“**PREDIKSI SUHU MENGGUNAKAN LONG SHORT-TERM MEMORY (LSTM) UNTUK PERAMALAN CUACA JANGKA PENDEK”**

**Hilmi Nurpadilah**

**Prodi Teknik Informatika - Fakultas Teknologi Informasi**

**Universitas Bale Bandung, Jl. R.A.A Wiranata Kusumah No.7, Baleendah,**

**Kec. Baleendah, Kabupaten Bandung, Jawa Barat 40375**

[hilminurpadilah8@gmail.com](mailto:Hilminurpadilah8@gmail.com) [rektorat@unibba.ac.id](mailto:rektorat@unibba.ac.id)

**ABSTRAK**

**Prediksi suhu menjadi komponen penting dalam peramalan cuaca yang bermanfaat bagi berbagai sektor seperti pertanian, energi, dan aktivitas harian. Penelitian ini menerapkan model **Long Short-Term Memory (LSTM)** untuk memprediksi suhu udara berdasarkan data time series dengan interval 10 menit. Dataset yang digunakan mencakup **52.696 data** selama periode **Januari 2020 hingga Januari 2021**, terdiri dari **8 fitur utama** meliputi suhu, kelembaban, tekanan udara, kecepatan angin, curah hujan, dan radiasi. Model **LSTM multivariate** dengan dua lapisan tersembunyi (100 dan 50 unit) dilatih menggunakan optimizer **Adam** dan fungsi loss **MSE**. Hasil evaluasi menunjukkan performa model yang baik dengan nilai **RMSE, MAE,** dan **R²** yang memuaskan. Model ini kemudian diimplementasikan ke dalam aplikasi web berbasis Flask yang mampu memprediksi suhu jangka pendek secara real-time. Penelitian ini menunjukkan bahwa LSTM efektif digunakan untuk peramalan cuaca jangka pendek dengan akurasi tinggi dan potensi pengembangan pada sistem prediksi berbasis data real-time.**

****Kata kunci**: LSTM, Time Series, Prediksi Suhu, Deep Learning, FlaskKata kunci:** LSTM, Time Series, Prediksi Suhu, Deep Learning, Flask

***ABSTRACT***

*Temperature prediction is an essential component of weather forecasting that plays a significant role in various sectors such as agriculture, energy management, and daily activities. This study applies a Long Short-Term Memory (LSTM) model to predict air temperature based on time series weather data recorded at 10-minute intervals. The dataset contains 52,696 records collected between January 2020 and January 2021, consisting of eight main features including temperature, humidity, air pressure, wind speed, rainfall, and shortwave radiation. A multivariate LSTM model with two hidden layers (100 and 50 units) was trained using the Adam optimizer and Mean Squared Error (MSE) as the loss function. The evaluation results indicate good model performance, achieving satisfactory RMSE, MAE, and R² values. The trained model was deployed into a Flask-based web application capable of performing short-term temperature predictions in real time. This research demonstrates that LSTM is an effective and practical approach for short-term weather forecasting, offering high accuracy and potential for integration into real-time data-driven prediction systems.*

***Keywords****: LSTM, Time Series, Temperature Prediction, Deep Learning, Flask.*

**PENDAHULUAN**

* 1. **Latar Belakang**

Prediksi suhu udara merupakan salah satu aspek penting dalam sistem peramalan cuaca karena berpengaruh terhadap berbagai bidang kehidupan, seperti pertanian, transportasi, energi, hingga perencanaan aktivitas harian. Informasi suhu yang akurat membantu masyarakat dan industri dalam pengambilan keputusan, misalnya penjadwalan kegiatan outdoor, manajemen sistem pendingin ruangan (HVAC), serta efisiensi penggunaan energi.

Metode tradisional seperti *Numerical Weather Prediction (NWP)* sering membutuhkan sumber daya komputasi besar dan tidak efisien untuk peramalan jangka pendek. Dengan kemajuan teknologi machine learning dan deep learning, metode berbasis data kini menjadi alternatif yang efektif dan efisien untuk memprediksi data deret waktu (time series).

Salah satu algoritma deep learning yang paling banyak digunakan untuk peramalan time series adalah **Long Short-Term Memory (LSTM).** LSTM merupakan varian dari *Recurrent Neural Network (RNN)* yang mampu mempelajari pola temporal jangka panjang tanpa mengalami permasalahan vanishing gradient. Dengan kemampuannya dalam mengingat konteks historis dan mengenali pola non-linear, LSTM menjadi pilihan yang tepat untuk prediksi suhu berdasarkan data cuaca yang bersifat kontinu dan dinamis.

Dalam penelitian ini, model LSTM dikembangkan untuk memprediksi suhu udara menggunakan data cuaca dengan interval 10 menit selama satu tahun pengamatan. Model kemudian diimplementasikan ke dalam **aplikasi web berbasis Flask**, sehingga pengguna dapat melakukan prediksi suhu jangka pendek secara interaktif dan real-time.

**1.2 Rumusan Masalah**

Berdasarkan latar belakang di atas, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana membangun model LSTM untuk memprediksi suhu udara berdasarkan data cuaca historis?
2. Fitur cuaca apa saja yang paling berpengaruh terhadap hasil prediksi suhu?
3. Bagaimana performa model LSTM dalam melakukan prediksi suhu jangka pendek (10 menit hingga 8 jam)?
4. Bagaimana mengimplementasikan model ke dalam aplikasi web yang interaktif dan mudah digunakan?

**1.3 Tujuan Penelitian**

Tujuan dari penelitian ini adalah untuk:

1. Mengembangkan model **LSTM multivariate** untuk memprediksi suhu berdasarkan data cuaca time series.
2. Menganalisis pengaruh fitur cuaca terhadap akurasi hasil prediksi suhu.
3. Mengevaluasi performa model menggunakan metrik RMSE, MAE, dan R².
4. Mengimplementasikan model LSTM ke dalam aplikasi web berbasis Flask untuk prediksi suhu secara real-time.

**1.4 Manfaat Penelitian**

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Bagi masyarakat umum, hasil penelitian ini dapat digunakan untuk membantu perencanaan aktivitas yang dipengaruhi oleh perubahan suhu dan cuaca.
2. Bagi akademisi dan peneliti, penelitian ini dapat menjadi referensi dalam pengembangan sistem prediksi berbasis deep learning.
3. Bagi industri dan pemerintah, hasil penelitian ini dapat digunakan dalam sistem pemantauan cuaca jangka pendek yang mendukung pengambilan keputusan cepat dan efisien.
4. Bagi pengembang sistem, penelitian ini dapat dijadikan dasar untuk pengembangan aplikasi prediksi cuaca real-time yang terintegrasi dengan sensor IoT.

**TINJAUAN PUSTAKA**

**2.1 Data Time Series**

Data time series merupakan sekumpulan data yang direkam secara berurutan berdasarkan waktu dengan interval tertentu. Ciri khas dari data time series adalah adanya ketergantungan antara data masa lalu dan masa depan (temporal dependency). Dalam konteks cuaca, time series forecasting digunakan untuk memprediksi kondisi atmosfer seperti suhu, kelembaban, tekanan, atau curah hujan berdasarkan pola historis. Pendekatan konvensional seperti **Autoregressive Integrated Moving Average (ARIMA)** mampu menangani pola linier, namun kurang efektif dalam memodelkan hubungan non-linear yang kompleks. Oleh karena itu, pendekatan **deep learning** seperti LSTM lebih sesuai karena mampu mengenali pola jangka panjang dan non-linear secara simultan.

**2.2 Deep Learning untuk Prediksi Cuaca**

**Deep learning** merupakan cabang dari machine learning yang menggunakan jaringan saraf tiruan (*artificial neural network*) dengan banyak lapisan untuk mengekstraksi pola kompleks dari data. Dalam prediksi cuaca, model deep learning telah banyak digunakan untuk memprediksi suhu, curah hujan, dan kecepatan angin dengan akurasi tinggi.

Model deep learning unggul karena kemampuannya dalam menangani:

1. Data berdimensi besar dan kompleks.
2. Hubungan non-linear antar variabel cuaca.
3. Ketergantungan jangka panjang antar data historis.

Beberapa arsitektur yang sering digunakan antara lain **Feedforward Neural Network (FNN), Convolutional Neural Network (CNN)** untuk data spasial, serta **Recurrent Neural Network (RNN)** dan **LSTM** untuk data sekuensial seperti time series.

**2.3 Long Short-Term Memory (LSTM)**

**Long Short-Term Memory (LSTM)** merupakan pengembangan dari **Recurrent Neural Network (RNN)** yang dirancang untuk mengatasi masalah *vanishing gradient*. LSTM menggunakan tiga gerbang utama yaitu:

1. **Forget Gate** – menentukan informasi apa yang perlu dihapus dari memori.
2. **Input Gate** – mengontrol informasi baru yang akan disimpan.
3. **Output Gate** – menentukan informasi mana yang akan digunakan sebagai keluaran.

Dengan struktur tersebut, LSTM mampu mengingat pola jangka panjang dan menyesuaikan pembelajaran terhadap konteks data sebelumnya. Model ini sangat cocok untuk data *time series* yang memiliki hubungan antar waktu, seperti data suhu udara.

Kelebihan utama LSTM meliputi:

* Dapat menangkap temporal dependency jangka panjang.
* Mampu mengenali pola non-linear dan kompleks.
* Lebih stabil dibanding RNN standar saat proses pelatihan.

**2.4 Penelitian Terkait**

Berbagai penelitian sebelumnya telah membuktikan efektivitas LSTM dalam prediksi cuaca:

* **Siami-Namini et al. (2019)** menunjukkan bahwa LSTM dan BiLSTM memberikan performa lebih baik dibandingkan ARIMA dalam prediksi suhu dan curah hujan.
* **Greff et al. (2017)** melakukan studi komprehensif terhadap arsitektur LSTM dan menemukan bahwa variasi jumlah unit serta lapisan berpengaruh signifikan terhadap akurasi prediksi.
* **Salinas et al. (2020)** mengembangkan DeepAR, model berbasis LSTM untuk prediksi probabilistik time series dengan hasil akurasi tinggi.
* **Hyndman dan Athanasopoulos (2018)** menekankan pentingnya pemilihan fitur temporal seperti bulan dan hari dalam meningkatkan performa prediksi musiman.

Penelitian ini mengadopsi pendekatan serupa dengan fokus pada prediksi suhu udara jangka pendek menggunakan LSTM multivariate dan implementasi dalam aplikasi web berbasis Flask, yang menjadi nilai tambah dari penelitian ini dibandingkan studi sebelumnya.

**2.5 Kerangka Teori Penelitian**

Berdasarkan studi literatur di atas, penelitian ini berlandaskan pada teori *time series forecasting* dan model LSTM. Alur penelitian dimulai dari pengumpulan data cuaca, pra-pemrosesan (normalisasi, feature engineering, dan pembentukan sekuens), pelatihan model LSTM, evaluasi performa menggunakan RMSE, MAE, dan R², hingga implementasi model dalam aplikasi web untuk prediksi suhu secara real-time.

Dengan kerangka ini, diharapkan penelitian dapat menghasilkan model prediksi suhu yang akurat dan efisien untuk mendukung sistem peramalan cuaca jangka pendek berbasis data.

**METODOLOGI PENELITIAN**

* 1. **Desain Penelitian**

Penelitian ini menggunakan pendekatan eksperimen kuantitatif berbasis data time series, di mana model Long Short-Term Memory (LSTM) dikembangkan untuk memprediksi suhu udara berdasarkan data cuaca historis. Penelitian dilakukan melalui beberapa tahapan utama, yaitu:

1. Pengumpulan dan pemahaman dataset cuaca.
2. Pra-pemrosesan dan pembuatan fitur temporal.
3. Pembuatan model LSTM multivariate.
4. Pelatihan dan evaluasi model.

Implementasi model ke dalam aplikasi web berbasis Flask.

* 1. **Dataset**

Dataset yang digunakan merupakan data cuaca yang dikumpulkan secara kontinu dengan interval waktu **10 menit** selama periode **1 Januari 2020 hingga 1 Januari 2021**. Dataset ini berisi **52.696 data observasi** dengan 20 fitur cuaca. Data ini mencakup parameter fisik atmosfer seperti suhu, kelembaban, tekanan udara, kecepatan angin, curah hujan, dan radiasi gelombang pendek.

**Karakteristik Dataset:**

* **Jumlah data**: 52.696 baris
* **Periode waktu**: Januari 2020 – Januari 2021
* **Interval waktu**: 10 menit
* **Jumlah fitur**: 20 kolom cuaca
* **Target variabel**: Suhu udara (**T** dalam °C)

**Fitur yang Digunakan untuk Pemodelan (8 fitur utama):**

1. Suhu udara (°C)
2. Kelembaban relatif (%)
3. Tekanan udara (hPa)
4. Kecepatan angin (m/s)
5. Curah hujan (mm)
6. Radiasi gelombang pendek (W/m²)
7. Bulan (1–12)
8. Hari dalam tahun (1–366)
   1. **Exploratoty Data Analysis (EDA)**

Tahap ini dilakukan untuk memahami karakteristik dan pola data sebelum dilakukan pemodelan.

**Langkah-langkah EDA**:

1. **Analisis deskriptif**: Melihat nilai minimum, maksimum, mean, dan standar deviasi dari setiap fitur.
2. **Analisis tren**: Melihat perubahan suhu terhadap waktu untuk memahami pola musiman dan fluktuasi harian.
3. **Analisis korelasi**: Menghitung hubungan antar fitur menggunakan correlation matrix untuk menentukan variabel yang paling berpengaruh terhadap suhu.
4. **Visualisasi data**: Membuat grafik line plot dan heatmap untuk menampilkan hubungan antar fitur cuaca.
   1. **Pra-Pemrosesan Data (Preprocessing)**

Tahapan ini bertujuan untuk menyiapkan data agar sesuai dengan format yang dibutuhkan oleh model LSTM.

* + 1. **Penanganan Missing Values**

# Cek missing values

missing\_values = df.isna().sum()

missing\_percent = (missing\_values / len(df)) \* 100

missing\_df = pd.DataFrame({

    'Kolom': missing\_values.index,

    'Jumlah Missing': missing\_values.values,

    'Persentase (%)': missing\_percent.values

})

missing\_df = missing\_df[missing\_df['Jumlah Missing'] > 0].sort\_values('Jumlah Missing', ascending=False)

if len(missing\_df) > 0:

    print("⚠️ MISSING VALUES DITEMUKAN:")

    display(missing\_df)

    # Handling missing values

    print("\nMelakukan interpolasi untuk mengisi missing values...")

    df = df.interpolate(method='time')

    df = df.fillna(method='bfill')

    print(f"✅ Missing values setelah interpolasi: {df.isna().sum().sum()}")

else:

    print("✅ TIDAK ADA MISSING VALUES!")

Dataset telah melalui proses pembersihan sehingga tidak ditemukan nilai kosong (missing values).

* + 1. **Feature Engineering**

Dibuat beberapa fitur temporal tambahan dari kolom waktu untuk meningkatkan kemampuan model dalam mengenali pola musiman:

* **Bulan**: 1–12
* **Hari dalam tahun**: 1–366
* **Hari dalam minggu**: 0–6
* **Jam**: 0–23
  + 1. **Normalisasi Data**

Seluruh fitur dinormalisasi menggunakan MinMaxScaler ke rentang [0, 1] untuk mempercepat proses training dan mencegah dominasi nilai besar.

# Scaling menggunakan MinMaxScaler

scaler = MinMaxScaler(feature\_range=(0, 1))

data\_scaled = scaler.fit\_transform(df\_model)

print("✅ Data berhasil di-scale!")

print(f"Shape data setelah scaling: {data\_scaled.shape}")

print(f"\nRange data setelah scaling:")

print(f"  Min: {data\_scaled.min():.4f}")

print(f"  Max: {data\_scaled.max():.4f}")

print(f"\nScaler akan disimpan untuk digunakan saat prediksi!")

* + 1. **Pembuatan Sekuens Data**

Untuk melatih model LSTM, data perlu diubah menjadi bentuk sekuensial (sliding window).

# Fungsi untuk membuat sequences

def create\_sequences(data, timesteps=60, target\_col\_idx=0):

    """

    Membuat sequences untuk LSTM

    Parameters:

    - data: array data yang sudah di-scale

    - timesteps: jumlah timesteps untuk window

    - target\_col\_idx: index kolom target (0 untuk suhu)

    Returns:

    - X: array input sequences

    - y: array target values

    """

    X, y = [], []

    for i in range(timesteps, len(data)):

        X.append(data[i-timesteps:i])

        y.append(data[i, target\_col\_idx])

    return np.array(X), np.array(y)

# Set timesteps

timesteps = 60

print(f"Membuat sequences dengan timesteps = {timesteps}...")

X, y = create\_sequences(data\_scaled, timesteps=timesteps, target\_col\_idx=0)

print(f"\n✅ Sequences berhasil dibuat!")

print(f"Shape X (input): {X.shape}")

print(f"Shape y (target): {y.shape}")

print(f"\nPenjelasan shape X: (samples, timesteps, features)")

print(f"  - Samples: {X.shape[0]}")

print(f"  - Timesteps: {X.shape[1]}")

print(f"  - Features: {X.shape[2]}")

Membuat sequences dengan timesteps = 60...

✅ Sequences berhasil dibuat!

Shape X (input): (52636, 60, 8)

Shape y (target): (52636,)

Penjelasan shape X: (samples, timesteps, features)

- Samples: 52636

- Timesteps: 60

- Features: 8

* 1. **Pembagian Data**

Dataset dibagi menjadi tiga bagian secara berurutan (tanpa acakan) untuk menghindari data leakage dan mensimulasikan kondisi prediksi sebenarnya:

| **Jenis Data** | **Persentase** | **Jumlah Sampel** |
| --- | --- | --- |
| Training | 70% | 36.845 |
| Validation | 15% | 7.895 |
| Testing | 15% | 7.896 |

* 1. **Arsitektur Model LSTM**

Model yang digunakan adalah **LSTM multivariate** dengan arsitektur dua lapisan tersembunyi:

| **Lapisan** | **Jenis Layer** | **Unit** | **Fungsi Aktivasi** |
| --- | --- | --- | --- |
| 1 | LSTM | 100 | tanh |
| 2 | Dropout | 0.2 | - |
| 3 | LSTM | 50 | tanh |
| 4 | Dropout | 0.2 | - |
| 5 | Dense | 25 | relu |
| 6 | Dense | 1 | linear |

**Hyperparameter**:

* **Optimizer**: Adam (learning\_rate=0.001)
* **Loss Function**: Mean Squared Error (MSE)
* **Metrics**: Mean Absolute Error (MAE)
* **Batch Size**: 32
* **Epochs**: 100 (dengan early stopping)
  1. **Proses Pelatihan Model**

Proses training dilakukan dengan menambahkan beberapa callback functions untuk menjaga performa model:

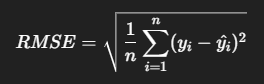
1. **EarlyStopping** – menghentikan training jika validation loss tidak membaik selama 10 epoch berturut-turut.
2. **ModelCheckpoint** – menyimpan model terbaik berdasarkan validation loss.
3. **ReduceLROnPlateau** – menurunkan learning rate sebesar 50% jika performa stagnan.

Model dilatih hingga mencapai konvergensi pada **epoch ke-15** dengan performa validasi stabil.

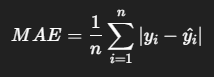
* 1. **Evaluasi Model**

Kinerja model dievaluasi menggunakan tiga metrik utama:

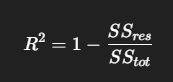
1. **Root Mean Squared Error (RMSE)**



1. **Mean Absolute Error (MAE)**



1. **R-squared**



Metrik ini digunakan untuk mengukur seberapa baik model memprediksi suhu dibandingkan data aktual.

* 1. **Implementasi Aplikasi Web**

Model yang telah dilatih disimpan dalam format **.h5** dan diintegrasikan ke dalam **aplikasi web berbasis Flas**k. Aplikasi ini memiliki fitur:

1. Visualisasi data suhu historis.
2. Prediksi suhu satu langkah (10 menit) dan multi-langkah (hingga 8 jam).
3. Tampilan *dashboard* interaktif dan *responsive*.

Struktur sistem web secara umum terdiri dari tiga komponen utama:

* **Frontend**: HTML, CSS, dan JavaScript untuk antarmuka pengguna.
* **Backend**: Flask untuk logika aplikasi dan pemanggilan model LSTM.
* **Model**: File **.h5** hasil training yang digunakan untuk melakukan prediksi.

**HASIL DAN PEMBAHASAN**

* 1. **Hasil Visualisasi Awal Data**

Tahap awal penelitian dimulai dengan melakukan visualisasi terhadap data suhu untuk memahami pola dan tren yang terjadi sepanjang periode pengamatan.