

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Studi ini merujuk pada penelitian sebelumnya yang telah dilakukan. Beberapa makalah sebelumnya yang menggunakan pendekatan peramalan ARIMA, ANN, dan LSTM untuk memprediksi data deret waktu dikutip karena mencakup tema-tema yang relevan dan terkait dengan penelitian ini.

Pada penelitian yang dilakukan [8] yang berjudul “*Stock Price Prediction Using ARIMA Model*”, menyajikan metode intensif membangun model ARIMA untuk prediksi nilai saham jangka pendek. Dari tiga sektor saham yang diuji, hasil error yang diperoleh pada penelitian ini dengan order ARIMA (0,1,0) menunjukkan angka 0.007, 0.018, dan 0.012. Hasil yang diperoleh memberikan potensi kekuatan model ARIMA untuk menghasilkan prediksi jangka pendek investor yang akan membantu proses investasi.

Penelitian selanjutnya yang dilakukan oleh [13] yang berjudul “Analisa Prediksi Index Harga Saham Gabungan Menggunakan Metode ARIMA”, dilakukan penelitian dengan metode ARIMA yang menggunakan beberapa parameter seperti R-Squared, Schwarz Criterion, dan Akaike Info Criterion. Hasil peramalan yang diperoleh dengan data aktual sebesar 6394.609 memiliki *dynamic forecast* sebesar 6387.551 dengan selisih - 7.05799 dan *statistic forecast* 6400.653 dengan selisih 6.043909. Selain itu, parameter untuk menguji penelitian yang diperoleh yaitu nilai R-Squared sebesar 0.0145000, Schwarz Criterion sebesar 10.83497 dan Akaike Info Criterion sebesar 10.77973. Hasil penelitian ini juga memiliki tingkat keakuratan yang baik dengan metode *statistic forecast* yang memiliki nilai MAE sebesar 39.66145 dan nilai MAPE sebesar 0.649627.

Penelitian lain dengan judul “Prediksi Harga Saham Garuda Indonesia di Tengah Pandemi *Covid-19* Menggunakan Metode ARIMA”

yang dilakukan oleh [20], dengan menggunakan data historis saham harian dari 22 April 2019 hingga 20 April 2020, juga melakukan riset untuk memperkirakan harga saham Garuda Indonesia. Tujuannya adalah untuk memprediksi harga penutupan saham dari 21 April 2020 hingga 13 Juli 2020. Hasil diperoleh nilai RMSE terkecil sebesar 38.03 pada order ARIMA (3, 1, 2) yang model tersebut kemudian digunakan untuk peramalan harga penutupan saham Garuda Indonesia. Hasil peramalan dengan model tersebut, harga penutupan saham Garuda Indonesia cenderung mengalami penurunan.

Pada penelitian yang dilakukan oleh [17] dengan judul “Peramalan Harga Saham Menggunakan Jaringan Syaraf Tiruan Secara Supervised Learning dengan Algoritma Backpropagation”. Penelitian ini menggunakan ANN sebagai metode untuk melatih model prediksi pada harga saham. Hasil nilai akurasi yang didapat pada saham Astra Gaphia memiliki nilai yang cukup tinggi sebesar 98,28%. Hasil akurasi pada saham Indofood Sukses juga memiliki nilai yang cukup tinggi sebesar 99,61%. Oleh karena itu, jika dibandingkan dengan data sebenarnya, nilai akurasi prediksi memanfaatkan pendekatan ANN memiliki tingkat akurasi di atas 98%.

Penelitian lainnya yang dilakukan oleh [23] dengan judul “Analisis Perbandingan Hasil Peramalan Harga Saham Menggunakan Model Autoregressive Integrated Moving Average dan Long Short Term Memory” yaitu membandingkan antara 2 metode ARIMA dan LSTM. Hasil nilai akurasi setiap model terbaik pada saham tertentu yang didapat pada model LSTM pada saham BBKA dengan nilai MAPE sebesar 0,864%. Sedangkan ARIMA (1,1,0) di saham BBTN yang memiliki nilai model paling akurat dengan nilai MAPE sebesar 1,875%.

2.2. Saham

Saham masih menjadi instrumen investasi yang paling banyak diminati oleh masyarakat pada saat ini di bursa efek. Dalam pasar modal, saham merupakan salah satu produk yang diperjualbelikan dan berbentuk sertifikat yang digunakan untuk membuktikan kepemilikan suatu

perusahaan. Jika saham yang dimiliki oleh seseorang di perusahaan tersebut semakin banyak, maka jumlah uang yang diberikan kepada perusahaan tersebut semakin besar [4].

Untuk menurunkan risiko pembelian dan penjualan saham, investor harus memeriksa data saham, yang berupa data deret waktu dengan rutin. Pergerakan harga saham dipengaruhi oleh beberapa faktor seperti ramai atau tidaknya permintaan dan penawaran pada pasar. Harga saham sering kali akan turun jika penawaran penjual melebihi permintaan pembeli. Di sisi lain, harga saham sering kali akan naik jika permintaan pembeli melebihi penawaran penjual. Karena fluktuasi harga saham cenderung berkembang seiring waktu dan membentuk pola tertentu, tidak ada saham yang terus-menerus tumbuh atau menurun. Dalam jangka waktu tertentu, pergerakan harga saham umumnya membentuk pola yang khas.

2.3. Peramalan (*Forecasting*)

Peramalan merupakan metode atau pendekatan yang berguna untuk memprediksi suatu nilai atau peristiwa yang dapat terjadi pada waktu kedepan dengan mengolah data dari informasi masa lalu. Untuk perencanaan jangka pendek, menengah, dan panjang suatu perusahaan, peramalan sendiri dapat berfungsi sebagai referensi atau pedoman yang berguna [3]. Tiga kategori peramalan dapat dibedakan berdasarkan periode waktu berikut:

1. Prediksi jangka pendek, umumnya berjarak antara 1 hingga 5 minggu. Peramalan ini berguna dalam keputusan kontrol yang bersifat jangka pendek seperti pengaturan lembur dan penjadwalan kerja.
2. Prediksi jangka menengah, umumnya berjarak antara 1 sampai 24 bulan. Prediksi menengah lebih tertarget yang berbeda dengan prediksi jangka panjang. Pada jangka menengah, peramalan berguna untuk menentukan aliran kas, perencanaan produksi, dan penentuan anggaran.
3. Prediksi jangka panjang biasanya berlangsung selama dua hingga sepuluh tahun. Peramalan membantu perencanaan sumber daya dan produk dalam jangka panjang..

2.4. *Autoregressive Integrated Moving Average (ARIMA)*

Metode *Autoregressive Integrated Moving Average* atau dapat disingkat ARIMA, merupakan teknik peramalan untuk model deret waktu yang diciptakan George Box dan Gwilym Jenkins. Teknik ini dilakukan berdasarkan perilaku variabel atau fenomena musiman yang diamati yang memiliki periode relatif sama. ARIMA mendapatkan luaran prakiraan jangka pendek yang akurat dengan memanfaatkan data masa lampau dan data saat ini dari seluruh data yang diambil. Bila terdapat hubungan statistik antara pengamatan dari deret waktu yang saling terkait (dependen), ARIMA dapat diterapkan. Namun, dalam peramalan jangka panjang, akurasi peramalannya seringkali kurang memuaskan dan cenderung stabil [14].

ARIMA terdapat 3 komponen yaitu AR, I, dan MA. AR merupakan keterkaitan regresi antar variabel yang berubah berdasarkan nilai sebelumnya. I mengacu pada pemodelan perbedaan antar observasi sehingga deret waktu dapat dianggap stasioner. MA merupakan keterkaitan antar observasi dan sisa kesalahan (sisa error) dari model moving average yang diterapkan pada observasi sebelumnya. Pemodelan deret waktu bergantung pada proses pembuatan deret tersebut menjadi stasioner dengan melakukan diferensiasi. Diferensiasi dilambangkan dengan (d) dan dilakukan 1 atau 2 kali sehingga diperoleh data yang stasioner. Parameter model autoregression dilambangkan dengan (p) dan parameter model moving average sebagai (q), sehingga dapat ditulis sebagai model ARIMA (p, d, q) dengan persamaan umum sebagai berikut [26]:

$$Z_t = \mu + (1 + \phi_1)Z_{t-1} + (\phi_2 + \phi_1)Z_{t-2} + \dots + (\phi_p + \phi_{p-1})Z_{t-p} \\ - \phi_p Z_{t-p-1} + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q}$$

Keterangan:

Z_t = data waktu t, t = 1, 2, 3, ..., n

μ = konstanta

p = order (AR)

- d = order differensiasi
 q = order (MA)
 ϕ_p = koefisien AR ke-p
 θ_q = koefisien MA ke-q
 Z_{t-p} = variabel bebas saat t-p
 a_{t-q} = sisaan pada saat t-q

2.4.1. Model *Autoregressive* (AR)

Model *Autoregressive* yang menyatakan selang waktu variabel dependen sebagai variabel independen, menjelaskan bagaimana variabel dependen pada periode masa lalu memengaruhi variabel dependen di masa mendatang [20]. Pada model *Autoregressive*, dilakukan peramalan variabel yang dipilih menggunakan kombinasi linear dari nilai masa lalu dari variabel tersebut. Syarat dari *autoregression* yaitu mengindikasikan kejadian regresi dari variabel terhadap variabel itu sendiri. Model *Autoregressive* memiliki ordo atau parameter (p), sehingga dapat ditulis sebagai model AR(p) dengan persamaan sebagai berikut [13]:

$$Y_t = \mu + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + \varepsilon_t$$

Keterangan:

- Y_t = variabel tidak bebas
 μ = konstanta
 ϕ_p = koefisien parameter AR ke-p
 Y_{t-p} = variabel bebas saat t-p
 ε_t = sisaan saat ke-t

2.4.2. Model *Moving Average* (MA)

Pada model *Moving Average*, keterkaitan yang saling bergantung antara output observasi dengan output kesalahan (error value) secara berurutan pada masa sebelumnya dinyatakan dengan prosedur yang dilakukan pada rata-rata bergerak orde (q) [20]. Model *Moving Average* memiliki ordo atau parameter (q), sehingga dapat ditulis sebagai model MA(q) dengan persamaan sebagai berikut [13]:

$$Y_t = \mu - \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_p \varepsilon_{t-q}$$

Keterangan:

Y_t = variabel tidak bebas

μ = konstanta

θ_q = koefisien parameter MA ke-q

ε_{t-q} = variabel bebas saat t-q

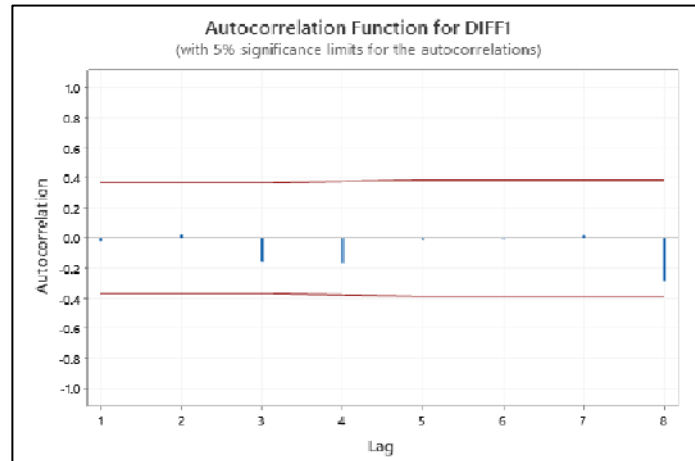
ε_t = sisaan saat ke-t

2.5. *Autocorrelation Function* (ACF) dan *Partial Autocorrelation Function* (PACF)

Keluaran korelogram ACF dan PACF dapat digunakan untuk mendapatkan parameter model AR(p) dan MA(q) dalam teknik ARIMA (p, d, q). Dengan melihat plot PACF dan ACF, parameter model ARIMA dapat ditentukan untuk menunjang penelitian. Contoh analisis yang dapat dilakukan untuk menentukan parameter ARIMA sebagai berikut.

2.5.1. Parameter Nilai AR(p)

Karakteristik yang terjadi pada plot ACF dicontohkan pada gambar 2.1.



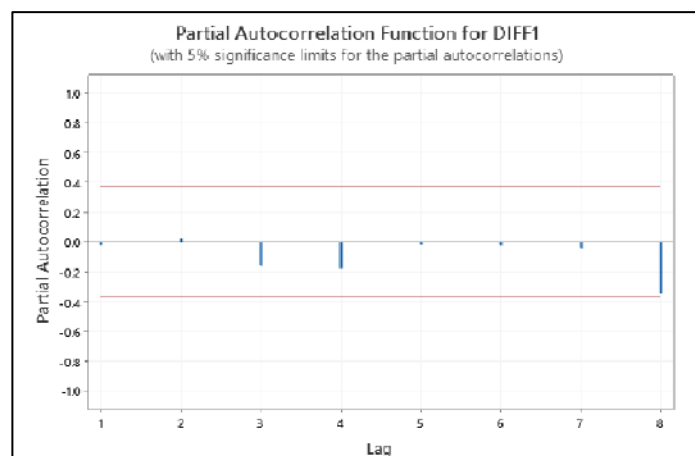
Gambar 2. 1 Contoh plot ACF

(Sumber: (Yulianti & Arliani, 2022))

Data tersebut stasioner pada diferensiasi orde pertama, menurut analisis plot deret waktu pada fungsi ACF. Proses diferensiasi pertama dapat dilihat bahwa tidak ada nilai yang menyentuh garis merah. Ini membuktikan bahwa tidak ada lag dari nilai *autocorrelation* yang diperoleh [26].

2.5.2. Parameter Nilai MA(q)

Karakteristik yang terjadi pada plot PACF dicontohkan pada gambar 2.2.



Gambar 2. 2 Contoh plot PACF

(Sumber: (Yulianti & Arliani, 2022))

Data tersebut stasioner dalam diferensiasi orde pertama, menurut analisis plot deret waktu pada fungsi PACF. Proses diferensiasi pertama dapat dilihat bahwa tidak ada nilai yang menyentuh garis merah. Ini membuktikan bahwa tidak ada lag dari nilai *partial autocorrelation* yang diperoleh [26].

2.6. Uji Stasioneritas menggunakan ADF Test (*Augmented Dickey-Fuller Test*)

Terdapat beberapa cara untuk menguji stasioneritas data dalam *time series*. Untuk menganalisis data dalam *time series*, diperlukan kestabilan data pada model ARIMA. Kumpulan data deret waktu diyakini stasioner jika rerata, varian, dan autovariannya (dalam variasi lag) konstan selama pembentukannya. Model deret waktu lebih stabil ketika data stasioner digunakan [19]. Uji akar unit merupakan teknik yang dapat digunakan untuk menilai stasioneritas data. Pengujian tersebut berkembang dengan sebutan *Augmented Dickey-Fuller Test*. Uji stasioneritas menggunakan *Augmented Dickey-Fuller Test* (ADF test) merupakan uji akar unit pada model AR yang digunakan untuk menganalisis kestasioneran data. Uji tersebut akan mengoreksi korelasi pada orde yang lebih tinggi secara parametrik dengan menggunakan jumlah lag yang lebih tinggi pada sisi kanan persamaan.

ADF test memiliki tiga model persamaan yang dapat digunakan untuk menguji *unit root*, sebagai berikut [19]:

$$y_t = a_1 y_{t-1} + \varepsilon_t$$

$$y_t = a_0 + a_1 y_{t-1} + \varepsilon_t$$

$$y_t = a_0 + a_1 y_{t-1} + a_2 t + \varepsilon_t$$

Parameter yang menjadi acuan dalam model tersebut adalah a_1 . Jika $a_1 = 1$, maka y_t berarti tidak stasioner. Jika $|a_1| < 1$, maka y_t berarti stasioner. Kemudian dapat dilakukan pengujian nilai a_1 melalui pengujian hipotesis seperti di bawah [19]:

$$H_0: a_1 = 1$$

$$H_0: |a_1| < 1$$

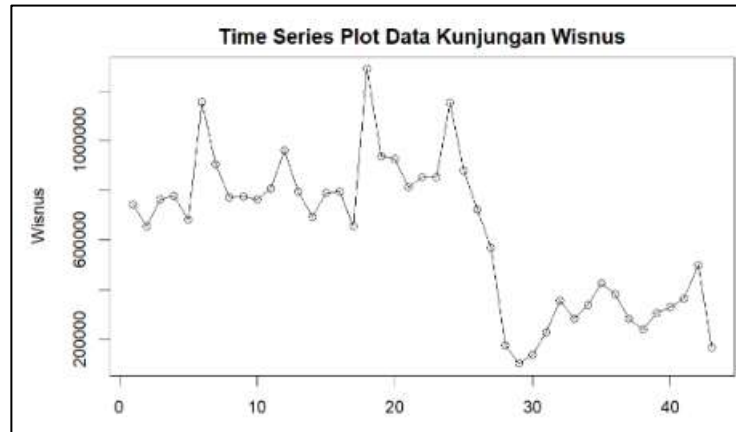
Selanjutnya dapat dilakukan pengujian menggunakan statistik t untuk menentukan variabel y_t mempunyai *unit root* atau tidak.

Pengambilan keputusan dapat dilakukan dengan 2 kriteria. Ketika ADF *test* memiliki kondisi *statistic test* < *critical value test* (*critical value* $\alpha = 5\%$) maka H_0 ditolak. Ketika ADF *test* memiliki kondisi *statistic test* > *critical value test* (*critical value* $\alpha = 5\%$) maka H_0 diterima.

2.7. Diferensiasi

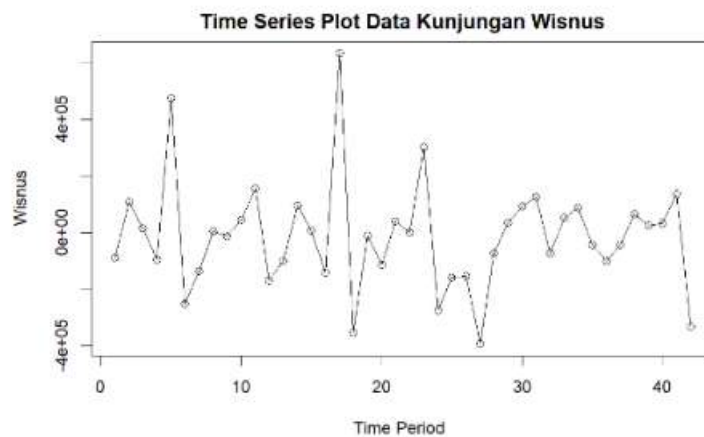
Proses diferensiasi merupakan proses yang digunakan dengan tujuan untuk menghilangkan ketergantungan pada *time series*. Ketergantungan ini termasuk pada tren dan musim pada data [16]. Diferensiasi juga berguna untuk menyetarakan mean dari *time series* dengan menghilangkan perbedaan pada level *time series*.

Analisis deret waktu seringkali dilakukan memakai dataset yang stasioner. Prakteknya seringkali ditemukan dataset yang tidak stasioner. Untuk mendapatkan data yang stasioner agar dapat dianalisis, teknik diferensiasi diperlukan untuk membuat model *time series*. Seringkali, praktik diferensiasi tidak hanya dilakukan sekali saja melainkan dua hingga tiga kali sesuai kebutuhan untuk menghilangkan variasi data yang beragam agar data *time series* yang didapat telah stasioner. Untuk mengetahui perbedaan data sebelum stasioner dan sesudah stasioner pada data dari *time series*, data disajikan gambar 2.3 (sebelum stasioner) dan gambar 2.4 (sesudah stasioner).



Gambar 2. 3 Contoh data aktual data deret waktu

(Sumber: (Riestiansyah et al., 2022))



Gambar 2. 4 Contoh data deret waktu setelah differensiasi

(Sumber: (Riestiansyah et al., 2022))

Dapat terlihat, data aktual (data asli) menunjukkan bahwa data memiliki tren yang tidak stasioner dan penyebarannya cenderung tidak rata. Ketika dilakukan differensiasi, data sudah terlihat stasioner dan penyebarannya cenderung rata.

2.8. ANN (*Artificial Neural Network*)

ANN merupakan teknik yang mampu digunakan dalam berbagai pola nonlinier dalam data. Hal tersebut dikarenakan ANN merupakan sebuah kecerdasan buatan yang dapat meniru sifat jaringan syaraf pada manusia melalui operasi matematika. ANN terdiri dari beberapa lapisan

buatan yang membentuk suatu arsitektur-arsitektur tertentu. Komponen lapisan dan neuron pada input, tersembunyi, dan output disusun dalam arsitektur dan digabungkan dengan bobot, fungsi pembelajaran, dan fungsi aktivasi [7]. Neuron pada satu lapisan terhubung dalam lapisan lainnya yang berdekatan. Maka dari itu arsitektur yang akan digunakan yaitu MLP (*Multi Layer Perceptron*) yang merupakan jaringan interkoneksi antar neuron yang dapat menyelesaikan masalah regresi, non linier dengan fungsi terdiferensiasi [25]. Masalah yang lebih rumit diselesaikan melalui jaringan multilayer [31]. Berikut ini adalah elemen-elemen yang membentuk ANN (MLP):

1. *Input Layer* (Lapisan Input)

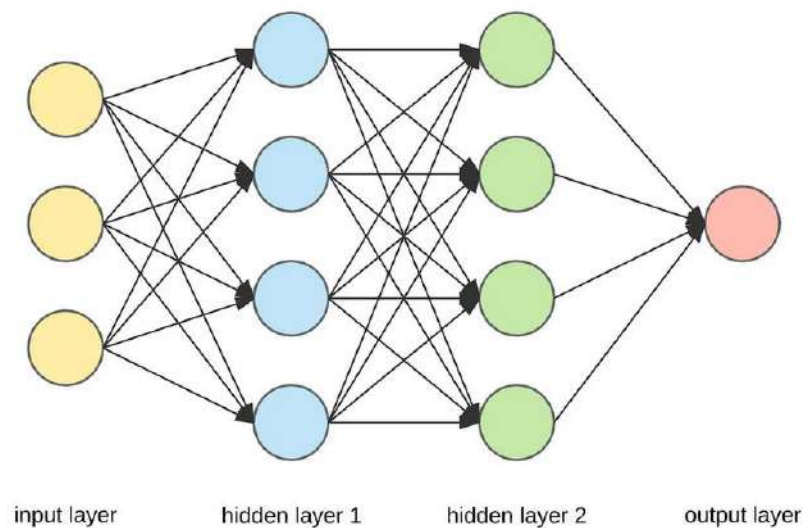
Input Layer merupakan data masukan yang data masukan tersebut kemudian dilakukan proses data pada ANN. Setiap neuron dalam lapisan ini menerima satu fitur input dari data yang diberikan

2. *Hidden Layer* (Lapisan Tersembunyi)

Lapisan input dan lapisan output dipisahkan oleh lapisan ini. Bergantung pada seberapa kompleks jaringannya, ANN mungkin menggunakan satu bisa lebih lapisan tersembunyi. Neuron dari lapisan ini menerima sinyal dari neuron pada lapisan sebelumnya, melakukan perhitungan tertentu seperti fungsi aktivasi dan meneruskan hasil perhitungan ke lapisan berikutnya

3. *Output Layer* (Lapisan Output)

Produk akhir akan dihasilkan oleh lapisan akhir ini. Hanya satu nilai keluaran dari neuron sebelumnya yang disediakan oleh setiap neuron di lapisan keluaran.



Gambar 2. 5 Struktur arsitektur ANN (MLP)

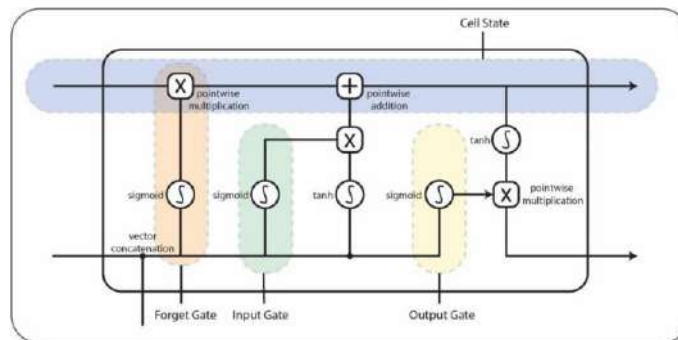
(sumber: <https://towardsdatascience.com/applied-deep-learning-part-1-artificial-neural-networks-d7834f67a4f6>)

Pada Gambar 2.5, dijelaskan bahwa bagian yang berwarna kuning merupakan *input layer* dari ANN. Lapisan tersebut berisi tentang data yang telah dimiliki oleh peneliti. Bagian yang berwarna biru dan hijau merupakan *hidden layer* yang berfungsi untuk memproses data inputan pada *input layer*. Output, yang merupakan hasil pemrosesan data, adalah bagian berwarna merah. Setiap tahap akan melewati garis penghubung yang akan disesuaikan dengan nilai bobot dan bias tertentu.

2.9. LSTM

LSTM adalah variasi dari RNN yang mampu belajar dari data dalam jangka Panjang untuk digunakan dalam peramalan[27]. Metode tersebut diperkenalkan oleh Hochreiter & Schmidhuber pada tahun 1997 kemudian dikembangkan oleh banyak orang pada penelitian-penelitian terbaru. LSTM dibuat untuk memberikan solusi terhadap kendala memori yang ada pada RNN. Memasukkan kontrol non-linier yang bergantung pada data ke dalam sel RNN sangat penting untuk arsitektur LSTM [30]. Setiap siklus dalam LSTM terdiri dari kumpulan aliran data yang diproses sebelumnya. LSTM menyimpan memori sebelumnya pada bagian atas model yang bertindak

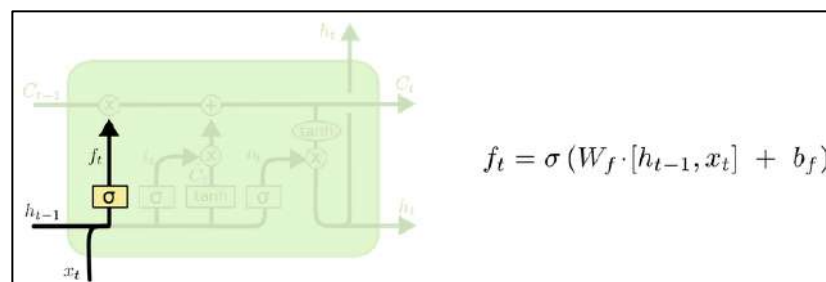
sebagai status memori [28]. Memori sebelumnya nanti akan tergantikan atau tertimpa oleh memori baru.



Gambar 2. 6 Arsitektur LSTM

2.9.1 Forget Gate

Forget gate bertugas untuk menentukan sebuah informasi harus dihapus atau tidak dari *cell state*. Output yang dihasilkan yaitu berupa angka 0 dan 1. Nilai 0 berarti output tidak diperlukan dan harus dihapus, sedangkan nilai 1 berarti output dapat digunakan dan tetap disimpan.



Gambar 2. 7 Forget gate

(sumber: <https://medium.com/bina-nusantara-it-division/lstm-long-short-term-memory-d29779e2ebf8>)

Keterangan:

f_t : Forget gate

σ : Fungsi sigmoid

W_f : Bobot input waktu ke t

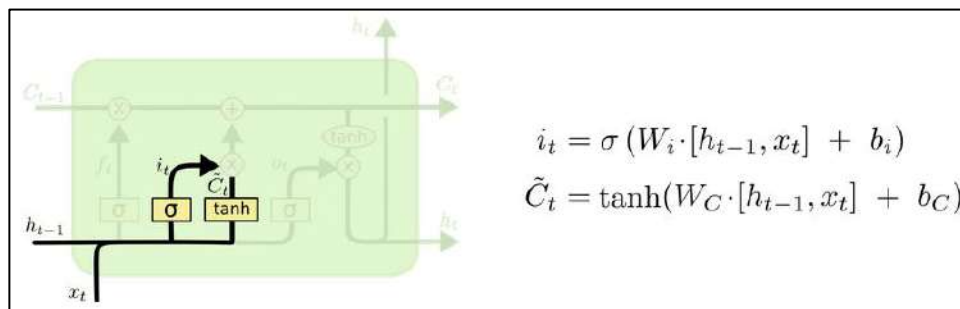
h_{t-1} : Output waktu t-1

x_t : Input waktu ke t

b_f : Bias forget gate

2.9.2 Input Gate

Input gate bertugas untuk menambahkan informasi yang berguna ke dalam *cell state*. Informasi diolah menggunakan fungsi sigmoid dan memilah nilai-nilai yang akan diingat dengan inputan h_{t-1} dan x_t . Kemudian diolah menggunakan fungsi tanh dan menghasilkan vector output yang bernilai dari -1 hingga +1.



Gambar 2. 8 Input gate

(sumber: <https://medium.com/bina-nusantara-it-division/lstm-long-short-term-memory-d29779e2ebf8>)

Keterangan:

i_t : Input gate

σ : Fungsi sigmoid

W_i : Bobot input waktu ke t

h_{t-1} : Output waktu ke t-1

x_t : Input waktu ke t

b_i : Bias input gate

$\tilde{C}_t$: Kandidat cell state

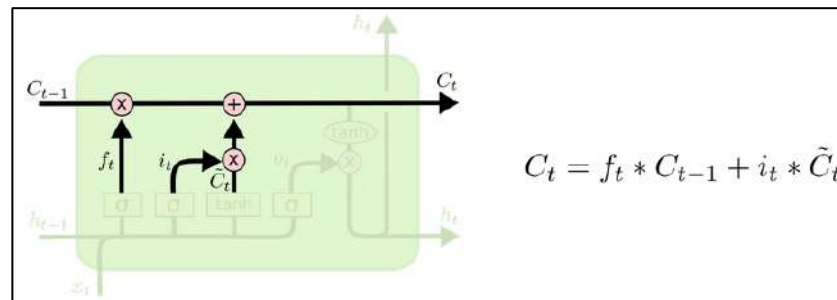
$\tanh$: Fungsi hyperbolic tangent

w_c : Pembobotan nilai input cell ke c

b_c : Bias pada cell ke c

2.9.3 Cell State (Memory Cell)

Cell state bertugas sebagai memori atau ingatan pada layer LSTM. Nilai *cell state* pada tahap ini mengalami pembaruan dari C_{t-1} menjadi C_t . tahap ini memproses nilai dari state sebelumnya. Proses ini bekerja dengan mengalikan nilai *forget gate* (f_t) dengan nilai C_{t-1} . Kemudian hasilnya ditambahkan dengan hasil perkalian dari nilai *Input gate* (i_t) dengan C_t . Hasil tersebut akan menjadi nilai terbaru dari *cell state* (C_t) dan dapat digunakan pada layer selanjutnya dan penentuan nilai *output gate*.



Gambar 2. 9 Cell state

(sumber: <https://medium.com/bina-nusantara-it-division/lstm-long-short-term-memory-d29779e2ebf8>)

Keterangan:

C_t : Nilai yang ada pada memori *cell state*

f_t : nilai pada *forget gate*

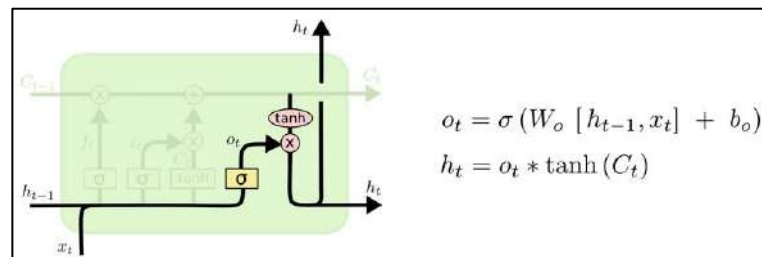
C_{t-1} : Nilai pada memori *cell state* pada cell sebelumnya

i_t : Nilai yang didapat dari *input gate*

$\tilde{C}_t$: Nilai kandidat pada memori *cell state*

2.9.4 Output Gate

Output gate bertugas untuk menentukan hasil luaran yang bergantung pada nilai *cell state* dan nilai input. *Output gate* akan melewati proses penyaringan menggunakan fungsi sigmoid untuk menentukan nilai yang akan digunakan untuk menjadi output akhir. Kemudian, *cell state* akan melewati fungsi *tanh* yang akan mengubah nilai ke dalam rentang -1 dan 1. Hasilnya akan dikalikan dengan nilai yang melewati fungsi sigmoid tadi.



Gambar 2. 10 Output gate

(sumber: <https://medium.com/bina-nusantara-it-division/lstm-long-short-term-memory-d29779e2ebf8>)

Keterangan:

o_t : Output gate

σ : Fungsi sigmoid

W_o : Bobot nilai input pada waktu ke t

h_{t-1} : Output waktu ke t-1

x_t : Input waktu ke t

b_o : Bias *output gate*

h_t : Hasil akhir perhitungan *output*

$\tanh$: Fungsi *hyperbolic tangent*

C_t : Nilai baru memori *cell state*

2.10. Fungsi Aktivasi

Fungsi aktivasi dapat digunakan untuk menjamin koneksi antara simpul masukan dan keluaran jaringan saraf tiruan (ANN) [18]. Menentukan tingkat nonlinieritas dengan fungsi aktivasi adalah tujuan utama saat menggunakan ANN dan LSTM. Saat menerapkan fungsi aktivasi di ANN dan LSTM, sejumlah fungsi dapat digunakan, termasuk:

1. Fungsi Sigmoid

$$f(x) = (1 + e^{-x})^{-1}$$

Keterangan:

e : bilangan euler

x : data

2. Fungsi Tan Hiperbolik

$$f(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$$

Keterangan:

e : bilangan euler

x : data

3. Fungsi Sinus atau Cosinus

$$f(x) = \sin(x)$$

$$f(x) = \cos(x)$$

Keterangan:

x : data

4. Fungsi Linear

$$f(x) = x$$

Keterangan:

x : data

5. Fungsi ReLU

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

Keterangan:

x : data

Fungsi aktivasi sigmoid pada *nodes output* paling cocok digunakan pada kasus klasifikasi. Namun, penggunaan fungsi aktivasi linear disarankan digunakan pada data *time series* [18].

2.11 Moving Average

Teknik peramalan yang disebut *Moving Average* (MA) memperkirakan nilai yang akan terjadi pada periode mendatang dengan menentukan nilai rata-rata dalam periode tertentu (Heizer & Render, 2015). Jika tidak ada tren atau elemen musiman dalam data historis, MA digunakan. MA sering digunakan untuk memastikan tren kumpulan data deret waktu (Ananda, 2023). Nilai MA dapat diketahui dengan merata-ratakan gabungan nilai data pada data deret waktu. Nilai MA dapat dicari dengan menentukan periode (t) terlebih dahulu. Setelah itu, MA pada periode (t) dapat dihitung nilai rata-ratanya. Hasilnya akan digunakan untuk ramalan pada periode kedepan. Pada MA, setiap data baru akan digunakan lagi dan tidak lagi menggunakan data lama karena jumlah periode selalu konstan (Nurlifa & Kusumadewi, 2017).

MA dapat dituliskan sebagai berikut (Maricar, 2019):

Keterangan:

$$MA = \frac{X_{t-1} + X_{t-2} + \dots + X_{t-n}}{n}$$

MA : Nilai moving average

X_{t-1} : Nilai periode sebelum periode yang akan dicari nilainya

n : Jumlah pada periode tertentu

2.12 Moving Average Convergence Divergence (MACD)

Indikator pengikut tren yang disebut MACD digunakan untuk melihat pergeseran tren. (Ong, 2017). MACD terdiri dari dua periode yaitu *slow periode* dan *fast period*. MACD memiliki dua garis yaitu *signal line* dan *MACD line*. *Signal line* biasanya memiliki nilai 9 dan berwarna merah dan *MACD line* biasanya memiliki nilai EMA 26 – EMA 12 dan memiliki warna biru. MACD mengeluarkan sinyal beli ketika *MACD line* memotong keatas *signal line*. MACD mengeluarkan sinyal jual ketika *MACD line* memotong kebawah *signal line* (Ong, 2017). Area di atas dan di bawah garis 0 merupakan dua bagian dari area MACD. Saat pergerakan MACD berada di wilayah negatif, ada tren penurunan yang kuat; saat berada di wilayah positif, ada tren kenaikan yang kuat.

MACD mendapat beberapa pengembangan namun masih terdapat masalah kelambatan (lag) pada indikatornya, sehingga banyak mendapat kritikan dalam kegagalannya merepresentasikan kondisi pasar. Maka dari itu, sekarang ini MACD hanya digunakan untuk *trend following* atau sebagai konfirmasi dalam melakukan aktivitas trading.

2.13 Relative Strength Index (RSI)

RSI adalah indikator yang dipakai untuk merepresentasikan keadaan jenuh jual dan jenuh beli (Ong, 2017). RSI berguna untuk membantu investor untuk melihat kondisi harga apakah pasar mengalami jenuh jual atau jenuh beli (Nor & Wickremasinghe, 2014). Cara kerja RSI yaitu membandingkan nilai kenaikan dan penurunan saat ini untuk mendapatkan momentum harga pada pergerakan harga seperti saham, crypto, dan forex. Terdapat dua area ekstrim pada RSI. Ketika nilai RSI menunjukkan nilai diatas 70 maka area tersebut dapat dikatakan area ekstrim atas. Ketika nilai RSI berada pada area tersebut, pasar mengalami kondisi jenuh beli. Ketika nilai RSI menunjukkan nilai dibawah 30 maka area tersebut dapat dikatakan area ekstrim bawah. Ketika nilai RSI berada pada area tersebut, pasar

mengalami kondisi jenuh jenuh jual. RSI juga dapat merepresentasikan sinyal jual dan beli dengan melihat nilai RSI berada pada area atas atau bawah. Sinyal jual ditentukan ketika nilai RSI berada diatas 70 dan sinyal beli ditentukan ketika nilai RSI berada dibawah 30.

2.14 Perbedaan Tujuan setiap metode

Penelitian ini mengacu pada perbedaan tujuan dari setiap metode sebelum penelitian ini dilakukan. Sejumlah penelitian sebelumnya yang menggunakan metodologi Moving Average, MACD, RSI, ARIMA, ANN, dan LSTM dikutip.

Tabel 2. 1 Perbedaan tujuan setiap metode

Metode	Peneliti	Tujuan
ARIMA	A. L. Putra, & A. K. Kurniawati, 2021	ARIMA memberikan ramalan jangka pendek yang akurat dengan memanfaatkan data masa lalu dan data sekarang.
ANN	A. I. Sakti, dkk., 2024	Analisis data deret waktu dalam peramalan adalah masalah yang dipecahkan dengan pendekatan ANN. Pendekatan ini dianggap menarik, rumit, dan populer.
LSTM	S. S. Abubaker, & S. R. Farid, 2022	LSTM mampu mempelajari data dalam jangka Panjang untuk digunakan dalam peramalan.
Moving Average	Ananda, 2023	Untuk memastikan pola data deret waktu, MA sering digunakan.
MACD	Ong, 2017	MACD mengeluarkan sinyal beli ketika MACD <i>line</i> memotong keatas <i>signal line</i> . MACD mengeluarkan sinyal jual ketika MACD <i>line</i> memotong kebawah <i>signal line</i> .

RSI	Ong, 2017	Keadaan jenuh jual dan jenuh beli dapat direpresentasikan oleh RSI.
-----	-----------	---

Dari tabel 2.1 menunjukkan bahwa ARIMA, ANN, dan LSTM memiliki tujuan yang sama yaitu untuk peramalan deret waktu. Sedangkan moving average berguna untuk mengetahui arah tren, MACD bertujuan untuk mengetahui sinyal beli dan jual, dan RSI bertujuan untuk mengetahui kondisi overbought/oversold. Sehingga untuk mencapai tujuan penelitian yang sama, ARIMA, ANN, dan LSTM sangat cocok dipakai pada penelitian ini.

2.15. Normalisasi Data

Normalisasi data adalah teknik yang digunakan pada tahap *pre-processing* data dengan mengubah rentang nilai yang ada menjadi rentang nilai yang baru. Normalisasi ini dapat membantu untuk membuat nilai menjadi normal dengan melakukan standarisasi nilai yang memiliki interval 0 sampai dengan 1 (Amalia, 2016). Berikut merupakan metode yang digunakan untuk proses normalisasi:

2.15.1 Min-Max Normalization

Untuk membuat data lebih mudah diproses tanpa kehilangan isinya, normalisasi min-max adalah teknik normalisasi yang menggunakan transformasi linear data asli dari satu rentang ke rentang nilai baru. [29]. *Min-max normalization* dapat dihitung dengan persamaan sebagai berikut:

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

Keterangan:

x : Data asli

x' : Hasil Normalisasi

x_{max} : Nilai tertinggi pada data

x_{min} : Nilai terendah pada data

2.16. Evaluasi Nilai Error

Dalam pembuatan model peramalan, evaluasi sangat diperlukan yang berguna untuk mendapatkan nilai keakuratan pada hasil peramalan. Meskipun keakuratan yang mutlak tidak dapat tercapai, hasil peramalan tersebut dapat dijadikan sebagai pertimbangan dan acuan pada pengambilan keputusan di masa depan.

Pengukurann yang dapat dipakai untuk menguji keakuratan yaitu *scale-dependent error* (pengukuran skala tergantung) dan *percentage error* (pengukuran persentase). Pengukuran skala tergantung umumnya dilakukan dengan menggunakan MAE dan RMSE. Sedangkan pengukuran skala persentase umumnya dilakukan dengan menggunakan MAPE. Pengujian tingkat akurasi yang dipakai penelitian ini yaitu MAPE dan RMSE. Berikut adalah beberapa evaluasi yang akan digunakan:

2.16.1. Mean Absolute Percentage Error (MAPE)

Teknik untuk menghitung rata-rata perbedaan antara data aktual dan data ramal adalah MAPE. Saat menganalisis data perkiraan, MAPE digunakan untuk menentukan seberapa akurat nilai aktual dan yang diantisipasi. Tingkat keakuratan hasil peramalan dapat diketahui ketika nilai presentase kesalahan pada nilai MAPE semakin rendah yang berarti semakin akurat hasil peramalan tersebut. Rumus untuk menghitung hasil MAPE ditulis dengan persamaan [11]:

$$MAPE = \frac{\sum_{t=1}^n \left| \frac{a_t - f_t}{a_t} \right|}{n} \times 100\%$$

$MAPE$ = nilai *mean absolute percentage error*

a_t = nilai data aktual periode ke-t

f_t = nilai data hasil prediksi periode ke-t

n = jumlah data

2.16.2. *Root Mean Square Error (RMSE)*

Salah satu cara untuk mengukur apakah model dapat memprediksi data kuantitatif secara akurat adalah dengan menggunakan RMSE. Persamaan RMSE yang dipakai yaitu [11]:

$$RMSE = \sqrt{\frac{\sum (a_t - f_t)^2}{n}}$$

Keterangan:

$RMSE$ = nilai *root mean square error*

a_t = nilai data aktual periode ke-t

f_t = nilai data hasil prediksi periode ke-t

n = jumlah data

Halaman ini sengaja dikosongkan