

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Peninjauan terhadap penelitian-penelitian terdahulu menjadi langkah penting dalam memahami perkembangan metode prediksi gempa bumi serta menempatkan kontribusi penelitian ini dalam konteks yang tepat. Pada Tabel 2.1 merangkum penelitian terdahulu yang relevan, meliputi judul, algoritma yang digunakan, serta hasil utama dari masing-masing penelitian.

Tabel 2. 1 Penelitian Terdahulu

No.	Judul Artikel	Detail Penelitian	Hasil	Analisis Kritis
1.	[9]	Algoritma : LSTM dan Bi-LSTM Dataset : Tokai-Japan	Model LSTM mencapai akurasi 76% dan menunjukkan kinerja yang lebih baik dibandingkan Bi-LSTM berdasarkan metrik spesifikasi dan AUC.	LSTM lebih baik untuk klasifikasi kejadian, tetapi fokus pada klasifikasi.
2.	[10]	Algoritma : RNN, LSTM, BLSTM Dataset : Indonesia (Bulanan)	RNN nilai MAE 0.0558 dan RMSE 0.0803 pada prediksi, BLSTM memperoleh nilai MAE 0.0668 dan RMSE 0.858 pada prediksi.	LSTM unggul pada data temporal kompleks, namun prediksi jumlah gempa.
3.	[11]	Algoritma : <i>Attention-driven</i> LSTM	Model ini mencapai tingkat akurasi 90% dalam mengklasifikasikan semua gempa bumi, yang menunjukkan tingkat presisi yang tinggi dalam prediksinya.	Penelitian ini melakukan klasifikasi gempa bumi.

No.	Judul Artikel	Detail Penelitian	Hasil	Analisis Kritis
4.	[7]	Algoritma : LSTM, FFNN Dataset : Global	Model LSTM dengan hasil evaluasi MSE 0.6 dan R ² score -0,252.	Hasil model menunjukkan keunggulan LSTM dibandingkan FFNM biasa dalam tugas pemodelan rangkaian gempa bumi.
5.	[8]	Algoritma : RNN, LSTM Dataset : Indonesia	Dataset pola temporal yang kompleks LSTM lebih baik dengan hasil MAE 0,6226 dan RMSE 0,7731.	LSTM lebih baik dari RNN untuk data temporal kompleks, tetapi error masih relatif tinggi.
6.	[12]	Algoritma : Residual LSTM dengan Attention Dataset : Sulawesi (1975-2024)	Metode <i>attention-based residual</i> LSTM menunjukkan performa tinggi dengan nilai Testing MAE 0.0717, \Testing RMSE 0.0951, dan Huber Loss 0.0045 yang stabil untuk data training dan testing.	Stabil dan generalisasi bagus untuk return periods, tetapi dataset lokal terbatas.
7.	[13]	Algoritma : LSTM, Genetic Algorithm, Artificial Immune Network Dataset : AETA stations (6 stasiun)	Model SEGA-AIN-LSTM menunjukkan performa yang lebih baik dibandingkan metode tradisional.	Model yang dipilih mengalami generalisasi yang lemah ke tiap wilayah lain.
8.	[14]	Algoritma : <i>K-Means Clustering</i> , LSTM Dataset : Sulawesi	Hasil LSTM rata-rata evaluasi MSE: 0.379816, RMSE: 0.616292, dan MAE: 0.414586.	<i>Clustering</i> bantu prediksi lokasi-spesifik, tetapi error lebih tinggi dari model <i>advanced</i> .
9.	[15]	Algoritma : <i>Stacked Normalized Recurrent Neural</i>	Penelitian ini fokus pada wilayah <i>Horn of Africa</i> dengan rentang magnitudo 3.5–6.0 serta memakai model	Fokus pada transformer kompleks, komputasi tinggi.

No.	Judul Artikel	Detail Penelitian	Hasil	Analisis Kritis
		<i>Network (SNRNN)</i> Dataset : Horn of Africa (3.5-6.0 Mw)	transformer yang kompleks dan membutuhkan sumber daya tinggi.	
10.	[16]	Algoritma : <i>Hybrid Machine Learning dan Deep Learning</i> Dataset : Turki	Model Transformer+GP menunjukkan performa terbaik. RMSE terendah: 0.063 dan R ² tertinggi: 0.62	Penelitian menggunakan LSTM menangkap ketergantungan sekuensial tetapi mungkin masih kesulitan dengan pergeseran temporal yang kompleks.

Berdasarkan Tabel 2.1, *Long Short-Term Memory* (LSTM) menunjukkan keunggulan yang signifikan dalam prediksi gempa bumi, khususnya dalam menangani data temporal yang kompleks. Penelitian [10] dan [8] menunjukkan bahwa LSTM mampu mengungguli *Recurrent Neural Network* (RNN) dan Bi-LSTM dalam memodelkan pola temporal pada data gempa bumi di Indonesia, baik untuk prediksi jumlah kejadian maupun magnitudo, yang ditunjukkan oleh nilai MAE dan RMSE yang lebih rendah. Temuan ini mengindikasikan bahwa LSTM memiliki kemampuan yang lebih baik dalam menangkap dependensi temporal dibandingkan arsitektur sekuensial lainnya, sehingga sering dijadikan model dasar dalam penelitian prediksi gempa bumi.

Meskipun demikian, LSTM dasar masih memiliki keterbatasan yang perlu diperhatikan. Penelitian [7] menunjukkan bahwa performa LSTM dasar relatif terbatas, tercermin dari nilai MSE sebesar 0,6 dan nilai R² negatif, yang mengindikasikan kemampuan model yang rendah dalam menjelaskan variasi data serta kecenderungan *overfitting*, terutama pada dataset berukuran kecil. Selain itu, LSTM dasar juga kurang optimal dalam menangkap ketergantungan jangka panjang secara konsisten, sehingga performanya dapat

menurun pada data dengan pola yang sangat kompleks atau mengandung noise tinggi.

Demi mengatasi keterbatasan tersebut, penelitian [11] dan [7] mengusulkan integrasi *attention mechanism* ke dalam arsitektur LSTM. *Attention mechanism* memungkinkan model untuk memfokuskan pembelajaran pada informasi temporal yang paling relevan, sehingga meningkatkan stabilitas dan akurasi prediksi. Namun demikian, penerapan attention juga menimbulkan tantangan baru, yaitu *over-reliance*, di mana model menjadi terlalu bergantung pada mekanisme *attention* dan berpotensi mengabaikan kemampuan memori internal LSTM dalam menangkap ketergantungan jangka panjang.

Sebagai upaya untuk mengatasi permasalahan *over-reliance* sekaligus meningkatkan stabilitas pembelajaran, penelitian [12] mengusulkan penggunaan *residual connection* pada arsitektur LSTM+Attention. *Residual connection* berperan dalam menjaga stabilitas gradien dan memperlancar aliran informasi antar-lapisan, sehingga mampu mengurangi permasalahan *vanishing gradient* serta mencegah degradasi kinerja pada jaringan yang lebih dalam. Kombinasi *attention mechanism* dan *residual connection* memungkinkan model untuk belajar secara lebih seimbang, menghasilkan performa yang lebih stabil serta kemampuan generalisasi yang lebih baik pada data dengan pola temporal yang kompleks.

Hal ini sejalan dengan penelitian [13] yang menunjukkan bahwa penggunaan LSTM dengan *residual connection* tanpa *attention mechanism* dapat menimbulkan permasalahan baru, yaitu aliran informasi yang terlalu dominan sehingga mengurangi efektivitas *gating mechanism* pada LSTM. Kondisi ini berpotensi menyebabkan *overfitting* atau hilangnya informasi penting. Oleh karena itu, integrasi *attention mechanism* dan *residual connection* secara bersamaan menjadi pendekatan yang lebih tepat untuk mencapai performa optimal dalam prediksi gempa bumi.

Berdasarkan rangkuman penelitian yang disajikan pada Tabel 2.1, sebagian besar studi terdahulu masih menitikberatkan pada pemanfaatan data

temporal, seperti waktu kejadian dan magnitudo gempa bumi. Hal ini terlihat pada penelitian [10], [7], dan [8] yang berfokus pada prediksi jumlah kejadian atau magnitudo gempa berdasarkan data temporal di wilayah Indonesia dan Tokai–Jepang. Sementara itu, pemanfaatan data spasial, seperti koordinat geografis dan lokasi episentrum, masih relatif terbatas, meskipun dimensi ini memiliki potensi besar dalam meningkatkan akurasi prediksi serta memberikan pemahaman yang lebih komprehensif terhadap karakteristik gempa bumi. Beberapa penelitian, seperti penelitian [12] dan [14], mulai mengintegrasikan pendekatan spasial melalui *clustering* untuk analisis lokasi-spesifik, namun eksplorasinya masih belum mendalam.

Selain keterbatasan pada dimensi data, aspek metodologis dalam pemodelan juga masih menyisakan celah pengembangan. Pertama, sebagian besar penelitian hanya menguji model menggunakan satu konfigurasi parameter tertentu tanpa melakukan eksplorasi *hyperparameter* secara sistematis, sebagaimana terlihat pada penelitian [7] dan [8]. Padahal, proses *tuning hyperparameter* memiliki peran penting dalam mengoptimalkan kinerja model LSTM. Kedua, pengujian model pada rentang data yang lebih panjang atau pada dataset multsumber masih jarang dilakukan, sehingga kemampuan generalisasi model ke wilayah atau kondisi seismik yang berbeda menjadi terbatas. Hal ini sejalan dengan temuan pada penelitian [13] dan [16] yang menunjukkan bahwa model cenderung mengalami penurunan performa ketika diterapkan pada wilayah di luar data pelatihan.

2.2 Teori yang berkaitan

2.2.1 Gempa Bumi

Gempa bumi merupakan peristiwa bergetarnya permukaan bumi yang disebabkan oleh pelepasan energi secara tiba-tiba dari dalam bumi, yang menghasilkan gelombang seismik[17]. Peristiwa ini dapat terjadi akibat berbagai penyebab, seperti tumbukan antar lempeng tektonik, aktivitas sesar atau patahan, aktivitas vulkanik, hingga keruntuhan struktur bawah tanah seperti gua atau bekas tambang[18].

Secara umum, gempa bumi sendiri dapat dikategorikan menurut sumber penyebabnya sebagai berikut:

- 1) Gempa Runtuhan (*Collapse Earthquake*) yaitu gempa ini disebabkan oleh kerusakan atau ambruknya rongga di bawah tanah, seperti gua atau lubang bekas tambang[19].
- 2) Gempa Vulkanik (*Volcanic Earthquake*) yaitu gempa ini terjadi akibat pergerakan magma dan aktivitas di dalam gunung berapi[19].
- 3) Gempa Tektonik (*Tectonic Earthquake*) yaitu gempa ini merupakan yang paling berdampak dan sering terjadi. Gempa tektonik disebabkan oleh pergerakan lempeng di kerak bumi, seperti pergeseran patahan, pembentukan lipatan, atau proses geologis lain yang membentuk gunung[19].

Gempa bumi memiliki beberapa karakteristik penting. Salah satunya berlangsung dalam waktu yang sangat singkat dan terjadi pada lokasi tertentu. Selain itu, kekuatan dari peristiwa ini dapat diukur menggunakan alat yang disebut seismometer[20]. Salah satu metode pengukuran yang sering digunakan adalah *Moment Magnitude* (Mw), yang menggambarkan total energi yang dilepaskan selama gempa[21]. Dalam memahami dan memprediksi gempa bumi secara lebih mendalam, diperlukan pemahaman terhadap konsep-konsep dasar dalam ilmu geofisika, termasuk dinamika lempeng tektonik dan propagasi gelombang seismik[22].

2.2.2 Data Seismik

Data seismik merupakan hasil rekaman gelombang seismik oleh seismometer yang timbul akibat aktivitas gempa bumi[23]. Data ini menyimpan informasi penting seperti waktu kejadian gempa, lokasi episenter, kedalaman hiposenter, serta magnitudo. Pemanfaatan data seismik sangat krusial dalam menganalisis pola gempa, mengidentifikasi sumber gempa, hingga membangun model prediksi seismik untuk mendukung upaya mitigasi bencana[24].

Selain digunakan dalam konteks gempa, data seismik juga memiliki aplikasi luas dalam eksplorasi geologi. Melalui data ini, dapat diidentifikasi berbagai fitur geologi seperti struktur batuan, stratigrafi, patahan, hingga lapisan batuan dalam kerangka kronostratigrafi. Studi [25] menunjukkan bagaimana integrasi data seismik 3D resolusi tinggi dengan data sumur berhasil memetakan sistem delta paleogen dengan akurasi stratigrafi yang belum pernah dicapai sebelumnya. Fitur-fitur tersebut terekam sebagai reflektor yang terbentuk akibat perbedaan impedansi akustik antar lapisan. Ketika lapisan bawah lebih kompak, impedansi akustiknya meningkat dan menghasilkan refleksi positif. Amplitudo gelombang seismik sangat dipengaruhi oleh besar kecilnya kontras impedansi; semakin besar perbedaannya, semakin kuat pula amplitudo yang muncul[26]. Selain impedansi, faktor litologi, porositas, dan keberadaan fluida turut memengaruhi kekuatan amplitudo. Kehadiran fluida bahkan sering dikenali melalui indikator khusus yang disebut *Direct Hydrocarbon Indicator* (DHI), misalnya *flat event, bright spot, dan phase reversal*[27].

Lebih lanjut, data seismik memiliki variasi fase dan polaritas yang bergantung pada sumber gelombang maupun proses pengolahannya. Dalam tahap interpretasi, salah satu proses penting adalah *picking horizon*, yaitu penentuan batas lapisan batuan yang direpresentasikan oleh puncak gelombang (*wavelet peak*) pada data *zero phase*, atau titik *crossing* tertentu pada data dengan fase minimum[28]. Pada tahap ini, polaritas serta nilai impedansi akustik harus diperhatikan dengan cermat agar interpretasi lapisan dapat dilakukan secara akurat dan dapat dipercaya.

2.2.3 Data Spasial-Temporal

Data gempa bumi memiliki karakteristik spasial dan temporal yang sangat kompleks, yang menjadikannya subjek penelitian menantang dalam bidang geofisika[29].

- a) Dimensi spasial (ruang), data ini mencakup lokasi geografis episenter gempa (garis lintang dan bujur), kedalaman sumber gempa, serta sebaran area yang terkena dampak. Komponen spasial ini penting karena aktivitas seismik sering kali tidak terdistribusi secara acak, melainkan cenderung terkonsentrasi di zona-zona tektonik tertentu, seperti sepanjang batas lempeng. Analisis spasial bertujuan untuk mengidentifikasi pola, konsentrasi, dan korelasi antar kejadian dalam konteks geografis[30].
- b) Dimensi temporal (waktu), data gempa bumi mencakup waktu kejadian yang bersifat tidak beraturan dan dapat terjadi kapan saja. Pola kejadian gempa sering kali mengikuti urutan peristiwa yang kompleks, termasuk gempa pembuka (*foreshock*), gempa utama (*mainshock*), dan gempa susulan (*aftershock*)[29]. Karakteristik ini membuat pemrosesan data menjadi tantangan besar, sebab pola gempa bersifat non-linier dan sering kali berkelompok dalam *cluster* spasial-temporal[19].

Data spasial-temporal pada prediksi gempa bumi merupakan data yang menggabungkan informasi lokasi geografis (spasial) seperti koordinat lintang dan bujur dengan informasi waktu (temporal) kejadian gempa bumi. Data ini memungkinkan analisis pola kejadian gempa dalam dimensi ruang dan waktu secara simultan. Dalam konteks prediksi gempa, data spasial-temporal biasanya memuat atribut seperti waktu terjadinya gempa, lokasi episenter, kedalaman, dan magnitudo gempa[24]. Pendekatan ini memungkinkan pemodelan hubungan dinamis antara peristiwa gempa yang terjadi secara berurutan dalam ruang dan waktu, contohnya model ETAS (Epidemic Type Aftershock Sequence) spasial-temporal yang mengintegrasikan fungsi intensitas kejadian gempa berdasarkan koordinat geografis dan waktu kejadian,

sehingga dapat memprediksi aktivitas gempa susulan secara lebih akurat[25].

2.2.4 Magnitudo

Magnitudo adalah parameter objektif yang menyatakan besarnya energi seismik yang dilepaskan oleh sumber gempa, dihitung berdasarkan data rekaman seismograf. Nilai magnitudo bersifat absolut, artinya sama di semua lokasi pengamatan untuk satu kejadian gempa, tidak seperti intensitas yang bersifat relatif dan bervariasi tergantung lokasi. Secara umum, magnitudo gempa dapat dihitung menggunakan rumus empiris yang mempertimbangkan amplitudo maksimum gelombang seismik, periode gelombang, jarak episenter, dan kedalaman gempa. Salah satu rumus dasar yang dikembangkan oleh Charles F. Richter adalah:

$$M = \log_{10}(a) + f(\Delta, h) + C_S + C_R$$

Rumus 2. 1 Rumus Empiris Magnitudo

Keterangan :

- a. M = magnitudo
- b. a = amplitudo gerakan tanah (dalam mikron)
- c. T = periode gelombang
- d. $f(\Delta, h)$ = koreksi jarak dan kedalaman
- e. C_S, C_R = koreksi stasiuan dan daerah sumber

Magnitudo momen (Mw) merupakan skala yang paling akurat untuk gempa besar karena didasarkan pada momen seismik yang dihitung dari perubahan bentuk batuan selama gempa. Skala ini tidak mengalami saturasi pada gempa besar dan mampu menggambarkan energi secara lebih lengkap dibandingkan skala lainnya. Sementara itu, magnitudo gelombang badan (mb) dan magnitudo gelombang permukaan (Ms) digunakan untuk mengukur kekuatan gempa berdasarkan jenis gelombang yang direkam oleh seismograf, dengan ms lebih cocok untuk gempa dangkal dan berjarak jauh dari stasiun pengamatan.

2.2.5 Intensitas Gempa

Intensitas gempa bumi merupakan ukuran yang menggambarkan kekuatan guncangan atau dampak gempa yang dirasakan di suatu lokasi tertentu, bukan ukuran absolut seperti magnitudo. Intensitas sangat dipengaruhi oleh berbagai faktor lokal, seperti jarak dari pusat gempa, kedalaman sumber gempa, kondisi geologi tanah, dan jenis struktur bangunan di lokasi tersebut. Semakin dekat lokasi dengan pusat gempa, semakin tinggi intensitas yang dirasakan, begitu pula dengan daerah yang memiliki tanah lunak cenderung mengalami getaran lebih besar dibandingkan dengan daerah bertanah keras.

Skala intensitas yang paling umum digunakan adalah Skala Mercalli, yang menggunakan angka Romawi (I–XII) untuk menggambarkan dampak gempa mulai dari tidak terasa sama sekali hingga kehancuran total. Skala ini bersifat kualitatif, didasarkan pada pengamatan langsung terhadap reaksi manusia, kerusakan bangunan, dan perubahan permukaan tanah. Di samping itu, Skala Modified Mercalli (MMI) juga digunakan secara luas di berbagai negara, termasuk Indonesia, untuk menilai kerusakan dan dampak gempa di lokasi tertentu. Skala MMI lebih rinci dan mempertimbangkan berbagai aspek seperti kerusakan pada bangunan, reaksi manusia, dan perubahan pada lingkungan.

Intensitas bersifat relatif, artinya nilainya berbeda-beda tergantung lokasi pengamatan dan kondisi lokal. Hal ini membuat intensitas sangat penting dalam mitigasi bencana karena memberikan informasi yang relevan tentang dampak gempa di lokasi tertentu, yang tidak dapat dijelaskan hanya dengan magnitudo. Dengan memahami intensitas, pihak terkait dapat merancang bangunan tahan gempa, menentukan zona rawan bencana, dan menyiapkan rencana evakuasi yang efektif.

2.2.6 Mitigasi Bencana

Mitigasi bencana gempa bumi merupakan serangkaian upaya sistematis yang bertujuan untuk mengurangi risiko dan dampak negatif yang ditimbulkan oleh aktivitas seismik[29]. Upaya ini mencakup pengembangan sistem peringatan dini (*early warning system*) yang mampu memberikan informasi secara cepat dan akurat sebelum guncangan gempa utama terjadi, sehingga masyarakat dapat segera melakukan langkah penyelamatan. Selain itu, mitigasi juga diwujudkan melalui perencanaan tata ruang yang mempertimbangkan tingkat kerawanan gempa, misalnya dengan menghindari pembangunan di atas sesar aktif atau daerah dengan potensi likuefaksi tinggi. Dalam aspek fisik, pembangunan infrastruktur tahan gempa menjadi prioritas penting, yakni dengan menerapkan standar rekayasa struktur yang sesuai dengan regulasi bangunan tahan gempa agar konstruksi mampu menahan getaran dalam intensitas tertentu.

Tidak kalah penting, edukasi dan sosialisasi kepada masyarakat menjadi bagian integral dalam strategi mitigasi. Peningkatan kesadaran dan pengetahuan masyarakat mengenai prosedur evakuasi, penggunaan jalur aman, hingga simulasi bencana secara berkala dapat meningkatkan tingkat kesiapsiagaan. Di sisi lain, mitigasi juga memerlukan dukungan informasi prediksi dan analisis gempa yang akurat. Data seismik dan model prediksi berbasis kecerdasan buatan, misalnya, dapat memberikan gambaran mengenai potensi terjadinya gempa, distribusi intensitas getaran, maupun wilayah yang berisiko terdampak. Informasi ini sangat krusial mengingat gempa bumi memiliki sifat tiba-tiba (*sudden onset disaster*) dan berpotensi menimbulkan kerusakan dalam skala besar[29]. Dengan demikian, mitigasi bencana gempa bukan hanya tanggung jawab pemerintah, tetapi juga membutuhkan sinergi antara lembaga penelitian, institusi pendidikan, sektor swasta, serta partisipasi aktif masyarakat.

2.3 Framework/Algoritma yang digunakan

2.3.1. Knowledge Discovery in Databases (KDD)

Knowledge Discovery in Databases (KDD) merupakan suatu metodologi yang digunakan untuk mengekstraksi pengetahuan bernilai dari sekumpulan data melalui tahapan yang sistematis, terstruktur, dan saling berkaitan. KDD tidak hanya berfokus pada penerapan algoritma *data mining*, tetapi mencakup keseluruhan proses mulai dari pemahaman terhadap data hingga interpretasi hasil yang dihasilkan. Metodologi ini dirancang untuk memastikan bahwa pengetahuan yang diperoleh bersifat valid, relevan, dan dapat dimanfaatkan dalam pengambilan keputusan. Secara umum, proses KDD terdiri dari beberapa tahapan utama yang saling berkaitan dan dilakukan secara berurutan :



Gambar 2. 1 Diagram CRISP-DM

1) *Data Selection*

Tahap ini mencakup proses pemilihan data yang relevan dari berbagai sumber untuk memastikan bahwa data yang digunakan sesuai dengan tujuan analisis dan mengandung informasi yang diperlukan dalam proses penemuan pengetahuan.

2) *Data Preprocessing*

Pada tahap ini dilakukan pembersihan data untuk meningkatkan kualitas dataset, termasuk penanganan nilai hilang, penghapusan data duplikat, serta pengurangan *noise* dan *outlier* agar data menjadi lebih konsisten dan andal.

3) *Data Transformation*

Tahap *data transformation* bertujuan untuk mengubah data ke dalam format atau struktur yang sesuai untuk proses analisis dan pemodelan, seperti normalisasi, reduksi dimensi, atau pembentukan representasi fitur yang lebih informatif.

4) *Data Mining*

Tahap ini merupakan inti dari proses KDD, yaitu penerapan teknik atau algoritma tertentu untuk menemukan pola, hubungan, atau model prediktif dari data yang telah dipersiapkan.

5) *Evaluation dan Interpretation*

Tahap evaluasi dilakukan untuk menilai kualitas dan validitas pola atau model yang dihasilkan, serta menginterpretasikan hasil temuan agar dapat memberikan pemahaman yang bermakna dan bermanfaat bagi pengambilan keputusan.

2.3.2. Analisis Spasial-Temporal

Analisis spasial-temporal adalah metode yang mengkaji fenomena dengan mempertimbangkan aspek ruang (lokasi) dan waktu secara bersamaan[31]. Pendekatan ini lebih kaya dibandingkan analisis spasial atau temporal saja karena dapat menunjukkan pola yang bertahan dari waktu ke waktu sekaligus mengidentifikasi pola tidak biasa, seperti adanya kluster data yang mungkin terkait dengan risiko lingkungan baru atau kesalahan pencatatan[32]. Data spasial-temporal muncul ketika suatu informasi memiliki dimensi ruang (misalnya wilayah, koordinat, atau batas administrasi) sekaligus dimensi waktu (periode atau titik waktu tertentu)[33].

1) Kebutuhan Analisis Spasial-Temporal dalam Seismologi

Dalam penelitian prediksi gempa bumi, analisis spasial-temporal sangat dibutuhkan karena beberapa alasan kritis :

- a. Heterogenitas Tektonik: Gempa bumi di wilayah yang berbeda memiliki mekanisme pemicuan yang berbeda pula. Misalnya, gempa di Palung Sumatra (akibat subduksi lempeng) memiliki karakteristik kedalaman dan frekuensi yang berbeda dengan gempa di Sesar Palu-Koro (akibat pergeseran mendatar). Mengabaikan lokasi (spasial) dan memperlakukannya sebagai satu deret waktu tunggal akan mengaburkan pola unik dari setiap mekanisme tektonik[34][35].
- b. Variasi Pola Frekuensi dan Magnitudo: Pola frekuensi (seberapa sering gempa terjadi) dan pola magnitudo (seberapa besar energinya) bersifat spesifik lokasi (spasial). Klaster geografis yang berbeda akan menunjukkan pola temporal yang berbeda. Oleh karena itu, model prediktif harus mampu mengaitkan karakteristik temporal suatu kejadian dengan lokasi spasial-nya.
- c. Meningkatkan Akurasi Prediksi Lokal: Dengan menganalisis klaster spasial yang homogen, model time series (seperti LSTM) akan dilatih pada data yang memiliki pola perilaku yang lebih konsisten. Hal ini secara teoritis dapat menghasilkan prediksi temporal (magnitudo) yang lebih akurat dan relevan untuk zona geografis tertentu [34][35].

2.3.3. Clustering K-Means

K-Means Clustering sebagai Pra-Pemrosesan Spasial untuk mengintegrasikan dimensi spasial ke dalam model Deep Learning yang berfokus pada dimensi temporal (LSTM), diperlukan langkah pra-pemrosesan yang efektif. K-Means *clustering* merupakan salah satu metode *unsupervised learning* yang sering digunakan untuk mengelompokkan data berdasarkan kemiripan karakteristik spasial atau

geografis[36]. Dalam konteks analisis data gempa bumi, K-Means *clustering* dapat diterapkan untuk mengidentifikasi pola distribusi spasial kejadian gempa dengan membagi data ke dalam sejumlah kelompok (*cluster*) berdasarkan koordinat geografis seperti lintang dan bujur. Tujuan utama dari metode ini adalah meminimalkan jarak antara titik data dengan pusat *cluster* (*centroid*) yang mewakili kelompok tersebut, sehingga setiap *cluster* memiliki karakteristik spasial yang relatif homogen[36].

Proses *clustering* dimulai dengan menentukan jumlah *cluster* yang optimal, biasanya menggunakan metode seperti *elbow method* atau *silhouette analysis*. Setelah jumlah *cluster* ditentukan, algoritma K-Means akan mengelompokkan data gempa berdasarkan kedekatan geografis, sehingga dapat membantu mengurangi bias seleksi akibat disparitas geografis pada data. Hasil *clustering* ini memungkinkan peneliti untuk memahami pola distribusi spasial kejadian gempa, mengidentifikasi area rawan gempa, serta mempermudah analisis lebih lanjut terhadap karakteristik gempa di tiap kelompok.

2.3.4. Long Short-Term Memory

Long Short-Term Memory (LSTM) adalah jenis arsitektur jaringan saraf yang dirancang khusus untuk menangani masalah terkait urutan atau *time series* dalam *machine learning* dan *deep learning*[37]. LSTM memecahkan beberapa masalah yang dihadapi oleh model tradisional, seperti kesulitan dalam mengingat informasi jangka panjang. LSTM memiliki struktur yang lebih kompleks dibandingkan dengan *recurrent neural networks* (RNNs) konvensional. Komponen utama serta persamaan dalam arsitektur LSTM melibatkan:

1. *Cell State* (Sel): Memegang informasi jangka panjang dan mampu menjaga atau menghapus informasi berdasarkan kondisi tertentu.

$$C_t = f_t * C_{\{t-1\}} + i_t * \tilde{C}_t$$

Rumus 2. 2 Rumus cell state

Keterangan :

- a. C_t : Cell state pada waktu
 - b. t : memori utama LSTM yang menyimpan informasi jangka panjang.
 - c. f_t : Menentukan porsi memori lama $C_{\{t-1\}}$ yang dipertahankan (mendekati 1) atau dibuang (mendekati 0).
 - d. $C_{\{t-1\}}$: Cell state dari langkah sebelumnya yang membawa memori jangka panjang.
 - e. i_t : Mengatur seberapa banyak informasi baru akan dimasukkan ke dalam cell state.
 - f. \tilde{C}_t : Informasi baru hasil aktivasi tanh yang berpotensi ditambahkan ke memori utama.
2. *Forget Gate* (Pintu Penghilang): Menentukan sejauh mana informasi lama dalam sel harus diabaikan.

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

Rumus 2.3 Rumus forget gate

Keterangan:

- a. σ adalah fungsi sigmoid.
- b. h_{t-1} adalah hidden state dari timestep sebelumnya.
- c. x_t adalah input pada timestep saat ini.
- d. W_f dan b_f adalah parameter yang harus dipelajari.

3. *Input Gate* (Pintu Masukan): Mengatur sejauh mana informasi baru harus dimasukkan ke dalam sel.

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

Rumus 2.4 Rumus input gate LSTM

Keterangan:

- a. σ adalah fungsi sigmoid.
 - b. h_{t-1} adalah hidden state dari timestep sebelumnya.
 - c. x_t adalah input pada timestep saat ini.
 - d. W_f dan b_i adalah parameter yang harus dipelajari.
4. *Cell Update*: Menambahkan informasi baru ke dalam sel, yang diperbarui melalui pintu masukan.

$$\tilde{C}_t = \tanh(W_C \cdot [h_{\{t-1\}}, x_t] + b_C)$$

Rumus 2. 5 Rumus cell update LSTM

Keterangan :

- a. $h_{\{t-1\}}, x_t$ adalah penggabungan output LSTM sebelumnya (h_{t-1}) dan input pada waktu sekarang (x_t) menjadi satu vektor.
- b. W_C adalah *matrix* bobot yang dipakai untuk mengolah gabungan informasi tersebut. Semua pola dipelajari di sini.
- c. b_C adalah bias untuk menyesuaikan hasil perhitungan supaya lebih fleksibel.
- d. \tanh adalah aktivasi tanh akan mengubah hasil menjadi rentang (-1 sampai 1).
- e. \tilde{C}_t adalah *candidate cell state*, yaitu kandidat memori baru yang mungkin akan dimasukkan ke memori utama cell state.

5. *Output Gate* (Pintu Keluar): Menentukan sejauh mana informasi dalam sel harus disertakan dalam output model.

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

Rumus 2. 6 Rumus output gate LSTM

Keterangan :

- a. $[h_{\{t-1\}}, x_t]$ adalah gabungan *hidden state* sebelumnya dan input saat ini.
- b. W_o adalah bobot yang digunakan untuk menghitung output gate.
- c. b_o adalah bias untuk menyesuaikan hasil perhitungan.
- d. σ adalah aktivasi sigmoid yang menghasilkan nilai antara 0–1.
- e. o_t adalah output gate yang menentukan seberapa besar memori akan dikeluarkan ke *hidden state*.

LSTM memungkinkan informasi untuk mengalir melalui jaringan dengan lebih baik daripada RNN tradisional, membuatnya cocok untuk tugas-tugas yang melibatkan pemahaman konteks jangka panjang, seperti terjemahan bahasa, analisis wacana, dan prediksi deret waktu. LSTM adalah pendekatan yang menjanjikan untuk memprediksi data deret waktu, namun penelitian lebih lanjut diperlukan untuk mengeksplorasi potensinya dalam berbagai skenario[37]. Tabel 2.2 menunjukkan pseudo code algoritma LSTM :

Tabel 2. 2 Algoritma LSTM[38]

Algoritma	LSTM
Input	Input data sequence (x_t), previous cell state ($C_{\{t-1\}}$), previous hidden state (h_{t-1})
Output	Current cell state (C_t), current hidden state (h_t)
Langkah – langkah	<ol style="list-style-type: none"> 1. Inisialisasi parameter dan variabel 2. Hitung <i>forget gate</i> : $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$ <ol style="list-style-type: none"> 3. Hitung <i>input gate</i> : $it = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$ <ol style="list-style-type: none"> 4. Hitung <i>candidate cell state</i> :

$$\tilde{C}_t = \tanh(W_C \cdot [h_{\{t-1\}}, x_t] + b_C)$$

5. Update cell state :

$$C_t = f_t * C_{\{t-1\}} + i_t * \tilde{C}_t$$

6. Hitung output gate :

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

7. Hitung hidden state (output) :

$$h_{t-1} = o_t * \tanh(C_t)$$

8. Hitung metrik evaluasi antara prediksi dan nilai aktual

9. Akhiri iterasi

2.3.5. Attention Mechanism

Mekanisme *attention* adalah teknik dalam *deep learning* yang awalnya diperkenalkan untuk tugas *sequence-to-sequence*[39]. Teknik ini memungkinkan model untuk fokus pada bagian tertentu dari urutan input ketika menghasilkan keluaran[40]. Dengan cara ini, *attention* mengatasi keterbatasan memori internal berukuran tetap (seperti final *hidden state* pada *encoder*) dengan memberikan bobot yang berbeda pada tiap elemen input secara dinamis di setiap langkah keluaran. Sejak diperkenalkan, mekanisme ini menjadi landasan bagi banyak hasil state-of-the-art di bidang pemrosesan bahasa alami (NLP), terutama dengan hadirnya model Transformer yang sangat bergantung pada *self-attention* untuk memproses data[39]. Namun, mekanisme *attention* tidak hanya digunakan pada Transformer, tetapi juga dapat dipadukan dengan arsitektur rekuren seperti LSTM dan GRU untuk meningkatkan kemampuan dalam menangkap ketergantungan jangka panjang[39].

Secara sederhana, mekanisme *attention* memungkinkan model memusatkan perhatian pada bagian input yang paling relevan, mirip dengan cara manusia memperhatikan detail penting sambil mengabaikan hal yang kurang signifikan[40]. Mekanisme ini bekerja dengan menghitung bobot perhatian (*attention weights*) yang

menunjukkan seberapa besar pengaruh masing-masing elemen input terhadap keluaran yang dihasilkan. Mekanisme *attention* pada struktur encoder-decoder LSTM yang mana tiap keadaan tersembunyi h_s biasanya merupakan gabungan dari keadaan sel maju dan mundur LSTM pada saat itu s .

1) Hitung skor perhatian (*alignment score*) :

Pada tiap pasangan (h_t, h_s) dihitung skor yang menunjukkan seberapa relevan h_s terhadap h_t . Contoh fungsi penilaian umum :

1) *Dot-product* (kesederhaan) :

$$score(h_t, h_s) = h_t \cdot h_s$$

Rumus 2. 7 Rumus Skor Relevansi Multiplikatif

2) *Multiplicative / bilinear* (dengan bobot) :

$$score(h_t, h_s) = h_t \cdot W \cdot h_s$$

Rumus 2. 8 Rumus Skor Relevansi Multiplikatif dengan Matriks Bobot

3) Aditif (Bahdanau) :

$$score(h_t, h_s) = v \tanh(W_1 h_t + W_2 h_s)$$

Rumus 2. 9 Rumus hitung Skor Relevansi Aditif

Keterangan :

- a. h_t adalah *hidden state* pada waktu t (sebagai *query*).
- b. h_s adalah *hidden state* pada waktu s (sebagai *key*).
- c. $h_t \cdot h_s$ adalah perkalian *dot product* antara dua vektor tersebut.
- d. W adalah *matriks* bobot yang dipelajari untuk mentransformasikan h_s sebelum dihitung skornya.
- e. v adalah vektor bobot yang memproyeksikan hasil aktivasi menjadi satu nilai skor.
- f. \tanh adalah aktivasi yang menghasilkan representasi gabungan dari kedua *hidden state*
- g. $score(h_t, h_s)$ adalah nilai kesesuaian (*similarity*).

2) Hitung bobot perhatian (*attention weights*) :

Normalkan skor untuk menghasilkan distribusi probabilitas menggunakan fungsi *softmax* :

$$a_{t,s} = \frac{\exp(score(h_t, h_s))}{\sum_{s'} \exp(score(h_t, h_{s'}))}$$

Rumus 2. 10 Rumus hitung bobot attention

Keterangan :

- a. $score(h_t, h_s)$: Nilai kesesuaian antara *hidden state* pada waktu t dan s.
- b. $\exp(score(h_t, h_s))$: Mengubah skor menjadi nilai positif sehingga dapat dibandingkan secara proporsional.
- c. $\sum_{s'} \exp(score(h_t, h_{s'}))$: Total seluruh nilai eksponensial skor untuk semua posisi s' , digunakan sebagai penyebut untuk normalisasi.
- d. $a_{t,s}$: Bobot perhatian (*attention weight*); menunjukkan seberapa besar perhatian posisi t diberikan kepada posisi s. Nilainya berada antara 0–1.

Sebuah a_{ts} mewakili bobot perhatian untuk keadaan tersembunyi encoder h_s ketika decoding pada langkah waktu t .

3) Hitung vektor konteks (*context vector*)

Hitung vektor konteks untuk langkah waktu decoder t sebagai jumlah tertimbang dari keadaan tersembunyi encoder :

$$c_t = \sum_{s=1}^T a_{ts} \times h_s$$

Rumus 2. 11 Rumus hitung vektor konteks.

Keterangan :

- a. a_{ts} : Bobot perhatian yang menunjukkan seberapa besar kontribusi *hidden state* ke-s terhadap posisi waktu t.
- b. h_s : *Hidden state* pada posisi waktu s, yaitu informasi yang dibawa oleh LSTM pada langkah tersebut.
- c. Penjumlahan $\sum_{s=1}^T$: Menjumlahkan kontribusi seluruh *hidden state* sepanjang urutan untuk menghasilkan satu vektor konteks.
- d. c_t : Vektor konteks akhir yang mewakili informasi penting dari seluruh *time steps* berdasarkan bobot perhatian.

4) Menggabungkan konteks dengan keadaan decoder.

Biasanya ini melibatkan penggabungan c_t dan h_t dan memasukkan kombinasi ini ke lapisan padat untuk menghasilkan masukan untuk lapisan LSTM atau prediksi keluaran untuk langkah waktu tersebut.

$$\tilde{h}_t = \tanh(W_c[c_t; h_t])$$

Rumus 2.12 Rumus Penggabungan (Notasi Lanjutan)

Keterangan :

- a. c_t : Vektor konteks yang berisi informasi penting dari seluruh *time steps* berdasarkan perhatian (*attention*).
- b. h_t : *Hidden state* LSTM pada langkah waktu ke-t, yang menyimpan informasi lokal dari input saat itu.
- c. W_c : Matriks bobot yang memproyeksikan hasil gabungan tersebut ke ruang representasi baru.
- d. \tanh : Fungsi aktivasi yang memberi non-linearitas dan menormalkan output agar stabil.
- e. \tilde{h}_t : Vektor representasi baru yang telah “diperkaya” oleh mekanisme *attention* untuk langkah waktu ke-t.

2.3.6. Residual Connection

Residual connection dalam LSTM adalah teknik yang digunakan untuk mengatasi masalah hilangnya informasi pada jaringan neural yang dalam, terutama mengurangi masalah *vanishing gradient* yang kerap terjadi pada pelatihan model *deep learning* berlapis. *Residual connection* atau *residual learning* memperkenalkan *shortcut* atau jalur penghubung langsung yang melewati satu atau beberapa lapisan, sehingga output suatu lapisan bukan hanya hasil transformasi fungsi $F(x)$, tetapi juga dijumlahkan dengan input asli x [41]. Dalam konteks jaringan neural, ini dapat dituliskan sebagai:

$$y = F(x) + x$$

Rumus 2. 13 Rumus residual connection

Keterangan :

- a. x adalah input ke lapisan,
- b. $F(x)$ adalah fungsi transformasi lapisan tersebut,
- c. y adalah output lapisan dengan *residual connection*.

Residual connection dalam LSTM adalah teknik yang digunakan untuk mengatasi degradasi performa pada jaringan yang sangat dalam dengan cara menambahkan *shortcut* atau *skip connection* yang melewatkannya output dari lapisan sebelumnya langsung ke lapisan yang lebih dalam[42]. Dengan demikian, output pada lapisan ke- l pada waktu t , yang dinotasikan $h_t^{(l)}$, dapat dituliskan sebagai penjumlahan dari hasil transformasi non-linear pada output lapisan sebelumnya $f_l(h_t^{(l-1)})$ dan output lapisan sebelumnya yang dilewatkannya secara langsung $h_t^{(l-n)}$, dimana n adalah jumlah lapisan yang dilewati (biasanya 1 atau 2) :

$$\hat{h}_t^{(l)} = f_l(h_t^{(l-1)}) + h_t^{(l-n)}$$

Rumus 2. 14 Rumus detail residual connection

Keterangan :

- a. $h_t^{(l-1)}$: *Hidden state* dari layer sebelumnya, menjadi input utama layer saat ini.
- b. f_l : Fungsi transformasi pada layer ke-l yang mengolah input (misalnya linear + aktivasi).
- c. $h_t^{(l-n)}$: *Skip connection* dari layer lebih awal untuk menjaga informasi dan stabilitas gradien.
- d. $\hat{h}_t^{(l)}$: Output layer yang sudah diperbaiki oleh *residual connection*.

Pendekatan ini efektif mengatasi masalah *vanishing gradient* yang sering terjadi saat melatih jaringan yang dalam, sehingga memastikan informasi penting tetap mengalir dan tidak hilang selama proses pelatihan. Manfaat utama *residual connection* dalam LSTM antara lain[43]:

- 1) Mengurangi masalah *vanishing gradient* saat *training* jaringan dalam.
- 2) Mempercepat konvergensi pada pelatihan model.
- 3) Memungkinkan pembuatan model LSTM yang lebih dalam tanpa penurunan performa signifikan.
- 4) Meningkatkan akurasi dan stabilitas prediksi pada data sekuensial yang kompleks, seperti data gempa bumi.

Teknik ini merupakan salah satu cara efektif untuk meningkatkan performa model LSTM yang memiliki banyak lapisan dan data yang kompleks.

2.3.7. Adam Optimizer

Adam, singkatan dari *Adaptive Moment Estimation*, adalah algoritma optimasi yang sering digunakan dalam pelatihan model jaringan saraf[44]. Adam dapat dipandang sebagai kombinasi dari RMSprop dan *Stochastic Gradient Descent* dengan momentum. Dikembangkan oleh Diederik P. Kingma dan Jimmy Ba pada tahun

2014, Adam telah menjadi pilihan utama bagi banyak praktisi pembelajaran mesin. Ini menggunakan gradien kuadrat untuk skala *learning rate* seperti RMSprop, dan mengambil keuntungan dari momentum dengan menggunakan rata-rata bergerak dari gradien daripada gradien itu sendiri, seperti SGD dengan momentum. Ini menggabungkan *Dynamic Learning Rate* dan *Smoothening* untuk mencapai minimum global. Salah satu fitur *optimizer* Adam yaitu menyesuaikan tingkat pembelajaran untuk setiap parameter secara individual. Ini menghitung rata-rata bergerak dari momen orde pertama (rata-rata gradien) dan momen orde kedua (variasi gradien yang tidak terpusat) untuk skala *learning rate* secara adaptif[44]. Ini membuatnya sangat cocok untuk masalah dengan gradien yang jarang atau data yang bising.

2.3.8. Hyperparameter Tuning Random Search

Random search merupakan salah satu metode penyetelan parameter untuk model algoritma *machine learning*[39]. Metode random search melibatkan pengambilan sampel berulang kali dan mengevaluasi desain sistem berdasarkan riwayat yang diamati untuk mencari desain terbaik yang layak. Dalam konteks pengembangan model, ruang parameter merujuk pada kombinasi nilai-nilai yang mungkin untuk setiap parameter yang perlu dioptimalkan[45]. Dengan nilai parameter yang tepat dapat membantu mengoptimalkan hasil dari model algortima yang digunakan. Langkah-langkahnya melibatkan definisi ruang parameter, pemilihan acak kombinasi parameter, pelatihan model untuk setiap kombinasi, evaluasi kinerja model, pencatatan hasil, dan akhirnya, pemilihan kombinasi parameter terbaik berdasarkan hasil evaluasi[45].

2.3.9. Mean Squared Error

Mean Squared Error (MSE) adalah metrik evaluasi yang mengukur rata-rata kuadrat selisih antara nilai prediksi dan nilai aktual[46]. Hal ini dikarenakan penekanan pada kuadrat kesalahan,

MSE memberi bobot lebih besar pada kesalahan prediksi yang besar, sehingga sensitif terhadap outlier. Nilai MSE yang kecil menunjukkan model memiliki performa prediksi yang baik.

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

Rumus 2. 15 Rumus MSE

Keterangan :

- a. n adalah jumlah data sampel,
- b. y_i adalah nilai aktual dari sampel ke- i
- c. \hat{y}_i adalah nilai prediksi dari sampel ke- i .

2.3.10. Mean Absolute Error

Mean Absolute Error (MAE) adalah metrik evaluasi yang digunakan untuk mengukur tingkat kesalahan model prediksi dengan cara menghitung rata-rata nilai mutlak selisih antara nilai prediksi dengan nilai aktual. MAE memberikan gambaran seberapa jauh rata-rata kesalahan prediksi dalam satuan yang sama dengan data asli dan semakin kecil nilai MAE, maka performa model semakin baik.

Persamaan MAE dinyatakan sebagai:

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

Rumus 2. 16 Rumus MAE

Keterangan :

- a. n adalah jumlah data sampel,
- b. y_i adalah nilai aktual dari sampel ke- i
- c. \hat{y}_i adalah nilai prediksi dari sampel ke- i .

2.3.11. Root Mean Squared Error

Root Mean Square Error (RMSE) adalah metrik evaluasi yang umum digunakan untuk mengukur sejauh mana perbedaan antara nilai prediksi model dengan nilai aktual dalam suatu dataset. RMSE

memberikan gambaran tentang seberapa baik model dapat memperkirakan nilai sebenarnya, dan seberapa besar kesalahan yang dihasilkan[46]. Penggunaan RMSE pada model LSTM melibatkan perhitungan kesalahan kuadrat rata-rata antara prediksi dan nilai sebenarnya. Pada RMSE hasil yang dihasilkan semakin kecil yaitu mendekati 0 maka nilai RMSE tersebut memiliki hasil yang baik[47].

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

Rumus 2. 17 Rumus RMSE

Keterangan :

- a. y_i adalah nilai actual dari data observasional.
- b. \hat{y}_i adalah nilai prediksi dari model.
- c. n adalah jumlah observasi atau ukuran dataset.

2.4 Tools/software yang digunakan

2.4.1 Google Colaboratory

Google Colab, atau Colaboratory, adalah platform cloud computing gratis yang disediakan oleh Google untuk mendukung pengembangan dan pelatihan model *machine learning* seperti LSTM[48]. Platform ini didukung oleh layanan Google Cloud dan memungkinkan pengguna untuk membuat dan menjalankan *notebook Jupyter* di cloud tanpa memerlukan instalasi perangkat lunak di lokal. Colab mendukung berbagai pustaka dan *framework machine learning* terkemuka, termasuk TensorFlow, Keras, dan PyTorch. Ini membuatnya sangat mudah digunakan oleh para peneliti dan pengembang yang bekerja dengan berbagai alat dan teknologi *machine learning*[49]. Fitur yang sangat membantu pada Colab ialah kemampuan dalam memberikan akses ke unit pemrosesan grafis (GPU) dan uni pemrosesan tensor (TPU) tanpa biaya tambahan. Sehingga

dapat mempercepat waktu pelatihan model *machine learning* dan mengoptimalkan performa algoritma yang digunakan.

2.4.2 Python

Python adalah bahasa pemrograman yang populer dan mudah dipelajari. Keunggulan utamanya terletak pada sintaksis yang bersahabat dan fleksibilitasnya yang luar biasa. Python digunakan di berbagai bidang, mulai dari pengembangan web dan ilmu data hingga kecerdasan buatan dan pemrograman perangkat keras. Kekuatan Python terletak pada berbagai pustaka dan *framework* yang mendukung berbagai kebutuhan pengembangan. Salah satu fitur menonjol Python adalah mudah dibaca dan ditulis[49]. Sintaksisnya bersifat ekspresif, mirip dengan bahasa Inggris, sehingga meminimalkan kesalahan penulisan kode. Hal ini membuat Python sangat cocok untuk pemula sekaligus memberikan kecepatan dan kejelasan dalam pengembangan[49]. Python juga memiliki ekosistem pustaka yang kaya. Pustaka seperti NumPy, Pandas, dan Matplotlib mendukung analisis data dan visualisasi dengan mudah. Flask dan Django mempermudah pembuatan aplikasi web, sedangkan TensorFlow dan PyTorch menyediakan alat kuat untuk pengembangan model kecerdasan buatan.