

BAB 4

HASIL PENELITIAN

4.1 RINGKASAN HASIL PENELITIAN

Pada bab ini, penelitian berfokus pada hasil prediksi metode Regresi Linear Berganda terhadap *Data Time Series* Tingkat Pengangguran Terbuka (TPT) di Provinsi Banten. Pengumpulan data dilakukan melalui observasi dan wawancara dengan pegawai bidang pelayanan.

4.2 HASIL PENELITIAN

Pada periode 22 April 2024 hingga 13 Mei 2024, peneliti telah melaksanakan pengumpulan data di Badan Pusat Statistik (BPS) Provinsi Banten terkait dengan *Data Time Series* Tingkat Pengangguran Terbuka (TPT) di Provinsi Banten. Data yang dikumpulkan mencakup rentang waktu dari tahun 2015 hingga 2023. Pengumpulan data ini bertujuan untuk mendapatkan informasi yang akurat mengenai perkembangan TPT di Provinsi Banten selama periode tersebut.

1. Hasil Pengumpulan Data

Dengan pengambilan data melalui metode observasi dan wawancara, penelitian ini berhasil memperoleh data yang diperlukan dengan cepat. Hasil dari pengumpulan data tersebut telah disajikan dalam **Tabel 4.1** sebagai *Data Time Series* Tingkat Pengangguran Terbuka (TPT) Provinsi Banten.

Tabel 4.1 Data Time Series

Inflasi (X1)	Kenaikan UMP (X2)	RLS (X3)	LP (X4)	TPT (Y)			Tahun
				Semester 1	Semester 2	Rata Rata Pertahun	
0,99	20,75	8,27	5,61	8,58	9,55	9,07	2015
0,61	11,5	8,37	5,28	7,95	8,92	8,44	2016
0,7	8,25	8,53	5,75	7,75	9,28	8,52	2017
0,63	8,71	8,26	5,77	7,77	8,52	8,15	2018
0,21	8,03	8,74	5,26	7,58	8,11	7,85	2019
0,21	8,51	8,89	-3,39	8,01	10,64	9,33	2020
0,69	0	8,93	4,49	9,01	8,98	9,00	2021
0,45	1,63	9,13	5,03	8,53	8,09	8,31	2022
0,48	6,4	9,15	4,81	7,97	7,52	7,75	2023

Data yang telah berhasil dikumpulkan selanjutnya akan dikelola sesuai dengan tahapan dari metode Regresi Linear Berganda, dalam pengolahan data tersebut variabel independen yaitu Inflasi, Kenaikan UMP, RLS dan LP, sedangkan variabel dependennya ialah TPT dengan nilai yang digunakan untuk perhitungannya ialah rata-rata dari TPT Semester 1 dan 2. Peneliti melakukan 2 tahap pengolahan data yaitu menggunakan *Microsoft Excel* dan pembangunan sistem dengan python.

2. Pemodelan Data

a. Menentukan Konstanta dan Koefisien

Dalam pemodelan data Regresi Linear Berganda perlu adanya konstanta (α) untuk menyediakan nilai dasar dari prediksi sebelum variabel independen diperhitungkan. Sedangkan koefisien (b) digunakan untuk menghitung prediksi nilai variabel dependen berdasarkan nilai-nilai variabel independen. **Gambar 4.1** menampilkan hasil konstanta dan koefisien dari perhitungan secara manual melalui *Microsoft Excel*.

	Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%	Lower 95,0%	Upper 95,0%
Intercept	11,0017	4,6272	2,3776	0,0762	-1,8454	23,8487	-1,8454	23,8487
X Variable 1	1,9511	0,6213	3,1404	0,0348	0,2261	3,6761	0,2261	3,6761
X Variable 2	-0,0219	0,0284	-0,7707	0,4839	-0,1008	0,0570	-0,1008	0,0570
X Variable 3	-0,2910	0,5003	-0,5815	0,5921	-1,6801	1,0982	-1,6801	1,0982
X Variable 4	-0,2057	0,0475	-4,3303	0,0123	-0,3376	-0,0738	-0,3376	-0,0738

Gambar 4.1 Mencari Nilai Konstanta dan Koefisien

Dengan menggunakan persamaan 2 untuk menghitung konstanta, persamaan 3 dan 4 untuk menghitung koefisien maka akan ditampilkan nilai dari perhitungan tersebut yang akan digunakan untuk prediksi, hasil dari konstanta menghasilkan sebesar **11,001%** dan nilai koefisien **b1 = 1,951**, **b2 = -0,021**, **b3 = -0.291**, dan **b4 = -0,205**. Berdasarkan nilai koefisien tersebut dapat dikatakan bahwa semua variabel independen dalam penelitian ini memiliki pengaruh terhadap hasil prediksi.

b. Melakukan Prediksi

Setelah melakukan pemodelan data selanjutnya melakukan prediksi secara manual dengan menggunakan *Microsoft Excel*. Prediksi dilakukan untuk memprediksi nilai dari variabel dependen berdasarkan nilai baru dari variabel independen. Pada **Gambar 4.2** menampilkan hasil prediksi terhadap nilai-nilai baru dari variabel independen melalui *Microsoft Excel*.

Inflasi (X1)	Kenaikan UMP (X2)	RLS (X3)	LP (X4)	TPT (Y)			Tahun	Hasil Prediksi Y'
				Semester 1	Semester 2	Rata Rata Tahunan		
0,99	20,75	8,27	5,61	8,58	9,55	9,07	2015	8,92
0,61	11,5	8,37	5,28	7,95	8,92	8,44	2016	8,42
0,7	8,25	8,53	5,75	7,75	9,28	8,52	2017	8,52
0,63	8,71	8,26	5,77	7,77	8,52	8,15	2018	8,45
0,21	8,03	8,74	5,26	7,58	8,11	7,85	2019	7,61
0,21	8,51	8,89	-3,39	8,01	10,64	9,33	2020	9,34
0,69	0	8,93	4,49	9,01	8,98	9,00	2021	8,83
0,45	1,63	9,13	5,03	8,53	8,09	8,31	2022	8,15
0,48	6,4	9,15	4,81	7,97	7,52	7,75	2023	8,15
-0,52	2,5	9,15	4,81	7,02	6,28	6,65	2024	6,28
Variabel Independen				Variabel Dependen				

Gambar 4.2 Hasil Prediksi

Dalam melakukan prediksi dapat menggunakan rumus pada persamaan 1, hasil dari perhitungan tersebut dapat menunjukkan hasil pada tahun 2015 sampai 2023, kemudian untuk tahun 2024 menggunakan data terbaru yang sebelumnya tidak dihitung untuk mencari konstanta dan koefisien. Dari perhitungan tersebut, Tingkat Pengangguran Provinsi Banten pada 2024 diprediksi turun menjadi **6.28%**.

3. Evaluasi Model

Dalam evaluasi model memerlukan beberapa penggunaan set pengujian untuk mengevaluasi kinerja model. Metode evaluasi umum yaitu *Mean Square Error (MSE)*, *Root Mean Square Error (RMSE)*, *R-square (R^2)* dan *Mean Absolute Error (MAE)*. **Gambar 4.3** menampilkan hasil dari evaluasi model secara manual melalui *Microsoft Excel*

Hasil Prediksi				Evaluasi Model		
Y'	$Y-Y'$	$(Y-Y')^2$	Absolute	MSE	RMSE	MAE
8,92	0,15	0,02	0,15	0,0427	0,2066	0,1608
8,42	0,02	0,00	0,02			
8,52	-0,01	0,00	0,01		Regression Statistics	
8,45	-0,30	0,09	0,30		Multiple R	0,917
7,61	0,23	0,06	0,23		R Square	0,841
9,34	-0,01	0,00	0,01		Adjusted R Square	0,681
8,83	0,17	0,03	0,17		Standard Error	0,310
8,15	0,16	0,02	0,16		Observations	9
8,15	-0,40	0,16	0,40			
6,28						

Gambar 4.3 Evaluasi Model

Dengan menggunakan rumus persamaan 5,6,7,8 pada fitur perhitungan Regresi dalam *Microsoft Excel* maka akan ditampilkan *Summary Output* hasil dari evaluasi model. Hasil dari evaluasi model ialah *R Square* sebesar 84,1%, *MSE* 0,042%, *RMSE* 0,206% dan *MAE* sebesar 0,160% yang menunjukkan bahwa model regresi linear berganda terhadap prediksi TPT memiliki performa yang sangat baik. Dengan nilai error yang kecil berarti bahwa prediksi yang dihasilkan oleh model sangat dekat dengan nilai aktual, dengan hanya sedikit kesalahan.

4.3 IMPLEMENTASI MODEL

Sistem yang dibangun menggunakan bahasa pemrograman python. Sistem ini dapat melakukan prediksi sesuai dengan data yang diperlukan, dan pengguna sistem dapat melakukan Tindakan lain seperti unduh hasil prediksi dan melihat grafik Tingkat Pengangguran Terbuka (TPT) beberapa tahun lalu.

1. Pembuatan skrip *app.py*

File *app.py* merupakan hal yang penting untuk pembangunan sistem dengan bahasa pemrograman python, *app.py* memiliki fungsi untuk menentukan semua rute dan fungsi yang akan dilakukan untuk setiap tindakan. Pada sistem prediksi ini *app.py* memungkinkan untuk memprediksi tingkat pengangguran terbuka (TPT) berdasarkan data yang diperlukan. Berikut adalah penjelasan mengenai fungsi-fungsi yang ada di dalam pembangunan sistem prediksi TPT Provinsi Banten.

a. Import dan inisialisasi Flask

Fungsi skrip ini untuk mengimpor pustaka yang diperlukan dan menginisialisasi aplikasi Flask. Flask sendiri memiliki fungsi untuk membuat aplikasi berbasis flask yang didalamnya terdapat *request* yang berfungsi untuk mengakses data yang dikirimkan oleh pengguna dalam permintaan HTTP, *render_template* berfungsi untuk merender template HTML menggunakan Jinja2, *jsonify* berfungsi untuk mengubah data python menjadi JSON dan mengembalikannya sebagai respons dari HTTP. `sklearn.linear_model.LinearRegression` merupakan algoritma regresi linear berganda dari scikit-learn error untuk menghitung *mean* kesalahan kuadrat serta menghitung *mean* kesalahan absolut.

```
from flask import Flask, request, render_template, jsonify, send_file
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error
import os

app = Flask(__name__)
```

b. Input *Data Time Series*

Skrip ini berfungsi untuk mendefinisikan *dataset time series* yang berisi data independen dan dependen dari tahun 2021 hingga 2023. Dalam skrip tersebut memiliki beberapa fungsi seperti membuat DataFrame baru menggunakan pandas (`'pd.DataFrame()'`) dan argumen yang diberikan ialah *dictionary* dimana kunci adalah nama kolom dan nilai adalah daftar nilai untuk setiap kolom. Selain itu terdapat variabel X DataFrame yang berisikan fitur-fitur yang akan digunakan sebagai input untuk pemodelan dan y menjadi DataFrame yang berisi target yang akan diprediksi oleh model.

```
df = pd.DataFrame({
    'Inflasi': [0.99, 0.61, 0.70, 0.63, 0.21, 0.21, 0.69, 0.45, 0.48],
    'UMP': [20.75, 11.5, 8.25, 8.71, 8.03, 8.51, 0.0, 1.63, 6.4],
    'RLS': [8.27, 8.37, 8.53, 8.26, 8.74, 8.89, 8.93, 9.13, 9.15],
    'LP': [5.61, 5.28, 5.75, 5.77, 5.26, -3.39, 4.49, 5.03, 4.81],
    'TPT': [9.07, 8.44, 8.52, 8.15, 7.85, 9.33, 9.00, 8.31, 7.75],
    'Tahun': [2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023]
```

```

}))
X = df[['Inflasi', 'UMP', 'RLS', 'LP']]
y = df['TPT']

```

c. Membuat dan melatih model regresi linear berganda

Skrip pada bagian ini berfungsi untuk membuat model regresi linear berganda, melatihnya dengan data yang tersedia, dan menghitung konstanta (*intercept*) serta koefisien untuk setiap variabel independen. Selain itu, beberapa metrik evaluasi model juga dihitung, seperti *R-square*, *Mean Square Error (MSE)*, *Root Mean Square Error (RMSE)* dan *Mean Absolute Error (MAE)*. Dalam skrip untuk membuat dan melatih model Regresi Linear Berganda menggunakan fungsi dari 'LinearRegression()' dan `model.fit(X,y)` untuk memetakan hubungan antara fitur dan target. Fungsi '`model.intercept_`' ialah mengambil nilai *intercept* (konstanta) dari model regresi linear dan '`model.coef_`' berfungsi mengambil koefisien dari fitur-fitur dalam model serta menunjukkan seberapa besar pengaruh masing-masing fitur terhadap prediksi. '`model.score(X, y)`' berfungsi untuk menghitung nilai *R-squared* (koefisien determinasi) dari model. Nilai ini menunjukkan seberapa baik model menjelaskan variabilitas data target, nilai *R-square* berkisar antara 0 dan 1 dengan nilai mendekati 1 menunjukkan model yang baik. fungsi `mse`, `rmse`, `mae` hamper memiliki fungsi yang sama apabila hasilnya semakin kecil maka model prediksi dapat dikatakan semakin baik.

```

model = LinearRegression()
model.fit(X, y)

# Menghitung konstanta (intercept) dan koefisien
intercept = model.intercept_
coefficients = model.coef_

r_squared = model.score(X, y)
mse = mean_squared_error(y, model.predict(X))
rmse = np.sqrt(mse)
mae = mean_absolute_error(y, model.predict(X))

```

d. Route untuk halaman utama

Skrip route ini berfungsi untuk menampilkan halaman utama (**index.html**) dengan mengirimkan data metrik evaluasi model, konstanta, koefisien, dan dataset

yang ada. `@app.route('/')` mendefinisikan rute untuk URL root (/) ketika pengguna mengunjungi URL ini maka fungsi `home` akan dipanggil. sedangkan `'return render_template('index.html', ...)'` berfungsi untuk merender template HTML bernama `index.html` dan mengirimkan data seperti `r_squared`, `mse`, `rmse`, `mae` ke template. Beda halnya dengan fungsi `'df=df.to_dict(orient='records')'` yang berfungsi mengkonversi DataFrame `df` menjadi daftar kamus (*list of dictionaries*) dan mengirimkannya ke file *template*.

```
@app.route('/')
def home():
    return render_template('index.html', r_squared=r_squared, mse=mse,
                           rmse=rmse, mae=mae, df=df.to_dict(orient='records'),
                           intercept=intercept, coefficients=coefficients)
```

e. Route untuk mendapatkan data

Jika fungsi route sebelumnya untuk halaman utama, fungsi Route ini adalah mengirimkan data tahun dan TPT sebagai JSON untuk keperluan pengambilan data. `'@app.route('/get_data')'` berfungsi untuk mendefinisikan rute untuk URL `'/get_data'`. Ketika pengguna mengunjungi URL ini, fungsi `get_data` akan dipanggil. data merupakan dictionary yang akan berisi 2 variabel yaitu tahun dan tpt yang nantinya metode `'tolist()'` akan mengkonversi kolom DataFrame menjadi daftar (list). Serta yang terakhir yaitu fungsi `'jsonify(data)'` untuk mengkonversi kamus data menjadi format JSON dan mengembalikannya sebagai respons HTTP.

```
@app.route('/get_data')
def get_data():
    # Mengirim data TPT sebagai JSON
    data = {
        'tahun': df['Tahun'].tolist(),
        'tpt': df['TPT'].tolist()
    }
    return jsonify(data)
```

f. Route untuk prediksi data

Fungsi dari skrip route ini ialah menerima input dari formulir, melakukan prediksi TPT menggunakan model, dan mengupdate dataset serta matrix model. Hasil prediksi dan metrik yang diperbarui kemudian dikirim kembali ke halaman

index.html. Dataset yang diperbarui juga disimpan ke dalam file CSV. 'def predict()' akan mendefinisikan fungsi predict yang akan dijalankan ketika rute ini diakses. 'global df, model, r_squared, mse, rmse, mae, intercept, coefficients' berfungsi untuk menyatakan variabel-variabel global yang akan di update dalam fungsi tersebut dan 'float(...)' akan mengkonversi nilai yang diambil dari formulir menjadi tipe float. Fungsi 'new_tahun = df['Tahun'].max() + 1' ialah menghitung tahun baru dengan menambah satu tahun dari tahun terakhir yang ada di DataFrame. 'new_data = {...}' untuk membuat kamus dengan data baru termasuk hasil prediksi. 'model.fit(X, y)' berfungsi melatih ulang model regresi linear berganda dengan data yang telah diperbarui. Fungsi dari 'df.to_csv('static/data/predictions.csv', index=False)' ialah untuk menyimpan DataFrame yang diperbarui ke file CSV di lokasi 'static/data/predictions.csv' pada visual studio code tanpa menyertakan indeksinya.

```
@app.route('/predict', methods=['POST'])
def predict():
    try:
        global df, model, r_squared, mse, rmse, mae, intercept, coefficients
        Inflasi = float(request.form['Inflasi'])
        UMP = float(request.form['UMP'])
        RLS = float(request.form['RLS'])
        LP = float(request.form['LP'])

        prediction = model.predict(np.array([[Inflasi, UMP, RLS, LP]]))
        output = round(prediction[0], 2)

        new_tahun = df['Tahun'].max() + 1
        new_data = {'Inflasi': Inflasi, 'UMP': UMP, 'RLS': RLS, 'LP': LP,
                    'TPT': output, 'Tahun': new_tahun}
        df = df.append(new_data, ignore_index=True)

        X = df[['Inflasi', 'UMP', 'RLS', 'LP']]
        y = df['TPT']

        model.fit(X, y)

        # Update model metrics and parameters
        r_squared = model.score(X, y)
        mse = mean_squared_error(y, model.predict(X))
        rmse = np.sqrt(mse)
        mae = mean_absolute_error(y, model.predict(X))
```



```

intercept = model.intercept_
coefficients = model.coef_

# Menyimpan DataFrame ke CSV
if not os.path.exists('static/data'):
    os.makedirs('static/data')
df.to_csv('static/data/predictions.csv', index=False)

return render_template('index.html', prediction_text=f'Tingkat
Pengangguran Terbuka yang diprediksi: {output}%', r_squared=r_squared,
mse=mse, rmse=rmse, mae=mae, df=df.to_dict(orient='records'),
intercept=intercept, coefficients=coefficients)
except KeyError as e:
    return f"Input form missing: {e}", 400

```

g. Route untuk mengunduh CSV

Fungsi Route ini memungkinkan pengguna untuk mengunduh dataset prediksi yang telah diperbarui dalam format CSV. Meski belum melakukan prediksi, data CSV sudah dapat di unduh untuk melihat data 9 Tahun terakhir. 'path = 'static/data/predictions.csv'' memiliki fungsi untuk menetapkan jalur file CSV yang akan diunduh. File ini disimpan di direktori 'static/data' dan dinamai 'predictions.csv' serta fungsi 'send_file(path, as_attachment=True, download_name='predictions.csv')' menggunakan fungsi 'send_file' dari Flask untuk mengirim file ke klien, terdapat fungsi 'as_attachment=True' bahwa file harus dikirim sebagai lampiran sehingga browser akan memulai unduh file. Fungsi Terakhir yaitu 'download_name='predictions.csv' yang berfungsi untuk menentukan nama file yang akan diunduh oleh pengguna.

```

@app.route('/download_csv')
def download_csv():
    path = 'static/data/predictions.csv'
    return send_file(path, as_attachment=True,
download_name='predictions.csv')

```

h. Menjalankan aplikasi flask

Fungsi skrip ini hanya untuk menjalankan aplikasi flask dalam mode debug. Fungsi 'if __name__ == "__main__":' Ini adalah untuk konstruksi standar di Python untuk memastikan bahwa kode di dalam blok ini hanya dijalankan jika skrip dijalankan secara langsung, 'debug=True:' mengaktifkan mode debug, yang dapat

melakukan debugging interaktif jika ada kesalahan runtime dan secara otomatis me-reload server setiap kali ada perubahan pada skrip terdeteksi.

```
if __name__ == "__main__":
    app.run(debug=True)
```

2. Pembuatan antarmuka dan skrip *index.html*

Jika skrip *app.py* memiliki fungsi untuk menentukan semua rute dan fungsi yang akan dilakukan untuk setiap Tindakan prediksi. Skrip ***index.html*** memiliki fungsi untuk menampilkan antarmuka guna prediksi Tingkat Pengangguran Terbuka (TPT) di Provinsi Banten. Berikut adalah penjelasan fungsi dari *index.html*.

a. Skrip Struktur dasar HTML

Pada skrip pertama ini ialah untuk mendefinisikan struktur dasar HTML, termasuk meta tag untuk pengaturan karakter dan viewport, serta link ke stylesheet eksternal (Bootstrap, Google Fonts, Font Awesome) dan script eksternal (jQuery, Bootstrap, Chart.js).

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
<title>PREDIKSI TPT</title>
<link
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap
.min.css" rel="stylesheet">
<link
href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;500;700
&display=swap" rel="stylesheet">
<link href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.4/css/all.min.css" rel="stylesheet">
<style>
/* CSS styles here */
</style>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.9.4/Chart.min.j
s"></script>
</head>
<body>
<!-- Body content here -->
```

```
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.m
in.js"></script>
</body>
</html>
```

b. Skrip Header

```
<div class="header">


<h1>PREDIKSI TINGKAT PENGANGGURAN PROVINSI BANTEN</h1>
</div>
```



Gambar 4.4 Tampilan Header

Skrip Header diatas berfungsi untuk menampilkan header halaman yang mencakup dua logo (kiri dan kanan) serta judul halaman, seperti yang dapat dilihat pada **Gambar 4.4** diatas.

c. Skrip Konten Utama

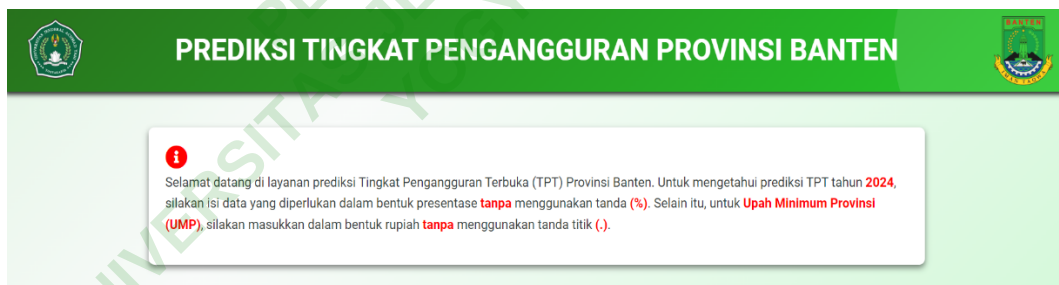
```
<div class="container">
<div class="row">
<div class="col-md-12">
<div class="intro">
<i class="fas fa-info-circle icon"></i>
<p>Selamat datang di layanan prediksi ...</p>
</div>
<form action="/predict" method="post" class="mt-4">
<div class="card">
<div class="card-body">
<!-- Form inputs here -->
</div>
</div>
</form>
{% if prediction_text %}
```

```

<h2 class="text-center mt-4">{{ prediction_text }}</h2>
{% endif %}
<div class="content-section">
<h3 class="text-center section-heading">GRAFIK TINGKAT PENGANGGURAN
PROVINSI BANTEN</h3>
<div class="card">
<div class="card-body">
<canvas id="myChart"></canvas>
</div>
</div>
</div>
<div class="content-section">
<h3 class="text-center section-heading">DATA PREDIKSI</h3>
<a href="/download_csv" class="btn btn-primary btn-block">Download
CSV</a>
<div class="card mt-4">
<div class="card-body">
<table class="table table-striped">
<thead>
<!-- Table headers here -->
</thead>
<tbody>
{% for row in df %}
<tr>
<td>{{ row.Tahun }}</td>
<td>{{ row.Inflasi }}</td>
<td>{{ row.UMP }}</td>
<td>{{ row.RLS }}</td>
<td>{{ row.LP }}</td>
<td>{{ row.TPT }}</td>
</tr>
{% endfor %}
</tbody>
</table>
</div>
</div>
</div>
<div class="row content-section">
<div class="col-md-6">
<h3 class="text-center section-heading">EVALUASI MODEL</h3>
<div class="card">
<div class="card-body text-center">
<p class="data-point">R-squared: {{ r_squared }}</p>
<p class="data-point">MSE: {{ mse }}</p>
<p class="data-point">RMSE: {{ rmse }}</p>
<p class="data-point">MAE: {{ mae }}</p>

```

```
</div>
</div>
</div>
<div class="col-md-6">
<h3 class="text-center section-heading">KONSTANTA & KOEFISIEN
MODEL</h3>
<div class="card">
<div class="card-body text-center">
<p class="data-point">Intercept: {{ intercept }}</p>
<p class="data-point">Koefisien:</p>
<ul class="list-unstyled">
<li class="data-point">Inflasi: {{ coefficients[0] }}</li>
<li class="data-point">UMP: {{ coefficients[1] }}</li>
<li class="data-point">RLS: {{ coefficients[2] }}</li>
<li class="data-point">IP: {{ coefficients[3] }}</li>
</ul>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
```



Gambar 4.5 Tampilan Informasi Awal

Scrip konten utama memiliki fungsi untuk memberikan intro secara singkat seperti yang terlihat pada **Gambar 4.5** dimana permintaan yang dilakukan pada scrip akan ditampilkan pada tampilan informasi awal.

i

Selamat datang di layanan prediksi Tingkat Pengangguran Terbuka (TPT) Provinsi Banten. Untuk mengetahui prediksi TPT tahun **2024**, silakan isi data yang diperlukan dalam bentuk presentase **tanpa** menggunakan tanda (%). Selain itu, untuk **Upah Minimum Provinsi (UMP)**, silakan masukkan dalam bentuk rupiah **tanpa** menggunakan tanda titik (.).

Inflasi:

Kenalkan UMP (Upah Minimum Provinsi):

Rata-rata Lama Sekolah (RLS):

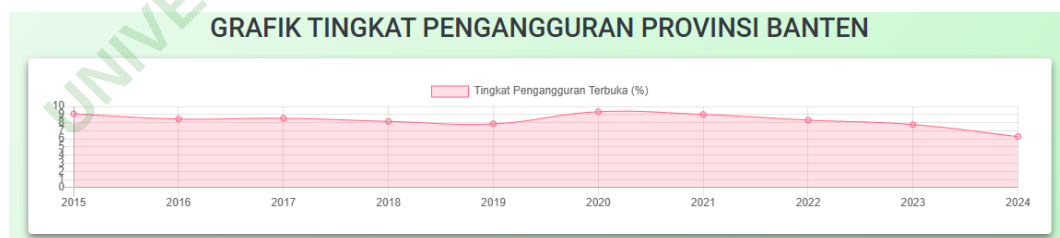
Lapangan Pekerjaan (LP):

Prediksi

Tingkat Pengangguran Terbuka yang diprediksi: 6.28%

Gambar 4.6 Tampilan Formulir

Menampilkan formulir untuk input data yang diperlukan guna prediksi seperti yang dapat dilihat pada **Gambar 4.6**. Tampilan form diatas pengguna dapat menginputkan *data time series* yang diperlukan, apabila data yang telah di-inputkan sesuai dengan data yang sesuai maka selanjutnya pengguna hanya perlu melakukan klik prediksi, dan hasil prediksi untuk Tingkat Pengangguran Terbuka 2024 akan ditampilkan oleh sistem. Peneliti telah melakukan prediksi TPT Provinsi Banten Tahun 2024 dan hasil prediksi TPT melalui sistem yang dibangun ialah sebesar **6,28%** yang mengartikan akan ada penurunan TPT di Provinsi Banten.



Gambar 4.7 Tampilan Grafik TPT

Menampilkan grafik tingkat pengangguran terbuka seperti pada **Gambar 4.7**, Pada gambar diatas nilai TPT yang ditampilkan ialah dari tahun 2015 sampai 2023 sehingga ketika pengguna telah melakukan input data prediksi pada form sistem akan secara otomatis menambahkan hasil TPT 2024 ke dalam grafik.

DATA PREDIKSI					
Download CSV					
Tahun	Inflasi	UMP	RLS	LP	TPT
2015.0	0.99	20.75	8.27	5.61	9.07
2016.0	0.61	11.5	8.37	5.28	8.44
2017.0	0.7	8.25	8.53	5.75	8.52
2018.0	0.63	8.71	8.26	5.77	8.15
2019.0	0.21	8.03	8.74	5.26	7.85
2020.0	0.21	8.51	8.89	-3.39	9.33
2021.0	0.69	0.0	8.93	4.49	9.0
2022.0	0.45	1.63	9.13	5.03	8.31
2023.0	0.48	6.4	9.15	4.81	7.75
2024.0	-0.52	2.5	9.15	4.81	6.28

Gambar 4.8 Tampilan Data Hasil Prediksi

Menampilkan data prediksi yang telah diperbarui seperti pada **Gambar 4.8**, pada gambar diatas setelah pengguna melakukan prediksi, tabel data prediksi juga akan mengalami penambahan data sesuai dengan yang telah pengguna inputkan. Selain itu, dalam sistem ini juga dapat melakukan unduh data dalam bentuk CSV baik sebelum atau sesudah melakukan prediksi.

EVALUASI MODEL	KONSTANTA & KOEFISIEN MODEL
R-squared: 0.8413158790752929 MSE: 0.04250422746181026 RMSE: 0.2061655341268522 MAE: 0.16046254264547846	Intercept: 11.026561144415759 Koefisien: Inflasi: 1.9509630490477357 UMP: -0.021885280134509008 RLS: -0.2932495793900656 LP: -0.2058533050071112

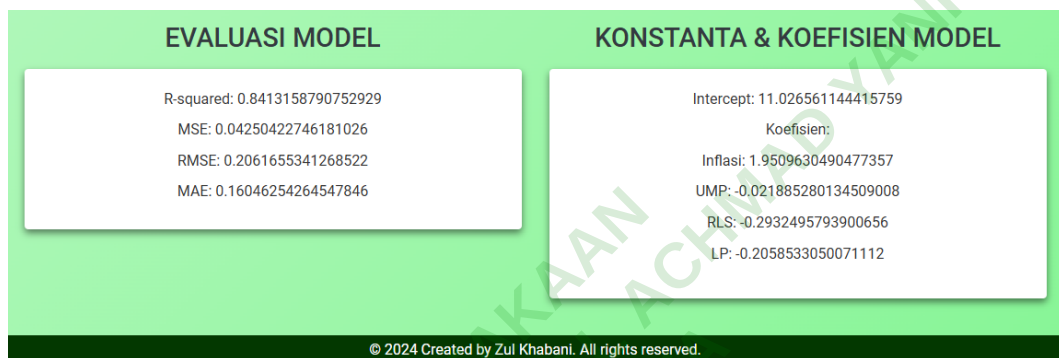
Gambar 4.9 Tampilan Evaluasi Model

Menampilkan hasil dari evaluasi model, konstanta dan koefisien model seperti pada **Gambar 4.9**. Pemodelan dan evaluasi model ditampilkan pada sistem guna memberikan penjelasan bahwa perhitungan yang dilakukan oleh sistem ini berjalan

sesuai dengan algoritma dari regresi linear berganda dan sesuai dengan hasil perhitungan manual melalui excel.

d. Skrip Footer

```
<footer id="footer">
<p>&copy; 2024 Created by Zul Khabani. All rights reserved.</p>
</footer>
```



Gambar 4.10 Tampilan Footer

Pada **Gambar 4.10** diatas menampilkan footer halaman dengan menampilkan informasi hak cipta. Seperti yang dapat dilihat, peneliti menambahkan informasi hak cipta pembuatan sistem yang terdapat di bagian footer.

e. Skrip JavaScript

Pada skrip javascript digunakan untuk mengambil data dari server guna menampilkan grafik menggunakan Chart.js, serta menampilkan footer saat halaman discroll lebih dari 100px.

```
<script>
document.addEventListener('DOMContentLoaded', function () {
  fetch('/get_data')
    .then(response => response.json())
    .then(data => {
      var ctx = document.getElementById('myChart').getContext('2d');
      var myChart = new Chart(ctx, {
        type: 'line',
        data: {
          labels: data.tahun,
```



```

datasets: [{
  label: 'Tingkat Pengangguran Terbuka (%)',
  data: data.tpt,
  backgroundColor: 'rgba(255, 99, 132, 0.2)',
  borderColor: 'rgba(255, 99, 132, 1)',
  borderWidth: 1
}]
},
options: {
  responsive: true,
  maintainAspectRatio: false,
  scales: {
    yAxes: [{
      ticks: {
        beginAtZero: true
      }
    }]
  }
});
});
window.addEventListener('scroll', function () {
  const footer = document.getElementById('footer');
  if (window.scrollY > 100) {
    footer.classList.add('visible');
  } else {
    footer.classList.remove('visible');
  }
});
});
</script>

```

4.4 HASIL PEMODELAN

Berdasarkan hasil pemodelan menggunakan regresi linear berganda menggunakan *Microsoft excel* dan sistem yang telah dibangun, hasil yang diperoleh dapat dilihat pada **Gambar 4.1** dan **Gambar 4.9** selanjutnya, hasil tersebut dapat dijelaskan bahwa

$$Y = 11,001 + 1,951X_1 - 0,021X_2 - 0,291X_3 - 0,205X_4$$

1. Berdasarkan hasil pemodelan menggunakan regresi linear berganda, jika variabel independen X_1 (inflasi), X_2 (kenaikan UMP), X_3 (rata-rata lama sekolah) dan X_4 (lapangan pekerjaan) tidak mengalami perubahan, maka nilai **konstanta (intersep)** dalam model akan menjadi nilai prediksi Tingkat Pengangguran Terbuka (TPT) di Provinsi Banten sebesar **11,001 unit**.
2. Apabila inflasi mengalami kenaikan sebesar 1 unit, maka TPT di Provinsi Banten diperkirakan akan mengalami kenaikan sebesar 1,951 unit. Dengan kata lain, setiap kenaikan 1 unit pada inflasi akan mengakibatkan peningkatan TPT sebesar 1,951 unit.
3. Apabila UMP mengalami kenaikan sebesar 1 unit, maka TPT di Provinsi Banten diperkirakan akan mengalami penurunan sebesar 0,021 unit. Dengan kata lain, setiap kenaikan 1 unit pada UMP akan mengakibatkan penurunan TPT sebesar 0,021 unit.
4. Apabila rata-rata lama sekolah (RLS) mengalami kenaikan sebesar 1 unit, maka TPT di Provinsi Banten diperkirakan akan mengalami penurunan sebesar 0,291 unit. Dengan kata lain, setiap kenaikan 1 unit pada RLS akan mengakibatkan penurunan TPT sebesar 0,291 unit, dan
5. Apabila lapangan pekerjaan mengalami peningkatan sebesar 1 unit, maka TPT di Provinsi Banten diperkirakan akan mengalami penurunan sebesar 0,205 unit. Dengan kata lain, setiap kenaikan 1 unit pada lapangan pekerjaan akan mengakibatkan penurunan TPT sebesar 0,205 unit.

4.5 HASIL PREDIKSI

Hasil prediksi yang telah dilakukan menggunakan Regresi Linear Berganda berdasarkan *data time series* dengan hasil TPT Provinsi Banten akan mengalami penurunan atau TPT Provinsi Banten pada tahun 2024 sebesar **6,28%** dari tahun sebelumnya sebesar 7,52%. Hasil tersebut juga didukung dengan hasil evaluasi pemodelan *MSE*, *RMSE*, *MAE* yang memiliki nilai kecil dan *R-Squared* sebesar 84,1%.