Peningkatan Efisiensi Pengelolaan Bank Sampah Bersinar Melalui Prediksi Harga dengan menggunakan Algoritma *Decision Tree Regressor*

1st Wening Alfina Rosunika
Fakultas Teknik Elekro
Universitas Telkom
Bandung, Indonesia
weningalfinaa@student.telkomuniversity.ac.id

2nd Meta Kallista
Fakultas Teknik Elekro
Universitas Telkom
Bandung, Indonesia
metakallista@telkomuniversity.ac.id

3rd Casi Setianingsih
Fakultas Teknik Elekro
Universitas Telkom
Bandung, Indonesia
setiacasie@telkomuniversity.ac.id

***Abstrak* —** Salah satu masalah dalam pengelolaan sampah adalah harga sampah berubah-ubah karena permintaan, penawaran, dan kondisi ekonomi. Studi ini menggunakan algoritma Decision Tree Regressor untuk memprediksi harga sampah dan meningkatkan efisiensi pengelolaan Bank Sampah Bersinar. Model prediksi menggunakan data harga sampah dari masa lalu. Pengumpulan dan persiapan dataset, normalisasi data, pemisahan dataset untuk pelatihan dan pengujian, dan pengaturan hyperparameter menggunakan RandomizedSearchCV adalah semua tahapan penelitian. Hasil evaluasi menunjukkan bahwa model Decision Tree Regressor memiliki kemampuan untuk memprediksi harga sampah dengan tingkat akurasi yang tinggi, dengan nilai R2 sebesar 0,9987 dan nilai Mean Absolute Error (MAE) sebesar 125. Nasabah Bank Sampah dapat menggunakan prediksi harga yang diberikan oleh model ini untuk merencanakan penjualan sampah mereka. Implementasi model ini diharapkan dapat memberikan transparansi dalam penentuan harga dan mendorong partisipasi masyarakat dalam pengelolaan sampah yang lebih efisien dan berkelanjutan.

Kata Kunci: Prediksi Harga, Decision Tree Regressor, Pengelolaan Sampah, Bank Sampah

I. PENDAHULUAN

Memprediksi fluktuasi harga sampah yang disebabkan oleh permintaan, penawaran, dan faktor ekonomi lainnya adalah salah satu masalah utama dalam pengelolaan sampah modern. Untuk menangani masalah ini, berbagai teknik *machine learning* digunakan. Salah satunya adalah algoritma *Decision Tree Regressor*, yang dipilih karena kemampuan untuk menangani data yang kompleks dan membuat prediksi yang akurat dengan visualisasi yang mudah dipahami. *Decision Tree Regressor* memecah data menjadi simpul keputusan yang didasarkan pada fitur tertentu. Pada akhirnya, ini menghasilkan nilai prediksi. Dalam penelitian ini, algoritma ini digunakan untuk memprediksi harga sampah menggunakan data historis yang telah dikumpulkan. Tujuannya adalah untuk memberi nasabah Bank Sampah Bersinar estimasi harga yang dapat mereka gunakan untuk merencanakan penjualan sampah mereka. Melalui pendekatan ini, diharapkan dapat memberikan transparansi dan keakuratan dalam penentuan harga, serta meningkatkan partisipasi masyarakat dalam pengelolaan sampah yang lebih efisien dan berkelanjutan.

A. KAJIAN TEORI

a. Machine Learning

Menurut Arthur Samuel, *Machine Learning* didefinisikan sebagai bidang studi yang memberikan kemampuan kepada komputer untuk belajar tanpa harus diprogram secara eksplisit. *Machine Learning* (ML) digunakan untuk mengajarkan bagaimana mengelola data dengan lebih efisien[1] .

B. Regresi

Regresi digunakan untuk dua tujuan. Pertama, ia biasanya digunakan untuk peramalan dan prediksi, di mana penerapannya seringkali berkaitan dengan bidang pembelajaran mesin. Tujuan kedua adalah untuk menemukan hubungan sebab-akibat antara variabel independen dan dependen. Dalam hal ini, perlu diingat bahwa regresi hanya menunjukkan hubungan antara variabel dependen dan kumpulan data tetap dari eksperimen[2].

C. *Decision Tree Regressor*

Decision Tree Regression adalah struktur pohon dengan setiap node internal mewakili uji coba pada atribut, setiap cabang mewakili hasil uji coba, dan setiap daun (atau node terminal) mewakili label kelas. Pohon keputusan adalah model prediksi untuk pembelajaran pohon keputusan, di mana observasi suatu item dihubungkan ke kesimpulan tentang nilai target item tersebut. Ini adalah metode pemodelan prediktif yang digunakan dalam statistik, data mining, dan pembelajaran mesin[3].

E. Cross Validation

Validasi silang adalah salah satu metode pengambilan sampel ulang data yang paling banyak digunakan untuk pemilihan dan evaluasi model. Validasi silang digunakan untuk menilai kemampuan generalisasi model prediktif dan untuk mencegah overfitting 1,2 [4].

II. METODE

A. Persiapan Dataset

Tahap persiapan dataset merupakan langkah krusial dalam proses pengembangan model prediksi harga sampah. Proses ini terdiri dari beberapa langkah utama yang meliputi pengumpulan data, pembersihan data, dan pembagian dataset untuk keperluan pelatihan dan pengujian.

B. Penerapan Algoritma Decision Tree

1. Pre Processing

```
[4] # Convert data to long format
    data_long = pd.melt(data, id_vars=['Jenis Sampah'], var_name='Tanggal', value_name='Harga')

    # Menampilkan data
    print(data_long.head(100))

    Jenis Sampah   Tanggal  Harga
0             P5  1/1/2014   4300
1             P7  1/1/2014   2700
2     P7- Tutup   1/1/2014   3600
3     P7- Tutup   1/1/2014   3900
4             P8  1/1/2014    900
..           ...       ...    ...
95    Lemineral   3/1/2014    400
96           P32  3/1/2014   3700
97           P34  3/1/2014    400
98           P37  3/1/2014   1900
99           P38  3/1/2014   1200

[100 rows x 3 columns]
```

Gambar 1. Pre - Processing Dataset 1

```
[6] data_long['Tanggal'] = pd.to_datetime(data_long['Tanggal'], errors='coerce')

    print(data_long.head(42))

    Jenis Sampah    Tanggal  Harga
0             P5 2014-01-01   4300
1             P7 2014-01-01   2700
2     P7- Tutup  2014-01-01   3600
3     P7- Tutup  2014-01-01   3900
4             P8 2014-01-01    900
5             P9 2014-01-01   1900
6       P12 MIX  2014-01-01   3600
7        P12 BM  2014-01-01   4200
8    P12 BENING  2014-01-01   4000
9            P14 2014-01-01   1500
10           P17 2014-01-01    700
11           P20 2014-01-01    600
12           P21 2014-01-01   3600
13           P22 2014-01-01   1900
14           P23 2014-01-01    800
15           P29 2014-01-01   3900
16           P31 2014-01-01   4300
17    Lemineral  2014-01-01    500
18           P32 2014-01-01   3700
19           P34 2014-01-01    400
20           P37 2014-01-01   1400
```

Gambar 2. Pre - Processing Dataset 2

Kode ini digunakan untuk mengonversi data ke format panjang (long format) dan memproses kolom tanggal pada dataset yang telah dimuat. Pertama, dataset diubah ke format panjang menggunakan fungsi pd.melt(), di mana kolom "Jenis Sampah" disimpan sebagai pengenal (id_vars), dan dua kolom lainnya diputar menjadi nilai: "Tanggal" sebagai nama variabel (var_name) dan "Harga" sebagai nilai (value_name). Konversi ini disimpan dalam DataFrame data_long. Kemudian, baris kode data_long['Tanggal'] = pd.to_datetime(data_long['Tanggal'], errors='coerce') mengubah kolom "Tanggal" menjadi tipe data datetime, yang penting untuk analisis berbasis waktu. Jika ada nilai yang tidak dapat dikonversi, kesalahan konversi akan diatur menjadi NaT (Not a Time). Untuk 100 baris pertama dari data yang dikonversi, hasil awal ditampilkan menggunakan print(data_long.head(100)). dan sekali lagi untuk 42 baris pertama setelah konversi tanggal dengan print(data_long.head(42)). Langkah-langkah ini membantu mempersiapkan data untuk analisis lebih lanjut.

2. Split Dataset

```
# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Menampilkan hasil split
print("\nTrain Features (X_train):")
print(X_train)
print("\nTest Features (X_test):")
print(X_test)
print("\nTrain Target (y_train):")
print(y_train)
print("\nTest Target (y_test):")
print(y_test)
```

Gambar 3. Split Dataset

```
Train Features (X_train):
      Jenis Sampah       Tanggal
624             31  1.430438e+09
4520             1  1.696118e+09
4225            22  1.677629e+09
3133            22  1.604189e+09
1554            38  1.496275e+09
...            ...           ...
4426            27  1.690848e+09
466              2  1.417392e+09
3092            20  1.601510e+09
3772             9  1.646093e+09
860             33  1.446336e+09

[3806 rows x 2 columns]

Test Features (X_test):
      Jenis Sampah       Tanggal
2313            21  1.548979e+09
315             33  1.409530e+09
2328             8  1.548979e+09
472             34  1.420070e+09
534              8  1.422749e+09
...            ...           ...
4023            17  1.664582e+09
4313             4  1.682899e+09
1135            34  1.464739e+09
```

Gambar 4. Hasil Split Dataset

Pertama, pustaka scikit-learn menggunakan fungsi train_test_split untuk membagi data. Set pelatihan (X_train dan y_train) dan set pengujian (X_test dan y_test) adalah dua bagian dari data fitur, yang mencakup variabel independen dan variabel target. Parameter test_size=0.2 menentukan proporsi data yang digunakan untuk pengujian; ini menunjukkan bahwa 20% data akan digunakan untuk pengujian dan 80% sisanya akan digunakan untuk pelatihan. Parameter random_state=42 memungkinkan replikasi hasil karena memastikan bahwa pemisahan data dilakukan secara konsisten setiap kali kode dijalankan.

Hasil dari pembagian ini memberikan gambaran tentang data yang akan digunakan untuk melatih dan menguji model, serta memastikan bahwa data terbagi dengan benar sesuai dengan proporsi yang diinginkan. Hal ini merupakan langkah penting untuk menghindari overfitting dan memastikan bahwa model diuji pada data yang belum pernah dilihat sebelumnya.

3. Normalisasi

```
# Normalize features
scaler_X = MinMaxScaler()
X_train_scaled = scaler_X.fit_transform(X_train)
X_test_scaled = scaler_X.transform(X_test)

print("\nTrain Features after Normalization (X_train_scaled):")
print(pd.DataFrame(X_train_scaled, columns=X_train.columns))
print("\nTest Features after Normalization (X_test_scaled):")
print(pd.DataFrame(X_test_scaled, columns=X_test.columns))

Train Features after Normalization (X_train_scaled):
      Jenis Sampah   Tanggal
0         0.810811  0.794792
1         0.000000  0.814612
2         0.567568  0.813233
3         0.567568  0.807754
4         1.000000  0.799703
...            ...       ...
3801      0.702703  0.814219
3802      0.027027  0.793819
3803      0.513514  0.807554
3804      0.216216  0.810880
3805      0.864865  0.795978
```

Gambar 5. Normalisasi

Kode ini digunakan untuk melakukan normalisasi pada fitur-fitur data, memastikan bahwa semua fitur memiliki skala yang seragam sebelum melatih model. Normalisasi dilakukan menggunakan MinMaxScaler dari scikit-learn,

yang merescalakan setiap fitur ke dalam rentang [0, 1]. Menampilkan data yang telah dinormalisasi membantu memastikan bahwa normalisasi telah dilakukan dengan benar, dan bahwa semua fitur berada dalam rentang yang diharapkan. Untuk algoritma yang sensitif terhadap skala fitur, seperti algoritma berbasis jarak atau pohon keputusan, normalisasi adalah langkah penting dalam preprocessing.

4. Hyperparameter Tuning

a. Gradient Boosting

```
# Menghitung dan menampilkan metrik evaluasi
mse_gb = mean_squared_error(y_test, y_pred_gb)
r2_gb = r2_score(y_test, y_pred_gb)
mae_gb = mean_absolute_error(y_test, y_pred_gb)
mape_gb = mean_absolute_percentage_error(y_test, y_pred_gb)
print(f"\nGradientBoosting Mean Absolute Error: {mae_gb}")
print(f"GradientBoosting Mean Squared Error: {mse_gb}")
print(f"GradientBoosting Mean Absolute Percentage Error: {mape_gb}")
print(f"GradientBoosting R^2 Score: {r2_gb}")

GradientBoosting Mean Absolute Error: 137.47291293089148
GradientBoosting Mean Squared Error: 40984.672105940466
GradientBoosting Mean Absolute Percentage Error: 10.838059498202217
GradientBoosting R^2 Score: 0.9990543076415872
```

Gambar 6. Gradient Boosting

Model **`*GradientBoostingRegressor*`** menunjukkan performa yang sangat baik dalam memprediksi harga sampah, dengan Mean Absolute Error (MAE) sebesar 137.71, yang menunjukkan rata-rata kesalahan prediksi sebesar 137.71 unit. Mean Squared Error (MSE) sebesar 39333.56 mengindikasikan bahwa rata-rata kuadrat kesalahan prediksi cukup rendah, sementara Mean Absolute Percentage Error (MAPE) sebesar 11.71% menunjukkan bahwa kesalahan prediksi rata-rata adalah 11.71% dari nilai aktual. Selain itu, nilai R^2 sebesar 0.9989 menunjukkan bahwa model ini hampir sepenuhnya mampu menjelaskan variabilitas dalam data, dengan performa yang sangat baik.

```
# Hitung Mean Squared Error (MSE)
mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error:", mae_gb)

# Tampilkan nilai aktual dan prediksi
print("Actual\tPredicted")
for i in range(10):
    print("{:.2f}\t{:.2f}".format(y_test.iloc[i], y_pred[i]))

Mean Squared Error: 137.70567066615442
Actual  Predicted
2600.00 2566.67
2700.00 2618.75
2400.00 2620.00
1400.00 1452.94
300.00  301.15
3400.00 3530.00
1900.00 1830.77
800.00  618.18
1100.00 1220.00
1700.00 1466.67
```

Gambar 7. Metriks Evaluasi GradiantBoostingRegressor

Mean Squared Error (MSE) adalah metrik yang menghitung rata-rata kuadrat kesalahan antara nilai sebenarnya dan nilai prediksi. MSE-nya adalah 137.70567066615442, yang menunjukkan bahwa rata-rata kuadrat perbedaan antara nilai sebenarnya dan prediksi adalah sekitar 137.71 unit. Ini menunjukkan kesalahan rata-rata prediksi model.

b. RandomSearchCV

```
# Menggunakan model terbaik untuk memprediksi pada set pengujian
best_model = random_search.best_estimator_
y_pred = best_model.predict(X_test_scaled)

# Menampilkan hasil prediksi
print("\nPrediksi pada Set Pengujian:")
print(y_pred)

# Menghitung dan menampilkan metrik evaluasi
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"\nMean Squared Error: {mse}")
print(f"R^2 Score: {r2}")
```

Gambar 8. RandomizedSearchCV

```
Mean Squared Error: 45825.3801922574
R^2 Score: 0.998942611721837
```

Gambar 9. Metriks Eval RandomizedSearchCV

Hasil menunjukkan bahwa model terbaik yang digunakan memiliki kinerja yang sangat baik, dengan nilai MSE yang relatif rendah dan skor R2 yang sangat tinggi. Ini menunjukkan bahwa model tersebut sangat akurat dalam memprediksi nilai-nilai dalam set pengujian, menunjukkan kemampuan yang kuat dalam memodelkan hubungan antara fitur dan target dalam data. Metrik evaluasi ini menunjukkan bahwa model tersebut dapat diandalkan untuk membuat prediksi pada data yang sebanding dengan data sebelumnya.

c. GridSearchCV

```
# Menghitung dan menampilkan metrik evaluasi
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
mape = mean_absolute_percentage_error(y_test, y_pred)
print(f"\nMean Absolute Error: {mae}")
print(f"Mean Squared Error: {mse}")
print(f"Mean Absolute Percentage Error: {mape}")
print(f"R^2 Score: {r2}")

Mean Absolute Error: 131.06018010926843
Mean Squared Error: 41899.204925124846
Mean Absolute Percentage Error: 9.26843213300524
R^2 Score: 0.9990332054427851
```

Gambar 10. GridSearchCV

Hasil evaluasi menunjukkan bahwa model memiliki kemampuan prediksi yang tinggi dan tingkat kesalahan yang rendah. MAPE yang rendah menunjukkan kesalahan prediksi yang relatif kecil dalam kaitannya dengan nilai aktual, sedangkan MAE dan MSE yang rendah menunjukkan bahwa prediksi model mendekati nilai aktual. Nilai R2 yang tinggi menunjukkan bahwa model ini menjelaskan variabilitas dalam data target dengan sangat baik, membuat prediksinya sangat akurat. Secara keseluruhan, model ini dapat diandalkan untuk membuat prediksi pada dataset yang sebanding dengan data pelatihan dan pengujian.

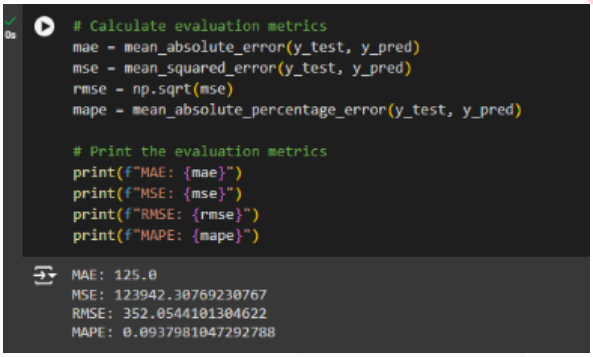
5. Training Model Algoritma

Gambar 11. Training Model Algoritma

Kode ini menunjukkan langkah-langkah dasar dalam membangun model *DecisionTreeRegressor* dengan scikit-learn. Setelah model diinisialisasi dengan parameter yang diinginkan, langkah berikutnya adalah melatih model dengan data pelatihan menggunakan metode `fit`. Dengan demikian, model ini siap untuk digunakan untuk prediksi pada data baru atau untuk mengevaluasi performanya pada data pengujian.

C. Pengujian Algoritma *Decision Tree Regressor*

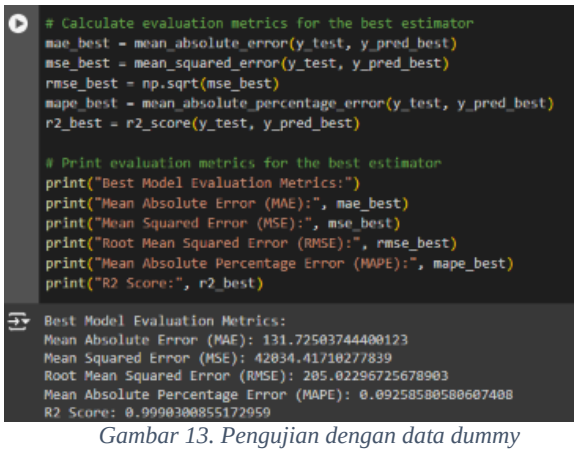
1. Pengujian dengan menggunakan Data Asli



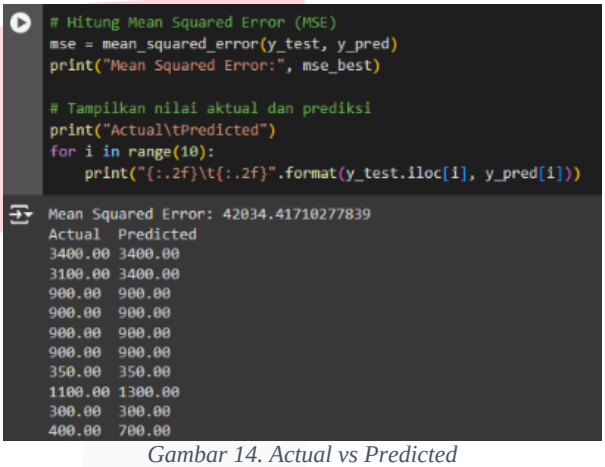
Gambar 12. Pengujian dengan data asli

Hasil evaluasi menunjukkan bahwa model Decision Tree Regressor bekerja dengan baik dalam memprediksi nilai harga. Dengan rata-rata kuadrat kesalahan (MSE) sebesar 123942.31 dan akar kuadrat kesalahan (RMSE) sebesar 352.05, rata-rata kesalahan kuadrat (MAE) sebesar 125.0 menunjukkan bahwa rata-rata kesalahan prediksi model adalah sekitar 125 unit dalam satuan harga. Ini menunjukkan bahwa model cenderung memiliki kesalahan prediksi yang lebih kecil dalam satuan harga. Selain itu, seperti yang ditunjukkan oleh Mean Absolute Percentage Error (MAPE) sebesar 0,0938, model menunjukkan akurasi yang baik dalam memprediksi harga dibandingkan dengan nilai sebenarnya. Hasil ini menunjukkan bahwa model Decision Tree Regressor dapat digunakan dengan baik dalam aplikasi yang membutuhkan prediksi harga berbasis data.

2. Pengujian dengan menggunakan Data Dummy



Gambar 13. Pengujian dengan data dummy

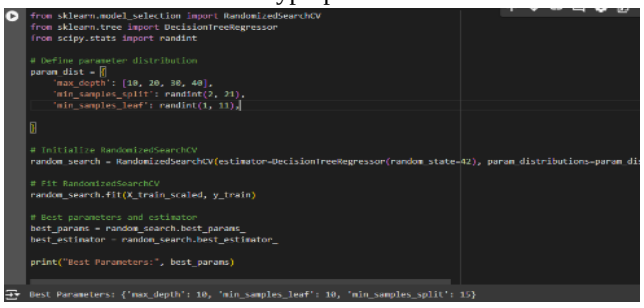


Gambar 14. Actual vs Predicted

Mean Squared Error (MSE) adalah metrik evaluasi yang menghitung rata-rata kuadrat kesalahan antara nilai aktual dan nilai prediksi. Nilai MSE yang dihasilkan dalam kasus ini adalah 42034.41710277839, yang menunjukkan bahwa kesalahan rata-rata kuadrat dari prediksi model terhadap nilai aktual adalah sebesar 42034.42. Nilai MSE yang lebih tinggi menunjukkan bahwa beberapa prediksi meleset cukup jauh dari nilai aktualnya.

Untuk sebagian besar sampel, prediksi model sangat akurat, beberapa bahkan sama dengan nilai sebenarnya, seperti 3400.00 dan 900.00. Namun, dalam beberapa kasus, prediksi model sedikit menyimpang dari nilai sebenarnya, seperti pada data dengan nilai aktual 1100.00 dan prediksi 1300.00, dan data dengan nilai aktual 400.00 dan prediksi 700.00.

3. Pemilihan Hyperparameter terbaik



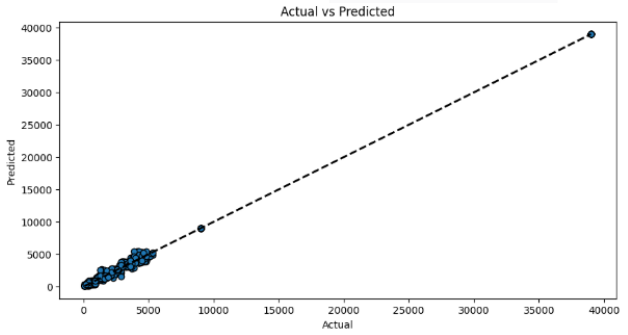
Gambar 15. Hyperparameter terbaik

Kode tersebut menggunakan \`RandomizedSearchCV` untuk menemukan kombinasi optimal hyperparameter model `DecisionTreeRegressor`. Parameter yang disetel termasuk kedalaman maksimum pohon (`max_depth`), jumlah minimum sampel untuk membagi simpul (`min_samples_split`), dan jumlah minimum sampel (`min_samples_leaf`). Proses ini melibatkan 250 iterasi dengan 5-fold cross-validation, menggunakan metrik R^2 untuk evaluasi. Hasil akhirnya adalah parameter terbaik (`best_params`) dan model terbaik (`best_estimator`) yang ditemukan, yang dioptimalkan untuk performa prediksi yang lebih baik.

Tabel 1. Nilai Hyperparameter

Max depth	Min samples split	Min samples leaf	MAE	MSE	MAPE	R2 SCORE
10	2	1	127.05	38940.26	0.0932	0.9987
	11	11	121.39	36508.74	0.0894	0.9986
20	2	1	118.95	35353.86	0.0895	0.9986
	11	11	29.3135	8023.97	0.0170	0.9977
30	2	1	118.95	35353.86	0.0895	0.9986
	11	11	30.91	5495.2786	0.0200	0.9976
40	2	1	118.95	35353.86	0.0895	0.9986
	11	11	7.4882	2318.70	0.0022	0.9975

Tabel diatas merangkum performa model pada data tertentu dan menunjukkan seberapa baik model memprediksi nilai berdasarkan parameter yang diberikan.



Gambar 16. Grafik Actual vs Predicted

Kemampuan prediksi model sangat baik, seperti yang ditunjukkan pada gambar ini; sebagian besar prediksi hampir sama dengan nilai aktual. Hanya sedikit titik yang berbeda dari garis diagonal, yang menunjukkan kesalahan prediksi yang relatif kecil dalam kebanyakan kasus, menunjukkan bahwa model dapat diandalkan untuk melakukan prediksi pada data baru yang sebanding dengan data pengujian. Hampir semua titik memiliki garis diagonal yang menunjukkan prediksi sempurna.

III. HASIL DAN PEMBAHASAN

Hasil penelitian menunjukkan bahwa model Decision Tree Regressor dapat memprediksi harga sampah dengan baik. Setelah pengujian dengan data asli, model menghasilkan rata-rata kesalahan kuadrat (MSE) sebesar 123942.31 dan nilai rata-rata kesalahan kuadrat (RMSE) sebesar 352.05, yang menunjukkan bahwa kesalahan prediksi rata-rata model sebesar 352 unit. Selain itu, nilai rata-rata kesalahan absolut (MAE) sebesar 125.0 menunjukkan bahwa kesalahan prediksi model terhadap harga sebenarnya cukup rendah.

Untuk menilai kemampuan generalisasi model, pengujian tambahan dilakukan dengan data dummy. Hasil menunjukkan bahwa model tetap menunjukkan akurasi yang memadai dalam sebagian besar kasus, meskipun terdapat beberapa perbedaan dalam prediksi. Beberapa prediksi sangat dekat dengan nilai aktual, hanya beberapa yang berbeda.

RandomizedSearchCV digunakan untuk mengoptimalkan parameter model saat memilih hyperparameter terbaik. Hasil menunjukkan bahwa konfigurasi terbaik mampu mencapai nilai R2 sebesar 0,9987, yang menunjukkan bahwa model dapat menjelaskan hampir semua variabilitas data. Ini menunjukkan bahwa model Decision Tree Regressor sangat cocok untuk digunakan dalam skenario prediksi harga sampah yang memerlukan ketepatan tinggi.

IV. KESIMPULAN

Penelitian ini berhasil mengimplementasikan algoritma *Decision Tree Regressor* untuk memprediksi harga sampah berdasarkan data historis. Model memiliki tingkat kesalahan yang rendah dan akurasi yang tinggi. Oleh karena itu, model ini dapat digunakan sebagai alat yang dapat diandalkan untuk memberikan estimasi harga yang akurat. Penambahan variabel tambahan yang dapat memengaruhi prediksi harga dan pengoptimalan metode pengaturan hyperparameter untuk hasil yang lebih akurat merupakan langkah maju yang dapat dilakukan.

REFERENSI

[1] B. Mahesh, “Machine Learning Algorithms - A Review,” *International Journal of Science and Research (IJSR)*, vol. 9, no. 1, pp. 381–386, 2020, doi: 10.21275/art20203995.

[2] D. Maulud and A. M. Abdulazeez, “A Review on Linear Regression Comprehensive in Machine Learning,” *Journal of Applied Science and Technology Trends*, vol. 1, no. 2, pp. 140–147, 2020, doi: 10.38094/jastt1457.

[3] H. Sharma and S. Kumar, “A Survey on Decision Tree Algorithms of Classification in Data Mining,” *International Journal of Science and Research (IJSR)*, vol. 5, no. 4, pp. 2094–2097, 2016, doi: 10.21275/v5i4.nov162954.

[4] D. Berrar, “Cross-validation 1,” pp. 1–13.