BAB II
KAJIAN LITERATUR

2.1 Prediksi Penjualan (*Sales Forecasting*)

Teknologi informasi dan penetrasi internet yang semakin luas dalam perdagangan online telah mengalami pertumbuhan yang pesat, memberikan peluang besar bagi pelaku usaha untuk meningkatkan penjualan mereka secara signifikan. Hanya, di tengah persaingan tidak cukup bagi perusahaan *e-commerce* hanya memiliki platform online yang menarik. Penting, memahami perilaku pelanggan dan tren pasar yang berperan krusial sebagai teknik *data mining* dalam mengoptimalkan penjualan online. Penelitian ini bertujuan untuk menjelajahi teknik *data mining* berbasis *deep learning,* yaitu model LSTM yang dapat diterapkan secara efektif dalam konteks *e-commerce* untuk memprediksi penjualan dan tren pasar. Dengan memanfaatkan informasi analisis yang ada pada dataset *e-commerce*, yang dapat membuat keputusan yang lebih baik dan merancang strategi pemasaran yang efektif dan memprediksi tren penjualan [13].

Prediksi penjualan merupakan proses penting dalam dunia bisnis yang digunakan untuk memperkirakan jumlah produk atau layanan yang akan terjual di masa depan, serta menjadi dasar dalam berbagai pengambilan keputusan strategis seperti perencanaan inventori, penjadwalan produksi, pengelolaan arus kas, dan penentuan target penjualan. Dalam konteks *e-commerce*, peran prediksi penjualan menjadi semakin krusial karena karakteristik data yang cenderung fluktuatif, tidak stabil, dan dipengaruhi oleh berbagai faktor seperti perubahan trafik, harga, serta aktivitas pasar lainnya. Selain itu, proses prediksi penjualan tidak hanya bergantung pada data historis, tetapi juga melibatkan berbagai informasi multidisiplin seperti tren pasar, data pelanggan, strategi bisnis, serta analisis kompetitor. Oleh karena itu, diperlukan pendekatan yang tepat dalam memilih metode prediksi agar dapat menghasilkan akurasi yang tinggi dan mendukung keberhasilan operasional bisnis secara keseluruhan [14].

Pentingnya prediksi penjualan juga terlihat dari perannya dalam meningkatkan efisiensi operasional perusahaan, khususnya dalam pengelolaan sumber daya seperti penjadwalan tenaga kerja dan perencanaan kegiatan operasional lainnya. Dalam praktiknya, prediksi penjualan dapat dilakukan melalui berbagai pendekatan, antara lain berdasarkan intuisi manajerial dan pemodelan ekonomi. Seiring dengan perkembangan teknologi, metode *machine learning* mulai banyak diterapkan karena dinilai lebih mampu menangkap

pola kompleks dibandingkan metode statistik tradisional. Salah satu pendekatan yang berkembang adalah penggunaan model berbasis *Recurrent Neural Network* (RNN), khususnya LSTM, yang terbukti mampu meningkatkan performa prediksi dibandingkan metode klasik seperti ARIMA. Hal ini didukung oleh karakteristik data penjualan yang umumnya memiliki pola musiman dan dependensi jangka panjang, sehingga memerlukan metode yang mampu memodelkan hubungan temporal secara lebih efektif [15].

Dalam perkembangan lebih lanjut, kebutuhan akan prediksi penjualan yang akurat semakin meningkat seiring kompleksitas sistem *e-commerce* dan manajemen inventori modern. Prediksi yang tepat menjadi dasar dalam perencanaan jaringan gudang, pengalokasian sumber daya, serta pengambilan keputusan strategis untuk meningkatkan efisiensi operasional dan mengurangi biaya. Namun, data penjualan modern yang bersifat dinamis, *non-linear*, dan memiliki fluktuasi tinggi menjadi tantangan bagi metode tradisional. Oleh karena itu, dikembangkan pendekatan yang lebih canggih dengan mengombinasikan model *non-linear*, seperti penggunaan LSTM untuk memodelkan fluktuasi *non-linear* dan frekuensi tinggi yang melekat kompleks pada data penjualan harian dan dependensi jangka panjang pada data, dan ini terbukti menghasilkan peningkatan signifikan daam akurasi peramalan [16].

Agar hasil prediksi dapat secara efektif menjawab masalah yang ada, prediksi sebaiknya mengikuti tahapan baku sebagai berikut ini :

1. Perumusan masalah dan pengumpulan data.

   Tahap pertama yang penting dan menentukan keberhasilan prediksi adalah menentukan masalah tentang apa yang akan diprediksi. Formulasi maslah yang jelas akan menuntun pada ketetapan jenis dan banyaknya data yang akan dikumpulkan. Apabila masalah telah ditetapkan, namun data tidak tersedia, maka harus dilakukan perumusan ulang atau mengubah metode prediksi.

2. Persiapan data.

   Setelah masalah dirumuskan dan data telah terkumpul, tahap selanjutnya adalah menyiapkan data hingga data diproses dengan benar. Hal ini diperlukan, karena dalam praktek ada beberapa maslah yang berkaitan dengan data yang terkumpul, yaitu :

   a. Jumlah data terlalu banyak. Pada umumnya, semakin banyak data akan semakin valid hasil prediksi, namun demikian, jumlah data yang sangat banyak justru berakibat hasil prediksi tidak dapat menjelaskan situasi sebenarnya, karena *time*

horizon dapat menjadi sangat panjang, yang dapat berakibat banyak data tidak relevan lagi.

b. Jumlah data justru terlalu sedikit. Beberapa metode prediksi pada umumnya jumlah data dibawah sepuluh dianggap tidak memadai untuk kegiatan prediksi secara kuantitatif.
c. Data harus diproses terlebih dahulu.
d. Data tersedia, namun rentang waktu data tidak sesuai dengan masalah yang ada.
e. Data tersedia, namun cukup banyak data yang hilang (*missing*), yakni data yang tidak lengkap, hal ini mengakibatkan hasil prediksi akan kurang valid, biasanya akan dilakukan perlakuan data missing, seperti melakukan rata-rata diantara dua data yang lengkap atau cara lain

3. Membangun model.

   Setelah data dianggap memadai dan siap dilakukan kegiatan prediksi, proses selanjutnya adalah memilih model yang tepat untuk melakukan peramalan pada data tersebut.

4. Implementasi model.

   Setelah metode prediksi ditetapkan, maka model dapat diterapkan pada data yang dapat dilakukan prediksi pada data untuk beberapa periode kedepan.

5. Evaluasi prediksi.

   Hasil prediksi yang telah ada kemudian dibandingkan dengan data aktual. Tentu saja tidak ada model prediksi yang dapat memprediksi data dimasa depan secara tepat, yang ada adalah ketepatan prediksi yang nantinya akan dipakai sebagai acuan dari data aktual sehingga dapat mengambil keputusan.

2.2 Preprocessing Data

LSTM memiliki keunggulan dalam menghasilkan data yang akurat atau bersih, efektif untuk kumpulan data yang memiliki rentang waktu. *Preprocessing data* merupakan langkah yang sangat penting dalam pembuatan model LSTM, karena hal ini dapat mempengaruhi hasil akhir analisis prediksi secara signifikan. Namun, meskipun LSTM telah memberikan hasil yang baik dalam berbagai kasus, keberhasilan sangat tergantung pada pemilihan fitur atau atribut yang relevan [17]. Tahapan *preprocessing* pada penelitian ini terdiri dari Seleksi Data, Pembersihan Data, Transformasi Data, Penanganan *Missing Value*, Rekayasa Fitur, Seleksi Atribut, Encoding Data, Normalisasi Data, dan Pengecekan Outlier.

Pertama, Seleksi Data merupakan tahap penting dalam analisis data, melibatkan pemilihan subnet atribut yang paling relevan dan informatif dari sekumpulan atribut data yang tersedia untuk digunakan dalam pembuatan model [18]. Oleh karena itu, seleksi pesanan berkategori tunggal dilakukan dengan hanya mempertahankan transaksi yang memuat satu produk. Pesanan yang memuat lebih dari satu kategori dibuang karena jumlah unit terjualnya tidak tidak dapat dipisahkan. Dengan begitu, setiap unit penjualan dapat dipastikan berasal dari satu kategori sehingga penjualan per kategori dapat dihitung secara akurat.

Langkah berikutnya adalah Pembersihan Data, yaitu mengonversi penanda waktu (*timestamp*) ke dalam format tanggal-waktu yang baku agar data dapat diurutkan dan di proses sebagai deret waktu. Selain konversi format, permasalahan umum pada data waktu meliputi *timestamp* yang hilang, duplikat, maupun urutan yang tidak berurutan, sehingga baris dengan waktu yang tidak valid perlu ditangani agar tidak merusak susunan deret [19]. Pada penelitian ini, kolo Waktu Pesanan Dibuat diuraikan (*parsing*) menjadi format tanggal, kemudian data yang kosong akan dihapus, serta menghapus atribut yang tidak diperlukan dalam pengolahan data nantinya, yang bertujuan untuk mengurangi kompleksitas atribut saat penerapan algoritma [20].

Selanjutnya Transformasi Data, yaitu menggabungkan data transaksi yang bersifat per kejadian ke dalam interval waktu tertentu (misalnya harian atau mingguan) sehingga terbentuk suatu deret waktu yang teratur. Agregasi temporal merupakan langkah *preprocessing* penting dalam analisis *time series* karena menyederhanakan struktur data dan mengurangi derau (*noise*), serta pemilihan tingkat agregasinya dapat mempengaruhi performa model prediksi [21]. Pada penelitian ini, data transaksi diagregasi menjadi total penjualan harian untuk setiap kategori produk agar pola penjualan lebih stabil dan dapat dimodelkan sebagai *time series*. Penggabungan kategori minor, kovariat kategorikal sering ditemui dalam banyak tugas prediksi. Kovariat semacam itu sering kali memiliki distribusi berekor panjang (*long tailed*) dengan banyak kategori langka. Hal ini menimbulkan masalah karena tidak ada cara untuk memperkirakan respons yang terkait dengan kategori-kategori langka tersebut secara akurat. Salah satu solusi untuk masalah ini adalah melakukan *preprocessing data* dengan mengelompokkan kategori-kategori yang hanya memiliki beberapa pengamatan kedalam satu kategori langka [22]. Pada penelitian ini, kategori dengan jumlah hari aktif kurang dari 100 hari digabung menjadi kategori “Lainnya” agar setiap deret memiliki cukup data untuk dilakukan pemrosesan model.

Kemudian dalam prediksi, keberadaan outliers atau data yang menyimpang juga menjadi masalah yang sering mengganggu akurasi model. Outlier dapat menyebabkan model yang tidak akurat dengan menghasilkan estimasi parameter yang bias dan mengarah pada misspesifikasi model. Oleh karena itu, deteksi outlier yang tepat menjadi langkah penting dalam meningkatkan kualitas model prediksi, terutama pada data deret waktu. Outlier yang tidak terdeteksi dapat merusak hasil prediksi dan mengurangi efektivitas model dalam menangkap pola data yang relevan. Deteksi outlier yang efektif dapat membantu menyaring data yang tidak relevan atau tidak wajar sebelum digunakan dalam pelatihan model, yang pada gilirannya meningkatkan akurasi dan stabilitas model [23].

Penanganan data yang tidak lengkap merupakan salah satu tahapan penting dalam preprocessing karena keberadaan nilai yang hilang dapat menurunkan kualitas data dan memengaruhi hasil analisis maupun proses pembelajaran model. Oleh karena itu, diperlukan strategi penanganan yang sesuai agar data yang digunakan dalam proses pelatihan memiliki kualitas yang baik dan mampu menghasilkan model prediksi yang lebih andal [24].

Rekayasa fitur membentuk fitur baru dari data yang ada untuk membantu model mengenali pola. Pada peramalan penjualan ritel, penjualan dipengaruhi oleh musim, hari libur, akhir pekan, dan acara khusus lainnya, sehingga penambahan variabel kalender seperti hari dalam minggu, bulan, dan tahun dapat mengidentifikasi pengaruh-pengaruh tersebut [25]. Pada penelitian ini, dari kolom tanggal dibentuk fitur kalender berupa dari dalam minggu, bulan.

Seleksi atribut merupakan salah satu tahapan penting dalam preprocessing yang bertujuan memilih atribut-atribut yang relevan dengan tujuan prediksi. Proses ini dilakukan untuk mengurangi atribut yang bersifat redundan, mengandung noise, atau tidak memberikan informasi yang signifikan terhadap model. Dengan menggunakan atribut yang relevan, kompleksitas model dapat dikurangi sekaligus meningkatkan efisiensi komputasi dan performa prediksi [26].

Encoding kategori diperlukan karena sebagian besar model hanya memproses masukan numerik, sehingga variabel kategorikal harus dipresentasikan sebagai nilai numerik. Pengkodean berurutan (ordinal) beresiko membuat model mengasumsikan adanya urutan yang sebenarnya tidak ada, sedangkan *one-hot encoding* merepresentasikan tiap kategori sebagai vektor biner tanpa memaksa asumsi urutan tersebut [27]. Pada penelitian ini, setiap kategori diberi kode numerik (kategori_id) yang kemudian direpresentasikan dalam bentuk *one-hot* pada tahap pemodelan agar seluruh kategori diperlakukan setara.

Min-Max Normalization, yang merupakan salah satu teknik dalam tahap persiapan data yang diterapkan setelah pembagian data yang digunakan untuk mengubah nilai atribut numerik ke dalam rentang tertentu, umumnya [0,1]. Teknik ini bertujuan untuk menyamakan skala antar fitur sehingga setiap variabel memiliki kontribusi yang seimbang dalam proses pelatihan model. Berikut ini rumus dari *Min-Max Normalization* [28].

$$x' = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (2.1)$$

Keterangan:

x` = Hasil Normalisasi

xi = Data ke i

$xmin$ = Data dengan nilai minimum

$xmax$ = Data dengan nilai maksimum

Sebagai ilustrasi, Tabel 2.1 menyajikan contoh normalisasi pada data penjualan harian (total_qty) salah satu kategori, yaitu Nampan/Tray, dengan nilai minimum 0 dan maksimum 262. Untuk mencegah kebocoran informasi, parameter *x-min* dan *x-max* sebaiknya diperoleh hanya dari data latih lalu diterapkan ke seluruh data .

Tabel 2.1 Contoh Perhitungan Normalisasi Min-Max pada Penjualan Harian Kategori Nampan/Tray

Penjualan Harian (total_qty)	x_min	x_max	Hasil Normalisasi (x′)
0	0	262	0,0000
17	0	262	0,0649
20	0	262	0,0763
210	0	262	0,8015
262	0	262	1,0000

2.3 Pembagian Data

Pembagian data adalah proses membagi dataset menjadi dua atau lebih subnet yang berbeda, biasanya untuk keperluan pelatihan dan pengujian model *deep learning*. Tujuan utama dari pembagian data adalah untuk mengukur kinerja model pada data yang belum pernah dilihat selama proses pelatihan. Ada beberapa jenis pembagian data yang umum digunakan:

1. Pelatihan data

   Data training digunakan untuk pengembangan model dan melatih algoritma. Subnet dari dataset yang digunakan untuk melatih model *deep learning.* Data pelatihan ini berisi contoh-contoh yang diberi label atau ouput yang diinginkan, dan model akan menggunakan data ini untuk belajar dan membuat prediksi pada data baru yang belum pernah dilihat sebelumnya.

2. Pengujian data

   Ketika dihadapkan pada data baru yang tidak teridentifikasi, pengujian dilakukan untuk mengamati performa algoritma yang telah dilatih sebelumnya. Subnet dari dataset yang digunakan untuk menguji kinerja model *deep learning* setelah model dilatih dengan data pelatihan. Testing data berfungsi untuk mengukur sejauh mana model mampu melakukan generalisasi pada data yang belum pernah dilihat sebelumnya, yang disebut sebagai data baru atau data yang belum terlihat [29]. Data yang telah diproses akan dibagi menjadi dua data, yaitu data *training* dan data *testing.* Data *training* akan digunakan untuk melatih model LSTM, sedangkan data *testing* akan digunakan untuk menguji performa model LSTM. Pembagian dilakukan dengan menggunakan dataset, dengan proporsi 80% untuk data *training* dan 20% untuk data *testing* [30].

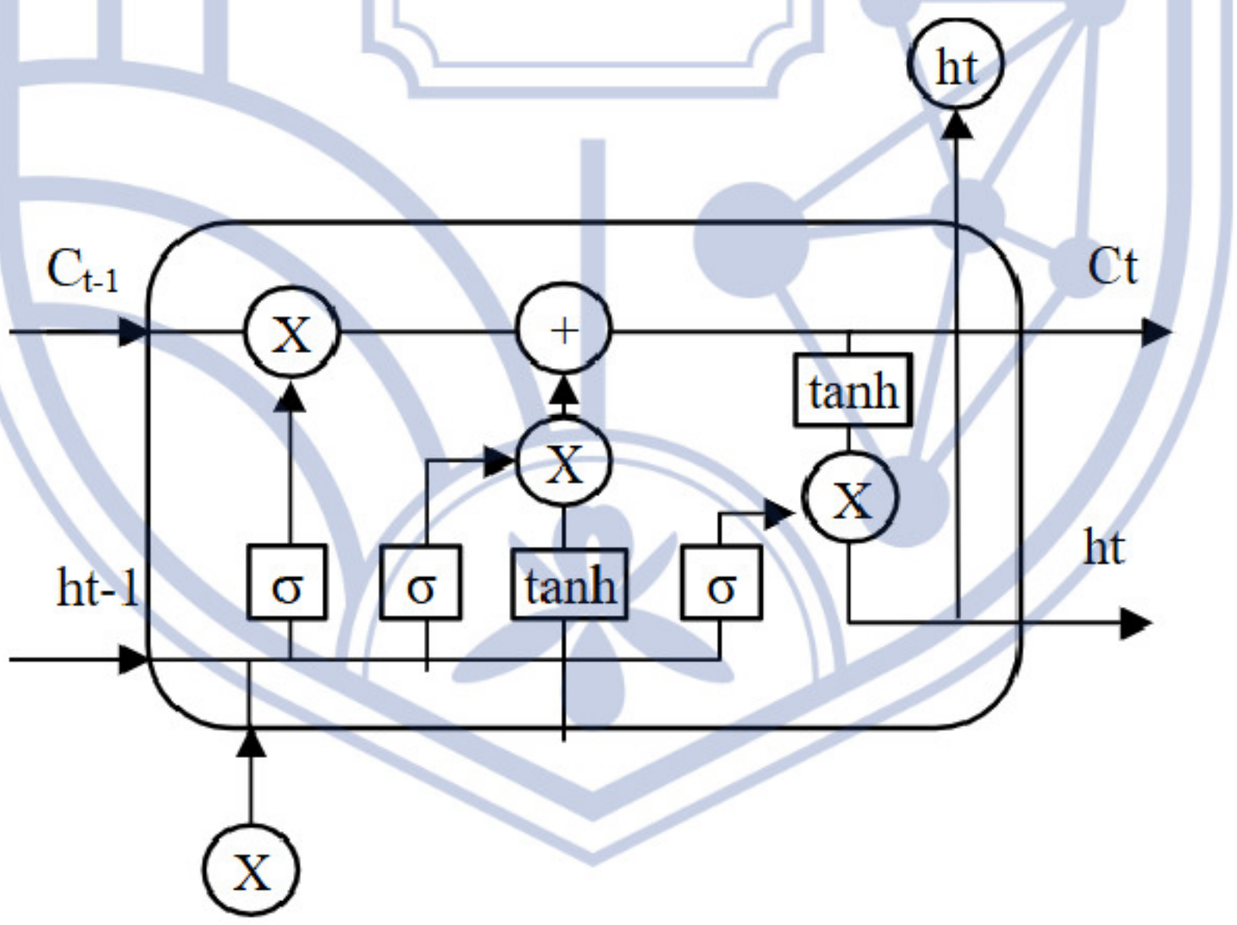
Setelah data dibagi menjadi data latih dan data uji, dilakukan normalisasi agar seluruh fitur numerik berada pada skala yang seragam sehingga proses pelatihan model berjalan optimal. Setelah dinormalisasi, deret waktu diubah menjadi pasangan masukan - keluaran menggunakan teknik *sliding window*, yaitu mengambil sejumlah nilai berurutan sebagai masukan untuk memprediksi nilai pada periode berikutnya sehingga model dapat mempelajari pola antar waktu. Pada penelitian ini digunakan jendela sepanjang tujuh hari untuk memprediksi penjualan hari ke-delapan ($7 \rightarrow 1$), dan jendela dibentuk terpisah untuk tiap kategori agar tidak menyeberang antar kategori.

Pada pendekatan *sliding window,* data dibagi secara dinamis dengan melatih model pada jendela waktu tertentu dan menguji model dengan memprediksi satu hari berikutnya. Jendela pelatihan bergeser sebanyak satu hari setiap kali periode pengujian selesai, dengan ukuran jendela 7 hari untuk melihat bagaimana model dapat beradaptasi. *Sliding window* lebih adaptif terhadap perubahan jangka pendek karena model dilatih ulang secara priodik berdasarkan jendela waktu yang bergerak [31].

2.4 LSTM (Long Short-Term Memory)

Long Short-Term Memory (LSTM) adalah jenis jaringan saraf berulang, (*Recurrent Neural Networks* atau RNN) khusus yang mampu mempelajari depedensi jangka panjang yang diperkenalkan oleh (Hochreiter & Schmidhuber 1997) lalu disempurnakan dan dipopulerkan oleh banyak orang. LSTM memiliki keterampilan mengingat informasi dalam jangka waktu yang lama. Tidak seperti RNN yang punya masalah dengan memori jangka pendek. Jika suatu urutan cukup panjang, RNN akan kesulitan membawa informasi dari tahapan waktu sebelumnya ke tahapan waktu selanjutnya . Hal ini memungkinkan LSTM untuk belajar dari data dengan waktu yang lebih panjang (lebih dari 1000 langkah waktu). Untuk mencapai hal tersebut, LSTM menyimpan informasi lebih banyak di luar aliran jaringan saraf tradisional dalam struktur yang disebut sel berpintu (*gated cells*) [32].

Berbeda dengan RNN standar yang hanya memiliki satu *hidden state,* LSTM dilengkapi dengan mekanisme *cell state* yang berfungsi sebagai memori jangka panjang. Mekanisme ini memungkinkan LSTM untuk secara efektif menyimpan, memperbarui, dan melupakan informasi melalui struktur *gate* yang terdiri dari *forget gate, input gate,* dan *ouput gate* [33]. Kemampuan tersebut membuat LSTM efektif digunakan sebagai model prediksi berbasis data historis, termasuk untuk kasis prediksi jumlah penjualan produk *e-commerce.*



Gambar 2.1 Arsitektur LSTM [28]

LSTM terdiri dari rangkaian sel memori yang unik dan model LSTM menyaring informasi melalui struktur gerbang. Pada LSTM terdapat 4 gerbang atau *Gates Unit,* yaitu *forgate gates, input gate, cell gates,* dan *output gates.* Persamaan dari setiap *gate* sebagai berikut:

1. *Forget Gate:*

$$f_t = \sigma(W_f \times X_t + U_f \times h_{t-1} + b_f) \quad \text{..... (2.2)}$$

Keterangan :
σ = Fungsi aktivasi sigmoid (Menghasilkan nilai antara 0-1)
W_f = Matriks bobot untuk *forgte gate*
h_{t-1} = *Hidden state* dari langkah waktu sebelumnya
X_t = Input data pada waktu saat ini t
b_f = Nilai bias pada *forget gate*

2. *Input Gate:*

$$i_t = \sigma(W_i \times X_t + U_i \times h_{t-1} + b_i) \quad \text{..... (2.3)}$$

Keterangan:
W_i = Matriks bobot untuk *input gate*
b_i = Nilai bias untuk *input gate*

3. *New Candidate:*

$$c'_t = tanh(W_c \times X_t + U_c \times h_{t-1} + b_c) \quad \text{..... (2.4)}$$

Keterangan :
W_c = Matriks bobot untuk *candidate cell*
Tanh = Fungsi aktivasi tangen hiperbolik (menghasilkan nilai antara -1 sampai 1)
b_c = Nilai bias untuk *candidate cell*

4. *Cell State:*

$$c_t = f_t \odot c_{t-1} + i_t \odot c'_t \quad \text{..... (2.5)}$$

Keterangan :
c_t = Nilai *cell state* yang baru pada waktu t
c_{t-1} = Nilai *cell state* dari waktu sebelumnya
$\odot$ = Perkalian elemen demi elemen

5. *Output Gate:*

$$o_t = \sigma(W_o \times X_t + U_o \times h_{t-1} + b_o) \quad \text{..... (2.6)}$$

Keterangan :
o_t = Nilai dari *output gate*
W_o = Matriks bobot untuk *output gate*
b_o = Nilai bias untuk *output gate*
h_{t-1} = *Hidden state* pada langkah waktu sebelumnya (t - 1)

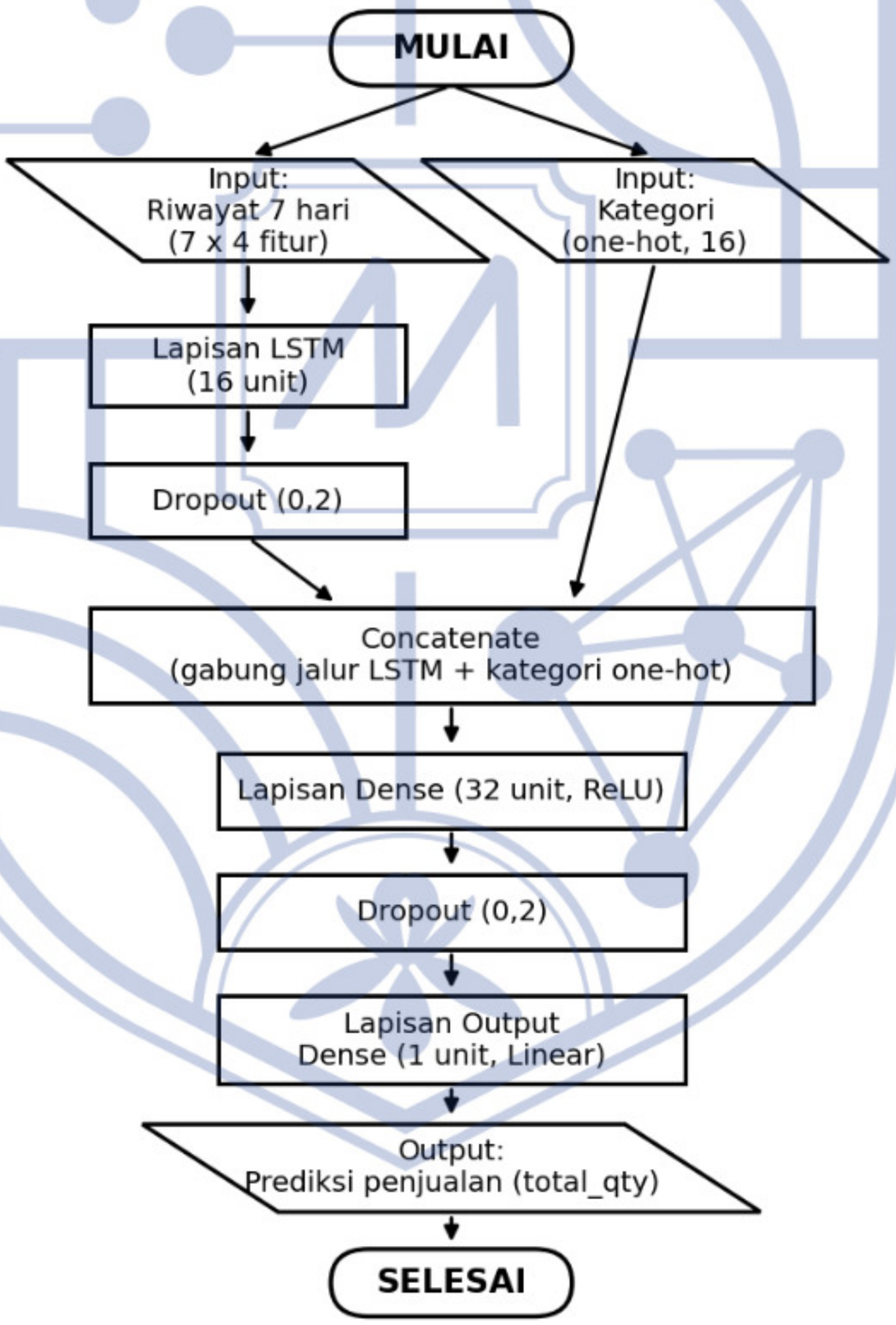
6. *Hidden State:*

$$h_t = o_t \odot tanh(c_t) \quad \text{..... (2.7)}$$

Keterangan :
h_t = *Hidden state* baru yang dikirim ke langkah waktu berikutnya
o_t = Nilai dari *output gate*
$\odot$ = Perkalian elemen demi elemen
Tanh = Fungsi aktivasi tangen hiperbolik (menghasilkan nilai antara -1 sampai 1)
c_t = Nilai *cell state* yang baru pada waktu t

2.4.1 Arsitektur LSTM yang Digunakan

Penelitian ini menggunakan arsitektur LSTM satu lapisan (*single LSTM*) yang dikombinasikan dengan identitas kategori dalam bentuk *one-hot*. Model memiliki dua jalur masukan, yaitu yang pertama memproses masukan sekuensial berupa riwayat tujuh hari (7 x 4 fitur) melalui satu lapisan LSTM 16 unit dan Dropout 0,2. Kemudian jalur yang kedua menerima identitas kategori dalam bentuk *one-hot* secara langsung. *Output* kedua jalur tersebut digabungkan (*concatenate*), lalu diteruskan ke lapisan Dense 32 unit beraktivasi ReLU, Dropout 0,2, dan lapisan keluaran Dense 1 unit beraktivasi *linear* yang menghasilkan prediksi jumlah penjualan. Secara keseluruhan, arsitektur ini memiliki 2.433 parameter. Struktur arsitektur yang digunakan dapat di lihat pada Gambar 2.2 berikut.



Gambar 2.2 Arsitektur Model LSTM yang Digunakan

2.4.2 Fungsi Aktivasi Dalam LSTM

Dalam arsitektur LSTM, terdapat beberapa fungsi aktivasi yang digunakan dengan peran yang berbeda-beda. Pemilihan fungsi aktivasi yang tepat pada setiap komponen sangat mempengaruhi kemampuan model dalam mempelajari pola dari data.

1. Fungsi Sigmoid (σ)

   Sigmoid adalah fungsi aktivasi utama yang digunakan pada seluruh *gate* dalam LSTM. Fungsi ini mengubah semua nilai input menjadi angka dalam rentang [0,1], sehingga bersifat sebagai katup yang mengontrol seberapa banyak informasi yang dilewatkan atau diblokir.

   $$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.8)$$

   Keterangan :

   x = Nilai input

   e = Bilangan Euler

   $\sigma(x)$ = Nilai keluaran fungsi Sigmoid

2. Fungsi Tanh

   Tanh digunakan untuk merepresentasikan isi informasi, baik pada saat pembaruan *cell state* maupun pada *hidden state output.* Tidak seperti sigmoid, tanh menghasilkan nilai dalam rentang [-1, 1] sehingga mampu merepresntasikan informasi bernilai positif maupun negatif.

   $$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \quad (2.9)$$

   Keterangan :

   x = Nilai input

   e = Bilangan Euler

   $tanh(x)$ = Nilai keluaran fungsi Tanh

3. Fungsi ReLU

   Fungsi *Rectified Linear Unit* (ReLU) digunakan pada lapisan dense yang berada setelah lapisan LSTM. ReLU dipilih karena kesederhanaan komputasinya serta kemampuannya mengatasi masalah *vanishing gradient* pada lapisan *non-recurrent.* Fungsi ini menghasilkan nilai nol untuk semua input negatif dan mengembalikan nilai asli untuk input positif [34].

$$ReLU(x) = max(0, x) \quad (2.10)$$

Keterangan :

x = Nilai input

$ReLU(x)$ = Nilai keluaran fungsi ReLU

Tabel 2.2 Perbandingan Fungsi Aktivasi Model LSTM

Fungsi	Output	Digunakan di	Peran dalam LSTM
Sigmoid (σ)	[0, 1]	*Forget, Input, Output Gate*	Mengontrol aliran informasi sebagai katup buka/tutup
Tanh	[-1, 1]	*Cell update, Hidden state*	Merepresentasikan isi informasi baru dan output
ReLU	[0, ∞)	*Layer Dense*	Transformasi fitur non-linear setelah LSTM cell
Linear	(-∞, ∞)	*Output layer*	Menghasilkan nilai prediksi numerik kontinu

2.4.3 Proses Pelatihan LSTM

Proses pelatihan model LSTM pada penelitian ini dilakukan melalui beberapa tahapan. Pertama, data dinormalisasi menggunakan *Min-Max Normalization* ke rentang [0, 1] agar pembelajaran berjalan optimal karena LSTM sensitif terhadap skala masukan. Kedua, data dibagi menjadi data latih dan data uji dengan rasio 80:20 secara kronologis. Ketiga, model dilatih menggunakan fungsi kerugian MSE dengan *optimizer Adam* yang mengadaptasi laju pembelajaran secara otomatis. Keempat, pelatihan dijalankan pada beberapa kombinasi learning rate (0,001 dan 0,0001), jumlah epoch (10, 20, 30, 40, 50, dan 100), dan ukuran batch (16 dan 32) untuk menentukan konfigurasi dengan galat terendah [35].

Pada setiap kombinasi, model memproses seluruh data latih secara berurutan melalui lapisan LSTM, menghitung nilai kesalahan, lalu memperbarui bobot menggunakan algoritma *Backpropagation Through Time* (BPTT). Pelatihan tidak menggunakan early stopping, setiap kombinasi dijalankan hingga jumlah epoch yang ditentukan, kemudian kombinasi dengan galat terendah pada data uji dipilih sebagai model akhir.

2.5 Evaluasi Model MSE, RMSE dan MAE

Evaluasi model dilakukan untuk mengukur seberapa akurat model LSTM dalam memprediksi penjualan harian per kategori (total_qty) pada data uji. MSE digunakan sebagai

fungsi *loss* selama pelatihan, sedangkan RMSE dan MAE digunakan sebagai metrik evaluasi performa. Berikut penjelasan ketiga metrik tersebut.

MSE adalah fungsi kerugian (*loss function*) selama proses pelatihan MSE mengukur selisih antara hasil prediksi dan hasil yang diharapkan model, dengan tujuan mengoptimalkan parameter jaringan [36]. MSE mengukur rata-rata dari kuadrat selisih antara nilai sebenarnya dan nilai prediksi, sehingga setiap kesalahan yang besar akan dikuadratkan dan memberi penalti yang jauh lebih berat dibanding kesalahan kecil. Sifat inilah yang membuat MSE efektif memaksa model menekan kesalahan besar, sekaligus menghasilkan permukaan kerugian yang mulus dan mudah dioptimasi melalui penurunan gradien. MSE dirumuskan sebagai berikut:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad \text{.....} \quad (2.11)$$

Keterangan :

y_i = nilai aktual penjualan harian ke-i
$\hat{y}_i$ = nilai prediksi ke-i
n = jumlah data

Nilai RMSE merupakan akar kuadrat dari MSE (RMSE = √MSE), sehingga RMSE menyatakan kesalahan dalam satuan unit penjualan yang sama dengan data aslinya.

Evaluasi model dilakukan untuk mengukur seberapa akurat model LSTM dalam memprediksi data *time series* (total_qty) pada data uji. Metrik evaluasi digunakan untuk mengukur performa dari prediksi model. Untuk menilai efisiensi dan akurasi model, dua indikator digunakan, yaitu RMSE dan (MAE) [37].

1. *Root Mean Square Error* (RMSE)

   RMSE adalah ukuran error yang diperoleh dari akar kuadrat rata-rata selisih kuadrat antara nilai aktual dan nilai prediksi. RMSE digunakan untuk mengukur seberapa besar kesalahan prediksi model terhadap data aktual. Semakin kecil nilai RMSE, maka semakin baik kemampuan model dalam melakukan prediksi [38]. Berikut rumus RMSE :

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \quad (2.12)$$

Keterangan :

y_i = nilai aktual penjualan harian ke-i

$\hat{y}_i$ = nilai prediksi ke-i

n = jumlah data uji

2. *Mean Absolute Error* (MAE)

MAE digunakan untuk mengevaluasi keakuratan hasil prediksi menggunakan metode mengkalkulasi rata-rata dari selisih absolut antara nilai sebenarnya dan nilai prediksi [39], semakin kecil nilai MAE maka semakin akurat hasil prediksi model. Rumus menghitung nilai MAE dapat dilihat sebagai berikut :

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (2.13)$$

Keterangan :

y_i = nilai aktual penjualan harian ke-i

$\hat{y}_i$ = nilai prediksi ke-i

n = jumlah data uji

2.6 E-Commerce

E-commerce merupakan salah satu konsep bisnis berbasis teknologi informasi yang cukup berkembang saat ini. Konsep *e-commerce* memberikan banyak kemudahan dan kelebihan jika dibandingkan dengan konsep belanja konvensional. Kemudahan pada *e-commerce* adalah semua informasi yang diinginkan konsumen dapat diakses lebih detail, cepat tanpa dibatasi tempat dan waktu, *e-commerce* memberikan kemudahan dalam proses transaksi, sehingga dengan penerapan sistem ini akan mempermudah dan lebih menguntungkan berbagai pihak yang terlibat, baik pihak konsumen maupun penjual [40].

Perkembangan *e-commerce* yang pesat menghasilkan volume data yang sangat besar, membuka peluang untuk menganalisis pola permintaan produk secara lebih akurat [41]. Data yang dihasilkan mencakip transaksi penjualan, pencarian produk, hingga ulasan pelanggan yang dapat diolah menggunakan teknik analisis data dan *machine learning* untuk medukung perencanaan persediaan serta pengambilan keputusan bisnis.

Di Indonesia, e-commerce telah menjadi salah satu pilar utama ekonomi digital dengan kontribusi terhadap perekonomian yang terus meningkat, tercermin dari nilai

transaksi nasional yang naik dari Rp205,5 triliun pada 2019 menjadi Rp487,01 triliun pada 2024. Pertumbuhan tersebut menghasilkan data penjualan berbentuk deret waktu (*time-series*) dalam jumlah besar, sehingga menciptakan peluang untuk memprediksi tren penjualan menggunakan metode statistik maupun machine learning.

Selain data transaksi, aktivitas pengguna pada platform *e-commerce* seperti ulasan produk juga menghasilkan data yang bernilai. Analisis terhadap ulasan pengguna dapat membantu platform mengidentifikasi aspek layanan yang paling sering dibahas sehingga mendukung pengambilan keputusan yang lebih terarah. Beragamnya jenis data yang dihasilkan *e-commerce* menjadikannya sumber informasi penting untuk analisis dan prediksi, termasuk prediksi penjualan yang menjadi fokus penelitian ini [42].