Tutorial Lengkap: Penerapan ANN untuk Prediksi Nilai Ekspor Migas dengan Flask

## Tugas 4: Membangun Aplikasi Web Prediksi Ekspor Migas menggunakan ANN dan Flask

### Persiapan Awal

1. \*\*Buat folder proyek baru\*\*:

```bash

mkdir tugas4\_ekspor\_migas

cd tugas4\_ekspor\_migas

```

2. \*\*Buat virtual environment\*\*:

```bash

python -m venv venv

source venv/bin/activate # Untuk Linux/Mac

venv\Scripts\activate # Untuk Windows

```

3. \*\*Instal library yang diperlukan\*\*:

```bash

pip install flask tensorflow pandas numpy scikit-learn matplotlib

```

### Struktur Folder Proyek

```

tugas4\_ekspor\_migas/

├── app.py # File utama Flask

├── model.py # Script untuk membangun model ANN

├── model\_ekspor.h5 # Model ANN yang sudah dilatih

├── templates/

│ ├── index.html # Halaman utama

│ └── hasil.html # Halaman hasil prediksi

├── static/

│ ├── style.css # File CSS

│ └── plot.png # Visualisasi data

└── data\_ekspor.csv # Dataset ekspor migas

```

### Langkah 1: Persiapan Dataset

Buat file `data\_ekspor.csv` dengan isi berikut (berdasarkan data dari tabel):

```csv

Tahun,Bulan,Nilai\_Ekspor\_Migas

2024,1,1285.2

2024,2,1127.9

2024,3,1452.9

2025,1,1127.9

2025,2,1216.3

2025,3,1452.9

```

### Langkah 2: Membangun Model ANN

Buat file `model.py`:

```python

import numpy as np

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import MinMaxScaler

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

import matplotlib.pyplot as plt

# 1. Load dataset

df = pd.read\_csv('data\_ekspor.csv')

# 2. Preprocessing

# Gabungkan tahun dan bulan menjadi satu fitur waktu

df['Waktu'] = df['Tahun'] + (df['Bulan']-1)/12

# Normalisasi data

scaler = MinMaxScaler()

df['Waktu\_norm'] = scaler.fit\_transform(df[['Waktu']])

df['Ekspor\_norm'] = scaler.fit\_transform(df[['Nilai\_Ekspor\_Migas']])

X = df['Waktu\_norm'].values.reshape(-1, 1)

y = df['Ekspor\_norm'].values

# 3. Split data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# 4. Bangun model ANN

model = Sequential([

Dense(16, activation='relu', input\_shape=(1,)),

Dense(16, activation='relu'),

Dense(1, activation='linear')

])

model.compile(optimizer='adam', loss='mse', metrics=['mae'])

# 5. Latih model

history = model.fit(X\_train, y\_train, epochs=200, validation\_data=(X\_test, y\_test), verbose=0)

# 6. Evaluasi model

loss, mae = model.evaluate(X\_test, y\_test)

print(f"Test MAE: {mae:.4f}")

# 7. Simpan model

model.save('model\_ekspor.h5')

# 8. Visualisasi hasil training

plt.plot(history.history['loss'], label='Training Loss')

plt.plot(history.history['val\_loss'], label='Validation Loss')

plt.xlabel('Epoch')

plt.ylabel('Loss')

plt.legend()

plt.savefig('static/training\_plot.png')

plt.close()

# 9. Prediksi untuk visualisasi

X\_future = np.array([df['Waktu'].max() + i/12 for i in range(1, 13)]).reshape(-1, 1)

X\_future\_norm = scaler.transform(X\_future)

y\_future\_norm = model.predict(X\_future\_norm)

y\_future = scaler.inverse\_transform(y\_future\_norm)

# Gabungkan data aktual dan prediksi

df\_plot = df[['Waktu', 'Nilai\_Ekspor\_Migas']].copy()

df\_plot['Type'] = 'Aktual'

df\_pred = pd.DataFrame({'Waktu': X\_future.flatten(), 'Nilai\_Ekspor\_Migas': y\_future.flatten(), 'Type': 'Prediksi'})

df\_combined = pd.concat([df\_plot, df\_pred])

# Visualisasi

plt.figure(figsize=(10, 6))

plt.plot(df\_combined[df\_combined['Type']=='Aktual']['Waktu'],

df\_combined[df\_combined['Type']=='Aktual']['Nilai\_Ekspor\_Migas'],

'bo-', label='Data Aktual')

plt.plot(df\_combined[df\_combined['Type']=='Prediksi']['Waktu'],

df\_combined[df\_combined['Type']=='Prediksi']['Nilai\_Ekspor\_Migas'],

'ro--', label='Prediksi')

plt.xlabel('Waktu (Tahun)')

plt.ylabel('Nilai Ekspor Migas (juta US$)')

plt.title('Prediksi Nilai Ekspor Migas Indonesia')

plt.legend()

plt.grid(True)

plt.savefig('static/prediction\_plot.png')

plt.close()

```

Jalankan script di atas untuk melatih model:

```bash

python model.py

```

### Langkah 3: Membangun Aplikasi Flask

Buat file `app.py`:

```python

from flask import Flask, render\_template, request

import numpy as np

import pandas as pd

from tensorflow.keras.models import load\_model

from sklearn.preprocessing import MinMaxScaler

import joblib

app = Flask(\_\_name\_\_)

# Load model dan scaler

model = load\_model('model\_ekspor.h5')

scaler = joblib.load('scaler.save') # Simpan scaler saat preprocessing

@app.route('/')

def home():

return render\_template('index.html')

@app.route('/predict', methods=['POST'])

def predict():

# Ambil input dari form

tahun = float(request.form['tahun'])

bulan = float(request.form['bulan'])

# Preprocessing input

waktu = tahun + (bulan-1)/12

waktu\_norm = scaler.transform([[waktu]])

# Prediksi

pred\_norm = model.predict(waktu\_norm)

prediksi = scaler.inverse\_transform(pred\_norm)[0][0]

return render\_template('hasil.html',

tahun=int(tahun),

bulan=int(bulan),

prediksi=f"{prediksi:,.2f}")

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

```

### Langkah 4: Membuat Template HTML

Buat folder `templates` dan file `index.html`:

```html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Prediksi Ekspor Migas Indonesia</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

<link rel="stylesheet" href="/static/style.css">

</head>

<body>

<div class="container mt-5">

<h1 class="text-center mb-4">Prediksi Nilai Ekspor Migas Indonesia</h1>

<div class="row">

<div class="col-md-6">

<div class="card">

<div class="card-header">

<h5>Masukkan Data</h5>

</div>

<div class="card-body">

<form action="/predict" method="post">

<div class="mb-3">

<label for="tahun" class="form-label">Tahun</label>

<input type="number" class="form-control" id="tahun" name="tahun" min="2024" max="2030" required>

</div>

<div class="mb-3">

<label for="bulan" class="form-label">Bulan</label>

<select class="form-select" id="bulan" name="bulan" required>

<option value="1">Januari</option>

<option value="2">Februari</option>

<option value="3">Maret</option>

<option value="4">April</option>

<option value="5">Mei</option>

<option value="6">Juni</option>

<option value="7">Juli</option>

<option value="8">Agustus</option>

<option value="9">September</option>

<option value="10">Oktober</option>

<option value="11">November</option>

<option value="12">Desember</option>

</select>

</div>

<button type="submit" class="btn btn-primary">Prediksi</button>

</form>

</div>

</div>

</div>

<div class="col-md-6">

<div class="card">

<div class="card-header">

<h5>Visualisasi Data</h5>

</div>

<div class="card-body">

<img src="/static/prediction\_plot.png" alt="Grafik Prediksi" class="img-fluid">

</div>

</div>

</div>

</div>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

```

Buat file `hasil.html`:

```html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Hasil Prediksi</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<div class="container mt-5">

<div class="card">

<div class="card-header bg-primary text-white">

<h3>Hasil Prediksi Nilai Ekspor Migas</h3>

</div>

<div class="card-body">

<h4>Prediksi untuk Bulan {{ bulan }}/{{ tahun }}:</h4>

<div class="alert alert-success mt-3">

<h5>Nilai Ekspor Migas Diprediksi: <strong>{{ prediksi }} juta US$</strong></h5>

</div>

<a href="/" class="btn btn-primary">Kembali</a>

</div>

</div>

</div>

</body>

</html>

```

### Langkah 5: Menjalankan Aplikasi

1. Jalankan aplikasi Flask:

```bash

python app.py

```

2. Buka browser dan akses `http://localhost:5000`

### Langkah 6: Deployment ke Heroku (Opsional)

1. Buat file `requirements.txt`:

```bash

pip freeze > requirements.txt

```

2. Buat file `Procfile`:

```

web: gunicorn app:app

```

3. Ikuti panduan deployment Heroku untuk mengunggah aplikasi Anda.

## Kesimpulan

Dalam tutorial ini, kita telah membangun:

1. Model ANN untuk memprediksi nilai ekspor migas Indonesia

2. Aplikasi web berbasis Flask untuk memvisualisasikan data dan hasil prediksi

3. Antarmuka pengguna yang memungkinkan input tahun dan bulan untuk prediksi

Aplikasi ini dapat dikembangkan lebih lanjut dengan:

- Menambahkan lebih banyak data historis

- Meningkatkan arsitektur model ANN

- Menambahkan fitur ekspor hasil prediksi

- Mengintegrasikan dengan database untuk menyimpan hasil prediksi

## Referensi

1. Dokumentasi Flask: https://flask.palletsprojects.com/

2. Dokumentasi TensorFlow: https://www.tensorflow.org/

3. Panduan Deployment Heroku: https://devcenter.heroku.com/

4. Dataset BPS: https://www.bps.go.id/