musim panas) serta 98,08% (musim gugur),	Relevansinya terletak pada aspek pemodelan musiman. Karena faktor cuaca, hari besar, dan pola waktu turut dipertimbangkan dalam pemodelan, studi ini dijadikan pendukung bahwa penangkapan pola musiman

					pengisian, pengadaan energi, dan stabilitas jaringan listrik.	sedangkan model *ANFIS* mencapai akurasi 99,35% (dingin), 98,04% (semi), 98,56% (panas), dan 97,06% (gugur) sehingga secara keseluruhan *ANFIS* menunjukkan performa paling konsisten pada peramalan beban pengisian *EV* lintas musim.	secara eksplisit merupakan hal yang penting.
10	[51]	*Bitcoin Price Forecasting: A Comparative Study of Machine Learning, Statistical and Deep Learning Models*	*Linear Regressor, Random Forest, XGBoost Regressor, ARIMA, Prophet, CNN, RNN, LSTM, GRU, Extra Trees Classifier*	*MAPE*	Memberikan penjelasan komparatif pemilihan model pada pasar *crypto* yang volatil sehingga bermanfaat untuk analisis risiko dan strategi keputusan berbasis prediksi yang lebih *robust*.	*ExtraTreesRegressor* menghasilkan *MAPE* uji terendah 0,0609% pada horizon 36, mengungguli model *Deep Learning* dan statistik yang dibandingkan pada kajian.	Relevansinya terletak pada dilakukannya perbandingan antara *XGBoost, Prophet,* dan *LSTM* dalam satu studi. Hal ini dapat dijadikan acuan dalam justifikasi pemilihan ketiga model *baseline* yang digunakan.