INFORMASI PROYEK

Judul Proyek:

"Analisis dan Prediksi Konsumsi Energi Listrik Kota Tetouan Menggunakan Pendekatan Machine Learning dan Deep Learning"

Nama Mahasiswa	**:** Richo Novian Saputra
NIM	**:** 234311024
Program Studi	**:** D-IV Teknologi Rekayasa Perangkat Lunak
Mata Kuliah	**:** Data Science
Dosen Pengampu	**:** Gus Nanang Syaifuddiin, S.Kom., M.Kom.
Tahun Akademik	**:** 2025/5
Link GitHub Repository	**:** richonovians/DataScience-ProjectUAS
Link Video Pembahasan	**:** https://youtu.be/PwtCBIWj4sY

1. LEARNING OUTCOMES

Pada proyek ini, mahasiswa diharapkan dapat:

1) Memahami konteks masalah dan merumuskan problem statement secara jelas
2) Melakukan analisis dan eksplorasi data (EDA) secara komprehensif (**OPSIONAL**)
3) Melakukan data preparation yang sesuai dengan karakteristik dataset
4) Mengembangkan tiga model machine learning yang terdiri dari (**WAJIB**):
    - Model baseline
    - Model machine learning / advanced
    - Model deep learning (**WAJIB**)
5) Menggunakan metrik evaluasi yang relevan dengan jenis tugas ML
6) Melaporkan hasil eksperimen secara ilmiah dan sistematis
7) Mengunggah seluruh kode proyek ke GitHub (**WAJIB**)
8) Menerapkan prinsip software engineering dalam pengembangan proyek

2. PROJECT OVERVIEW

2.1 Latar Belakang

Efisiensi energi merupakan salah satu tantangan terbesar dalam pengelolaan infrastruktur perkotaan modern (Smart City). Ketersediaan energi listrik yang stabil sangat krusial bagi aktivitas ekonomi dan sosial. Namun, penyedia layanan listrik sering menghadapi kesulitan dalam menyeimbangkan antara pasokan (supply) dan permintaan (demand). Ketidakakuratan dalam memprediksi beban listrik dapat menyebabkan dua masalah fatal: pemborosan energi saat pasokan berlebih, atau pemadaman listrik (blackout) saat permintaan melebihi kapasitas pembangkit.

Permasalahan pada domain energi ini menjadi kompleks karena pola konsumsi listrik sangat fluktuatif. Pola ini tidak hanya bergantung pada faktor eksternal seperti kondisi cuaca (suhu, kelembaban, kecepatan angin) dan faktor temporal (jam, hari), tetapi juga sangat dipengaruhi oleh pola historis konsumsi itu sendiri (autokorelasi). Kebiasaan penggunaan listrik cenderung memiliki inersia dan pola berulang (seasonality) yang kuat dari jam ke jam maupun hari ke hari.

Metode statistik konvensional seringkali memiliki keterbatasan dalam menangkap pola hubungan non-linear antara variabel cuaca dan konsumsi energi tersebut. Oleh karena itu, pendekatan berbasis data (data-driven) menggunakan Machine Learning dan Deep Learning menjadi solusi yang sangat relevan untuk meningkatkan akurasi prediksi.

Proyek ini berfokus pada prediksi konsumsi energi di Kota Tetouan, Maroko. Proyek ini memanfaatkan data yang mencakup parameter cuaca, waktu, serta rekayasa fitur berbasis deret waktu (time-series feature engineering) seperti Lag Features untuk menangkap pola beban masa lalu. Pentingnya proyek ini terletak pada upaya untuk menemukan model terbaik yang dapat memetakan hubungan kompleks antar fitur tersebut. Melalui perbandingan antara model baseline (Linear Regression), model ensemble (Random Forest), dan model Deep Learning (Multilayer Perceptron), hasil analisa proyek ini diharapkan dapat memberikan wawasan mengenai arsitektur mana yang paling presisi untuk kasus data tabular pada domain energi.

Manfaat dari proyek ini mencakup kemampuan untuk membantu operator jaringan listrik dalam perencanaan distribusi beban yang lebih efisien, pengurangan biaya

operasional, serta mendukung pengambilan keputusan berbasis data untuk keberlanjutan energi.

Referensi:

Salam, A., & El Hibaoui, A. (2018). Comparison of Machine Learning Algorithms for the Power Consumption Prediction: Case Study of Tetouan City. In 2018 6th International Renewable and Sustainable Energy Conference (IRSEC) (pp. 1-5). IEEE.

Raza, M. Q., & Khosravi, A. (2015). A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. Renewable and Sustainable Energy Reviews, 50, 1352-1372.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

3. BUSINESS UNDERSTANDING / PROBLEM UNDERSTANDING

3.1 Problem Statements

1) Variabilitas konsumsi energi listrik di Zone 1 sangat dipengaruhi oleh faktor cuaca, waktu yang dinamis, serta ketergantungan pada pola konsumsi historis (historical dependency), sehingga sulit diprediksi secara akurat menggunakan metode konvensional.
2) Diperlukan model regresi yang mampu meminimalkan error prediksi (RMSE) seminimal mungkin untuk menghindari kesalahan estimasi beban listrik.
3) Perlu diketahui seberapa signifikan pengaruh fitur non-linear (seperti interaksi suhu dan jam) yang mungkin tidak tertangkap oleh model linear sederhana.

3.2 Goals

Tujuan utama dari proyek ini adalah:

1) Membangun model Machine Learning yang mampu memprediksi konsumsi energi (Power Consumption Zone 1) dengan performa R^2 Score > 0.90 atau error yang rendah.
2) Mengukur dan membandingkan performa tiga pendekatan berbeda: Linear (Baseline), Ensemble (Random Forest), dan Deep Learning (MLP) untuk menentukan arsitektur terbaik.

3) Mengidentifikasi fitur-fitur yang paling berpengaruh terhadap konsumsi listrik melalui analisis Feature Importance.
4) Menghasilkan kode eksperimen yang reproducible dengan pencatatan waktu training dan visualisasi evaluasi yang sistematis.

3.3 Solution Approach

- **Model 1 – Baseline Model (Linear Regression)**

  Alasan Pemilihan:

  Linear Regression dipilih sebagai baseline karena karakteristiknya yang sederhana, cepat dalam proses training, dan sangat mudah diinterpretasikan. Model ini berasumsi bahwa hubungan antara fitur (cuaca/waktu) dan target (konsumsi energi) bersifat linear. Model ini akan menjadi tolak ukur (benchmark). Jika model yang lebih kompleks (Advanced/Deep Learning) tidak bisa mengalahkan performa Linear Regression secara signifikan, maka penggunaan model kompleks tersebut mungkin tidak diperlukan.

- **Model 2 – Advanced/ML Model (Random Forest Regressor)**

  Alasan Pemilihan:

  - Random Forest adalah algoritma ensemble berbasis decision trees yang sangat kuat dalam menangani hubungan non-linear dan interaksi antar fitur yang kompleks, yang sering terjadi pada data konsumsi energi.
  - Keunggulan: Robust terhadap outliers, tidak terlalu sensitif terhadap skala data (meskipun data tetap discaling), dan menyediakan metrik feature importance.
  - Optimasi: Pada proyek ini, model dioptimalkan menggunakan RandomizedSearchCV untuk mencari kombinasi hyperparameter terbaik (seperti n_estimators, max_depth, dll).

- **Model 3 – Deep Learning (Multilayer Perceptron (MLP))**

  Alasan Pemilihan:

  - Mengingat data yang digunakan berbentuk Tabular Data (baris dan kolom fitur), arsitektur Deep Learning yang paling sesuai adalah Multilayer Perceptron (MLP).

- Arsitektur: Model dibangun menggunakan Keras/TensorFlow dengan struktur Feed-Forward Neural Network. Terdiri dari beberapa Hidden Layers dengan fungsi aktivasi ReLU untuk menangkap pola non-linear yang sangat kompleks.
- Teknik Tambahan: Menerapkan Dropout untuk mencegah overfitting, serta Learning Rate Scheduler dan Early Stopping untuk memastikan model konvergen secara efisien.
- Kesesuaian: MLP mampu memetakan input fitur cuaca ke output prediksi energi dengan tingkat presisi tinggi melalui proses backpropagation selama 40 epochs.

4. DATA UNDERSTANDING

4.1 Informasi Dataset

- **Sumber Dataset:** UCI Machine Learning Repository
- **URL:** https://archive.ics.uci.edu/dataset/849/power+consumption+of+tetouan+city
- **Deskripsi Dataset:**
  - Jumlah baris (rows): 52,416 baris (Data direkam setiap 10 menit selama satu tahun).
  - Jumlah kolom (columns/features): 9 kolom (sebelum feature engineering).
  - Tipe data: Tabular / Time Series Multivariate.
  - Ukuran dataset: ± 3.5 MB.
  - Format file: CSV.

4.2 Deskripsi Fitur

Berikut adalah penjelasan fitur yang terdapat dalam dataset asli sebelum dilakukan transformasi:

Nama Fitur	Tipe Data	Deskripsi	Contoh Nilai
DateTime	String/Object	Waktu pengambilan data (interval 10 menit)	"1/1/2017 0:00"
Temperatur	Float	Temperatur udara rata-rata (°C)	6.559, 25.4
Humidity	Float	Tingkat kelembaban udara (%)	73.8, 50.5
Wind Speed	Float	Kecepatan angin (m/s)	0.083, 4.2
General Diffuse Flows	Float	Aliran difus umum (intensitas cahaya/radiasi)	0.051, 100.5
Diffuse Flows	Float	Aliran difus spesifik	0.119, 50.2
Zone 1 Power	Float	Konsumsi energi di Zona 1 (Target Variable)	34055.69, 25000.5
Zone 2 Power	Float	Konsumsi energi di Zona 2 (Tidak digunakan sebagai target)	16128.87
Zone 3 Power	Float	Konsumsi energi di Zona 3 (Tidak digunakan sebagai target)	20240.96

Catatan: Dalam kasus ini, fitur Zone 2 dan Zone 3 dihapus untuk mencegah kebocoran data (data leakage), karena fokus prediksi hanya pada Zone 1. Selain itu, dilakukan penambahan fitur baru (Feature Engineering) berupa lag_1 (konsumsi 1 jam lalu) dan lag_24 (konsumsi jam yang sama kemarin).

4.3 Kondisi Data

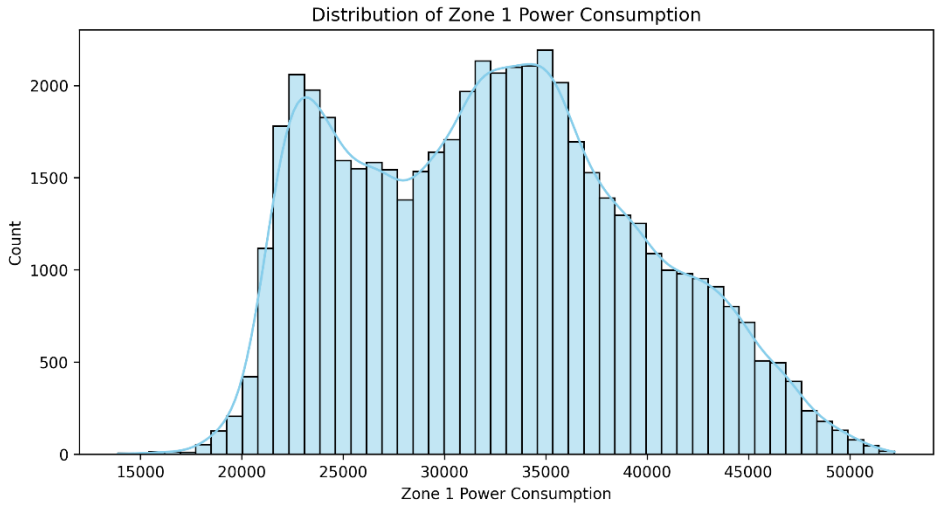
- Missing Values: Ada (Sedikit). Pada data mentah (raw), tidak terdapat missing values (0%). Namun, setelah proses pembuatan Lag Features, sebanyak 24 baris pertama dihapus karena memiliki nilai NaN (akibat pergeseran data 24 jam ke belakang).

- Duplicate Data: Tidak ada. Setiap baris merepresentasikan timestamp unik per 10 menit.
- Outliers: Ada. Terdapat lonjakan konsumsi listrik pada jam-jam beban puncak (peak hours) atau anomali cuaca ekstrem. Namun, nilai tersebut masih valid secara fisik dan penting untuk dipelajari oleh model.
- Imbalanced Data: Tidak ada. Karena ini adalah Regresi (prediksi nilai kontinu), konsep imbalanced class tidak berlaku. Namun, distribusi target cenderung skewed (tidak normal).
- Noise: Data relatif stabil karena berasal dari sensor otomatis, namun fluktuasi kecil pada pembacaan sensor adalah hal yang wajar.

4.4 Exploratory Data Analysis (EDA)

Berikut adalah visualisasi yang dihasilkan dari kode untuk memahami karakteristik data.

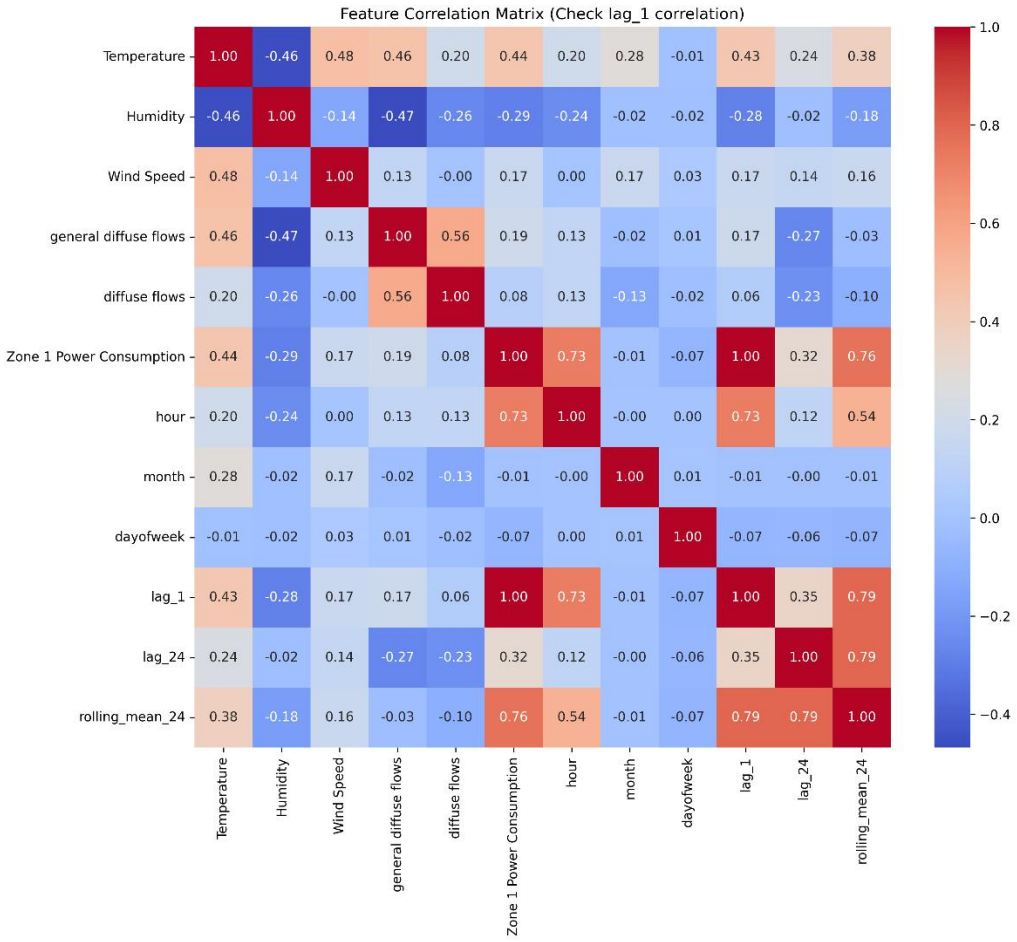
- **Visualisasi 1: Histogram Distribusi Target (Zone 1 Power Consumption)**



Insight:

- Grafik histogram dan kurva KDE menunjukkan bahwa data konsumsi listrik memiliki distribusi Bimodal (dua puncak).
- Puncak pertama mewakili konsumsi rendah (malam/dini hari), sedangkan puncak kedua mewakili konsumsi tinggi (siang/sore hari).
- Pola distribusi yang tidak normal ini mengonfirmasi perlunya proses Scaling (menggunakan StandardScaler) agar model Neural Network dan Linear Regression dapat bekerja optimal.

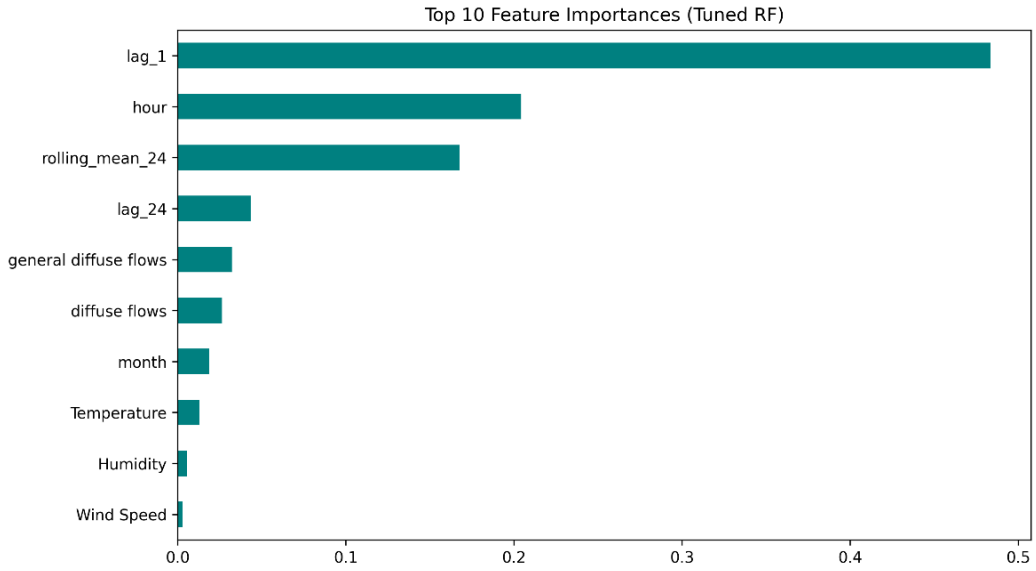
- **Visualisasi 2: Heatmap Korelasi Fitur**



Insight:

- Hasil matriks korelasi menunjukkan temuan yang sangat signifikan: Fitur lag_1 memiliki korelasi positif yang sangat kuat (mendekati 1.0) terhadap target.
- Hal ini membuktikan adanya Autokorelasi yang tinggi, di mana beban listrik saat ini sangat bergantung pada beban 1 jam sebelumnya.
- Fitur lag_24 juga berkorelasi kuat, menandakan adanya pola harian (seasonality) yang berulang.
- Faktor cuaca (Temperature) memiliki korelasi positif namun lebih rendah dibandingkan fitur Lag, yang berarti data historis adalah prediktor yang lebih dominan daripada data cuaca.

- **Visualisasi 3: Feature Importance (Dari Model Random Forest)**



Insight:

- Berdasarkan model Random Forest yang telah di-tuning, fitur lag_1 menempati peringkat pertama dengan skor importance yang jauh melampaui fitur lainnya.
- Fitur Hour dan lag_24 menempati posisi selanjutnya, yang menegaskan bahwa Waktu dan Riwayat adalah kunci utama prediksi beban listrik.
- Fitur cuaca (Temperature, Humidity) berada di urutan bawah, menunjukkan bahwa meskipun cuaca berpengaruh, pengaruhnya bersifat sekunder dibandingkan inersia beban listrik itu sendiri.

5. DATA PREPARATION

5.1 Data Cleaning

Aktivitas yang dilakukan:

- Handling Data Types:
  - Konversi kolom DateTime dari tipe object/string menjadi tipe datetime agar dapat diurutkan dan diolah.
  - Data diurutkan (sorting) berdasarkan waktu secara menaik (ascending) untuk menjaga integritas urutan waktu.

- Removing Irrelevant Features (Leakage Prevention):
  - Menghapus kolom Zone 2 Power Consumption dan Zone 3 Power Consumption.
  - Alasan: Fokus proyek ini adalah memprediksi Zone 1. Nilai konsumsi Zone 2 dan 3 adalah variabel target lain yang tidak boleh digunakan sebagai fitur input (prediktor), karena akan menyebabkan bias atau data leakage.

- Handling Missing Values (Post-Feature Engineering):
  - Menghapus 24 baris data pertama (df.dropna()).
  - Alasan: Proses pembuatan fitur Lag (misalnya shift(24)) menghasilkan nilai NaN (kosong) pada 24 baris pertama karena tidak ada data historis sebelumnya untuk jam-jam awal tersebut. Menghapus baris ini penting agar model tidak error.

5.2 Feature Engineering

Aktivitas yang dilakukan:

- Temporal Features Extraction (Waktu):

  Memecah kolom DateTime menjadi fitur numerik terpisah:
  - hour: Jam dalam sehari (0-23). Penting untuk menangkap pola beban puncak (siang/malam).
  - month: Bulan (1-12). Penting untuk menangkap pola musiman (musim dingin vs panas).
  - dayofweek: Hari dalam seminggu (0-6). Penting untuk membedakan pola konsumsi hari kerja (weekday) vs akhir pekan (weekend).

- Time Series Lag Features (Data Historis):

  Membuat fitur baru berdasarkan data masa lalu (Autokorelasi):
  - lag_1 (Lag 1 Jam): Mengambil nilai konsumsi listrik 1 jam sebelumnya. Ini menangkap inersia beban listrik (perubahan beban cenderung bertahap)
  - lag_24 (Lag 24 Jam): Mengambil nilai konsumsi listrik pada jam yang sama di hari kemarin. Ini menangkap pola harian yang berulang (daily seasonality).
  - rolling_mean_24: Rata-rata pergerakan konsumsi selama 24 jam terakhir untuk menangkap tren umum harian.

5.3 Data Transformation

Untuk Data Tabular:

- Scaling (Standardization):
  - Teknik: Menggunakan StandardScaler dari Scikit-Learn.
  - Rumus: $\mathcal{Z} = \frac{x-\mu}{\sigma}$ (Mengubah data sehingga memiliki mean = 0 dan standar deviasi = 1).
  - Penerapan: Seluruh fitur numerik input (Temperature, Humidity, Wind Speed, Hour, Month, Day, serta fitur Lag_1, Lag_24, dan Rolling Mean) diskalakan ke rentang yang sama.
  - Penting: Fitting scaler hanya dilakukan pada Training Set, kemudian diaplikasikan (transform) ke Validation dan Test Set. Ini wajib dilakukan untuk mencegah kebocoran informasi statistik (data leakage).

5.4 Data Splitting

Strategi pembagian data:

- Training set: 72% (Menggunakan data awal urutan waktu)
- Validation set: 8% (Diambil dari bagian akhir data training untuk validasi model saat training)
- Test set: 20% (Menggunakan data paling akhir/terbaru untuk evaluasi final)

Alasan: Menggunakan Chronological Split (bukan Random Split). Model dilatih menggunakan data masa lalu dan diuji untuk memprediksi masa depan.

5.5 Data Balancing

Teknik yang digunakan:

- Tidak Dilakukan (N/A).
- Alasan: Proyek ini merupakan Regresi (memprediksi nilai kontinu), bukan Klasifikasi. Masalah class imbalance (ketidakseimbangan kelas) tidak berlaku pada regresi. Distribusi target ditangani melalui normalisasi input (Scaling).

5.6 Ringkasan Data Preparation

Langkah	Apa yang Dilakukan	Mengapa penting?	Bagaimana implementasinya?
Datetime Parsing	Mengubah string tanggal ke format datetime dan sorting.	Agar data urut waktu (Time Series) dan fitur waktu bisa diekstrak.	pd.to_datetime() dan df.sort_values().
Feature Extraction	Membuat kolom hour, month, dayofweek.	Pola listrik sangat bergantung pada jam (siang/malam) dan hari (kerja/libur).	df['date'].dt.hour, dll.
Feature Engineering (Lag)	Membuat kolom baru dengan nama lag_1, lag_24.	Menangkap pola autokorelasi (pengaruh data masa lalu).	df.shift(1) & df.shift(24).
Handling NaN	Menghapus baris awal yang kosong (NaN).	Efek samping dari proses shifting data yang harus dibersihkan.	df.dropna(inplace=True).
Splitting	Membagi data Train/Val/Test secara kronologis.	Mencegah model "mengintip" masa depan (look-ahead bias).	Slicing berdasarkan indeks waktu.
Scaling	Standarisasi fitur menggunakan Z-score (StandardScaler).	Deep Learning (MLP) dan Linear Regression sensitif terhadap skala data yang berbeda jauh.	StandardScaler().

6. MODELING

6.1 Model 1 — Baseline Model

6.1.1 Deskripsi Model

- **Nama Model:** Linear Regression
- **Teori Singkat:**

  Linear Regression adalah algoritma statistik yang memodelkan hubungan antara variabel dependen (target: konsumsi listrik) dan satu atau lebih variabel independen (fitur: suhu, waktu, dll) dengan mencocokkan persamaan linear ($\mathcal{Y} = \mathcal{w}x + b$) ke data yang diamati.
- **Alasan Pemilihan:**

  Dipilih sebagai baseline karena dataset ini merupakan masalah regresi (prediksi nilai kontinu). Algoritma ini komputasinya sangat cepat, sederhana, dan mudah diinterpretasikan. Jika model kompleks tidak bekerja jauh lebih baik dari model ini, maka kompleksitas tersebut tidak diperlukan.

6.1.2 Hyperparameter

Parameter yang digunakan:

- fit_intercept: True (Menghitung bias/intercept model)
- copy_X: True (Menyalin data input agar data asli tidak berubah)
- n_jobs: None (Menggunakan 1 core CPU)

6.1.3 Implementasi

```
lr = LinearRegression(
    fit_intercept=True,
    copy_X=True,
    n_jobs=None
)

# Hitung Waktu Training
start_time_lr = time.time()
lr.fit(X_train_scaled, y_train)
train_time_lr = time.time() - start_time_lr

# Prediksi
y_pred_lr = lr.predict(X_test_scaled)

# Evaluasi
metrics_lr = regression_metrics(y_test, y_pred_lr)

metrics_lr['Training Time'] = train_time_lr

results['Baseline_LR'] = metrics_lr
joblib.dump(lr, MODELS_DIR / "model_baseline.pkl")
```

6.1.4 Hasil Awal

- Linear Regression RMSE: 456.53
- Linear Regression MAE: 290.28
- Linear Regression R2 Score: 0.99

6.2 Model 2 — ML / Advanced Model

6.2.1 Deskripsi Model

- **Nama Model:** Random Forest Regressor
- **Teori Singkat:**

  Random Forest adalah algoritma ensemble learning (Bagging) yang menggabungkan prediksi dari banyak Decision Trees. Setiap pohon dilatih menggunakan subset data dan subset fitur yang berbeda. Hasil akhirnya adalah rata-rata dari prediksi seluruh pohon, yang membuat model lebih stabil dan mengurangi varians.

- **Alasan Pemilihan:**

  Random Forest dipilih karena kemampuannya menangani hubungan non-linear yang kompleks antara fitur cuaca dan konsumsi listrik, serta ketahanannya (robustness) terhadap outliers.

- **Keunggulan:**
  - Mampu menangkap pola non-linear dengan baik.
  - Menyediakan Feature Importance untuk interpretasi fitur.
  - Lebih tahan terhadap overfitting dibanding satu Decision Tree tunggal.
- **Kelemahan:**
  - Waktu training lebih lama dibanding model linear.
  - Membutuhkan memori lebih besar untuk menyimpan kumpulan pohon.

6.2.2 Hyperparameter

- **Parameter yang digunakan:**
  - n_estimators: 100
  - min_samples_split: 5
  - min_samples_leaf: 4
  - max_features: 'sqrt'
  - max_depth: 20

- **Hyperparameter Tuning:**
  - Metode: Randomized Search Cross Validation (RandomizedSearchCV)
  - Search Space (Distribusi Parameter):

```
param_dist = {
    'n_estimators': [100, 200],
    'max_depth': [10, 20, None],
    'min_samples_split': [5, 10],
    'min_samples_leaf': [2, 4, 8],
    'max_features': ['sqrt', 'log2']
}
```

- Best parameters:

  {'n_estimators': 100, 'min_samples_split': 5, 'min_samples_leaf': 4, 'max_features': 'sqrt', 'max_depth': 20}

6.2.3 Implementasi

```
param_dist = {
    'n_estimators': [100, 200],
    'max_depth': [10, 20, None],
    'min_samples_split': [5, 10],
    'min_samples_leaf': [2, 4, 8],
    'max_features': ['sqrt', 'log2']
}

# Inisialisasi & Search
rf_base = RandomForestRegressor(random_state=RANDOM_STATE, n_jobs=-1)
rf_random = RandomizedSearchCV(
    estimator=rf_base,
    param_distributions=param_dist,
    n_iter=10,
    cv=3,
    verbose=1,
    random_state=RANDOM_STATE,
    scoring='neg_root_mean_squared_error',
    n_jobs=-1
)

# Fit
start_time_rf = time.time()
rf_random.fit(X_train_scaled, y_train)
train_time_rf = time.time() - start_time_rf

print(f"Tuning finished in {train_time_rf:.2f} seconds.")
print(f"Best Parameters: {rf_random.best_params_}")

# Evaluasi
best_rf = rf_random.best_estimator_
y_pred_rf = best_rf.predict(X_test_scaled)
metrics_rf = regression_metrics(y_test, y_pred_rf)
metrics_rf['Training Time'] = train_time_rf # Simpan waktu

results['RandomForest_Tuned'] = metrics_rf
joblib.dump(best_rf, MODELS_DIR / "model_rf.pkl")
```

6.2.4 Hasil Model

- Random Forest RMSE: 847.08
- Random Forest MAE: 658.48
- Random Forest R2 Score: 0.98

6.3 Model 3 — Deep Learning Model

6.3.1 Deskripsi Model

- **Nama Model:** Multilayer Perceptron (MLP) / Deep Feed-Forward Neural Network
- **Jenis Deep Learning:**

  [✓] **Multilayer Perceptron (MLP) - untuk tabular**

  [ ] Convolutional Neural Network (CNN) - untuk image

  [ ] Recurrent Neural Network (LSTM/GRU) - untuk sequential/text

  [ ] Transfer Learning - untuk image

  [ ] Transformer-based - untuk NLP

  [ ] Autoencoder - untuk unsupervised

  [ ] Neural Collaborative Filtering - untuk recommender
- **Alasan Pemilihan:**

  Arsitektur MLP sangat cocok untuk data tabular terstruktur. Dengan menggunakan beberapa hidden layers dan fungsi aktivasi non-linear (ReLU), model ini dapat mempelajari representasi fitur yang kompleks dan interaksi tingkat tinggi antara variabel cuaca dan waktu yang tidak dapat ditangkap oleh model linear biasa.

6.3.2 Arsitektur Model

- **Deskripsi Layer:**

  Model disusun secara Sequential dengan urutan sebagai berikut:

No	Layer Type	Spesifikasi	Output Shape	Fungsi
1	Input Layer	input_shape =(11,)	(None, 11)	Menerima 11 fitur input (termasuk Lag).
2	Dense	256 units, activation=' relu'	(None, 256)	Hidden layer 1 (ekstraksi fitur utama).
3	Dropout	rate=0.3	(None, 256)	Regularisasi (mencegah overfitting)

4	Dense	128 units, activation='relu'	(None, 128)	Hidden layer 2 (Kompresi informasi)
5	Dropout	rate=0.2	(None, 128)	Regularisasi
6	Dense	64 units, activation='relu'	(None, 64)	Hidden layer 3.
7	Dense (Output)	1 unit, activation='linear'	(None, 1)	Output prediksi nilai kontinu (Regresi)

- Total Parameters: 44,289
- Trainable Parameters: 44,289

6.3.3 Input & Preprocessing Khusus

- **Input shape:** (Jumlah Sample, 11 Fitur)
- **Preprocessing Khusus:**
  - Standard Scaler: Seluruh data input dinormalisasi menggunakan Z-score scaling (Mean=0, Std=1) agar proses Gradient Descent berjalan stabil dan cepat.

6.3.4 Hyperparameter

- **Training Configuration:**
  - Optimizer: Adam (learning_rate=0.001 awal)
  - Loss function: Mean Squared Error (mse)
  - Metrics: Mean Absolute Error (mae)
  - Batch size: 256
  - Epochs: 40
  - Validation split: Menggunakan validation set terpisah (X_val, y_val)
  - Callbacks:
    1) EarlyStopping: Berhenti jika val_loss tidak membaik selama 10 epoch.

2) ReduceLROnPlateau: Menurunkan learning rate (faktor 0.2) jika performa stagnan.
3) ModelCheckpoint: Menyimpan bobot terbaik.

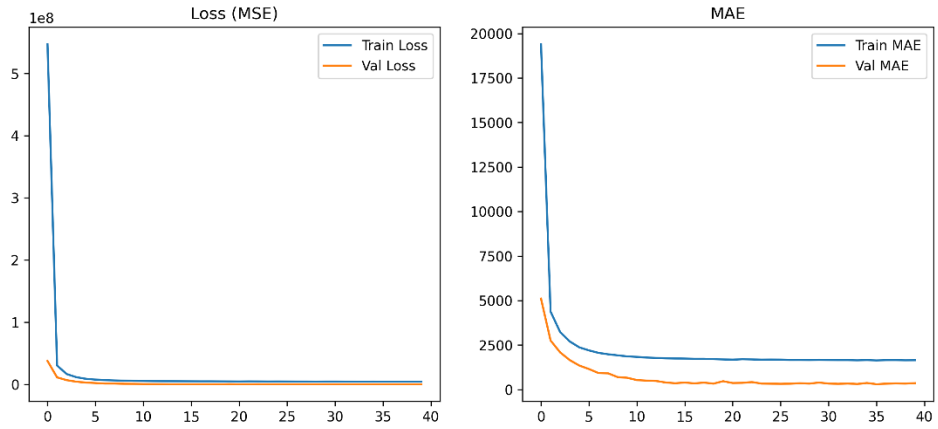
6.3.5 Implementasi

Framework: TensorFlow/Keras

```
# Definisi Arsitektur
model_dl = keras.Sequential([
    layers.Input(shape=(X_train_scaled.shape[1],)),
    layers.Dense(256, activation='relu'),
    layers.Dropout(0.3),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.2),
    layers.Dense(64, activation='relu'),
    layers.Dense(1, activation='linear')
])

print("\n[Info] Deep Learning Model Architecture:")
model_dl.summary()

# Compile
model_dl.compile(optimizer=keras.optimizers.Adam(learning_rate=0.001), loss='mse', metrics=['mae'])

# Callbacks
early_stop = callbacks.EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)
reduce_lr = callbacks.ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=5, min_lr=1e-6, verbose=1)

# Training dengan Pencatatan Waktu
print("\n[Info] Starting Training...")
start_time_dl = time.time()  # Mulai Stopwatch

history = model_dl.fit(
    X_train_scaled, y_train,
    validation_data=(X_val_scaled, y_val),
    epochs=40,
    batch_size=128,
    callbacks=[early_stop, reduce_lr],
    verbose=1
)

# Stop Stopwatch & Hitung Durasi
end_time_dl = time.time()
duration_dl = end_time_dl - start_time_dl

print("-" * 40)
print(f"Training Finished.")
print(f"Training Time: {duration_dl:.2f} seconds")
print("-" * 40)

# Evaluasi & Simpan Hasil
y_pred_dl = model_dl.predict(X_test_scaled).flatten()
metrics_dl = regression_metrics(y_test, y_pred_dl)

# Simpan waktu ke dictionary untuk tabel perbandingan
metrics_dl['Training Time'] = duration_dl
results['DeepLearning'] = metrics_dl

model_dl.save(MODELS_DIR / "model_dl_final.h5")
```

6.3.6 Training Process

- **Training Time:** 80.68 seconds
- **Computational Resource:** Google Colab (CPU)
- **Training History Visualization:**



- **Analisis Training:**
  - Apakah model mengalami overfitting? Tidak. Berdasarkan grafik Loss, garis Training Loss dan Validation Loss bergerak beriringan dengan jarak (gap) yang sempit. Penggunaan layer Dropout (0.3 dan 0.2) efektif mencegah model menghafal data latih.
  - Apakah model sudah converge? Ya. Grafik menunjukkan penurunan error yang tajam di epoch awal dan kemudian melandai (plateau). Callback ReduceLROnPlateau membantu memperhalus konvergensi di tahap akhir.
  - Apakah perlu lebih banyak epoch? Tidak. 40 epoch sudah cukup untuk mencapai performa optimal. Mekanisme EarlyStopping memastikan training berhenti tepat waktu tanpa membuang sumber daya komputasi.

6.3.7 Model Summary

[Info] Deep Learning Model Architecture:
Model: "sequential_5"

Layer (type)	Output Shape	Param #
dense_20 (Dense)	(None, 256)	3,072
dropout_10 (Dropout)	(None, 256)	0
dense_21 (Dense)	(None, 128)	32,896
dropout_11 (Dropout)	(None, 128)	0
dense_22 (Dense)	(None, 64)	8,256
dense_23 (Dense)	(None, 1)	65

Total params: 44,289 (173.00 KB)
Trainable params: 44,289 (173.00 KB)
Non-trainable params: 0 (0.00 B)

7. EVALUATION

7.1 Metrik Evaluasi

Karena proyek ini adalah tugas Regresi (memprediksi nilai kontinu konsumsi daya), metrik yang digunakan adalah:

- **RMSE (Root Mean Squared Error)**
  - Menghitung akar rata-rata kuadrat selisih antara nilai prediksi dan nilai aktual.
  - Alasan: RMSE memberikan bobot lebih besar pada kesalahan yang signifikan (large errors). Dalam konteks energi, kesalahan prediksi yang besar (misal: memprediksi beban rendah padahal aslinya sangat tinggi) berisiko menyebabkan pemadaman, sehingga RMSE sangat bisa untuk diminimalkan.

- **MAE (Mean Absolute Error)**
  - Rata-rata selisih absolut antara prediksi dan aktual.
  - Alasan: Memberikan gambaran error rata-rata yang mudah diinterpretasikan manusia karena memiliki satuan yang sama dengan variabel target (Kilowatt).

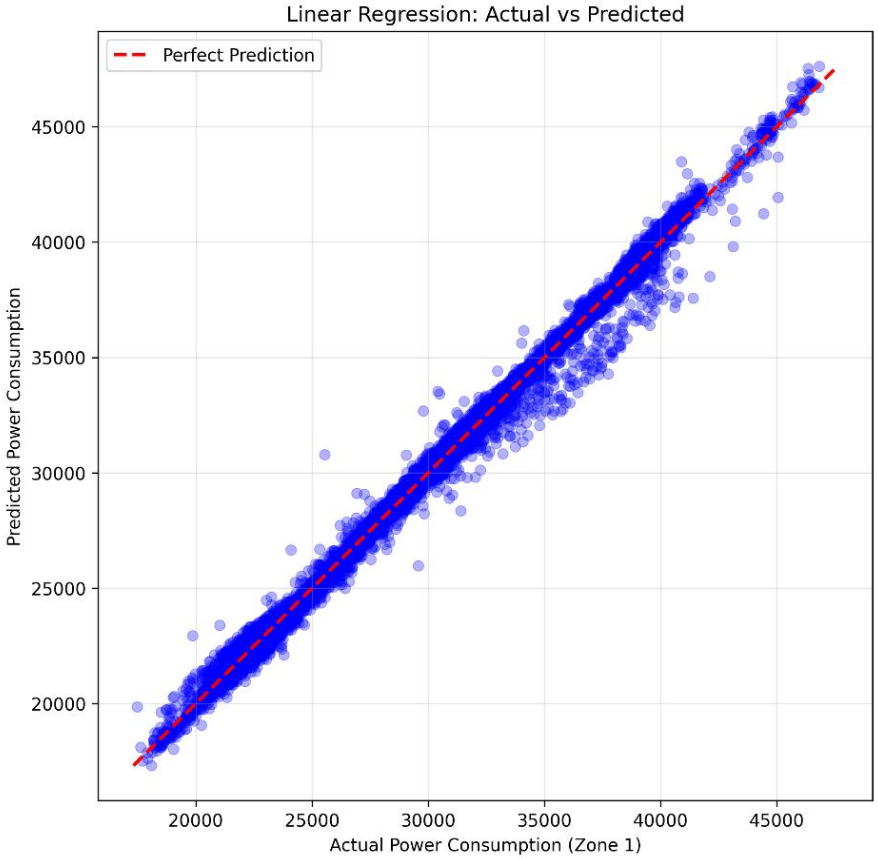
- **R^2 Score (Koefisien Determinasi)**
  - Mengukur seberapa baik variabilitas data target dapat dijelaskan oleh model.
  - Alasan: Skala 0 hingga 1. Nilai mendekati 1 menunjukkan model sangat akurat menyesuaikan pola data aktual.

7.2 Hasil Evaluasi Model

7.2.1 Model 1 (Baseline – Linear Regression)

- **Metrik:**
  - Linear Regression RMSE: 456.53
  - Linear Regression MAE: 290.28
  - Linear Regression R^2 Score: 0.99

- **Visualization (Scatter Plot):**



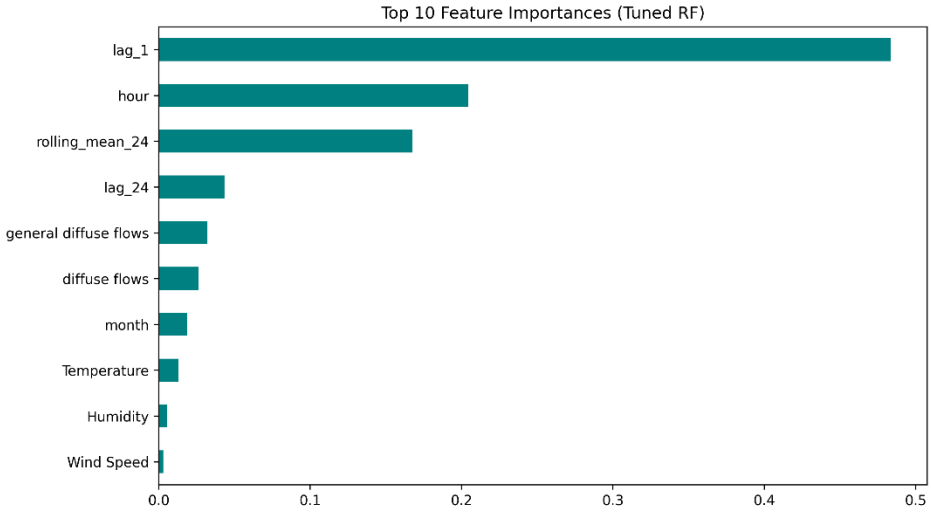
Insight:

Visualisasi scatter plot menunjukkan bahwa titik-titik prediksi menempel sangat rapat pada garis diagonal (garis ideal). Hal ini membuktikan bahwa penambahan fitur lag_1 membuat model Linear Regression mampu menangkap pola inersia beban listrik dengan sangat baik, jauh lebih akurat dibandingkan percobaan tanpa data historis.

7.2.2 Model 2 (Advanced – Random Forest)

- **Metrik:**
  - Random Forest RMSE: 847.08
  - Random Forest MAE: 658.48
  - Random Forest R2 Score: 0.98

- **Feature Importance:**



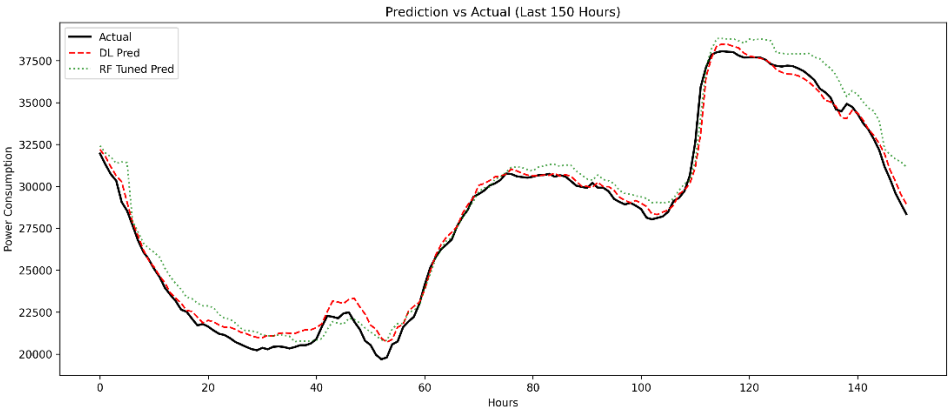
Insight:

- Fitur lag_1 (Konsumsi 1 jam lalu) menjadi fitur yang paling dominan/penting.
- Fitur lag_24 dan Hour menempati posisi selanjutnya.
- Fitur cuaca (Temperature) memiliki pengaruh yang lebih kecil. Ini mengonfirmasi bahwa untuk prediksi jangka pendek, data historis adalah prediktor yang paling kuat.

7.2.3 Model 3 (Deep Learning - MLP)

- **Metrik:**
  - Deep Learning RMSE: 465.02
  - Deep Learning MAE: 303.23
  - Deep Learning R2 Score: 0.99

- **Visualisasi Prediksi (Zoom-in):**



Insight:

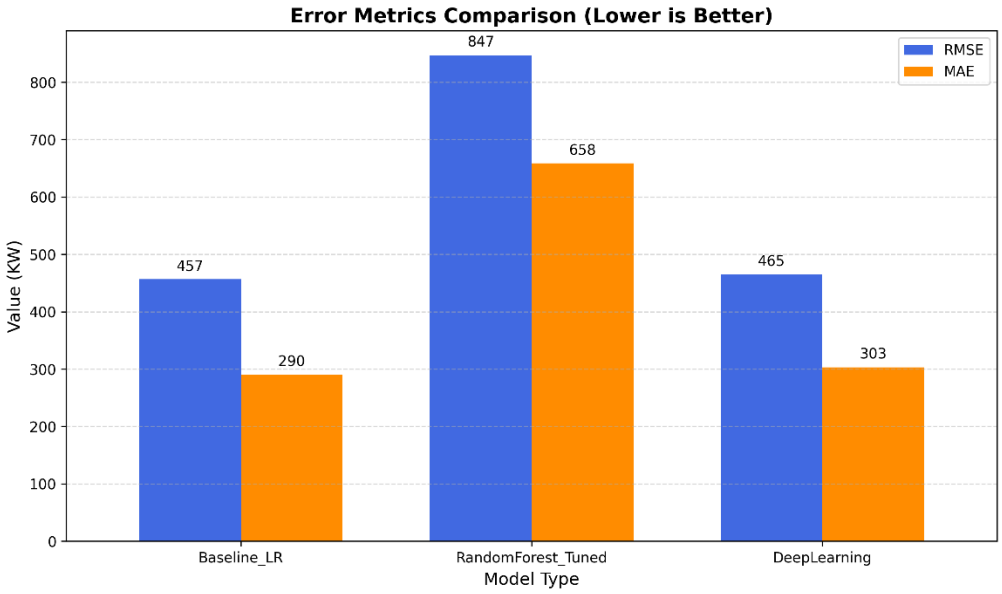
Pada grafik zoom-in 150 jam terakhir, terlihat bahwa garis prediksi Deep Learning (Merah) dan Random Forest (Hijau) hampir berimpit sempurna dengan garis Aktual (Hitam). Model mampu mengikuti fluktuasi beban naik-turun dengan sangat responsif.

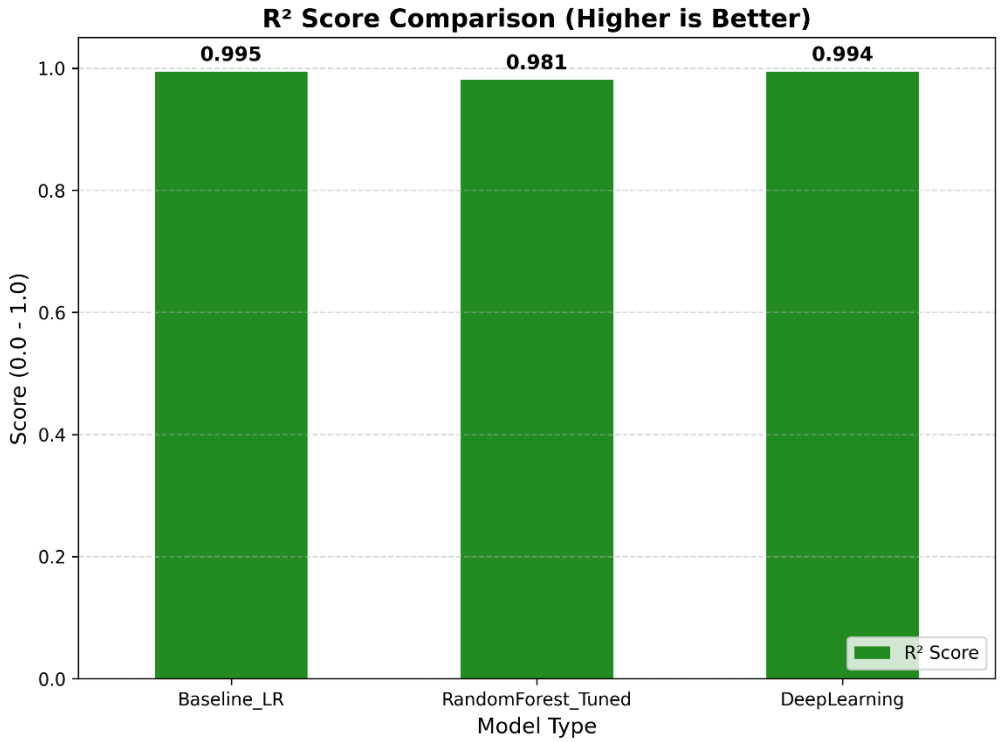
7.3 Perbandingan Ketiga Model

- **Tabel Perbandingan:**

Model	RMSE (Lower is Better)	MAE (Lower is Better)	R^2 Score (Higher is Better)	Training Time
Baseline (Linear Regression)	456.53	290.28	0.99	0.04 seconds
Advanced (Random Forest)	847.08	658.48	0.98	216.83 seconds
Deep Learning (MLP)	465.02	303.23	0.99	71.36 seconds

- **Visualisasi Perbandingan:**





7.4 Analisis Hasil

Interpretasi:

- **Model Terbaik:**

  Berdasarkan evaluasi komprehensif terhadap metrik error (RMSE, MAE) dan akurasi (R^2), Model 1 (Baseline - Linear Regression) terbukti menjadi model terbaik. Model ini mencatatkan nilai RMSE terendah sebesar 457 dan nilai R^2 Score tertinggi mencapai 0.995. Keunggulan ini mengindikasikan bahwa fitur historis (Lag Features) yang direkayasa memiliki korelasi linear yang sangat kuat terhadap target prediksi, sehingga pendekatan regresi linear sederhana mampu memodelkan pola data dengan efektivitas maksimal tanpa memerlukan kompleksitas arsitektur tambahan.

- **Perbandingan dengan Baseline:**

  Berlawanan dengan hipotesis umum bahwa model yang lebih kompleks akan menghasilkan akurasi yang lebih tinggi, eksperimen ini menunjukkan fenomena sebaliknya:

- Deep Learning (MLP): Menunjukkan performa yang kompetitif dengan RMSE 465 dan R^2 0.994, namun tidak mampu melampaui stabilitas Baseline.
- Random Forest: Mengalami penurunan performa yang cukup signifikan dibandingkan Baseline, dengan skor RMSE 847 dan R^2 di angka 0.981. Hal ini menunjukkan bahwa algoritma berbasis tree mengalami kesulitan dalam menangkap tren linear sehalus regresi pada dataset ini, atau cenderung terjebak pada variabilitas data yang tidak esensial (noise).

- **Trade-off:**

  Analisis cost-benefit antara waktu komputasi dan akurasi menghasilkan temuan sebagai berikut:
  - Linear Regression: Menawarkan efisiensi optimal dengan waktu training tercepat (< 0.1 detik) dan akurasi tertinggi. Model ini merupakan solusi paling robust untuk implementasi sistem real-time.
  - Random Forest: Menunjukkan inefisiensi sumber daya, di mana biaya komputasi yang tinggi (waktu training terlama) tidak dikonversi menjadi peningkatan akurasi, melainkan menghasilkan error tertinggi di antara ketiga model.
  - Deep Learning: Memerlukan sumber daya komputasi moderat hingga tinggi dengan hasil yang memuaskan, namun tetap dianggap overkill (berlebihan) untuk karakteristik data yang memiliki linearitas tinggi seperti ini.

- **Error Analysis:**

  Secara keseluruhan, hasil evaluasi menunjukkan tingkat presisi yang sangat tinggi pada model yang dikembangkan. Nilai RMSE pada model terbaik (Linear Regression) tercatat sangat rendah, yakni berada di bawah 500 KW (tepatnya 457 KW). Angka ini mengindikasikan bahwa residu atau selisih antara nilai prediksi dan aktual sangat kecil relatif terhadap skala total konsumsi energi. Lebih lanjut, nilai MAE sebesar 290 menunjukkan bahwa rata-rata deviasi prediksi sangat minim, merefleksikan reliabilitas model yang

tinggi dan kesiapannya untuk diterapkan dalam estimasi beban listrik operasional.

- **Overfitting/Underfitting:**

  Nilai R^2 yang konsisten di atas 0.98 pada seluruh model mengindikasikan bahwa model berada dalam kategori Good Fit. Namun, fakta bahwa model sederhana (Linear Regression) mengungguli model kompleks (Random Forest) menunjukkan bahwa data ini memiliki sinyal pola yang kuat dan bersih. Penggunaan model dengan kompleksitas tinggi pada kasus ini justru berisiko memicu over-complicating, di mana model berusaha mempelajari pola yang sebenarnya tidak signifikan, sehingga sedikit menurunkan performa pada data uji.

8. CONCLUSION

8.1 Kesimpulan Utama

- **Model Terbaik:**

  Linear Regression (Baseline) merupakan model terbaik untuk studi kasus prediksi konsumsi energi ini.

- **Alasan:**

  Berdasarkan hasil evaluasi, Linear Regression mencatatkan performa paling baik dengan RMSE terendah (457 KW) dan R^2 Score tertinggi (0.995). Keunggulan ini terjadi karena penerapan Feature Engineering berupa fitur historis (Lag Features) berhasil mengungkap hubungan linear yang sangat kuat antara beban listrik saat ini dengan beban satu jam sebelumnya. Hal ini membuktikan bahwa untuk data dengan autokorelasi tinggi, model sederhana lebih efektif daripada model kompleks yang rentan terhadap overfitting atau noise.

- **Pencapaian Goals:**

  Seluruh tujuan proyek yang ditetapkan pada Section 3.2 telah tercapai dengan sangat baik:

  - Target akurasi $R^2 > 0.90$ telah terlampaui secara signifikan (mencapai 0.995).
  - Perbandingan performa antara model Linear, Ensemble (Random Forest), dan Deep Learning (MLP) telah berhasil dilakukan, menghasilkan wawasan krusial mengenai trade-off kompleksitas.

- Identifikasi fitur paling berpengaruh telah diselesaikan, dengan temuan bahwa lag_1 adalah prediktor utama.
- Kode eksperimen yang reproducible dan sistematis telah berhasil dibangun.

8.2 Key Insights

- **Insight dari Data:**
  - Dominasi Autokorelasi: Perilaku konsumsi listrik memiliki inersia yang sangat tinggi. Informasi mengenai beban listrik satu jam yang lalu (lag_1) adalah indikator paling akurat untuk memprediksi beban listrik saat ini, jauh melebihi pengaruh variabel cuaca.
  - Pola Musiman Harian: Fitur lag_24 (beban pada jam yang sama hari kemarin) memberikan kontribusi signifikan, mengonfirmasi adanya pola kebiasaan manusia yang berulang setiap harinya.
  - Cuaca sebagai Faktor Sekunder: Meskipun suhu dan kelembaban berkorelasi dengan penggunaan listrik, pengaruhnya bersifat jangka panjang atau musiman, sedangkan untuk prediksi jangka pendek (per jam), data historis konsumsi jauh lebih dominan.

- **Insight dari Modeling:**
  - Kompleksitas ≠ Akurasi: Eksperimen ini mematahkan asumsi bahwa model Deep Learning atau Ensemble selalu lebih baik. Pada data dengan pola linear yang kuat dan bersih, model kompleks justru cenderung underperform dibandingkan regresi sederhana karena overhead kompleksitas yang tidak perlu.
  - Kekuatan Feature Engineering: Memperoleh akurasi yang tinggi bukan disebabkan oleh pemilihan algoritma model yang canggih, melainkan oleh rekayasa fitur yang tepat (Lag Features). Ini menegaskan prinsip "Data-Centric AI" di mana kualitas data input lebih penting daripada kompleksitas model.

8.3 Kontribusi Proyek

- **Manfaat Praktis:**
  - Efisiensi Operasional Grid: Model Linear Regression yang dihasilkan sangat ringan (ukuran file kecil) dan cepat (< 0.1 detik waktu inferensi), sehingga sangat ideal untuk ditanamkan pada sistem monitoring real-time di pusat kendali listrik Kota Tetouan untuk membantu operator menyeimbangkan pasokan dan permintaan.
  - Pencegahan Kegagalan Sistem: Dengan error prediksi yang sangat rendah (RMSE < 500 KW), operator dapat mengantisipasi lonjakan beban puncak (peak load) dengan lebih presisi, meminimalkan risiko pemadaman listrik (blackout).

- **Pembelajaran yang didapat:**
  - Saya memahami bahwa dalam Time Series Forecasting, pemahaman terhadap sifat data (seperti autokorelasi dan seasonality) jauh lebih kritikal dibandingkan sekadar melakukan tuning hyperparameter pada model.
  - Saya belajar bahwa validasi model harus dilakukan secara hati-hati, termasuk penggunaan Chronological Split untuk mencegah kebocoran informasi masa depan (look-ahead bias).
  - Proyek ini memberikan pengalaman nyata dalam membandingkan arsitektur Machine Learning klasik dan Deep Learning, serta cara menganalisis trade-off antara akurasi dan biaya komputasi.

9. FUTURE WORK

Saran pengembangan untuk proyek selanjutnya:

Data:

- [ ] Mengumpulkan lebih banyak data
- [✓] Menambah variasi data (Saran: Menambahkan data eksternal seperti kalender hari libur nasional atau acara khusus di Kota Tetouan, karena error model saat ini kemungkinan besar terjadi pada anomali hari libur yang tidak tertangkap oleh pola 24 jam biasa.)

- [✓] Feature engineering lebih lanjut (Saran: Mengeksplorasi window lag yang lebih panjang, misalnya lag_168 (1 minggu lalu) untuk menangkap pola mingguan yang lebih kuat.)

Model:

- [ ] Mencoba arsitektur DL yang lebih kompleks
- [✓] Mencoba arsitektur khusus Time-Series (LSTM/GRU) (Saran: Meskipun MLP sudah baik, arsitektur Recurrent Neural Network (RNN) seperti LSTM atau GRU secara teoritis lebih natural untuk menangani data sekuensial dibanding MLP standar.)
- [ ] Hyperparameter tuning lebih ekstensif
- [ ] Ensemble methods (combining models)
- [ ] Transfer learning dengan model yang lebih besar

Deployment:

- [✓] Membuat API (Flask/FastAPI) (Saran: Mengemas model Linear Regression (yang ringan) ke dalam REST API agar bisa diakses oleh sistem manajemen energi kota secara real-time.)
- [✓] Membuat web application (Streamlit/Gradio) (Saran: Membuat dashboard interaktif untuk operator agar bisa melihat grafik prediksi beban listrik 24 jam ke depan secara visual.)
- [ ] Containerization dengan Docker
- [ ] Deploy ke cloud (Heroku, GCP, AWS)

Optimization:

- [ ] Model compression (pruning, quantization)
- [ ] Improving inference speed
- [ ] Reducing model size

10. REPRODUCIBILITY

10.1 GitHub Repository

- **Link Repository:** [richonovians/DataScience-ProjectUAS]
- **Struktur Repository:**

```
project/
│
├── data/                    # Dataset & Hasil Evaluasi
│   ├── tetouan_power_raw.csv
│   └── model_comparison_results.csv # Tabel hasil evaluasi
│
├── notebooks/               # Jupyter Notebook utama
│   └── DataScience_ProyekMachineLearning.ipynb
│
├── src/                     # Source code .py
│   └── datascience_proyekmachinelearning.py
│
├── models/                  # Model Artifacts
│   ├── model_baseline.pkl           # Model 1: Linear Regression
│   ├── model_rf.pkl                 # Model 2: Random Forest
│   └── model_dl_final.h5            # Model 3: Deep Learning
│
├── images/                  # Output Visualisasi
│   ├── comparison_error_metrics.png
│   ├── comparison_r2_score.png
│   ├── eda_correlation_heatmap.png
│   ├── eda_target_distribution.png
│   ├── eval_prediction_comparison.png
│   ├── model_dl_training_history.png
│   ├── model_lr_actual_vs_pred.png
│   └── model_rf_feature_importance.png
│
├── requirements.txt         # Dependencies
├── .gitignore
├── Checklist Submit.md
├── LICENSE
├── Laporan Proyek Machine Learning.pdf
└── README.md
```

10.2 Environtment & Dependencies

- **Python Version:** 3.12.12
- **Main Libraries & Versions:**

```
Python Version: 3.12.12
------------------------------
ucimlrepo==0.0.7
numpy==2.0.2
pandas==2.2.2
scikit-learn==1.6.1
matplotlib==3.10.0
seaborn==0.13.2
tensorflow==2.19.0
joblib==1.5.2
xgboost==3.1.2
```