

BAB II

TINJAUAN PUSTAKA

Bab tinjauan pustaka ditujukan untuk mengklarifikasi dan memperkuat gagasan berdasarkan teori dan penelitian-penelitian sebelumnya. Tinjauan Pustaka akan membahas penelusuran literatur, pencapaian terdahulu, serta tinjauan teoritis.

2.1 Pencapaian Terdahulu

Pada penelitian ini, referensi penelitian sebelumnya digunakan sebagai bahan rujukan, landasan, dan penguat argumen. Oleh Karena itu, referensi yang mencangkup topik relevan ditinjau mengenai penerapan model LSTM dengan studi kasus yang berbeda-beda. Lihat Tabel 2.1 Pencapaian Terdahulu

Tabel 2. 1 *Pencapaian Terdahulu*

Literatur ke-1	
Nama Penulis	U. M. Butt, S. Letchmunan, M. Ali, F. H. Hassan, A. Baqir, dan H. H. R. Sherazi (2021)
Judul	<i>Machine Learning Based Diabetes Classification and Prediction for Healthcare Applications</i>
Hasil	Penelitian ini bertujuan untuk mengembangkan pendekatan berbasis machine learning dalam klasifikasi, deteksi dini, dan prediksi diabetes, guna meningkatkan efisiensi dan akurasi deteksi sekaligus mencegah komplikasi serius akibat diabetes mellitus. Dengan menggunakan dataset <i>PIMA Indian Diabetes</i> , penelitian ini menguji tiga classifier— <i>Random Forest (RF)</i> , <i>Multilayer Perceptron (MLP)</i> , dan <i>Logistic Regression (LR)</i> —serta metode prediktif seperti <i>Long Short-Term Memory (LSTM)</i> , <i>Moving Averages (MA)</i> , dan <i>Linear Regression (LR)</i> . Hasil menunjukkan bahwa MLP mencapai akurasi 86,08% dalam klasifikasi, sedangkan LSTM unggul dalam prediksi dengan akurasi 87,26%, menjadikan pendekatan ini efektif untuk aplikasi perawatan kesehatan.
Literatur ke-2	
Nama Penulis	Nizar Alsharif (2022)
Judul	<i>Fake opinion detection in an e-commerce business based on a long-short memory algorithm</i>
Hasil	Penelitian ini mengusulkan penggunaan algoritma Long Short-Term Memory (LSTM) untuk mendeteksi opini palsu dalam bisnis e-commerce, yang sering kali memengaruhi keputusan pembelian konsumen. Model

	<p>LSTM dilatih menggunakan dataset ulasan produk dari Yelp, dengan menambahkan fitur linguistik untuk membedakan ulasan asli dan palsu. Hasil pengujian menunjukkan bahwa model ini mencapai akurasi dan skor F1 sebesar 98%, membuktikan efektivitasnya dalam mengklasifikasikan ulasan sebagai palsu atau asli. Pendekatan ini memberikan kontribusi penting dalam membantu konsumen membuat keputusan pembelian yang lebih tepat dan mengurangi dampak negatif opini palsu di platform e-commerce.</p>
Literatur ke-3	
Nama Penulis	Jiang Han (2024)
Judul	<i>Prediction Model of User Investment Behavior Based on Deep Learning</i>
Hasil	<p>Penelitian ini mengembangkan model prediksi perilaku investasi pengguna menggunakan algoritma Long Short-Term Memory (LSTM) yang telah dioptimalkan untuk meningkatkan akurasi dan efisiensi dalam memprediksi perilaku investasi di pasar saham. Dengan menganalisis dataset yang mencakup 5120 pengguna dan 231497 catatan perilaku investasi dari indeks saham Shanghai, model ini mengatasi keterbatasan metode tradisional seperti RNN melalui penyesuaian struktur jaringan, parameter, dan mekanisme dropout untuk mengurangi overfitting. Hasil penelitian menunjukkan bahwa model ini mampu memberikan analisis perilaku investasi yang akurat, memprediksi tren pasar saham, serta menghasilkan prediksi harga opsi yang lebih baik dibandingkan metode tradisional. Model ini juga dapat memberikan wawasan yang lebih baik terkait tren investasi jangka panjang dan membantu menstabilkan harga opsi serta keseimbangan pasar.</p>
Literatur ke-4	
Nama Penulis	E. Lee, D. Kim, dan H. Bae (2021)
Judul	<i>Container Volume Prediction Using Time-Series Decomposition with a Long Short-Term Memory Models</i>
Hasil	<p>Penelitian ini mengusulkan model prediksi volume kontainer di Pelabuhan Busan dengan menggabungkan dekomposisi deret waktu dan variabel eksternal ke dalam model deep learning berbasis Long Short-Term Memory (LSTM). Model ini dirancang untuk meningkatkan akurasi prediksi volume kontainer, yang sebelumnya menggunakan metode statistik tradisional seperti ARIMA dan regresi namun tidak cukup efektif dalam menangkap kompleksitas dan variabilitas data akibat pengaruh lingkungan eksternal. Dengan pendekatan multivariat yang mempertimbangkan dekomposisi deret waktu dan variabel eksternal, model ini menunjukkan kinerja lebih</p>

	baik dibandingkan LSTM tradisional, serta mampu mengikuti tren volume kontainer secara lebih akurat. Penelitian ini memberikan kontribusi signifikan bagi perencanaan dan operasi pelabuhan yang lebih efisien.
Literatur ke-5	
Nama Penulis	Y. Wang, M. A. bin Abdullah, dan J. Y. T. Hwang (2024)
Judul	<i>Time Series Analysis and Optimization of the Prediction Model of Agricultural Insurance Loss Ratio</i>
Hasil	Penelitian ini bertujuan untuk mengembangkan pendekatan berbasis machine learning dalam klasifikasi, deteksi dini, dan prediksi diabetes, guna meningkatkan efisiensi dan akurasi deteksi sekaligus mencegah komplikasi serius akibat diabetes mellitus. Dengan menggunakan dataset <i>PIMA Indian Diabetes</i> , penelitian ini menguji tiga classifier— <i>Random Forest</i> (RF), <i>Multilayer Perceptron</i> (MLP), dan <i>Logistic Regression</i> (LR)—serta metode prediktif seperti <i>Long Short-Term Memory</i> (LSTM), <i>Moving Averages</i> (MA), dan <i>Linear Regression</i> (LR). Hasil menunjukkan bahwa MLP mencapai akurasi 86,08% dalam klasifikasi, sedangkan LSTM unggul dalam prediksi dengan akurasi 87,26%, menjadikan pendekatan ini efektif untuk aplikasi perawatan kesehatan.
Literatur ke-6	
Nama Penulis	S. Banik, N. Sharma, M. Mangla, S. N. Mohanty, S. Shitharth (2022)
Judul	LSTM Based Decision Support System for Swing Trading in Stock Market
Hasil	Penelitian ini mengembangkan sistem pendukung keputusan berbasis Long Short-Term Memory (LSTM) untuk membantu trader melakukan swing trading di pasar saham. Model LSTM digunakan untuk memprediksi harga saham di masa depan berdasarkan data historis, dengan mempertimbangkan indikator teknis seperti Money Flow Index (MFI), Relative Strength Index (RSI), level Support dan Resistance, serta analisis Moving Average Convergence Divergence (MACD) dan garis sinyal. Hasil penelitian menunjukkan bahwa model LSTM yang diusulkan memiliki kinerja yang lebih baik dibandingkan model tradisional, dengan nilai Root Mean Squared Error (RMSE) yang lebih rendah pada prediksi harga pembukaan, harga penutupan, harga tertinggi, harga terendah, dan volume perdagangan saham. Sistem ini juga menyediakan laporan analisis investasi yang mencakup skor kesuksesan investasi, membantu trader membuat keputusan yang lebih tepat dan mengurangi risiko investasi di pasar saham.
Literatur ke-7	

Nama Penulis	S. Kumar, R. Sharma, T. Tsunoda, T. Kumarevel, dan A. Sharma.(2021)
Judul	Forecasting the spread of COVID-19 using LSTM network
Hasil	Penelitian ini mengembangkan model prediksi berbasis Long Short-Term Memory (LSTM) untuk meramalkan penyebaran COVID-19. Model ini divalidasi menggunakan data dari Selandia Baru, yang berhasil menekan jumlah kasus harian COVID-19 hingga nol, dan menunjukkan bahwa LSTM mampu memprediksi dengan akurat periode waktu pengendalian penyebaran COVID-19. Selain itu, model ini digunakan untuk meramalkan tanggal perkiraan pengendalian di negara lain. Meskipun prediksi bersifat estimasi berdasarkan data saat ini, hasil ini diharapkan dapat membantu pemerintah dan pembuat kebijakan dalam merancang strategi untuk mengendalikan pandemi. Penelitian ini juga menekankan perlunya pembatasan tambahan di banyak negara, karena tanpa intervensi lebih lanjut, diperlukan waktu lebih dari tiga bulan untuk mengendalikan penyebaran COVID-19..

2.2 Tinjauan Teoritis

Pada bagian ini menguraikan teori-teori yang mendasari dalam penelitian. Penguraian setiap teori akan disampaikan secara rinci untuk memberikan landasan konseptual yang jelas bagi penelitian, sebagaimana dijabarkan berikut.

2.2.1 Prediksi

Prediksi atau *forecasting* adalah proses memperkirakan nilai atau kejadian di masa depan berdasarkan analisis data historis. Prediksi digunakan oleh berbagai pihak, termasuk organisasi, perusahaan, pemerintah, dan individu, untuk mendukung pengambilan keputusan strategis. Proses ini dilakukan dalam berbagai rentang waktu bergantung pada kebutuhan, mulai dari harian hingga tahunan. Senapati et al. (2023) Prediksi merupakan tugas mendasar dalam ilmu pengetahuan dan rekayasa yang dilakukan dengan menggunakan model matematika untuk meramalkan peristiwa atau hasil di masa depan melalui analisis pola dari pengamatan masa lalu, sehingga pada dasarnya menjawab apa yang kemungkinan besar akan terjadi berdasarkan perilaku yang telah diketahui sebelumnya. Penerapan prediksi telah dilakukan di berbagai sektor seperti pasar keuangan, kesehatan, bisnis, dan teknologi, baik pada skala lokal maupun global. Tujuan

utama prediksi adalah untuk mengurangi ketidakpastian dan memberikan wawasan yang mendukung perencanaan dan pengelolaan risiko. Savadatti et al. (2022) Prediksi dalam machine learning merujuk pada proses penggunaan algoritma untuk menganalisis data historis dan meramalkan hasil di masa depan. Proses ini melibatkan pengidentifikasian pola dan tren dalam data untuk mendukung pengambilan keputusan yang terinformasi di berbagai bidang, seperti bisnis, kedokteran, dan pendidikan. Proses prediksi dilakukan dengan menggunakan metode berbasis *machine learning* seperti *Long Short-Term Memory* (LSTM), yang mampu menangani data kompleks dan menghasilkan prediksi yang lebih akurat. Menurut (Oak et al., 2024) Model LSTM multivariat mampu menangani beberapa deret waktu secara bersamaan, sehingga dapat menangkap interaksi antar variabel. Pendekatan ini meningkatkan akurasi peramalan dengan mengintegrasikan hubungan yang kompleks. LSTM mampu melakukan prediksi dengan memanfaatkan mekanisme *gates* (*input, forget, dan output*) serta *cell state* untuk menangkap hubungan temporal dan pola jangka panjang dalam data berurutan, sehingga menghasilkan prediksi yang akurat.

2.2.2 Machine Learning

Machine learning adalah cabang kecerdasan buatan (AI) yang memungkinkan sistem untuk belajar dan meningkatkan kinerjanya dari data tanpa diprogram secara eksplisit. Saini et al. (2024a) Algoritma *machine learning* merupakan bagian dari kecerdasan buatan yang memungkinkan komputer untuk belajar dan membuat prediksi atau keputusan tanpa pemrograman eksplisit, dengan menggunakan teknik statistik untuk mengidentifikasi pola dalam data guna melakukan prediksi atau klasifikasi di berbagai aplikasi. Machine learning terbagi menjadi beberapa jenis, seperti *supervised learning*, *unsupervised learning*, *reinforcement learning*, dan *semi-supervised learning*. Menurut (Saini et al., 2024b) Algoritma *machine learning* terbagi menjadi tiga kategori utama: pembelajaran terawasi (*supervised learning*), pembelajaran tak terawasi (*unsupervised learning*), dan pembelajaran penguatan (*reinforcement learning*). Setiap kategori menggunakan teknik yang berbeda untuk belajar dari data, memungkinkan berbagai aplikasi di berbagai bidang seperti pertanian, kesehatan, dan keuangan. Proses pembuatan *machine learning* mencakup pengumpulan dan *preprocessing* data,

pelatihan model, evaluasi kinerja menggunakan metrik seperti RMSE dan MAE, serta penerapan pada data baru. Menurut (Scutari & Malvestio, 2023) Proses pengembangan *machine learning* melibatkan pembuatan infrastruktur perangkat lunak dan pelaksanaan langkah-langkah analisis data. Alur kerja yang bersifat iteratif ini menggabungkan praktik rekayasa perangkat lunak dengan ilmu data, dengan fokus pada pembangunan dan pengoperasian model serta persiapan dan analisis data secara efisien. *Machine learning* diterapkan di berbagai bidang, termasuk pengenalan wajah, deteksi penipuan, analisis pasar, dan penilaian kelayakan produk asuransi.

2.2.3 Pengumpulan Data

Pengumpulan data adalah tahap awal dalam proses analisis yang bertujuan untuk memperoleh informasi relevan yang diperlukan untuk mendukung penelitian atau pengembangan model. Menurut (Salmia, 2023) Proses pengumpulan data melibatkan penggunaan berbagai instrumen penelitian untuk mengumpulkan *field* data yang kemudian diolah baik secara kuantitatif maupun kualitatif. Pengaturan sistematis ini membantu menjawab pertanyaan penelitian atau menguji hipotesis, memastikan validitas dan keandalan data. Dalam konteks ini, proses pengumpulan data dilakukan dari perusahaan asuransi swasta yang memperizinkan penggunaan data *internal* yaitu laporan keuangan yang berisikan informasi sensitif. Pengumpulan data berlangsung pada tahap awal proyek untuk memastikan ketersediaan data yang dibutuhkan sebelum proses analisis dan pengembangan model dilakukan. Data diperoleh dari sistem informasi operasional harian perusahaan. Tujuan utama pengumpulan data adalah untuk memastikan bahwa dataset yang digunakan mencerminkan kondisi aktual perusahaan, relevan dengan kebutuhan analisis, dan cukup representatif untuk mendukung pelatihan model atau pengambilan keputusan perusahaan. Menurut (Gebru et al., 2022) Dataset dalam *machine learning* berfungsi sebagai elemen dasar untuk melatih dan mengevaluasi model. Karakteristiknya memengaruhi perilaku model secara signifikan, sehingga pemilihan dan pemahaman dataset menjadi krusial untuk mencapai transparansi, akuntabilitas, dan reproduktifitas dalam hasil *machine learning*. Proses pengumpulan data melibatkan akses ke sistem internal perusahaan, penyaringan dokumen digital, serta pengumpulan data di dalam perusahaan.

2.2.4 Preprocessing Data

Preprocessing data adalah tahap penting yang bertujuan untuk mempersiapkan dataset agar layak digunakan oleh model *machine learning*. Menurut (Bala & Behal, 2024) Preproses data mencakup berbagai teknik, seperti pembersihan data, normalisasi, pemilihan fitur, dan pengurangan dimensi. Langkah-langkah ini bertujuan untuk meningkatkan kualitas data sekaligus mengoptimalkan kinerja model pembelajaran mesin dan pembelajaran mendalam, sehingga mendukung analisis yang lebih akurat terhadap data mentah dan tidak terstruktur dari beragam bidang. Tahap ini dilakukan dengan memanfaatkan data yang telah dikumpulkan dari perusahaan sebelumnya. Preprocessing dilakukan setelah data dikumpulkan dan sebelum pelatihan model dimulai. Tahap ini diterapkan menggunakan perangkat lunak alat *machine learning*. Tujuan *preprocessing* adalah untuk memastikan kualitas data dapat ditingkatkan melalui proses pembersihan, penyusunan, dan penyesuaian format data agar sesuai untuk analisis, sehingga akurasi hasil dapat ditingkatkan. Langkah-langkah *preprocessing* mencakup penanganan data hilang, penghapusan duplikasi, normalisasi, penghapusan outlier, dan pembagian data menjadi set pelatihan, pengujian, dan validasi untuk memastikan model dapat bekerja secara optimal.

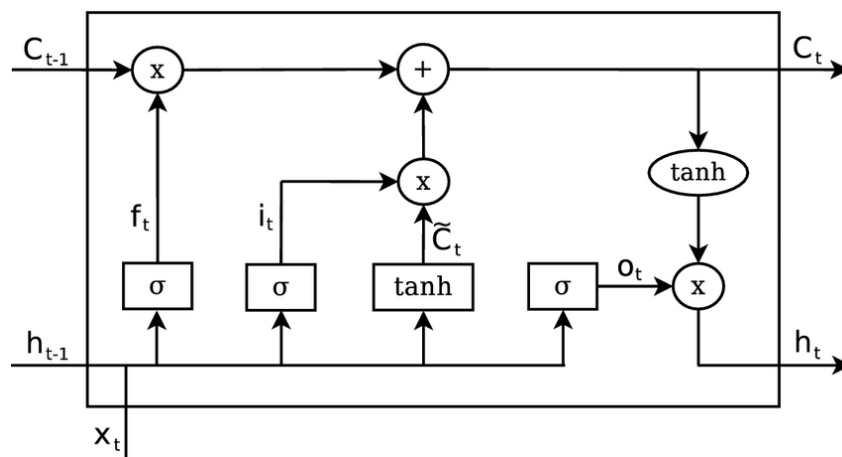
2.2.5 Split Data

Split data adalah proses membagi dataset menjadi beberapa subset untuk mendukung pelatihan dan evaluasi model *machine learning*. Menurut (Oktafiani et al., 2023) Penelitian ini mengkaji rasio pembagian data train-test dari 60:40 hingga 95:05, yang menunjukkan variasi signifikan dalam kinerja model. Penelitian ini menekankan pentingnya memilih rasio yang optimal untuk menyeimbangkan overfitting dan underfitting. Tujuan penggunaan metode pembagian data dalam *machine learning* adalah untuk mengevaluasi kinerja model dengan membagi data menjadi set pelatihan dan pengujian, sehingga menghasilkan identifikasi yang akurat. Pendekatan ini membantu menentukan komposisi data terbaik untuk mencapai prediksi yang optimal. Proses ini dilakukan sebelum pelatihan model dimulai untuk memastikan bahwa model dapat dievaluasi secara objektif dan menghasilkan generalisasi yang baik. Pembagian data diterapkan pada dataset yang telah diproses. Tujuan dari split data adalah untuk memastikan bahwa model dapat

diuji pada data yang tidak pernah digunakan selama pelatihan, sehingga meminimalkan risiko *overfitting* dan meningkatkan keakuratan hasil. Menurut (Sivakumar et al., 2024) Proses pembagian data biasanya dilakukan dengan membagi dataset menjadi tiga subset utama: set pelatihan, validasi, dan pengujian, menggunakan rasio umum seperti 70:20:10 atau 80:20:10. Rasio ini mengalokasikan sebagian besar data untuk pelatihan, sementara data validasi digunakan untuk memastikan model dapat menggeneralisasi dengan baik, dan data pengujian digunakan untuk mengevaluasi kinerja akhir model secara objektif pada data yang belum pernah dilihat sebelumnya.

2.2.6 Metode LSTM

Long Short-Term Memory (LSTM) adalah sebuah jenis model *deep learning* dari kelompok jaringan syaraf tiruan (*Neural Network*). LSTM merupakan hasil modifikasi dari *Recurrent Neural Network* (RNN). Model LSTM dirancang untuk mengelolah data berurutan seperti *time series*, teks, dan suara. Menurut Ravikumar (2023) LSTM merupakan jenis khusus dari RNN yang dirancang untuk secara efisien menangkap ketergantungan jangka panjang dalam data berurutan. LSTM mampu mengatasi masalah *vanishing gradient* dan *exploding gradient* yang sering menjadi kendala pada RNN konvensional. Keterbatasan RNN yang dimodifikasi menjadi LSTM terletak pada arsitektur jaringan dan mekanisme penyimpanan informasi. RNN menggunakan *looping* sederhana, dimana informasi dari langkah sebelumnya langsung diteruskan ke langkah berikutnya. Sehingga, menyebabkan kehilangan dependensi data dalam jangka panjang. Sedangkan kelebihan LSTM menggunakan mekanisme gerbang (*gates*) untuk mengelola informasi, menggunakan informasi, dan melupakan informasi yang disimpan sebelumnya untuk proses selanjutnya. Oleh karena itu, LSTM lebih umum digunakan untuk mengolah data yang lebih kompleks seperti prediksi *time series*, pemrosesan bahasa alami, dan pengenalan suara. Lihat Gambar 2.1 Arsitektur LSTM.



Gambar 2. 1 Arsitektur LSTM

Berdasarkan arsitektur LSTM pada gambar terbagi menjadi tiga jenis gerbang utama yaitu *forget gates*, *input gates*, *output gates*. Gerbang-gerbang tersebut dapat direpresentasikan dalam bentuk matematis sebagai berikut.

a) *Forget Gate*

Forget gate berperan untuk melupakan informasi yang tidak relevan dari data sebelumnya atau sudah berada di memori lama. Sehingga dapat direpresentasikan dalam bentuk matematis sebagai berikut.

$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.1)$$

Keterangan,

- $f_t = Output\ forget\ gate$
- $[h_{t-1}, x_t] = Hidden\ state\ sebelumnya$
- $x_t = Input\ saat\ ini$
- $W_f, b_f = Bobot\ dan\ bias\ forget\ gate$

b) *Input Gate*

Input gate merupakan gerbang yang memutuskan informasi baru apa saja yang perlu ditambahkan ke *cell state*. Sehingga representasi matematis dari *input gate* sebagai berikut.

$$i_t = \sigma (W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.2)$$

$$C_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.4)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C_t \quad (2.5)$$

Keterangan,

- C_t = *Cell state* (Menyimpan Jangka Panjang)
- f_t = *Output forget gate*
- $[h_{t-1}, x_t]$ = *Hidden state* sebelumnya
- i_t = *Input gate*

c) *Output Gate*

Output gate merupakan gerbang akhir yang menghasilkan nilai prediksi dari *hidden state* berdasarkan *cell state* saat ini. Sehingga Representasi matematis dari *output gate* sebagai berikut.

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

$$h_t = o_t \cdot \tanh(C_t) \quad (2.7)$$

Keterangan,

- C_t = *Cell state*
- h_t = *Hidden state*

Dalam penelitian ini LSTM akan diterapkan untuk mengelola data laporan keuangan untuk mengidentifikasi kelayakan produk asuransi. Gerbang-gerbang LSTM seperti *forget gate* akan melupakan informasi historis yang tidak relevan, seperti informasi nilai *loss ratio*, *disconto rate*, *expenses*, *risk adjustment*, dan nilai *CSM* yang tidak berpengaruh pada tren selanjutnya. *Input gate* yang menambah informasi baru dari data laporan keuangan diperiode berikutnya. Lalu, *output gate* yang menghasilkan prediksi *loss ratio* dan komponen nilai *CSM* berdasarkan pola yang telah dipelajari. Sehingga menghasilkan tambahan informasi untuk membantu perusahaan dalam mengelola risiko perusahaan dan menilai kelayakan dari produk asuransi.

2.2.7 Pelatihan Model

Pelatihan model adalah tahapan krusial dalam *machine learning*, dimana model dilatih menggunakan sub data pelatihan dan validasi untuk mempelajari pola dan hubungan dalam dataset. Proses ini dilakukan dengan algoritma *machine learning*. Pelatihan dilakukan setelah preprocessing data selesai dan sebelum pengujian model dilakukan. Menurut (Munde, 2024) Proses pelatihan dalam pembelajaran mesin melibatkan penggunaan data berlabel pada pembelajaran terawasi untuk membangun model prediktif, sedangkan pembelajaran tak terawasi bertujuan mengidentifikasi pola dalam data tanpa label. Langkah-langkah utama meliputi praproses data, rekayasa fitur, dan pemilihan model. Tujuan utama pelatihan model adalah untuk menemukan parameter optimal yang meminimalkan kesalahan prediksi, sehingga menghasilkan model dengan performa yang baik dan mampu memprediksi secara akurat. Proses pelatihan melibatkan input data pelatihan ke dalam model, perhitungan output prediksi, penghitungan error antara prediksi dan nilai sebenarnya. Proses ini diulang secara iteratif hingga model mencapai tingkat akurasi yang diinginkan atau memenuhi kriteria penghentian tertentu. Dalam metode LSTM, pelatihan model melibatkan pengolahan data deret waktu secara berurutan, memanfaatkan mekanisme gates untuk menangkap pola temporal yang kompleks dan menghasilkan prediksi yang lebih akurat.

2.2.8 Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) adalah sebuah metrik pengujian yang telah umum digunakan untuk menguji tingkat hasil akurasi model prediksi atau identifikasi kelayakan produk asuransi. Prinsip mekanisme RMSE adalah dengan menghitung nilai akar rata-rata dari error antara selisih nilai prediksi dengan nilai aktual dalam data, sehingga menciptakan hasil pengujian performa model yang jelas. RMSE efektif digunakan untuk kesalahan yang terdistribusi secara normal, menjadikannya pilihan utama dalam berbagai aplikasi, termasuk peramalan dan analisis regresi. Menurut (Hodson, 2022) *Root Mean Squared Error* (RMSE) adalah akar kuadrat dari *Mean Squared Error* (MSE), yang mewakili kesalahan tipikal untuk kesalahan yang terdistribusi normal. RMSE memberikan metrik dengan satuan yang sama dengan nilai yang diamati, sehingga memudahkan

interpretasi. Perolehan nilai RMSE dapat direpresentasikan dalam bentuk matematika melalui persamaan sebagai berikut.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y})^2} \quad (2.8)$$

Keterangan,

- y_i = Nilai aktual
- \hat{y} = Nilai prediksi dari model
- n = Jumlah data

2.2.9 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) adalah metrik pengujian yang mengukur rata-rata selisih absolut antara nilai prediksi dan nilai sebenarnya, memberikan gambaran tentang seberapa besar kesalahan model secara keseluruhan tanpa memperhitungkan arah kesalahan. Menurut (Robeson & Willmott, 2023) MAE adalah ukuran statistik yang digunakan untuk mengevaluasi akurasi model dengan mengukur rata-rata besar kesalahan dalam serangkaian prediksi, tanpa memperhitungkan arah kesalahan. MAE dihitung sebagai rata-rata dari selisih absolut antara nilai prediksi dan nilai sebenarnya. MAE berguna karena memberikan interpretasi yang jelas tentang besar kesalahan, sehingga lebih mudah dipahami dibandingkan dengan metrik lain seperti Mean Squared Error (MSE) atau Root Mean Squared Error (RMSE). Perolehan nilai MAE dapat direpresentasikan dalam bentuk matematika melalui persamaan sebagai berikut.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.9)$$

Keterangan,

- y_i = Nilai aktual
- \hat{y}_i = Nilai prediksi dari model
- n = Jumlah data

2.2.10 Deployment Model

Deployment model adalah tahap akhir dalam siklus pengembangan model *machine learning*, di mana model yang telah diuji dan disempurnakan diimplementasikan ke dalam aplikasi. Proses ini berlangsung dalam tahap pengembangan interaksi pengguna untuk memastikan bahwa model dapat memproses data dan memberikan hasil yang sesuai. Deployment dilakukan setelah model berhasil melalui tahap pengujian prototyping, dan siap untuk diintegrasikan ke dalam sistem operasional perusahaan. Tujuan utama dari deployment model adalah untuk menyediakan solusi berbasis *machine learning* yang dapat mendukung identifikasi kelayakan produk asuransi dan memastikan model menghasilkan output yang sesuai. Dalam konteks identifikasi kelayakan produk asuransi, deployment model mencakup integrasi model *Long Short-Term Memory* (LSTM) ke dalam bentuk aplikasi website dan memungkinkan model memproses data seperti *loss ratio*, *discount rate*, *expenses*, dan *risk adjustment* untuk memberikan rekomendasi kelayakan produk. Proses deployment melibatkan pengemasan model ke dalam aplikasi website yang mampu mengidentifikasi kelayakan produk asuransi. Dengan deployment yang berhasil, perusahaan asuransi dapat menggunakan model ini untuk meningkatkan efisiensi operasional, membuat keputusan yang lebih tepat, dan memaksimalkan keuntungan melalui pengelolaan produk asuransi yang lebih baik.

2.2.11 Metode Konvensional

Metode konvensional merujuk pada pendekatan manual dalam analisis data, seperti penggunaan Microsoft Excel atau Tableau untuk pengolahan dan visualisasi data. Proses ini umumnya dilakukan oleh analis data atau staf keuangan di perusahaan asuransi. Metode ini digunakan sejak lama dalam aktivitas operasional dan pelaporan. Penggunaannya masih lazim di berbagai sektor dan diterapkan di sistem lokal perusahaan. Tujuan utama metode konvensional adalah menyediakan solusi sederhana untuk analisis dasar atau pelaporan. Namun, metode konvensional mempunyai keterbatasan yaitu memakan waktu, sumber daya teknologi yang cukup tinggi, dan proses yang dilakukan secara manual lainnya. Hal tersebut yang menjadi kelemahan saat menganalisis kelayakan produk asuransi dengan analisa konvensional.

2.2.12 Data Laporan Keuangan Perusahaan

Data laporan keuangan perusahaan mencakup informasi operasional seperti pendapatan premi, klaim yang dibayarkan, biaya operasional, dan laba bersih, yang digunakan untuk analisis kelayakan produk asuransi. Data ini disediakan oleh pihak perusahaan asuransi. Pengumpulan dan analisis data dilakukan secara berkala sesuai dengan operasional yang ditentukan oleh perusahaan. Data ini diambil dari sistem internal perusahaan, seperti sistem akuntansi atau basis data keuangan. Tujuan utamanya adalah untuk menyediakan informasi yang akurat dan relevan bagi analisis dan pengambilan keputusan. Prosesnya melibatkan ekstraksi data, pengolahan, dan integrasi dengan model analitik untuk mendukung penilaian kelayakan produk asuransi.

2.2.13 Faktor-faktor penentu Kelayakan Produk Asuransi

Faktor-faktor penentu kelayakan produk asuransi mencakup elemen-elemen seperti *loss ratio*, *discount rate*, *expenses*, *risk adjustment* yang menjadi indikator utama dalam menilai kelayakan suatu produk. Faktor-faktor ini umumnya dianalisis oleh tim aktuaria. Evaluasi dilakukan secara berkala untuk penyesuaian atau selama fase pengembangan produk baru. Data yang digunakan diambil dari laporan keuangan perusahaan asuransi. Tujuannya adalah untuk memastikan bahwa produk asuransi memberikan manfaat optimal bagi nasabah sekaligus menguntungkan bagi perusahaan. Analisis faktor-faktor ini melibatkan pendekatan statistik dan *machine learning*, seperti model berbasis LSTM, untuk mengidentifikasi pola dan hubungan antarparameter untuk menentukan kelayakan produk asuransi.

2.2.14 *Contract Service Margin (CSM)*

Contract Service Margin (CSM) adalah margin keuntungan yang belum direalisasikan dari kontrak asuransi, dihitung saat pengakuan awal kontrak dan diakui secara bertahap selama masa layanan. CSM penting untuk menjaga transparansi dan sistematisasi pengakuan pendapatan, memastikan kepatuhan terhadap standar akuntansi seperti IFRS 17 atau PSAK 117. CSM dihitung dengan rumus:

$$CSM = \text{Gross Premi} - (\text{Nilai Gross Premi saat ini} + \text{Expenses} + \text{Expenses})$$

(2.10)

Contoh perolehan nilai CSM dapat ditentukan melalui nilai klaim saat ini yang diperoleh dari diskonto arus kas masa depan, *risk adjustment* dihitung sebagai persentase dari klaim, dan *expenses* dihitung dari persentase premi. Sebagai contoh, jika premi Rp 1.000.000, nilai kini klaim Rp 614.000, risk adjustment Rp 36.840, dan biaya operasional Rp 150.000, maka CSM sebesar Rp 199.160. CSM digunakan oleh perusahaan asuransi untuk menilai profitabilitas produk dan memastikan kelayakan berdasarkan nilai ekonomi masa depan.

2.2.15 Identifikasi Kelayakan Produk Asuransi

Identifikasi kelayakan produk asuransi adalah proses evaluasi untuk menentukan apakah suatu produk asuransi layak ditawarkan kepada pasar berdasarkan analisis parameter tertentu seperti risiko, profitabilitas, dan kesesuaian dengan kebutuhan konsumen. Proses ini dilakukan oleh perusahaan asuransi, aktuaria, dan analis risiko untuk memastikan produk yang ditawarkan memberikan nilai tambah bagi perusahaan dan nasabah. Evaluasi dilakukan pada tahap pengembangan produk atau secara berkala selama siklus hidup produk, sesuai dengan kebutuhan perusahaan atau perubahan kondisi dunia bisnis asuransi salah satunya adalah regulasi. Analisis ini diterapkan pada produk asuransi di berbagai sektor, seperti kesehatan, kendaraan, jiwa, dan properti, baik di tingkat lokal maupun global. Tujuan utama identifikasi ini adalah untuk memastikan bahwa produk yang ditawarkan dapat memenuhi kebutuhan pasar, memberikan keuntungan finansial, serta mematuhi regulasi yang berlaku. Prosesnya melibatkan analisis data historis, penggunaan model prediksi risiko, dan penilaian parameter seperti loss ratio, discount rate, dan expenses, dengan memanfaatkan teknologi seperti machine learning untuk melerai hambatan pada metode konvensional.

2.2.17 PSAK 117

Pernyataan Standar Akuntansi Keuangan (PSAK) 117 adalah standar akuntansi keuangan yang mengatur perlakuan akuntansi untuk kontrak asuransi, bertujuan untuk meningkatkan transparansi, akuntabilitas, dan konsistensi laporan keuangan perusahaan asuransi. Standar ini berlaku bagi perusahaan asuransi dan diatur oleh Otoritas Jasa Keuangan (OJK) di Indonesia. Implementasinya dimulai sejak PSAK ini diberlakukan secara resmi untuk laporan keuangan tahun berjalan. PSAK 117 diterapkan dalam seluruh aspek pelaporan keuangan perusahaan

asuransi di Indonesia. Tujuan utama standar ini adalah untuk memastikan laporan keuangan perusahaan asuransi memberikan gambaran yang wajar mengenai posisi keuangan dan kinerja operasionalnya. Proses penerapannya mencakup perubahan metode pencatatan, pengukuran, dan pengungkapan yang sesuai dengan ketentuan yang diatur dalam PSAK 117, termasuk perlakuan terhadap aset dan liabilitas kontrak asuransi.

2.2.18 Bahasa Pemrograman Python

Dalam melakukan penelitian digunakan bahasa pemrograman Python sebagai bahasa pengembangan aplikasi. Bahasa pemrograman Python mempunyai beberapa keunggulan yaitu open source, kode skrip yang lebih sederhana dan high language, dan kumpulan fungsi library yang tersedia, siap pakai, dan dapat disesuaikan dengan kebutuhan sistem. Sintaks kode didukung oleh berbagai fungsionalitas dari perpustakaan yang luas dan komprehensif. Selain itu, tersedia banyak perpustakaan tambahan yang mempermudah pengguna dalam menulis kode sumber. (Justica et al., 2022).

2.2.18.1 Library Numpy

Numpy merupakan salah satu pustaka yang umum digunakan dalam bahasa pemrograman Python. Pustaka ini menyediakan beragam fungsi matematis untuk pengolahan angka, dan termasuk bilangan bulat, matriks (Nofiyanti & Oki Nur Haryanto, 2021). Numpy sering digunakan sebagai alat untuk melakukan komputasi ilmiah yang melibatkan manipulasi data dalam skala besar. Dikenal karena kemampuannya dalam perhitungan matriks dan pengolahan data yang efisien. Sehingga Numpy berkontribusi dalam proses pengembangan model machine learning untuk memprediksi keuangan perusahaan di masa depan.

2.2.18.2 Library Matplotlib

Matplotlib merupakan sekumpulan berbagai fungsi yang dibuat untuk mendukung penggambaran data dalam dua dimensi atau tiga dimensi. (Reynaldi et al., 2021). Matplotlib mendukung program Python untuk melakukan yang berkaitan dengan gambar, plot, dan format penyimpanan. Matplotlib dapat memberikan visualisasi berdasarkan hasil dari prediksi model *machine learning*. Hal tersebut dapat memudahkan pengguna aplikasi menarik simpulan dari hasil prediksi yang didapatkan.

2.2.18.3 Library Pandas

Pandas merupakan perpustakaan Python yang menawarkan struktur data serta alat analisis data yang mudah digunakan dan memiliki kinerja tinggi. Pustaka ini banyak digunakan untuk tugas-tugas manipulasi data seperti pembersihan, analisis, dan persiapan data untuk mengembangkan machine learning. Pandas mempunyai dua komponen utama yaitu series dan data frame. Sebuah series adalah array berlabel satu dimensi yang mampu menyimpan data dari jenis apa pun, sementara data frame adalah struktur data dua dimensi berlabel yang menyerupai tabel atau lembar kerja, terdiri dari baris dan kolom.

2.2.18.4 Library Flask

Flask merupakan sebuah framework yang dikembangkan dengan bahasa dasar yaitu python dalam menjawab kerpeluan pengembangan aplikasi web. Framework ini terkenal karena kesederhaannya dan kemudahan dalam melakukan integrasi model yang telah dikembangkan. Menurut Aryudha Hattu et al. (2024) Flask menawarkan keunggulan dalam hal kesederhanaan, fleksibilitas, dan skalabilitas, sehingga menjadi kerangka kerja yang ideal untuk pengembangan sistem aplikasi operasional. Dengan sifatnya yang ringan, Flask memungkinkan kustomisasi yang mudah serta mendukung implementasi konsep seperti Flask Overriding, yang berkontribusi pada peningkatan fungsionalitas dan adaptabilitas sesuai dengan kebutuhan spesifik dalam konteks bisnis. Dalam penelitian ini Flask digunakan untuk media pengembangan website yang akan berinteraksi dengan model *machine learning* yang telah dikembangkan.

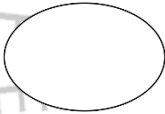

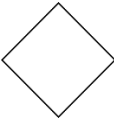

2.2.19 Website

Website adalah sekumpulan halaman yang tersedia di internet dan berfungsi untuk menyajikan informasi, konten, atau layanan dalam berbagai format, seperti teks, gambar, video, atau elemen interaktif lainnya. Tujuan utama website adalah mempermudah penyampaian informasi dan mendukung kolaborasi atau layanan. Website dibangun menggunakan HTML sebagai kerangka, CSS untuk estetika, dan JavaScript untuk membuatnya interaktif. Website dapat diakses melalui berbagai perangkat, seperti desktop, tablet, dan ponsel, namun penelitian ini berfokus pada optimalisasi penggunaan di desktop.

2.2.20 Flowchart

Flowchart merupakan gambaran visual dari proses kegiatan yang saling berhubungan.. Menurut Gomes et al., (2020), diagram alir sering digunakan untuk merepresentasikan aliran sebuah proses algoritma. Sedangkan Menurut Hermawan et al., (2019), diagram ini dapat memberikan solusi secara bertahap yang terdapat dalam suatu prosedur atau algoritma. Diagram tersebut dapat menggambarkan solusi yang terstruktur mengikuti proses yang terdapat pada sistem. *Flowchart* mempunyai beragam simbol dan masing-masing mempunyai kegunaan yang berbeda tergantung proses alur informasi yang ingin dibuat. Kegunaan lain dari diagram alur adalah untuk mendokumentasikan tahapan dan rencana dalam sebuah penelitian. Berikut adalah beberapa jenis *flowchart* dan penjelasannya yang digunakan. Lihat Tabel 2.2 *Flowchart*.

Tabel 2. 2 Flowchart

Simbol	Keterangan
	Simbol <i>Terminator</i> Simbol yang menjukan permulaan atau penutupan dari suatu fase atau aktivitas sistem.
	Simbol <i>Processing</i> Simbol yang merepresentasikan bahwa sebuah proses sudah usai atau telah dilaksanakan dalam sistem.
	Simbol <i>Decision</i> Berfungsi untuk menentukan pilihan berdasarkan kondisi-kondisi yang tersedia.
	Simbol <i>Input-Output</i> Simbol Input-Output merupakan representasi visual yang diartikan sebagai proses penerimaan data yang masuk serta pengeluaran data pada sistem.

2.2.21 Unified Modelling Language (UML) Diagram

Unified Modeling Language (UML) merupakan suatu pendekatan atau kerangka kerja dalam proses perancangan sistem berbasis komputer atau objek. Hal ini memberikan kejelasan terkait informasi, mendokumentasikan, dan memahami

susunan serta karakteristik yang dikembangkan (Anwar et al., 2020). UML umumnya digunakan sebagai alat bantu untuk mempermudah pengembangan perangkat lunak secara sistematis, terstruktur, dan efektif. Dalam penelitian ini, beberapa langkah penerapan UML dilakukan sebagai berikut.


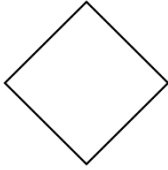
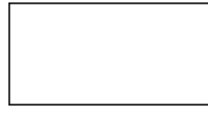
a) Use Case Diagram

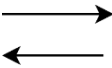

Penggunaan diagram *usecase* bertujuan untuk merepresentasi bentuk visual dari sebuah informasi dengan menggambarkan bagaimana aplikasi dapat berinteraksi dengan pengguna. Menurut Setiyati et al. (2021) Diagram ini membantu dalam tahap pengembangan perangkat lunak.

b) Activity Diagram

Activity diagram adalah representasi pemodelan sistem yang dirancang untuk menunjukkan alur sistematis dari aktivitas yang terjadi antara pengguna dan aplikasi dalam bentuk diagram. Diagram ini sering digunakan dalam pengembangan sistem aplikasi untuk menggambarkan urutan proses atau alur kerja aplikasi secara keseluruhan. Menurut Nistrina et al. (2022) beberapa simbol dalam *Activity* diagram digunakan untuk melihat sistem, dengan penjelasan sebagai berikut. Lihat Tabel 2.3 *Activity* Diagram.

Tabel 2. 3 Activity Diagram





Simbol	Keterangan
	Simbol Status Awal Simbol bulat berwarna hitam ini menandakan dimulainya suatu kegiatan.
	Simbol <i>Decision</i> Simbol yang berfungsi untuk pemilihan dari keadaan yang ada.
	Simbol Proses Simbol tersebut digunakan untuk menjalin komunikasi antara sistem-sistem yang terlibat.


	<p>Simbol Panah</p> <p>Simbol tersebut digunakan untuk menunjukkan arah dari proses yang berlangsung.</p>
	<p>Simbol <i>Association</i></p> <p>Simbol yang menandakan akhir dari hubungan komunikasi antar objek pada suatu aktivitas.</p>

c) *Sequence Diagram*

Sequence diagram merupakan tipe diagram dalam *Unified Modeling Language* (UML) yang ditujukan untuk memvisualisasikan dan menjelaskan keterlibatan antara komponen yang terdapat pada sistem aplikasi dan masih dikembangkan. Diagram ini menunjukkan hubungan antara objek-objek tersebut selama proses sistem berlangsung, yang digambarkan melalui berbagai simbol dan elemen visual. Penjelasan mengenai simbol-simbol yang diterapkan dilihat pada tabel 2.4 *Sequence Diagram*.

Tabel 2. 4 *Sequence Diagram*

Simbol	Keterangan
	<p>Simbol <i>Actor</i></p> <p>Simbol yang menunjukkan partisipasi dalam interaksi dengan sistem.</p>
	<p>Simbol <i>Lifeline</i></p> <p>Simbol yang digunakan untuk mengartikan pelaksanaan dalam suatu eksekusi dalam objek.</p>
	<p>Simbol <i>Time Active</i></p> <p>Simbol yang memungkinkan untuk menandai suatu interaksi sebagai aktif.</p>
	<p>Simbol <i>Message</i></p>

	Simbol ini berguna untuk menunjukkan aktivitas yang terjadi di antara objek dalam suatu interaksi.
	Simbol <i>Return Message</i> Simbol ini berguna untuk mengembalikan pesan objek yang akan menerimanya.

