

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Berdasarkan berbagai penelitian terdahulu, penggunaan algoritma Long Short-Term Memory (LSTM) telah banyak diterapkan dalam berbagai studi prediksi, khususnya pada data time series. Metode ini dikenal mampu mempelajari pola jangka panjang dalam data sehingga sering digunakan dalam penelitian yang berkaitan dengan prediksi harga saham, peramalan ekonomi, maupun analisis data sekuensial lainnya.[4]

Masih sedikit penelitian terdahulu yang menggunakan xLSTM untuk prediksi saham, terutama pada minute level trading atau scalping. Hal ini dikarenakan masih baru nya model xLSTM dan adopsi penggunaannya masih terlalu minim, terutama di kalangan akademik. Penelitian terdahulu masih lebih berfokus pada data yang berbasis timeseries yang lebih terstruktur seperti failure rate atau linear growth pada time series.

Selain LSTM, beberapa penelitian terbaru juga mulai mengembangkan variasi model seperti xLSTM yang dirancang untuk meningkatkan kemampuan pemodelan terhadap data sekuensial yang kompleks. Model ini dikembangkan untuk mengatasi beberapa keterbatasan pada arsitektur LSTM tradisional, terutama dalam menangani dependensi data yang lebih panjang dan meningkatkan efisiensi dalam proses pembelajaran model.[5]

Namun demikian, masih terdapat research gap yang sangat signifikan dalam literatur akademis saat ini. Berdasarkan penelusuran data pada database Scopus, sejauh ini hanya ditemukan satu jurnal ilmiah yang mengkaji penerapan arsitektur xLSTM untuk prediksi harga saham. Penerapan penelitian yang membahas penerapan LSTM maupun satu yang menggunakan xLSTM dalam prediksi saham umumnya menggunakan data harga saham dengan interval waktu harian (daily

data), sehingga pola yang dipelajari oleh model cenderung lebih stabil dan memiliki tingkat noise yang relatif lebih rendah.[6]

Selain jumlahnya yang sangat terbatas, penelitian terdahulu tersebut masih memiliki kelemahan mendasar karena evaluasinya hanya berfokus pada ranah teoritis, yaitu sebatas mengukur metrik akurasi statistik model di atas kertas. Penelitian tersebut belum mengaitkan maupun menguji performa model ke dalam skenario implementasi dunia nyata (real-world case), seperti bagaimana dampaknya jika sinyal tersebut dieksekusi dalam simulasi perdagangan aktif yang melibatkan biaya transaksi dan dinamika pasar riil. Oleh karena itu, penelitian ini hadir untuk mengisi celah tersebut dengan menguji model secara praktis pada simulasi trading tingkat menit.

Oleh karena itu, penelitian ini juga mengembangkan model LSTM menjadi beberapa varian yaitu xLSTM, mLSTM, dan sLSTM. Penggunaan beberapa arsitektur ini diharapkan dapat meningkatkan kemampuan model dalam menangani noise pada data per menit sehingga dapat menghasilkan performa prediksi dan tingkat akurasi yang lebih baik.[7]

Tabel 2. 1 Penelitian Terdahulu

Tahun	Penulis	Masalah Penelitian	Metode yang Digunakan	Relevansi dengan Penelitian
2024	X Fan; C Tao; J Zhao	Improve stock price prediction accuracy using xLSTM models	xLSTM	Terkait dengan saham dan juga xLSTM dan akurasinya dalam memprediksi saham secara jangka panjang dan metode untuk meningkatkan akurasi dari algoritma

				tersebut
2024	M Alharthi; A Mahmood	Long-term time series forecasting using xLSTM architecture	xLSTM	Terkait dengan saham dan juga xLSTM dan prediksinya terkait akurasi dalam preddiksi jangka panjang
2025	R Zhang; X Wan; S Bu; M Zhou; Q Zeng; Z Zhang	Predict power output from multiple photovoltaic stations	xLSTM	Prediksi timeseries menggunakan XLSTM dan dampaknya pada data time series dan masalah yang ditemukan
2025	H Wan; J Wang; Q Gan; Y Xia; Y Chang; H Yan	Improve wind and photovoltaic power forecasting with hybrid models	xLSTM - TCCNN	Metode hybrid gabungan xLSTM dan cara untuk meningkatkan akurasi model tersebut di bidang yang mengalami peningkatan drastis tak terduga (noise)
2025	Q Lai; W Zhou	Predict sea surface temperature	xLSTM - CNN	Metode hybrid gabungan

		using CNN-xLSTM		xLSTM dan CNN untuk memprediksi data timeseries di bidang yang stabil
2025	X Wang; Y Zhang; J Chen; X Wu; D Du; et all	Predict SSD hardware failures using hybrid LSTM-xLSTM	LSTM-xLSTM	Perbandingan antara LSTM dan XLSTM dan keunggulan setiap metode di bidang-bidang berbeda dengan banyaknya variabel yang diolah
2025	Huang C	Predict traffic patern in a high noise situation utilizing xLSTM	xLSTM	Penggunaan dataset yang sangat variatif dan besar untuk membuat model prediksi perpetual yang akurat meskipun penuh dengan noise
2025	Wang J	Predicting model Trajectory based on hybrid xLSTM model in table tenis	xLSTM	Melihat arah pergerakan model dan drop off akurasi dari model varian

				xLSTM seperti mLSTM, dan sLSTM
2025	Chen J	Prediction model featuring multi variable in a chaotic high noise enviroment time series	Hybrid xLSTM	Prediksi yang akurat di dataset timeseries yang berantakan dan penuh dengan noise dan outliers untuk melihat kapabilitas dan limitasi model
2025	Zheng D	Resarching adaptability of xlstm variants for multi-horizon stock price	xLSTM	Melihat adaptibilitas model xLSTM jika memprediksi dataset yang memiliki variabel data yang bergerak secara dinamis

Berdasarkan temuan dari berbagai penelitian terdahulu, dapat disimpulkan bahwa penggunaan arsitektur Extended Long Short-Term Memory (xLSTM) menunjukkan tingkat akurasi yang lebih tinggi dalam memprediksi tren data deret waktu (time-series) jika dibandingkan dengan model LSTM tradisional[8]. Keunggulan ini utamanya lahir dari integrasi struktur memori baru, seperti matrix memory pada mLSTM dan alokasi dinamis pada sLSTM, yang memungkinkan model untuk menangkap ketergantungan konteks jangka panjang secara lebih efektif tanpa kehilangan informasi krusial di sepanjang urutan data.[9]

Meskipun memiliki performa prediktif yang lebih superior, xLSTM memiliki karakteristik khusus yang menuntut perhatian lebih pada fase pelatihannya. Salah satu kelemahan yang sering ditemukan pada arsitektur ini adalah kerentanannya terhadap fenomena "halusinasi prediksi," di mana model menghasilkan proyeksi tren yang menyimpang dari pola historis aktual akibat kompleksitas ruang dimensi tinggi yang digunakannya[10]. Jika tidak dikendalikan, variasi representasi di dalam sel xLSTM dapat membuat model menjadi terlalu sensitif terhadap gangguan kecil (noise) pada data pasar yang dinamis.[11]

Oleh karena itu, penerapan strategi optimisasi yang tepat memegang peranan yang sangat krusial agar model xLSTM dapat meminimalkan risiko halusinasi tersebut dan menghasilkan prediksi yang benar-benar akurat.[12] Pengaturan hiperparameter yang presisi, seperti pemilihan learning rate yang adaptif melalui optimizer ADAM serta pembatasan jumlah epoch yang optimal, berfungsi sebagai jangkar untuk menjaga stabilitas konvergensi model[13] [14][15]Proses optimisasi yang terstruktur ini memastikan bahwa kekuatan pemrosesan sekuensial xLSTM tetap terarah pada jalur tren yang valid dan objektif.

2.2 Teori yang berkaitan

2.2.1 Saham dan Pasar Modal

Saham merupakan sebuah instrumen keuangan yang menjadi bukti atas kepemilikan sah seseorang atau suatu lembaga terhadap aset dan kekayaan sebuah perusahaan. Di dalam ekosistem pasar modal, saham berfungsi sebagai komoditas investasi yang diperdagangkan secara aktif melalui bursa efek resmi. Salah satu pusat perdagangan saham paling berpengaruh di dunia adalah pasar saham Amerika Serikat (AS), seperti bursa NASDAQ, yang menjadi rumah bagi saham-saham perusahaan teknologi raksasa, termasuk Advanced Micro Devices (AMD). Pergerakan nilai saham di bursa ini menjadi indikator penting yang mencerminkan sentimen ekonomi serta tingkat kepercayaan investor secara global.

Untuk kebutuhan analisis, data pergerakan saham tersebut dapat diakses secara terbuka melalui berbagai platform penyedia informasi keuangan terkemuka, di mana salah satu yang paling populer digunakan adalah Yahoo Finance. Platform ini menyediakan bank data yang sangat lengkap mengenai informasi historis pergerakan harga, volume perdagangan, hingga berbagai indikator pasar teknikal lainnya. Ketersediaan akses data yang mudah dan terstruktur dari Yahoo Finance ini menjadikannya salah satu modal utama bagi para analis dan

pengembang sistem untuk melakukan riset pasar maupun menguji strategi perdagangan secara berkala.

Di dalam kumpulan data saham tersebut, terdapat lima komponen metrik utama yang biasa dikenal dengan istilah OHLCV (Open, High, Low, Close, Volume). Opening price merupakan harga awal saat saham mulai ditransaksikan ketika bursa dibuka, sedangkan closing price adalah harga transaksi terakhir saat sesi perdagangan hari tersebut resmi ditutup. Di antara kedua waktu tersebut, tercatat pula high price sebagai titik harga tertinggi dan low price sebagai titik harga terendah yang sempat dicapai, serta volume yang menunjukkan total kuantitas lembar saham yang berpindah tangan. Kombinasi seluruh informasi ini sangat krusial dalam analisis data keuangan, terutama ketika digunakan sebagai variabel masukan dalam penelitian pemodelan prediksi berbasis time series.

2.2.2 Prediktif Analisis

Model prediktif berbasis Long Short-Term Memory (LSTM) merupakan salah satu pendekatan dalam deep learning yang banyak digunakan untuk menganalisis dan memprediksi data time series, termasuk pergerakan harga saham. LSTM adalah pengembangan dari Recurrent Neural Network (RNN) yang dirancang untuk mampu menangkap pola ketergantungan jangka panjang pada data berurutan melalui mekanisme gates seperti input gate, forget gate, dan output gate[16]. Mekanisme ini memungkinkan model untuk menyimpan, memperbarui, dan melupakan informasi secara selektif sehingga dapat mempelajari pola historis yang kompleks dari data waktu ke waktu[17]. Dalam konteks prediksi harga saham, model LSTM memanfaatkan data historis seperti opening price, closing price, high price, low price, serta volume perdagangan untuk mempelajari tren dan pola pergerakan pasar. Dengan memproses urutan data historis tersebut, model dapat menghasilkan prediksi harga pada periode berikutnya berdasarkan pola yang telah dipelajari sebelumnya. Keunggulan utama yang membuat model deep learning seperti LSTM jauh lebih sesuai untuk prediksi harga saham dibandingkan dengan model statistik konvensional atau machine learning biasa terletak pada kemampuannya dalam mengekstrak fitur secara mandiri (automated feature extraction) dari data yang bersifat non-linear dan dipenuhi noise. Model statistik tradisional seperti ARIMA bekerja dengan asumsi linearitas yang ketat dan membutuhkan data yang stasioner, sehingga sering kali gagal total ketika dihadapkan pada volatilitas ekstrem dan perubahan rezim pasar modal yang dinamis. Oleh karena itu, LSTM sering digunakan dalam penelitian analisis prediktif karena kemampuannya yang baik dalam menangani data time series yang bersifat dinamis, fluktuatif, dan memiliki ketergantungan temporal[18].

2.2.3 Evaluasi Kinerja Model Prediksi

Evaluasi model LSTM dilakukan untuk mengukur tingkat akurasi dan kinerja model dalam memprediksi data time series, khususnya pada prediksi harga saham. Proses evaluasi ini bertujuan untuk mengetahui seberapa baik model mampu mempelajari pola dari data historis dan menghasilkan prediksi yang mendekati nilai sebenarnya. Beberapa metrik evaluasi yang umum digunakan dalam penelitian prediksi harga saham antara lain Mean Squared Error (MSE), Root Mean Squared Error (RMSE), dan Mean Absolute Error (MAE)[19]. Metrik-metrik tersebut digunakan untuk menghitung selisih antara nilai prediksi yang dihasilkan model dengan nilai aktual pada data uji, sehingga dapat memberikan gambaran tingkat kesalahan prediksi yang dihasilkan oleh model.

Selain untuk mengukur akurasi, evaluasi juga dilakukan untuk membandingkan kinerja beberapa model yang digunakan dalam penelitian, seperti LSTM dan variasinya seperti XLSTM, MLSTM, dan SLSTM. Dengan melakukan perbandingan nilai metrik evaluasi dari masing-masing model, peneliti dapat menentukan model mana yang memiliki performa paling baik dalam memprediksi data saham. Model dengan nilai kesalahan yang lebih kecil menunjukkan kemampuan prediksi yang lebih baik. Hasil evaluasi ini kemudian menjadi dasar dalam menentukan model yang paling optimal untuk digunakan dalam analisis prediksi harga saham berbasis time series[20][21].

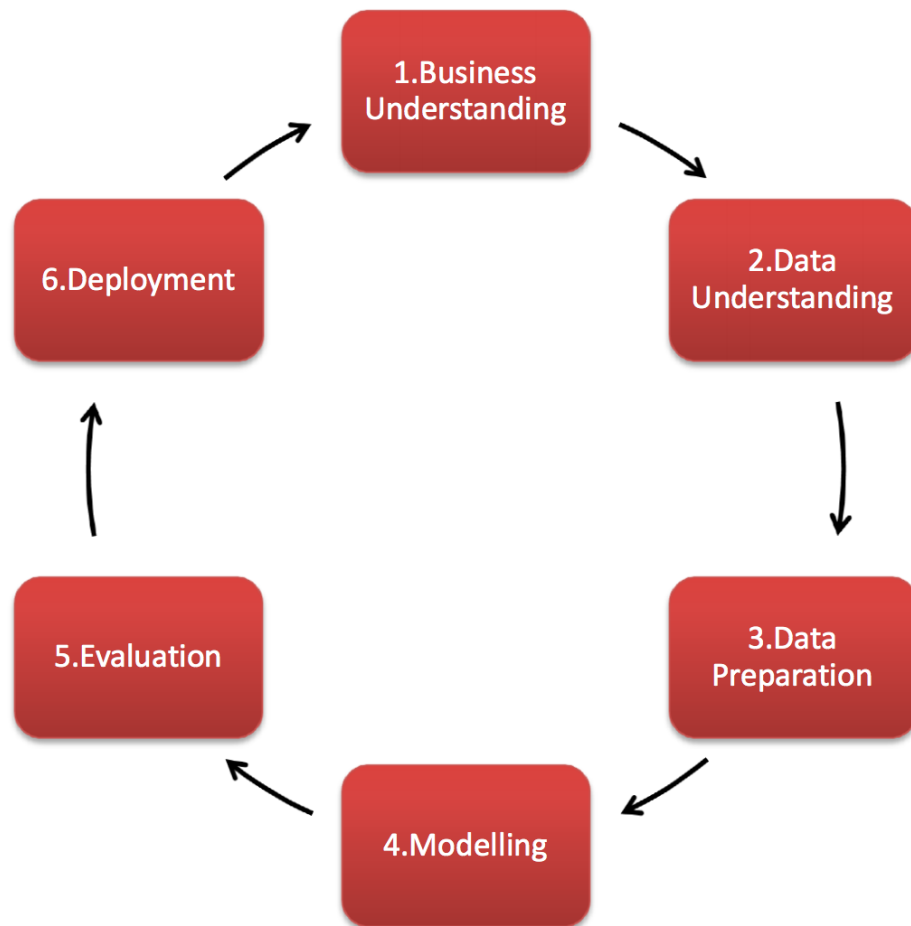
2.3 Framework/Algoritma yang digunakan

2.3.1 Framework

CRISP-DM atau Cross Industry Standard Process for Data Mining adalah sebuah kerangka kerja yang menjadi standar proses global dalam mengelola siklus hidup proyek data science dan machine learning. Kerangka kerja ini menyediakan pendekatan yang terstruktur, adaptif, dan teruji untuk mengubah data mentah menjadi solusi bisnis yang bernilai tinggi. Karena sifatnya yang netral terhadap industri dan teknologi tertentu, CRISP-DM dapat diterapkan pada berbagai skala proyek, mulai dari analisis data sederhana hingga pengembangan model kecerdasan buatan yang kompleks.

Metodologi ini membagi proyek data science ke dalam enam tahapan utama yang saling berkaitan secara berulang, yaitu pemahaman bisnis, pemahaman data, penyiapan data, pemodelan, evaluasi, dan penerapan. Siklus ini digambarkan berbentuk lingkaran karena prosesnya tidak selalu berjalan satu arah; hasil dari satu tahapan sering kali mengharuskan tim untuk kembali ke tahapan sebelumnya demi menyempurnakan hasil. Dengan menerapkan kerangka kerja ini, organisasi dapat meminimalkan risiko kegagalan proyek,

menjaga fokus pada tujuan bisnis, serta memastikan seluruh proses pengolahan data berjalan secara efisien dan terorganisir.



Gambar 2. 1 Alur CRISP-DM

Tahapan pertama adalah business understanding yang berfokus pada pemahaman mendalam tentang tujuan dan kebutuhan dari sudut pandang bisnis. Pada fase ini, dapat ditentukan masalah apa yang ingin diselesaikan, menerjemahkan tujuan bisnis tersebut menjadi tujuan analisis data, serta membuat rencana proyek yang matang. Tanpa pemahaman bisnis yang kuat, model yang dihasilkan nanti bisa jadi tidak relevan atau tidak menyelesaikan masalah utama perusahaan.

Tahapan kedua adalah data understanding yang dimulai dengan mengumpulkan data awal yang diperlukan untuk proyek. Setelah data terkumpul, dapat dilakukan eksplorasi untuk mengenali karakteristik data, mendeteksi masalah

kualitas seperti data yang hilang atau tidak konsisten, serta mencari wawasan awal yang menarik. Langkah ini penting untuk memastikan bahwa data yang dimiliki sesuai dan cukup untuk menjawab kebutuhan bisnis yang sudah ditentukan sebelumnya.

Tahapan ketiga adalah data preparation yang biasanya memakan waktu paling banyak dalam seluruh proses. Di sini akan dilakukan pembersihan data, memilih variabel atau fitur yang relevan, mengubah format data, serta menggabungkan data dari berbagai sumber berbeda menjadi satu kesatuan. Hasil akhir dari tahapan ini adalah kumpulan data bersih yang sudah siap dimasukkan ke dalam algoritma atau model pemrograman.

Tahapan keempat adalah modelling yang merupakan proses inti dari pembuatan kecerdasan buatan atau analisis prediktif. Pada fase ini, ada pemilihan dan penerapan berbagai teknik atau algoritma kecerdasan buatan yang sesuai dengan karakteristik masalah bisnis. Dapat juga dilakukan pengaturan parameter dari algoritma tersebut dan melakukan pengujian berulang kali untuk menemukan model dengan performa terbaik dan akurasi paling tinggi.

Tahapan kelima adalah evaluation yang bertujuan untuk menilai model yang sudah dibuat secara lebih mendalam sebelum benar-benar digunakan. Di sini dapat dilihat kembali apakah hasil dari model tersebut sudah sejalan dengan tujuan bisnis yang ingin dicapai pada tahapan pertama. Selain itu, model akan diperiksa apakah ada langkah atau detail penting dalam proses pembuatan model yang terlewat atau perlu diperbaiki lagi.

Tahapan keenam dan terakhir adalah deployment yang merupakan proses menerapkan model yang sudah dievaluasi ke dalam sistem operasional perusahaan. Model yang sudah jadi akan diintegrasikan agar bisa digunakan secara langsung oleh pengguna atau sistem lain untuk mengambil keputusan berdasarkan data baru. Proses ini juga mencakup pembuatan rencana pemeliharaan dan pemantauan agar performa model tetap terjaga dengan baik dalam jangka panjang.

Alasan utama CRISP-DM dipilih dalam proyek ini adalah karena kerangka kerja tersebut sangat sesuai dengan struktur eksperimen yang terencana. Karakteristik siklusnya yang iteratif atau berulang memungkinkan proses pengujian model dilakukan secara fleksibel, di mana hasil dari satu tahapan dapat digunakan untuk memperbaiki tahapan sebelumnya. Hal ini memberikan ruang bagi tim untuk terus melakukan eksperimen dan penyesuaian parameter secara sistematis demi mendapatkan hasil akhir yang optimal.

Selain itu, metodologi ini memiliki sifat yang sangat data-centric, artinya seluruh alur kerja berpusat pada pemahaman dan pengolahan data yang mendalam. Fokus yang kuat pada kualitas data sejak awal mulai dari tahap eksplorasi hingga penyiapan data memastikan bahwa algoritma yang digunakan nantinya bekerja dengan informasi yang akurat. Pendekatan ini meminimalkan

risiko terjadinya kesalahan analisis akibat data yang kotor atau tidak relevan, sehingga model prediktif yang dibangun menjadi jauh lebih andal.

Keunggulan lain dari CRISP-DM terletak pada fase evaluasinya yang sangat terpadu serta siklus penelitian yang jelas dari awal hingga akhir. Sebelum model benar-benar diterapkan, ada proses peninjauan mendalam untuk memastikan bahwa performa teknis model sudah sejalan dengan tujuan bisnis yang ingin dicapai. Alur kerja yang transparan dan terukur ini membuat seluruh perkembangan proyek dapat dipantau dengan mudah, memberikan batasan yang jelas pada setiap fase, dan menjamin efisiensi dalam penyelesaian proyek data science.

2.3.2 LSTM

Long Short-Term Memory Network atau LSTM merupakan algoritma hasil modifikasi dari Recurrent Neural Network (RNN) yang dirancang untuk mengatasi kelemahan mendasar pada struktur standar RNN. Meskipun ada banyak variasi modifikasi RNN yang dikembangkan, LSTM tetap menjadi salah satu yang paling populer dan banyak digunakan hingga saat ini. Alasan utama popularitasnya adalah kemampuan LSTM dalam melengkapi kekurangan RNN standar, khususnya masalah *vanishing gradient*, yang membuat RNN konvensional kesulitan atau bahkan tidak dapat memprediksi data berdasarkan informasi lampau yang tersimpan dalam jangka waktu yang lama[22][23].

Kemampuan memproses data jangka panjang atau *time-series* ini dapat terwujud karena LSTM memiliki struktur sel yang lebih kompleks dibandingkan RNN biasa[24]. Di dalam setiap sel LSTM, terdapat mekanisme gerbang (*gates*) yang mengatur informasi apa saja yang perlu dibuang, disimpan, atau diteruskan ke tahapan berikutnya. Melalui pengaturan yang dinamis ini, LSTM mampu mempertahankan informasi penting dari masa lalu yang jaraknya sangat jauh dan mengabaikan informasi yang sudah tidak relevan, sehingga sangat andal untuk digunakan dalam analisis data deret waktu, prediksi harga saham, hingga pemrosesan bahasa alami[25].

A) LSTM

Cara kerja LSTM (Long Short-Term Memory) mirip seperti sistem penyaringan informasi digital, di mana data baru yang masuk akan diseleksi secara ketat melalui tiga gerbang utama: pertama, informasi masa lalu yang sudah tidak berguna akan dibuang (*forget gate*); kedua, data baru yang penting akan dipilih untuk disimpan (*input gate*); dan ketiga, semua informasi tersebut digabungkan ke dalam memori utama sel untuk menghasilkan prediksi akhir (*output gate*).

Tabel 2. 2 Definisi variabel

Variabel	Penjelasan	Lambang / Nilai
c_t	Cell state pada waktu t , gabungan antara informasi lama dan informasi baru melalui forget gate dan input gate	R
h_t	Hidden state atau output LSTM pada waktu t	[-1, 1]
$\hat{h}_t$	Output aktivasi dari cell state c_t menggunakan fungsi aktivasi ψ	[-1, 1]
z_t	Input kandidat memori baru setelah fungsi aktivasi ϕ diterapkan pada $\hat{z}_t$	R
$\hat{z}_t$	Kombinasi linear dari input x_t dan hidden state sebelumnya h_{t-1} , dikontrol oleh bobot ωz dan rz serta bias bz	R
i_t	Input gate, menentukan seberapa banyak informasi baru (z_t) yang ditambahkan ke cell state	[0,1]
$\hat{i}_t$	Nilai sebelum aktivasi sigmoid pada input gate	R
f_t	Forget gate, menentukan seberapa banyak informasi lama (c_{t-1}) yang disimpan	[0,1]
$\hat{f}_t$	Nilai sebelum aktivasi sigmoid pada forget gate	R
o_t	Output gate, menentukan seberapa banyak dari cell state yang digunakan sebagai output (h_t)	[0,1]
$\hat{o}_t$	Nilai sebelum aktivasi sigmoid pada output gate	R
(σ)	Memiliki interval nilai [0,1], jika mendekati 0 akan dihapuskan, dan dilanjutkan jika nilainya 1	[0,1]

Tabel di atas menyajikan rangkuman komprehensif mengenai daftar lambang, simbol matematis, beserta pengertian fungsionalnya masing-masing yang terdapat di dalam formula rumus Long Short-Term Memory (LSTM), sehingga mempermudah pembaca dalam memahami peran setiap variabel di setiap gerbang pemrosesan informasi sel.

Gate Name	Equation
Input Gate	$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$ $\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$
Forget Gate	$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$
Update Gate	$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$ $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$
Output Gate	$h_t = o_t \odot \tanh(c_t)$

Gambar 2. 2 Rumus LSTM

Gambar di atas menunjukkan daftar rumus matematika yang menyusun arsitektur Long Short-Term Memory atau LSTM. Jika di bedah satu per satu, semua rumus ini sebenarnya bekerja bersama-sama di dalam setiap sel jaringan untuk mengatur aliran memori, menentukan informasi apa yang perlu disimpan, dan membuang data yang sudah tidak relevan lagi seiring berjalannya waktu.

Tahap pertama dimulai dari proses di bagian Forget Gate. fungsi utama dari gerbang ini adalah untuk menyaring informasi dari masa lalu. di sini, bobot matriks dan bias akan dikalikan dengan input saat ini serta memori dari langkah sebelumnya, lalu dilewatkan ke fungsi aktivasi sigmoid yang menghasilkan nilai antara 0 sampai 1. kalau nilainya mendekati 0 berarti informasi lama tersebut akan dihapus, sedangkan kalau mendekati 1 berarti memorinya akan dipertahankan.

Setelah menentukan apa yang harus dibuang, alur berlanjut ke bagian Input Gate yang terdiri dari dua baris rumus kalkulasi yang berfungsi untuk memutuskan nilai mana saja dari input baru yang layak untuk diperbarui ke dalam memori sel. sama seperti sebelumnya, fungsi sigmoid di sini berperan sebagai penjaga gerbang yang memberikan skala prioritas terhadap data baru yang masuk ke dalam sistem.

Sementara itu, rumus kedua di dalam Input Gate menggunakan fungsi aktivasi tanh untuk membuat sebuah vektor kandidat memori baru yang berisi nilai-nilai

alternatif potensial. hasil dari ini nantinya akan dikombinasikan dengan seleksi untuk digabungkan ke dalam memori utama sel pada tahap berikutnya.

Masuk ke bagian Update Gate, di sinilah pembaruan kondisi memori sel yang sebenarnya terjadi melalui ini secara elegan menggabungkan memori dari masa lalu dan masa kini. bagian ini melambangkan seberapa banyak memori lama yang tetap dipertahankan setelah disaring oleh forget gate, sedangkan bagian selanjutnya melambangkan seberapa banyak kandidat informasi baru yang berhasil dimasukkan ke dalam sistem.

Masih di dalam bagian Update Gate, ada rumus pendukung lainnya yang sering diidentifikasi sebagai persiapan untuk gerbang keluaran. rumus ini berfungsi untuk menentukan bagian mana dari kondisi memori sel yang sudah diperbarui tadi yang layak untuk dikeluarkan sebagai output pada langkah waktu saat ini. fungsi aktivasi sigmoid kembali digunakan untuk menyaring porsi informasi tersebut secara presisi.

Tahap paling akhir dari seluruh rangkaian kalkulasi sel ini ditutup oleh bagian Output Gate melalui rumus tadi. di sini, kondisi memori sel yang baru saja diperbarui akan dilewatkan terlebih dahulu ke dalam fungsi aktivasi agar nilainya berada di rentang -1 sampai 1.

Nilai inilah yang menjadi output akhir dari sel LSTM pada time-step berjalan. fungsi ini sangat krusial karena ia memegang dua peran sekaligus: pertama, sebagai hasil prediksi atau luaran untuk layer berikutnya, dan kedua, sebagai umpan balik yang akan dikirimkan kembali ke langkah waktu berikutnya bersama dengan nilai memori sel. siklus matematis yang rapi inilah yang membuat LSTM bisa mengingat pola jangka panjang dengan sangat stabil tanpa kehilangan arah tren data.

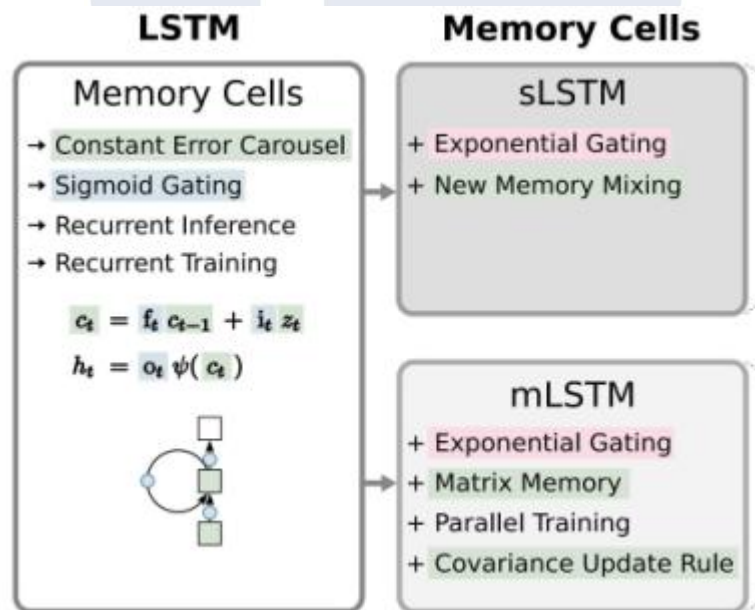
Kemampuan utama yang membuat algoritma LSTM sangat unggul adalah kapasitasnya untuk mengingat sekumpulan informasi penting yang sudah disimpan dalam jangka waktu yang panjang. Tidak hanya mempertahankan data masa lalu, struktur ini juga dirancang untuk secara otomatis menghapus informasi lama yang dinilai sudah tidak lagi relevan dengan kondisi terkini. Mekanisme penyaringan yang cerdas ini memastikan bahwa memori yang tersimpan di dalam sistem selalu optimal dan siap digunakan untuk menghasilkan prediksi yang akurat.

Proses seleksi informasi tersebut dapat berjalan dengan sangat efisien karena adanya sistem gerbang (*gates*) internal yang mengatur aliran data di dalam setiap selnya. Melalui interaksi matematika antar variabel yang ada, LSTM mampu mengontrol kapan harus membuka jalan bagi informasi baru, kapan harus mempertahankan informasi lama, dan kapan harus membuang data yang tidak berguna. Efisiensi dalam pemrosesan ini membuat model tidak terbebani oleh tumpukan data masa lalu yang tidak berkontribusi pada hasil akhir.

Oleh karena itu, dalam berbagai pengujian dan implementasi, LSTM terbukti jauh lebih efisien dalam melakukan proses, memprediksi, sekaligus melakukan klasifikasi data yang berdasarkan urutan waktu atau *time-series* tertentu. Karakteristik khusus ini menjadikannya solusi yang jauh lebih bertenaga dan stabil jika dibandingkan dengan algoritma pendahulunya, yaitu algoritma RNN konvensional. Dengan mengatasi masalah hilangnya memori jangka panjang, LSTM menjadi standar utama dalam menangani data yang memiliki ketergantungan waktu yang kuat.

B) XLSTM

xLSTM merupakan pengembangan dari arsitektur Long Short-Term Memory (LSTM) yang dirancang untuk meningkatkan kemampuan model dalam memahami pola data berurutan, seperti teks, suara, maupun data time series. Berbeda dengan LSTM biasa, xLSTM menggunakan mekanisme memori yang lebih modern dan efisien sehingga mampu menangani konteks jangka panjang dengan lebih baik, memiliki performa yang lebih stabil, serta dapat digunakan pada kebutuhan kecerdasan buatan dan deep learning yang lebih kompleks[26].



Gambar 2. 3 Proses xLSTM

Alur evolusi arsitektur xLSTM ini bisa dilihat dari kiri ke kanan pada gambar diatas. bagian paling kiri menunjukkan struktur dari LSTM tradisional atau klasik yang biasa diketahui. dari situ arsitekturnya mulai berkembang dan pecah jadi dua varian baru yaitu algoritma sLSTM dan mLSTM[27]. kedua varian inilah yang nantinya dipakai sebagai pondasi dasar untuk menyusun blok xLSTM.

Setelah itu blok-blok xLSTM ini akan disusun secara bertingkat atau stacked sampai akhirnya membentuk arsitektur xLSTM penuh yang ada di bagian

paling kanan gambar. arsitektur akhir ini punya sifat yang modular dan efisien karena memang dari awal dirancang khusus sebagai algoritma pemodelan untuk menangani konteks data jangka panjang. susunan blok arsitektur inilah yang membuat xLSTM bisa terbentuk dengan utuh[28].

$$\begin{aligned}
C_t &= f_t C_{t-1} + i_t v_t k_t^\top && \text{cell state (19)} \\
n_t &= f_t n_{t-1} + i_t k_t && \text{normalizer state (20)} \\
h_t &= o_t \odot \tilde{h}_t, \quad \tilde{h}_t = C_t q_t / \max \left\{ \left| n_t^\top q_t \right|, 1 \right\} && \text{hidden state (21)} \\
q_t &= W_q x_t + b_q && \text{query input (22)} \\
k_t &= \frac{1}{\sqrt{d}} W_k x_t + b_k && \text{key input (23)} \\
v_t &= W_v x_t + b_v && \text{value input (24)} \\
i_t &= \exp(\tilde{i}_t), \quad \tilde{i}_t = w_i^\top x_t + b_i && \text{input gate (25)} \\
f_t &= \sigma(\tilde{f}_t) \text{ OR } \exp(\tilde{f}_t), \quad \tilde{f}_t = w_f^\top x_t + b_f && \text{forget gate (26)} \\
o_t &= \sigma(\tilde{o}_t), \quad \tilde{o}_t = W_o x_t + b_o && \text{output gate (27)}
\end{aligned}$$

Gambar 2. 4 Rumus xLSTM

Berdasarkan algoritma tersebut, Extended Long Short-Term Memory (xLSTM) hadir sebagai arsitektur lanjutan dari LSTM tradisional yang dimodifikasi secara masif. Algoritma baru ini dikembangkan dengan cara menggabungkan dua varian mutasi dari LSTM sendiri, yaitu sLSTM (Scalar LSTM) dan mLSTM (Matrix LSTM). Melalui kombinasi inovatif ini, xLSTM mampu mengatasi keterbatasan struktural yang selama ini ada pada model pemrosesan sekuensial generasi sebelumnya.

Di dalam arsitekturnya, sLSTM berperan dengan menggunakan memori skalar yang dirancang sangat ringan dan efisien untuk merangkum serta menyimpan informasi penting dalam ruang asli (*low-dimensional space*). Di sisi lain, mLSTM bekerja menggunakan pendekatan memori berbasis matriks yang memberikan kapasitas penyimpanan jauh lebih besar. Penggunaan memori matriks ini juga memungkinkan terjadinya pemrosesan data secara paralel yang sangat efisien di dalam ruang berdimensi tinggi (*high-dimensional space*).

Gabungan antara sLSTM dan mLSTM di dalam satu kesatuan arsitektur xLSTM akhirnya memberikan fleksibilitas yang luar biasa dalam merangkum konteks sekuensial, baik di ruang rendah maupun tinggi. Hasilnya, model ini memiliki kemampuan yang jauh lebih efektif dalam menangani berbagai jenis data yang memiliki kompleksitas temporal yang rumit dan beragam. Fleksibilitas serta efisiensi komputasi inilah yang membuat xLSTM menjadi jauh lebih bertenaga dibandingkan dengan algoritma LSTM tradisional.

C) MLSTM

mLSTM atau Matrix LSTM merupakan salah satu komponen inti sekaligus turunan dari metode xLSTM yang dirancang untuk membawa perubahan besar pada cara model menyimpan informasi. Berbeda dengan LSTM tradisional yang menggunakan memori skalar, mLSTM memperkenalkan struktur memori berbasis matriks di dalam selnya. Pendekatan ini sengaja dikembangkan untuk mendobrak batasan kapasitas penyimpanan memori pada arsitektur pemodelan sekuensial generasi sebelumnya[29].



Gambar 2. 5 Fungsi mLSTM

Dengan menggabungkan konsep *matrix memory* tersebut, algoritma ini mampu meningkatkan akurasi dari hasil prediksi secara signifikan. Struktur matriks memungkinkan model untuk menangani hubungan antar-data yang rumit dan menyimpan informasi penting dalam ruang berdimensi tinggi tanpa kehilangan konteks masa lalu. Hal ini membuat mLSTM sangat sensitif terhadap pola-pola halus dalam data deret waktu yang biasanya terlewatkan oleh algoritma biasa[30].

Selain unggul dalam hal akurasi, penerapan memori matriks ini juga memungkinkan model untuk melakukan prediksi dari *dataset* yang berukuran sangat besar. Keunggulan utamanya terletak pada kemampuan pemrosesan secara paralel yang jauh lebih efisien, sehingga waktu komputasi bisa dipangkas secara drastis saat menangani data masif. Kombinasi antara kapasitas ruang memori yang besar dan kecepatan proses inilah yang membuat mLSTM mampu menghasilkan prediksi yang jauh lebih akurat dan stabil untuk kebutuhan analisis tingkat lanjut[31].

D) SLSTM

sLSTM atau Scalar LSTM merupakan salah satu varian turunan dari metode xLSTM yang memperkenalkan pendekatan unik dalam mengelola kapasitas memori. Berbeda dengan arsitektur tradisional yang memiliki batasan ruang

penyimpanan yang kaku, sLSTM dirancang untuk menjadi metode yang dinamis. Algoritma ini memiliki kemampuan bawaan untuk menambahkan memori secara fleksibel tepat pada saat dibutuhkan selama proses pengolahan data berlangsung.



Gambar 2. 6 Fungsi sLSTM

Secara teori, mekanisme tracking memori sLSTM secara teoritis optimal untuk konteks urutan panjang, dibatasi oleh alokasi RAM hardware. Selama perangkat keras yang digunakan masih memiliki sisa ruang memori yang tersedia, sLSTM akan terus memperluas kapasitas penyimpanannya untuk menampung ketergantungan informasi jangka panjang. Kemampuan ini mendobrak batas bawah yang biasanya menjadi kendala utama pada pemodelan konteks data yang sangat besar[32].

Penggunaan memori skalar yang fleksibel ini membuat sLSTM menjadi pilihan yang sangat efisien untuk merangkum informasi tanpa membebani daya komputasi secara berlebihan sejak awal proyek. Model hanya akan menggunakan sumber daya yang benar-benar diperlukan sesuai dengan kompleksitas urutan data yang sedang dihadapi. Kombinasi antara efisiensi skalar dan potensi kapasitas tanpa batas ini memberikan keunggulan besar bagi sLSTM dalam menangani pemodelan urutan data yang sangat panjang.

2.4 Tools/software yang digunakan

2.4.1 Visual Studio Code

Visual Studio Code (VS Code) merupakan salah satu perangkat lunak code editor gratis dan terbuka paling populer yang dikembangkan oleh Microsoft. Meskipun sering dikategorikan sebagai Integrated Development Environment (IDE) karena fiturnya yang kaya, VS Code sebenarnya adalah text editor yang sangat ringan namun dapat ditingkatkan menjadi IDE yang tangguh. Desain antarmukanya yang bersih, modern, dan layout yang mudah dibaca membuat para pengembang dapat fokus menulis kode tanpa terganggu oleh penataan visual yang rumit.

Salah satu keunggulan utama dari VS Code adalah dukungan bawaannya terhadap berbagai fitur esensial seperti alat debugging (pelacakan kesalahan kode) dan integrasi Git secara langsung. Tanpa perlu memasang aplikasi tambahan, pengguna dapat langsung menjalankan script, memasang titik henti kode (breakpoints), dan memeriksa variabel untuk menemukan kesalahan dalam program. Hal ini membuat proses pengembangan perangkat lunak menjadi jauh lebih cepat, efisien, dan meminimalkan hambatan teknis saat aplikasi sedang diuji.

Selain itu, kekuatan terbesar VS Code terletak pada ekosistemnya yang sangat fleksibel berkat dukungan ribuan ekstensi pihak ketiga. Melalui Extension Marketplace, pengguna dapat dengan mudah menyesuaikan VS Code agar mendukung berbagai bahasa pemrograman, mulai dari Python, JavaScript, C++, hingga Java. Kemampuan adaptasi yang tinggi inilah yang membuat VS Code menjadi pilihan utama bagi pengembang lintas platform, baik untuk membangun aplikasi web, mengolah data, hingga mengelola arsitektur cloud.

2.4.2 Python

Python adalah bahasa pemrograman tingkat tinggi yang dirancang agar mudah dibaca, dipelajari, dan digunakan oleh pengembang dari berbagai tingkat keahlian. Mengusung filosofi sintaksis yang bersih dan lugas, Python memungkinkan penulisan kode dalam jumlah baris yang lebih sedikit dibandingkan bahasa pemrograman lain seperti Java atau C++. Karena sifatnya yang serbaguna dan didukung oleh ekosistem komunitas yang sangat besar, Python kini menjadi salah satu bahasa paling populer di dunia untuk pengembangan web, kecerdasan buatan, hingga otomatisasi sistem.

Di bidang analisis data, Python memegang peran yang sangat dominan berkat ketersediaan pustaka (library) khusus yang sangat kuat. Analisis data menggunakan pustaka seperti Pandas dan NumPy untuk membersihkan, memanipulasi, dan mengolah kumpulan data berukuran besar dengan cepat. Selain itu, untuk keperluan visualisasi data, Python menyediakan alat seperti Matplotlib yang mampu mengubah angka-angka rumit menjadi grafik, diagram batang, yang interaktif dan mudah dipahami.

Untuk menulis dan menjalankan kode tersebut, Visual Studio Code (VS Code) hadir sebagai salah satu lingkungan IDE terbaik untuk Python. Melalui dukungan ekstensi resmi, VS Code memungkinkan pengguna untuk menulis skrip, melakukan debugging (pelacakan kesalahan kode), serta menjalankan blok kode secara interaktif langsung di dalam satu aplikasi. Integrasi yang mulus ini membuat proses analisis data menggunakan Python di VS Code menjadi jauh lebih produktif, rapi, dan efisien.

2.4.3 Yahoo Finance

Yahoo Finance adalah salah satu library yang ada bagi python, library ini sebenarnya memiliki nama asli yfinance dan fungsinya sangat berguna untuk melakukan scraping data keuangan langsung dari situs yahoo finance. dengan menggunakan library ini bisa dengan mudah mengambil berbagai macam data pasar keuangan seperti harga saham, obligasi, mata uang kripto, hingga data laporan keuangan perusahaan tanpa perlu membayar biaya langganan API yang mahal.

selain itu library ini juga banyak digunakan karena cara penggunaannya yang sangat sederhana untuk mengunduh data historis dalam bentuk dataframe pandas sehingga bisa langsung diolah untuk analisis atau backtesting strategi trading. data yang bisa diambil juga cukup lengkap mulai dari data harian, mingguan, bulanan, bahkan sampai data dengan interval menit untuk kebutuhan analisis teknikal yang lebih mendalam pada pergerakan saham.

