

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terdahulu**

Pada bagian ini dilakukan tinjauan terhadap penelitian-penelitian sebelumnya untuk memperdalam pemahaman dalam penelitian ini. Penelitian ini merujuk pada studi-studi sebelumnya untuk memperkuat dasar teori dan metodologi yang digunakan, sekaligus memastikan orisinalitas penelitian. Setelah meninjau penelitian terdahulu, peneliti dapat memahami perkembangan topik serta mengidentifikasi kekurangan atau celah yang belum ditangani. Berikut adalah beberapa penelitian terdahulu yang relevan dengan penggunaan metode LSTM:

Penelitian yang dilakukan oleh Hanafiah dkk., (2023) memfokuskan pada prediksi harga saham menggunakan metode LSTM pada data saham BBNI. Pengujian dilakukan dengan dua skenario jumlah *epoch*, yaitu 10 dan 20, dengan *batch size* 32. Hasil terbaik pada *epoch* 20 didapatkan nilai MAE sebesar 0.0150 dan MAPE sebesar 0.0257. Hasil pengujian dengan *epoch* 10 menghasilkan nilai MAE sebesar 0.0286 dan MAPE sebesar 0.0488, sementara dengan *epoch* 20 didapatkan nilai MAE sebesar 0.0150 dan MAPE sebesar 0.0257. Penurunan nilai MAE dan MAPE tersebut menunjukkan bahwa model dengan jumlah *epoch* 20 menghasilkan performa yang lebih baik dibanding model dengan jumlah *epoch* 10, sehingga dapat meningkatkan ketepatan prediksi.

Penelitian yang dilakukan oleh Riyantoko dkk., (2020) menggunakan algoritma LSTM untuk memprediksi harga saham di sektor perbankan. Hasil penelitian menunjukkan bahwa LSTM memiliki prediksi yang akurat dengan ditunjukkan pada nilai RMSE serta model data yang di dapatkan pada variasi nilai *epoch*. Nilai RMSE terendah sebesar 57,31 diperoleh pada saham BRI. *Optimizer* Adam dengan 100 *epoch* menunjukkan kinerja terbaik dalam hal akurasi dan nilai loss untuk sebagian besar data saham.

Penelitian yang dilakukan oleh Julian & Pribadi, (2021) menggunakan algoritma LSTM dengan fokus pada sektor pertambangan. Penelitian ini menggunakan data selama lima tahun dengan total 1.261 baris dan menunjukkan

bahwa penambahan jumlah *epoch* berpengaruh terhadap akurasi prediksi, dengan hasil terbaik tercapai pada 200 *epoch*. Nilai RMSE optimal yang dihasilkan oleh metode LSTM untuk emiten TINS adalah sebesar 31.71.

Penelitian yang dilakukan oleh Kartika & Karmilasari, (2022) menggunakan metode LSTM untuk melakukan prediksi harga saham PT Aneka Tambang Tbk (PT Antam). Penelitian ini menggunakan satu *layer* LSTM dengan 64 *neuron*. Pada penelitian ini didapatkan hasil prediksi terbaik pada epoch 100, didapatkan nilai RMSE yang mendekati nilai 0 sebesar 0.3304 dan nilai R2 yang hampir mendekati 1 yaitu sebesar 0.9902.

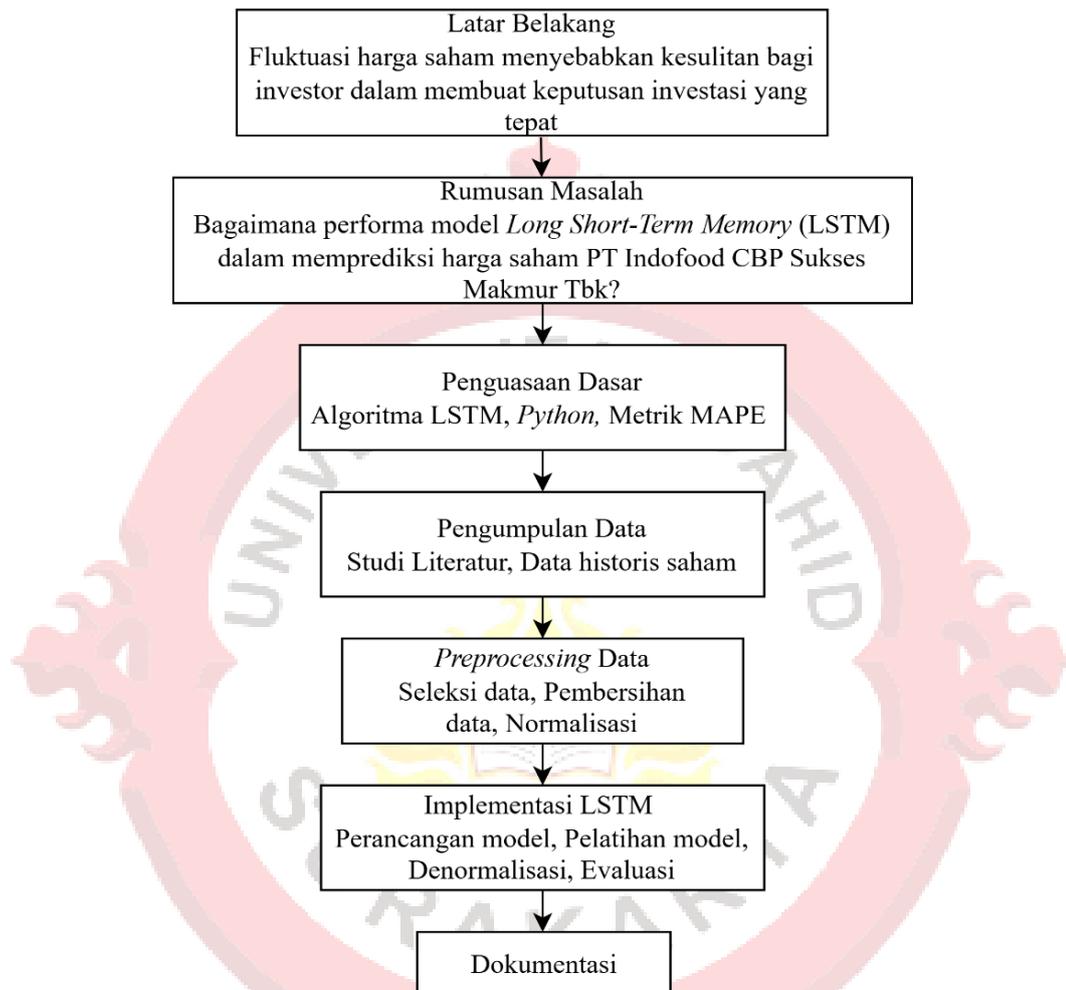
Penelitian oleh Bhandari dkk., (2022) membandingkan performa antara arsitektur LSTM satu *layer* dan *multi-layer* dalam memprediksi harga penutupan indeks S&P 500. Hasil penelitian menunjukkan bahwa model LSTM satu *layer* menghasilkan akurasi prediksi yang lebih tinggi dibandingkan model *multi-layer*, dengan nilai *Correlation Coefficient* (R) yang lebih tinggi yaitu 0.9976.

Penelitian oleh Li (2024) membahas optimalisasi arsitektur LSTM untuk prediksi harga saham, dengan fokus pada variasi jumlah unit *neuron* per *layer*. Hasil penelitian menunjukkan bahwa konfigurasi dengan 64 unit per *layer* memberikan performa terbaik, ditunjukkan oleh nilai RMSE terendah sebesar 0,6689, dibandingkan dengan konfigurasi 16 unit, dan 512 unit.

Berdasarkan studi-studi tersebut, dapat disimpulkan bahwa metode LSTM telah banyak digunakan untuk prediksi harga saham dengan tingkat akurasi yang bervariasi. Sebagian besar penelitian terfokus pada sektor perbankan dan pertambangan. Penelitian ini berfokus pada saham sektor konsumen, yaitu PT Indofood CBP Sukses Makmur Tbk (ICBP), serta menguji performa model dengan kombinasi beberapa konfigurasi *epoch* dan *batch size*. Percobaan dilakukan dengan 4 variasi *epoch* (10, 20, 50, 100) dan 2 variasi *batch size* (32 dan 64). Kombinasi ini ditujukan untuk mengevaluasi performa model secara menyeluruh berdasarkan nilai MAPE sebagai metrik utama.

## 2.2 Kerangka Pemikiran

Berikut merupakan kerangka penelitian yang menggambarkan alur penelitian ini, ditunjukkan pada Gambar 2.1.



Gambar 2.1 Kerangka Pemikiran

### 1. Latar Belakang

Tahapan paling awal, yakni penjelasan mengenai isu atau permasalahan yang menjadi fokus penelitian. Hal ini mencakup alasan mengapa masalah tersebut penting, relevan dan perlu diteliti

### 2. Rumusan Masalah

Penyimpulan latar belakang masalah menjadi suatu rumusan masalah yang akan diangkat menjadi bahan penelitian.

### 3. Penguasaan Dasar

Tahapan untuk mempelajari dasar-dasar algoritma LSTM dan pemrograman *python* yang bertujuan untuk lebih menguasai penerapan model LSTM untuk prediksi data *time series*.

### 4. Pengumpulan Data

Pengumpulan data saham dilakukan dengan cara mengambil *dataset* dari situs Yahoo Finance menggunakan *library yfinance* dari Python.

### 5. *Pre-processing*

*Pre-processing* bertujuan untuk membersihkan dan mempersiapkan data. Serta akan dilakukan pembagian data menjadi data latih dan data uji.

### 6. Implementasi model LSTM

Pada tahap ini dimulai dengan perancangan model LSTM, yang kemudian dilatih menggunakan data latih yang telah dinormalisasikan. Setelah dilakukan pelatihan model diuji menggunakan data uji untuk menghasilkan prediksi. Selanjutnya, model diuji menggunakan data uji dan dievaluasi dengan metrik seperti RMSE untuk mengukur akurasi prediksi.

### 7. Dokumentasi

Tahap dokumentasi dilakukan untuk mencatat dan menyimpan semua proses yang telah dilakukan dalam penelitian.

## 2.3 Landasan Teori

### 2.3.1 Pasar Modal

Pasar modal merupakan pasar untuk berbagai instrumen keuangan jangka panjang yang bisa diperjualbelikan seperti obligasi, saham, reksa dana, instrumen derivatif maupun instrumen lainnya. Pasar modal merupakan sarana pendanaan bagi perusahaan maupun institusi lain (misalnya pemerintah), dan sebagai sarana bagi kegiatan berinvestasi (Finthariasari dkk., 2020). Undang-Undang Republik Indonesia No. 8 Tahun 1995 mendefinisikan pasar modal sebagai kegiatan yang bersangkutan dengan Penawaran Umum dan perdagangan Efek, Perusahaan Publik

yang berkaitan dengan Efek yang diterbitkan, serta lembaga dan profesi yang berkaitan dengan Efek (Undang-Undang RI No. 8, 1995).

### **2.3.2 Saham**

Saham merupakan salah satu instrumen pasar modal yang paling banyak diminati investor karena memberikan tingkat keuntungan yang menarik. Saham dapat didefinisikan sebagai tanda penyertaan modal seseorang atau sepihak (badan usaha) dalam suatu perusahaan atau perseroan terbatas. Dengan menyertakan modal, maka pihak tersebut memiliki klaim atas pendapatan perusahaan, klaim aset perusahaan, dan berhak hadir dalam rapat umum pemegang saham (Aryawati dkk., 2022).

### **2.3.3 Prediksi**

Prediksi adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan masa sekarang yang dimiliki, agar kesalahannya (selisih antara sesuatu yang terjadi dengan perkiraan) dapat diperkecil (Mukhlisin dkk., 2020). Prediksi tidak harus memberikan jawaban secara pasti kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban sedekat mungkin yang akan terjadi (Wathani dkk., 2023).

### **2.3.4 Time Series**

*Time series* merupakan salah satu prosedur statistika yang diterapkan untuk meramalkan struktur probabilitas keadaan yang terjadi di masa yang akan datang dalam rangka pengambilan keputusan untuk sebuah perencanaan tertentu (Wiharja & Ningrum, 2020).

Salah satu jenis data di dalam ilmu statistika, salah satunya adalah jenis data menurut waktu pengumpulannya yaitu data time series. Data runtun waktu adalah nilai– nilai suatu variabel yang berurutan menurut waktu periode data tersebut, sebagai contoh periode hari, minggu, bulan, atau tahun (Ayunda dkk., 2021).

### **2.3.5 Python**

Python merupakan bahasa pemrograman tingkat tinggi (*high level programming*) yang dikembangkan oleh Guido van Rossum pada tahun 1989.

Selain itu, Python mudah dipelajari karena memiliki sintaks jelas dan elegan, yang dikombinasikan dengan penggunaan modul - modul yang mempunyai struktur data tingkat tinggi, efisien, dan siap langsung digunakan (Sarno dkk., 2023). Sebagai salah satu bahasa pemrograman yang paling populer, menyediakan berbagai pustaka yang kuat untuk analisis data, di antaranya Pandas dan NumPy (Candra, 2025). Pandas dan NumPy adalah dua pustaka Python yang paling banyak digunakan dalam komunitas data *science*. Pandas dirancang untuk memudahkan manipulasi dan analisis data, sementara NumPy memberikan dukungan untuk operasi numerik yang efisien (Candra, 2025). Bahasa Python dipilih sebagai bahasa pemrograman dalam penelitian ini karena python memiliki penulisan sintaksis yang mudah selain itu juga memiliki *library* yang lengkap serta bersifat *open source*.

### **2.3.6 Google colab**

*Google Colab* (singkatan dari *Google Colaboratory*) adalah *platform* berbasis *cloud computing* yang disediakan oleh *Google*. *Google Colab* sering digunakan oleh para ilmuwan data, peneliti, dan pengembang untuk melakukan berbagai jenis pekerjaan, termasuk pemrosesan data, pengembangan model kecerdasan buatan, analisis data, dan pelatihan model mesin (Andarsyah & Yanuar, 2024). Pada penelitian ini *google colab* digunakan untuk data *cleaning*, transformasi data, visualisasi data, *machine learning*, alat simulasi, dan lain sebagainya.

### **2.3.7 Data Mining**

*Data mining* atau penambangan data adalah teknik yang relatif cepat dan mudah untuk menemukan pengetahuan, pola dan/atau relasi antar data secara otomatis (Suyanto, 2019). Secara umum, kegunaan data mining dapat dibagi menjadi dua: deskriptif dan prediktif. Deskriptif berarti data mining digunakan untuk mencari pola-pola yang dapat dipahami manusia yang menjelaskan karakteristik data. Sedangkan prediktif berarti data mining digunakan untuk membentuk sebuah model pengetahuan yang akan digunakan untuk melakukan prediksi (Suyanto, 2019).

### 2.3.8 Machine Learning

*Machine Learning* merupakan sub dari *Artificial intelligence* yang memiliki tujuan untuk meningkatkan pengetahuan atau performa (Julian & Pribadi, 2021). *Machine Learning* menggunakan beberapa jenis algoritma yang dilakukan secara berulang-ulang menuntun komputer untuk belajar dari data. Komputer akan diberi *input* berupa data secara terus menerus. Sedangkan algoritma akan “berubah” sesuai dengan data yang diberikan. Dengan demikian kemampuan berpikir komputer menjadi meningkat, mampu memahami data *input* (Sarno dkk., 2023).

### 2.3.9 Normalisasi

Normalisasi merupakan proses penskalaan ulang data dari nilai aslinya ke dalam rentang tertentu. Proses ini memerlukan informasi mengenai nilai minimum dan maksimum dari data yang diamati, yang dapat diperoleh atau diperkirakan secara akurat berdasarkan data yang tersedia. Normalisasi dilakukan pada *dataset* untuk meminimalkan *error*, mengubah data aktual menjadi nilai dengan rentang 0-1 dengan teknik MinMaxScaler (Julian & Pribadi, 2021). Pengubahan nilai ke dalam rentang 0-1 pada normalisasi *min-max* memiliki tujuan agar nilai yang terdapat pada setiap data memiliki proporsi yang sama dalam pemrosesannya (Marthasari dkk., 2021). Rumus dari normalisasi *min-max* ditunjukkan pada Persamaan 2.1.

$$x' = \frac{(x-x_{min})}{(x_{max}-x_{min})} \dots \dots \dots \text{Persamaan(2.1)}$$

Keterangan:

- $x'$  : Data hasil normalisasi
- $x$  : Nilai data yang akan dinormalisasi
- $x_{max}$  : Nilai maksimal
- $x_{min}$  : Nilai minimal

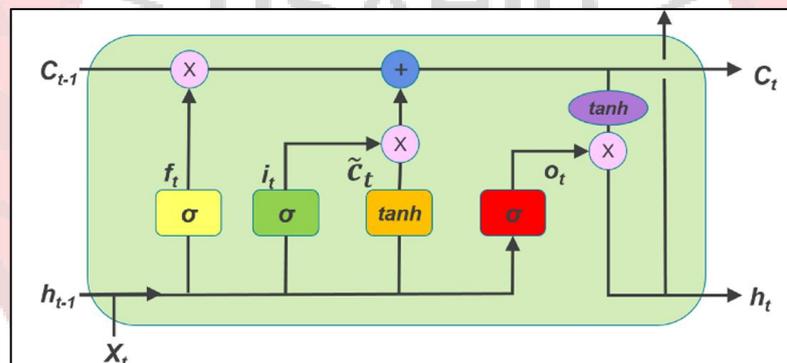
### 2.3.10 Pemisahan Data

Pemisahan data adalah proses membagi dataset menjadi dua atau lebih subset, umumnya terdiri dari data *training* dan data *testing*. Data *training* adalah bagian yang digunakan untuk melatih model, sedangkan data *testing* merupakan

data yang diuji dalam model akhir dan dibandingkan dengan kumpulan data sebelumnya (Sarno dkk., 2023). Rasio pemisahan data yang sering digunakan adalah 80:20, namun rasio lainnya seperti 70:30, 60:40, atau bahkan 50:50 juga diterapkan dalam praktik. Meskipun demikian, belum ada panduan yang jelas tentang rasio yang paling tepat atau optimal untuk setiap *dataset* (Joseph, 2022).

### 2.3.11 Long Short-Term Memory (LSTM)

LSTM merupakan pengembangan dari RNN yang diperkenalkan oleh *Sepp Hochreiter* dan *Jürgen Schmidhuber* pada tahun 1997. LSTM mampu mengatasi *vanishing gradient* atau keadaan nilai gradien adalah 0 atau dekat dengan 0 dengan mekanisme *gate* (Julian & Pribadi, 2021). *Vanishing gradient* merupakan keadaan yang disebabkan oleh nilai gradien yang selalu mengecil hingga *layer* terakhir sehingga nilai bobot tidak berubah dan menyebabkan model tidak pernah memperoleh hasil yang lebih baik (Rowan dkk., 2022).



Gambar 2.2 Arsitektur Jaringan LSTM (Sari dkk., 2020)

LSTM dapat mengatur informasi dalam status sel dengan menambahkan atau menghapusnya secara terstruktur menggunakan mekanisme yang dikenal sebagai *gate* (gerbang). LSTM memiliki tiga jenis *gate*, yaitu *input gate*, *forget gate*, dan *output gate*. Ketiga gerbang ini adalah gerbang analog yang menggunakan fungsi sigmoid dengan rentang nilai antara 0 hingga 1 (Yadav dkk., 2020).

*Forget gate* akan menentukan informasi mana yang akan dibuang dari sel memori (*cell state*). *Forget gate* dihitung menggunakan fungsi sigmoid yang menerima  $h_{t-1}$  dan  $x_t$  yang menghasilkan nilai antara 0 hingga 1. Jika nilai mendekati 0, maka informasi dari *cell state* akan dihapus, dan jika nilai mendekati

1, maka informasi akan disimpan. Persamaan yang digunakan untuk menghitung *forget gate* pada Persamaan 2.2:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \dots \dots \dots \text{Persamaan(2.2)}$$

Keterangan:

$f_t$  : *forget gate*

$\sigma$  : fungsi sigmoid

$W_f$  : nilai bobot (*weight*) untuk *forget gate*

$h_{t-1}$  : *hidden state* pada *timestep* sebelumnya

$x_t$  : nilai *Input*

$b_f$  : nilai bias *forget gate*

*Input gate* menggunakan fungsi aktivasi sigmoid untuk menentukan bagian mana yang akan diperbaharui. Fungsi aktivasi tanh digunakan untuk membuat vektor dari nilai kandidat baru. *Input gate* dihitung menggunakan Persamaan 2.3 dan Persamaan 2.4

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \dots \dots \dots \text{Persamaan(2.3)}$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \dots \dots \dots \text{Persamaan(2.4)}$$

Keterangan:

$i_t$  : *input gate*

$\tilde{C}_t$  : kandidat vektor baru

$\sigma$  : fungsi sigmoid

$\tanh$  : fungsi tanh

$W_i, W_c$  : nilai bobot (*weight*)

$h_{t-1}$  : *hidden state* pada *timestep* sebelumnya

$x_t$  : nilai *Input* periode ke  $t$

$b_i, b_c$  : nilai bias

Kemudian *cell state* diperbaharui melalui Persamaan 2.5.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \dots \dots \dots \text{Persamaan(2.5)}$$

- $C_t$  : *cell state*  
 $f_t$  : *forget gate*  
 $C_{t-1}$  : *cell state* sebelumnya  
 $i_t$  : *input gate*  
 $\tilde{C}_t$  : kandidat vektor baru

Gerbang terakhir, yaitu *output gate* yang berperan untuk menentukan nilai pada bagian memory cell mana yang akan dikeluarkan dengan menggunakan fungsi aktivasi sigmoid (Selle dkk., 2022). Kemudian *cell state* diperbaharui menggunakan fungsi tanh kemudian dikalikan dengan *output* dari *output gate* untuk menghasilkan *hidden state* yang baru. Persamaan *output gate* diuraikan pada Persamaan 2.6 dan 2.7.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \dots \dots \dots \text{Persamaan(2.6)}$$

$$h_t = o_t * \tanh(C_t) \dots \dots \dots \text{Persamaan(2.7)}$$

Keterangan:

- $o_t$  : *output gate*  
 $\sigma$  : fungsi sigmoid  
 $\tanh$  : fungsi tanh  
 $W_o$  : nilai bobot (*weight*) untuk *output gate*  
 $h_{t-1}$  : *hidden state* pada *timestep* sebelumnya  
 $x_t$  : nilai *Input*  
 $b_o$  : nilai bias *output gate*  
 $C_t$  : *cell state*

### 2.3.12 TensorFlow

TensorFlow adalah *library* yang dikembangkan oleh *google* dan merupakan salah satu *library* yang paling populer serta banyak digunakan untuk mengembangkan dan menerapkan *machine learning* dan algoritma lain yang memiliki banyak operasi matematika untuk dilakukan (Sarno dkk., 2023). TensorFlow dapat melatih dan menjalankan jaringan saraf tiruan (*neural network*),

termasuk LSTM. Dalam penelitian ini digunakan untuk melatih model prediksi harga saham menggunakan LSTM.

### 2.3.13 Mean Absolute Percentage Error (MAPE)

MAPE (*Mean Absolute Percentage Error*) adalah suatu metrik evaluasi yang dihitung dengan cara mencari rata-rata dari selisih mutlak antara nilai aktual dan nilai prediksi, lalu dibagi dengan nilai aktual (Budiprasetyo dkk., 2023). Persamaan MAPE dapat dilihat pada Persamaan 2.8 sebagai berikut:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100 \dots \dots \dots \text{Persamaan(2.8)}$$

Keterangan:

- $n$  : banyaknya data
- $y_i$  : nilai aktual dari data ke-i
- $\hat{y}_i$  : nilai prediksi dari data ke-i

Nilai MAPE yang dihasilkan dikelompokkan menjadi beberapa kategori berdasarkan rentang nilai untuk menentukan kualitas dari model yang digunakan dalam memprediksi harga saham.

Tabel 2.1 Kategori Nilai MAPE

Rentang Nilai MAPE	Kategori Akurasi
< 10	Performa model prediksi akurat
10 – 20	Performa model prediksi baik
20 – 50	Performa model prediksi layak
> 50	Performa model prediksi tidak akurat

**Sumber:** Budiprasetyo dkk., (2023)

### 2.3.14 Mean Squared Error (MSE)

*Mean Squared Error* (MSE) merupakan metrik yang digunakan untuk mengukur seberapa besar kesalahan antara nilai prediksi dan nilai aktual. MSE menghitung rata-rata dari selisih kuadrat antara nilai target dan nilai prediksi (Sarno dkk., 2023). Dalam penelitian ini, MSE digunakan untuk mengukur *loss* pada saat pelatihan model. MSE dapat dihitung menggunakan Persamaan 2.9.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \dots \dots \dots \text{Persamaan(2.9)}$$

Keterangan:

$y_i$  : nilai aktual

$\hat{y}_i$  : nilai prediksi

$n$  : jumlah data

### 2.3.15 *R-Squared* ( $R^2$ )

Koefisien determinasi ( $R^2$ ) adalah metrik yang dihitung berdasarkan jumlah kesalahan kuadrat, dan digunakan untuk menunjukkan berapa persen variasi nilai target yang dapat dijelaskan oleh model prediksi (Sarno dkk., 2023).  $R^2$  juga berfungsi untuk menilai tingkat akurasi prediksi. Nilai  $R^2$  berada dalam rentang 0 sampai 1, di mana semakin mendekati 1, maka model dianggap memiliki performa yang semakin baik (Kartika & Karmilasari, 2022).  $R^2$  dapat dihitung menggunakan Persamaan 2.10.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \dots \dots \dots \text{Persamaan(2.10)}$$

Keterangan:

$y_i$  : nilai aktual

$\hat{y}_i$  : nilai prediksi

$n$  : jumlah data

$\bar{y}_i$  : nilai rata-rata dari data aktual