

LAMPIRAN

Lampiran 1 Hasil Turnitin



**UNIVERSITAS DARMA PERSADA
UPT PERPUSTAKAAN**

Gedung Rektorat Lantai 3,
Jl. Taman Malaka Selatan, Pondok Kelapa – Jakarta Timur 13450

**SURAT KETERANGAN
HASIL PENGECEKAN TURNITIN**

UPT Perpustakaan Universitas Darma Persada menerangkan telah selesai melakukan pemeriksaan duplikasi/*similarity* menggunakan perangkat lunak Turnitin terhadap hasil karya sebagai berikut:

Judul : KLASIFIKASI PENGELOLAAN SAMPAH MENGGUNAKAN
ALGORITMA DECISION TREE DAN K-MEANS CLUSTERING BERDASARKAN
NILAI EKONOMIS

Penulis : Anggy Iis Mawati

NIM : 2021230009

Tgl pemeriksaan : 5 Februari 2025

Dengan hasil Tingkat Kesamaan (*similarity index*) **13%**

Demikian Surat Keterangan kami buat, untuk dipergunakan sebagaimana mestinya.

Jakarta, 5 Februari 2025

Ka.UPT Perpustakaan Unsada

Yus Rusmiyati, SS., MM

Batas maksimal similarity 30% untuk Fakultas Sastra dan Ekonomi

Batas maksimal similarity 25% untuk Fakultas Teknik, Kelautan
dan Pasca Sarjana

Lampiran 2 Source Code Hasil Clustering dengan K-Means

```
1  st.subheader("Hasil Clustering")
2  st.write(df[['tahun','provinsi',
'kabupatenkota','kertas_karton', 'plastik', 'logam', 'kaca',
'hasil_cluster']])
3  silhouette_avg = silhouette_score(data_normalized,
data_clustering['cluster'])
4  # Tombol download untuk Hasil Clustering
5  st.download_button(
6      label="Unduh Hasil Clustering (Excel)",
7      data=to_excel(df[['provinsi', 'kabupatenkota',
'kertas_karton','plastik', 'logam', 'kaca', 'hasil_cluster']]),
8      file_name="hasil_clustering.xlsx",
9      mime="application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet"
10 )
11 # Menambahkan desain kustom untuk tombol
12 st.markdown("""
13     <style>
14         .stDownloadButton>button {
15             background-color: #dbd28c;
16             color: white;
17             font-size: 16px;
18             font-weight: bold;
19             border: none;
20             padding: 10px 20px;
21             border-radius: 5px;
22             cursor: pointer;
23             transition: background-color 0.3s ease;
24         }
25
26         .stDownloadButton>button:hover {
27             background-color: #4F4D793;
28         }
29     """)
```

```

30         .stDownloadButton>button:focus {
31             outline: none;
32         }
33     </style>
34     """ , unsafe_allow_html=True)
35     st.write(f"Rata-rata Silhouette Score untuk seluruh dataset:
36     {silhouette_avg}")
37     # Visualisasi dengan deskripsi cluster, centroid, dan label
38     cluster
37     st.subheader("Visualisasi Clustering dengan Deskripsi
38     Cluster")
39     plt.figure(figsize=(8, 6))
40     scatter = plt.scatter(data_clustering['plastik'],
41     data_clustering['logam'],
42     c=data_clustering['cluster'],
43     cmap='viridis')
44     centroids = kmeans.cluster_centers_
45     plt.scatter(centroids[:, 0], centroids[:, 1], s=200, c='red',
46     marker='X', label='Centroid')
47     # Menambahkan deskripsi warna sesuai cluster
48     cluster_labels = ['Rendah', 'Sedang', 'Tinggi']
49     colors = plt.cm.viridis(np.linspace(0, 1, 3))
50     for i, color in enumerate(colors):
51         plt.scatter([], [], c=[color], label=f'Cluster {i}:
52         {cluster_labels[i]}')
53     plt.xlabel('Plastik')
54     plt.ylabel('Logam')
55     plt.title('Clustering: Plastik vs Logam')
56     # plt.colorbar(scatter, label='Cluster')
57     plt.legend()
58     st.pyplot(plt)
59     # Menghitung jumlah data per cluster
60     cluster_counts = data_clustering['cluster'].value_counts()
61     # Menampilkan diagram batang untuk jumlah data per cluster
62     st.subheader("Distribusi Jumlah Data per Cluster")
63     plt.figure(figsize=(8, 6))

```

```

59 cluster_counts.plot(kind='bar', color=['#dbd28c', '#4F4D79',
    '#8cdbd2'])
60 plt.xlabel('Cluster')
61 plt.ylabel('Jumlah Data')
62 plt.title('Distribusi Jumlah Data per Cluster')
63 # Menambahkan label untuk setiap cluster dengan rotasi
    vertikal
64 cluster_labels = ['Cluster 0', 'Cluster 1', 'Cluster 2']
65 plt.xticks(ticks=range(len(cluster_counts)),
    labels=cluster_labels, rotation=0, ha='center')
66 # Menambahkan jumlah data di atas batang
67 for i, v in enumerate(cluster_counts):
68     plt.text(i, v + 0.5, str(v), ha='center', va='bottom',
    fontsize=12)
69 st.pyplot(plt)

```

Lampiran 3 Source Code Hasil Klasifikasi dengan Decision Tree

```

1 # Model Decision Tree
2 X = df[['kertas_karton', 'plastik', 'logam', 'kaca']]
3 y = df['hasil_cluster']
4 X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size=0.3, random_state=42)
5 clf = DecisionTreeClassifier(random_state=42)
6 clf.fit(X_train, y_train)
7 y_pred = clf.predict(X_test)
8 st.subheader("Evaluasi Model Klasifikasi (Decision Tree)")
9 st.write("Akurasi Model Decision Tree:",
    accuracy_score(y_test, y_pred))
10 df['tahun'] = df['tahun'].apply(lambda x: f"{int(x):d}") #
    Menambahkan format integer untuk tahun
11 # Menambahkan hasil prediksi ke DataFrame
12 df['Cluster'] = clf.predict(X)

13 # Fungsi untuk memberikan keterangan nilai ekonomis
    berdasarkan cluster
14 def nilai_ekonomis(cluster):

```

```

15     if cluster == 'Cluster 0':
16         return 'Rendah'
17     elif cluster == 'Cluster 1':
18         return 'Sedang'
19     elif cluster == 'Cluster 2':
20         return 'Tinggi'
21     else:
22         return 'Tidak Dikenal'
23 # Menambahkan kolom prediksi nilai ekonomis
24 df['prediksi_nilai_ekonomis'] =
df['Cluster'].apply(nilai_ekonomis)
25 # Fungsi untuk memberikan keterangan deskriptif terkait nilai
ekonomis
26 def keterangan_nilai_ekonomis(nilai_ekonomis):
27     if nilai_ekonomis == 'Tinggi':
28         return 'Menunjukkan potensi yang sangat menguntungkan
dalam pengelolaan sampah.'
29     elif nilai_ekonomis == 'Sedang':
30         return 'Potensi ekonomi yang memerlukan pemrosesan
lebih lanjut untuk hasil optimal.'
31     elif nilai_ekonomis == 'Rendah':
32         return 'Sedikit potensi untuk dimanfaatkan dalam
pengelolaan sampah.'
33     else:
34         return 'Tidak Dikenal'
35 # Menambahkan kolom keterangan nilai ekonomis
36 df['keterangan_prediksi'] =
df['prediksi_nilai_ekonomis'].apply(keterangan_nilai_ekonomis)
37 # Menampilkan tabel dengan provinsi dan kabupaten/kota
38 st.write("Hasil klasifikasi dengan kolom Provinsi dan
Kabupaten/Kota:")
39 hasil_klasifikasi = df[['tahun', 'provinsi', 'kabupatenkota',
'kertas_karton', 'plastik', 'logam', 'kaca', 'Cluster',
'prediksi_nilai_ekonomis', 'keterangan_prediksi']]
40 st.write(hasil_klasifikasi)

```

Lampiran 4 Source Code Hasil Evaluasi Kedua Model

```
1  # Model Decision Tree
2  X = df[['kertas_karton', 'plastik', 'logam', 'kaca']]
3  y = df['hasil_cluster']
4  X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)
5  clf = DecisionTreeClassifier(random_state=42)
6  clf.fit(X_train, y_train)
7  y_pred = clf.predict(X_test)
8  st.subheader("Evaluasi Model Klasifikasi (Decision Tree)")
9  st.write("Akurasi Model Decision Tree:",
accuracy_score(y_test, y_pred))
10 df['tahun'] = df['tahun'].apply(lambda x: f"{int(x):d}") #
Menambahkan format integer untuk tahun
11 # Menambahkan hasil prediksi ke DataFrame
12 df['Cluster'] = clf.predict(X)
13 # Fungsi untuk memberikan keterangan nilai ekonomis
berdasarkan cluster
14 def nilai_ekonomis(cluster):
15     if cluster == 'Cluster 0':
16         return 'Rendah'
17     elif cluster == 'Cluster 1':
18         return 'Sedang'
19     elif cluster == 'Cluster 2':
20         return 'Tinggi'
21     else:
22         return 'Tidak Dikenal'
23 # Menambahkan kolom prediksi nilai ekonomis
24 df['prediksi_nilai_ekonomis'] =
df['Cluster'].apply(nilai_ekonomis)
25 # Fungsi untuk memberikan keterangan deskriptif terkait nilai
ekonomis
26 def keterangan_nilai_ekonomis(nilai_ekonomis):
27     if nilai_ekonomis == 'Tinggi':
28         return 'Menunjukkan potensi yang sangat menguntungkan
dalam pengelolaan sampah.'
```

```

29     elif nilai_ekonomis == 'Sedang':
30         return 'Potensi ekonomi yang memerlukan pemrosesan
lebih lanjut untuk hasil optimal.'
31     elif nilai_ekonomis == 'Rendah':
32         return 'Sedikit potensi untuk dimanfaatkan dalam
pengelolaan sampah.'
33     else:
34         return 'Tidak Dikenal'
35 # Menambahkan kolom keterangan nilai ekonomis
36 df['keterangan_prediksi'] =
df['prediksi_nilai_ekonomis'].apply(keterangan_nilai_ekonomis)
37 # Menampilkan tabel dengan provinsi dan kabupaten/kota
38 st.write("Hasil klasifikasi dengan kolom Provinsi dan
Kabupaten/Kota:")
39 hasil_klasifikasi = df[['tahun', 'provinsi', 'kabupatenkota',
'kertas_karton', 'plastik', 'logam', 'kaca', 'Cluster',
'prediksi_nilai_ekonomis', 'keterangan_prediksi']]
40 st.write(hasil_klasifikasi)

```