

BAB II

KAJIAN LITERATUR

2.1. Pasar Saham

Pasar saham merupakan bagian dari sistem keuangan yang berfungsi sebagai sarana pertemuan antara pihak yang memiliki kelebihan dana dan pihak yang membutuhkan dana melalui mekanisme perdagangan saham, di mana harga saham mencerminkan nilai ekspektasi investor terhadap kinerja perusahaan di masa depan [24]. Harga saham di pasar saham bersifat dinamis karena dipengaruhi oleh berbagai faktor yang saling berinteraksi, seperti kondisi ekonomi, informasi perusahaan, faktor politik, serta perilaku dan sentimen investor, sehingga pergerakannya sulit diprediksi secara sederhana [24]. Karakteristik utama pasar saham adalah volatilitas dan ketidakpastian yang tinggi, yang disebabkan oleh sifat data pasar saham yang nonlinier, *non-stationary*, dan sangat sensitif terhadap informasi baru [25]. Fluktuasi harga saham terjadi sebagai respons pasar terhadap informasi yang tersedia, baik informasi fundamental perusahaan, indikator teknikal, maupun informasi eksternal seperti berita ekonomi dan isu global, sehingga pasar saham menjadi sistem yang kompleks dan adaptif [26].

Dalam kajian teori keuangan, pasar saham sering dikaitkan dengan konsep efisiensi pasar, yang menyatakan bahwa harga saham telah mencerminkan seluruh informasi yang tersedia di pasar [24]. Namun, penelitian empiris menunjukkan bahwa pasar saham tidak selalu berada dalam kondisi efisien sempurna, karena masih terdapat pola dan anomali harga yang dapat dianalisis dan dimanfaatkan untuk tujuan prediksi [25]. Ketidakefisienan pasar inilah yang membuka peluang bagi penggunaan pendekatan analitis dan komputasional dalam mempelajari pergerakan harga saham [27]. Seiring perkembangan teknologi dan ketersediaan data dalam jumlah besar, penelitian pasar saham mengalami pergeseran dari metode statistik tradisional menuju pendekatan berbasis kecerdasan buatan dan pembelajaran mesin [27]. Metode konvensional seperti model linear dan ekonometrika klasik dinilai kurang mampu menangkap kompleksitas pasar saham yang bersifat nonlinier dan dipengaruhi oleh banyak variabel secara simultan [27]. Oleh karena itu, pendekatan berbasis *machine learning* dan *deep learning* banyak digunakan dalam penelitian pasar saham untuk meningkatkan akurasi analisis dan prediksi harga saham [27].

Penelitian terdahulu menunjukkan bahwa analisis pasar saham yang hanya mengandalkan data historis harga memiliki keterbatasan, sehingga diperlukan integrasi

berbagai jenis data untuk memperoleh pemahaman yang lebih komprehensif [27]. Penggabungan data harga saham dengan indikator teknikal dan sentimen pasar mampu memberikan hasil analisis yang lebih baik dibandingkan pendekatan tunggal [26]. Hal ini menegaskan bahwa pasar saham dipengaruhi tidak hanya oleh faktor kuantitatif, tetapi juga oleh faktor kualitatif yang mencerminkan perilaku investor [26].

Perilaku dan sentimen investor merupakan faktor penting dalam membentuk dinamika pasar saham, karena keputusan investasi sering kali dipengaruhi oleh emosi, persepsi risiko, dan ekspektasi terhadap kondisi pasar di masa depan [24]. Sentimen positif atau negatif yang berkembang di pasar dapat memicu aksi beli atau jual secara masif, sehingga menyebabkan fluktuasi harga saham yang signifikan meskipun tidak terjadi perubahan fundamental perusahaan [24]. Kondisi ini menunjukkan bahwa pasar saham tidak sepenuhnya rasional dan dapat mengalami distorsi akibat faktor psikologis investor [25]. Berbagai penelitian terdahulu membuktikan bahwa penggunaan model berbasis *neural network* dan *deep learning* mampu menangkap pola kompleks dalam pergerakan pasar saham dengan lebih baik dibandingkan metode tradisional [25]. Terdapat penelitian yang menyatakan bahwa model *deep learning* memiliki keunggulan dalam mempelajari ketergantungan jangka panjang dan hubungan nonlinier pada data pasar saham, sehingga lebih sesuai digunakan dalam konteks prediksi harga saham [27]. Temuan ini memperkuat relevansi penggunaan pendekatan analisis modern dalam penelitian pasar saham, khususnya untuk saham dengan volatilitas tinggi [27].

2.2. Prediksi Harga Saham

Prediksi harga saham merupakan proses memperkirakan nilai harga saham di masa mendatang dengan memanfaatkan data historis serta pola pergerakan harga yang terbentuk di pasar saham. Pasar saham sendiri bekerja berdasarkan mekanisme permintaan dan penawaran, di mana harga saham mencerminkan reaksi investor terhadap informasi, kondisi ekonomi, serta kinerja perusahaan yang bersangkutan [28]. Dalam konteks pasar modal, pergerakan harga saham bersifat dinamis dan cenderung fluktuatif, sehingga diperlukan pendekatan analitis yang mampu menangkap pola historis dari data tersebut [29].

Secara teoritis, prediksi harga saham banyak dilakukan menggunakan pendekatan *time series forecasting*, di mana waktu dijadikan sebagai variabel utama dalam analisis. Pendekatan ini berfokus pada pola historis seperti *trend*, musiman, dan autokorelasi untuk memproyeksikan nilai harga saham di masa depan [29]. Karakteristik data harga saham yang sering kali non-

stationary mengharuskan adanya proses pengujian dan penyesuaian data agar model prediksi dapat bekerja secara optimal [29].

Perkembangan teknologi mendorong pemanfaatan metode *machine learning* dan *deep learning* dalam prediksi harga saham karena kemampuannya dalam menangkap hubungan nonlinier dan pola kompleks yang sulit diidentifikasi oleh metode statistik konvensional. Algoritma *machine learning* seperti *Support Vector Machine (SVM)*, *Random Forest*, dan *K-Nearest Neighbors (KNN)* memiliki potensi yang signifikan dalam memprediksi pergerakan harga saham dengan tingkat akurasi yang cukup tinggi, tergantung pada karakteristik data dan parameter model yang digunakan [29]. Hal ini menunjukkan bahwa pemilihan metode prediksi sangat berpengaruh terhadap hasil prediksi harga saham.

Terdapat penelitian yang membandingkan model *SARIMA* dan *Prophet* dalam memprediksi harga saham berdasarkan data historis harga penutupan. Hasil penelitian tersebut menunjukkan bahwa kedua model mampu menangkap pola pergerakan harga saham, namun model *Prophet* cenderung memberikan hasil prediksi yang lebih baik dalam kondisi tertentu dibandingkan model *SARIMA* [29]. Meskipun demikian, penelitian tersebut juga menegaskan bahwa prediksi harga saham tetap memiliki keterbatasan karena sifat pasar saham yang sangat dipengaruhi oleh faktor eksternal dan peristiwa tak terduga [29].

Prediksi harga saham dalam penelitian ini diposisikan sebagai upaya untuk memanfaatkan data historis pasar saham guna memperoleh gambaran pergerakan harga saham di masa depan. Data harga saham dianalisis menggunakan metode prediksi yang sesuai untuk mengidentifikasi pola pergerakan harga, sehingga hasil prediksi yang diperoleh dapat dijadikan sebagai bahan pertimbangan dalam pengambilan keputusan [29]. Dengan demikian, kajian mengenai prediksi harga saham menjadi dasar penting dalam penelitian ini untuk menghubungkan antara data pasar saham, metode analisis, dan hasil prediksi yang dihasilkan

2.3. OSEMN

OSEMN adalah sebuah kerangka kerja dalam *data science* yang terdiri dari lima langkah utama, yaitu *Obtain*, *Scrub*, *Explore*, *Model*, dan *Interpret* [30]. Kerangka kerja ini dirancang sebagai metode literatur dalam proses analisis data untuk memastikan setiap langkah penting yang dilakukan secara sistematis, sehingga hasil analisis yang diperoleh memiliki tingkat keakuratan yang tinggi dan dapat diandalkan. Metode *OSEMN* digunakan untuk mengekstrak informasi dan pengetahuan dari data, serta dapat diterapkan pada

berbagai jenis proyek *data science* [31]. Adapun proses *OSEMN* dapat diuraikan sebagai berikut [30].

1. *Obtain Data*

Obtain Data menggambarkan proses pengumpulan data yang diperlukan untuk penelitian atau proyek tertentu, informasi bisa diperoleh dari berbagai sumber, termasuk basis data (*database*), *Application Programming Interface (API)*, serta metode pengambilan data menggunakan *web scraping* [30]. Data yang diperoleh perlu memiliki relevansi dengan tujuan penelitian dan mutu yang tinggi agar bisa dipakai dalam tahap analisis selanjutnya. Setelah data terkumpul, proses analisis dilanjutkan ke tahap *Scrub*, *Explore*, *Model*, dan *Interpret*.

2. *Scrub Data*

Scrub Data adalah proses pembersihan dan prapemrosesan data yang bertujuan untuk memastikan data dalam keadaan siap untuk dianalisis [30]. Proses ini mencakup penghapusan data yang berulang, penanganan nilai-nilai yang hilang, serta penyesuaian format data agar memenuhi kebutuhan analisis. Tahap *scrub* memiliki fungsi yang sangat krusial karena data yang berkualitas tinggi akan memengaruhi akurasi hasil analisis [32]. Melalui pembersihan data yang komprehensif, kesalahan dan ketidakkonsistenan dalam data bisa diminimalkan sebelum melanjutkan ke proses analisis berikutnya

3. *Explore Data*

Explore Data adalah tahapan penting yang harus dilakukan sebelum melanjutkan ke proses pembuatan model *Machine learning (ML)* [30]. Proses ini, biasanya dikenal sebagai *Explore Data Analysis (EDA)*, melibatkan analisis menyeluruh terhadap semua sifat data untuk memastikan kebersihan dan kesiapannya. Melalui penerapan teknik statistik, kita dapat menganalisis struktur data yang paling penting untuk diproses pada tahap pemodelan selanjutnya [31]

4. *Model Data*

Tahap pemodelan data sering dipandang sebagai pusat dari proyek *data science* karena dalam proses ini prediksi dilakukan [30]. Pada tahap ini, dipilih model yang paling tepat untuk memproses data dan menghasilkan estimasi atau prediksi di masa depan. Kunci keberhasilan pada tahap ini terletak pada pemilihan *fitur* atau karakteristik data yang paling penting agar hasil prediksi yang diperoleh akurat dan sesuai tujuan [31].

5. *Interpret Data*

Interpret Data adalah tahap terakhir dalam metode *OSEMN* yang berfokus pada penyampaian hasil untuk menjelaskan masalah bisnis serta memberikan wawasan strategis

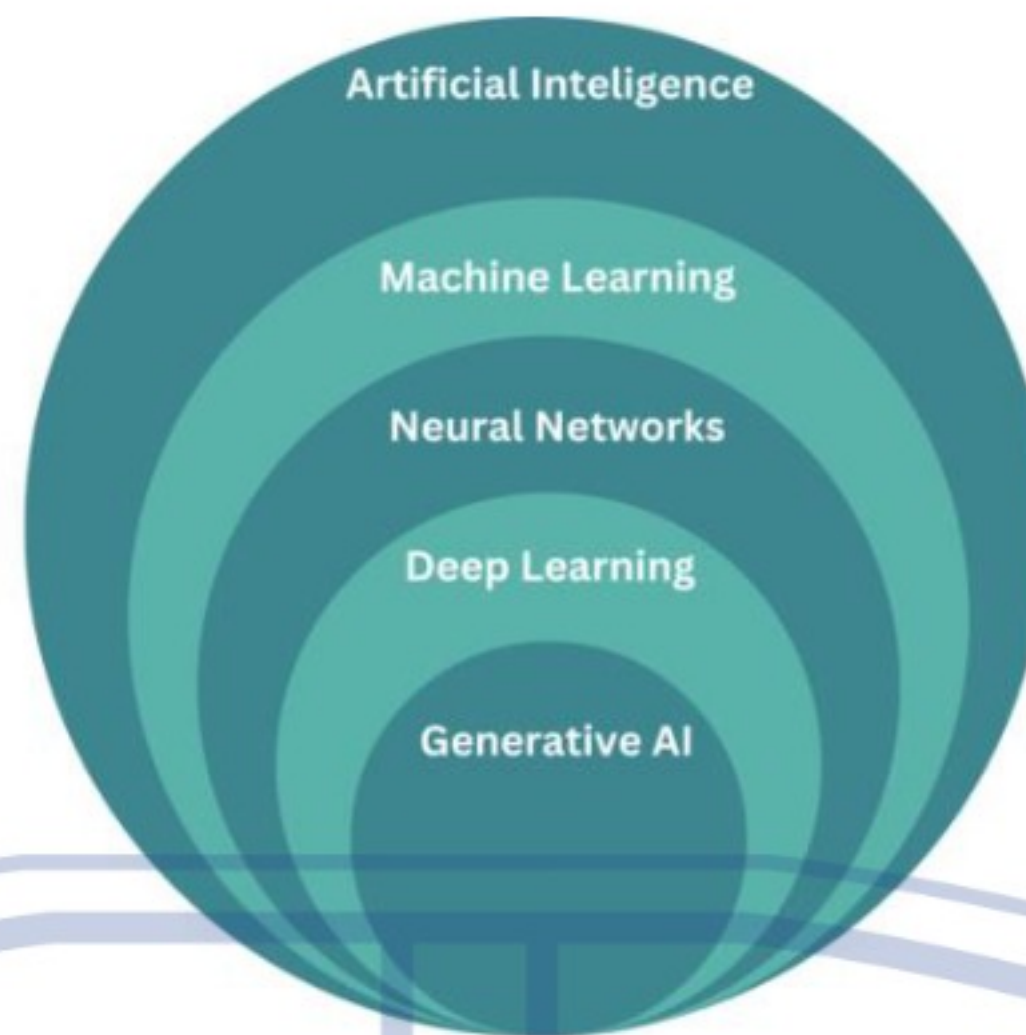
yang dapat diimplementasikan melalui data science [30]. Walaupun bukan fase teknis, tahap ini membutuhkan pemahaman mendalam untuk meringkas hasil model secara objektif dan kritis sehingga kesimpulan yang diperoleh memiliki dasar yang kokoh dan dapat dibandingkan dengan penelitian lain [32]. Dengan menguraikan hasil secara terperinci dan logis, proses ini berperan dalam menentukan langkah-langkah tindakan ke depan, termasuk menggunakan analitik prediktif untuk mengulangi keberhasilan atau menghindari dampak negatif di waktu yang akan datang.



Gambar 2.1 Tahapan *OSEMN*

2.4. Deep learning

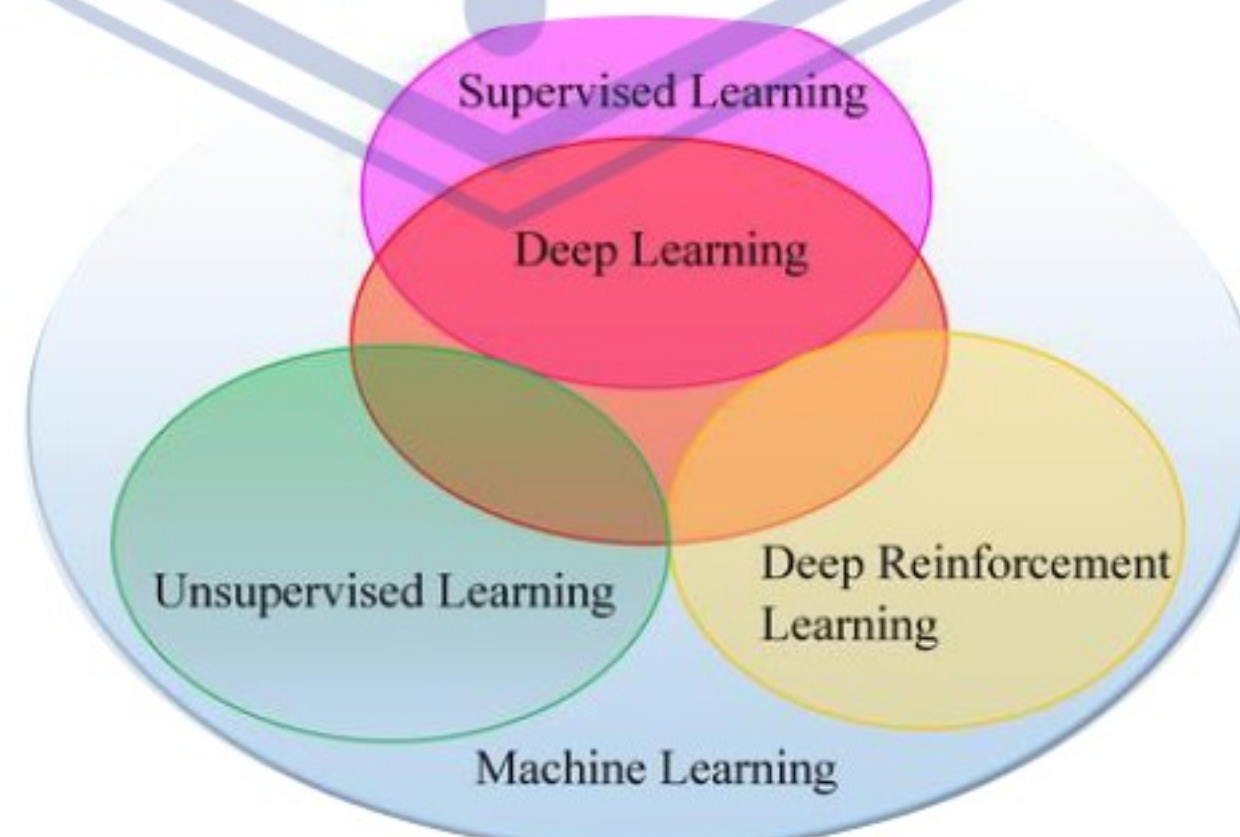
Deep learning merupakan bagian dari *Machine learning* yang memanfaatkan jaringan saraf buatan (*Artificial neural network*) dengan banyak lapisan (*deep neural networks*) untuk menganalisis pola rumit dari data dalam skala besar [33]. Pendekatan ini terinspirasi dari cara kerja sistem saraf manusia dalam mengolah informasi. *Deep learning* dapat secara otomatis mengekstrak *fitur* dari data mentah tanpa perlu proses rekayasa *fitur* yang rumit seperti pada metode konvensional [33]. Taksonomi ini dapat dilihat pada Gambar 2.2 di bawah ini.



Gambar 2.2 Taksonomi *Deep learning*

Deep learning juga merupakan metode yang menggunakan beberapa lapisan tersembunyi (*hidden layer*) yang terletak di antara lapisan *input layer* dan lapisan *output layer*. Struktur bertingkat ini memungkinkan proses pemodelan *non-linear* dilakukan melalui berbagai tahap, sehingga dapat dimanfaatkan untuk menganalisis *fitur* serta mengenali pola dari data. Setiap lapisan dalam *deep learning* mempunyai peran untuk merepresentasikan tingkat abstraksi yang bervariasi, mulai dari *fitur* dasar hingga *fitur* yang lebih rumit [34].

Deep learning memiliki beberapa lapisan yang tertata secara berurutan (*cascade*) untuk melakukan ekstraksi dan transformasi *fitur*. Hasil yang diperoleh dari satu lapisan akan menjadi masukan untuk lapisan selanjutnya, sehingga informasi dapat dipelajari secara bertahap. Algoritma *deep learning* dapat digunakan dalam pendekatan *supervised* dan *unsupervised*, dimana masing-masing dipakai untuk tujuan klasifikasi (*supervised*) dan analisis pola (*unsupervised*) pada data [35].



Gambar 2.3 Metode *Deep learning* [36]

Beragam metode dalam *Deep learning* dapat dibagi menjadi beberapa kategori, yaitu *supervised*, *semi-supervised*, dan *unsupervised*. Selain itu, ada metode lain seperti *Reinforcement learning (RL)* atau *Deep reinforcement learning (DRL)* yang dalam beberapa referensi sering dikategorikan sebagai bagian dari *semi-supervised* atau *unsupervised learning*. Klasifikasi itu dapat diilustrasikan dalam gambar 2.3 di atas dan dijelaskan sebagai berikut [36]:

1. *Deep Supervised learning* adalah teknik *learning* yang menggunakan data *training* yang sudah dilabeli sebelumnya (*labeled data*). Metode ini sering diterapkan dalam tugas klasifikasi dan prediksi dimana model dilatih menggunakan pasangan data *input* dan *output* yang telah diketahui. Contoh algoritma yang tergolong dalam kategori ini meliputi *Deep Neural networks (DNN)*, *Convolutional Neural networks (CNN)*, *NeuralProphet*, *Gated Recurrent Unit (GRU)* dan *Recurrent Neural networks (RNN)*.
2. *Deep Semi-supervised learning* memanfaatkan gabungan data berlabel dan data tidak berlabel sebelumnya. Metode ini bertujuan untuk memperbaiki kinerja model saat jumlah data berlabel yang tersedia terbatas. Berbagai metode yang masuk dalam kategori mencakup *Deep reinforcement learning (DRL)* dan *Generative Adversarial Networks (GAN)*.
3. *Deep Unsupervised learning* adalah metode yang tidak memerlukan data yang sudah diberi label. Model dalam kategori ini menekankan pada proses pengambilan pola penyajian data secara mandiri. Contoh algoritma yang sering dipakai adalah *Autoencoders (AE)*, *Restricted Boltzmann Machines (RBM)* dan berbagai variasi dari *Generative Adversarial Networks (GAN)*.
4. Sementara itu, *Deep reinforcement learning* adalah metode yang digunakan dalam lingkungan yang belum sepenuhnya dipahami (*unknown environments*). Model ini berinteraksi dengan lingkungan untuk mengoptimalkan hadiah tertentu. Perkembangan *Deep reinforcement learning* mulai menarik perhatian besar sejak tahun 2013, sejalan dengan munculnya penelitian yang dikembangkan oleh *Google DeepMind*.
5. Salah satu kelebihan utama dari *deep learning* adalah kemampuannya untuk mengelola hubungan *non-linear* dan data yang sangat kompleks. Dengan demikian, *deep learning* banyak diterapkan di berbagai sektor, seperti pengenalan gambar, pemrosesan bahasa alami, dan analisis data *time-series*. Dalam konteks prediksi runtun waktu, *deep learning* dapat mempelajari hubungan jangka pendek dan

jangka panjang antar data, yang umumnya sulit ditangkap oleh model statistik tradisional [37].

2.5. Data *Time-series*

Data *time-series* adalah sekumpulan observasi yang dicatat secara berurutan berdasarkan waktu dengan interval konsisten, sehingga nilai pada satu titik waktu dipengaruhi oleh nilai-nilai sebelumnya dan memiliki keterkaitan temporal yang kuat [38]. Karakteristik data *time series* mencakup *trend*, musiman, siklus, dan komponen acak (*noise*), yang perlu dipahami secara teoritis sebelum dilakukan pemodelan karena komponen-komponen tersebut mempengaruhi pola prediksi jangka panjang dan pendek [38]. Data *time series* pada sektor keuangan dan pasar saham umumnya bersifat nonlinier dan volatil, sehingga menimbulkan tantangan dalam pemodelan dan prediksi menggunakan pendekatan konvensional [38].

Pendekatan ini bertujuan untuk mengidentifikasi pola yang muncul dari data lama serta menarik kesimpulan dari pola tersebut untuk periode yang akan datang, sehingga hasilnya dapat digunakan sebagai dasar atau pedoman dalam melakukan prediksi nilai di masa depan

2.5.1. Plot Runtun Waktu

Plot runtun waktu merupakan representasi yang digunakan untuk menunjukkan data yang terorganisir menurut urutan waktu tertentu [39]. Plot runtun waktu bertujuan untuk mendukung pemahaman tentang perilaku data sepanjang waktu, termasuk mengidentifikasi pola *trend*, musiman (*seasonality*), siklus, dan fluktuasi atau anomali yang mungkin terjadi [39]. Contoh plot runtun waktu dapat dilihat pada gambar 2.4



Gambar 2.4 Contoh Plot Runtun Waktu [40]

Berikut adalah jenis-jenis dari plot yang digunakan analisis *time-series*:

1. Plot Waktu

Plot waktu merupakan representasi grafis data runtun waktu yang menunjukkan nilai variabel seiring dengan urutan waktu secara berurutan. Plot ini digunakan untuk mengamati pola umum dalam data, seperti *trend* jangka panjang, variasi, perubahan *structural*, dan tahap awal dalam analisis *time-series* karena memberikan gambaran awal tentang karakteristik data [40]

2. Plot Musiman

Plot musiman digunakan untuk mengenali pola yang berulang dan terjadi secara periodik dalam jangka waktu tertentu, seperti bulanan atau tahunan. Dalam plot ini, data dikelompokkan menurut periode musiman yang serupa, sehingga pola musiman dapat dibandingkan dari tahun ke tahun. Plot musiman sangat bermanfaat untuk mengidentifikasi konsistensi dan kekuatan pengaruh musiman dalam data seri waktu [40]

3. Plot Musiman Polar

Plot musiman polar adalah variasi dari plot musiman yang ditampilkan dalam format radial atau melingkar. Setiap titik di plot merepresentasikan periode tertentu dengan sudut yang mencerminkan musim dan jarak dari pusat menunjukkan nilai data [40].

4. Plot Dekomposisi *Time-series*

Plot dekomposisi *time-series* berfungsi untuk memisahkan data runtun waktu menjadi beberapa komponen utama, yaitu *trend*, musiman, dan residual (*noise*). Dekomposisi ini memberikan pemahaman yang lebih rinci tentang struktur data dengan memisahkan setiap komponen, sehingga mempermudah proses analisis, pemodelan, dan prediksi pada data *time-series* [40].

5. Plot Autokorelasi

Plot autokorelasi digunakan untuk menilai hubungan antara nilai data pada waktu tertentu dan nilai pada waktu yang lebih awal (*lag*). Plot ini menunjukkan koefisien pada beberapa *lag*, sehingga dapat dimanfaatkan untuk mengidentifikasi ketergantungan temporal dalam data [40]

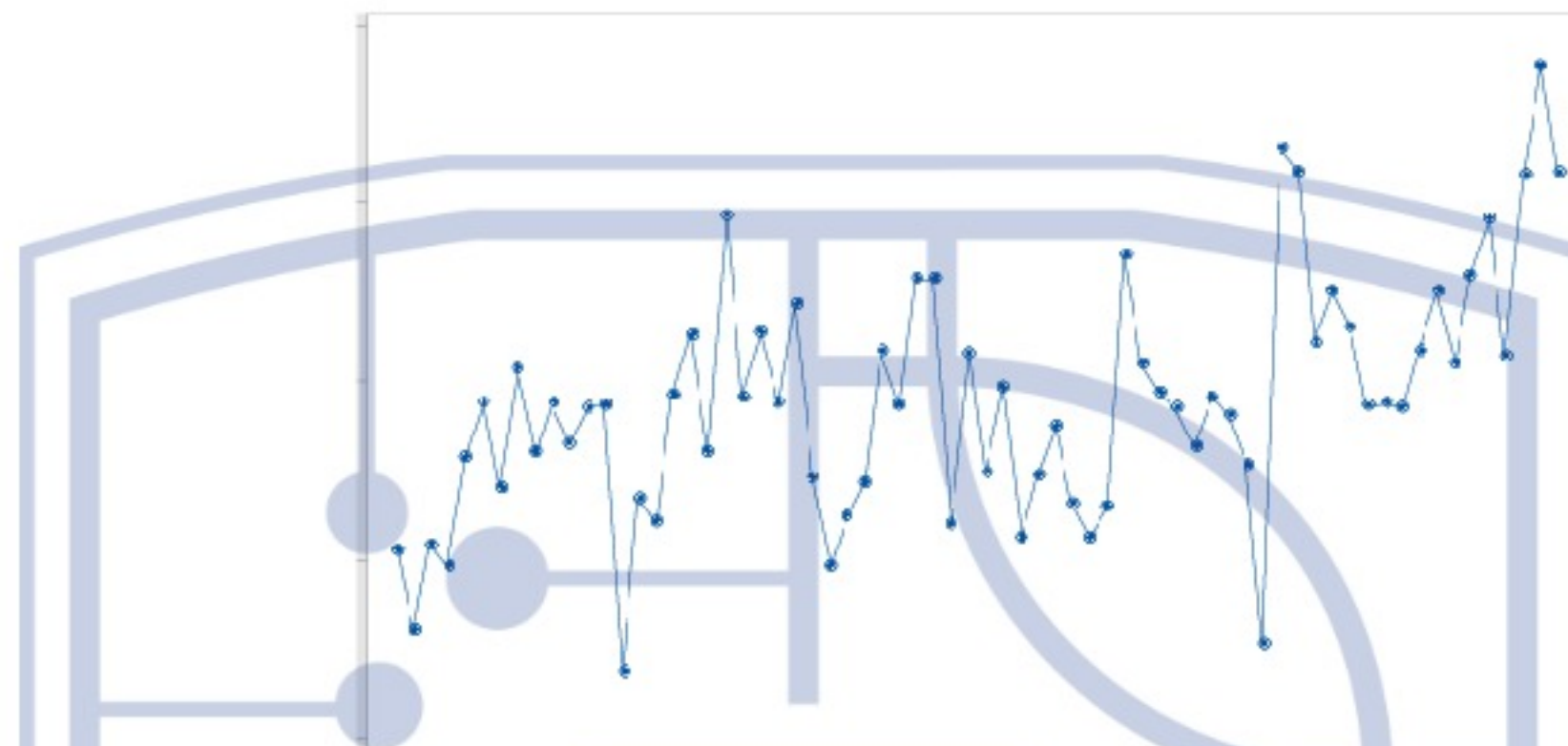
2.5.2. Pola Runtun Waktu

Pola runtun waktu (*time series patterns*) merupakan karakteristik atau struktur spesifik yang terbentuk dari data yang diobservasi secara berurutan berdasarkan waktu. Pola ini menunjukkan karakteristik data seiring berjalannya waktu dan menjadi dasar penting

dalam analisis serta pemodelan *time-series*. Terdapat beberapa pola dalam runtun waktu yaitu:

1. Horizontal

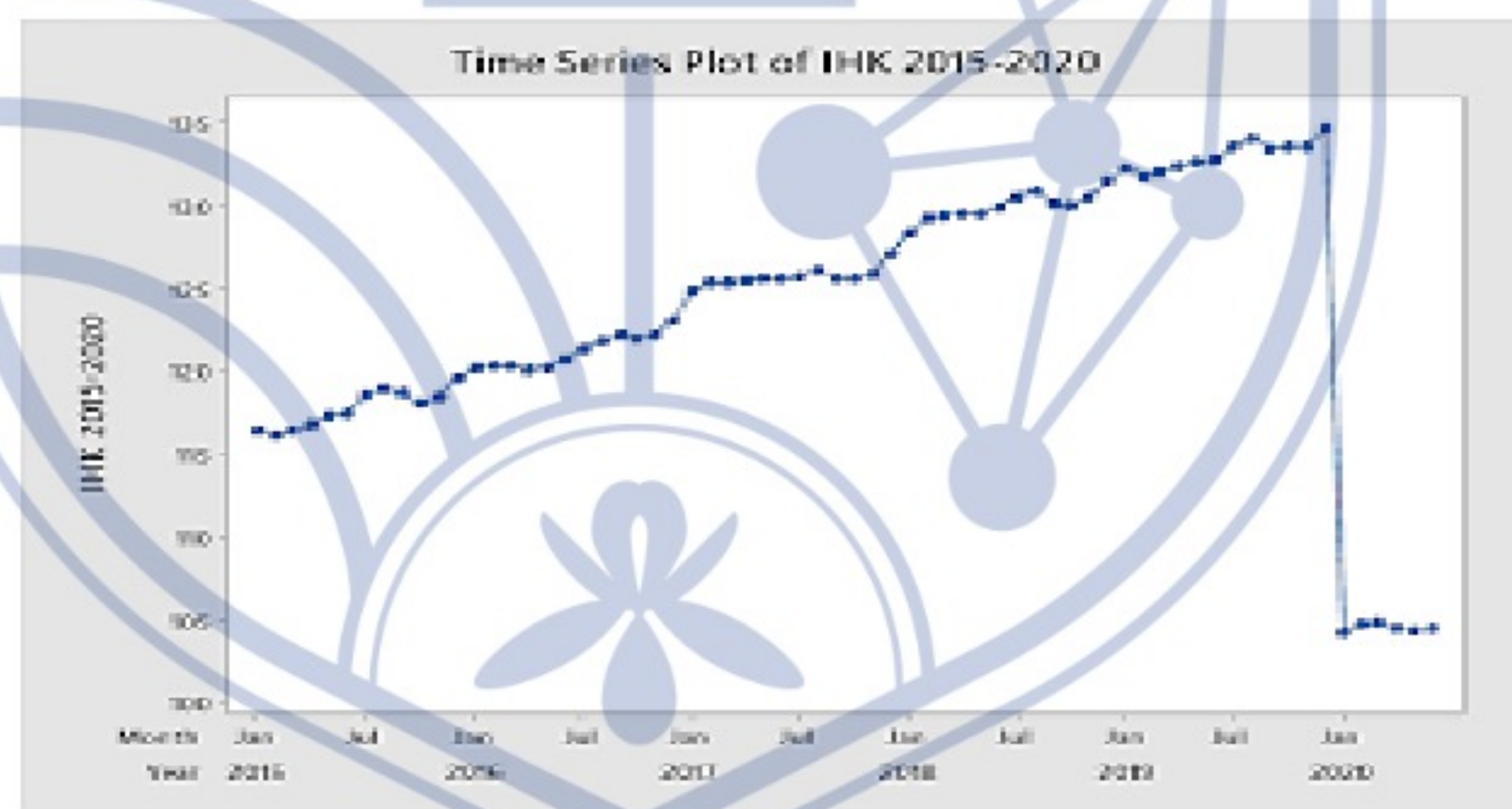
Pola horizontal adalah pola runtun waktu di mana data bergerak fluktuatif di sekitar nilai rata-rata yang cukup stabil tanpa menunjukkan arah naik atau turun dalam periode panjang [41].



Gambar 2.5 Contoh Horizontal [41]

2. Trend

Pola *trend* merupakan pola runtun waktu yang menunjukkan kecenderungan pergerakan data secara teratur dalam jangka panjang, baik yang mengalami kenaikan (*trend* naik) maupun penurunan (*trend* turun) [42]

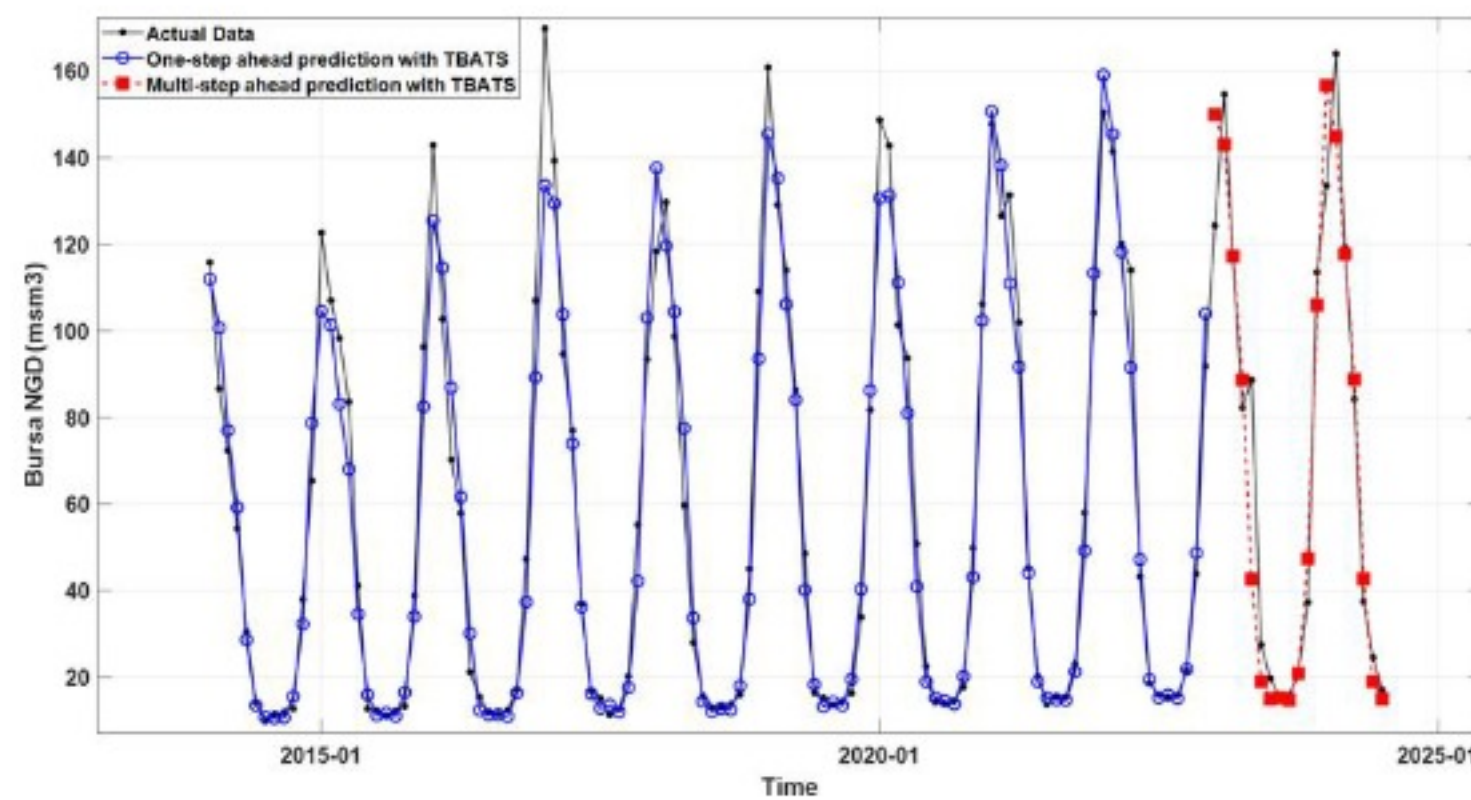


[42]

Gambar 2.6 Contoh *Trend* [35]

3. Musiman

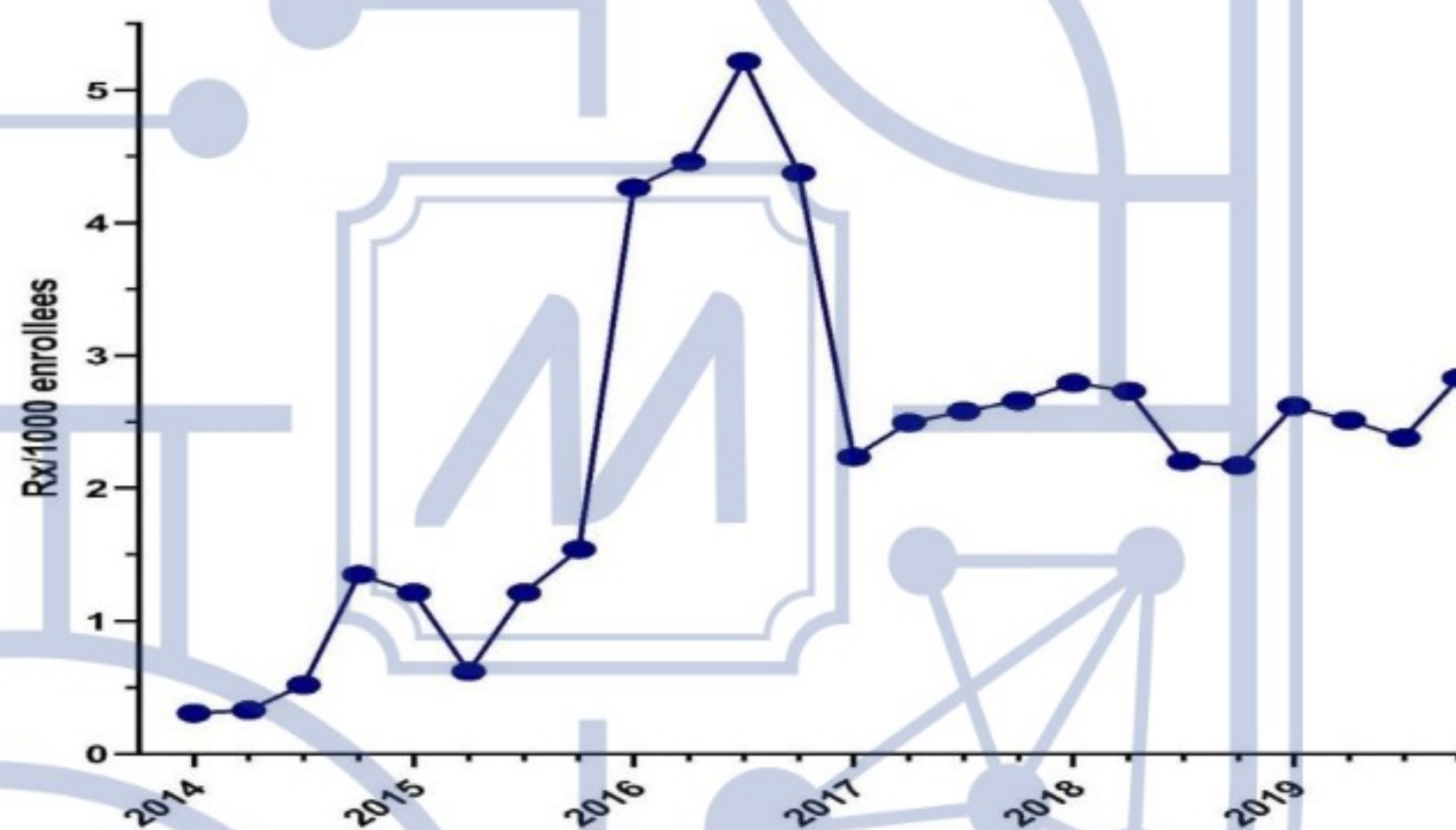
Pola musiman adalah pola yang ditandai dengan variasi data yang terjadi secara berulang dan teratur dalam rentang waktu tertentu, seperti harian, bulanan, atau tahunan [43]



Gambar 2.7 Contoh Musiman [43]

4. Siklus

Pola siklus menggambarkan variasi data yang berlangsung dalam periode panjang dan tidak memiliki interval tetap seperti pola musiman [43]



Gambar 2.8 Contoh Siklus [43]

2.6. Analisis *Time series*

Analisis *time series* atau *time-series* merupakan metode kuantitatif yang digunakan untuk memodelkan serta memprediksi data yang tersusun secara berurutan berdasarkan interval waktu tertentu. Pendekatan ini menitikberatkan pada hubungan temporal antar observasi, di mana suatu nilai pada periode tertentu dipengaruhi oleh nilai pada periode sebelumnya dalam rangkaian waktu yang sama. Selain untuk memahami pola historis, analisis *time series* juga bertujuan menghasilkan estimasi yang dapat dimanfaatkan sebagai dasar pengambilan keputusan strategis pada masa mendatang [44]. Secara konseptual, data *time-series* memiliki beberapa komponen utama, yaitu *trend* (*trend*), pola musiman (*seasonality*), siklus (*cyclical*), serta variasi acak (*random variations*). Identifikasi terhadap setiap komponen tersebut merupakan tahapan penting dalam proses analisis, karena perbedaan struktur pola akan menentukan jenis model yang digunakan dalam pemodelan

[45]. Dalam bidang ekonomi dan keuangan, data *time series* umumnya bersifat *non-stationary* dan menunjukkan tingkat volatilitas yang dinamis, sehingga diperlukan transformasi data seperti *differencing* untuk mencapai kondisi *stationary* sebelum dilakukan proses pemodelan [46]. Penerapan analisis *time series* telah meluas ke berbagai bidang, antara lain ekonomi regional, prediksi penjualan, pergerakan harga saham, hingga prediksi kasus pandemi. Pada penelitian terkait dinamika ekonomi regional, metode *time-series* terbukti mampu mengidentifikasi *trend* jangka panjang, pola musiman, serta fluktuasi acak yang kompleks pada variabel seperti *PDRB*, inflasi, dan investasi [47]. Temuan tersebut menunjukkan bahwa analisis *time-series* tidak hanya berfungsi sebagai alat analisis statistik, tetapi juga sebagai instrumen perencanaan pembangunan yang berbasis data empiris.

Dalam implementasinya, model *time series* dapat dikelompokkan menjadi model statistik klasik dan model berbasis *machine learning*. Model statistik klasik seperti *ARIMA* dan *SARIMA* berfokus pada pola linier serta autokorelasi dalam data, dan dinilai efektif untuk prediksi jangka pendek hingga menengah [46]. Model *SARIMA* secara khusus dikembangkan untuk mengakomodasi pola musiman yang berulang dalam data *time-series*, sebagaimana dibuktikan dalam penelitian pada data penjualan industri otomotif yang menghasilkan tingkat akurasi tinggi melalui pemodelan musiman yang tepat [45]. Seiring perkembangan metode modern, model seperti *Prophet* menawarkan fleksibilitas dalam memodelkan *trend* nonlinier, perubahan struktural (*structural break*), serta pola musiman kompleks melalui pendekatan adaptif [44]. *Prophet* juga terbukti mampu menangani *missing value* dan *outlier* secara efektif, sehingga sesuai untuk data dengan dinamika perubahan yang signifikan. Dalam studi komparatif terhadap data pariwisata dan jumlah penumpang bandara, model *SARIMA*, *LSTM*, dan *Prophet* menunjukkan tingkat performa yang bervariasi bergantung pada karakteristik data yang dianalisis, yang mengindikasikan bahwa tidak ada satu model yang secara universal paling unggul untuk seluruh jenis *time-series* [48]. Lebih lanjut, penerapan analisis *time series* pada prediksi penjualan menunjukkan bahwa model *ARIMA* dapat mendukung perencanaan produksi dan pengambilan keputusan manajerial melalui estimasi permintaan yang lebih terukur berdasarkan data historis. Hal ini mempertegas bahwa analisis *time-series* memiliki implikasi praktis yang signifikan dalam mendukung perencanaan operasional serta strategi bisnis. Secara metodologis, tahapan analisis *time series* umumnya meliputi eksplorasi data, pengujian *stationary* dari data *time series*, identifikasi model melalui analisis *ACF* dan *PACF*, estimasi parameter, serta evaluasi performa model menggunakan Metrik kuantitatif seperti *MAE*, *RMSE*, atau *MAPE* [44], [46]. Rangkaian proses tersebut bertujuan memastikan bahwa model yang dibangun tidak hanya

mampu merepresentasikan pola historis secara akurat, tetapi juga memiliki kemampuan prediktif yang andal.

Dalam penelitian ini, analisis *time series* digunakan untuk mengevaluasi pergerakan harga saham *BYD* sebagai salah satu perusahaan pada sektor kendaraan listrik. Saham pada sektor kendaraan listrik cenderung memiliki dinamika yang tinggi karena dipengaruhi oleh perkembangan teknologi baterai, permintaan kendaraan listrik, kebijakan pemerintah, sentimen pasar, serta persaingan antarperusahaan otomotif. Kondisi tersebut menyebabkan data harga saham *BYD* berpotensi mengalami volatilitas tinggi, perubahan *trend*, *non-stationary*, dan *structural break*. Oleh karena itu, diperlukan perbandingan beberapa model *time series* yang memiliki karakteristik berbeda. *SARIMA* digunakan sebagai model statistik klasik yang mampu menangkap pola historis, autokorelasi, dan musiman yang relatif stabil. *Prophet* digunakan karena memiliki kemampuan dalam memodelkan perubahan *trend* dan *change point* secara fleksibel. Sementara itu, *NeuralProphet* digunakan karena menggabungkan pendekatan *Prophet* dengan *neural network* sehingga lebih mampu menangkap pola *non-linear* dan dinamika data yang kompleks. Dengan demikian, perbandingan ketiga model tersebut menjadi dasar logis untuk mengetahui model mana yang paling sesuai dalam memprediksi harga saham *BYD* pada sektor kendaraan listrik.

Dengan demikian, analisis *time series* merupakan pendekatan ilmiah yang sistematis dalam memahami dinamika data berbasis waktu. Metode ini memungkinkan identifikasi pola struktural, pemodelan hubungan temporal, serta pengembangan prediksi yang dapat mendukung pengambilan keputusan strategis di berbagai sektor, termasuk ekonomi, bisnis, dan keuangan. Dalam konteks penelitian ini, analisis *time series* menjadi pondasi utama dalam mengevaluasi serta membandingkan kinerja model *SARIMA*, *Prophet*, dan *NeuralProphet* pada data harga saham *BYD* yang bersifat dinamis, volatil, dan dipengaruhi oleh perkembangan industri kendaraan listrik..

2.7. Model *Time series*

Model *time series* adalah metode statistik dan komputasi yang digunakan untuk menganalisis dan meramalkan data yang terorganisir berdasarkan kronologi. Dalam model *time series*, setiap pengamatan dianggap memiliki hubungan dengan nilai pada periode sebelumnya, sehingga pola yang ada dapat digunakan untuk meramalkan nilai di masa depan. Secara umum, model *time series* bertujuan untuk mengenali dan memodelkan beberapa elemen utama dalam data, yaitu *trend* (*trend*), musiman (*seasonality*), siklus (*cycle*), dan fluktuasi acak (*noise*). Dalam konteks data keuangan seperti harga saham, sering

muncul karakteristik seperti *non-stationary*, volatilitas yang tinggi, serta adanya kemungkinan perubahan struktur *trend*. Karena itu, memilih model yang sesuai adalah faktor krusial dalam menghasilkan prediksi yang tepat dan konsisten. Dalam penelitian ini diterapkan tiga pendekatan utama, yaitu *SARIMA* sebagai model statistik konvensional, *Prophet* sebagai model aditif modern yang berfokus pada regresi waktu, dan *NeuralProphet* sebagai pengembangan berbasis jaringan saraf yang dapat menangkap pola nonlinier

2.7.1 *Seasonal Autoregressive Integrated Moving average (SARIMA)*

Seasonal Autoregressive Integrated Moving average (SARIMA) merupakan pengembangan dari model *Autoregressive Integrated Moving average (ARIMA)* yang dirancang untuk memodelkan data *time-series* yang memiliki pola musiman yang berulang secara periodik [49]. Model *SARIMA* banyak digunakan dalam prediksi data *time series* karena kemampuannya dalam menangkap komponen *trend*, fluktuasi acak, serta pola musiman yang tidak dapat dimodelkan secara optimal oleh *ARIMA* konvensional [50]. *SARIMA* diperkenalkan dalam kerangka metodologi *Box-Jenkins* sebagai model statistik parametrik yang efektif untuk prediksi jangka pendek hingga menengah pada data *time-series* musiman [51]. *Box-Jenkins* yang mengasumsikan bahwa pola historis data dapat digunakan untuk memprediksi nilai di masa depan melalui hubungan linier antar observasi pada waktu yang berbeda [51]. Keunggulan utama *SARIMA* dibandingkan *ARIMA* konvensional terletak pada kemampuannya dalam memodelkan efek musiman yang tidak dapat ditangkap oleh model *ARIMA* non-musiman [52]. Model ini telah diaplikasikan secara luas dalam berbagai bidang seperti ekonomi, kesehatan masyarakat, transportasi, dan keberlanjutan, termasuk dalam prediksi kecelakaan lalu lintas, profit perusahaan, serta data epidemiologi [49], [52].

Secara umum, model *SARIMA* dinotasikan sebagai $SARIMA(p, d, q)(P, D, Q)_s$, di mana parameter non-musiman dan musiman dimodelkan secara terpisah untuk menangkap dinamika data yang kompleks [53]. Parameter (p, d, q) merepresentasikan komponen non-musiman, sedangkan $(P, D, Q)_s$ merepresentasikan komponen musiman dengan periode musiman s . Parameter p menunjukkan orde *autoregressive* (AR), d menunjukkan tingkat diferensiasi (*differencing*), dan q menunjukkan orde *moving average* (MA), sementara P , D , dan Q masing-masing menunjukkan komponen AR musiman, *differencing* musiman, dan MA musiman. Model *SARIMA* secara matematis dapat dinyatakan sebagai berikut [49], [51]:

$$6. \quad \Phi_P(B^s)\phi_p(B)(1-B)^d(1-B^s)^D y_t = \theta_Q(B^s)\theta_q(B)\varepsilon_t \quad (1)$$

Di mana y_t merupakan nilai aktual *time-series* pada waktu ke- t , B adalah operator backshift, dan ε_t merupakan *error* acak yang diasumsikan berdistribusi normal dengan rata-rata nol dan variansi konstan [51]. Operator backshift B didefinisikan sebagai $By_t = y_{t-1}$, yang memungkinkan model menangkap ketergantungan nilai masa lalu terhadap nilai saat ini [52].

Komponen $\phi_p(B)$ merepresentasikan bagian *autoregressive* non-musiman, yang dapat dituliskan sebagai:

$$7. \quad \phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p \quad (2)$$

Sementara itu, komponen *moving average* non-musiman dinyatakan sebagai:

$$8. \quad \theta_q(B) = 1 + \theta_1 B + \theta_2 B^2 + \dots + \theta_q B^q \quad (3)$$

Untuk menangkap pola musiman, *SARIMA* menambahkan komponen *autoregressive* dan *moving average* musiman yang masing-masing dinyatakan sebagai:

$$\Phi_P(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{2s} - \dots - \Phi_P B^{Ps} \quad (4)$$

$$\Theta_Q(B^s) = 1 + \theta_1 B^s + \theta_2 B^{2s} + \dots + \theta_Q B^{Qs} \quad (5)$$

Pendekatan ini memungkinkan *SARIMA* untuk menangkap hubungan linier jangka pendek, *trend* jangka panjang, serta pola musiman yang berulang dalam interval waktu tertentu secara efektif [52]. Diferensiasi non-musiman $(1 - B)^d$ digunakan untuk menghilangkan *trend* pada data sehingga data menjadi *stationary*, yang merupakan syarat utama dalam pemodelan *ARIMA* dan *SARIMA* [51]. Selain itu, diferensiasi musiman $(1 - B^s)^D$ digunakan untuk menghilangkan pola musiman yang berulang agar model dapat mempelajari struktur data dengan lebih akurat [53].

Proses identifikasi parameter *SARIMA* umumnya dilakukan dengan menganalisis grafik *Autocorrelation Function (ACF)* dan *Partial Autocorrelation Function (PACF)* untuk menentukan nilai p , q , P , dan Q yang optimal [50]. Model yang dihasilkan kemudian dievaluasi menggunakan kriteria statistik seperti Akaike Information Criterion (AIC) dan Mean Absolute Percentage Error (MAPE) untuk memastikan kinerja prediksi yang optimal [49]. Model ini terbukti efektif untuk prediksi jangka pendek hingga menengah pada data *time-series* yang memiliki pola musiman yang stabil [53].

Namun demikian, *SARIMA* memiliki keterbatasan karena hanya mampu memodelkan hubungan linier dan kurang efektif dalam menangkap pola nonlinier yang kompleks dibandingkan dengan pendekatan berbasis *machine learning* dan *deep learning* [51]. Oleh karena itu, dalam beberapa penelitian, *SARIMA* sering dibandingkan atau dikombinasikan

dengan model lain seperti *LSTM* atau *NeuralProphet* untuk meningkatkan akurasi prediksi [52], [53].

2.7.2. *Prophet*

Prophet adalah model prediksi yang dikembangkan oleh *Facebook* untuk membuat model prediksi untuk proyek internal mereka berdasarkan model aditif di mana *trend non-linear* diperkirakan dengan musiman tahunan, mingguan, dan harian, ditambah efek hari libur [54]. model ini dirancang untuk menangani pola musiman yang kuat dan mampu menyesuaikan perubahan *trend* yang terjadi secara tidak teratur.

Prophet menunjukkan kinerja yang baik pada data historis yang mengandung nilai hilang, serta perubahan struktur pola *trend*, sehingga sering digunakan dalam analisis *time-series* dunia nyata [54]. Keunggulan lain dari *Prophet* adalah kemampuannya menggunakan keseluruhan data historis dalam proses pemodelan tanpa memerlukan pemisahan data yang ketat seperti pada beberapa metode berbasis jaringan saraf [54]. Hal ini menjadikan *Prophet* sebagai metode yang efisien dan stabil untuk prediksi jangka pendek maupun jangka menengah pada data *time-series* dengan pola musiman yang kompleks. Model *Prophet* secara matematis dirumuskan sebagai berikut [54]:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \quad (6)$$

model *Prophet*, $y(t)$ menyatakan nilai observasi *time-series* pada waktu ke- t , sedangkan $g(t)$ merupakan fungsi *trend* yang memodelkan perubahan non-periodik. Komponen $s(t)$ merepresentasikan pola musiman periodik, seperti musiman mingguan dan tahunan, sementara $h(t)$ menunjukkan pengaruh hari libur atau kejadian khusus yang terjadi secara tidak teratur. Adapun ε_t merupakan komponen galat yang tidak dapat dijelaskan oleh model, di mana *Prophet* menggunakan waktu sebagai variabel regresor untuk menyesuaikan fungsi linier maupun nonlinier sebagai komponen penyusun model. Terdapat dua model *trend* utama yang digunakan dalam *Prophet*, yaitu model pertumbuhan nonlinier yang bersifat jenuh (*saturating growth*) dan model linier tersegmentasi (*piecewise linear*).

Model nonlinier pada *Prophet* umumnya dimodelkan menggunakan model pertumbuhan logistik, yang dalam bentuk paling dasar dirumuskan sebagai berikut [54]:

$$g(t) = \frac{C}{1 + \exp(-k(t - m))} \quad (7)$$

Parameter C menyatakan *carrying capacity* atau kapasitas maksimum yang dapat dicapai oleh sistem, sedangkan k merupakan laju pertumbuhan (*growth rate*) yang mengatur kecepatan perubahan nilai *trend*. Parameter m berfungsi sebagai parameter pergeseran

waktu (*offset*) yang menentukan posisi titik tengah pertumbuhan pada fungsi logistik. Ketika nilai laju pertumbuhan k mengalami perubahan, maka parameter pergeseran m juga harus disesuaikan agar hubungan antarsegmen tetap kontinu

Model pertumbuhan logistik tersegmentasi dalam *Prophet* dirumuskan sebagai berikut :

$$g(t) = \frac{C(t)}{1 + \exp\left(-(k + a(t)^T \delta)(t - (m + a(t)^T \gamma))\right)} \quad (8)$$

dengan keterangan:

- $C(t)$ adalah kapasitas maksimum yang dapat berubah terhadap waktu,
- $a(t)$ adalah variabel indikator yang menunjukkan apakah waktu t telah melewati titik perubahan (*changepoint*),
- $17t$ atis parameter penyesuaian laju pertumbuhan,
- $17t$ atis parameter penyesuaian parameter *offset*.

Perubahan laju pertumbuhan akibat adanya *changepoint* menyebabkan *trend* dimodelkan sebagai berikut :

$$g(t) = (k + a(t)^T \delta)t + (m + a(t)^T \gamma) \quad (9)$$

Pada model *trend*, parameter k menunjukkan laju pertumbuhan awal, sedangkan m merupakan parameter *offset* awal yang menentukan posisi awal *trend*. Parameter $17t$ atis penyesuaian laju pertumbuhan yang merepresentasikan perubahan kemiringan *trend* pada setiap *changepoint*. Untuk menjaga agar fungsi *trend* tetap kontinu, parameter γ ditetapkan sebesar $-s_j \delta_j$. Dalam proses pemilihan *changepoint* secara otomatis, parameter δ_j diasumsikan mengikuti distribusi Laplace(0, τ). Hal ini bertujuan untuk mengontrol kompleksitas model tren agar tidak terjadi *overfitting*.

Model *trend* linier tersegmentasi (*piecewise linear trend*) pada *Prophet* dirumuskan sebagai berikut :

$$g(t) = (k + a(t)^T \delta)t + (m + a(t)^T \gamma) \quad (10)$$

Pada fungsi *trend*, parameter k menyatakan laju pertumbuhan (*growth rate*), sedangkan m merupakan parameter pergeseran (*offset parameter*). Parameter δ berfungsi sebagai penyesuaian laju pertumbuhan yang merepresentasikan perubahan *trend* pada titik perubahan tertentu. Untuk menjaga kesinambungan fungsi *trend*, parameter γ_j ditetapkan sebesar $-s_j \delta_j$. Sementara itu, $a(t)$ merupakan statis indikator yang menunjukkan apakah waktu t telah melewati titik perubahan (*changepoint*).

Prophet menggunakan deret *Fourier* untuk memodelkan efek musiman karena pendekatan ini bersifat fleksibel dalam menangkap pola 18statis pada data *time-series*. Efek musiman direpresentasikan menggunakan persamaan berikut :

$$s(t) = \sum_{n=1}^N \left(a_n \cos \left(\frac{2\pi nt}{P} \right) + b_n \sin \left(\frac{2\pi nt}{P} \right) \right) \quad (11)$$

Pada komponen musiman, parameter P menyatakan periode statis, seperti 7 untuk musiman mingguan atau 365,25 untuk musiman tahunan, sedangkan N menunjukkan jumlah komponen *Fourier* yang digunakan. Parameter a_n dan b_n merupakan koefisien *Fourier* yang diestimasi dari data historis. Pendekatan deret *Fourier* ini memungkinkan model *Prophet* menangkap pola musiman yang kompleks dan tidak terbatas pada bentuk gelombang sederhana.

2.7.3. *NeuralProphet*

NeuralProphet merupakan sebuah model yang memprediksi data *time-series* yang menggabungkan pendekatan statistik tradisional dengan teknik *learning* yang mendalam [55]. Model ini dirancang sebagai pengembangan dari algoritma *Prophet* yang diperkenalkan oleh *Facebook (Meta)*, dengan menambahkan elemen jaringan saraf untuk meningkatkan kemampuan model dalam mengidentifikasi pola *non-linear* dan kompleks pada data *time-series*. *NeuralProphet* dirancang untuk mengatasi kekurangan model prediksi tradisional yang biasanya kurang mampu memodelkan hubungan *non-linear* dan dinamika perubahan data yang cepat [17].

NeuralProphet secara matematis memodelkan data *time series* sebagai gabungan dari beberapa komponen utama yang diatur secara aditif termasuk komponen *trend*, musiman, berbasis jaringan *neural* yang *autoregressive*, dan *error* [56]. Model ini dapat dinyatakan dalam persamaan sebagai berikut [56]:

$$y(t) = g(t) + s(t) + a(t) + \varepsilon(t) \quad (12)$$

Di mana $y(t)$ merupakan nilai aktual pada waktu ke- t , $g(t)$ merepresentasikan *trend* jangka panjang, $s(t)$ menunjukkan pola musiman, $a(t)$ merupakan komponen *autoregressive* nonlinier, dan $\varepsilon(t)$ adalah komponen kesalahan [56]. Pendekatan ini memungkinkan *NeuralProphet* menangkap dinamika data baik dalam jangka pendek maupun jangka panjang.

NeuralProphet memodelkan komponen musiman dengan pendekatan *fourier* untuk menangkap pola yang berulang dalam jangka waktu tertentu [56]. Komponen musiman tersebut dapat dinyatakan sebagai berikut [17]:

$$s(t) = \sum_{n=1}^N \left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right) \right) \quad (13)$$

Dalam hal ini, P menunjukkan periode musiman dan a_n serta b_n merupakan parameter yang dipelajari selama proses pelatihan [17]. Pendekatan ini memungkinkan *NeuralProphet* mengenali pola musiman yang kompleks, baik musiman tunggal maupun ganda

Elemen penting yang membedakan *NeuralProphet* dari *Prophet* adalah pemanfaatan jaringan saraf *autoregressive* untuk memodelkan ketergantungan waktu [17]. Elemen ini memanfaatkan nilai historis sebagai *input*, yang dapat dinyatakan dalam bentuk berikut [17]:

$$a(t) = f(y_{t-1}, y_{t-2}, \dots, y_{t-p}) \quad (14)$$

Dengan p sebagai jumlah *lag* dan $f(\dots)$ sebagai fungsi nonlinier jaringan saraf [17]. Fungsi umumnya dipresentasikan sebagai $f(x) = \sigma(W_x + b)$, di mana W merupakan matriks bobot, b adalah bias, dan $\sigma(\cdot)$ merupakan fungsi aktivasi [17]. Pendekatan ini memungkinkan *NeuralProphet* untuk memahami hubungan *nonlinier* antara nilai *time series* dengan lebih efisien dibandingkan dengan model *autoregressive* tradisional. Pendekatan ini memungkinkan *NeuralProphet* untuk memahami hubungan nonlinier antara variabel. Dalam praktiknya, *NeuralProphet* menunjukkan keunggulan dalam mengelola data *time-series* yang rumit dan dinamis, seperti harga saham yang memiliki berfluktuasi dan dipengaruhi oleh berbagai faktor eksternal.

2.8. Evaluasi Model

Evaluasi model adalah langkah penting dalam proses *machine learning* yang bertujuan untuk mengukur kinerja dan akurasi dari model yang sudah dilakukan. Dengan melakukan evaluasi, peneliti dapat memahami sejauh mana model dapat melakukan tugas yang diharapkan baik dalam Permasalahan klasifikasi, regresi, maupun jenis tugas prediktif lainnya. Evaluasi ini dilaksanakan dengan menerapkan berbagai ukuran dan teknik penilaian yang disesuaikan dengan karakteristik serta tujuan masalah yang diteliti.

Dalam penerapannya, penilaian model biasanya memakai metrik kesalahan seperti *Mean Absolute Error (MAE)*, dan *Root Mean Squared Error (RMSE)* untuk mengukur seberapa tepat hasil prediksi. Metrik tersebut menggambarkan perbedaan antara nilai prediksi dan nilai aktual, sehingga dapat digunakan untuk menilai seberapa efektif model dalam memprediksi data. Evaluasi yang dilakukan dengan benar merupakan langkah kunci dalam pengembangan model yang tepat dan sesuai dengan tujuan penelitian

Selain itu, pemilihan metrik evaluasi yang sesuai dengan konteks masalah memiliki peran penting dalam memahami kinerja model secara keseluruhan. Dengan menerapkan

metrik yang tepat, hasil evaluasi dapat menjadi landasan dalam pengambilan keputusan serta penilaian kualitas model berdasarkan *output* prediksi yang dihasilkan

Untuk mengukur kinerja algoritma prediksi dengan mengevaluasi perbandingan antara hasil prediksi dan hasil aktualnya dapat dilakukan dengan cara berikut:

1. *Mean Absolute Error (MAE)*

Mean Absolute Error (MAE) adalah salah satu ukuran evaluasi yang digunakan untuk menilai rata-rata kesalahan prediksi dengan menghitung nilai absolut dari perbedaan antara hasil prediksi dan nilai sebenarnya pada setiap data [57]. Metrik ini memberikan gambaran langsung tentang seberapa besar kesalahan prediksi tanpa mempertimbangkan arah kesalahan tersebut [58]. Nilai *MAE* yang lebih rendah menunjukkan bahwa model memiliki tingkat akurasi yang lebih baik dalam melakukan prediksi, karena perbedaan antara nilai prediksi dan nilai aktual semakin kecil. Sehingga, *MAE* sering dijadikan sebagai ukuran kinerja model untuk menilai kemampuan prediksi, terutama pada masalah regresi dan prediksi *time series*. Persamaan *MAE* adalah sebagai berikut [58]:

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

(15)

Keterangan:

MAE : *Mean Absolute Error*

n : Jumlah total data atau observasi

y_i : Nilai aktual dari observasi ke- i

$\hat{y}$: Nilai prediksi dari observasi ke- i

Σ : Penjumlahan dari kesalahan absolut pada seluruh data

2. *Mean Square Error (MSE)*

Mean Squared Error (MSE) adalah Metrik evaluasi yang digunakan untuk mengukur tingkat kesalahan prediksi dengan menghitung rata-rata kuadrat dari perbedaan antara nilai prediksi dan nilai aktual di setiap data [59]. Dengan mengkuadratkan selisih itu, *MSE* memberikan hukuman yang lebih signifikan untuk kesalahan prediksi yang memiliki nilai besar. Semakin rendah nilai *MSE* yang diperoleh, maka semakin tinggi tingkat akurasi model dalam memberikan prediksi yang mendekati nilai sebenarnya [60]. Persamaan *MSE* adalah sebagai berikut [61]:

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

(16)

Keterangan

MSE : *Mean Square Error*

n : Jumlah total data atau observasi

γ_i : Nilai aktual dari observasi ke- i

$\hat{\gamma}$: Nilai prediksi dari observasi ke- i

Σ : Penjumlahan dari kesalahan absolut pada seluruh data

3. *Mean Absolute Percentage Error (MAPE)*

Mean Absolute Percentage Error (MAPE) adalah metrik yang digunakan untuk mengevaluasi seberapa besar kesalahan dalam prediksi secara relatif dengan menghitung rata-rata perbedaan persentase antara nilai sebenarnya dan nilai yang diprediksi [62]. Metrik ini menggambarkan seberapa akurat hasil prediksi dibandingkan dengan nilai sebenarnya dalam bentuk persentase, sehingga mempermudah pemahaman tentang tingkat akurasi model. Nilai *MAPE* yang semakin rendah menunjukkan bahwa model semakin baik dalam melakukan prediksi, karena persentase perbedaan antara nilai prediksi dan nilai aktual semakin kecil. Persamaan *MAPE* adalah sebagai berikut [62]:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{\gamma_i - \hat{\gamma}_i}{\gamma_i} \right| \quad (17)$$

Keterangan:

MAPE : *Mean Absolute Percentage Error*

n : Jumlah total data atau observasi

γ_i : Nilai aktual dari observasi ke- i

$\hat{\gamma}$: Nilai prediksi dari observasi ke- i

Σ : Penjumlahan dari kesalahan absolut pada seluruh data

Nilai *MAPE* dapat diartikan ke dalam 4 kategori, yaitu [62]:

Tabel 2.1 Rentang nilai *MAPE*

Rentang <i>MAPE</i>	Arti nilai
<10%	Kemampuan model prediksi sangat baik
10 – 20%	Kemampuan model prediksi baik
20 – 50%	Kemampuan model prediksi layak
>50%	Kemampuan model prediksi buruk

4. *Root Mean Squared Error (RMSE)*

Root Mean Squared Error (RMSE) adalah sebuah Metrik evaluasi yang umum digunakan untuk mengukur tingkat kesalahan dalam model prediksi data kuantitatif[59]. Metrik ini

menghitung akar dari rata-rata kuadrat perbedaan antara nilai prediksi dan nilai aktual, sehingga dapat menunjukkan seberapa besar deviasi prediksi dibandingkan dengan data yang sebenarnya. *RMSE* juga biasanya digunakan untuk menilai seberapa dekat penyebaran data prediksi dengan garis regresi linear. Semakin rendah nilai *RMSE*, semakin baik dalam menggambarkan pola data yang diprediksi, Persamaan *RMSE* adalah sebagai berikut[59]:

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

(18)

Keterangan:

RMSE : *Root Mean Square Error*

n : Jumlah total data atau observasi

y_i : Nilai aktual dari observasi ke-*i*

$\hat{y}$: Nilai prediksi dari observasi ke-*i*

Σ : Penjumlahan dari kesalahan absolut pada seluruh data

5. *Mean Absolute Scaled Error (MASE)*

MASE (Mean Absolute Scaled Error) adalah metrik evaluasi yang mengukur akurasi Prediksi dengan cara membandingkan *MAE* model yang diuji terhadap *MAE* model *naïve* sebagai pembanding [63]. Metrik ini dirancang agar tidak bergantung pada skala data, sehingga dapat digunakan untuk membandingkan kinerja berbagai model pada beragam jenis data *time-series*. Keunggulan *MASE* dibanding metrik lain terletak pada fleksibilitasnya yang tetap andal meski data mengandung fluktuasi, tren, maupun pola musiman. Semakin kecil nilai *MASE*, semakin baik kemampuan prediksi model dan apabila nilainya di bawah satu, model tersebut terbukti menghasilkan prediksi yang lebih akurat dibandingkan model *naïve* [63]. Karena sifatnya yang robust dan mudah diinterpretasikan, *MASE* banyak dijadikan ukuran evaluasi utama dalam analisis *time-series*, terutama pada data yang berpola musiman. Persamaan *MASE* adalah sebagai berikut [63]:

$$MASE = \frac{MAE_{model}}{MAE_{naive}}$$

(19)

Keterangan:

MASE: *Mean Absolute Scaled Error*

MAE_{model}: Nilai *Mean Absolute Error* dari model prediksi yang digunakan

MAEnaive: Nilai *Mean Absolute Error* dari model *naive* sebagai pembandingan

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

Symmetric Mean Absolute Percentage Error (sMAPE) merupakan metrik evaluasi akurasi Prediksi yang dikembangkan sebagai alternatif atas keterbatasan *Mean Absolute Percentage Error (MAPE)*, khususnya pada kondisi di mana nilai aktual mendekati nol sehingga menyebabkan pembagian yang tidak terdefinisi [64]. *sMAPE* menghitung persentase kesalahan dengan membagi nilai absolut selisih antara hasil prediksi dan nilai aktual terhadap rata-rata dari keduanya, sehingga menghasilkan ukuran yang lebih seimbang dibandingkan *MAPE* [64]. Nilai *sMAPE* selalu berada dalam rentang 0% hingga 200%, di mana nilai yang lebih kecil menunjukkan tingkat akurasi prediksi yang lebih baik [64]. Meskipun pada awalnya *sMAPE* diusulkan karena dianggap tidak memihak estimasi rendah dan lebih stabil terhadap kesalahan pengukuran pada data aktual, sejumlah kajian empiris menunjukkan bahwa klaim tersebut tidak selalu terbukti dalam berbagai kondisi data [64]. Selain itu, *sMAPE* hanya bersifat simetris dalam hal pertukaran nilai prediksi dan nilai aktual, namun tidak simetris terhadap kondisi *overforecasting* dan *underforecasting* dalam jumlah yang sama, yang berarti kesalahan prediksi ke bawah dan ke atas tidak dihukum secara setara [64]. Oleh karena itu, penggunaan *sMAPE* lebih disarankan pada data dengan *Volume* tinggi dan variasi yang tidak terlalu ekstrem, di mana nilainya cenderung mendekati ekspektasi aktual, sehingga dapat memberikan gambaran akurasi yang lebih bermakna [64]. Persamaan *sMAPE* adalah sebagai berikut [64]:

(20)

$$sMAPE = \frac{1}{n} \sum_{i=1}^n \frac{|\hat{y}_i - y_i|}{(|\hat{y}_i| + |y_i|)/2} \times 100\%$$

Keterangan:

sMAPE : *Symmetric Mean Absolute Percentage Error*

n : Jumlah total data atau observasi

y_i : Nilai aktual dari observasi ke-*i*

y_i : Nilai prediksi dari observasi ke-*i*

2.9. Protokol Evaluasi

Protokol evaluasi adalah elemen penting dalam penelitian prediksi *time-series* karena menetapkan cara kinerja model diuji dan dibandingkan secara objektif [9]. Dalam konteks prediksi harga saham, pemilihan protokol evaluasi yang tepat sangat penting mengingat data bersifat berurutan, tidak *stationary*, dan sangat dipengaruhi oleh dinamika waktu. Pemanfaatan metode evaluasi yang mengabaikan aspek temporal, seperti pembagian acak atau pembagian *simple train-test*, dapat menyebabkan bias evaluasi (*look-ahead bias*) dan memberikan estimasi kinerja model yang kurang realistis [9].

Dalam penelitian-penelitian *time series* terkini, protokol evaluasi dirumuskan untuk menirukan keadaan prediksi di dunia nyata, di mana model hanya dapat mengakses data historis hingga waktu tertentu dan digunakan untuk meramalkan nilai di masa yang akan datang [65], [66], [67]. Dengan demikian, protokol evaluasi yang telah distandarisasi seperti *walk-forward validation* dan *multi-horizon forecasting* sangat disarankan dalam literatur sebagai metode yang lebih dapat diandalkan untuk menilai model prediksi *time-series*, terutama dalam konteks data keuangan dan pasar saham.

Untuk mengawasi kinerja evaluasi model prediksi dapat dilakukan dengan cara berikut:

1. *Walk-forward validation*

Walk-forward validation adalah protokol penilaian yang dirancang khusus bagi data *time-series* dengan mempertahankan urutan kronologis data [68]. Dalam pendekatan ini, model dilatih dengan data sejarah hingga waktu tertentu, lalu diuji pada periode waktu selanjutnya. Selanjutnya, jendela data pelatihan diperluas atau digeser ke depan, dan proses pelatihan serta pengujian diulang secara berulang hingga semua data digunakan [68].

Metode *walk-forward validation* lebih tepat diterapkan pada data *time-series* (*time series*) karena secara eksplisit memperhitungkan adanya ketergantungan temporal antara pengamatan pada waktu t_n dan pengamatan pada waktu selanjutnya t_{n+1} [69]. Dalam data *time-series*, nilai pada waktu tertentu tidak independen, tetapi dipengaruhi oleh nilai-nilai yang ada sebelumnya, sehingga pengujian model perlu menjaga urutan kronologis data. Pendekatan *walk-forward validation* menjamin bahwa pelatihan model hanya memanfaatkan data sampai waktu tertentu t_n , sedangkan pengujian dilaksanakan pada data selanjutnya, yaitu pada saat t_{n+1} , sehingga mencerminkan situasi prediksi yang realistis dan menghindari munculnya *look-ahead bias* [69].

Selain itu, dengan secara bertahap memperbarui jendela pelatihan dari t_n ke t_{n+1} , metode ini memungkinkan penilaian kinerja model dilakukan secara berkala pada berbagai waktu.

[69]. Hal itu sangat penting dalam konteks *forecasting* harga saham yang dinamis dan *non-stationary*, sebab perubahan pola data seiring waktu dapat memengaruhi ketepatan model.

2. *Multi-Horizon forecasting*

Multi-horizon forecasting adalah metode evaluasi di mana model prediksi diuji pada berbagai horizon waktu secara bersamaan, seperti prediksi satu langkah ke depan (jangka pendek), beberapa langkah ke depan (jangka menengah), hingga horizon yang lebih jauh. Pendekatan ini krusial karena tingkat kesulitan dan ketidakpastian ramalan cenderung bertambah seiring dengan bertambahnya jangka waktu [69]. Evaluasi *multi-horizon* memungkinkan analisis yang lebih mendalam terhadap kemampuan model dalam menangkap dinamika jangka pendek serta jangka panjang. Model yang menunjukkan kinerja baik di horizon pendek belum tentu tetap stabil pada horizon yang lebih panjang, khususnya saat data memiliki volatilitas tinggi dan perubahan mendadak [70]

2.10. *Literature Review*

Dalam penyusunan tugas akhir ini, terdapat penelitian terdahulu yang telah dilakukan oleh peneliti sebelumnya. Model, data, serta temuan yang dihasilkan dari penelitian-penelitian tersebut dimanfaatkan sebagai sumber referensi utama yang mendukung penelitian ini. Ringkasan mengenai penelitian-penelitian terkait yang telah dilakukan dapat dilihat dalam Tabel 2.2

Tabel 2.2 Literature Review

No	Referensi	Model	Judul	Area Aplikasi	Hasil	Relevansi
1.	[13]	<i>SARIMA dan LSTM</i>	<i>Predicting Stock Prices for the Next 7 Days: A Comparative Analysis of SARIMA and LSTM Models</i>	Ekonomi	model <i>SARIMA</i> lebih unggul dalam memprediksi harga saham untuk jangka pendek, yang dibuktikan dengan nilai <i>MAE</i> , <i>MSE</i> , dan <i>RMSE</i> yang lebih rendah dibandingkan <i>LSTM</i> . Namun kinerjanya menurun untuk prediksi jangka panjang, di mana <i>LSTM</i> menunjukkan hasil yang lebih baik	Penelitian ini relevan, karena menunjukkan bahwa model <i>SARIMA</i> masih memiliki daya saing dalam prediksi jangka panjang. Oleh karena itu, penelitian ini mengembangkan studi komparatif dengan memperluas model yang diuji, yaitu <i>SARIMA</i> , <i>Prophet</i> , dan <i>NeuralProphet</i> , serta menerapkan evaluasi multi-horizon untuk memperoleh pemahaman yang lebih komprehensif terhadap kinerja model dalam konteks volatilitas saham sektor kendaraan listrik.
2.	[12]	<i>Prophet, NeuralProphet, dan ARIMA</i>	<i>Forecasting Stock Prices Using Multi-Macroeconomic Regressors Based on the Facebook Prophet Model</i>	Ekonomi	Model <i>Prophet</i> merupakan model terbaik dalam memprediksi harga saham dibandingkan dengan <i>NeuralProphet</i> dan <i>ARIMA</i> . Model <i>Prophet</i>	Penelitian ini relevan dengan penelitian yang dilakukan pada saham <i>BYD</i> karena menunjukkan bahwa model <i>Prophet</i> dan <i>NeuralProphet</i> memiliki fleksibilitas dalam

					menghasilkan nilai <i>Mean Absolute Error (MAE)</i> sebesar 14,08259, yang merupakan nilai terendah di antara dua model lainnya	menangkap pola <i>trend</i> dan musiman pada data <i>time series</i> saham. Penelitian ini mengembangkan pendekatan tersebut dengan melakukan analisis komparatif terhadap <i>Prophet</i>, <i>NeuralProphet</i>, dan <i>SARIMA</i> pada saham sektor kendaraan listrik, serta mengevaluasi performa model pada berbagai horizon prediksi untuk memperoleh pemahaman yang lebih komprehensif terhadap stabilitas dan reliabilitas model.
3.	[50]	<i>ARIMA, SARIMA, dan LSTM</i>	<i>Profit Prediction Using ARIMA, SARIMA, and LSTM Models in Time series Forecasting: A Comparison</i>	Keuangan dan Bisnis	Hasil penelitian menunjukkan bahwa model <i>LSTM</i> memberikan performa prediksi terbaik dibandingkan model <i>ARIMA</i> dan <i>SARIMA</i>, terutama dalam menangkap pola <i>non-linear</i> pada data <i>time-series</i> keuntungan. Hasil penelitian menunjukkan bahwa model <i>ARIMA</i> memperoleh	Penelitian ini relevan dengan penelitian pada saham <i>BYD</i> karena menunjukkan pentingnya evaluasi komparatif antar model sebelum menentukan model terbaik. Oleh karena itu, penelitian ini mengembangkan pendekatan komparatif dengan menguji <i>SARIMA</i>, <i>Prophet</i>, dan <i>NeuralProphet</i> pada data harga saham sektor

					akurasi sebesar 93,84% (<i>MAPE</i> 6,16%), <i>SARIMA</i> sebesar 94,378% (<i>MAPE</i> 5,62%), dan <i>LSTM</i> sebesar 97,01% (<i>MAPE</i> 2,99%). Model <i>LSTM</i> memiliki performa terbaik dan paling akurat dalam memprediksi <i>profit</i> dibandingkan model <i>ARIMA</i> dan <i>SARIMA</i>	kendaraan listrik untuk memperoleh pemahaman yang lebih komprehensif mengenai stabilitas dan akurasi model dalam berbagai horizon prediksi.
4.	[53]	<i>SARIMA</i> dan <i>LSTM</i>	<i>Predictive Analytics for Demand Forecasting – A Comparison of SARIMA and LSTM in Retail SCM</i>	<i>Supply chain management (retail)</i>	Dalam penelitian tersebut, <i>LSTM</i> lebih unggul pada data dengan pola permintaan yang stabil, sedangkan <i>SARIMA</i> lebih baik pada data yang memiliki pola berulang atau fluktuasi musiman, seperti <i>cucumber</i> yang menghasilkan <i>MAPE</i> lebih rendah dibandingkan <i>LSTM</i>	Penelitian tersebut relevan dengan penelitian ini karena harga saham <i>BYD</i> sebagai perusahaan kendaraan listrik memiliki dinamika yang kompleks dan fluktuatif. Dengan mempertimbangkan kompleksitas tersebut, penelitian ini melakukan analisis komparatif terhadap model <i>SARIMA</i> , <i>Prophet</i> , dan <i>NeuralProphet</i> untuk mengevaluasi performa masing-masing model secara lebih menyeluruh pada berbagai horizon prediksi.

5.	[71]	<i>PROPHE T, LSTM, CNN, dan SARIMA</i>	<i>Time-series Forecasting of seasonal items sales using machine learning – A comparative analysis</i>	Bisnis	<i>LSTM</i> memiliki akurasi terbaik dibandingkan 3 model lainnya, ditunjukkan oleh <i>Mean Absolute Percentage Error (MAPE)</i> terendah yaitu 17.34	Penelitian ini menegaskan bahwa pemodelan <i>time series</i> pada data saham tidak dapat mengandalkan satu pendekatan tunggal, mengingat karakteristik pasar yang dinamis. Kondisi ini sejalan dengan pergerakan harga saham <i>BYD Company</i> yang cenderung <i>non-stationary</i> serta memiliki tingkat volatilitas yang tinggi. Berdasarkan kompleksitas tersebut, penelitian ini merancang kerangka analisis komparatif dengan menguji model <i>SARIMA</i>, <i>Prophet</i>, dan <i>NeuralProphet</i> guna menilai konsistensi dan ketepatan prediksi pada berbagai horizon prediksi.
----	------	--	--	--------	---	---