

# **Laporan Tugas Praktikum 07**



**Muhammad Bintang Gunawan - 0110224003**

Teknik Informatika, STT Terpadu Nurul Fikri, Depok

[0110224003@student.nurulfikri.ac.id](mailto:0110224003@student.nurulfikri.ac.id)

## Abstract

Pada praktikum ini bertujuan untuk membangun dan mengevaluasi model regresi menggunakan algoritma Support Vector Regression (SVR) untuk memprediksi Kandungan Nitrogen (N) dalam tanah berdasarkan fitur-fitur citra satelit.Tugas Mandiri

### 1. Berikut adalah kode berserta penjelasan dari praktikum Tugas Praktikum Mandiri.

```
▶ import pandas as pd
    import seaborn as sns
    import matplotlib.pyplot as plt
    from sklearn.preprocessing import LabelEncoder
    from sklearn.model_selection import train_test_split
    from sklearn.svm import SVC
    from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

    df = pd.read_csv("/content/drive/MyDrive/PRAKTIKUM ML/Praktikum07/Data/dataset_satelit.csv")
    print("Data berhasil dimuat!\n")
    print(df.head())
```

- pandas (pd): Digunakan untuk membaca, memproses, dan memanipulasi data tabular (seperti tabel atau spreadsheet).
- seaborn (sns) & matplotlib.pyplot (plt): Digunakan untuk membuat visualisasi data.
- sklearn (Scikit-learn): Library utama untuk Machine Learning.
- LabelEncoder: Untuk mengubah label teks menjadi format numerik.
- train\_test\_split: Untuk membagi data menjadi set pelatihan dan set pengujian.
- SVC: Model Support Vector Classification (klasifikasi).
- accuracy\_score, classification\_report, confusion\_matrix: Metrik untuk mengevaluasi kinerja model.

```

*** Data berhasil dimuat!

      No   Longitude  Lattitude    N     P     K     Ca    Mg     Fe     Mn  \
0    1  103.036658 -0.604417  2.64  0.15  0.415  0.51  0.31  119.96  463.23
1    2  103.037201 -0.604689  2.75  0.17  0.568  0.76  0.58  102.63  493.81
2    3  103.036359 -0.603012  1.77  0.12  0.339  0.49  0.6   107.37  460.93
3    4  103.036950 -0.603219  2.30  0.15  0.460  0.74  0.67  96.02  338.17
4    5  103.036802 -0.601969  2.05  0.14  0.308  0.64  0.72  87.01  384.33

      ...      b1  Sigma_VV  Sigma_VH     plia      lia     iafe  gamma0_vv  \
0 ... 0.0433  0.18183  0.04461  35.74446  35.79744  35.41161  0.22331
1 ... 0.0465  0.22079  0.04640  35.12096  35.14591  35.41510  0.27116
2 ... 0.0417  0.18926  0.03992  35.07724  35.07730  35.41135  0.23242
3 ... 0.0367  0.14769  0.03622  36.08078  36.08469  35.41583  0.18138
4 ... 0.0361  0.18205  0.03797  32.68855  32.69293  35.41592  0.22359

      gamma0_vh  beta0_vv  beta0_vh
0  0.05479  0.31325  0.07686
1  0.05699  0.38033  0.07993
2  0.04902  0.32604  0.06876
3  0.04448  0.25440  0.06238
4  0.04664  0.31359  0.06541

```

Tabel yang ditampilkan adalah lima baris pertama (`df.head()`) dari *dataset* tersebut, yang berisi banyak kolom (variabel).

- Kolom Geografis: Longitude (bujur) dan Lattitude (lintang).
- Kolom Kimia Tanah: N, P, K, Ca, Mg, Fe, Mn (mungkin kandungan unsur hara).
- Kolom Satelit/Penginderaan Jauh: Kolom dengan nama teknis seperti b1, Sigma VV, Sigma VH, plia, iafe, dan gamma0 (mungkin hasil pengukuran dari citra satelit seperti SAR atau optik).

```

df.columns

Index(['No', 'Longitude', 'Latitude', 'N', 'P', 'K', 'Ca', 'Mg', 'Fe', 'Mn',
       'Cu', 'Zn', 'B', 'b12', 'b11', 'b9', 'b8a', 'b8', 'b7', 'b6', 'b5',
       'b4', 'b3', 'b2', 'b1', 'Sigma_VV', 'Sigma_VH', 'plia', 'lia', 'iafe',
       'gamma0_vv', 'gamma0_vh', 'beta0_vv', 'beta0_vh'],
      dtype='object')

df["N"].unique()

array([2.64, 2.75, 1.77, 2.3 , 2.05, 2.2 , 1.95, 2.35, 2.34, 2.72, 2.59,
       2.63, 2.83, 2.41, 2.26, 2.42, 1.96, 2.31, 2.39, 2.22, 2.27, 2.84,
       2.52, 2.81, 2.47, 2.36, 2.95, 2.69, 2.51, 2.12, 2.46, 2.55, 2.09,
       2.07, 2.13, 2.02, 2.57, 2.49, 2.4 , 2.15, 2.25, 2.06, 1.89, 2.29,
       2.28, 2.11, 1.9 , 2.5 , 2.19, 2.03, 2.1 , 2.01, 2.23, 1.91, 2.45,
       2.87, 2.7 , 3.02, 2.43, 2.74, 2.16, 1.8 , 2.24, 2.6 , 2.37, 2.21,
       2.04, 2.53, 2.71, 1.78, 2.67, 2.61, 2.56, 2.14, 2.32, 2.77, 2.79,
       2.17, 3.13, 2.62, 3.12, 2.44, 2.89, 2. , 1.85, 1.98, 2.48, 1.87,
       2.86, 2.73, 2.68, 2.38, 1.71, 2.18, 2.66, 2.9 , 2.54, 2.82, 1.94,
       1.92, 1.79, 1.52, 1.99, 1.5 , 1.76, 1.72, 1.56, 1.73, 1.41, 1.83,
       1.39, 1.53, 1.66, 1.61, 1.7 , 1.57, 1.47, 1.67, 1.35, 1.82, 1.43,
       1.68, 1.69, 1.3 , 1.49, 1.75, 1.6 , 2.08, 1.88, 1.97, 1.84, 1.81,
       1.93, 1.58, 1.37, 1.44, 1.86, 1.48, 1.29, 1.45, 1.34, 1.54, 1.55,
       1.63, 1.14, 1.28, 1.42, 1.74, 2.58, 2.33, 2.93, 2.94, 2.96, 2.8 ,
       2.78, 2.85, 3.06, 2.92, 3.03, 3.23, 2.88])

```

`df.columns` menampilkan semua nama kolom (features) yang ada di dalam DataFrame (df).

`df['N'].unique()` mengambil semua nilai unik yang ada di dalam kolom spesifik 'N'.

```

X = df[['b1', 'Sigma_VV', 'Sigma_VH', 'plia', 'lia', 'iafe',
        'gamma0_vv', 'gamma0_vh', 'beta0_vv', 'beta0_vh']]

y = df['N']

```

`X = df[['b1', 'Sigma_VV', 'Sigma_VH', 'plia', 'lia', 'iafe',
 'gamma0_vv', 'gamma0_vh', 'beta0_vv', 'beta0_vh']]`

membuat variabel baru bernama X yang akan bertindak sebagai Variabel Independen atau Fitur (*Features*)

`y = df['N']` membuat variabel baru bernama y yang akan bertindak sebagai Variabel Dependental atau Target (Target Variable)

▶ x.head()

	b1	Sigma_VV	Sigma_VH	plia	lia	iafe	gamma0_vv	gamma0_vh	beta0_vv	beta0_vh
0	0.0433	0.18183	0.04461	35.74446	35.79744	35.41161	0.22331	0.05479	0.31325	0.07686
1	0.0465	0.22079	0.04640	35.12096	35.14591	35.41510	0.27116	0.05699	0.38033	0.07993
2	0.0417	0.18926	0.03992	35.07724	35.07730	35.41135	0.23242	0.04902	0.32604	0.06876
3	0.0367	0.14769	0.03622	36.08078	36.08469	35.41583	0.18138	0.04448	0.25440	0.06238
4	0.0361	0.18205	0.03797	32.68855	32.69293	35.41592	0.22359	0.04664	0.31359	0.06541

▶ y.head()

	N
0	2.64
1	2.75
2	1.77
3	2.30
4	2.05

**dtype:** float64

menampilkan 5 baris pertama dari *Dataframe* X dan y yang telah Anda definisikan sebelumnya.

```

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVR
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = SVR(kernel='linear')
model.fit(X_train, y_train)

```

▼      SVR       ⓘ ⓘ ?  
 SVR(kernel='linear')

mempersiapkan data dan melatih model regresi menggunakan algoritma *Support Vector Regression* (SVR) untuk memprediksi Kandungan Nitrogen (y) berdasarkan fitur satelit (X).

```

▶ from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

y_pred = model.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print("Hasil Evaluasi Model SVR:")
print(f"MAE : {mae:.3f}")
print(f"MSE : {mse:.3f}")
print(f"RMSE : {rmse:.3f}")
print(f"R² : {r2:.3f}")

...

```

Hasil Evaluasi Model SVR:

```

MAE : 0.228
MSE : 0.081
RMSE : 0.285
R² : 0.464

```

- `y_pred`: Model SVR (model) digunakan untuk membuat prediksi terhadap data pengujian fitur (`X_test`). Hasil prediksi disimpan dalam variabel `y_pred`.
- MAE, MSE, RMSE, R<sup>2</sup>: Selanjutnya, metrik-metrik regresi dihitung dengan membandingkan nilai aktual (`y_test`) dan nilai prediksi (`y_pred`). Metrik ini mengukur tingkat kesalahan dan seberapa baik model mencocokkan data.

```

import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(7, 5))
sns.scatterplot(x=y_test, y=y_pred, color="dodgerblue", alpha=0.6)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], color="red", linestyle="--")
plt.xlabel("Nilai Aktual")
plt.ylabel("Nilai Prediksi")
plt.title("Perbandingan Nilai Aktual vs Prediksi")
plt.show()

residuals = y_test - y_pred
plt.figure(figsize=(7, 5))
sns.histplot(residuals, bins=20, kde=True, color="orange")
plt.xlabel("Error (Residual)")
plt.ylabel("Frekuensi")
plt.title("Distribusi Error (Residuals)")
plt.show()

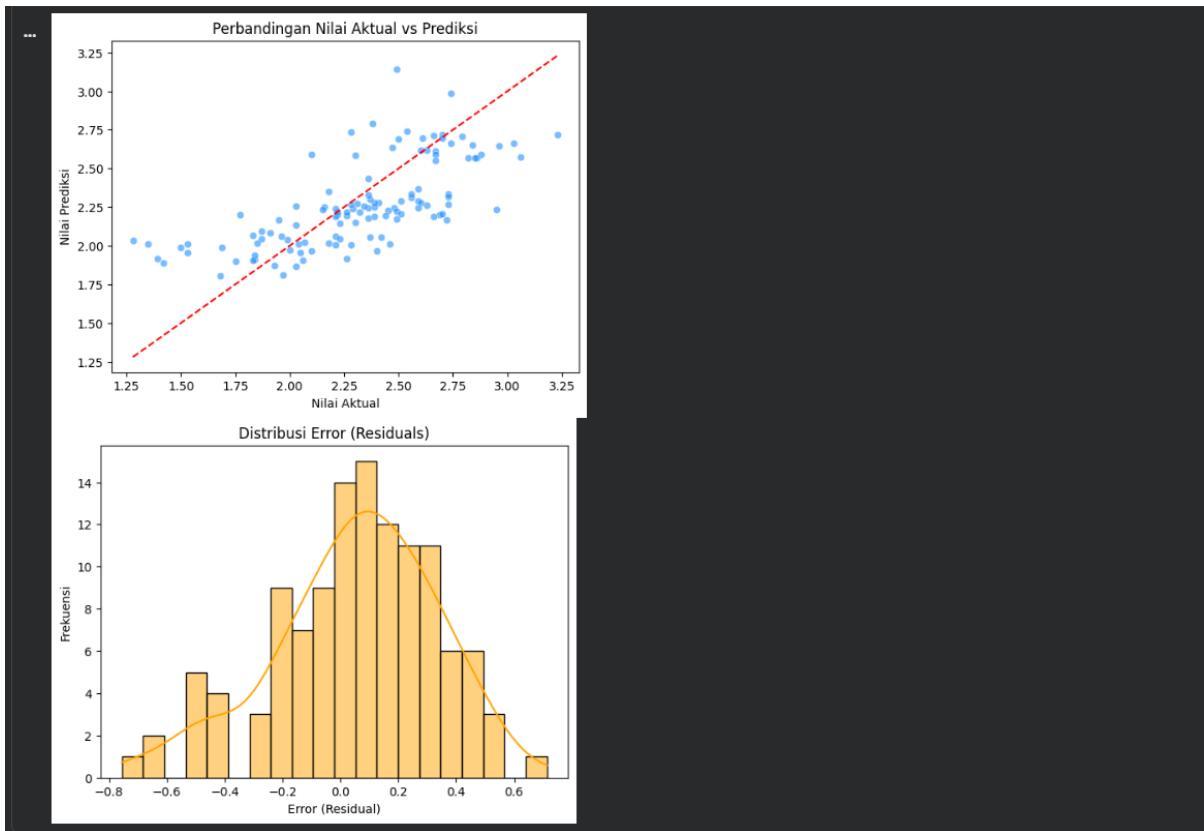
```

## 1. Scatter Plot Perbandingan Aktual vs Prediksi

- Membuat *scatter plot* untuk melihat seberapa dekat nilai prediksi ( $y_{pred}$ ) dengan nilai aktual ( $y_{test}$ ).
- Titik Biru: Menunjukkan setiap pasangan data aktual ( $x-axis$ ) dan prediksi ( $y-axis$ ).
- Garis Merah Putus-putus: Menarik garis ideal  $y=x$ , di mana setiap titik seharusnya berada jika prediksi sempurna (Nilai Aktual = Nilai Prediksi).

## 2. Histogram Distribusi

- Tujuan Kode: Menghitung selisih antara nilai aktual dan prediksi ( $residuals = y_{test} - y_{pred}$ ) dan memvisualisasikan distribusinya.
- Histogram Oranye: Menunjukkan frekuensi (ketinggian batang) dari nilai-nilai kesalahan (Error).
- KDE (Garis Lengkung): Adalah perkiraan kurva distribusi normal.



1. Sebagian besar titik biru mengumpul di sekitar garis merah, menunjukkan bahwa model SVR cukup baik dalam memprediksi nilai Kandungan Nitrogen.
  - Penyebaran titik di sekitar garis menunjukkan adanya kesalahan prediksi (*residual*). Distribusi error terpusat di sekitar nol, dan bentuknya mirip dengan kurva normal (berbentuk lonceng).
    - Interpretasi: Ini adalah kondisi ideal untuk model regresi yang baik. Distribusi error yang normal dan terpusat di nol menunjukkan bahwa model tidak secara sistematis melebih-lebihkan atau meremehkan prediksi.

Link GitHub:

[https://github.com/XiaoMao15/ML\\_Praktikum-dan-Tugas.git](https://github.com/XiaoMao15/ML_Praktikum-dan-Tugas.git)