BAB II
KAJIAN LITERATUR

2.1 Perubahan Iklim dan Emisi Karbon

Perubahan iklim global merupakan fenomena kompleks yang ditandai dengan peningkatan suhu rata-rata permukaan bumi akibat meningkatnya konsentrasi gas rumah kaca di atmosfer, terutama karbon dioksida (CO_2). Gas rumah kaca berperan dalam menjebak panas melalui mekanisme efek rumah kaca yang secara alami menjaga keseimbangan suhu bumi. Namun, peningkatan konsentrasi gas ini akibat aktivitas manusia telah menyebabkan ketidakseimbangan sistem iklim global [1,2]. Emisi karbon CO_2 dihasilkan dari berbagai aktivitas antropogenik seperti pembakaran bahan bakar fosil, produksi energi, transportasi, serta kegiatan industri. Sektor energi menjadi penyumbang terbesar emisi karbon global, diikuti oleh sektor industri dan transportasi. Selain CO_2, gas lain seperti metana (CH_4) dan dinitrogen oksida (N_2O) juga berkontribusi terhadap pemanasan global meskipun dalam jumlah yang lebih kecil [3].

Peningkatan emisi karbon memberikan dampak signifikan terhadap sistem iklim bumi, termasuk kenaikan suhu global, perubahan pola curah hujan, serta meningkatnya frekuensi kejadian cuaca ekstrem seperti badai dan gelombang panas. Data global menunjukkan bahwa emisi karbon CO_2 terus mengalami peningkatan dalam beberapa dekade terakhir seiring dengan pertumbuhan populasi dan kebutuhan energi [3]. Penelitian sebelumnya juga menunjukkan bahwa peningkatan konsentrasi CO_2 sejak era revolusi industri berkontribusi besar terhadap pemanasan global dan perubahan iklim [1,2]. Selain dampak lingkungan, peningkatan emisi karbon juga memengaruhi sektor ekonomi dan sosial. Ketidakstabilan iklim dapat berdampak pada produktivitas pertanian, ketersediaan sumber daya air, serta kesehatan manusia. Oleh karena itu, pemahaman terhadap pola dan tren emisi karbon menjadi sangat penting dalam mendukung pengambilan keputusan terkait perubahan iklim [10,11].

Dalam konteks analisis data, emisi karbon termasuk dalam kategori data deret waktu (*time series*) yang memiliki pola historis, tren, serta ketergantungan antar waktu. Karakteristik ini memungkinkan penggunaan pendekatan berbasis pembelajaran mesin, khususnya *deep learning*, untuk melakukan prediksi tren emisi karbon di masa mendatang. Model berbasis *deep learning* mampu menangkap pola kompleks dan hubungan *non-linear* dalam data. Dengan memahami pola historis tersebut, model prediksi dapat membantu dalam

pengambilan keputusan berbasis data. Pendekatan ini diharapkan dapat memberikan kontribusi dalam upaya mitigasi perubahan iklim secara lebih efektif.

2.1.1 Sumber Emisi Karbon

Sumber utama emisi karbon berasal dari aktivitas manusia yang berkaitan dengan penggunaan energi berbasis fosil. Beberapa sektor utama penyumbang emisi karbon dapat dilihat pada Tabel 2.1 di bawah ini:

Tabel 2.1 Sumber Utama Emisi Karbon Global [3]

No	Sumber Emisi	Deskripsi
1	Energi	Pembakaran batu bara, minyak, dan gas
2	Transportasi	Kendaraan bermotor dan penerbangan
3	Industri	Produksi manufaktur dan bahan kimia
4	Pertanian	Aktivitas peternakan dan penggunaan pupuk
5	Deforestasi	Penggundulan hutan dan perubahan lahan

Tabel 2.1 menunjukkan bahwa sektor energi menjadi kontributor terbesar dalam emisi karbon global, yang disebabkan oleh tingginya ketergantungan terhadap bahan bakar fosil seperti batu bara, minyak bumi, dan gas alam dalam memenuhi kebutuhan energi dunia. Penggunaan bahan bakar fosil tersebut tidak hanya dominan pada sektor pembangkit listrik, tetapi juga pada berbagai sektor lain seperti industri dan transportasi yang secara tidak langsung bergantung pada energi berbasis fosil. Selain itu, pertumbuhan populasi global serta peningkatan aktivitas ekonomi dan industrialisasi turut mendorong peningkatan permintaan energi secara signifikan. Kondisi ini menyebabkan konsumsi energi berbasis fosil terus meningkat dari waktu ke waktu sehingga berkontribusi langsung terhadap peningkatan emisi karbon di atmosfer. Ketergantungan yang tinggi terhadap energi fosil juga menunjukkan bahwa proses transisi menuju energi terbarukan masih menghadapi berbagai tantangan, baik dari sisi teknologi, biaya, maupun kebijakan. Oleh karena itu, sektor energi menjadi fokus utama dalam upaya pengurangan emisi karbon global. Analisis terhadap pola konsumsi energi dan emisi karbon pada sektor ini menjadi sangat penting untuk memahami tren emisi di masa mendatang serta sebagai dasar dalam pengembangan model prediksi yang akurat [3].

2.1.2 Dampak Emisi Karbon terhadap Lingkungan

Peningkatan emisi karbon memberikan berbagai dampak terhadap lingkungan global. Dampak utama yang ditimbulkan antara lain:

1. Pemanasan global

   Peningkatan suhu rata-rata bumi akibat akumulasi gas rumah kaca di atmosfer.

2. Perubahan pola iklim

   Perubahan musim, curah hujan, dan distribusi suhu global.

3. Kenaikan permukaan air laut

   Akibat mencairnya es di kutub dan pemuaian air laut.

4. Cuaca ekstrem

   Meningkatnya frekuensi badai, banjir, dan kekeringan.

Dampak-dampak tersebut menunjukkan bahwa emisi karbon memiliki pengaruh luas terhadap stabilitas lingkungan dan kehidupan manusia. Oleh karena itu, diperlukan pendekatan analitis untuk memahami dan memprediksi tren emisi karbon sebagai dasar dalam mendukung pengambilan keputusan terkait pengelolaan lingkungan [10,11].

2.2 Data Deret Waktu (*Time Series*)

Data deret waktu (*time series*) merupakan jenis data yang dikumpulkan dan diamati secara berurutan berdasarkan interval waktu tertentu, seperti harian, bulanan, atau tahunan. Data ini memiliki karakteristik utama berupa keterkaitan antar nilai pada waktu yang berbeda, sehingga nilai pada suatu waktu sangat dipengaruhi oleh nilai pada waktu sebelumnya. Dalam konteks emisi karbon, data yang digunakan umumnya berupa data historis tahunan yang mencerminkan perkembangan emisi dari waktu ke waktu [12]. Keunikan data *time series* terletak pada adanya pola tertentu yang dapat dianalisis dan dimanfaatkan untuk melakukan prediksi di masa mendatang. Pola-pola tersebut mencerminkan dinamika sistem yang mendasari data, sehingga pemodelan *time series* tidak hanya berfokus pada nilai data, tetapi juga pada hubungan temporal antar data. Oleh karena itu, analisis data deret waktu menjadi sangat penting dalam berbagai bidang, termasuk lingkungan, ekonomi, dan energi.

2.2.1 Karakteristik Data *Time Series*

Data *time series* memiliki beberapa karakteristik utama yang membedakannya dari jenis data lainnya. Karakteristik tersebut meliputi [12,13]:

1. Tren (*Trend*)

   Tren menunjukkan kecenderungan umum data dalam jangka panjang, apakah mengalami peningkatan, penurunan, atau relatif stabil. Dalam kasus emisi karbon, tren

biasanya menunjukkan peningkatan seiring dengan pertumbuhan industri dan konsumsi energi.

2. Musiman (*Seasonality*)

   Pola musiman merupakan pola berulang dalam periode waktu tertentu, seperti bulanan atau tahunan. Meskipun pada data emisi karbon global pola musiman tidak terlalu dominan, pada beberapa sektor tertentu pola ini tetap dapat muncul.

3. Fluktuasi Acak (*Noise*)

   Fluktuasi acak merupakan variasi yang tidak dapat diprediksi dan disebabkan oleh faktor eksternal yang tidak terkontrol.

4. Ketergantungan Waktu (*Temporal Dependency*)

   Nilai data pada waktu tertentu dipengaruhi oleh nilai sebelumnya, sehingga terdapat hubungan antar waktu yang kuat dalam data.

Karakteristik-karakteristik ini membuat data *time series* menjadi kompleks dan memerlukan metode khusus dalam proses analisis dan pemodelannya [12,13].

2.2.2 Contoh Pola Data *Time Series*

Untuk memahami pola dalam data *time series*, ilustrasi umum dapat dilihat pada Tabel 2.2 berikut.

Tabel 2.2 Tren Emisi karbon CO_2 Global Tahun 2019–2024 [3]

Tahun	Emisi karbon CO_2 (Gigaton)
2019	36.7
2020	34.8
2021	36.3
2022	37.1
2023	37.4
2024	38.6

Tabel 2.2 menunjukkan bahwa tren emisi karbon global mengalami fluktuasi dalam beberapa tahun terakhir. Pada tahun 2020 terjadi penurunan emisi karbon yang signifikan akibat pembatasan aktivitas global selama pandemi COVID-19. Namun, setelah itu emisi kembali meningkat pada tahun-tahun berikutnya seiring dengan pemulihan aktivitas ekonomi dan industri. Pola ini menunjukkan bahwa meskipun terdapat fluktuasi jangka pendek, tren jangka panjang emisi karbon tetap cenderung meningkat. Hal ini mengindikasikan adanya ketergantungan yang kuat terhadap aktivitas ekonomi global serta penggunaan energi berbasis fosil [3].

2.2.3 Relevansi *Time Series* dalam Prediksi Emisi Karbon

Dalam penelitian ini, data emisi karbon diperlakukan sebagai data deret waktu (*time series*) karena setiap nilai tersusun menurut tahun dan berkaitan dengan pengamatan sebelumnya. Istilah *non-linear* berarti perubahan emisi tidak selalu mengikuti hubungan lurus atau kenaikan yang tetap dari satu tahun ke tahun berikutnya. Kenaikan satu fitur juga tidak selalu menyebabkan perubahan CO_2 dengan besar yang sama pada semua negara dan periode. Kondisi tersebut dapat terlihat ketika emisi turun pada suatu tahun, kemudian meningkat kembali pada tahun berikutnya walaupun kecenderungan jangka panjangnya masih naik. Karakteristik ini menunjukkan bahwa pola emisi tidak cukup dijelaskan hanya dengan hubungan sederhana antarvariabel. Oleh karena itu, model perlu mempelajari hubungan waktu dan perubahan antarfaktor secara bersamaan.

LSTM digunakan karena dapat menerima data dalam bentuk urutan dan mempertahankan informasi dari pengamatan sebelumnya. Kemampuan tersebut membantu model mempelajari tren, perubahan mendadak, serta hubungan jangka panjang pada data emisi. Dalam penelitian ini, sepuluh pengamatan sebelumnya digunakan untuk memprediksi satu nilai CO_2 berikutnya. Delapan fitur terpilih dan riwayat CO_2 diproses bersama agar model tidak hanya bergantung pada satu indikator. Pendekatan ini sesuai dengan karakteristik data emisi yang memiliki pola berbeda antarnegara dan antarperiode. Dengan demikian, LSTM digunakan untuk membentuk prediksi berdasarkan pola historis yang tersedia, bukan untuk memastikan kondisi emisi pada masa depan [12-15].

2.2.4 Metode Pemodelan *Time Series*

Dalam analisis *time series*, terdapat dua pendekatan utama yang sering digunakan, yaitu metode statistik tradisional dan metode berbasis pembelajaran mesin. Metode statistik tradisional seperti *Autoregressive Integrated Moving Average* (ARIMA) banyak digunakan karena kemampuannya dalam memodelkan data *linear*, namun metode ini memiliki keterbatasan dalam menangani pola *non-linear* dan dependensi jangka panjang yang kompleks. Seiring dengan perkembangan teknologi, pendekatan berbasis *machine learning* dan *deep learning* mulai banyak digunakan dalam pemodelan *time series*. Metode *deep learning*, khususnya yang berbasis jaringan saraf, memiliki kemampuan untuk menangkap hubungan *non-linear* serta pola kompleks dalam data. Penelitian menunjukkan bahwa model *deep learning* seperti *Recurrent Neural Network* (RNN) dan *Long Short-Term Memory* (LSTM) mampu memberikan hasil prediksi yang lebih akurat dibandingkan metode tradisional dalam berbagai kasus *time series* kompleks [14,15]. Selain itu, model *hybrid*

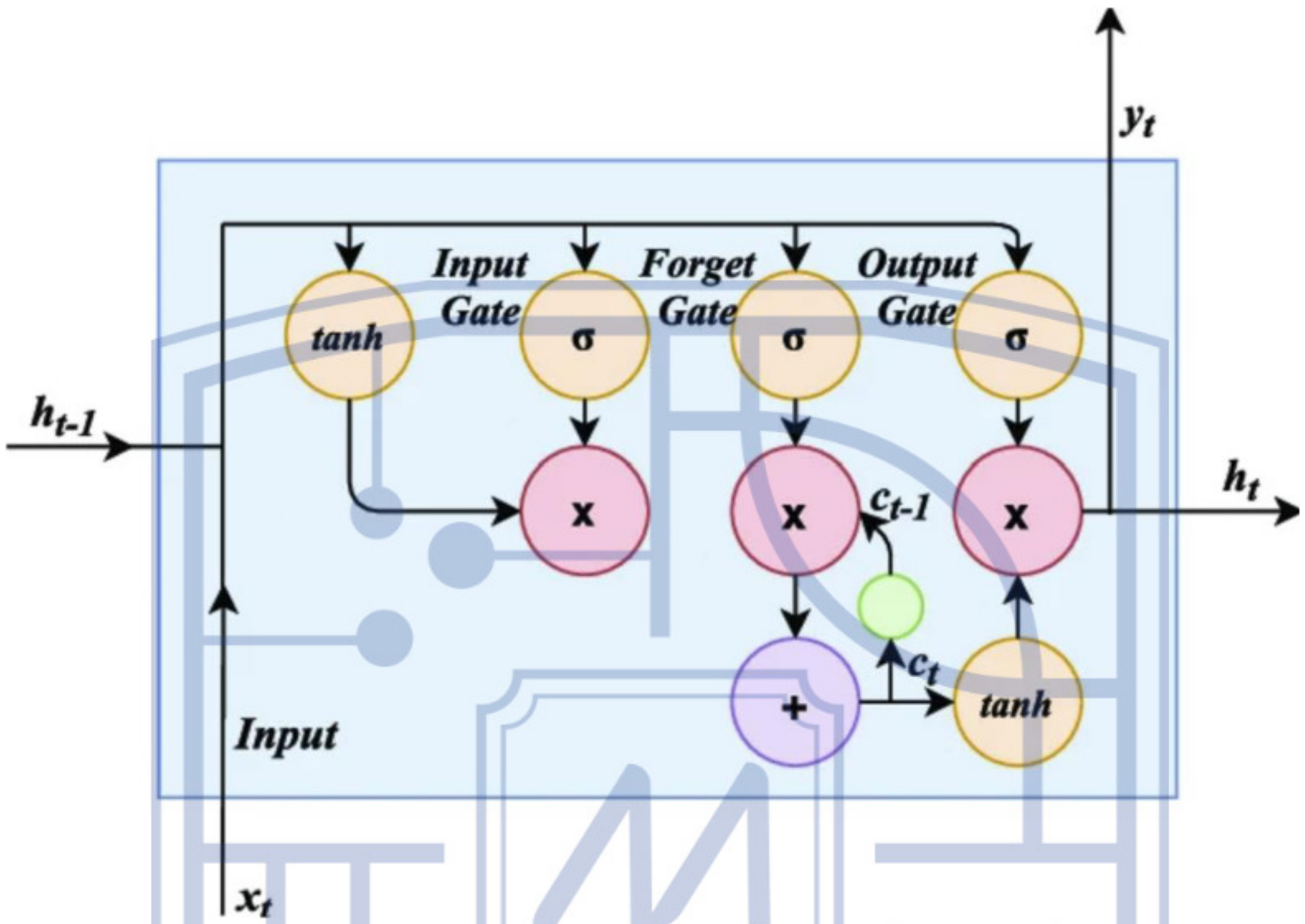
berbasis *deep learning* juga telah digunakan dalam prediksi emisi karbon untuk menangkap hubungan antar variabel dan pola data yang kompleks [16].

2.3 *Long Short-Term Memory* (LSTM)

Long Short-Term Memory (LSTM) merupakan pengembangan dari *Recurrent Neural Network* (RNN) yang dirancang untuk mengatasi keterbatasan dalam menangkap dependensi jangka panjang, khususnya akibat permasalahan *vanishing gradient*. LSTM mampu mempertahankan informasi dalam jangka waktu yang lebih panjang melalui mekanisme memori internal yang lebih kompleks dibandingkan RNN konvensional [8,17]. Keunggulan utama LSTM terletak pada kemampuannya dalam mengontrol aliran informasi yang masuk, disimpan, dan dikeluarkan dari memori melalui mekanisme *gates*. Mekanisme ini memungkinkan model untuk mempertahankan informasi penting serta mengabaikan informasi yang tidak relevan. Dengan kemampuan tersebut, LSTM menjadi efektif digunakan pada data sekuensial seperti data deret waktu (*time series*) [17,18]. Pada penelitian ini, model dilatih menggunakan data dari banyak entitas. Agar satu model dapat membedakan pola masing-masing entitas, identitas negara diubah menjadi *country_id* dan diproses melalui *country embedding*. *Embedding* menghasilkan representasi numerik berdimensi rendah yang dipelajari bersama model, sehingga identitas negara tidak diperlakukan sebagai angka berurutan.

2.3.1 Arsitektur LSTM

Arsitektur *Long Short-Term Memory* (LSTM) dapat dilihat pada Gambar 2.1 di bawah ini.



Gambar 2.1 Arsitektur *Long Short-Term Memory* (LSTM) [19]

Gambar 2.1 menunjukkan arsitektur *Long Short-Term Memory* (LSTM) yang terdiri dari beberapa komponen utama, yaitu *cell state* (c_t), *hidden state* (h_t), serta tiga mekanisme gerbang (*gate*), yaitu *input gate*, *forget gate*, dan *output gate*. Pada setiap langkah waktu t, data masukan x_t dan *hidden state* sebelumnya h_{t-1} diproses secara bersamaan untuk menghasilkan pembaruan pada sistem memori. *Input gate* (i_t) berfungsi untuk mengontrol seberapa besar informasi baru yang akan disimpan ke dalam *cell state*. *Forget gate* (f_t) menentukan bagian informasi dari *cell state* sebelumnya (c_{t-1}) yang perlu dipertahankan atau dihapus. Selanjutnya, informasi baru yang telah difilter akan dikombinasikan dengan informasi lama untuk menghasilkan *cell state* terbaru (c_t).

Output gate (o_t) kemudian mengatur informasi yang akan dikeluarkan sebagai *hidden state* (h_t), yang juga berfungsi sebagai *output* pada waktu tersebut. Proses ini melibatkan fungsi aktivasi seperti *sigmoid* (σ) untuk pengendalian gerbang dan *tanh* untuk normalisasi nilai. Mekanisme ini memungkinkan LSTM untuk mengontrol aliran informasi secara selektif dalam setiap langkah waktu. Struktur berulang ini memungkinkan LSTM untuk mempertahankan informasi jangka panjang serta mengatasi permasalahan *vanishing*

gradient yang umum terjadi pada model RNN tradisional. Dengan kemampuan tersebut, LSTM menjadi efektif digunakan pada data sekuensial seperti data deret waktu [19].

2.3.2 Mekanisme Kerja LSTM

Mekanisme kerja *Long Short-Term Memory* (LSTM) merupakan proses komputasi berulang yang terjadi pada setiap langkah waktu (*time step*) dalam data sekuensial. Pada waktu ke-t, model menerima *input* $x_t$serta *hidden state* sebelumnya h_{t-1}, yang kemudian diproses melalui beberapa gerbang (*gate*) untuk mengatur aliran informasi dalam jaringan. Tahapan proses LSTM dapat dijelaskan sebagai berikut:

1. *Forget Gate*

   Langkah pertama adalah menentukan informasi dari *cell state* sebelumnya ct−1 yang perlu dipertahankan atau dihapus. Proses ini dilakukan oleh *forget gate*:

   $$f_t = \sigma(U_f x_t + W_f h_{t-1} + b_f) \quad (2.1)$$

   Nilai f_t berada pada rentang 0 hingga 1, di mana nilai mendekati 0 menunjukkan informasi akan dihapus, sedangkan mendekati 1 berarti dipertahankan.

2. *Input Gate* dan Kandidat Memori

   Selanjutnya, *input gate* menentukan informasi baru yang akan ditambahkan ke dalam memori:

   $$i_t = \sigma(U_i x_t + W_i h_{t-1} + b_i) \quad (2.2)$$

   Nilai i_t berada pada rentang 0 hingga 1 dan menentukan seberapa besar informasi baru yang diteruskan untuk memperbarui *cell state*.

   Kandidat nilai memori baru dihitung sebagai:

   $$g_t = tanh(U_g x_t + W_g h_{t-1} + b_g) \quad (2.3)$$

   Nilai g_t merupakan kandidat informasi baru yang akan dipertimbangkan untuk dimasukkan ke dalam *cell state*. Nilai kandidat tersebut dihasilkan menggunakan fungsi aktivasi *tanh* agar informasi berada pada rentang −1 hingga 1.

3. Pembaruan *Cell State*

   Setelah informasi lama dan baru diperoleh, *cell state* diperbarui dengan menggabungkan keduanya:

   $$c_t = f_t \times c_{t-1} + i_t \times g_t) \quad (2.4)$$

   Persamaan tersebut menggabungkan informasi lama yang dipertahankan oleh *forget gate* dengan informasi baru yang dipilih oleh *input gate*. Hasil penggabungan tersebut membentuk *cell state* terbaru yang digunakan pada langkah waktu berikutnya.

4. *Output* Gate

Output gate menentukan bagian dari *cell state* yang akan dikeluarkan sebagai *output*:

$$o_t = \sigma(U_o x_t + W_o h_{t-1} + b_o) \quad (2.5)$$

Nilai o_t berada pada rentang 0 hingga 1 dan menentukan seberapa besar informasi dari *cell state* yang diteruskan sebagai *output*. Nilai yang mendekati 1 menunjukkan bahwa lebih banyak informasi diteruskan, sedangkan nilai yang mendekati 0 menunjukkan bahwa informasi lebih banyak ditahan.

5. *Hidden State* (*Output* Akhir)

   Output akhir dari LSTM dihitung sebagai:

$$h_t = o_t \times tanh(c_t) \quad (2.6)$$

   Nilai h_t digunakan sebagai *input* pada langkah waktu berikutnya sekaligus sebagai *output* LSTM pada waktu ke-*t*.

Secara keseluruhan, proses LSTM dimulai dengan seleksi informasi lama melalui *forget gate*, kemudian penambahan informasi baru melalui *input gate*, dilanjutkan dengan pembaruan memori pada *cell state*, dan diakhiri dengan produksi *output* melalui *output gate*. Mekanisme ini memungkinkan LSTM untuk mempertahankan informasi penting dalam jangka panjang serta mengatasi permasalahan *vanishing gradient* yang umum terjadi pada model *Recurrent Neural Network* (RNN) tradisional [19].

2.3.3 *Hyperparameter* pada LSTM

Dalam implementasi *Long Short-Term Memory* (LSTM), terdapat beberapa *hyperparameter* yang berperan penting dalam menentukan performa model dalam mempelajari pola data sekuensial. *Hyperparameter* merupakan parameter yang ditentukan sebelum proses pelatihan dan tidak diperbarui selama proses pembelajaran model. Beberapa *hyperparameter* utama pada LSTM antara lain [20]:

1. Jumlah unit LSTM (*units*)

   Menentukan jumlah neuron pada *layer* LSTM. Semakin banyak unit yang digunakan, semakin besar kemampuan model dalam menangkap pola kompleks pada data, namun juga meningkatkan risiko *overfitting*.

2. *Learning rate*

   Mengontrol besar kecilnya perubahan bobot pada setiap iterasi selama proses pelatihan. Nilai *learning rate* yang terlalu besar dapat menyebabkan model tidak konvergen, sedangkan nilai yang terlalu kecil dapat memperlambat proses pelatihan.

3. *Batch size*

Menentukan jumlah data yang diproses dalam satu iterasi pelatihan. *Batch size* yang kecil cenderung meningkatkan kemampuan generalisasi model, namun membutuhkan waktu pelatihan yang lebih lama.

4. *Epoch*

   Menunjukkan jumlah pengulangan pelatihan terhadap seluruh *dataset*. Pemilihan jumlah *epoch* yang tepat diperlukan untuk menghindari *underfitting* maupun *overfitting*.

5. *Dropout*

   Digunakan untuk mengurangi *overfitting* dengan cara menonaktifkan sebagian neuron secara acak selama proses pelatihan, sehingga model menjadi lebih *robust* (kemampuan model untuk tetap bekerja dengan baik meskipun ada gangguan pada data) terhadap variasi data.

6. *Optimizer*

   Merupakan algoritma yang digunakan untuk memperbarui bobot model, seperti Adam atau *Stochastic Gradient Descent* (SGD), yang berperan dalam mempercepat konvergensi model selama proses pelatihan.

Pemilihan *hyperparameter* yang tepat sangat penting dalam model LSTM karena memengaruhi kemampuan model dalam mempelajari pola temporal dan ketergantungan jangka panjang pada data deret waktu. Parameter seperti jumlah unit LSTM dan *epoch* berpengaruh terhadap kapasitas model dalam menangkap pola data, sedangkan *learning rate* dan *optimizer* memengaruhi kestabilan serta kecepatan konvergensi selama proses pelatihan. Selain itu, *batch size* dan *dropout* berperan dalam meningkatkan kemampuan generalisasi model serta mengurangi risiko *overfitting*. Oleh karena itu, pengaturan *hyperparameter* yang tepat diperlukan agar model LSTM dapat menghasilkan prediksi emisi karbon yang lebih akurat dan stabil [19,20].

EarlyStopping merupakan mekanisme yang dapat menghentikan pelatihan ketika metrik *validation* tidak lagi membaik. Mekanisme ini berguna untuk membatasi *overfitting*, tetapi juga membuat jumlah *epoch* efektif berbeda antar konfigurasi. Penelitian ini menguji *epoch* sebagai variabel eksperimen, sehingga setiap konfigurasi dijalankan sampai jumlah *epoch* yang ditetapkan agar paparan pelatihan dapat dibandingkan secara terkendali.

2.3.4 Kelebihan LSTM

LSTM memiliki beberapa keunggulan yang menjadikannya salah satu metode yang banyak digunakan dalam pemodelan data deret waktu. Keunggulan tersebut antara lain kemampuan dalam menangkap dependensi jangka panjang pada data sekuensial serta

stabilitas yang lebih baik dalam proses pelatihan dibandingkan RNN konvensional [17,18]. Selain itu, LSTM juga efektif dalam memodelkan hubungan *non-linear* dan mampu menangani data *time series* yang kompleks [14,21]. Berbagai penelitian menunjukkan bahwa LSTM memiliki performa yang baik dalam prediksi emisi karbon dan energi. Model ini terbukti mampu memberikan hasil yang lebih akurat dibandingkan metode konvensional dalam menangani data kompleks, khususnya pada kasus prediksi berbasis data historis [4-6].

2.4 *Preprocessing* Data

Preprocessing data merupakan tahap awal yang sangat penting dalam proses pengolahan data sebelum digunakan dalam pemodelan, khususnya pada pendekatan *deep learning*. Tahap ini bertujuan untuk meningkatkan kualitas data dengan cara mengurangi *noise*, menangani data yang tidak lengkap, serta memastikan data berada dalam format yang sesuai untuk diproses oleh model. Tanpa *preprocessing* yang baik, model berisiko menghasilkan prediksi yang tidak akurat karena belajar dari data yang tidak representatif [21]. Dalam konteks data deret waktu (*time series*) seperti emisi karbon, *preprocessing* memiliki peran yang lebih kompleks dibandingkan data statis. Hal ini disebabkan oleh adanya ketergantungan antar waktu (*temporal dependency*) serta pola historis seperti tren dan fluktuasi. Oleh karena itu, kesalahan dalam tahap *preprocessing* dapat menyebabkan distorsi pola temporal yang berdampak langsung pada penurunan performa model prediksi [12,21].

Selain itu, model *deep learning* seperti *Long Short-Term Memory* (LSTM) sangat sensitif terhadap skala dan distribusi data. Data yang tidak dinormalisasi atau mengandung anomali dapat mengganggu proses pembelajaran, memperlambat konvergensi, serta meningkatkan risiko *overfitting*. Kondisi ini menunjukkan bahwa *preprocessing* tidak hanya berfungsi sebagai tahap persiapan data, tetapi juga sebagai faktor penentu keberhasilan model dalam menangkap pola jangka panjang pada data deret waktu [12]. Dengan *preprocessing* yang tepat, model dapat belajar pola secara lebih efektif dan menghasilkan prediksi yang lebih stabil. Oleh karena itu, penerapan *preprocessing* yang sistematis menjadi langkah krusial dalam keseluruhan proses pemodelan.

Selain pembersihan dan normalisasi, penelitian ini menggunakan pemilihan fitur (*feature selection*) untuk membatasi jumlah variabel yang masuk ke model. Metode yang dipakai adalah *Recursive Feature Elimination* (RFE), yaitu proses menghapus fitur dengan tingkat kepentingan paling rendah secara bertahap sampai jumlah yang ditetapkan tercapai.

Random Forest Regressor digunakan sebagai penilai karena mampu membentuk banyak pohon keputusan dan menghitung tingkat kepentingan setiap fitur. Pada tahap ini, *Random Forest Regressor* tidak digunakan untuk menghasilkan prediksi akhir emisi karbon CO_2. Proses RFE dihentikan setelah diperoleh delapan fitur yang kemudian dipakai sebagai masukan LSTM. Pemilihan hanya menggunakan periode data *fit* agar data *validation* dan *testing* tidak memengaruhi fitur terpilih.

Landasan pemilihan fitur pada penelitian ini adalah prinsip parsimoni, yaitu mempertahankan sekumpulan variabel yang informatif dengan kompleksitas masukan yang lebih rendah. *Random Forest Regressor* digunakan sebagai *estimator* karena mampu menilai hubungan *non-linear* dan interaksi antarfaktor tanpa mensyaratkan bentuk hubungan tertentu. RFE kemudian menghapus fitur dengan tingkat kepentingan terendah secara bertahap. Jumlah delapan fitur ditetapkan sebagai rancangan masukan yang ringkas terhadap 58 kandidat yang lolos kelengkapan data, karena jumlah tersebut tidak diuji sebagai nilai optimum secara terpisah, istilah "fitur terbaik" dalam penelitian ini berarti delapan fitur berperingkat tertinggi menurut RFE pada data *fit*, bukan kombinasi terbaik secara universal.

Secara umum, *preprocessing data* terdiri dari beberapa tahapan utama sebagai berikut:

2.4.1 *Cleaning data*

Tahap *cleaning data* bertujuan untuk membersihkan data dari berbagai permasalahan seperti nilai yang hilang (*missing values*), data duplikat, serta data yang tidak konsisten agar kualitas data tetap terjaga sebelum proses pemodelan. Penanganan *missing values* dapat dilakukan dengan beberapa metode, seperti imputasi menggunakan nilai rata-rata (*mean imputation*), yaitu mengganti nilai yang hilang dengan rata-rata dari data yang tersedia pada variabel yang sama. Metode ini sederhana dan mudah diterapkan, namun dapat mengurangi variasi data jika digunakan secara berlebihan. Pada data deret waktu (*time series*), metode interpolasi juga sering digunakan untuk memperkirakan nilai yang hilang dengan menghubungkan titik data sebelum dan sesudahnya, misalnya menggunakan interpolasi *linear*. Pendekatan ini membantu menjaga kontinuitas pola data sehingga informasi temporal tetap terpelihara. Selain itu, penghapusan data (*deletion*) dapat dilakukan dengan menghapus baris yang memiliki nilai hilang apabila jumlahnya kecil dan tidak mempengaruhi distribusi data secara keseluruhan [22].

Imputasi median dipilih ketika distribusi fitur cenderung tidak simetris atau memuat nilai ekstrem, karena median ditentukan oleh posisi nilai tengah dan lebih tahan terhadap pengaruh observasi yang sangat besar atau sangat kecil dibandingkan rata-rata. Pada

penelitian ini, median dihitung per entitas dari data *fit* agar karakteristik setiap negara tetap dipertahankan. Median global dari data *fit* hanya digunakan sebagai *fallback* apabila median fitur pada suatu entitas tidak tersedia. Parameter tersebut kemudian diterapkan tanpa dihitung ulang pada *validation* dan *testing* sehingga tidak memasukkan informasi masa depan [22].

Selain penanganan *missing values*, pada data *time series* dilakukan deteksi dan penanganan *outlier*, yaitu nilai yang menyimpang secara signifikan dari pola normal data. *Outlier* dapat dideteksi menggunakan metode *Z-score*, di mana data dengan nilai lebih dari ±3 standar deviasi dari rata-rata dikategorikan sebagai *outlier*. Selain itu, metode *Interquartile Range* (IQR) juga dapat digunakan dengan menentukan batas bawah $Q1 - 1.5 \times IQR$ dan batas atas $Q3 + 1.5 \times IQR$. Data yang berada di luar rentang tersebut dianggap sebagai *outlier* yang perlu ditangani. Penanganan *outlier* dapat dilakukan dengan menghapus data, melakukan transformasi, atau mengganti nilai tersebut dengan nilai yang lebih representatif agar tidak menyebabkan bias dan ketidakstabilan pada hasil prediksi model [22].

2.4.2 Normalisasi Data

Normalisasi merupakan proses transformasi data ke dalam rentang tertentu, seperti [0,1] atau [-1,1], dengan tujuan untuk menyamakan skala antar fitur sehingga tidak terjadi dominasi nilai pada fitur tertentu. Tahap ini sangat penting dalam *deep learning* karena algoritma berbasis gradien sensitif terhadap perbedaan skala data, yang dapat mempengaruhi stabilitas dan kecepatan konvergensi model. Normalisasi diterapkan pada tahap pra-pemrosesan sebelum data digunakan dalam proses pelatihan model. Proses ini dilakukan dengan mentransformasikan seluruh fitur pada *dataset* pelatihan, kemudian parameter normalisasi yang diperoleh digunakan kembali pada data pengujian agar konsistensi skala tetap terjaga. Tanpa normalisasi, perbedaan skala antar fitur dapat menyebabkan model memberikan bobot yang tidak proporsional, memperlambat proses konvergensi, serta menurunkan stabilitas model. Dalam konteks data deret waktu (*time series*) dan model seperti LSTM, normalisasi membantu meningkatkan efisiensi pelatihan serta kestabilan model dalam menangkap pola data secara berurutan [20].

Salah satu metode normalisasi yang umum digunakan adalah *Min-Max Normalization*, yaitu metode yang mentransformasikan nilai data ke dalam rentang tertentu melalui penskalaan linier berdasarkan nilai minimum dan maksimum pada *dataset*. Secara matematis, metode ini dirumuskan sebagai berikut:

$$x' = (x - x\_min) / (x\_max - x\_min) \quad (2.7)$$

di mana x' merupakan nilai hasil normalisasi, x adalah nilai data asli, x_min adalah nilai minimum data, dan x_max adalah nilai maksimum data. Metode ini mempertahankan bentuk distribusi data serta hubungan antar nilai, sehingga cocok digunakan pada model berbasis *deep learning* yang memerlukan kestabilan skala *input*. Namun demikian, *Min-Max Normalization* memiliki kelemahan yaitu sensitif terhadap *outlier*, karena nilai minimum dan maksimum sangat mempengaruhi hasil transformasi. Oleh karena itu, metode ini umumnya digunakan setelah tahap *cleaning data* dilakukan untuk memastikan tidak terdapat nilai ekstrem yang dapat mengganggu proses normalisasi [22].

Nilai yang telah dinormalisasi perlu dikembalikan ke skala asli sebelum hasil prediksi dibaca dalam satuan CO_2. Proses pengembalian tersebut menggunakan *inverse transform* sebagai berikut:

$$x = x' \times (xmax - xmin) + xmin \quad (2.8)$$

Pada Persamaan (2.8), x′ adalah nilai pada skala normalisasi, sedangkan x adalah nilai yang telah dikembalikan ke skala asli. Nilai xmin dan xmax berasal dari data *fit* pada entitas yang sama.

2.4.3 Transformasi Data

Transformasi data dilakukan untuk mengubah struktur atau format data agar sesuai dengan kebutuhan model. Pada data *time series*, transformasi umumnya melibatkan pembentukan data menjadi urutan (*sequence*) berdasarkan jendela waktu tertentu (*sliding window*), sehingga model dapat mempelajari pola temporal dari data historis. Sebagai contoh, misalkan terdapat data emisi karbon dalam bentuk deret waktu: 100,110,120,130,140. Dengan menggunakan *sliding window* sebesar 3, maka data akan ditransformasikan menjadi:

1. *Input*: [100,110,120] →*Output*: 130
2. *Input*: [110,120,130] → *Output*: 140

Proses ini dilakukan dengan menggeser jendela secara bertahap sepanjang data, sehingga setiap urutan data sebelumnya digunakan untuk memprediksi nilai berikutnya. Dengan demikian, cara kerja transformasi ini adalah mengubah data deret waktu menjadi pasangan *input-output* berbentuk urutan, di mana model seperti LSTM menggunakan beberapa data sebelumnya sebagai masukan untuk memprediksi nilai pada waktu selanjutnya. Transformasi ini memungkinkan model untuk memahami hubungan antar waktu dan menangkap pola temporal secara lebih efektif [12].

2.4.4 Dampak *Preprocessing* terhadap Model

Tahap *preprocessing* memiliki pengaruh yang signifikan terhadap performa model *deep learning*. Data yang telah melalui *preprocessing* dengan baik cenderung menghasilkan model yang lebih stabil dan mampu belajar pola dengan lebih efektif. Selain itu, model juga akan memiliki tingkat kesalahan yang lebih rendah karena data yang digunakan lebih bersih dan terstruktur. Proses ini membantu model dalam melakukan generalisasi dengan lebih baik terhadap data baru yang belum pernah dilihat sebelumnya. Dengan demikian, kualitas *preprocessing* secara langsung mempengaruhi akurasi dan keandalan hasil prediksi model [20,21].

Sebaliknya, *preprocessing* yang tidak optimal dapat menyebabkan berbagai permasalahan dalam proses pelatihan model. Model dapat mengalami kesulitan dalam belajar pola yang sebenarnya terdapat dalam data, sehingga menghasilkan prediksi yang tidak akurat. Bahkan, dalam beberapa kasus, model dapat gagal menangkap informasi penting akibat adanya *noise*, data yang tidak konsisten, atau distribusi data yang tidak seimbang. Kondisi ini juga dapat menyebabkan proses pelatihan menjadi tidak stabil dan memerlukan waktu konvergensi yang lebih lama. Oleh karena itu, *preprocessing* tidak dapat dianggap sebagai tahap tambahan, melainkan sebagai komponen utama dalam *pipeline* pemodelan data yang menentukan keberhasilan model [12,21].

2.5 Pembagian Data

Pembagian data merupakan tahapan penting dalam proses pemodelan untuk mengevaluasi kinerja model secara objektif. Data umumnya dibagi menjadi dua bagian utama, yaitu data pelatihan (*training data*) dan data pengujian (*testing data*). Data pelatihan digunakan untuk melatih model dalam mempelajari pola dari data historis yang tersedia. Sementara itu, data pengujian digunakan untuk mengevaluasi kemampuan model dalam melakukan prediksi terhadap data yang belum pernah dilihat sebelumnya. Dengan adanya pemisahan ini, performa model dapat diukur secara lebih akurat dan tidak bias terhadap data yang digunakan selama proses pelatihan [12].

Dalam konteks data deret waktu (*time series*), pembagian data tidak dilakukan secara acak seperti pada data umum, melainkan harus mempertimbangkan urutan waktu. Hal ini bertujuan untuk menjaga konsistensi pola temporal serta menghindari kebocoran informasi (*data leakage*) dari masa depan ke masa lalu. Oleh karena itu, data dengan periode waktu lebih awal digunakan sebagai data pelatihan, sedangkan data pada periode berikutnya

digunakan sebagai data pengujian. Pendekatan ini memastikan bahwa model dilatih dan diuji dalam kondisi yang menyerupai situasi nyata. Dengan demikian, hasil evaluasi model menjadi lebih valid dan representatif terhadap kondisi sebenarnya [12,13].

Selain pembagian menjadi *training* dan *testing*, beberapa penelitian juga menambahkan data validasi (*validation data*) sebagai bagian dari proses pemodelan. Data validasi digunakan selama proses pelatihan untuk melakukan *tuning* parameter model agar diperoleh konfigurasi yang optimal. Penggunaan data validasi memungkinkan proses optimasi dilakukan tanpa mempengaruhi hasil evaluasi pada data pengujian. Selain itu, pembagian data juga bertujuan untuk menghindari *overfitting*, yaitu kondisi ketika model hanya menghafal data pelatihan namun gagal melakukan generalisasi pada data baru. Dengan strategi pembagian data yang tepat, model dapat menghasilkan performa yang lebih stabil dan dapat diandalkan.

2.6 Evaluasi Model

Evaluasi model merupakan tahapan penting untuk mengukur kinerja model dalam menghasilkan prediksi. Pada penelitian prediksi emisi karbon berbasis data deret waktu (*time series*), evaluasi dilakukan menggunakan *Mean Absolute Error* (MAE), *Root Mean Square Error* (RMSE), Koefisien Determinasi (R^2), dan *Mean Absolute Percentage Error* (MAPE) [12,13]. MAE dan RMSE mengukur besar kesalahan dalam satuan data, R^2 menilai kemampuan model mengikuti variasi nilai aktual, sedangkan MAPE menyatakan kesalahan relatif dalam persentase. Penggunaan keempat metrik secara bersamaan memberikan gambaran yang lebih lengkap daripada penggunaan satu metrik saja. Dengan demikian, evaluasi dapat menunjukkan ketepatan tingkat nilai sekaligus kemampuan model mengikuti perubahan data.

Dalam penelitian ini, hasil evaluasi dibedakan menjadi hasil sebelum koreksi dan hasil setelah koreksi. Hasil sebelum koreksi merupakan keluaran model sebelum penyesuaian, sedangkan hasil setelah koreksi diperoleh dengan menambahkan median *residual validation* pada setiap entitas. Nilai *residual validation* dihitung dari selisih antara nilai aktual dan hasil prediksi pada data *validation*. Data *testing* tidak digunakan dalam perhitungan koreksi agar hasil evaluasi tetap objektif dan tidak mengalami kebocoran data. Selain evaluasi prediksi satu langkah, penelitian ini juga menggunakan *recursive backtesting* untuk menilai kemampuan model ketika hasil prediksi sebelumnya digunakan kembali sebagai masukan untuk menghasilkan prediksi pada periode berikutnya.

Koreksi bias digunakan untuk mengurangi kecenderungan model yang terus-menerus menghasilkan prediksi terlalu rendah atau terlalu tinggi. Prosesnya dimulai dengan menghitung selisih antara nilai aktual dan nilai prediksi pada data *validation* untuk setiap entitas. Selisih tersebut disebut *residual*, kemudian nilai tengah atau median *residual* digunakan sebagai nilai koreksi karena lebih tahan terhadap selisih yang sangat besar atau ekstrem. Nilai koreksi selanjutnya ditambahkan pada hasil prediksi awal, jika median *residual* bernilai positif, prediksi akan dinaikkan, sedangkan jika bernilai negatif, prediksi akan diturunkan. Hasil koreksi juga dibatasi agar tidak menghasilkan nilai emisi di bawah nol. Apabila suatu entitas tidak memiliki data *validation* yang cukup untuk menghitung nilai koreksi, median *residual* dari seluruh entitas digunakan sebagai nilai pengganti.

Koreksi bias tidak melatih ulang model serta tidak mengubah arsitektur, bobot, atau pola yang telah dipelajari oleh LSTM. Proses ini hanya menyesuaikan tingkat hasil prediksi berdasarkan kecenderungan kesalahan yang ditemukan pada data *validation*. Data *testing* tidak digunakan untuk menentukan nilai koreksi karena data tersebut hanya dipakai untuk evaluasi akhir. Dengan demikian, proses koreksi bias tetap menjaga pemisahan data dan tidak memasukkan informasi dari data *testing* ke dalam hasil prediksi.

2.6.1 *Mean Absolute Error* (MAE)

MAE merupakan rata-rata dari nilai absolut selisih antara data aktual dan data hasil prediksi. MAE mengukur seberapa besar kesalahan prediksi tanpa memperhatikan arah kesalahan.

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

di mana:

yi = nilai aktual

$\hat{y}i$ = nilai prediksi

n = jumlah data

MAE memberikan interpretasi yang mudah karena berada dalam satuan yang sama dengan data asli, sehingga sering digunakan sebagai indikator dasar dalam evaluasi model [12].

2.6.2 *Root Mean Square Error* (RMSE)

RMSE merupakan akar dari rata-rata kuadrat selisih antara nilai aktual dan nilai prediksi. Metrik ini memberikan penalti yang lebih besar terhadap kesalahan yang besar (*large errors*).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (2.10)$$

RMSE sensitif terhadap *outlier* karena menggunakan kuadrat dari error. Oleh karena itu, metrik ini sering digunakan untuk mengevaluasi performa model pada data yang memiliki variasi tinggi [13].

2.6.3 *Mean Absolute Percentage Error* (MAPE)

MAPE mengukur rata-rata persentase kesalahan antara nilai aktual dan nilai prediksi. Metrik ini dinyatakan dalam bentuk persentase sehingga memudahkan interpretasi hasil evaluasi.

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{yi - \hat{y}i}{yi}\right| \quad (2.11)$$

MAPE banyak digunakan dalam analisis *time series* karena menyajikan besar kesalahan secara relatif terhadap nilai aktual. Nilai metrik dinyatakan dalam persentase sehingga hasilnya mudah dibandingkan antarperiode atau antarkumpulan data. Dalam implementasi penelitian ini, pengamatan dengan nilai aktual nol tidak dimasukkan ke dalam perhitungan agar tidak terjadi pembagian dengan nol. MAPE digunakan bersama MAE, RMSE, dan R^2 karena setiap metrik menjelaskan aspek kinerja yang berbeda. Penggunaan beberapa metrik memberikan penilaian yang lebih lengkap terhadap ketepatan hasil prediksi [12].

2.6.4 Koefisien Determinasi (R^2)

Koefisien determinasi (R^2) digunakan untuk melihat seberapa baik hasil prediksi mengikuti perubahan nilai aktual. Nilai yang mendekati 1 menunjukkan kesesuaian yang semakin baik. Nilai 0 menunjukkan bahwa hasil model tidak lebih baik daripada menggunakan rata-rata nilai aktual. R^2 dapat bernilai negatif apabila hasil prediksi lebih buruk daripada nilai rata-rata tersebut. Oleh karena itu, R^2 tetap dibaca bersama MAE, RMSE, dan MAPE [12,13].

$$R^2 = 1 - [\Sigma_{i=1}{}^{n}\,(y_i - \hat{y}_i)^2] \,/\, [\Sigma_{i=1}{}^{n}\,(y_i - \bar{y})^2] \quad (2.12)$$

di mana y_i adalah nilai aktual, $\hat{y}_i$ adalah nilai prediksi, $\bar{y}$ adalah rata-rata nilai aktual, dan n adalah jumlah data.

Dalam penelitian ini, R^2 dihitung setelah prediksi dikembalikan ke skala asli CO_2. Perhitungan dilakukan pada hasil sebelum dan setelah koreksi bias untuk kedua skema pembagian data. R^2 positif yang semakin besar menunjukkan bahwa model lebih mampu mengikuti variasi emisi antar pengamatan. R^2 negatif, terutama pada evaluasi per entitas atau *recursive backtesting*, menunjukkan bahwa perubahan tahunan belum diikuti model secara memadai. Interpretasi tersebut digunakan bersama besar galat absolut dan persentase untuk menjelaskan kemampuan serta keterbatasan model.

2.7 *Dataset Our World in Data*

Dataset yang digunakan dalam penelitian ini berasal dari *Our World in Data* (OWID), yaitu *platform* data terbuka yang menyediakan berbagai indikator global, termasuk data emisi karbon dioksida (CO_2). *Dataset* tersebut dihimpun dari berbagai sumber ilmiah dan lembaga internasional sehingga dapat digunakan sebagai data sekunder dalam penelitian [3]. Data emisi disajikan dalam bentuk deret waktu per negara dan entitas global, sehingga mendukung analisis tren serta pemodelan menggunakan *machine learning* dan *deep learning*.

Dataset OWID memiliki struktur yang konsisten dan dapat diakses dalam format CSV. Rentang data yang panjang memungkinkan model mempelajari perubahan emisi dari waktu ke waktu. Ketersediaan berbagai indikator juga mendukung proses pemilihan fitur, *preprocessing*, dan pemodelan LSTM. Dengan karakteristik tersebut, *dataset* ini sesuai untuk penelitian prediksi emisi karbon CO_2 berbasis data historis [10].

2.7.1 Karakteristik *Dataset*

Dataset emisi karbon dari *Our World in Data* memiliki beberapa karakteristik utama sebagai berikut [3]:

1. Bentuk data deret waktu (*time series*)

   Data disusun berdasarkan urutan waktu (misalnya tahunan), sehingga memungkinkan analisis tren historis.

2. Cakupan global

   Mencakup berbagai negara di dunia, sehingga dapat digunakan untuk analisis skala global maupun regional.

3. Variabel yang beragam

   Selain emisi karbon CO_2, *dataset* juga menyediakan variabel lain seperti konsumsi energi, populasi, dan indikator ekonomi yang dapat digunakan sebagai fitur tambahan.

4. Data terbuka dan terverifikasi

   Dataset dapat diakses secara bebas dan telah melalui proses kurasi dari sumber terpercaya.

2.7.2 Alasan Pemilihan *Dataset*

Pemilihan *dataset Our World in Data* dalam penelitian ini didasarkan pada beberapa pertimbangan [3,10]:

1. *Dataset* memiliki cakupan data yang luas dan representatif
2. Tersedia dalam bentuk *time series* yang sesuai untuk pemodelan LSTM
3. Memiliki kualitas data yang baik dan sering digunakan dalam penelitian ilmiah
4. Mendukung analisis tren jangka panjang emisi karbon

Dengan karakteristik tersebut, *dataset* ini dinilai mampu mendukung proses pengembangan model prediksi emisi karbon secara optimal serta menghasilkan hasil yang lebih akurat dan relevan terhadap kondisi nyata [10].

2.8 Kerangka Konseptual

Penelitian ini menggunakan pendekatan *deep learning* dengan model *Long Short-Term Memory* (LSTM) untuk memprediksi emisi karbon CO_2 berdasarkan data deret waktu (*time series*). Kerangka konseptual menggambarkan hubungan antara data masukan, penyiapan awal, pemilihan fitur, pembagian temporal, *preprocessing*, pembentukan *sequence*, pemodelan, evaluasi, dan keluaran prediksi. Setiap tahap menerima hasil dari tahap sebelumnya sehingga urutan proses dapat ditelusuri secara jelas. Parameter *preprocessing* dan pemilihan model hanya dibentuk menggunakan data *fit* serta *validation* agar data *testing* tidak memengaruhi proses pelatihan. Tahapan utama penelitian dijelaskan sebagai berikut:

1. Data Masukan

   Data masukan berasal dari *dataset Our World in Data* yang memuat 50.411 baris dan 79 atribut. Kolom *co2* digunakan sebagai *target*, sedangkan *country*, *year*, dan *iso_code* digunakan untuk mengidentifikasi entitas serta urutan waktu. Data disaring agar hanya memuat negara berkode ISO tiga huruf dan *World*. Pemodelan menggunakan periode 1970–2024 dan mempertahankan entitas yang mempunyai riwayat memadai. Hasil penyaringan membentuk data tahunan per entitas.

2. Penyaringan Data dan Pemilihan Fitur

Penyiapan awal dilakukan sebelum pembagian temporal untuk memastikan hanya data yang relevan digunakan dalam pemodelan. Tahapan yang diterapkan meliputi:

a. Penyaringan dan pembersihan data: mempertahankan negara berkode ISO tiga huruf dan entitas *World*, menghapus duplikat, mengeluarkan baris dengan *target* CO_2 kosong, serta membatasi periode pemodelan menjadi 1970–2024.
b. Pemeriksaan kandidat fitur: mempertahankan fitur yang memenuhi batas kelengkapan nilai dan memastikan setiap entitas mempunyai riwayat yang memadai untuk pembentukan *sequence*.
c. Pemilihan fitur: RFE dengan *Random Forest Regressor* memilih delapan fitur menggunakan periode data *fit* skema 70:30 tanpa melibatkan data *validation* dan *testing*.

3. Pembagian Temporal, *Preprocessing*, dan Pembentukan *Sequence*

Data yang telah melalui penyiapan awal selanjutnya dibagi secara kronologis menggunakan skema 70:30 dan 80:20. Setiap skema terdiri atas data *fit*, *validation*, dan *testing* dengan fungsi yang berbeda.

a. Data *fit* digunakan untuk memperbarui bobot model dan membentuk parameter *preprocessing*, sedangkan data *validation* digunakan untuk memilih konfigurasi serta menghitung koreksi bias.
b. Data *testing* hanya digunakan untuk evaluasi akhir. Setelah pembagian temporal, nilai kosong diisi, nilai ekstrem dibatasi, dan normalisasi dilakukan per entitas menggunakan parameter yang hanya dihitung dari data *fit*.

Data yang telah melalui *preprocessing* kemudian dibentuk menjadi *sequence* dengan *window* berukuran sepuluh pengamatan. Konteks sebelum batas *validation* dan *testing* tetap disertakan agar tahun pertama pada setiap bagian dapat diprediksi tanpa menjadikannya sebagai *target*. Urutan data tidak diacak sehingga hubungan temporal tetap terjaga dan *data leakage* dapat dicegah.

4. Pemodelan LSTM

Model menerima dua masukan yang diproses melalui cabang berbeda. Tahapan pemodelan mencakup:

a. *Sequence* numerik berukuran 10×9, yang terdiri atas delapan fitur terpilih dan satu riwayat CO_2 pada setiap langkah waktu.
b. *Country_id* disiapkan pada tahap *preprocessing* dan dipasangkan dengan setiap *sequence*. Di dalam model, identitas tersebut diubah menjadi *country embedding* berdimensi delapan agar tidak dianggap sebagai urutan angka.

c. Pengujian sembilan kombinasi *epoch* dan *batch size* dilakukan pada skema 70:30. Parameter model lainnya dibuat tetap, lalu kombinasi dengan MAE *validation* paling rendah digunakan pada model akhir skema 70:30 dan 80:20. Dua konfigurasi penelitian rujukan dibandingkan secara terpisah agar tidak bercampur dengan proses pemilihan [7].

Sequence dan *country_id* dipasangkan pada tahap *preprocessing*, tetapi keduanya tetap menjadi dua masukan yang berbeda. *Sequence* diproses oleh LSTM, sedangkan *country_id* diproses oleh *embedding* yang bobotnya dipelajari saat pelatihan. Hasil kedua jalur baru digabungkan di dalam model sebelum diteruskan ke *Dense*, *Dropout*, dan lapisan keluaran. Data *testing* tidak digunakan untuk memilih kombinasi. Susunan yang sama digunakan pada kedua skema agar perbandingan tetap adil.

5. Evaluasi Model

   Prediksi dikembalikan ke skala asli setiap entitas dan dievaluasi menggunakan:

   a. MAE dan RMSE untuk mengukur besar kesalahan prediksi dalam satuan CO_2.
   b. R^2 untuk menilai kemampuan model mengikuti variasi nilai aktual.
   c. MAPE untuk mengukur kesalahan relatif, disertai evaluasi sebelum dan setelah koreksi bias *validation*.

   Evaluasi dilakukan secara agregat dan secara khusus pada Indonesia serta *World*. Nilai aktual dan prediksi *World* tahun 2015–2024 dibandingkan pada periode yang sama. *Recursive backtesting* digunakan untuk mengukur dampak ketika prediksi sebelumnya menjadi riwayat bagi prediksi berikutnya. Koreksi bias dihitung dari median *residual validation* dan tidak menggunakan nilai aktual *testing*. Keempat metrik dibaca bersama untuk menjelaskan ketepatan nilai dan kemampuan mengikuti perubahan data.

6. Keluaran (Hasil Prediksi)

   Keluaran penelitian meliputi hasil pemilihan fitur, perbandingan *hyperparameter*, kurva pelatihan, metrik dua skema, dan evaluasi per entitas. Penelitian juga menghasilkan grafik aktual dan prediksi *World* tahun 2015–2024. *Recursive backtesting* tahun 2015–2024 dilakukan sebelum model diterapkan untuk proyeksi. Proyeksi *World* tahun 2025–2034 dilakukan secara *recursive* dengan memperbarui nilai CO_2 pada setiap langkah. Semua keluaran tersebut digunakan untuk menilai kemampuan dan batas model.

Kerangka konseptual ini menempatkan setiap proses sesuai urutan pelaksanaan penelitian. Data disiapkan dan fitur dipilih sebelum pembagian temporal dilakukan. Parameter *preprocessing* kemudian dibentuk dari data *fit* dan diterapkan pada data *validation* serta *testing* sebelum *sequence* dibentuk. Model selanjutnya dilatih, dievaluasi, dan diuji

melalui *recursive backtesting*. Proyeksi masa depan dilakukan setelah konfigurasi serta hasil evaluasi diperoleh sehingga hubungan antar tahap dapat dipahami dengan jelas.