BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Berikut merupakan penelitian terdahulu yang diambil sebagai referensi dan juga sebagai panduan dalam melakukan penelitian ini untuk mendapatkan referensi yang membahas mengenai prediksi *sales*, algoritma prediksi terbaik, dan penggunaan *dashboard* dalam bisnis.

Tabel 2.1 Daftar Penelitian Terdahulu

No.	Judul Penelitian	Tahun	Penulis	Jurnal	Algoritma	Kesimpulan
1	**Prediksi Harga Saham Syariah Menggunakan Algoritma *Long Short-Term Memory* (LSTM)** [12]	2022	Gunawan Budiprasetyo, Darin Zahira Aflah, Mamluatul Hani'ah	Jurnal Nasional Teknologi & System Informasi	LSTM	1) Penelitian menggunakan metode *Long Short-Term Memory* (LSTM) untuk memprediksi harga saham syariah. 2) Normalisasi data harga penutupan dilakukan sebelum pemodelan untuk meningkatkan akurasi prediksi. 3) Pengujian dilakukan dengan berbagai kombinasi parameter (layers, epoch, time step) untuk menyesuaikan pola data masing-masing saham. 4) Hasil menunjukkan nilai MAPE rendah, menunjukkan prediksi yang cukup akurat: a. Aneka Tambang Tbk: 2,64% b. Erajaya Swasembada Tbk: 2,24% c. Kalbe Farma Tbk: 1,51% d. Semen Indonesia Tbk: 1,83%

						e. Wijaya Karya Tbk: 2,66% 5) Model LSTM terbukti efektif dalam memprediksi harga saham syariah. 6) Rekomendasi untuk penelitian lanjutan: mempertimbangkan faktor eksternal (seperti kondisi ekonomi dan politik) yang mempengaruhi fluktuasi harga saham. 7) Disarankan untuk meningkatkan akurasi model dengan menggabungkan LSTM dan ensemble methods dalam pendekatan Deep Learning.
2	**Penerapan Algoritma *Random Forest* Untuk Prediksi Penjualan Dan Sistem Persediaan Produk** [13]	2024	Muhammad Syahrul Efend, Akhmad Khanif Zyen, Sarwido	Resolusi : Rekayasa Teknik Informatika dan Informasi	*Random Forest*	1) Algoritma *Random Forest* digunakan untuk memprediksi penjualan dan mengelola persediaan produk *Bolen Crispy* di Desa Pekalongan. 2) Model berhasil meningkatkan efisiensi bisnis dengan tingkat akurasi mencapai 85%. 3) Prediksi penjualan mingguan membantu pelaku usaha menghindari kelebihan/kekurangan stok dan mendukung pengambilan keputusan strategis berbasis data. 4) Keterbatasan penelitian mencakup ukuran dataset yang terbatas dan kurangnya variabel tambahan untuk peningkatan akurasi. 5) Secara keseluruhan, *Random Forest* menunjukkan potensi besar dalam meningkatkan daya saing dan pertumbuhan bisnis di wilayah pedesaan.
3	**Analisis dan Prediksi Data Penjualan Menggunakan**	2022	Ferdy Riza	Jurnal Data Science Indonesia	*XGBoost* *XGBoost*	1) Penelitian menggunakan *supervised machine learning* untuk memprediksi penjualan di berbagai lokasi Big Mart.

	***Machine learning* dengan Pendekatan Ilmu Data** [14]					2) *XGBoost* dan *XGBoost* terbukti memiliki tingkat kesalahan terendah berdasarkan evaluasi dengan MAE dan RMSE. 3) *XGBoost* mencapai skor R^2 tertinggi sebesar 0,61, diikuti oleh *XGBoost* dengan skor 0,60. 4) Terdapat korelasi signifikan antar atribut yang memengaruhi penjualan; lokasi berukuran menengah menghasilkan penjualan tertinggi. 5) Meskipun hasil sudah baik, diperlukan optimasi lebih lanjut untuk meningkatkan akurasi prediksi. *6)* Rekomendasi metode optimasi model di masa depan: *Grid Search, Random Search,* dan *Bayesian Optimization*.
4	***Sales* Forecasting using *Machine learning*** [15]	2023	Harsh Goel, Himanshi Dwivedi, Prithvi Krishna Prasad, Rohan M, Vidya R	International Journal of Advance Research and Innovative Ideas in Education	*XGBoost*	1) Proyek mengembangkan model prediksi penjualan menggunakan algoritma *XGBoost*. 2) Model menunjukkan efektivitas sangat tinggi dengan nilai R^2 sebesar 0.999, menandakan kemampuan luar biasa dalam menangkap pola data. 3) Model. memberikan wawasan akurat yang mendukung pengambilan keputusan bisnis 4) Evaluasi lanjutan tetap diperlukan untuk menguji keandalan model terhadap data baru (generalisasi). 5) Secara keseluruhan, proyek ini menghasilkan alat prediksi penjualan yang kuat dan aplikatif.

5	**Prediksi Jumlah Kasus COVID-19 Menggunakan Metode *Auto Regressive Integrated Moving Average* (ARIMA) (studi kasus Kabupaten Sidoarjo)** [16]	2021	Lailatul Ainiyah, Muflihah Bansori	Jurnal Sains Dasar	ARIMA	1) Data telah bersifat stasioner terhadap ragam setelah transformasi, tetapi tidak stasioner terhadap mean berdasarkan analisis autokorelasi. 2) Proses differencing dilakukan dua kali agar data menjadi stasioner terhadap mean, dikonfirmasi melalui *Partial Autocorrelation*. 3) Beberapa model ARIMA diuji dengan pendekatan *trial* and *error* untuk mencari nilai MSE terkecil. 4) Model terbaik untuk total kasus positif COVID-19 adalah ARIMA (2,2,1) dengan MSE = 1540,51. 5) Model terbaik untuk total kasus pasien sembuh COVID-19 adalah ARIMA (3,1,2) dengan MSE = 526,81. 6) Model ARIMA yang diperoleh mampu menghasilkan prediksi yang mendekati data aktual, menunjukkan keefektifannya dalam analisis dan peramalan tren COVID-19.
6	**A comparative study of automobile *Sales* forecasting with ARIMA, SARIMA and deep learning LSTM model** [17]	2022	Sharath Kariya Shetty, Rajesh Bukta	International Journal Advanced Opereations Management	ARIMA SARIMA LSTM	1) LSTM unggul dalam prediksi penjualan otomotif, mengungguli ARIMA dan SARIMA: a. Akurasi LSTM meningkat 92% dibanding ARIMA. b. Akurasi LSTM meningkat 42,5% dibanding SARIMA. c. Dengan nilai MAE LSTM (58,58), ARIMA (1,081.99), SARIMA (131.69) 2) Pemilihan *hyperparameter* yang optimal menjadi kunci keberhasilan model LSTM. 3) LSTM dapat meningkatkan akurasi prediksi dengan menambahkan variabel eksternal seperti: a. Harga minyak

						b. Indeks saham c. GDP 4) LSTM memiliki kemampuan mengintegrasikan analisis sentimen, yang tidak dimiliki oleh model *time series* tradisional. 5) Prediksi yang lebih akurat membantu perusahaan meminimalkan risiko kerugian dan mengoptimalkan strategi bisnis.
7	**Implementasi Metode Long-Short Term Memory Untuk Memprediksi Pergerakan Nilai Harga Emas** [18]	2022	Michael Owen, Vincent, Riama Br Ambarita, Evta Indra	Jurnal Tekinkom	LSTM	1) Algoritma LSTM berhasil memprediksi pergerakan harga emas di pasar Indonesia secara efektif. 2) Model menggunakan variabel prediktor: IHSG, harga minyak mentah, dan nilai tukar USD/IDR. 3) Evaluasi menunjukkan akurasi tinggi, berdasarkan metrik: a. MSE: 76.237 Rupiah²/gram² b. RMSE: 8.566 Rupiah/gram c. MAPE: 0,66% 4) Dataset yang digunakan mencakup periode 1 Januari 2018 – 31 Maret 2023; performa terbaik dicapai setelah 500 epoch pelatihan. 5) Validasi menggunakan *k-Fold Cross Validation* dan *Time series* Split menunjukkan adanya variasi akurasi tergantung metode pemisahan data. 6) Penelitian memberikan wawasan penting bagi sektor keuangan, khususnya dalam prediksi harga emas. 7) Perlu evaluasi lanjutan terhadap hasil prediksi, mengingat adanya faktor eksternal yang dapat memengaruhi harga. 8) Untuk peningkatan model di masa depan, disarankan menambahkan variabel prediktor dan memperluas cakupan dataset.

8	**Implementasi Intelejen Bisnis dengan Visualisasi Data Gaji dan Algoritma Linear Regresion** [19]	2024	Haryadi Tri Nugroho (2024)	Jurnal Indonesia Manajemen Informatika dan Komunikasi	Regresi Linear	1) Regresi linier digunakan untuk memprediksi *salary*, dengan hasil R-*squared* sebesar 0,263429, yang menunjukkan kemampuan prediksi masih terbatas. 2) Model memiliki potensi untuk ditingkatkan melalui: a. Penambahan data training, b. Optimasi parameter, c. Penggunaan model *machine learning* yang lebih kompleks. 3) *Dashboard* visualisasi masih memiliki kekurangan, yang perlu diperbaiki untuk meningkatkan kualitas analisis. 4) Penelitian lanjutan disarankan memperbarui proses training dengan dataset yang lebih besar guna meningkatkan akurasi. 5) Hasil penelitian ini dapat menjadi referensi awal bagi perusahaan dalam pengambilan keputusan berbasis data.
9	**Implementasi Framework Streamlit Sebagai Prediksi Harga Jual Rumah Dengan Linear Regresi** [20]	2023	Gita Ayu Syafarina, Zaenuddin	Media Teknologi Informasi dan Komputer Jurnal	*Linear Regresi Streamlit*	1) Penelitian menghasilkan aplikasi prediksi harga jual rumah berbasis AI menggunakan *framework* Streamlit. 2) Metode yang digunakan adalah *Linear Regression*, dengan antarmuka yang sederhana dan mudah digunakan. 3) Model dilatih menggunakan data historis harga rumah Kota Banjarmasin dan dievaluasi dengan metrik: a. *Mean Absolute Error* (MAE) b. *Mean Squared Error* (MSE) 4) Model menunjukkan akurasi sebesar 67,8%.

						5) Aplikasi diuji menggunakan data aktual yang tidak termasuk dalam pelatihan, menghasilkan estimasi harga yang cukup akurat dan kesalahan yang rendah. 6) Aplikasi ini berpotensi menjadi alat bantu efektif untuk perkiraan harga properti di Kota Banjarmasin.
10	**Rancang Bangun Aplikasi Prediksi Penjualan Pintu Baja dengan Metode Quadratic Trend (Studi Kasus PT. Jaya Bersama Saputra Perkasa)** [21]	2024	Triandi B	Journal of Software Engineering, Computer Science and Information Technology	*Least Squares*	1) Permasalahan Kios Tinta adalah ketidakteraturan pengelolaan stok dan pemenuhan permintaan konsumen yang tidak optimal. 2) Penyebab utama adalah belum adanya sistem prediksi penjualan yang akurat dan proses pencatatan stok yang masih manual. 3) Kondisi ini menyebabkan sering terjadi kekurangan atau penumpukan barang, yang mengganggu kelancaran penjualan. 4) Penelitian merancang sistem prediksi penjualan menggunakan metode *Least Squares* (deret waktu) untuk memanfaatkan data penjualan masa lalu. 5) Sistem ini dapat membantu Kios Tinta dalam memperkirakan kebutuhan stok bulanan dengan lebih tepat. 6) Diharapkan penerapan metode ini dapat: a) Meningkatkan efisiensi operasional, b) Mengurangi risiko kelebihan/kekurangan stok, c) Mendukung pengambilan keputusan lebih baik dalam pengelolaan persediaan.

Keseluruhan penelitian terdahulu membahas berbagai metode prediksi dalam domain keuangan, penjualan, dan properti menggunakan algoritma *machine learning* dan deep learning. Fokus utama adalah penerapan *Long Short-Term Memory* (LSTM), *Random Forest*, *XGBoost*, ARIMA, dan regresi linier untuk meningkatkan akurasi prediksi dan mendukung pengambilan keputusan berbasis data.

Dalam prediksi harga saham syariah, model LSTM digunakan setelah proses normalisasi data. Pengujian dengan berbagai kombinasi parameter menunjukkan bahwa model ini mampu memberikan prediksi yang cukup akurat dengan nilai *Mean Absolute Percentage Error* (MAPE) yang bervariasi antar emiten [12]. Namun, penelitian lebih lanjut diperlukan untuk mempertimbangkan faktor eksternal yang memengaruhi harga saham serta meningkatkan akurasi dengan metode ensemble.

Dalam bidang penjualan dan manajemen persediaan, algoritma *Random Forest* diterapkan untuk memprediksi penjualan *Bolen Crispy* di Desa Pekalongan, meningkatkan efisiensi bisnis dengan akurasi 85%, yang membantu menghindari kelebihan dan kekurangan stok. Sementara itu, *XGBoost* dan *XGBoost* digunakan dalam analisis data penjualan *Big Mart*, di mana *XGBoost* mencapai skor R^2 tertinggi sebesar 0,61, serta mengungkap korelasi antar atribut yang memengaruhi penjualan. Selain itu, *XGBoost* juga digunakan dalam proyek prediksi penjualan dengan efektivitas tinggi (R^2 = 0,999), meskipun evaluasi lebih lanjut masih diperlukan terhadap data baru [14].

Dalam prediksi tren kasus COVID-19, ARIMA diterapkan dengan analisis kestasioneran data yang menunjukkan perlunya differencing dua kali agar data stasioner terhadap mean. Model terbaik untuk memprediksi total kasus pasien COVID-19 adalah ARIMA(2,2,1) dengan MSE sebesar 1540,51, sedangkan untuk kasus pasien sembuh, model terbaik adalah ARIMA(3,1,2) dengan MSE sebesar 526,81. Hasil ini menunjukkan bahwa ARIMA efektif dalam peramalan tren kasus COVID-19 [16].

Dalam prediksi otomotif dan pergerakan harga emas, LSTM menunjukkan keunggulan dengan peningkatan akurasi hingga 92% dibandingkan ARIMA dan 42,5% dibandingkan SARIMA dalam prediksi penjualan otomotif [17]. Model ini juga dapat mengintegrasikan analisis sentimen untuk meningkatkan akurasi. Dalam prediksi harga emas, LSTM menggunakan variabel seperti Indeks Harga Saham Gabungan (IHSG), harga minyak mentah, dan nilai tukar USD/IDR. Dengan dataset dari 2018 hingga 2023, model ini mencapai MAPE sebesar 0,66%, menunjukkan performa yang sangat baik dalam memodelkan pergerakan harga emas[18].

Penggunaan *dashboard* dalam implementasi intelijen bisnis terbukti menjadi alat bantu penting dalam menyajikan hasil prediksi secara visual dan informatif. Dalam beberapa penelitian, *dashboard* visualisasi digunakan untuk menampilkan hasil prediksi gaji menggunakan regresi linier, meskipun masih memiliki beberapa kekurangan yang perlu disempurnakan agar analisis lebih optimal [19]. Dalam prediksi harga properti, aplikasi AI berbasis Streamlit dikembangkan untuk memprediksi harga jual rumah di Kota Banjarmasin menggunakan regresi linier. Model ini mencapai akurasi sebesar 67,8%, memberikan estimasi harga properti yang cukup akurat bagi pengguna [20]. Dengan demikian, *dashboard* berperan penting dalam menjembatani hasil analisis prediktif dan penerapannya dalam konteks bisnis nyata.

Secara keseluruhan, penelitian terdahulu menunjukkan bahwa model *machine learning* dan deep learning memiliki potensi besar dalam meningkatkan akurasi prediksi di berbagai domain. LSTM unggul dalam prediksi time-series, terutama dalam data yang kompleks dan non-linear, sementara *XGBoost* dan *Random Forest* efektif dalam analisis data penjualan. ARIMA tetap relevan untuk peramalan tren pada data yang memiliki pola musiman dan kestasioneran yang terjaga, seperti pada kasus COVID-19.

Model ARIMA, sebagai metode statistik tradisional, bekerja dengan baik pada data yang bersifat stasioner dan memiliki pola musiman atau tren yang dapat dimodelkan secara linear. Namun, model ini memiliki keterbatasan dalam menangani pola non-linear dan data yang sangat fluktuatif, sehingga performanya

menurun ketika diterapkan pada data yang memiliki dinamika kompleks, seperti pergerakan harga saham, harga emas, atau penjualan dengan faktor eksternal yang bervariasi.

Di sisi lain, LSTM sebagai bagian dari Deep Learning mampu menangani pola non-linear, ketergantungan jangka panjang (*long-term dependencies*), serta volatilitas data dengan lebih baik. Model ini menunjukkan peningkatan akurasi yang signifikan dibandingkan ARIMA dalam berbagai studi kasus, seperti:

1) Prediksi harga saham syariah, di mana LSTM memberikan nilai MAPE lebih rendah dibandingkan ARIMA [12].
2) Prediksi harga emas di Indonesia, dengan MSE dan RMSE yang lebih kecil dibandingkan metode statistic [18].
3) Prediksi penjualan otomotif, di mana LSTM menunjukkan peningkatan akurasi hingga 92% dibandingkan ARIMA [17].

Namun, meskipun LSTM lebih unggul dalam banyak aspek, model ini juga memiliki beberapa tantangan, seperti waktu komputasi yang lebih tinggi, kebutuhan dataset yang besar, dan *tuning hyperparameter* yang kompleks. Sebaliknya, ARIMA lebih cepat dan lebih mudah diinterpretasikan, sehingga masih relevan

Dengan demikian, penelitian kali ini memanfaatkan algoritma LSTM, ARIMA, dan SARIMA untuk melakukan prediksi penjualan berbasis data *time-series*. Hal ini didukung oleh masing-masing model yang memiliki keunggulan tersendiri. ARIMA dan SARIMA efektif untuk data yang bersifat stasioner dan memiliki pola musiman, sementara LSTM unggul dalam menangkap pola non-linear dan ketergantungan jangka panjang.

Sebagai pengembangan lebih lanjut, penelitian ini juga mencoba menerapkan pendekatan *Hybrid* antara LSTM-ARIMA dan LSTM-SARIMA dengan tujuan menggabungkan kekuatan model statistik dalam mengenali tren linear dan pola musiman, serta kemampuan LSTM dalam memahami dinamika data yang kompleks. Pendekatan *Hybrid* ini belum banyak dibahas dalam penelitian

terdahulu, sehingga menjadi kontribusi baru dalam pengembangan model prediksi yang lebih akurat dan adaptif terhadap perubahan data penjualan.

2.2 Teori tentang Topik Skripsi

2.2.1 Teori Data

Data merupakan elemen fundamental yang berfungsi sebagai representasi fakta atau gambaran suatu peristiwa yang terjadi dalam kurun waktu tertentu. Secara umum, data dapat diartikan sebagai kumpulan informasi yang diperoleh melalui proses pengamatan atau pengukuran, namun masih dalam bentuk mentah dan belum memiliki makna langsung bagi pengguna [22].

Menurut Hoffer et al. (2004), data merujuk pada sesuatu yang merepresentasikan objek maupun peristiwa yang memiliki nilai signifikan bagi pemakainya. Dalam bidang komputasi, data sering kali berupa simbol atau sinyal yang diproses lebih lanjut untuk menghasilkan informasi [23].

Ketika data telah diproses, ia berubah menjadi informasi, yaitu kumpulan data yang telah diolah sehingga memberikan makna bagi penerima. Informasi terbentuk dari data yang telah tersusun secara sistematis, sementara pengetahuan merupakan kumpulan informasi yang telah dipahami serta terstruktur dengan baik [22]

2.2.2 Teori Prediksi *Sales*

Prediksi penjualan atau forecasting adalah proses memperkirakan jumlah produk yang akan terjual dalam periode tertentu di masa depan. Proses ini melibatkan analisis data historis dan penggunaan model matematis untuk menghasilkan proyeksi yang dapat membantu perusahaan dalam pengambilan keputusan strategis. Menurut Tita Deitiana (2019), prediksi penjualan bertujuan untuk menentukan kuantitas produk yang akan diminta di masa depan, mengingat adanya risiko dan ketidakpastian yang mungkin dihadapi [24].

Prediksi penjualan bertujuan untuk membantu perusahaan dalam perencanaan produksi agar dapat memenuhi permintaan, mengelola stok dengan memastikan ketersediaan barang yang cukup tanpa kelebihan, serta

mendukung pengambilan keputusan strategis seperti investasi, pemasaran, dan pengembangan produk [25].

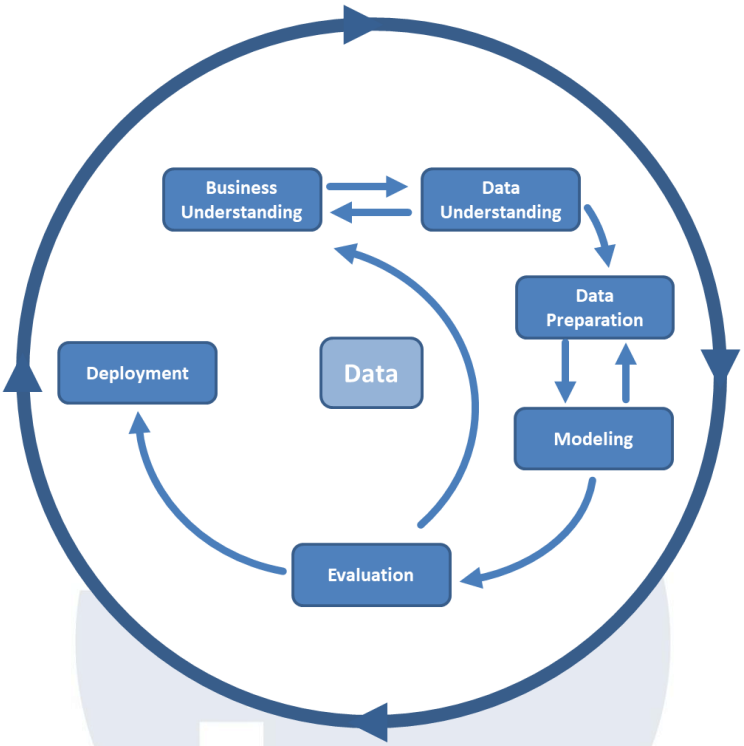
Berbagai metode digunakan dalam prediksi penjualan, antara lain:

1) **Metode Deret Waktu (*Time series*):** Menggunakan data historis untuk memprediksi nilai di masa depan dengan asumsi pola data berulang. Teknik yang umum digunakan adalah *Moving Average* untuk menghaluskan fluktuasi dan *Exponential Smoothing* yang memberi bobot lebih pada data terbaru untuk prediksi yang lebih responsif.
2) **Metode Kualitatif:** Melibatkan penilaian subjektif dari ahli atau survei pasar, seperti Metode Delphi, yang mengumpulkan pendapat para ahli untuk meramalkan tren.
3) **Model Regresi:** Digunakan untuk menganalisis hubungan antara variabel independen dan dependen, dengan data historis sebagai dasar untuk memprediksi penjualan berdasarkan faktor-faktor yang memengaruhinya.[26]

2.3 Teori tentang Framework/Algoritma yang digunakan

2.3.1 Framework CRISP-DM

Metode **CRISP-DM** (Cross-Industry Standard Process for Data Mining) adalah kerangka kerja standar yang digunakan dalam proyek data mining dan analisis data. Metode ini dirancang untuk membantu para profesional data memahami, merencanakan, dan mengelola proses analisis data secara sistematis. Dikembangkan pada akhir 1990-an, CRISP-DM telah menjadi salah satu metode yang paling populer untuk mendukung implementasi proyek berbasis data [27].



Gambar 2.1 Framework CRISP-DM

CRISP-DM terdiri dari enam tahap utama yang bersifat iteratif dan fleksibel, memungkinkan pengguna untuk kembali ke tahap sebelumnya jika diperlukan [34]. Berikut adalah penjelasan setiap tahap [28]:

1. *Business Understanding* (Pemahaman Bisnis)

Tahap ini berfokus pada memahami tujuan bisnis dan masalah yang ingin diselesaikan melalui analisis data. Tujuannya adalah untuk menghubungkan kebutuhan bisnis dengan strategi data mining yang akan digunakan. Aktivitas utama meliputi:

a) Mengidentifikasi tujuan proyek.
b) Memahami kebutuhan bisnis.
c) Menentukan masalah yang ingin diselesaikan dengan data.

2. *Data Understanding* (Pemahaman Data)

Tahap ini melibatkan pengumpulan dan eksplorasi data yang relevan. Data dianalisis secara mendalam untuk memahami karakteristiknya, seperti pola, anomali, atau kualitas data. Aktivitas utama meliputi:

a) Mengumpulkan data awal.
b) Mengeksplorasi data (deskriptif).
c) Memeriksa kualitas data (*missing values*, duplikasi, dll).

3. *Data Preparation* (Persiapan Data)

Pada tahap ini, data yang telah dikumpulkan diproses agar siap untuk dianalisis lebih lanjut. Aktivitas ini meliputi pembersihan data, transformasi, dan penggabungan dataset yang diperlukan. Tujuannya adalah memastikan data berkualitas tinggi. Aktivitas utama meliputi:

a) Pembersihan data dari *noise* atau *missing values*.
b) Transformasi data (normalisasi, *encoding*, *scaling*).
c) Penggabungan atau pengelompokan dataset.

4. *Modeling* (Pemodelan)

Di tahap ini, algoritma atau model yang relevan diterapkan untuk menganalisis data dan memprediksi hasil. Proses ini melibatkan pemilihan teknik yang sesuai, pengaturan parameter, dan pelatihan model. Aktivitas utama meliputi:

a) Memilih algoritma yang sesuai (regresi, *clustering*, dll).
b) Melatih model dengan data.
c) Menguji model untuk memastikan kinerjanya.

5. *Evaluation* (Evaluasi)

Setelah model selesai dibuat, hasilnya dievaluasi untuk memastikan bahwa model tersebut memenuhi tujuan bisnis yang telah ditetapkan. Evaluasi dilakukan dengan menggunakan metrik tertentu, seperti akurasi, presisi, recall, atau *F1-score*. Aktivitas utama meliputi:

a) Mengevaluasi kinerja model.

b) Membandingkan hasil model dengan kebutuhan bisnis.

c) Memastikan model layak untuk implementasi.

6. *Deployment* (Implementasi)

Tahap terakhir adalah mengimplementasikan hasil analisis ke dalam lingkungan bisnis. Hasil dapat berupa laporan, *dashboard*, atau integrasi model ke dalam sistem perusahaan. Aktivitas utama meliputi:

a) Menyediakan hasil analisis ke pengguna.

b) Mengintegrasikan model ke dalam proses operasional.

c) Memberikan pelatihan kepada pengguna jika diperlukan.

2.3.2 Teori *Machine learning*

Machine learning (ML) merupakan salah satu cabang dari kecerdasan buatan yang memungkinkan sistem untuk memperoleh pembelajaran dari data serta pengalaman tanpa harus diprogram secara eksplisit. Konsep ini pertama kali diperkenalkan oleh Arthur Samuel pada tahun 1959, yang mendefinisikan ML sebagai kemampuan komputer untuk belajar dari data dan meningkatkan kinerjanya seiring waktu [29].

Proses pembelajaran dalam *Machine learning* dilakukan melalui serangkaian algoritma yang bertugas menganalisis data, mengenali pola, serta menghasilkan keputusan atau prediksi berdasarkan informasi yang diperoleh [30].

Machine learning dibagi menjadi tiga kategori utama berdasarkan cara pembelajarannya:

1) ***Supervised Learning*:** Dalam metode ini, algoritma dilatih menggunakan data berlabel. Model belajar untuk memetakan input ke output berdasarkan contoh-contoh yang diberikan. Contoh aplikasi termasuk klasifikasi dan regresi.
2) ***Unsupervised Learning*:** Berbeda dengan supervised learning, metode ini tidak menggunakan data berlabel. Algoritma berusaha menemukan pola atau struktur dalam data yang tidak terlabel, seperti pengelompokan (*clustering*) dan asosiasi.
3) ***Reinforcement Learning*:** Metode ini melibatkan interaksi dengan lingkungan di mana agen belajar dari umpan balik yang diterima setelah melakukan tindakan tertentu. Tujuannya adalah untuk memaksimalkan *reward* atau mengurangi *penalty* [31].

2.3.3 Teori *Time series*

Deret waktu adalah sekumpulan data pengamatan yang diukur pada titik waktu tertentu dengan interval yang sama. Menurut Box dan Jenkins (1976), deret waktu merupakan serangkaian nilai pengamatan yang saling berkorelasi satu sama lain, di mana setiap pengamatan berhubungan dengan pengamatan sebelumnya. Dalam analisis statistik, deret waktu digunakan untuk mempelajari pola dan tren dari data yang dikumpulkan seiring waktu, seperti penjualan bulanan, suhu harian, atau harga saham [32].

Deret waktu memiliki beberapa karakteristik penting yang perlu diperhatikan:

1) **Tren (*Trend*)**: Pergerakan jangka panjang dari data, baik naik maupun turun.
2) **Musiman (*Seasonality*)**: Pola fluktuasi yang terjadi secara teratur dalam periode tertentu, seperti bulanan atau tahunan.
3) **Siklus (*Cyclic*)**: Fluktuasi yang terjadi dalam periode yang tidak tetap dan sering kali terkait dengan kondisi ekonomi.

4) **Variasi Acak (*Irregular Variation*)**: Fluktuasi yang tidak dapat diprediksi dan bersifat acak [33].

2.3.4 Algoritma ARIMA

Model *Auto Regressive Integrated Moving Average* (ARIMA) merupakan teknik statistik yang digunakan dalam analisis serta peramalan data deret waktu (*time series*). Metode ini tidak mempertimbangkan variabel independen dalam proses prediksi, melainkan hanya mengandalkan data historis dari variabel dependen untuk menghasilkan estimasi nilai di masa mendatang [34]. Model ini dinyatakan dalam notasi ARIMA(p, d, q), di mana:

1) **p**: Orde dari proses *Autoregressive* (AR).
2) **d**: Jumlah *differencing* yang diperlukan untuk membuat data stasioner.
3) **q**: Orde dari proses *Moving Average* (MA)

ARIMA terdiri dari tiga komponen utama:

1) ***AutoRegressive* (AR)**

   Komponen AR menggambarkan hubungan antara nilai saat ini dalam data *time series* dengan nilai-nilai sebelumnya (lag) sebagai variabel prediktor. Dalam model ARIMA, parameter **p** menunjukkan jumlah lag yang digunakan dalam model.

$$X_t = \emptyset_1 X_{t-1} + \emptyset_2 X_{t-2} + \cdots + \emptyset_p X_{t-p} + \epsilon_t$$

Rumus 2.1 Persamaan dasar AutoRegresive

Di mana:

X_t : Nilai *time series* pada waktu t

$\emptyset$: Koefisien *autoregressive*

ϵ_t : Error (residual)

2) ***Integrated* (I)**

Komponen I menangani aspek stasioneritas data. Data *time series* sering kali memiliki tren atau pola musiman sehingga perlu dilakukan proses differencing untuk menghilangkan pola tersebut. Parameter **d** menunjukkan jumlah kali differencing yang diterapkan untuk membuat data stasioner.

$$Y_t = X_t - X_{t-1}$$

Rumus 2.2 Persamaan Dasar Komponen Integrated

3) ***Moving Average* (MA)**

Komponen MA menangkap hubungan antara nilai *time series* saat ini dengan residual (error) dari lag sebelumnya. Dalam model ARIMA, parameter **q** menunjukkan jumlah residual lag yang digunakan dalam model.

$$X_t = \mu + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + \cdots + \theta_q\epsilon_{t-q}$$

Rumus 2.3 Persamaan Dasar Moving Avarage(MA)

Di mana:

μ : Rata rata dari *time series*

θ : Koefisien moving average

ϵ : Residual atau error

2.3.5 Algoritma SARIMA

SARIMA (Seasonal *Auto Regressive Integrated Moving Average*) adalah model statistik yang digunakan untuk menganalisis dan meramalkan data deret waktu (*time series*) yang memiliki pola musiman. Model ini merupakan pengembangan dari ARIMA (*Auto Regressive Integrated Moving Average*) dengan menambahkan komponen musiman untuk menangani data yang menunjukkan pola berulang pada interval waktu tertentu, seperti bulanan, kuartalan, atau tahunan [35].

Model SARIMA dinyatakan dalam notasi:

$$SARIMA(p, d, q)(P, D, Q)s$$

Rumus 2. 4 Notasi SARIMA

Dimana:

1) **p, d, q**: Parameter non-musiman dari model ARIMA.
    a) **p**: Jumlah lag *autoregresif* (AR).
    b) **d**: Tingkat *differencing* untuk membuat data stasioner.
    c) **q**: Jumlah lag *moving average* (MA).
2) **P, D, Q**: Parameter musiman.
    a) **P**: Jumlah lag *autoregresif* musiman.
    b) **D**: Tingkat *differencing* musiman.
    c) **Q**: Jumlah *lag moving average* musiman.
3) **s**: Panjang periode musiman (contoh: 12 untuk data bulanan).

2.3.6 Algoritma LSTM

Long Short-Term Memory (LSTM) merupakan salah satu arsitektur dalam *Recurrent Neural Network* (RNN) yang dirancang untuk mengatasi permasalahan vanishing gradient, yang kerap terjadi ketika RNN memproses data sekuensial dengan rentang panjang. LSTM memungkinkan model untuk mempertahankan informasi dalam periode waktu yang lebih lama, sehingga lebih efisien dalam menangkap ketergantungan jangka panjang dalam data [36].

Arsitektur LSTM terdiri dari beberapa komponen kunci yang berfungsi untuk mengelola informasi:

1) ***Cell State***: Merupakan komponen utama yang menyimpan informasi sepanjang waktu.
2) ***Gates***: Terdapat tiga jenis gate dalam LSTM:

a) ***Input Gate***: Mengontrol informasi baru yang akan ditambahkan ke cell state.

b) ***Forget Gate***: Menentukan informasi mana yang harus dihapus dari cell state.

c) ***Output Gate***: Mengatur informasi yang akan dikeluarkan dari cell state sebagai output.

Proses kerja LSTM dapat dijelaskan melalui beberapa langkah matematis [37]:

1. *Input Gate*: Menghitung nilai *input gate* i_t, menggunakan fungsi *sigmoid*:

$$i_t = \sigma(W_i x_t + U_i h_t + b_i)$$

Rumus 2.5 Fungsi Input Gate

2. *Candidate Cell State*: Menghitung nilai kandidiat *cell State* C_t:

$$C_t = tanh(W_c x_t + U_c h_{c-1} + b_c)$$

Rumus 2.6 Notasi Candidate Cell State

3. *Forget Gate*: Menghitung nilai *forget gate* f_t :

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$

Rumus 2.7 Fungsi Forget Gate

4. *Update Cell State*: Memperbarui *cell state* dengan menggabungkan informasi baru dan melupakan informasi lama:

$$C_t = i_t * C_t + f_t * C_{t-1}$$

Rumus 2.8 Notasi Update Cell State

5. *Output Gate*: Menghitung nilai *output* gate o_t

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$

Rumus 2.9 Fungsi Output Gate

6. *Final Outp*ut Menghasilkan *output* akhir h_t :

$$h_t = o_t * \tanh(C_t)$$

Rumus 2.10 Final Output

2.3.7 MAE

Mean Absolute Error (MAE) merupakan salah satu metrik evaluasi yang paling umum digunakan untuk menilai kinerja model *machine learning* dalam tugas regresi. Regresi sendiri adalah jenis pembelajaran mesin yang bertujuan untuk memprediksi nilai numerik yang bersifat kontinu. MAE menghitung rata-rata dari selisih absolut antara nilai prediksi dan nilai aktual, sehingga memberikan gambaran mengenai tingkat kesalahan model dalam melakukan estimasi. Secara matematis, MAE dihitung dengan rumus berikut:

$$MAE = (1/n) * \Sigma |y_i - \hat{y}_i|$$

Rumus 2.11 Rumus MAE

Dimana:

1) n adalah jumlah total data poin.
2) y_i adalah nilai sebenarnya dari data poin ke-i.
3) $\hat{y}_i$ adalah nilai prediksi dari data poin ke-i.
4) Σ menunjukkan penjumlahan dari semua data poin.
5) |...| menunjukkan nilai absolut

MAE memberikan gambaran tentang seberapa besar kesalahan rata-rata yang dilakukan oleh model dalam memprediksi nilai. Nilai MAE yang lebih rendah menunjukkan bahwa model memiliki kinerja yang lebih baik dalam memprediksi nilai yang mendekati nilai sebenarnya. Karena MAE menggunakan nilai absolut, ia memberikan bobot yang sama untuk semua kesalahan, baik kesalahan positif (prediksi lebih tinggi dari nilai sebenarnya) maupun kesalahan negatif (prediksi lebih rendah dari nilai sebenarnya) [38].

2.3.8 MSE

Mean Squared Error (MSE) merupakan salah satu metrik evaluasi yang sering digunakan untuk menilai kinerja model *machine learning* dalam tugas regresi. Sama seperti *Mean Absolute Error* (MAE), MSE mengukur rata-rata kesalahan antara nilai prediksi dan nilai aktual, namun dengan pendekatan yang

berbeda. MSE diperoleh dengan menghitung akar kuadrat dari rata-rata kuadrat selisih antara nilai prediksi dan nilai sebenarnya, sehingga memberikan bobot lebih besar pada kesalahan yang lebih besar. Secara matematis, MSE dihitung dengan rumus berikut:

$$MSE = (1/n) * \Sigma (y_i - \hat{y}_i)^2$$

Rumus 2.12 Rumus MSE

Dimana:

1) n adalah jumlah total data poin.
2) y_i adalah nilai sebenarnya dari data poin ke-i.
3) $\hat{y}_i$ adalah nilai prediksi dari data poin ke-i.
4) Σ menunjukkan penjumlahan dari semua data poin.

Mean Squared Error (MSE) memberikan indikasi mengenai besarnya rata-rata kesalahan yang terjadi dalam prediksi model. Semakin rendah nilai MSE, semakin baik kinerja model dalam melakukan estimasi. Karena MSE melibatkan proses pengkuadratan kesalahan sebelum dihitung rata-ratanya, metrik ini memberikan bobot lebih besar terhadap kesalahan yang lebih besar, sehingga lebih sensitif terhadap outlier dalam data [39].

2.3.9 R^2 (r-*Squared*)

R-*squared* (R^2), atau dikenal sebagai *koefisien determinasi*, adalah ukuran statistik yang digunakan dalam analisis regresi untuk menunjukkan seberapa besar variasi dalam variabel dependen (variabel yang diprediksi) dapat dijelaskan oleh variabel independen (variabel prediktor) dalam suatu model. Nilai R^2 berkisar antara 0 hingga 1, dengan nilai yang lebih tinggi menunjukkan kemampuan model yang lebih baik dalam menjelaskan data [40].

Rumus dasar untuk menghitung R^2 adalah:

$$R2 = \frac{SSR}{SST}$$

Rumus 2.13 Rumus R^2

Dimana:

1) SST (*Sum of Squares Total*): Jumlah kuadrat total dari variabilitas data.
2) SSR (*Sum of Squares Residual*): Jumlah kuadrat sisa yang tidak dijelaskan oleh model.

2.4 Teori tentang tools/software yang digunakan

2.4.1 Bahasa Program Python

Python adalah bahasa pemrograman tingkat tinggi yang pertama kali dikembangkan oleh Guido van Rossum pada akhir 1980-an dan dirilis pada tahun 1991. Python dirancang dengan fokus pada keterbacaan kode sehingga sintaksnya lebih sederhana dan mudah dipahami dibandingkan dengan banyak bahasa pemrograman lainnya. Hal ini menjadikannya pilihan populer di kalangan pemula maupun pengembang berpengalaman untuk berbagai aplikasi, mulai dari pengembangan web hingga analisis data dan *machine learning* [41].

Python digunakan dalam berbagai bidang, antara lain:

1) **Pengembangan Web**: Membangun aplikasi web menggunakan kerangka kerja seperti Django dan Flask.
2) **Ilmu Data dan Analisis**: Digunakan oleh ilmuwan data untuk analisis statistik dan visualisasi data.
3) **Automasi Tugas**: Memungkinkan otomatisasi proses rutin melalui *scripting*.
4) ***Machine learning***: Banyak digunakan dalam pengembangan model *machine learning* dengan pustaka seperti *TensorFlow* dan *Scikit-learn* [42].

2.4.2 Streamlit

Streamlit merupakan framework *Python open-source* yang dirancang untuk mempermudah pembuatan dan berbagi aplikasi web interaktif dalam proyek *machine learning* serta *data science*. Dengan Streamlit, pengguna dapat mengonversi script data menjadi aplikasi web yang siap dibagikan dalam waktu

singkat, tanpa memerlukan keahlian mendalam dalam pengembangan web [43]. Penggunaan Umum Streamlit adalah sebagai berikut:

1) Membuat *Dashboard*: Streamlit sangat cocok untuk membuat *dashboard* interaktif untuk memvisualisasikan data dan hasil analisis.
2) Membuat Aplikasi *Machine learning*: Anda dapat menggunakan Streamlit untuk membuat aplikasi yang memungkinkan pengguna untuk mengunggah data, melatih model *machine learning*, dan melihat hasilnya secara *real-time*.
3) Membuat Alat untuk Eksplorasi Data: Streamlit dapat digunakan untuk membuat alat yang memungkinkan pengguna untuk menjelajahi data secara interaktif dan menemukan *insight* baru.
4) Berbagi Hasil Penelitian: Streamlit memudahkan Anda untuk berbagi hasil penelitian Anda dengan orang lain dalam bentuk aplikasi web interaktif.