

## ASSESSMENT CLO PEMODELAN DAN OPTIMASI LANJUT

3 November 2025

Nama Anggota:

1. Muhammad Naufal Erza Farandi (203012510031)
2. Novia Natasya (203012510034)
3. Erlangga Dewa Sakti (203012510035)

Source code:

[https://drive.google.com/drive/folders/1IJq\\_VLAig3RWED9mYHDkHRxfHovpbGEf?usp=drive\\_link](https://drive.google.com/drive/folders/1IJq_VLAig3RWED9mYHDkHRxfHovpbGEf?usp=drive_link)

Jawaban & Pembahasan

### SOAL 1

Analisis ini bertujuan untuk membangun prediksi mengenai biaya rental mobil ( $y$ ) berdasarkan jarak tempuh ( $x$ ) dengan data yang dipakai sesuai yang telah diberikan pada soal, yaitu sebagai berikut:

- $x$  (Jarak dalam satuan km): [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]
- $y$  (Biaya Rupiah): [20500, 18000, 26500, 30000, 34500, 39500, 37500, 45500, 50500, 57500, 54500, 58000, 68000, 67500, 74500]

Dengan menggunakan model linear yang memiliki bentuk persamaan:

$$y = Wx + a$$

Keterangan:

$W$  = *slope* atau gradien/koeffisien regresi

$a$  = *intercept* (bias)

#### 1. Least Square Method.

- a) Berdasarkan implementasi pada source code yang telah diberikan, nilai  $W$  dan  $a$  dihitung dari rumus berikut.

$$W = \frac{\sum_{i=1}^N x_i(y_i - \bar{y})}{\sum_{i=1}^N x_i(x_i - \bar{x})}$$

$$a = \bar{y} - w\bar{x}$$

Hasil Perhitungan eksekusi kode menghasilkan *output* nilai koefisien sebagai berikut

- Koefisien Regresi ( $W$ ) = 3941.071429
- Intercept ( $a$ ) = 10030.357143

Dengan demikian, berdasarkan nilai yang telah didapatkan dari hasil perhitungan model regresi linear yang dihasilkan dari metode Least Square Method adalah:

$$y = 3941.071429x + 10030.357143$$

Kami juga menyertakan hasil output untuk menentukan nilai koefisien regresi, intercept, dan model regresi linear dengan *Least Square Method* (LSM) pada source code dengan output di bawah ini:

```
Menghitung Slope (W):
W = numerator / denominator
W = 1103500.00 / 280.00
W = 3941.071429

Menghitung Intercept (a):
a =  $\bar{y}$  - W ×  $\bar{x}$ 
a = 45500.00 - 3941.071429 × 9.0
a = 10030.357143
```

Model Regresi Linear (LSM):  $y = 3941.071429x + 10030.357143$

- b) Untuk mengestimasi biaya rental mobil yang harus dibayarkan untuk menempuh jarak 9.25 km, nilai  $x = 9.25$  disubstitusikan ke dalam model regresi linear yang telah dirumuskan pada nomor 1a dengan perhitungan sebagai berikut, dengan  $y$  sebagai biaya rental:

$$\begin{aligned} y &= 3941.071429(9.25) + 10030.357143 \\ y &= 36454.910714 + 10030.357143 \\ y &\approx 46485.27 \end{aligned}$$

Sehingga, perhitungan di atas mendapatkan nilai yang sama dengan hasil run code yang dilakukan pada sel ke-5, yaitu sebesar 46,485.27 rupiah untuk prediksi biaya rental jika jaraknya 9,25 km.

Hasil yang didapatkan juga setara dengan output dari source code yang kami buat dengan hasilnya di bawah ini:

```
Menggunakan model LSM:  $y = 3941.071429x + 10030.357143$ 

Perhitungan:
y = W × x + a
y = 3941.071429 × 9.25 + 10030.357143
y = 36454.910714 + 10030.357143
y = 46485.27
```

Estimasi harga untuk jarak 9.25 km = Rp 46,485.27

## 2. Gradient Descent

Sebelum masuk ke bagian implementasi, saat kami sedang melakukan percobaan terhadap *learning rate* 0.1, terdapat data yang bernilai *nan*. Hal tersebut terjadi karena ukuran *learning rate* yang dipilih, yaitu 0.1, terlalu besar untuk data yang tidak diskalakan.

```
=====
EKSPERIMEN 2: Learning Rate  $\eta = 0.1$ 
=====
Iterasi 0: W = 96613.3333, a = 9100.0000, Loss = 2366316666.6667
Iterasi 100: W = 114198911376900148529740440226493720183772090494076485073890468507931031963202039239004
Iterasi 200: W = 1420067261968923170831260094402218886477551661793684826195282551490900621311948336118912
Iterasi 300: W = nan, a = nan, Loss = nan
Iterasi 400: W = nan, a = nan, Loss = nan
Iterasi 500: W = nan, a = nan, Loss = nan
Iterasi 600: W = nan, a = nan, Loss = nan
Iterasi 700: W = nan, a = nan, Loss = nan
Iterasi 800: W = nan, a = nan, Loss = nan
Iterasi 900: W = nan, a = nan, Loss = nan
Iterasi 999: W = nan, a = nan, Loss = nan
```

Sebagai solusinya, kami menerapkan normalisasi pada data kemudian didenormalisasikan setelah mengimplementasikan *gradient descent* dengan harapan agar data dapat dijalankan dengan *learning rate* 0.1.

c) Berdasarkan implementasi yang dilakukan pada source code yang telah diberikan sebelumnya, berikut adalah hasil yang diperoleh serta *Mean Square Error (MSE)* untuk:

- **Skenario 1: Learning rate ( $\alpha$ ) = 0.001**

Koefisien Regresi ( $W$ ): 3259.685673

Intercept ( $a$ ): 16549.422594

Model:  $y = 3259.685673x + 16549.422594$

Final MSE (normalized): 0.0046844577

- **Skenario 2: Learning rate ( $\alpha$ ) = 0.1**

Koefisien Regresi ( $W$ ): 3941.071429

Intercept ( $a$ ): 10030.357143

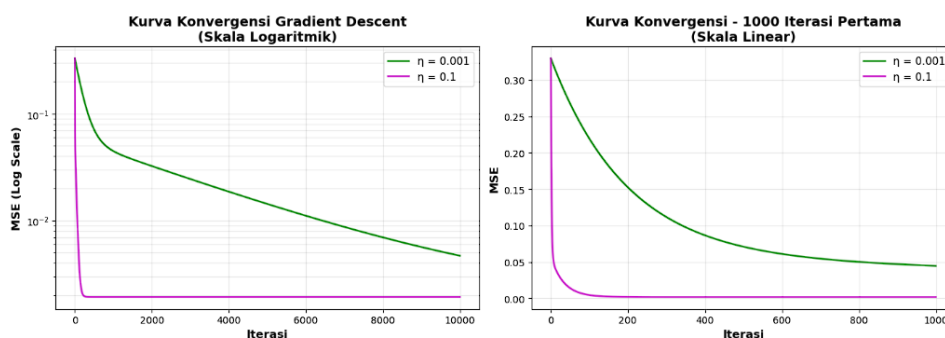
Model:  $y = 3941.071429x + 10030.357143$

Final MSE (normalized): 0.0019218979

## Analisis efektivitas Learning Rate

Dengan menggunakan data yang telah dinormalisasi, analisis ini dievaluasi berdasarkan kecepatan konvergensi dan akurasi parameter ( $W$  dan  $a$ ).

Skenario 1 menunjukkan konvergensi yang stabil namun sangat lambat. Bahkan setelah 10.000 iterasi, *log training* menunjukkan bahwa MSE (MSE\_norm = 0.004684) masih terus menurun, menandakan model belum sepenuhnya konvergen. Hal ini menyebabkan parameter akhirnya ( $W = 3259.69$ ,  $a = 16549.42$ ) masih memiliki deviasi signifikan dari *baseline* LSM. Sebaliknya, Skenario 2 terbukti jauh lebih efektif. *Log training* menunjukkan konvergensi yang sangat cepat, mencapai MSE\_norm minimum (0.001922) hanya dalam 1000 iterasi.

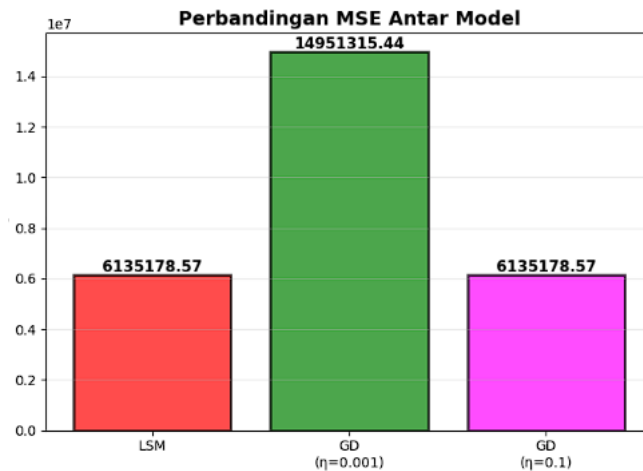


Analisis ini kami perkuat dengan menambahkan kurva konvergensi dengan skala logaritmik dan skala linear. Plot skala logaritmik (kiri) menunjukkan bahwa skenario 2 (ungu) konvergen sangat cepat ke MSE\_norm minimum (0.001922) dan kemudian stabil. Hal ini menjadi kesimpulan pertama bahwa skenario 2 adalah skenario dengan solusi yang paling optimal untuk LSM. Sebaliknya, Skenario 1 (hijau) menunjukkan penurunan *error* yang jauh lebih lambat dan landai, dan bahkan setelah 10.000 iterasi, MSE\_norm-nya (0.004684) masih lebih tinggi dan terlihat belum sepenuhnya konvergen. Plot skala linear (kanan) semakin menegaskan perbedaan kecepatan ini, di mana Skenario 2 telah mencapai solusi optimalnya sebelum iterasi ke-200, sementara Skenario 1 masih memiliki *error* yang jauh lebih tinggi pada iterasi ke-1000. Dengan begitu, visualisasi ini membuktikan bahwa skenario 2 dengan *learning rate* 0.1 jauh lebih unggul dan efisien dalam mencapai solusi optimal.

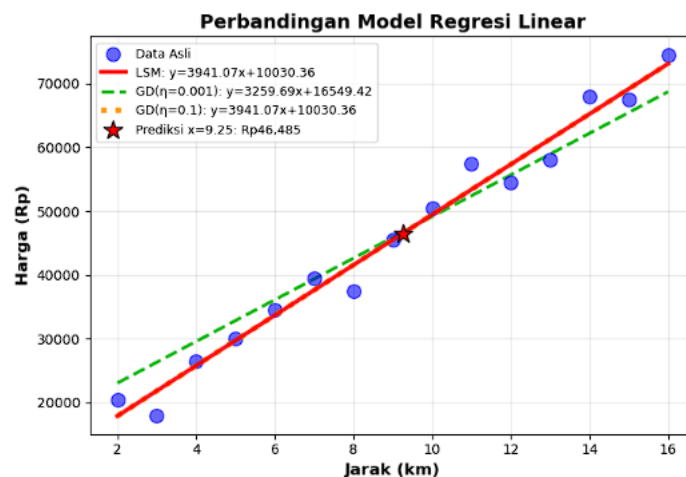
d) Di bawah ini adalah tabel perbandingan hasil antara *Least Square Method* (LSM) dengan Gradient Descent dengan membandingkan nilai *Mean Square Error* (MSE) untuk mencari model mana yang menghasilkan error paling kecil.

Model	W	a	MSE
<b>LSM (Baseline)</b>	<b>3941.071429</b>	<b>10030.357143</b>	<b>6135178.57</b>
GD ( $\alpha = 0.001$ )	3259.685673	16549.422594	14951315.44
<b>GD (<math>\alpha = 0.1</math>)</b>	<b>3941.071429</b>	<b>10030.357143</b>	<b>6135178.57</b>

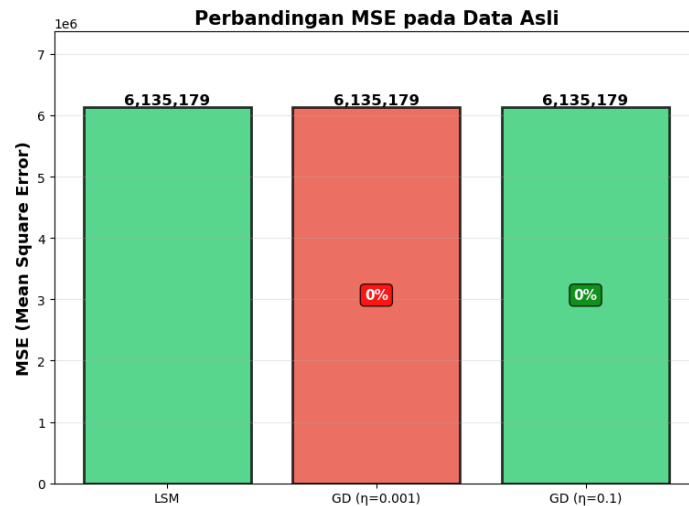
Berdasarkan tabel di atas, dapat disimpulkan bahwa model *Least Square Method* (LSM) dengan Gradient Descent ( $\alpha = 0.1$ ) memiliki nilai *Mean Square Error* (MSE) yang sama, yaitu 6135178.57. Perbandingan ini juga telah disajikan dengan visualisasi bar plot antar model dengan nilai *Mean Square Error* (MSE) masing-masing yang telah didapatkan sebelumnya.



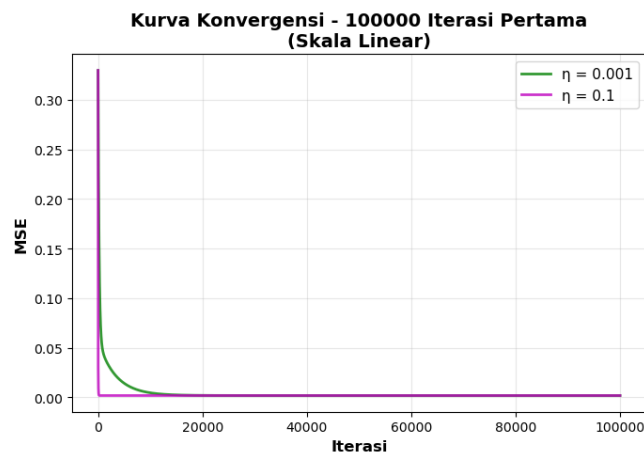
Lebih lanjut, kedua model ini dilakukan perbandingan model regresi linearnya dengan visualisasi yang dapat dilihat di bawah ini.



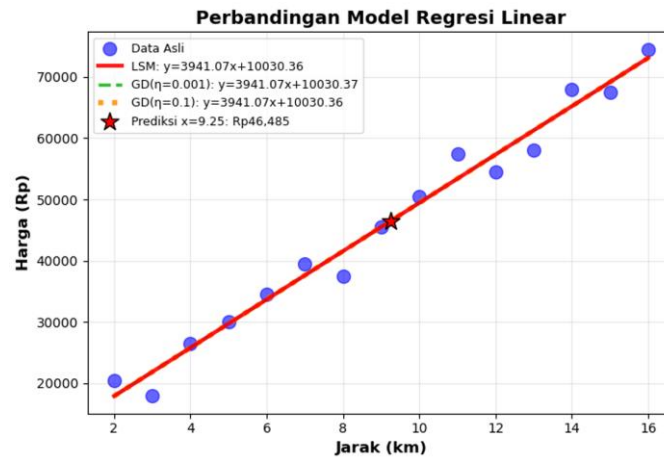
Grafik dengan jelas menunjukkan bahwa garis merah solid (LSM) dan garis oranye putus-putus (GD  $\alpha = 0.1$ ) tumpang tindih dengan sempurna, ini menunjukkan bahwa GD dengan *learning rate* 0.1 berhasil konvergen ke solusi optimal yang identik dengan LSM ( $y = 3941.07x + 10030.36$ ). Sebaliknya, garis hijau putus-putus (GD  $\alpha = 0.001$ ) menunjukkan *slope* yang lebih landai dan *intercept* yang lebih tinggi ( $y = 3259.69x + 16549.42$ ), yang secara visual tidak mewakili data asli seakurat dua model lainnya. Hal ini menandakan kalau *learning rate* 0.001 terlalu lambat dan belum mencapai konvergensi optimal. Titik prediksi (bintang hitam) untuk 9.25 km seharga Rp46,485 pun terletak tepat pada garis optimal LSM/GD 0.1, yang mengindikasikan prediksi dari model yang paling akurat.



Pada grafik di atas menunjukkan bahwa model Gradient Descent dengan *learning rate* 0.001 perlahan-lahan akan menyesuaikan nilai sehingga mendapatkan nilai *Mean Square Error* (MSE) akhir yang sama dengan model Gradient Descent dengan *learning rate* 0.1 serta model *Least Square Method* (LSM). Momen ini akan terjadi ketika epoch mencapai 100000 iterasi pertama seperti pada grafik di bawah.



Dengan adanya peristiwa ini menunjukkan bahwa ketiga model akan memiliki model regresi linear yang sama sehingga ketiga model memiliki regresi linear yang saling tumpang tindih seperti pada visualisasi di bawah ini.



## SOAL 2

Analisis ini bertujuan untuk mencari fitur mana yang korelasinya paling tinggi dengan target (Electricity Consumption (MWh)) dengan menggunakan correlation coefficient. Kemudian membangun model regresi linier dengan Least Square Method dari fitur dengan korelasi tertinggi tersebut dengan fiturnya guna memprediksi penggunaan listrik. Adapun data yang digunakan dapat dilihat pada gambar di bawah.

	TIME	Temperature (C)	Humidity (%)	Precipitation (l/m <sup>2</sup> )	Wind Speed (m/s)	Electricity Consumption (MWh)
0	2015-01-03 01:00:00	25.865259	0.018576	0.016174	21.850546	970.3450
1	2015-01-03 02:00:00	25.899255	0.018653	0.016418	22.166944	912.1755
2	2015-01-03 03:00:00	25.937280	0.018768	0.015480	22.454911	900.2688
3	2015-01-03 04:00:00	25.957544	0.018890	0.016273	22.110481	889.9538
4	2015-01-03 05:00:00	25.973840	0.018981	0.017281	21.186089	893.6865

### a. Correlation Coefficient

Correlation Coefficient adalah nilai yang mengukur hubungan antar dua variabel, dengan nilainya berada di kisaran  $-1$  sampai  $+1$ , Berikut equation dan kode yang digunakan pada penelitian kali ini.

$$CC = \frac{1}{N} \sum_{i=1}^N \frac{(X_i - \mu_X)(Y_i - \mu_Y)}{\sigma_X \sigma_Y}$$

```

1 def calculate_correlation(X, y):
2     if isinstance(X, pd.DataFrame):
3         feature_names = X.columns
4         X = X.values
5     else:
6         feature_names = [f'Feature_{i+1}' for i in range(X.shape[1])] if X.ndim > 1 else ['Feature_1']
7         X = np.array(X)
8
9     y = np.array(y, dtype=float)
10
11     print("Koefisien korelasi tiap fitur dengan target:")
12     print("-" * 45)
13
14     correlations = {}
15     if X.ndim == 1:
16         corr = np.corrcoef(X, y)[0, 1]
17         print(f"{feature_names[0]}: {corr:.6f}")
18         correlations[feature_names[0]] = corr
19     else:
20         for i, name in enumerate(feature_names):
21             corr = np.corrcoef(X[:, i], y)[0, 1]
22             print(f"{name}: {corr:.6f}")
23             correlations[name] = corr
24
25     return correlations

```

Berdasarkan perhitungan dari kode tersebut, didapat hasil sebagai berikut:

```

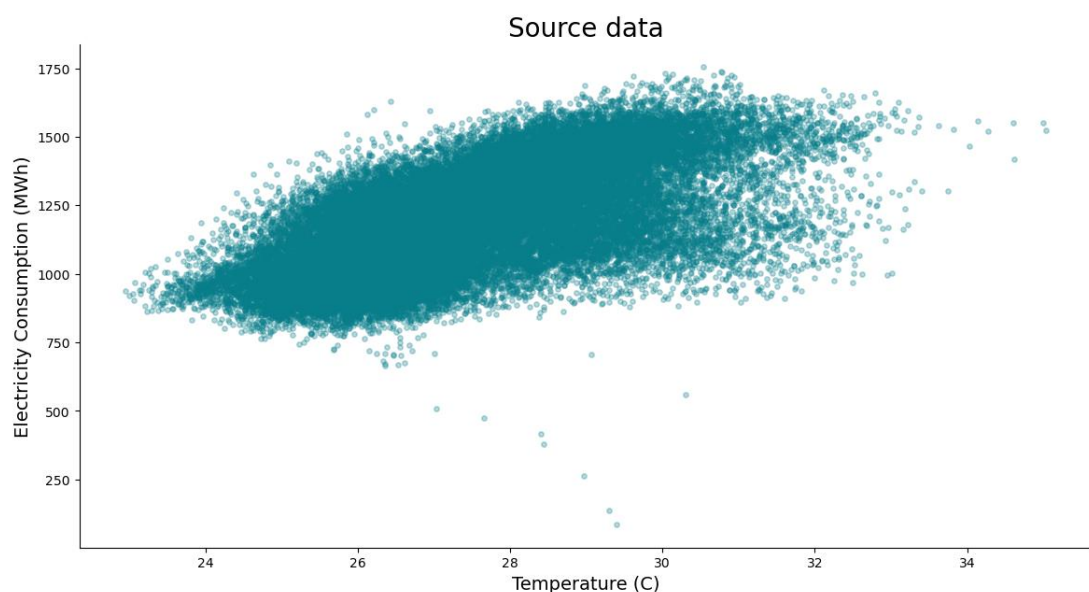
Koefisien korelasi tiap fitur dengan target:
-----
Temperature (C): 0.652811
Humidity (%): -0.036706
Precipitation (l/m^2): 0.073109
Wind Speed (m/s): 0.098435

```

Dapat dilihat bahwa fitur Temperature (C) memiliki korelasi paling tinggi jika dibandingkan dengan fitur lain, dengan nilai CC dari Temperature (C) adalah 0.652811.

## b. Machine Learning Model

Berdasarkan dari hasil CC di atas, maka akan digunakan Temperature (C) sebagai X dan Electricity Consumption (MWh) sebagai y, berikut adalah persebaran data antara Temperature (C) dan Electricity Consumption (MWh)





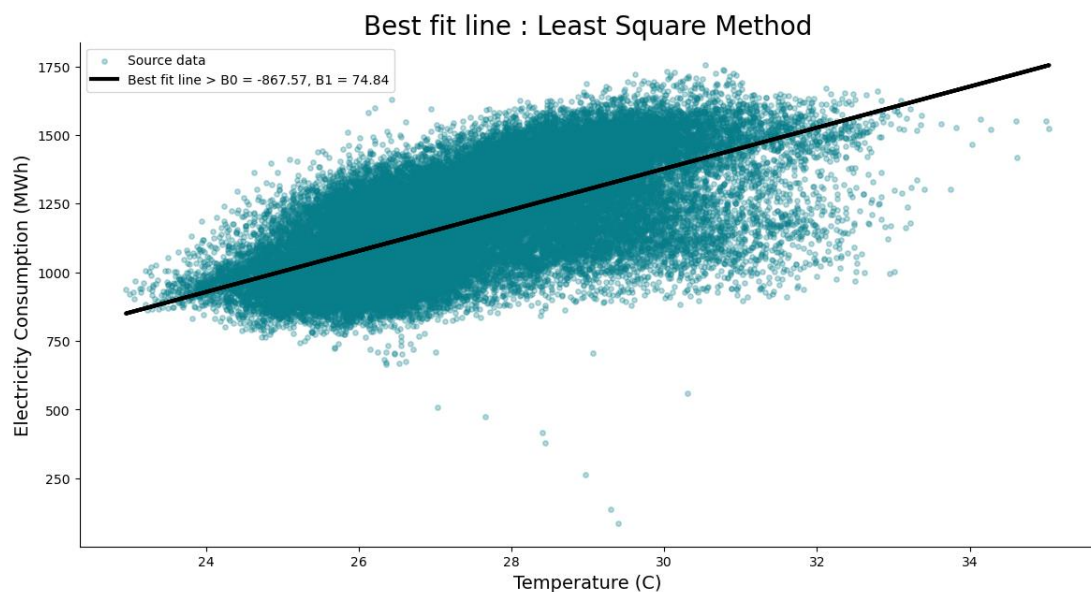
Setelah ini dapat kita implementasikan pada model linier regression dengan Least Square Method yang sama dengan nomor 1. Hasil perhitungannya didapatkan :

- ❖ Koefisien Regresi ( $W$ ) = -858.91
- ❖ Intercept ( $\alpha$ ) = 74.52

Dengan demikian, berdasarkan nilai yang telah didapatkan dari hasil perhitungan tersebut, model regresi linear yang dihasilkan dari metode Least Square Method adalah:

$$y = -858.91x + 74.52$$

Berikut adalah visualisasi implementasi model linier regression dengan Least Square Method dari Temperature (C) dengan Electricity Consumption (MWh) sebagai target.



Berdasarkan hasil dari model linier regression dengan Least Square Method menunjukkan bahwa semakin tinggi suhu maka penggunaan daya listrik juga meningkat, kemudian kita dapat mengevaluasi akurasi model dilakukan dengan metrik Root Mean Squared Error (RMSE). Adapun nilai RMSE pada model ini adalah 144.863 ini menunjukkan hasil dari prediksi model meleset 144.863 MWh. Ini menunjukkan model sudah cukup baik namun masih bisa di tingkatkan lagi, ini dapat terjadi karena penggunaan model yang sederhana yakni hanya dengan menggunakan 1 fitur saja. Korelasi dan suhu juga tidak benar-benar linier karena hanya ada di angka 0.65 saja. Ada banyak faktor lain yang mempengaruhi konsumsi daya listrik seperti waktu, hari libur, atau fitur lain yang tidak digunakan.