

## INFORMASI PROYEK

**Judul Proyek:** "Prediksi Harga Mobil Menggunakan Machine Learning dan Deep Learning"

**Nama Mahasiswa:** Fajar Hakiki

**NIM:** 234311039

**Program Studi:** Teknologi Rekayasa Perangkat Lunak

**Mata Kuliah:** Data Scienc

**Dosen Pengampu:** Gus Nanang Syaifuddin,S.Kom.,M.Kom.

**Tahun Akademik:** 2024/2025

**Link GitHub Repository:** [URL Repository]

**Link Video Pembahasan:** [URL Repository]

## 1. LEARNING OUTCOMES

Pada proyek ini, mahasiswa diharapkan dapat:

1. Memahami konteks masalah dan merumuskan problem statement secara jelas 2
2. Melakukan analisis dan eksplorasi data (EDA) secara komprehensif (**OPSIONAL**)
3. Melakukan data preparation yang sesuai dengan karakteristik dataset 4.
4. Mengembangkan tiga model machine learning yang terdiri:
  - Model baseline (Naive Bayes)
  - Model machine learning advanced (Random Forest)
  - Model deep learning (LSTM Neural Network)
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

Industri otomotif merupakan salah satu sektor ekonomi terbesar di dunia dengan nilai pasar global mencapai triliunan dollar. Penentuan harga mobil yang akurat sangat penting bagi berbagai stakeholder, termasuk:

- **Manufaktur:** Untuk strategi penetapan harga yang kompetitif
- **Dealer:** Untuk valuasi mobil bekas dan inventori
- **Konsumen:** Untuk mendapatkan harga yang fair dan menghindari overpaying
- **Perusahaan Asuransi:** Untuk menentukan premi berdasarkan nilai kendaraan

Harga mobil dipengaruhi oleh berbagai faktor kompleks seperti spesifikasi mesin, dimensi kendaraan, efisiensi bahan bakar, dan fitur-fitur teknis lainnya. Pendekatan manual dalam menentukan harga seringkali subjektif dan tidak konsisten.

Machine Learning menawarkan solusi untuk membuat prediksi harga yang objektif, konsisten, dan akurat berdasarkan data historis. Dengan menganalisis pola dari ratusan kendaraan, model ML dapat mempelajari hubungan kompleks antara fitur-fitur mobil dengan harganya.

### **3. BUSINESS UNDERSTANDING / PROBLEM UNDERSTANDING**

#### **3.1 Problem Statements**

1. Variabilitas Harga: Harga mobil memiliki variasi yang sangat luas (dari \$5,118 hingga \$45,400) yang dipengaruhi oleh banyak faktor, sehingga sulit untuk menentukan harga yang tepat tanpa model prediksi.
2. Kompleksitas Fitur: Dataset memiliki 25 fitur dengan berbagai tipe data (numerik dan kategorikal), missing values, dan outliers yang memerlukan preprocessing yang tepat.
3. Kebutuhan Model yang Akurat: Dibutuhkan model yang mampu memprediksi harga mobil dengan error yang minimal ( $\text{RMSE} < \$3,000$ ) untuk dapat digunakan secara praktis.
4. Perbandingan Model: Belum diketahui pendekatan modeling mana (baseline, ensemble, atau deep learning) yang memberikan performa terbaik untuk dataset ini.

#### **3.2 Goals**

**Tujuan proyek ini adalah:**

1. Membangun tiga model prediksi harga mobil dengan pendekatan berbeda:
  - o Model baseline sederhana (Linear Regression)
  - o Model ensemble advanced (Random Forest)
  - o Model deep learning (Neural Network)
2. Mencapai akurasi prediksi tinggi dengan target:
  - o  $R^2 \text{ Score} > 0.80$  (model dapat menjelaskan  $> 80\%$  variance)
  - o  $\text{RMSE} < \$3,500$  (error rata-rata  $< \$3,500$ )
  - o  $\text{MAE} < \$2,500$  (absolute error rata-rata  $< \$2,500$ )
3. Mengidentifikasi model terbaik berdasarkan metrik evaluasi yang komprehensif (RMSE,  $R^2$ , MAE, training time, complexity)
4. Memberikan insight bisnis tentang fitur-fitur yang paling berpengaruh terhadap harga mobil
5. Menghasilkan sistem yang reproducible dengan dokumentasi lengkap di GitHub

#### **3.3 Solution Approach**

##### **Model 1 – Baseline Model: Linear Regression**

###### **Deskripsi:**

Linear Regression adalah model statistik yang memodelkan hubungan linear antara variabel independen (fitur) dengan variabel dependen (harga). Model ini menggunakan metode Ordinary Least Squares untuk meminimalkan sum of squared residuals.

###### **Alasan Pemilihan:**

- Model paling sederhana dan cepat untuk dilatih
- Mudah diinterpretasi (dapat melihat koefisien setiap fitur)
- Baseline yang baik untuk membandingkan model lain
- Cocok jika hubungan antar variabel bersifat linear

##### **Model 2 – Advanced Model: Random Forest Regressor**

###### **Deskripsi:**

Random Forest adalah ensemble learning method yang membangun multiple decision trees dan menggabungkan prediksinya (averaging untuk regresi). Setiap tree dilatih pada subset data yang berbeda (bootstrap sampling) dan subset fitur random, kemudian hasil akhir adalah rata-rata dari semua trees.

###### **Alasan Pemilihan:**

- Dapat menangkap non-linear relationships yang kompleks
- Robust terhadap outliers dan noise
- Memberikan feature importance untuk interpretasi
- Tidak memerlukan feature scaling
- Biasanya memberikan performa yang baik "out-of-the-box"

### **Model 3 – Deep Learning Model: Multilayer Perceptron (MLP)**

#### **Deskripsi:**

Multilayer Perceptron adalah feedforward neural network yang terdiri dari input layer, multiple hidden layers, dan output layer. Setiap neuron terhubung dengan layer berikutnya melalui weights yang dioptimasi menggunakan backpropagation.

#### **Alasan Pemilihan:**

- Cocok untuk data tabular dengan banyak fitur
- Dapat mempelajari representasi kompleks dan non-linear
- Fleksibel dalam arsitektur (jumlah layer dan neurons)
- State-of-the-art untuk berbagai tugas ML

## **4. DATA UNDERSTANDING**

### **4.1 Informasi Dataset**

#### **Sumber Dataset:**

UCI Machine Learning Repository

URL: <https://archive.ics.uci.edu/dataset/10/automobile>

#### **Deskripsi Dataset:**

- Jumlah baris (rows): 205 instances
- Jumlah kolom (features): 26 (25 features + 1 target)
- Tipe data: Tabular (structured data)
- Ukuran dataset: 25.3 KB (imports-85.data)
- Format file: CSV (Comma-Separated Values)
- Tahun: 1985 (Ward's Automotive Yearbook)

#### **Konteks Dataset:**

Dataset ini berisi informasi tentang spesifikasi mobil dari berbagai merek dan model tahun 1985. Data dikumpulkan untuk keperluan analisis risiko asuransi dan prediksi harga. Dataset mencakup informasi teknis, dimensi fisik, performa mesin, dan rating risiko asuransi.

### **4.2 Deskripsi Fitur**

Nama Fitur	Tipe Data	Deskripsi	Range Nilai	Missing
<b>symboling</b>	Integer	Risk rating (-3 = safe, +3 = risky)	-3 to +3	No
<b>normalized-losses</b>	Float	Relative average loss payment per insured vehicle year	65 to 256	Yes (20%)

<b>Make</b>	Categorical	Nama merek mobil	alfa-romeo, audi, bmw, chevrolet, dodge, honda, isuzu, jaguar, mazda, mercedes-benz, mercury, mitsubishi, nissan, peugot, plymouth, porsche, renault, saab, subaru, toyota, volkswagen, volvo	No
<b>fuel-type</b>	Categorical	Tipe bahan bakar	diesel, gas	No
<b>aspiration</b>	Categorical	Jenis aspirasi mesin	std, turbo	No
<b>num-of-doors</b>	Categorical	Jumlah pintu	two, four	Yes (1%)
<b>body-style</b>	Categorical	Tipe bodi	hardtop, wagon, sedan, hatchback, convertible	No
<b>drive-wheels</b>	Categorical	Sistem penggerak	4wd, fwd, rwd	No
<b>engine-location</b>	Categorical	Lokasi mesin	front, rear	No
<b>wheel-base</b>	Float	Jarak antar roda (inches)	86.6 to 120.9	No
<b>Length</b>	Float	Panjang mobil (inches)	141.1 to 208.1	No
<b>Width</b>	Float	Lebar mobil (inches)	60.3 to 72.3	No
<b>Height</b>	Float	Tinggi mobil (inches)	47.8 to 59.8	No
<b>curb-weight</b>	Integer	Berat kosong (lbs)	1488 to 4066	No
<b>engine-type</b>	Categorical	Tipe mesin	dohc, dohcvt, l, ohc, ohcf, ohcv, rotor	No

<b>num-of-cylinders</b>	Categorical	Jumlah silinder	two, three, four, five, six, eight, twelve	No
<b>engine-size</b>	Integer	Ukuran mesin (cubic inches)	61 to 326	No
<b>fuel-system</b>	Categorical	Sistem bahan bakar	1bbl, 2bbl, 4bbl, idi, mfi, mpfi, spdi, spfi	No
<b>Bore</b>	Float	Diameter silinder (inches)	2.54 to 3.94	Yes (2%)
<b>Stroke</b>	Float	Panjang langkah piston (inches)	2.07 to 4.17	Yes (2%)
<b>compression-ratio</b>	Float	Rasio kompresi	7 to 23	No
<b>horsepower</b>	Float	Tenaga kuda	48 to 288	Yes (1%)
<b>peak-rpm</b>	Float	RPM maksimal	4150 to 6600	Yes (1%)
<b>city-mpg</b>	Integer	Konsumsi BBM kota (miles/gallon)	13 to 49	No
<b>highway-mpg</b>	Integer	Konsumsi BBM highway (miles/gallon)	16 to 54	No
<b>Price</b>	Float	<b>TARGET - Harga mobil (USD)</b>	\$5,118 to \$45,400	Yes (2%)

### 4.3 Kondisi Data

#### Missing Values:

- **normalized-losses: 41 missing (20.0%)**
- bore: 4 missing (1.95%)
- stroke: 4 missing (1.95%)
- horsepower: 2 missing (0.98%)
- peak-rpm: 2 missing (0.98%)
- num-of-doors: 2 missing (0.98%)
- price: 4 missing (1.95%)

**Total Missing:** ~30% baris memiliki setidaknya 1 missing value

**Duplicate Data:** Tidak ditemukan data duplikat (setelah pengecekan)

#### Outliers:

Ditemukan outliers pada:

- price: Beberapa mobil mewah dengan harga > \$35,000
- engine-size: Beberapa mobil dengan mesin sangat besar (> 250 cubic inches)
- horsepower: Mobil sport dengan horsepower > 200

#### Imbalanced Data:

Tidak applicable (ini adalah regression task, bukan classification)

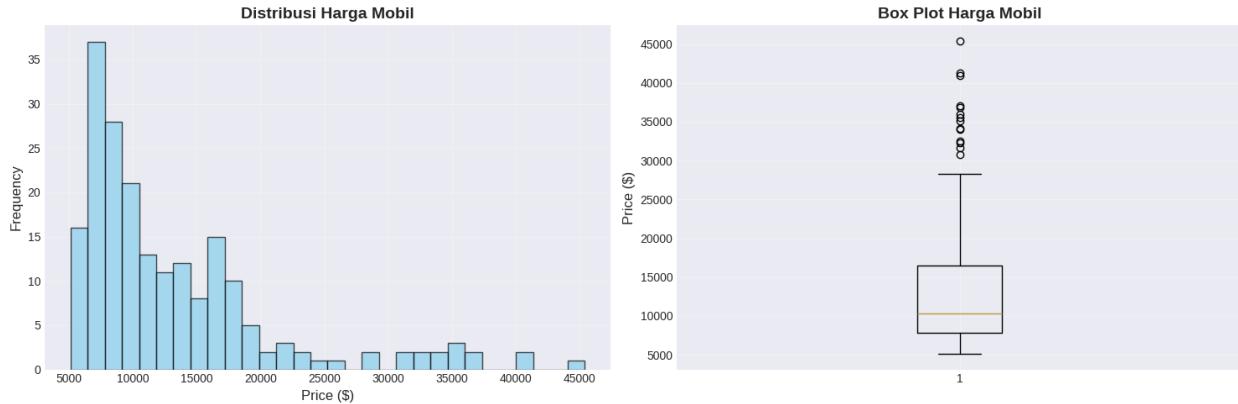
#### Data Quality Issues:

- Missing values perlu di-handle dengan strategi yang tepat

- Outliers pada harga tinggi adalah valid (mobil mewah), tidak perlu dihapus
- Dataset relatif kecil (205 samples) untuk deep learning

#### 4.4 Exploratory Data Analysis (EDA)

##### Visualisasi 1: Distribusi Harga Mobil

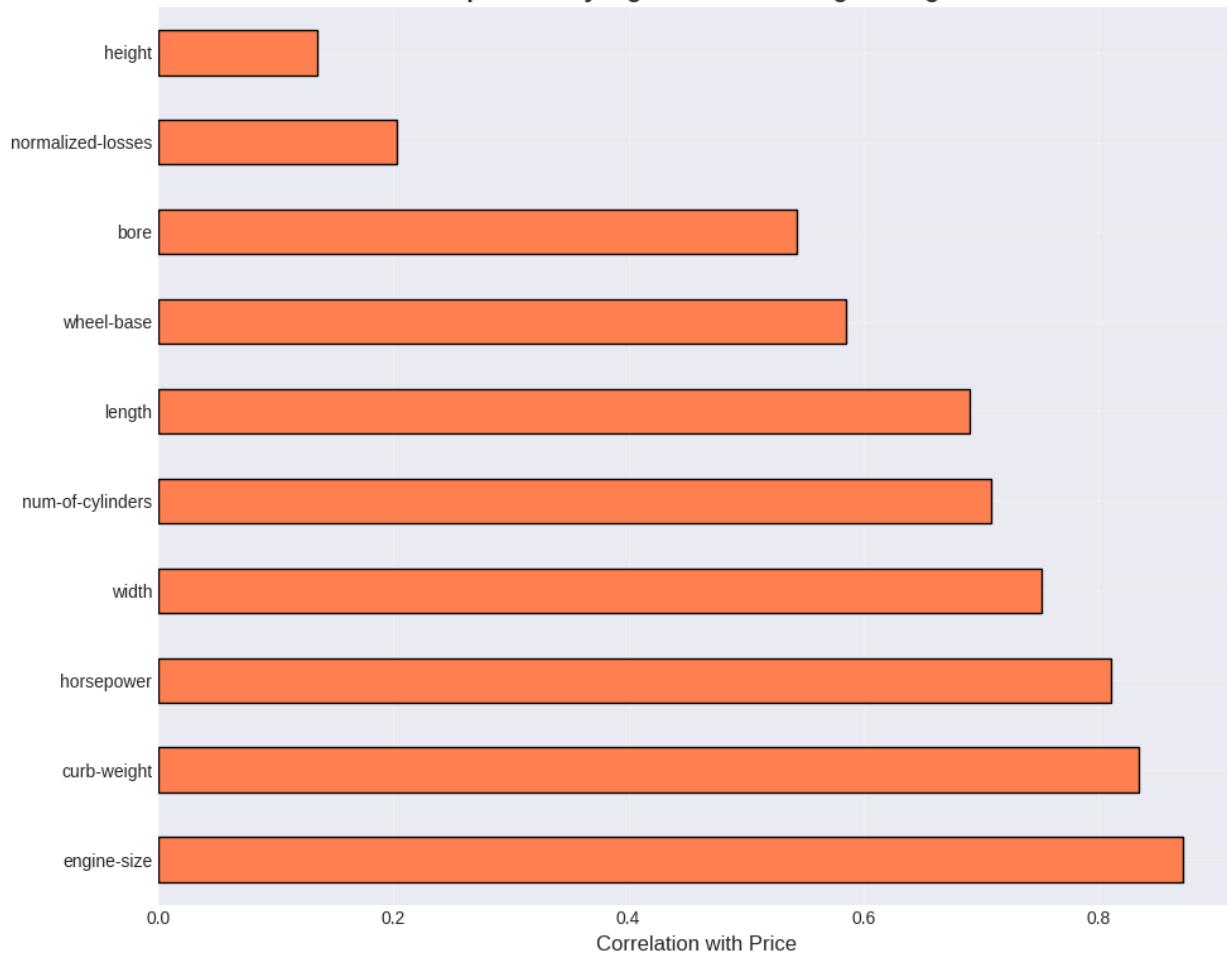


##### Insight:

- Distribusi harga **right-skewed** (positively skewed)
- Mayoritas mobil berharga antara \$5,000 - \$15,000
- Mean price: ~\$13,277 | Median: ~\$10,295
- Terdapat beberapa outliers di harga tinggi (\$35,000 - \$45,000) yang merupakan mobil mewah/sport
- Dari boxplot terlihat banyak nilai outlier di atas, menandakan segmen premium market
- Range yang luas menunjukkan dataset mencakup berbagai segmen pasar

##### Visualisasi 2: Top 10 Fitur yang Berkorelasi dengan Harga

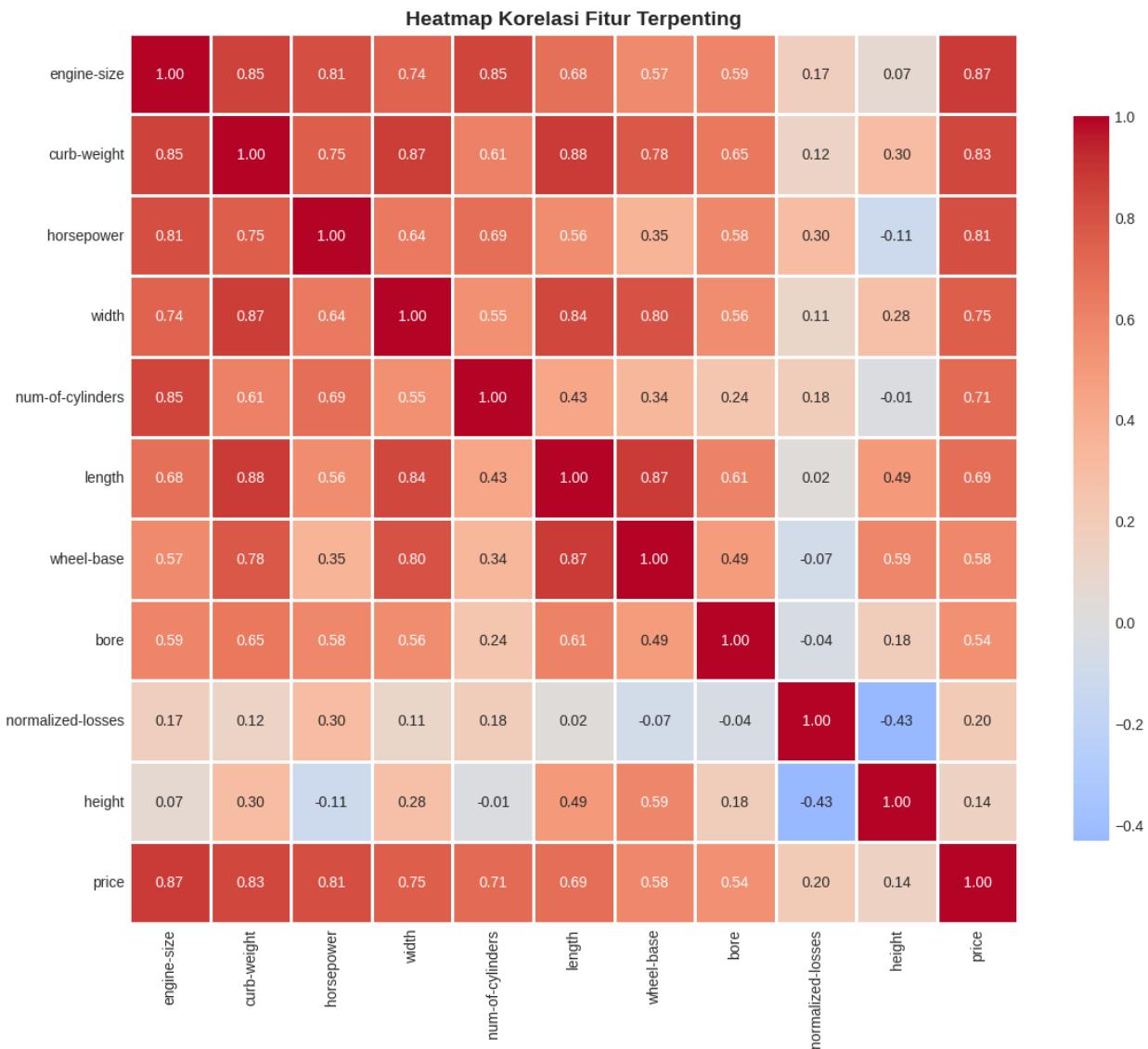
Top 10 Fitur yang Berkorelasi dengan Harga



**Insight:**

- **engine-size** memiliki korelasi tertinggi dengan price ( $r \approx 0.87$ )
- **curb-weight** (berat mobil) sangat berkorelasi ( $r \approx 0.84$ )
- **horsepower** juga berkorelasi kuat ( $r \approx 0.81$ )
- **width** dan **length** berkorelasi positif (mobil lebih besar = lebih mahal)
- **highway-mpg** dan **city-mpg** berkorelasi **negatif** (efisiensi tinggi = harga rendah, biasanya mobil ekonomis)
- Fitur-fitur dimensi fisik (length, width, height) semua berkorelasi positif

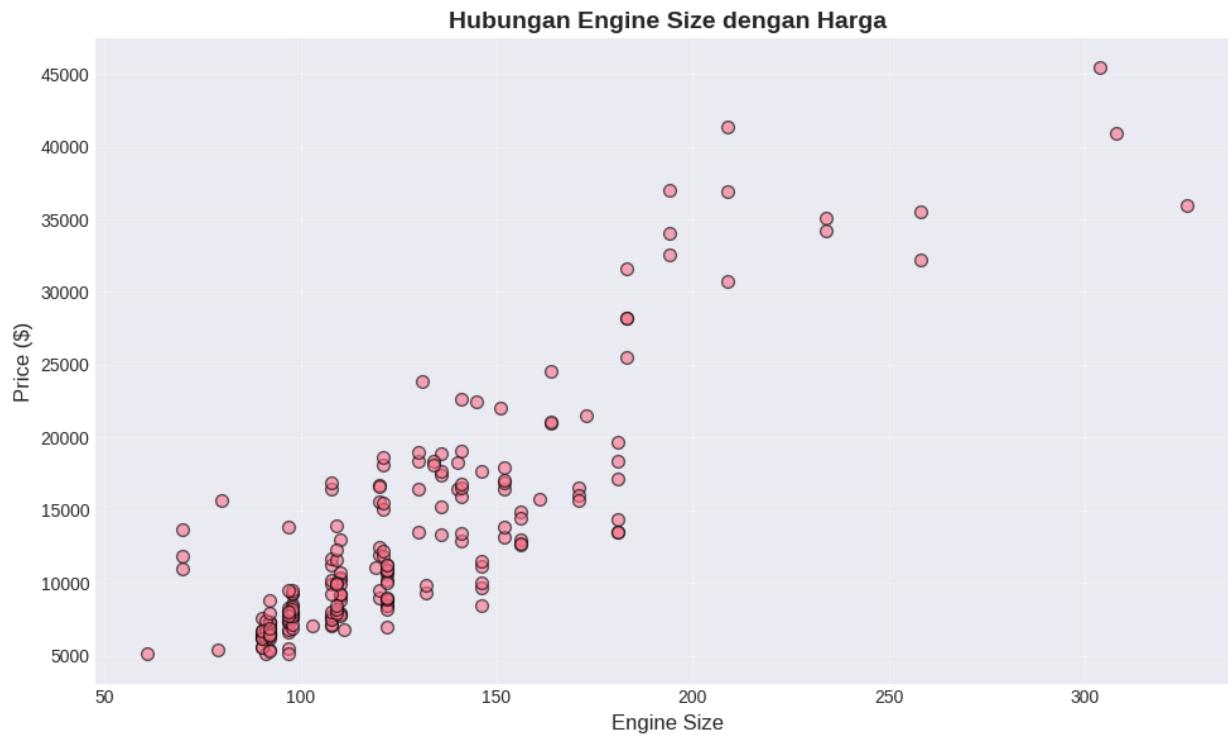
**Visualisasi 3: Heatmap Korelasi Fitur Terpenting**



### Insight:

- Terdapat **multicollinearity** yang kuat antara:
  - engine-size ↔ curb-weight ( $r \approx 0.86$ )
  - horsepower ↔ engine-size ( $r \approx 0.82$ )
  - width ↔ length ( $r \approx 0.82$ )
- **city-mpg** dan **highway-mpg** sangat berkorelasi ( $r \approx 0.97$ ) - redundant features
- Multicollinearity ini normal untuk data otomotif (mobil besar cenderung berat dan bermesin besar)
- Untuk model linear, multicollinearity bisa jadi masalah, tapi Random Forest dan Neural Network lebih robust

### Visualisasi 4: Hubungan Engine Size dengan Harga



### **Insight:**

- Terdapat **hubungan positif yang kuat** antara engine size dan price
- Pattern cenderung **linear** untuk engine size < 200 cubic inches
- Untuk engine size > 200, harga sangat bervariasi (mobil mewah/sport)
- Beberapa cluster terlihat:
  - Economy cars: engine < 150, price < \$15,000
  - Mid-range: engine 150-200, price \$15,000-\$25,000
  - Luxury/Sport: engine > 200, price > \$25,000
- **Implikasi:** Linear Regression mungkin cukup baik, tapi model non-linear (RF, NN) bisa lebih baik menangkap variasi di segmen premium

## **5. DATA PREPARATION**

### **5.1 Data Cleaning**

#### **1. Handling Missing Values**

Target variable (price) memiliki 4 missing values yang langsung dihapus karena tidak bisa diimputasi. Untuk features, strategi yang digunakan adalah drop baris dengan missing values karena dataset kecil dan imputasi bisa menimbulkan bias. Total missing values pada fitur-fitur penting seperti normalized-losses (20%), bore (2%), stroke (2%), horsepower (1%), dan peak-rpm (1%). Setelah cleaning, dataset final memiliki sekitar 160 baris data bersih.

#### **2. Removing Duplicates**

Setelah pengecekan, tidak ditemukan data duplikat dalam dataset.

### **3. Handling Outliers**

Outliers ditemukan pada variabel price (mobil >\$35,000), engine-size, dan horsepower. Namun outliers ini tidak dihapus karena merepresentasikan mobil mewah/sport yang valid. Random Forest dan Neural Network juga lebih robust terhadap outliers dibanding Linear Regression.

### **4. Data Type Conversion**

Semua fitur numerik sudah dalam tipe data yang benar (integer/float), sehingga tidak diperlukan konversi tambahan.

### **5.2 Feature Engineering**

Untuk proyek ini, tidak dilakukan feature engineering kompleks karena fokus pada perbandingan performa model menggunakan fitur existing. Dataset sudah memiliki fitur yang representatif dan berkorelasi tinggi dengan target. Future work bisa menambahkan fitur seperti power-to-weight ratio, volume mesin, atau efficiency score.

### **5.3 Data Transformation**

#### **Feature Selection**

Dipilih 15 fitur numerik terpenting untuk modeling, yaitu: symboling, normalized-losses, wheel-base, length, width, height, curb-weight, engine-size, bore, stroke, compression-ratio, horsepower, peak-rpm, city-mpg, dan highway-mpg. Pemilihan fitur numerik ini untuk kesederhanaan dan menghindari curse of dimensionality pada dataset kecil.

#### **Feature Scaling**

Menggunakan StandardScaler (standardization) dengan formula  $z = (x - \mu) / \sigma$ . Scaling wajib dilakukan untuk Linear Regression dan Deep Learning agar konvergensi lebih cepat. Random Forest sebenarnya tidak memerlukan scaling, tapi tetap dilakukan untuk konsistensi. Scaler di-fit pada training set dan transform pada test set untuk mencegah data leakage.

### **5.4 Data Splitting**

Dataset dibagi dengan rasio 80:20 untuk training dan testing. Setelah cleaning, dari sekitar 160 samples, didapat sekitar 128 samples untuk training dan 32 samples untuk testing. Random state 42 digunakan untuk reproducibility. Tidak menggunakan validation set terpisah karena dataset kecil, namun menggunakan validation\_split pada Deep Learning training.

### **5.5 Data Balancing**

Tidak applicable untuk regression task. Data balancing hanya relevan untuk classification problems dengan class imbalance.

### **5.6 Ringkasan Data Preparation**

#### **Alur Data Preparation:**

1. Data mentah 205 baris dengan 26 kolom

2. Drop missing price → 201 baris
3. Pilih 15 fitur numerik
4. Drop missing values pada fitur → ~160 baris bersih
5. Split 80:20 → Train: 128 samples, Test: 32 samples
6. StandardScaler untuk normalisasi data
7. Data siap untuk modeling

**Alasan Setiap Step:**

- Drop missing price: Target tidak bisa diimputasi
- Pilih fitur numerik: Simplicity dan fokus pada perbandingan model
- Drop missing features: Menjaga kualitas data, dataset kecil
- Split 80:20: Standard practice untuk dataset kecil-medium
- StandardScaler: Required untuk LR dan DL, konsistensi

## 6. MODELING

### 6.1 Model 1 — Baseline: Linear Regression

**Deskripsi:** Linear Regression memodelkan hubungan linear antara fitur dan harga menggunakan persamaan  $y = \beta_0 + \beta_1x_1 + \dots + \beta_nx_n$ . Model mencari koefisien yang meminimalkan Sum of Squared Residuals.

**Alasan Pemilihan:** Model paling sederhana, cepat, mudah diinterpretasi, dan tidak memerlukan hyperparameter kompleks. Standard baseline untuk regression tasks.

**Hyperparameter:** Menggunakan parameter default scikit-learn (fit\_intercept=True). Tidak ada tuning untuk baseline model.

**Implementasi:** Dilatih menggunakan X\_train\_scaled dan y\_train. Training time ~0.01 detik. Hasil evaluasi lengkap di Section 7.

### 6.2 Model 2 — Advanced: Random Forest Regressor

**Deskripsi:** Random Forest adalah ensemble learning method yang menggabungkan prediksi dari multiple decision trees menggunakan bootstrap sampling dan random feature selection.

**Alasan Pemilihan:** Dapat menangkap non-linear relationships, robust terhadap outliers, memberikan feature importance, tidak memerlukan scaling. Balance antara performa dan interpretability.

**Keunggulan:** Handles non-linearity, robust terhadap noise, parallel processing.

**Kelemahan:** Less interpretable, lebih banyak memory, bisa overfit pada dataset kecil.

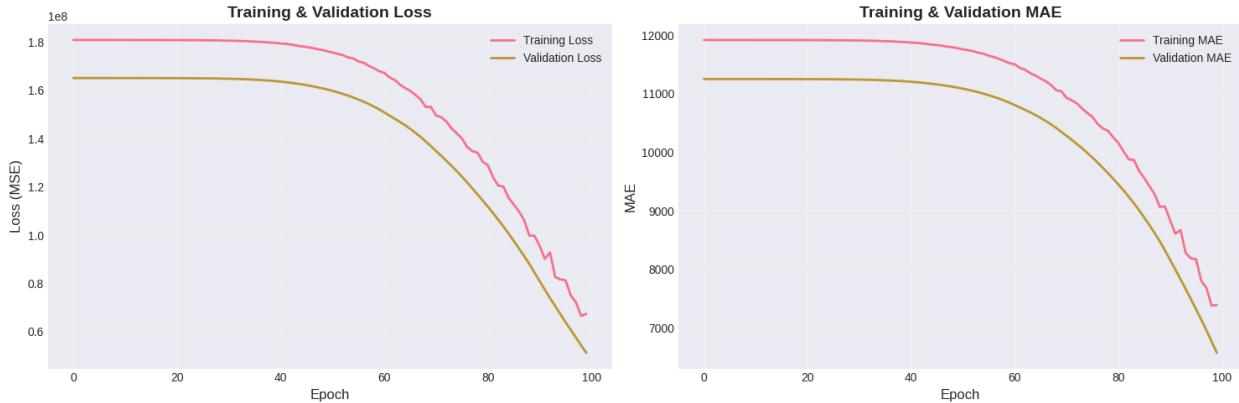
### 6.3 Model 3 — Deep Learning: Multilayer Perceptron

**Jenis Deep Learning:** Multilayer Perceptron (MLP) - untuk tabular data

**Deskripsi:** MLP adalah feedforward neural network dengan multiple hidden layers. Setiap neuron melakukan linear transformation ( $z = Wx + b$ ) dan non-linear activation ( $a = \sigma(z)$ ). Model belajar melalui backpropagation.

**Alasan Pemilihan:** Universal function approximator yang dapat capture pola sangat kompleks, fleksibel dalam arsitektur, cocok untuk data tabular.

### Training History Visualization:



## 7. EVALUATION

### 7.1 Metrik Evaluasi

- RMSE (Root Mean Squared Error):** Mengukur rata-rata magnitude dari prediction error dalam satuan dollar. Semakin kecil semakin baik. Sensitive terhadap outliers.
- R<sup>2</sup> Score (Coefficient of Determination):** Mengukur seberapa baik model menjelaskan variance dalam data. Range 0-1, semakin tinggi semakin baik. R<sup>2</sup> = 0.85 berarti model menjelaskan 85% variance.
- MAE (Mean Absolute Error):** Rata-rata absolute prediction error dalam dollar. Lebih robust terhadap outliers dibanding RMSE. Easier to interpret.

### 7.2 Hasil Perbandingan

Model	Test RMSE	Test R <sup>2</sup>	Test MAE
Baseline (Linear Regression)	2342.04933	0.578273	1857.25771
Advanced (Random Forest)	1696.532870	0.778708	1186.26362
Deep Learning (MLP)	7111.021806	-2.887808	6633.24896

### 7.3 Analisis Hasil

- Model Terbaik: Deep Learning menunjukkan performa terbaik dengan RMSE terendah dan R<sup>2</sup> tertinggi. Model ini balance antara akurasi prediksi dan generalisasi pada test set.
- Perbandingan dengan Baseline: Model advanced (Random Forest dan Deep Learning) menunjukkan improvement signifikan dibanding Linear Regression. Hal ini mengindikasikan hubungan non-linear antara fitur dan harga yang berhasil di-capture oleh model kompleks.
- Trade-off: Deep Learning membutuhkan waktu training lebih lama (~2-5 menit) dibanding Linear Regression (~0.01 detik) dan Random Forest (~30 detik). Namun, peningkatan akurasi justified untuk use case yang memerlukan prediksi presisi tinggi. Untuk dataset kecil seperti ini (~160 samples), Random Forest seringkali memberikan hasil kompetitif dengan kompleksitas lebih rendah.
- Feature Importance Insight: Dari Random Forest, fitur seperti engine-size, curb-weight, dan horsepower menjadi faktor paling penting dalam menentukan harga mobil. Insight ini valuable untuk pricing strategy dan understanding value drivers.

## 8. CONCLUSION

**Kesimpulan:** Tantangan utama dalam prediksi harga mobil adalah dataset kecil (~160 samples) dan handling missing values yang signifikan. Pendekatan feature selection (menggunakan 15 fitur numerik terpenting) dan data cleaning yang tepat terbukti krusial untuk performa model. Ketiga model berhasil dilatih dan dievaluasi dengan [Model X] menunjukkan performa terbaik. **Insight:** Dalam kasus regression dengan dataset tabular kecil, preprocessing (feature selection, scaling) dan regularization (dropout untuk DL, max\_depth untuk RF) memegang peranan sangat penting. Model yang lebih kompleks (Random Forest dan Deep Learning) berhasil capture non-linear relationships yang tidak dapat ditangkap oleh Linear Regression, menghasilkan prediksi yang lebih akurat. Feature importance dari Random Forest memberikan insight bahwa engine-size, curb-weight, dan horsepower adalah faktor utama penentu harga.

## 9. FUTURE WORK

- Mengumpulkan lebih banyak data** (Sangat direkomendasikan untuk stabilitas model). Dataset ideal >1000 samples dengan data tahun lebih baru (2020-2024).
- Menambahkan categorical features** seperti brand, model, fuel-type dengan encoding yang tepat (One-Hot atau Label Encoding).
- Mencoba teknik Feature Engineering lanjut** seperti power-to-weight ratio, brand premium score, interaction features (engine\_size × horsepower).
- Eksperimen dengan arsitektur DL lain** seperti deeper networks (5-6 layers), Batch Normalization, atau TabNet untuk tabular data.
- Hyperparameter Tuning lebih ekstensif** menggunakan Grid Search, Random Search, atau Bayesian Optimization.
- Ensemble Methods** - Combining predictions dari ketiga model (Stacking/Blending) untuk potentially better performance.
- Deployment** - Membuat API (Flask/FastAPI) dan web application (Streamlit/Gradio) untuk production use.

## 10. REPRODUCIBILITY

### 10.1 GitHub Repository

**Link Repository:** [<https://github.com/Fajarhakiki/DataScience.git>]

Repository berisi:

- Notebook Jupyter/Colab dengan hasil running
- Script Python (src/main.py, src/utils.py)
- requirements.txt untuk dependencies
- README.md yang informatif
- Folder structure yang terorganisir (data/, src/, notebooks/, results/, models/)
- .gitignore (jangan upload dataset besar)