

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian terdahulu merupakan salah satu dasar penting dalam penyusunan penelitian ini karena digunakan untuk memahami pendekatan, metode, hasil, serta keterbatasan dari penelitian yang telah dilakukan sebelumnya. Melalui penelitian terdahulu, peneliti dapat melihat bagaimana model *Deep Learning* diterapkan pada data berbasis *time series*, khususnya pada permasalahan prediksi yang memiliki pola berurutan, tidak linear, dan dipengaruhi oleh kondisi historis. Kajian ini juga digunakan untuk menemukan celah penelitian yang dapat dikembangkan lebih lanjut sesuai dengan fokus penelitian, yaitu prediksi nilai *sleep efficiency* menggunakan data *wearable* berbasis *time series*.

Berbagai penelitian sebelumnya telah memanfaatkan model *Deep Learning* seperti Long Short-Term Memory (LSTM), CNN-LSTM, CNN-BiLSTM, serta model hybrid lainnya untuk mempelajari pola temporal pada data. Model-model tersebut banyak digunakan dalam berbagai bidang, seperti kesehatan, energi, lingkungan, pertanian, ekonomi, dan finansial. LSTM digunakan karena mampu menangkap ketergantungan antarwaktu, sedangkan CNN dan BiLSTM sering dikombinasikan untuk menangkap pola lokal dan pola temporal secara lebih luas. Mekanisme *Attention* juga mulai digunakan dalam beberapa penelitian untuk membantu model mengidentifikasi bagian input yang lebih penting terhadap hasil prediksi.

Selain berfokus pada peningkatan performa model, beberapa penelitian juga mulai mengintegrasikan pendekatan *Explainable Artificial Intelligence (XAI)*, seperti *LIME*, *SHAP*, atau *Integrated Gradients Attribution*. Pendekatan ini digunakan untuk meningkatkan interpretabilitas model, terutama pada model *Deep Learning* yang sering dianggap sebagai *black box*. Dengan adanya *XAI*, hasil

prediksi tidak hanya dinilai berdasarkan nilai error atau akurasi, tetapi juga dapat dianalisis berdasarkan fitur-fitur yang paling berkontribusi terhadap output model.

Dalam penelitian ini, penelitian terdahulu digunakan sebagai acuan untuk menyusun pendekatan pemodelan, pemilihan algoritma, pemilihan metrik evaluasi, serta interpretasi hasil model. Kajian penelitian terdahulu juga menjadi dasar dalam membandingkan model LSTM, CNN-BiLSTM, dan CNN-BiLSTM dengan *Attention* untuk mengetahui model yang paling sesuai dalam memprediksi nilai *sleep efficiency* berbasis data *wearable*. Hasil kajian tersebut disajikan pada Tabel 2.1.

Tabel 2.1 Penelitian Terdahulu

Jurnal	Masalah	Metode	Hasil	Limitation
[9]	Kesulitan memetakan hubungan aktivitas fisik harian dengan waktu tidur nyenyak menggunakan data <i>wearable</i>	Perbandingan empat varian LSTM (stacked, dual input, autoencoder, 1D CNN-LSTM) pada data detak jantung dan langkah	Model Stacked LSTM sederhana memberikan hasil terbaik, model kompleks justru mengalami overfitting	RMSE/MAE masih cukup besar karena keterbatasan jumlah data, fitur fisik, dan faktor musiman
[11]	Prediksi daya solar photovoltaic sulit dilakukan karena data bersifat fluktuatif, tidak linear, dan dipengaruhi faktor meteorologi serta pola temporal yang kompleks.	Model deep hybrid CNN-BiLSTM yang dibandingkan dengan MLP, SVR, Random Forest, dan CNN-LSTM. CNN digunakan untuk menangkap pola lokal, sedangkan BiLSTM digunakan untuk menangkap dependensi temporal dua arah.	Model CNN-BiLSTM memperoleh performa terbaik dengan R^2 sebesar 0,99848, RMSE sebesar 0,5939, dan MAE sebesar 0,398, serta melampaui model klasik dan neural baseline.	Penelitian berfokus pada prediksi energi surya, sehingga penerapan pada domain lain seperti kesehatan atau <i>wearable</i> memerlukan penyesuaian karakteristik data.
[13]	Deteksi sleep apnea via PSG sangat mahal, serta model AI yang ada bersifat "kotak hitam" (tidak transparan)	<i>Explainable Artificial Intelligence (XAI)</i> menggunakan Decision Tree (ante-hoc) dan LIME (<i>post-hoc</i>) pada sinyal ECG	Mencapai akurasi 86,8% dan memberikan transparansi mengenai fitur klinis yang memengaruhi diagnosis	Perlunya investigasi fitur tambahan dan pengujian pada <i>dataset</i> yang lebih beragam
[12]	Model banjir fisik tradisional (TUFLOW) membutuhkan waktu simulasi yang sangat lama	Model surrogate hibrida CNN-LSTM; CNN untuk fitur spasial medan dan LSTM untuk fitur temporal hujan/pasut	Mengurangi waktu komputasi dari jam menjadi 3,2 menit dengan akurasi prediksi	Model tidak secara inheren mengodekan hukum fisik (seperti kekekalan massa) sehingga bisa

	untuk peringatan dini <i>real-time</i>		spasial genangan yang tinggi	terjadi inkonsistensi fisik
[14]	Kelangkaan data suhu air tambak ikan berkualitas tinggi yang membatasi performa model pembelajaran mesin murni	Physics-augmented LSTM yang mengintegrasikan model keseimbangan energi panas 1D ke dalam input LSTM	Meningkatkan RMSE sebesar 48,65% dibandingkan LSTM standar dan mengurangi ketergantungan pada jumlah data observasi	Prediksi terbatas pada jangka pendek (sampai 3 jam) dan ketergantungan pada kualitas input model fisik
[8]	Sulitnya memprediksi degradasi baterai litium-ion sejak dini karena sinyal penuaan pada 100 siklus pertama sangat samar	Kerangka kerja Enhanced Adaptive LSTM (EA-LSTM) dengan mekanisme adaptive gating dan modulasi berbasis atensi	Mengurangi kesalahan EoL sebesar 10-15% dan menunjukkan kemampuan transfer domain yang kuat antar protokol pengisian	Performa menurun drastis pada skenario zero-shot pada profil pemakaian yang sangat berbeda (<i>Mendeley dataset</i>)
[15]	Prediksi harga cryptocurrency sulit dilakukan karena pasar bersifat sangat volatil, tidak linear, dan memiliki ketergantungan temporal yang kompleks. Selain itu, model <i>Deep Learning</i> sering sulit dijelaskan.	Framework hybrid CNN-BiLSTM-TCN dengan analisis SHAP. CNN menangkap pola lokal, BiLSTM mempelajari dependensi temporal dua arah, TCN menangkap hubungan jangka panjang, dan SHAP digunakan untuk interpretasi fitur.	Model hybrid menghasilkan performa lebih baik dibanding model <i>Deep Learning</i> individual berdasarkan RMSE, MAE, dan R^2 . SHAP membantu menjelaskan fitur dan <i>time step</i> yang paling berpengaruh terhadap prediksi.	Penelitian hanya berfokus pada data cryptocurrency, khususnya Bitcoin, sehingga generalisasi ke domain lain perlu diuji kembali. Selain itu, model lebih kompleks karena menggabungkan beberapa arsitektur.
[16]	Prediksi GDP sulit dilakukan karena data ekonomi bersifat dinamis, tidak linear, dan memiliki ketergantungan temporal jangka panjang. Model statistik tradisional sering terbatas dalam menangkap pola tersebut.	Model CNN-BiLSTM yang dioptimasi menggunakan Genetic Algorithm, dilengkapi <i>feature selection</i> dan interpretasi menggunakan SHAP. CNN digunakan untuk menangkap pola lokal, sedangkan BiLSTM menangkap dependensi temporal dua arah.	Model CNN-BiLSTM memperoleh performa terbaik dibanding CNN, BiLSTM, GRU, Random Forest, XGBoost, Lasso, AdaBoost, dan VAR, dengan MSE 0,014943, MAE 0,008809, MAPE 0,890626%, dan R^2 0,985057.	Model membutuhkan proses optimasi dan <i>feature selection</i> yang cukup kompleks, sehingga memerlukan waktu eksperimen lebih besar dan bergantung pada kualitas indikator yang digunakan.
[17]	Prediksi hasil panen bit gula tradisional memakan waktu dan model yang	Model Stacked-LSTM menggunakan data deret waktu dari UAV (RGB &	Melampaui metode ML tradisional dengan R^2 0,761 untuk prediksi	Akurasi prediksi hasil akar dan hasil gula lebih rendah dibanding kandungan gula

	ada sering mengabaikan ketergantungan urutan pertumbuhan	multispektoral) dan faktor meteorologi	kandungan gula menggunakan data dari semua fase pertumbuhan	karena pengaruh tanah dan manajemen pertanian
[18]	Beban listrik perkotaan yang bersifat stokastik dan kurangnya transparansi model dalam memprediksi beban puncak	Model hibrida 1D-CNN-LSTM dengan IGA- <i>XAI</i> untuk memprediksi beban puncak siang, malam, dan luar puncak	Mencapai R^2 0,9515, sangat tangguh terhadap <i>noise</i> , dan memberikan interpretasi fitur meteorologi yang relevan	Penurunan performa jika prediksi dilakukan lebih dari 3-4 hari ke depan dan belum mempertimbangkan hari libur

Kajian penelitian terdahulu menunjukkan bahwa pemodelan *time series* pada berbagai bidang cenderung menghadapi persoalan yang sama, yaitu data bersifat berurutan, fluktuatif, tidak linear, dan dipengaruhi oleh kondisi historis. Pola tersebut terlihat pada data kesehatan berbasis *wearable* dan sinyal fisiologis [9][13], data lingkungan dan energi [11][12][14][18], data ekonomi dan cryptocurrency [15][16], data pertanian [17], serta data teknis seperti degradasi baterai [8]. Kesamaan karakter tersebut menunjukkan bahwa model prediksi tidak cukup hanya membaca nilai pada satu titik waktu, tetapi perlu memahami hubungan antarperiode agar pola historis dapat dimanfaatkan dalam proses prediksi.

Dari sisi tren metode, penelitian terdahulu memperlihatkan adanya pergeseran dari penggunaan model tunggal menuju model hybrid dan model yang lebih interpretatif. LSTM banyak digunakan karena mampu mempelajari ketergantungan temporal pada data berurutan, terutama pada data yang membutuhkan pemahaman pola historis seperti kualitas tidur berbasis *wearable*, suhu air tambak, dan hasil panen [9][14][17]. Pada penelitian lain, model hybrid seperti CNN-LSTM, CNN-BiLSTM, dan CNN-BiLSTM-TCN mulai digunakan untuk menggabungkan kemampuan ekstraksi pola lokal dan pembelajaran temporal, terutama pada data yang memiliki fluktuasi tinggi seperti energi surya, cryptocurrency, GDP, dan beban listrik [11][15][16][18]. Selain itu, *Attention* dan *XAI* juga mulai ditambahkan untuk membantu model mengenali bagian input yang lebih penting serta menjelaskan kontribusi fitur terhadap hasil prediksi, seperti pada degradasi baterai, sleep apnea, cryptocurrency, dan beban listrik [8][13][15][18].

Dari sisi hasil, model *Deep Learning* berbasis *time series* umumnya menunjukkan performa yang baik, tetapi hasilnya sangat dipengaruhi oleh karakteristik data dan target prediksi. Model CNN-BiLSTM pada prediksi daya energi surya memperoleh R^2 sebesar 0,99848, RMSE sebesar 0,5939, MAE sebesar 0,398, dan nRMSE sebesar 1,18% [11]. Pada prediksi GDP, CNN-BiLSTM yang dioptimasi menggunakan Genetic Algorithm memperoleh MSE sebesar 0,014943, MAE sebesar 0,008809, MAPE sebesar 0,890626%, dan R^2 sebesar 0,985057 [16]. Model 1D-CNN-LSTM pada prediksi beban listrik juga memperoleh R^2 sebesar 0,9602 untuk day peak, 0,948 untuk evening peak, dan 0,9464 untuk off-peak [18]. Hasil tersebut menunjukkan bahwa model hybrid dapat menghasilkan performa tinggi ketika struktur data memiliki pola lokal dan temporal yang kuat.

Namun, tren hasil penelitian juga menunjukkan bahwa arsitektur yang lebih kompleks tidak selalu menghasilkan performa terbaik. Pada penelitian kualitas tidur berbasis *wearable*, model LSTM yang lebih sederhana dapat memberikan hasil lebih baik dibandingkan model yang lebih kompleks karena model kompleks berpotensi mengalami overfitting [9]. Pada penelitian hasil panen bit gula, model stacked LSTM menghasilkan performa yang berbeda untuk setiap target, yaitu R^2 sebesar 0,761 untuk kandungan gula, 0,531 untuk hasil akar, dan 0,478 untuk hasil gula [17]. Selain itu, peningkatan performa pada model physics-augmented LSTM untuk prediksi suhu air tambak menunjukkan bahwa keberhasilan model tidak hanya ditentukan oleh arsitektur, tetapi juga oleh kesesuaian fitur dan informasi tambahan yang digunakan [14]. Dengan demikian, pemilihan model perlu disesuaikan dengan karakteristik data, fitur, target, dan risiko overfitting.

Gap penelitian yang terlihat adalah masih terbatasnya penelitian yang secara khusus memprediksi *sleep efficiency* sebagai nilai numerik kontinu berbasis data *wearable time series*. Penelitian terkait tidur masih lebih banyak membahas kualitas tidur secara umum atau deteksi gangguan tidur [9][13]. Sementara itu, penggunaan model hybrid, *Attention*, dan *XAI* sudah banyak diterapkan pada domain lain seperti baterai, energi, lingkungan, ekonomi, cryptocurrency, pertanian, dan beban listrik, tetapi belum banyak diarahkan pada prediksi *sleep efficiency* harian berbasis data *wearable*. Gap ini menunjukkan perlunya penelitian yang tidak hanya membangun

model prediksi, tetapi juga membandingkan beberapa arsitektur dan menjelaskan kontribusi fitur terhadap hasil prediksi.

Pemilihan LSTM, CNN-BiLSTM, dan CNN-BiLSTM dengan *Attention* dalam penelitian ini didasarkan pada tren dan gap tersebut. LSTM dipilih sebagai model dasar karena relevan untuk mempelajari pola historis pada data berurutan [9][14][17]. CNN-BiLSTM dipilih karena mampu menggabungkan ekstraksi pola lokal melalui CNN dan pembelajaran temporal dua arah melalui BiLSTM, sebagaimana ditunjukkan pada penelitian dengan performa tinggi di domain energi, ekonomi, dan beban listrik [11][16][18]. CNN-BiLSTM dengan *Attention* dipilih untuk menguji apakah *Attention* digunakan agar model dapat mengenali bagian *sequence* yang memiliki kontribusi lebih besar terhadap hasil prediksi, seperti pola aktivitas, stres, atau riwayat tidur pada periode tertentu, seperti yang digunakan pada penelitian degradasi baterai dan cryptocurrency [8][15]. Ketiga model tersebut dipilih dibandingkan metode *time series* klasik karena data penelitian ini bersifat multivariat, terdiri dari aktivitas harian, kondisi fisiologis, perilaku pengguna, stres, suasana hati, dan riwayat tidur, sehingga hubungan antarfiturnya berpotensi non-linear dan lebih sesuai dianalisis menggunakan pendekatan *Deep Learning* berbasis *sequence*.

2.2 Teori yang berkaitan

Teori yang berkaitan dalam penelitian ini digunakan sebagai landasan untuk memahami konsep utama yang mendukung proses prediksi *sleep efficiency* menggunakan data *wearable* berbasis *time series*. Bagian ini membahas teori mengenai kualitas tidur, *sleep efficiency*, *wearable device*, data *time series*, serta faktor-faktor yang berkaitan dengan efisiensi tidur. Pembahasan teori tersebut diperlukan agar variabel, data, dan pendekatan yang digunakan dalam penelitian memiliki dasar konseptual yang jelas.

Secara umum, penelitian ini berada pada ruang lingkup kesehatan digital dan analisis data berbasis *wearable device*. Data yang dihasilkan dari *wearable device* dapat merepresentasikan aktivitas harian dan kondisi fisiologis

pengguna, seperti jumlah langkah, detak jantung, tingkat stres, aktivitas olahraga, serta indikator tidur. Oleh karena itu, teori mengenai *wearable device* dan data *time series* menjadi penting untuk menjelaskan karakteristik data yang digunakan dalam penelitian.

Setelah landasan konseptual tersebut dijelaskan, bagian selanjutnya akan membahas masing-masing teori yang berkaitan dengan penelitian ini, dimulai dari konsep kualitas tidur.

2.2.1 Kualitas Tidur

Tidur merupakan proses biologis yang memiliki peran penting dalam menjaga keseimbangan fisiologis dan psikologis manusia. Selama tidur, tubuh melakukan berbagai proses pemulihan yang berhubungan dengan perbaikan energi, pengaturan hormon, pemulihan fungsi sistem saraf, serta pemeliharaan kondisi tubuh secara menyeluruh. Tidur juga berperan dalam fungsi kognitif, seperti konsolidasi memori, pengolahan informasi, regulasi emosi, serta kemampuan seseorang untuk berkonsentrasi dalam aktivitas sehari-hari [19].

Tidur yang berkualitas tidak cukup diukur dari jumlah jam tidur saja, melainkan juga dari bagaimana tidur tersebut berjalan, termasuk kenyamanan, kontinuitas, dan efektivitasnya. Seseorang dapat memiliki durasi tidur yang cukup, tetapi tetap merasa tidak segar apabila tidurnya sering terganggu atau tidak berlangsung secara efektif. Oleh karena itu, kualitas tidur perlu dilihat dari beberapa aspek, seperti durasi tidur, kontinuitas tidur, frekuensi terbangun pada malam hari, waktu yang dibutuhkan untuk mulai tidur, serta efisiensi tidur [6].

Kualitas tidur yang baik menunjukkan bahwa seseorang dapat memperoleh tidur yang cukup, nyenyak, dan bersifat restoratif. Tidur yang berkualitas dapat membantu tubuh melakukan pemulihan fisik dan mental setelah beraktivitas. Sebaliknya, kualitas tidur yang buruk dapat berdampak pada kelelahan, penurunan konsentrasi, gangguan suasana hati, serta menurunnya produktivitas harian [6].

Dalam konteks penelitian ini, kualitas tidur menjadi konsep utama yang menjadi dasar pemilihan variabel target. Namun, penelitian ini tidak melakukan diagnosis gangguan tidur secara klinis. Penelitian ini berfokus pada prediksi nilai *sleep efficiency* sebagai salah satu indikator numerik yang dapat merepresentasikan efisiensi tidur pengguna berdasarkan data *wearable* harian.

2.2.2 *Sleep efficiency*

Sleep efficiency merupakan salah satu indikator yang digunakan untuk menggambarkan seberapa efisien seseorang menggunakan waktu istirahatnya untuk tidur. Secara umum, *sleep efficiency* menunjukkan proporsi waktu tidur efektif dibandingkan dengan total waktu yang dihabiskan di tempat tidur. Semakin tinggi nilai *sleep efficiency*, maka semakin besar proporsi waktu di tempat tidur yang benar-benar digunakan untuk tidur. Sebaliknya, nilai *sleep efficiency* yang rendah dapat menunjukkan bahwa seseorang lebih banyak menghabiskan waktu di tempat tidur tanpa tidur secara efektif.

Sleep efficiency sering digunakan sebagai salah satu indikator kualitas tidur karena mampu menggambarkan efektivitas tidur seseorang secara kuantitatif. Indikator ini dapat digunakan sebagai ukuran objektif kualitas tidur karena merepresentasikan proporsi waktu tidur dibandingkan dengan waktu yang dihabiskan di tempat tidur. Pada pengukuran berbasis perangkat seperti actigraphy, *sleep efficiency* dapat dihitung dari total waktu tidur dibagi waktu di tempat tidur [20].

Selain melalui pengukuran objektif, *sleep efficiency* juga dapat dihitung melalui instrumen subjektif seperti Pittsburgh Sleep Quality Index (PSQI). Pada PSQI, komponen habitual *sleep efficiency* dihitung menggunakan rumus jumlah jam tidur dibagi jumlah jam yang dihabiskan di tempat tidur, kemudian dikalikan 100%. Nilai *sleep efficiency* yang lebih tinggi menunjukkan kondisi tidur yang lebih baik,

sedangkan nilai yang lebih rendah menunjukkan efisiensi tidur yang kurang baik [21].

Dalam konteks kesehatan, *sleep efficiency* juga memiliki relevansi karena berkaitan dengan kondisi fisiologis seseorang. Efisiensi tidur yang buruk dapat berhubungan dengan peningkatan risiko kesehatan tertentu, salah satunya risiko kardiovaskular pada kelompok dengan gangguan pernapasan tidur. Hal ini menunjukkan bahwa *sleep efficiency* tidak hanya menggambarkan kualitas tidur, tetapi juga dapat berkaitan dengan kondisi kesehatan secara lebih luas [21].

Dalam penelitian ini, *sleep efficiency* digunakan sebagai variabel target yang akan diprediksi. Variabel ini dipilih karena berbentuk numerik kontinu, sehingga sesuai digunakan dalam pendekatan regresi. Berbeda dengan klasifikasi yang membagi kualitas tidur ke dalam kategori tertentu, pendekatan regresi memungkinkan model memprediksi nilai *sleep efficiency* secara langsung. Dengan demikian, penelitian ini berfokus pada prediksi efisiensi tidur pengguna berdasarkan data *wearable* harian yang mencakup aktivitas, kondisi fisiologis, perilaku, suasana hati, dan riwayat tidur.

2.2.3 ***Wearable Device* untuk Pemantauan Tidur**

Wearable device merupakan perangkat teknologi yang dapat dikenakan pada tubuh pengguna untuk memantau aktivitas dan kondisi fisiologis secara berkelanjutan. Contoh *wearable device* yang umum digunakan adalah *smartwatch*, *fitness tracker*, dan gelang kesehatan. Perangkat ini umumnya dilengkapi dengan sensor yang dapat merekam berbagai data, seperti jumlah langkah, aktivitas fisik, detak jantung, variabilitas detak jantung, kadar oksigen dalam darah, hingga estimasi pola tidur [6].

Dalam pemantauan tidur, *wearable device* memiliki keunggulan karena dapat digunakan dalam kehidupan sehari-hari secara lebih praktis dan non-invasif. Berbeda dengan metode klinis seperti polysomnography yang membutuhkan peralatan khusus dan umumnya dilakukan dalam

lingkungan klinis, *wearable device* memungkinkan pengumpulan data secara berkelanjutan dalam kondisi aktivitas normal pengguna [6]. Hal ini membuat *wearable device* dapat memberikan gambaran yang lebih luas mengenai pola aktivitas dan tidur pengguna dalam jangka waktu tertentu.

Data yang dihasilkan oleh *wearable device* dapat digunakan untuk menganalisis hubungan antara aktivitas harian dan kualitas tidur. Misalnya, jumlah langkah dapat menggambarkan tingkat aktivitas fisik pengguna, detak jantung dapat merepresentasikan kondisi fisiologis, HRV dapat berkaitan dengan pemulihan tubuh dan stres fisiologis, sedangkan SpO₂ dapat memberikan informasi mengenai kadar oksigen dalam darah. Selain itu, fitur seperti konsumsi kafein, konsumsi alkohol, durasi olahraga, tingkat stres, dan suasana hati juga dapat digunakan sebagai variabel pendukung dalam memprediksi *sleep efficiency*.

Dalam penelitian ini, data *wearable* digunakan sebagai sumber utama dalam membangun model prediksi. Data yang digunakan mencakup aktivitas harian, kondisi fisiologis, perilaku pengguna, suasana hati, serta riwayat nilai *sleep efficiency*. Dengan memanfaatkan data tersebut, model diharapkan mampu mempelajari pola yang berhubungan dengan efisiensi tidur pengguna.

2.2.4 Data Time Series pada Wearable Device

Data *time series* merupakan data yang tersusun berdasarkan urutan waktu. Setiap data dicatat pada periode tertentu dan memiliki kemungkinan keterkaitan dengan data pada periode sebelumnya. Pada data *time series*, urutan data menjadi aspek penting karena perubahan nilai dari waktu ke waktu dapat membentuk pola tertentu yang relevan untuk dianalisis.

Data yang diperoleh dari *wearable device* termasuk ke dalam data *time series* karena perangkat tersebut dapat merekam aktivitas dan kondisi fisiologis pengguna secara berkelanjutan. Data seperti detak jantung, jumlah langkah, aktivitas fisik, dan indikator tidur dicatat berdasarkan waktu tertentu, sehingga dapat digunakan untuk mempelajari perubahan

kondisi pengguna dari waktu ke waktu. Penelitian sebelumnya menunjukkan bahwa data seperti heart rate dan jumlah langkah dari *wearable device* dapat digunakan sebagai fitur *time series* dalam pemodelan kualitas tidur [9].

Pada data *wearable*, pola temporal menjadi hal yang penting untuk diperhatikan. Aktivitas fisik, detak jantung, tingkat stres, dan indikator fisiologis lainnya tidak selalu berdiri sendiri pada satu waktu tertentu, tetapi dapat memiliki hubungan dengan kondisi pengguna pada waktu sebelumnya. Oleh karena itu, pendekatan *time series* dapat membantu model dalam menangkap pola historis yang muncul dari data *wearable*. Pemodelan berbasis *time series* juga dinilai sesuai untuk data *wearable* karena model perlu menangkap pola temporal dari parameter fisiologis dan aktivitas pengguna agar hasil prediksi menjadi lebih baik [9].

Selain data aktivitas seperti jumlah langkah, *wearable device* juga dapat menghasilkan data fisiologis lain seperti ECG dan accelerometer. Data tersebut menunjukkan bahwa *wearable device* dapat menangkap sinyal multimodal yang merepresentasikan kondisi fisiologis dan perilaku pengguna dalam kehidupan sehari-hari. Sinyal fisiologis dan perilaku tersebut memiliki struktur temporal, sehingga dapat digunakan dalam berbagai analisis berbasis *time series* [22].

Dalam penelitian ini, data *wearable* harian digunakan untuk memprediksi nilai *sleep efficiency*. Setiap pengguna memiliki data yang dicatat berdasarkan urutan tanggal, sehingga data tidak dapat diperlakukan sebagai data acak biasa. Data pada hari sebelumnya dapat berhubungan dengan nilai *sleep efficiency* pada hari berikutnya atau pada periode tertentu. Oleh karena itu, urutan waktu pada masing-masing pengguna perlu dipertahankan dalam proses analisis.

Untuk menangkap pola historis tersebut, penelitian ini membentuk data menjadi *sequence* menggunakan metode *sliding window*. Panjang window yang digunakan adalah 14 hari. Dengan pendekatan ini, setiap input model berisi data *wearable* pengguna selama 14 hari berturut-turut. Pembentukan *sequence* dilakukan secara terpisah berdasarkan *user_id* agar

data antar pengguna tidak tercampur dan pola waktu masing-masing pengguna tetap terjaga.

Penggunaan data *time series* pada penelitian ini bertujuan agar model dapat mempelajari pola aktivitas, kondisi fisiologis, perilaku, dan riwayat tidur pengguna dalam rentang waktu tertentu. Dengan demikian, model tidak hanya memanfaatkan informasi dari satu hari, tetapi juga mempertimbangkan pola historis yang terbentuk dari data *wearable* harian.

2.2.5 Faktor-Faktor yang Berkaitan dengan *Sleep efficiency*

Sleep efficiency dapat dipengaruhi oleh berbagai faktor yang berkaitan dengan aktivitas harian, kondisi fisiologis, perilaku, psikologis, dan riwayat tidur pengguna. Oleh karena itu, fitur yang digunakan dalam penelitian ini tidak hanya berasal dari satu jenis variabel, tetapi mencakup beberapa kelompok fitur yang memiliki keterkaitan dengan kualitas dan efisiensi tidur.

Salah satu faktor yang berkaitan dengan kualitas tidur adalah kondisi demografis dan kesehatan pengguna. Beberapa penelitian menunjukkan bahwa kualitas tidur dapat berhubungan dengan faktor seperti usia, jenis kelamin, kondisi ekonomi, tingkat pendidikan, diabetes, hipertensi, dan kebiasaan merokok. Faktor-faktor tersebut menunjukkan bahwa kualitas tidur tidak berdiri sendiri, tetapi dapat dipengaruhi oleh kondisi individu dan status kesehatan secara umum [23]. Dalam penelitian ini, aspek kondisi tubuh pengguna direpresentasikan melalui fitur seperti *bmi*, *avg_hr_day_bpm*, *resting_hr_bpm*, *hrv_rmssd_ms*, dan *spo2_avg_pct*.

Aktivitas fisik juga menjadi salah satu faktor yang dapat berkaitan dengan *sleep efficiency*. Aktivitas harian dalam penelitian ini direpresentasikan melalui fitur *steps*, *workout_minutes*, dan *steps_roll7*. Jumlah langkah dan durasi olahraga dapat menggambarkan tingkat aktivitas fisik pengguna dalam satu hari maupun dalam beberapa hari sebelumnya. Aktivitas fisik yang cukup dapat membantu tubuh mengalami proses pemulihan yang lebih baik saat tidur, meskipun pengaruhnya dapat berbeda pada setiap individu.

Faktor psikologis juga memiliki hubungan dengan kualitas tidur. Kondisi seperti stres, kecemasan, dan depresi dapat berkaitan dengan gangguan tidur atau kualitas tidur yang lebih rendah. Penelitian sebelumnya menunjukkan bahwa gangguan tidur dan faktor psikologis seperti kecemasan, depresi, dan stres dapat berkaitan dengan kondisi kesehatan dan tingkat keparahan penyakit tertentu [24]. Dalam penelitian ini, aspek psikologis dan kondisi emosional pengguna direpresentasikan melalui fitur `stress_score`, `stress_roll7`, dan `mood_encoded`.

Selain faktor psikologis, perilaku pengguna juga dapat memengaruhi efisiensi tidur. Fitur seperti `caffeine_mg`, `alcohol_units`, dan `screen_time_min` digunakan untuk merepresentasikan kebiasaan atau perilaku harian yang berpotensi berkaitan dengan tidur. Konsumsi kafein dapat memengaruhi kemampuan seseorang untuk tidur karena memiliki efek stimulan. Konsumsi alkohol juga dapat memengaruhi kualitas tidur, sedangkan durasi penggunaan layar dapat berkaitan dengan kebiasaan sebelum tidur yang dapat memengaruhi kesiapan tubuh untuk beristirahat.

Kondisi fisiologis pengguna juga menjadi faktor penting dalam prediksi *sleep efficiency*. Fitur `avg_hr_day_bpm` dan `resting_hr_bpm` digunakan untuk menggambarkan kondisi detak jantung pengguna, sedangkan `hrv_rmssd_ms` merepresentasikan variabilitas detak jantung yang berkaitan dengan respons tubuh dan pemulihan fisiologis. Fitur `spo2_avg_pct` digunakan untuk menggambarkan rata-rata kadar oksigen dalam darah. Kondisi fisiologis tersebut dapat memberikan gambaran mengenai keadaan tubuh pengguna yang berpotensi berkaitan dengan kualitas tidur.

Riwayat tidur juga menjadi faktor penting karena data yang digunakan dalam penelitian ini berbentuk *time series*. Fitur `sleep_lag1`, `sleep_lag2`, dan `sleep_roll7` digunakan untuk merepresentasikan pola *sleep efficiency* pada hari-hari sebelumnya. Penggunaan fitur riwayat tidur bertujuan agar model tidak hanya mempelajari kondisi pengguna pada satu hari tertentu, tetapi juga mempertimbangkan pola historis tidur pengguna.

Berdasarkan faktor-faktor tersebut, fitur yang digunakan dalam penelitian ini dipilih untuk merepresentasikan beberapa aspek utama yang berkaitan dengan *sleep efficiency*, yaitu aktivitas harian, kondisi fisiologis, perilaku pengguna, kondisi psikologis, suasana hati, dan riwayat tidur. Kombinasi fitur tersebut diharapkan dapat membantu model dalam memprediksi nilai *sleep efficiency* secara lebih baik berdasarkan data *wearable* berbasis *time series*.

2.3 Framework/Algoritma yang digunakan

Framework atau algoritma yang digunakan dalam penelitian ini berfungsi sebagai dasar dalam proses pembangunan model prediksi *sleep efficiency*. Karena data yang digunakan memiliki karakteristik *time series*, maka algoritma yang dipilih perlu mampu mempelajari pola data yang berurutan dari waktu ke waktu. Selain itu, karena target penelitian berupa nilai numerik kontinu, maka model yang digunakan diarahkan untuk menyelesaikan permasalahan regresi.

Penelitian ini menggunakan pendekatan *Deep Learning* karena data *wearable* memiliki pola yang kompleks dan tidak selalu linear. Aktivitas harian, kondisi fisiologis, tingkat stres, perilaku pengguna, serta riwayat tidur dapat saling berhubungan dan membentuk pola tertentu terhadap nilai *sleep efficiency*. Oleh karena itu, diperlukan model yang mampu menangkap hubungan kompleks tersebut dari data historis pengguna.

Model yang digunakan dalam penelitian ini terdiri dari LSTM, CNN-BiLSTM, dan CNN-BiLSTM dengan *Attention*. LSTM digunakan sebagai model dasar karena memiliki kemampuan dalam memproses data berurutan. CNN-BiLSTM digunakan untuk menggabungkan kemampuan CNN dalam menangkap pola lokal dan BiLSTM dalam mempelajari hubungan temporal dari dua arah. Sementara itu, CNN-BiLSTM dengan *Attention* digunakan untuk melihat apakah mekanisme perhatian dapat membantu model memberikan fokus pada bagian *sequence* yang lebih relevan terhadap prediksi.

Selain algoritma prediksi, penelitian ini juga menggunakan pendekatan *Explainable Artificial Intelligence (XAI)* dengan metode SHAP. Pendekatan ini

digunakan untuk membantu menginterpretasikan hasil prediksi model, sehingga penelitian tidak hanya menilai performa berdasarkan metrik evaluasi, tetapi juga memahami fitur-fitur yang berkontribusi terhadap prediksi *sleep efficiency*.

2.3.1 CRISP-DM

CRISP-DM atau Cross-Industry Standard Process for Data Mining merupakan salah satu kerangka kerja yang banyak digunakan dalam proses data mining dan analisis berbasis data. CRISP-DM digunakan untuk membantu peneliti atau praktisi dalam menyusun tahapan kerja secara sistematis, mulai dari pemahaman masalah, pemahaman data, persiapan data, pemodelan, evaluasi, hingga penerapan hasil. Dalam konteks penelitian berbasis *machine learning*, CRISP-DM berperan sebagai panduan alur kerja agar proses pengolahan data dan pembangunan model dapat dilakukan secara lebih terarah dan sesuai dengan tujuan penelitian [25]. CRISP-DM juga dipandang sebagai salah satu model proses data mining yang paling luas digunakan karena mampu memberikan struktur kerja yang jelas dalam pengembangan proyek berbasis data [26].

CRISP-DM bukan merupakan algoritma atau metode pemodelan, melainkan sebuah framework atau kerangka kerja untuk mengatur tahapan proses data mining. Dengan demikian, CRISP-DM tidak menggantikan algoritma *machine learning* atau *Deep Learning* yang digunakan dalam penelitian, tetapi membantu mengarahkan tahapan penelitian agar lebih sistematis. Hal ini penting karena penelitian berbasis data tidak hanya berfokus pada pembangunan model, tetapi juga melibatkan pemahaman masalah, pemeriksaan kualitas data, pemilihan fitur, proses pembersihan data, serta evaluasi hasil model. Oleh karena itu, CRISP-DM dapat digunakan untuk memastikan bahwa setiap tahapan penelitian memiliki hubungan yang jelas dengan tujuan analisis [26].

Secara umum, CRISP-DM terdiri dari enam tahapan utama, yaitu *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation*, dan *deployment*. Tahap *business understanding* berfokus pada pemahaman masalah dan penentuan tujuan penelitian. Tahap *data understanding* dilakukan untuk memahami karakteristik, struktur, kualitas, dan potensi data yang digunakan. Selanjutnya, tahap *data preparation* mencakup proses pembersihan data, transformasi, pemilihan fitur, dan penyusunan data agar siap digunakan dalam pemodelan. Tahap *modeling* dilakukan dengan membangun dan melatih model berdasarkan data yang telah dipersiapkan. Setelah itu, tahap *evaluation* digunakan untuk menilai performa model dan memastikan apakah hasil yang diperoleh telah sesuai dengan tujuan penelitian. Tahap terakhir adalah *deployment*, yaitu penerapan model atau hasil analisis ke dalam sistem atau lingkungan penggunaan yang lebih nyata [25].

Meskipun memiliki urutan tahapan yang jelas, proses CRISP-DM tidak selalu berjalan secara linear. Hasil dari satu tahapan dapat membuat peneliti perlu kembali ke tahapan sebelumnya untuk melakukan perbaikan. Misalnya, hasil evaluasi model dapat menunjukkan bahwa data perlu diproses ulang, fitur perlu ditinjau kembali, atau konfigurasi model perlu diperbaiki. Dengan sifatnya yang iteratif, CRISP-DM cocok digunakan dalam penelitian berbasis *machine learning* karena proses pengembangan model sering membutuhkan penyesuaian antara data, metode, dan hasil evaluasi [25].

2.3.2 Deep Learning

Deep Learning merupakan bagian dari *machine learning* yang memanfaatkan jaringan saraf tiruan berlapis untuk mengenali pola dalam data. Pendekatan ini banyak digunakan karena mampu mengolah data berukuran besar serta mempelajari hubungan antarvariabel yang kompleks. Pada metode *machine learning*

konvensional, fitur biasanya perlu ditentukan atau diekstraksi terlebih dahulu secara manual. Sementara itu, *Deep Learning* dapat mempelajari pola dan karakteristik data secara otomatis melalui susunan lapisan pemrosesan, sehingga representasi data yang dihasilkan menjadi lebih kaya dan kompleks.

Deep Learning banyak digunakan pada permasalahan yang memiliki pola tidak linear, berdimensi tinggi, serta sulit dianalisis menggunakan pendekatan statistik konvensional. Dalam berbagai bidang, *Deep Learning* telah diterapkan untuk pengenalan citra, pemrosesan bahasa alami, analisis sinyal, deteksi anomali, klasifikasi, maupun prediksi berbasis data *time series*. Kemampuan *Deep Learning* dalam mengenali pola kompleks membuat metode ini banyak digunakan dalam analisis data kesehatan dan data fisiologis, seperti EEG, ECG, serta sinyal lain yang diperoleh dari perangkat sensor [27][28].

Pada data kesehatan, hubungan antarvariabel sering kali tidak sederhana. Kondisi fisiologis seseorang dapat dipengaruhi oleh banyak faktor, seperti aktivitas fisik, detak jantung, stres, pola tidur, konsumsi tertentu, hingga kebiasaan harian. Hubungan antarvariabel tersebut dapat bersifat non-linear dan saling memengaruhi. Oleh karena itu, *Deep Learning* menjadi salah satu pendekatan yang relevan karena mampu mempelajari hubungan kompleks tersebut tanpa harus menentukan pola hubungan secara eksplisit sejak awal.

Deep Learning juga banyak digunakan pada data *time series*. Data *time series* memiliki karakteristik utama berupa urutan waktu, sehingga nilai pada suatu waktu dapat berkaitan dengan nilai pada waktu sebelumnya. Pada kasus seperti peramalan permintaan, prediksi energi, analisis sinyal, maupun prediksi kondisi kesehatan, model perlu memahami perubahan data dari waktu ke waktu. Penelitian sebelumnya menunjukkan bahwa model *Deep Learning* dapat digunakan untuk mempelajari pola temporal, fluktuasi, dan ketergantungan antarwaktu pada data berurutan [29].

Dalam konteks penelitian ini, data *wearable* memiliki karakteristik kompleks karena terdiri dari berbagai fitur aktivitas harian dan fisiologis, seperti jumlah langkah, detak jantung, tingkat stres, aktivitas olahraga, serta indikator tidur. Data tersebut juga berbentuk *time series* karena dicatat secara harian untuk setiap pengguna. Oleh sebab itu, *Deep Learning* digunakan sebagai pendekatan utama untuk membangun model prediksi nilai *sleep efficiency*. Model yang digunakan terdiri dari LSTM, CNN-BiLSTM, dan CNN-BiLSTM dengan *Attention*. Ketiga model tersebut digunakan untuk membandingkan kemampuan masing-masing arsitektur dalam mempelajari pola historis dan temporal pada data *wearable*.

2.3.3 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) merupakan salah satu arsitektur jaringan saraf tiruan yang dikembangkan dari Recurrent Neural Network (RNN). RNN pada dasarnya dirancang untuk memproses data berurutan dengan mempertimbangkan informasi dari waktu sebelumnya. Namun, RNN sederhana memiliki keterbatasan dalam mempelajari hubungan jangka panjang karena adanya masalah *vanishing gradient* dan *exploding gradient*. Permasalahan tersebut dapat menyebabkan informasi dari waktu yang jauh sebelumnya sulit dipertahankan selama proses pembelajaran. LSTM dikembangkan untuk mengatasi keterbatasan tersebut dengan menambahkan mekanisme memori yang mampu mempertahankan informasi penting dalam data sekuensial [9].

LSTM memiliki struktur utama berupa cell state dan beberapa mekanisme gerbang atau gate. Cell state berfungsi sebagai jalur memori yang membawa informasi dari waktu sebelumnya ke waktu berikutnya. Di dalam LSTM terdapat tiga komponen utama, yaitu input gate, forget gate, dan output gate. Input gate berfungsi untuk menentukan informasi baru yang akan dimasukkan ke dalam

memori. Forget gate menentukan informasi lama yang perlu dipertahankan atau dihapus. Sementara itu, output gate menentukan informasi yang akan dikeluarkan sebagai hasil dari proses pembelajaran. Dengan mekanisme tersebut, LSTM dapat memilih informasi yang relevan dan mengabaikan informasi yang kurang penting [9].

Kemampuan LSTM dalam mempertahankan informasi historis membuat arsitektur ini banyak digunakan pada data *time series*. Pada data berurutan, kondisi saat ini sering kali dipengaruhi oleh kondisi sebelumnya. Oleh karena itu, LSTM sesuai digunakan untuk mempelajari hubungan temporal pada data yang memiliki urutan waktu. Dalam konteks prediksi, LSTM dapat mempelajari pola perubahan data dari waktu ke waktu dan menangkap ketergantungan temporal yang mungkin tidak terlihat apabila data hanya dianalisis sebagai data tabular biasa.

LSTM juga banyak digunakan dalam penelitian prediksi berbasis data sensor dan data kesehatan. Pada penelitian prediksi kualitas tidur berbasis *wearable device*, LSTM digunakan untuk memodelkan data *time series* seperti heart rate dan jumlah langkah. Data tersebut dianggap relevan karena aktivitas fisik dan kondisi fisiologis pengguna dapat berkaitan dengan kualitas tidur. Penelitian tersebut juga menunjukkan bahwa pemodelan berbasis *time series* sesuai digunakan karena model perlu menangkap pola temporal dari parameter fisik untuk menghasilkan prediksi yang lebih baik [14].

Selain itu, penelitian berbasis *wearable device* menunjukkan bahwa model LSTM dapat digunakan untuk membandingkan beberapa variasi arsitektur dalam memprediksi kualitas tidur. Hasil penelitian tersebut menunjukkan bahwa model LSTM yang lebih sederhana dapat memberikan performa yang baik, sedangkan model yang lebih kompleks dapat mengalami overfitting apabila jumlah parameter yang dilatih terlalu besar [14]. Hal ini menunjukkan

bahwa kompleksitas model tidak selalu menjamin peningkatan performa, terutama apabila data memiliki karakteristik tertentu.

Dalam penelitian ini, LSTM digunakan sebagai model dasar atau baseline untuk memprediksi nilai *sleep efficiency*. Pemilihan LSTM didasarkan pada kemampuannya dalam memproses data berurutan dan mempelajari hubungan temporal dari data *wearable* harian. Data yang digunakan dalam penelitian ini memiliki bentuk *sequence*, sehingga LSTM digunakan untuk menangkap pola historis dari aktivitas harian, kondisi fisiologis, perilaku pengguna, dan riwayat tidur. Model LSTM juga menjadi pembanding utama untuk mengetahui apakah model yang lebih kompleks, yaitu CNN-BiLSTM dan CNN-BiLSTM dengan *Attention*, mampu memberikan peningkatan performa prediksi dibandingkan model dasar.

2.3.4 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan arsitektur *Deep Learning* yang dirancang untuk mengekstraksi fitur penting dari data. CNN awalnya banyak digunakan pada pengolahan citra karena memiliki kemampuan mengenali pola lokal, seperti garis, bentuk, tekstur, dan objek. Namun, CNN juga dapat digunakan pada data *time series* melalui pendekatan konvolusi satu dimensi atau Conv1D. Pada data berurutan, Conv1D dapat digunakan untuk menangkap pola lokal dalam *sequence*.

CNN bekerja dengan menggunakan filter atau kernel yang bergerak melintasi input data. Proses ini disebut konvolusi. Melalui proses konvolusi, model dapat mengenali pola tertentu yang muncul pada bagian input. Setelah proses konvolusi, biasanya digunakan fungsi aktivasi untuk menambahkan kemampuan non-linear pada model. Selain itu, pooling layer dapat digunakan untuk mereduksi ukuran representasi data dan mempertahankan informasi penting. Hasil dari proses ini kemudian diteruskan ke lapisan berikutnya untuk menghasilkan prediksi.

Dalam data *time series*, CNN dapat digunakan untuk menangkap pola lokal yang terjadi dalam rentang waktu tertentu. Misalnya, perubahan aktivitas fisik, detak jantung, atau tingkat stres dalam beberapa hari dapat menjadi pola lokal yang relevan terhadap prediksi. CNN tidak secara langsung mempelajari hubungan jangka panjang seperti LSTM, tetapi CNN efektif dalam mengekstraksi pola pendek yang muncul dalam *sequence*. Oleh karena itu, CNN sering dikombinasikan dengan model recurrent seperti LSTM atau BiLSTM agar model dapat mempelajari pola lokal sekaligus pola temporal.

Beberapa penelitian menunjukkan bahwa CNN dapat digunakan dalam analisis data sinyal kesehatan seperti ECG dan data sensor lainnya [27]. Selain itu, arsitektur hybrid CNN-LSTM juga banyak digunakan dalam berbagai kasus data *time series* karena CNN mampu menangkap fitur lokal, sedangkan LSTM mampu menangkap hubungan temporal [12][30]. Kombinasi tersebut menunjukkan bahwa CNN dapat berperan sebagai ekstraktor fitur sebelum data diproses oleh model sekuensial.

Dalam penelitian ini, CNN digunakan dalam bentuk Conv1D pada model CNN-BiLSTM dan CNN-BiLSTM dengan *Attention*. CNN digunakan untuk menangkap pola lokal dari *sequence* data *wearable* selama periode tertentu. Setelah pola lokal diekstraksi, hasilnya diteruskan ke BiLSTM agar model dapat mempelajari hubungan temporal secara lebih menyeluruh. Dengan demikian, CNN berperan sebagai tahap awal dalam proses ekstraksi fitur sebelum pemodelan temporal dilakukan.

2.3.5 Bidirectional Long Short-Term Memory (BiLSTM)

Bidirectional Long Short-Term Memory (BiLSTM) merupakan pengembangan dari LSTM yang memproses data sekuensial dari dua arah, yaitu arah maju dan arah mundur. Pada LSTM biasa, data diproses dari awal *sequence* menuju akhir

sequence. Artinya, model hanya mempertimbangkan informasi dari masa lalu menuju masa sekarang. Sementara itu, BiLSTM memproses *sequence* dari dua arah, sehingga model dapat memperoleh informasi dari konteks sebelum dan sesudah suatu titik dalam *sequence* [31].

BiLSTM terdiri dari dua lapisan LSTM yang berjalan secara berlawanan arah. Lapisan pertama memproses data dari awal ke akhir, sedangkan lapisan kedua memproses data dari akhir ke awal. Hasil dari kedua arah tersebut kemudian digabungkan untuk membentuk representasi yang lebih lengkap. Dengan cara ini, BiLSTM dapat menangkap pola temporal secara lebih luas dibandingkan LSTM satu arah [31].

Pada data *time series*, BiLSTM dapat membantu model memahami hubungan antarbagian dalam *sequence*. Meskipun data waktu secara alami bergerak dari masa lalu ke masa depan, dalam proses pelatihan model seluruh *sequence* input sudah tersedia. Oleh karena itu, pemrosesan dua arah dapat membantu model menemukan pola yang mungkin tidak tertangkap apabila hanya menggunakan arah maju. Pendekatan ini sering digunakan pada data sekuensial yang membutuhkan pemahaman konteks secara lebih menyeluruh.

BiLSTM banyak digunakan dalam berbagai penelitian berbasis *time series* dan prediksi karena kemampuannya dalam menangkap ketergantungan temporal dari dua arah. Penelitian sebelumnya menunjukkan bahwa BiLSTM dapat digunakan untuk meningkatkan pemodelan temporal, terutama ketika dikombinasikan dengan arsitektur lain seperti CNN atau model hybrid [32][33]. Selain itu, BiLSTM juga telah digunakan dalam tugas prediksi berbasis regresi pada data historis, sehingga menunjukkan bahwa arsitektur ini tidak hanya relevan untuk klasifikasi, tetapi juga dapat diterapkan pada permasalahan prediksi nilai kontinu [31].

Kombinasi 1D-CNN dan BiLSTM juga dapat memanfaatkan keunggulan CNN dalam ekstraksi fitur lokal serta keunggulan BiLSTM dalam memproses hubungan temporal. Dalam penelitian ini, BiLSTM digunakan pada model CNN-BiLSTM. CNN digunakan terlebih dahulu untuk mengekstraksi pola lokal pada data *wearable*, kemudian BiLSTM digunakan untuk mempelajari hubungan temporal dari hasil ekstraksi tersebut. Penggunaan BiLSTM bertujuan untuk mengetahui apakah pemrosesan *sequence* dari dua arah dapat memberikan performa yang lebih baik dibandingkan LSTM biasa dalam memprediksi nilai *sleep efficiency*.

2.3.6 **Attention dan MultiHeadAttention**

Attention merupakan mekanisme dalam *Deep Learning* yang digunakan untuk membantu model memberikan fokus lebih besar pada bagian input yang dianggap lebih relevan terhadap output. Pada data sekuensial, tidak semua bagian *sequence* memiliki pengaruh yang sama terhadap hasil prediksi. Beberapa titik waktu mungkin lebih penting dibandingkan titik waktu lainnya. Dengan mekanisme *Attention*, model dapat memberikan bobot yang berbeda pada setiap bagian input sesuai tingkat relevansinya.

Secara umum, *Attention* bekerja dengan menghitung hubungan antara elemen-elemen dalam *sequence*. Model akan mempelajari bagian mana dari input yang perlu diberi perhatian lebih besar. Bagian input yang dianggap lebih berkontribusi terhadap prediksi akan memperoleh bobot perhatian yang lebih tinggi. Dengan demikian, *Attention* dapat membantu model dalam memahami bagian *sequence* yang paling berpengaruh terhadap hasil akhir.

MultiHeadAttention merupakan pengembangan dari mekanisme *Attention* yang menggunakan beberapa head perhatian secara paralel. Setiap head dapat mempelajari pola hubungan yang berbeda dari input yang sama. Dengan menggunakan beberapa head,

model dapat menangkap berbagai jenis hubungan dalam *sequence* secara bersamaan. Hal ini membuat *MultiHeadAttention* lebih fleksibel dalam mempelajari representasi data yang kompleks dibandingkan *Attention* tunggal.

Mekanisme *MultiHeadAttention* banyak digunakan dalam arsitektur modern seperti Transformer. Pada berbagai penelitian, *MultiHeadAttention* digunakan untuk membantu model menangkap hubungan yang kompleks dalam data sekuensial dan meningkatkan interpretabilitas terhadap bagian input yang dianggap penting [34]. Selain itu, model berbasis multi-head feature fusion juga menunjukkan bahwa pendekatan multi-head dapat membantu model mempelajari representasi fitur yang lebih beragam dan adaptif [35].

Pada data *wearable*, *Attention* dapat digunakan untuk membantu model mengenali bagian *sequence* yang lebih penting terhadap prediksi *sleep efficiency*. Misalnya, aktivitas fisik, tingkat stres, atau kondisi fisiologis pada beberapa hari tertentu dapat lebih berpengaruh dibandingkan hari lainnya. Dengan *Attention*, model tidak hanya memproses seluruh *sequence* secara merata, tetapi dapat memberikan fokus lebih pada bagian input yang dianggap lebih relevan.

Dalam penelitian ini, *MultiHeadAttention* digunakan pada model CNN-BiLSTM dengan *Attention*. Model ini dibangun untuk melihat apakah penambahan *Attention* setelah proses CNN dan BiLSTM dapat membantu meningkatkan performa prediksi. CNN berfungsi untuk menangkap pola lokal, BiLSTM mempelajari hubungan temporal, sedangkan *MultiHeadAttention* membantu model memberikan bobot perhatian pada bagian *sequence* yang lebih penting. Dengan kombinasi tersebut, model diharapkan dapat mempelajari pola data *wearable* secara lebih mendalam dalam memprediksi nilai *sleep efficiency*.

2.3.7 Perbandingan Algoritma yang digunakan

Penelitian ini menggunakan tiga model *Deep Learning*, yaitu LSTM, CNN-BiLSTM, dan CNN-BiLSTM dengan *Attention*. Ketiga model tersebut dipilih karena memiliki kemampuan dalam mempelajari data berurutan atau *time series*, tetapi memiliki karakteristik arsitektur yang berbeda. Perbandingan model dilakukan untuk mengetahui model yang paling sesuai dalam memprediksi nilai *sleep_efficiency* berdasarkan data *wearable* harian yang disusun dalam bentuk *sequence*.

LSTM digunakan sebagai model dasar karena mampu mempelajari ketergantungan temporal pada data berurutan. Model ini memiliki mekanisme *gate* yang dapat membantu mengatur informasi mana yang perlu disimpan atau dilupakan selama proses pembelajaran. Dengan kemampuan tersebut, LSTM sesuai digunakan untuk data *time series* karena model dapat mempelajari pola historis dari data beberapa hari sebelumnya.

CNN-BiLSTM merupakan model gabungan yang memanfaatkan CNN dan BiLSTM. CNN digunakan untuk menangkap pola lokal pada data *sequence*, sedangkan BiLSTM digunakan untuk mempelajari pola temporal dari dua arah dalam rentang *window* historis. Kombinasi ini diharapkan dapat membantu model memahami pola lokal antarfitur sekaligus pola urutan waktu dalam data *wearable*. Namun, model ini memiliki arsitektur yang lebih kompleks dibandingkan LSTM, sehingga jumlah parameter yang dipelajari juga lebih besar.

CNN-BiLSTM dengan *Attention* merupakan pengembangan dari CNN-BiLSTM dengan tambahan mekanisme *Attention*. Mekanisme *Attention* digunakan untuk menguji apakah model dapat memberikan fokus yang lebih besar pada bagian *sequence* yang dianggap lebih relevan terhadap hasil prediksi. Dalam penelitian ini, *Attention* tidak digunakan dengan asumsi bahwa model pasti menghasilkan performa lebih baik, tetapi digunakan untuk menguji

apakah mekanisme pembobotan pada bagian tertentu dari *sequence* dapat membantu prediksi *sleep_efficiency*.

Perbandingan karakteristik ketiga model ditunjukkan pada Tabel 2.2.

Tabel 2.2 Perbandingan Model

Model	Karakteristik Utama	Kelebihan	Keterbatasan	Alasan Digunakan dalam Penelitian
LSTM	Model recurrent neural network yang dirancang untuk mempelajari data berurutan dengan mekanisme gate.	Mampu menangkap pola historis pada data <i>time series</i> dan lebih sederhana dibandingkan model hybrid.	Tidak secara khusus mengekstraksi pola lokal seperti CNN dan hanya membaca urutan dalam satu arah.	Digunakan sebagai model dasar untuk memprediksi <i>sleep_efficiency</i> berdasarkan pola historis data <i>wearable</i> .
CNN-BiLSTM	Model hybrid yang menggabungkan CNN dan BiLSTM. CNN menangkap pola lokal, sedangkan BiLSTM mempelajari pola temporal dua arah dalam window historis.	Dapat mempelajari pola lokal antarfitur dan pola temporal dalam <i>sequence</i> . BiLSTM juga dapat menangkap informasi dari dua arah dalam rentang input historis.	Arsitektur lebih kompleks, jumlah parameter lebih besar, dan berpotensi lebih sensitif terhadap <i>noise</i> atau variasi antar pengguna.	Digunakan untuk menguji apakah kombinasi ekstraksi pola lokal dan pembelajaran temporal dua arah dapat meningkatkan performa prediksi.
CNN-BiLSTM dengan <i>Attention</i>	Model CNN-BiLSTM yang ditambahkan mekanisme <i>Attention</i> untuk memberikan bobot pada bagian <i>sequence</i> tertentu.	Dapat membantu model memberikan fokus lebih besar pada bagian input yang dianggap lebih relevan terhadap prediksi.	Penambahan <i>Attention</i> membuat model lebih kompleks dan tidak selalu menghasilkan performa lebih baik, terutama jika data relatif terbatas, bervariasi,	Digunakan untuk menguji apakah mekanisme fokus pada bagian <i>sequence</i> tertentu dapat membantu prediksi <i>sleep_efficiency</i> .

			atau mengandung <i>noise</i> .	
--	--	--	--------------------------------	--

Berdasarkan perbandingan tersebut, LSTM berperan sebagai model dasar untuk mempelajari pola historis pada data *wearable time series*. CNN-BiLSTM digunakan untuk melihat apakah penambahan CNN dan BiLSTM dapat meningkatkan kemampuan model dalam menangkap pola lokal dan temporal. Sementara itu, CNN-BiLSTM dengan *Attention* digunakan untuk menguji apakah mekanisme pembobotan pada bagian tertentu dari *sequence* dapat membantu model menghasilkan prediksi yang lebih baik.

Dengan demikian, perbandingan ketiga model ini tidak hanya bertujuan untuk mencari model dengan performa terbaik, tetapi juga untuk melihat apakah peningkatan kompleksitas arsitektur memberikan dampak positif terhadap prediksi *sleep_efficiency*. Hal ini penting karena model yang lebih kompleks tidak selalu lebih unggul, terutama pada data *wearable* harian yang memiliki variasi antar pengguna dan kemungkinan mengandung *noise*.

2.3.8 ***Explainable Artificial Intelligence (XAI)***

Explainable Artificial Intelligence (XAI) digunakan untuk membuat hasil prediksi model kecerdasan buatan menjadi lebih transparan dan dapat dijelaskan. Model *machine learning* dan *Deep Learning* sering kali menghasilkan prediksi melalui proses internal yang kompleks, sehingga tidak mudah diketahui bagaimana suatu output terbentuk. Kondisi ini membuat model tersebut kerap dipandang sebagai *black box*. Dalam konteks kesehatan, transparansi model menjadi penting karena pengguna tidak hanya membutuhkan nilai prediksi, tetapi juga perlu mengetahui alasan atau faktor yang mendukung munculnya prediksi tersebut [36].

Dalam bidang kesehatan, interpretabilitas model menjadi aspek penting karena hasil prediksi dapat berkaitan dengan kondisi fisiologis atau kualitas hidup seseorang. Model yang memiliki

performa baik tetapi sulit dijelaskan dapat menimbulkan keterbatasan dalam proses analisis, terutama ketika peneliti ingin mengetahui fitur apa saja yang berkontribusi terhadap hasil prediksi. Oleh karena itu, *XAI* digunakan untuk membantu menjelaskan bagaimana model menggunakan fitur input dalam menghasilkan output [36].

XAI dapat digunakan untuk memberikan transparansi terhadap model yang kompleks. Pendekatan ini dapat membantu mengidentifikasi fitur yang paling berpengaruh, memahami arah kontribusi fitur, serta memberikan penjelasan yang lebih masuk akal terhadap hasil prediksi. Dengan adanya *XAI*, hasil model tidak hanya dinilai berdasarkan metrik evaluasi, tetapi juga berdasarkan kemampuan model untuk memberikan interpretasi terhadap hubungan antara fitur dan target [37].

Secara umum, metode *XAI* dapat digunakan untuk menjelaskan model secara global maupun lokal. Penjelasan global digunakan untuk melihat fitur yang berpengaruh terhadap model secara keseluruhan, sedangkan penjelasan lokal digunakan untuk memahami kontribusi fitur pada prediksi tertentu. Kedua bentuk penjelasan tersebut dapat membantu peneliti memahami perilaku model dengan lebih baik [37].

Dalam penelitian ini, *XAI* digunakan untuk membantu menginterpretasikan hasil prediksi model *Deep Learning* terhadap nilai *sleep efficiency*. Penggunaan *XAI* bertujuan untuk mengetahui fitur *wearable* mana yang paling berkontribusi terhadap prediksi, sehingga hasil penelitian tidak hanya menampilkan performa model, tetapi juga memberikan pemahaman mengenai faktor-faktor yang memengaruhi prediksi *sleep efficiency*.

2.3.9 SHAP

Selain mengevaluasi performa model menggunakan metrik regresi, penelitian ini juga menerapkan pendekatan *Explainable*

Artificial Intelligence (XAI) untuk membantu menjelaskan hasil prediksi model. XAI digunakan karena model *Deep Learning* seperti LSTM, CNN-BiLSTM, dan CNN-BiLSTM dengan *Attention* memiliki struktur yang kompleks, sehingga proses pengambilan keputusan model sulit dijelaskan secara langsung hanya berdasarkan nilai MSE, RMSE, MAE, dan R^2 . Metrik evaluasi hanya menunjukkan tingkat kesalahan dan kemampuan prediksi model, tetapi belum menjelaskan fitur apa saja yang berkontribusi terhadap hasil prediksi.

Metode XAI yang digunakan dalam penelitian ini adalah SHAP (SHapley Additive exPlanations). SHAP dipilih karena mampu memberikan nilai kontribusi fitur secara kuantitatif terhadap output prediksi model [37]. Dengan pendekatan ini, setiap fitur dapat dianalisis berdasarkan besar kontribusinya terhadap prediksi *sleep_efficiency*. Hal tersebut sesuai dengan kebutuhan penelitian ini, karena selain menentukan model terbaik, penelitian juga bertujuan untuk mengetahui fitur-fitur yang paling berkontribusi terhadap prediksi efisiensi tidur.

Pemilihan SHAP juga dilakukan karena metode ini dapat digunakan untuk interpretasi secara global maupun lokal [37]. Interpretasi global digunakan untuk mengetahui fitur mana yang secara umum paling berkontribusi terhadap prediksi model melalui *variable importance*. Sementara itu, interpretasi lokal dapat digunakan untuk melihat kontribusi fitur pada hasil prediksi tertentu. Dalam penelitian ini, interpretasi lebih difokuskan pada interpretasi global agar dapat diketahui fitur-fitur utama yang paling banyak digunakan model dalam memprediksi *sleep_efficiency*.

Dibandingkan dengan metode interpretasi yang hanya berfokus pada prediksi tertentu atau hanya mengukur perubahan performa model ketika suatu fitur dihilangkan, SHAP lebih sesuai digunakan dalam penelitian ini karena dapat menghasilkan nilai kontribusi fitur yang lebih terstruktur dan mudah dirangkum sebagai

variable importance. Hal ini penting karena input model dalam penelitian ini terdiri dari 18 fitur yang dibentuk dalam *sequence* 14 hari, sehingga diperlukan metode interpretasi yang dapat membantu merangkum kontribusi fitur terhadap prediksi model secara keseluruhan.

SHAP juga relevan dengan karakteristik data penelitian yang berupa data *wearable time series*. Fitur yang digunakan dalam model tidak hanya mencakup data aktivitas dan fisiologis harian, tetapi juga fitur historis hasil *feature engineering*, seperti *sleep_lag1*, *sleep_lag2*, *sleep_roll7*, *steps_roll7*, dan *stress_roll7*. Dengan menggunakan SHAP, kontribusi fitur harian dan fitur historis tersebut dapat dianalisis untuk melihat fitur mana yang paling berperan dalam hasil prediksi model.

Namun, interpretasi SHAP dalam penelitian ini tetap dibatasi sebagai penjelasan kontribusi fitur terhadap prediksi model, bukan sebagai pembuktian hubungan sebab-akibat secara klinis. Fitur dengan nilai SHAP tinggi tidak dapat langsung disimpulkan sebagai penyebab naik atau turunnya *sleep_efficiency*. Interpretasi yang digunakan adalah bahwa fitur tersebut memiliki kontribusi besar dalam proses prediksi yang dilakukan oleh model.

2.4 Metrik Evaluasi Regresi

Metrik evaluasi digunakan untuk mengukur seberapa baik model dalam menghasilkan prediksi terhadap nilai aktual. Karena penelitian ini menggunakan pendekatan regresi, maka metrik evaluasi yang digunakan berbeda dengan metrik klasifikasi. Pada penelitian regresi, model tidak dinilai berdasarkan benar atau salahnya kelas, tetapi berdasarkan seberapa dekat nilai prediksi dengan nilai aktual.

Dalam penelitian ini, variabel target yang diprediksi adalah *sleep_efficiency*, yaitu nilai numerik kontinu. Oleh karena itu, evaluasi model dilakukan menggunakan metrik Mean Squared Error (MSE), Root Mean

Squared Error (RMSE), Mean Absolute Error (MAE), dan koefisien determinasi (R^2). Keempat metrik tersebut digunakan untuk memberikan gambaran yang lebih menyeluruh mengenai tingkat kesalahan prediksi dan kemampuan model dalam menjelaskan variasi data target.

MSE, RMSE, dan MAE digunakan untuk melihat besar kesalahan prediksi model. Kinerja model dianggap lebih baik apabila nilai MSE, RMSE, dan MAE yang dihasilkan semakin kecil, karena hal tersebut menunjukkan bahwa hasil prediksi semakin mendekati nilai aktual. Adapun R^2 digunakan untuk menilai kemampuan model dalam menjelaskan perubahan atau variasi pada variabel *sleep_efficiency*. Nilai R^2 yang lebih tinggi menunjukkan bahwa model semakin mampu menangkap pola data dengan baik.

2.4.1 Mean Squared Error (MSE)

Mean Squared Error (MSE) merupakan metrik evaluasi regresi yang menghitung rata-rata dari kuadrat selisih antara nilai aktual dan nilai prediksi. MSE memberikan penalti yang lebih besar terhadap kesalahan prediksi yang besar karena setiap error dikuadratkan. Dengan demikian, MSE sensitif terhadap prediksi yang memiliki error tinggi.

Rumus MSE adalah sebagai berikut:

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

Rumus 2.1 Rumus MSE

Keterangan:

y_i = nilai aktual

$\hat{y}_i$ = nilai prediksi

n = jumlah data

Dalam penelitian ini, MSE digunakan untuk mengetahui rata-rata kesalahan kuadrat antara nilai *sleep_efficiency* aktual dan nilai *sleep_efficiency* hasil prediksi. Nilai MSE yang lebih kecil

menunjukkan bahwa model menghasilkan prediksi yang lebih dekat dengan nilai aktual.

2.4.2 Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) merupakan akar kuadrat dari MSE. RMSE digunakan untuk mengukur besar kesalahan prediksi dalam skala yang sama dengan variabel target. Karena target dalam penelitian ini adalah *sleep_efficiency*, maka nilai RMSE dapat diinterpretasikan sebagai rata-rata besar error prediksi dalam satuan nilai *sleep_efficiency*.

Rumus RMSE adalah sebagai berikut:

$$\text{RMSE} = \sqrt{\text{MSE}}$$

atau dapat dituliskan sebagai:

$$\text{RMSE} = \sqrt{\frac{\sum (y_i - \hat{y}_i)^2}{n}}$$

Rumus 2.2 Rumus RMSE

Keterangan:

y_i = nilai aktual

$\hat{y}_i$ = nilai prediksi

n = jumlah data

RMSE berguna untuk memberikan gambaran besar kesalahan prediksi secara lebih mudah dipahami karena berada pada skala yang sama dengan target. Semakin kecil nilai RMSE, maka semakin baik performa model dalam memprediksi nilai *sleep_efficiency*.

2.4.3 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) merupakan metrik evaluasi regresi yang menghitung rata-rata nilai absolut dari selisih antara nilai aktual dan nilai prediksi. Berbeda dengan MSE yang mengkuadratkan error, MAE menghitung besar error secara langsung tanpa memperhatikan arah kesalahan. Artinya, prediksi yang lebih

tinggi atau lebih rendah dari nilai aktual tetap dihitung sebagai kesalahan absolut.

Rumus MAE adalah sebagai berikut:

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

Rumus 2.3 Rumus MAE

Keterangan:

y_i = nilai aktual

$\hat{y}_i$ = nilai prediksi

n = jumlah data

Dalam penelitian ini, MAE digunakan untuk mengetahui rata-rata kesalahan absolut prediksi nilai *sleep_efficiency*. Nilai MAE yang lebih kecil menunjukkan bahwa prediksi model semakin mendekati nilai aktual. MAE juga lebih mudah dipahami karena menunjukkan rata-rata error secara langsung pada skala target.

2.4.4 Koefisien Determinasi (R^2)

Koefisien determinasi atau R^2 merupakan metrik evaluasi regresi yang digunakan untuk mengukur seberapa besar variasi nilai target dapat dijelaskan oleh model. Nilai R^2 umumnya berada pada rentang 0 sampai 1. Semakin mendekati 1, maka semakin besar kemampuan model dalam menjelaskan variasi data target. Sebaliknya, nilai R^2 yang rendah menunjukkan bahwa model belum mampu menjelaskan pola data dengan baik.

Rumus R^2 adalah sebagai berikut:

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

Rumus 2.4 Rumus R^2

Keterangan:

y_i = nilai aktual

$\hat{y}_i$ = nilai prediksi

$\bar{y}$ = rata-rata nilai aktual

n = jumlah data

Dalam penelitian ini, R^2 digunakan sebagai metrik utama untuk membandingkan performa model LSTM, CNN-BiLSTM, dan CNN-BiLSTM dengan *Attention*. Hal ini karena R^2 dapat menunjukkan seberapa baik model menjelaskan variasi nilai *sleep_efficiency*. Nilai R^2 yang lebih tinggi menunjukkan bahwa model memiliki kemampuan prediksi yang lebih baik.

2.4.5 Pemilihan Metrik Evaluasi

Pemilihan MSE, RMSE, MAE, dan R^2 dilakukan karena penelitian ini merupakan permasalahan regresi. MSE digunakan untuk melihat besar kesalahan kuadrat, RMSE digunakan untuk mengukur error dalam skala yang sama dengan target, MAE digunakan untuk melihat rata-rata kesalahan absolut, sedangkan R^2 digunakan untuk mengetahui kemampuan model dalam menjelaskan variasi nilai target.

Keempat metrik evaluasi tersebut digunakan agar penilaian performa model tidak hanya berfokus pada besar kecilnya kesalahan prediksi, tetapi juga pada kemampuan model dalam memahami variasi data. Dalam penelitian ini, model dianggap lebih baik apabila menghasilkan nilai MSE, RMSE, dan MAE yang rendah, serta nilai R^2 yang tinggi.

2.5 Tools/software yang digunakan

2.5.1 Python

Python banyak digunakan dalam penelitian berbasis data karena memiliki dukungan pustaka yang luas untuk analisis data, data science, *machine learning*, dan *Deep Learning*. Python memiliki sintaks yang relatif sederhana, fleksibel, serta didukung oleh ekosistem pustaka (library) yang luas. Dukungan pustaka tersebut membuat Python banyak digunakan dalam proses pengolahan data, pemodelan prediktif, visualisasi, dan evaluasi model. Dalam penelitian berbasis data, Python dapat digunakan untuk membangun alur kerja komputasional yang terstruktur, mulai

dari pembacaan data, *preprocessing*, pemodelan, hingga analisis hasil [38].

Dalam penelitian ini, Python digunakan sebagai bahasa pemrograman utama untuk menjalankan seluruh tahapan analisis. Tahapan tersebut meliputi pembacaan *dataset*, eksplorasi data, pembersihan data, transformasi fitur, pembentukan data *time series*, pelatihan model *Deep Learning*, evaluasi performa model, visualisasi hasil, serta interpretasi model. Penggunaan Python memungkinkan seluruh proses penelitian dilakukan dalam satu alur kerja yang saling terintegrasi.

Pada tahap pengolahan data, Python digunakan untuk membaca *dataset* dalam format CSV dan mengubahnya menjadi struktur data yang dapat dianalisis. Library pandas digunakan untuk memproses data berbentuk tabel, seperti melihat struktur *dataset*, memeriksa *missing value*, mengurutkan data berdasarkan *user_id* dan *date*, serta membentuk fitur tambahan seperti *lag feature* dan *rolling average*. Selain itu, library NumPy digunakan untuk melakukan operasi numerik dan membentuk array yang dibutuhkan dalam pemodelan *Deep Learning* berbasis *time series*.

Python juga digunakan dalam proses visualisasi data. Library Matplotlib dan Seaborn digunakan untuk menampilkan visualisasi seperti distribusi target *sleep_efficiency*, matriks korelasi antarfitur, grafik *actual vs predicted*, *training curve*, *error distribution*, serta grafik interpretasi hasil model. Visualisasi tersebut membantu proses analisis karena pola data dan hasil model dapat diamati secara lebih jelas.

Pada tahap *data preparation*, Python digunakan bersama library Scikit-learn untuk melakukan proses *scaling* dan evaluasi model. Dalam penelitian ini, RobustScaler digunakan untuk menyesuaikan skala fitur input dan target karena data *wearable* memiliki kemungkinan nilai ekstrem pada beberapa variabel. Selain itu, Scikit-learn digunakan untuk menghitung metrik evaluasi

regresi, yaitu Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), dan koefisien determinasi (R^2). Penggunaan library seperti pandas dan scikit-learn dalam Python juga didukung oleh penelitian sebelumnya yang menunjukkan bahwa Python-based *machine learning* tools dapat digunakan untuk membangun workflow pemodelan, validasi, dan evaluasi model secara terstruktur [38].

Untuk proses pemodelan, penelitian ini menggunakan TensorFlow dan Keras. TensorFlow merupakan framework yang banyak digunakan dalam pengembangan model *Deep Learning*, sedangkan Keras menyediakan API yang lebih sederhana untuk membangun arsitektur jaringan saraf. Dalam penelitian ini, TensorFlow/Keras digunakan untuk membangun model LSTM, CNN-BiLSTM, dan CNN-BiLSTM dengan *Attention*. Beberapa layer yang digunakan antara lain LSTM, Bidirectional LSTM, Conv1D, MaxPooling1D, *Dropout*, Dense, MultiHead*Attention*, Add, LayerNormalization, dan GlobalAveragePooling1D.

Selain digunakan untuk membangun model, Python juga digunakan untuk mengatur proses pelatihan model. Callback seperti EarlyStopping dan ReduceLROnPlateau digunakan untuk membantu proses *training* berjalan lebih stabil. EarlyStopping digunakan untuk menghentikan proses pelatihan ketika nilai *validation loss* tidak mengalami perbaikan, sedangkan ReduceLROnPlateau digunakan untuk menurunkan *learning rate* ketika performa model mulai stagnan. Dengan bantuan callback tersebut, proses pelatihan model dapat dikontrol agar tidak berjalan terlalu lama dan dapat mengurangi risiko overfitting.

Pada tahap interpretasi model, Python digunakan bersama library SHAP. SHAP digunakan untuk menganalisis kontribusi fitur terhadap hasil prediksi model. Dengan SHAP, penelitian ini dapat mengetahui fitur-fitur yang paling berpengaruh terhadap prediksi nilai *sleep efficiency*. Hal ini penting karena model *Deep Learning*

umumnya sulit diinterpretasikan secara langsung. Oleh karena itu, penggunaan SHAP membantu memberikan penjelasan yang lebih transparan terhadap hasil prediksi model.

Secara keseluruhan, Python dipilih karena mampu mendukung seluruh proses penelitian, mulai dari pengolahan data hingga interpretasi model. Ketersediaan library yang lengkap membuat Python sesuai digunakan dalam penelitian prediksi *sleep efficiency* berbasis data *wearable time series*. Dengan menggunakan Python, proses penelitian dapat dilakukan secara sistematis, terstruktur, dan dapat direplikasi kembali.

2.5.2 Jupyter Notebook

Jupyter Notebook merupakan lingkungan kerja interaktif yang banyak digunakan dalam bidang analisis data, *machine learning*, dan penelitian komputasional. Jupyter Notebook memungkinkan pengguna untuk menulis kode, menjalankan kode, menampilkan output, membuat visualisasi, serta menambahkan penjelasan dalam bentuk teks pada satu dokumen yang sama. Lingkungan ini mendukung penggabungan kode, data, output, visualisasi, dan narasi teks sehingga sesuai digunakan dalam proses analisis data dan penelitian ilmiah [39].

Dalam penelitian ini, Jupyter Notebook digunakan sebagai media utama untuk menjalankan proses analisis dan pemodelan. Seluruh tahapan penelitian, mulai dari eksplorasi data, *preprocessing*, *feature engineering*, pembentukan *sequence*, pelatihan model, evaluasi, visualisasi, hingga interpretasi SHAP dilakukan dalam lingkungan Jupyter Notebook. Dengan menggunakan notebook, setiap langkah penelitian dapat dijalankan secara berurutan dan hasilnya dapat langsung diamati.

Jupyter Notebook memiliki struktur berbasis cell, yaitu bagian-bagian kode dan teks yang dapat dijalankan atau ditulis

secara terpisah. Struktur ini memudahkan peneliti dalam melakukan eksperimen secara bertahap karena setiap bagian kode dapat dieksekusi secara independen. Jurnal terkait Jupyter Notebook menjelaskan bahwa notebook memiliki code cell untuk menjalankan kode, markdown cell untuk menuliskan dokumentasi atau narasi, serta output cell untuk menampilkan hasil eksekusi seperti teks, tabel, error, maupun visualisasi [39]. Karakteristik ini sesuai dengan kebutuhan penelitian yang memerlukan dokumentasi proses analisis dan tampilan hasil secara langsung.

Jupyter Notebook memudahkan proses eksplorasi *dataset* karena setiap kode dapat dijalankan per bagian atau per cell. Hal ini memungkinkan peneliti untuk memeriksa output dari setiap tahap secara langsung, seperti melihat bentuk *dataset*, mengecek *missing value*, menampilkan statistik deskriptif, dan membuat visualisasi distribusi data. Jika terdapat kesalahan pada tahap awal, peneliti dapat memperbaikinya tanpa harus menjalankan seluruh program dari awal.

Selain itu, Jupyter Notebook juga mendukung visualisasi secara langsung. Grafik seperti distribusi *sleep_efficiency*, matriks korelasi, *training curve*, actual vs predicted, scatter plot, error distribution, dan grafik SHAP dapat ditampilkan langsung setelah kode dijalankan. Kemampuan ini membantu peneliti dalam memahami hasil analisis secara visual dan memudahkan pemilihan gambar yang akan dimasukkan ke dalam laporan penelitian.

Dalam proses pemodelan, Jupyter Notebook memudahkan eksperimen model karena setiap model dapat dibangun, dilatih, dan dievaluasi pada bagian kode yang berbeda. Peneliti dapat membandingkan hasil dari model LSTM, CNN-BiLSTM, dan CNN-BiLSTM dengan *Attention* secara langsung. Jika terjadi perubahan pada fitur, parameter, atau arsitektur model, peneliti dapat menjalankan ulang bagian tertentu tanpa harus menjalankan seluruh kode dari awal.

Jupyter Notebook juga membantu dalam dokumentasi proses penelitian. Setiap tahapan dapat diberi penjelasan menggunakan markdown, sehingga alur kerja menjadi lebih mudah dipahami. Peneliti dapat menuliskan keterangan mengenai tujuan setiap proses, alasan pemilihan fitur, hasil evaluasi, dan interpretasi visualisasi secara langsung di dalam notebook. Hal ini membuat notebook tidak hanya berfungsi sebagai tempat menjalankan kode, tetapi juga sebagai catatan eksperimen yang terstruktur.

Pada penelitian ini, Jupyter Notebook juga mendukung proses reproduksibilitas. Kode yang digunakan untuk membaca data, melakukan *preprocessing*, membentuk *sequence*, melatih model, menghitung metrik evaluasi, dan menghasilkan visualisasi dapat disimpan dalam satu file notebook. Dengan demikian, proses penelitian dapat ditelusuri kembali dan dijalankan ulang apabila diperlukan. Hal ini sejalan dengan karakteristik Jupyter Notebook yang memungkinkan integrasi kode, output, visualisasi, dan narasi dalam satu dokumen, sehingga mendukung proses komunikasi dan reproduksi hasil penelitian [39].

Keunggulan lain dari Jupyter Notebook adalah kemampuannya untuk mengintegrasikan berbagai library Python dalam satu lingkungan kerja. Library seperti Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, TensorFlow/Keras, dan SHAP dapat digunakan secara bersamaan. Hal ini membuat proses penelitian lebih efisien karena seluruh kebutuhan analisis dan pemodelan dapat dilakukan dalam satu platform.

Secara keseluruhan, Jupyter Notebook digunakan dalam penelitian ini karena mendukung proses analisis data secara interaktif, fleksibel, dan terdokumentasi. Lingkungan ini memudahkan peneliti dalam melakukan eksperimen model *Deep Learning* berbasis *time series*, mengevaluasi performa model, serta menampilkan visualisasi hasil penelitian. Dengan demikian, Jupyter

Notebook menjadi alat yang penting dalam pelaksanaan penelitian prediksi *sleep efficiency* berbasis data *wearable*.



UMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA