

DAFTAR PUSTAKA

- Wijayanto, D. (2021). *Penerapan Algoritma Greedy pada Game Maze* (Doctoral dissertation, UIN Sunan Gunung Djati Bandung).
<https://digilib.uinsgd.ac.id/51094/>
- Febriansyah, A. R., & Arifin, T. (2023). Penerapan Algoritma K-Means Clustering Pada Agen Video Game Valorant. *eProsiding Teknik Informatika (PROTEKTIF)*, 4(1), 106-214.
<http://eprosiding.ars.ac.id/index.php/pti/article/view/982>
- Mayola, L., Hafizh, M., & Syahputra, H. (2025). Klasterisasi Rumah Sakit berdasarkan Kunjungan Pasien menggunakan Algoritma K-Means: Data 2019-2023. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 7(1), 15-21.
<https://jurnal.unidha.ac.id/index.php/jteksis/article/view/1703>
- Palevi, M. R., & Indra, Z. (2024). Implementasi Algoritma K-Means Clustering Dengan Pendekatan Active Learning Pada Siswa SMA Untuk Menentukan Jurusan Ke Perguruan Tinggi. *Jurnal SAINTIKOM (Jurnal Sains Manajemen Informatika dan Komputer)*, 23(1), 26-36.
<https://ojs.trigunadharma.ac.id/index.php/jis/article/view/9553>
- Sulaiman, D., & Utami, W. S. (2023). Rancang Bangun Sistem Rekomendasi Pemilihan Hero Pada Game Mobile Legends Menggunakan Algoritma Greedy. *INTECOMS: Journal of Information Technology and Computer Science*, 6(2), 998-1007.
<https://journal.ipm2kpe.or.id/index.php/INTECOM/article/view/8128>
- Wijayanti, R., Nugraha, W., & Kusri, K. (2021). Optimalisasi Penyelesaian Permainan pada Game Puzzle 8 dengan Perbandingan Algoritma A* dan Greedy. *Creative Information Technology Journal*, 7(1), 10-19.
<https://citec.amikom.ac.id/main/index.php/citec/article/view/230>
- Mempun, D. A. E., & Pieter, M. S. S. (2022). PENERAPAN ALGORITMA GREEDY DAN ALGORITMA BRANCH AND BOUND PADA GAME TIC TAC TOE. *Jurnal Teknologi Informasi*, 10(2), 10-22.
<https://ojs.ustj.ac.id/jti/article/view/1124>
- Waruwu, M. (2023). Pendekatan penelitian pendidikan: metode penelitian kualitatif, metode penelitian kuantitatif dan metode penelitian kombinasi (Mixed

Method). *Jurnal Pendidikan Tambusai*, 7(1), 2896-2910.

<https://jptam.org/index.php/jptam/article/view/6187>

Lakutu, N. F., Mahmud, S. L., Katili, M. R., & Yahya, N. I. (2023). Algoritma Dijkstra dan Algoritma Greedy Untuk Optimasi Rute Pengiriman Barang Pada Kantor Pos Gorontalo. *Euler: Jurnal Ilmiah Matematika, Sains dan Teknologi*, 11(1), 55-65.

<https://ejurnal.ung.ac.id/index.php/Euler/article/view/18244>

Putri, D. E., Putra, D. W. T., & Mandala, E. P. W. (2025). Implementasi Algoritma K-Means dalam Klasterisasi Penjualan Lauk Masakan Padang. *Progresif: Jurnal Ilmiah Komputer*, 21(1).

<https://ojs.stmik-banjarbaru.ac.id/index.php/progresif/article/view/2459>

Arviantino, F. N., Gata, W., Kurniawati, L., Setiawan, Y. A., & Priansyah, D. (2021). Penerapan Algoritma Greedy Dalam Pencarian Jalur Terpendek Pada Masjid–Masjid Di Kota Samarinda. *Metik Jurnal*, 5(1), 8-11.

<https://journal.universitasmulia.ac.id/index.php/metik/article/view/188>

Faisal, F. A., & Rahmatulloh, A. (2024). Analisis dan Rekomendasi Character Pada Game Genshin Impact Berdasarkan Revenue Banner Menggunakan Algoritma Clustering. *Jurnal Mahasiswa Teknik Informatika*, 3(1), 40-46.

<https://jurnal.unw.ac.id/index.php/jamastika/article/view/2741>

Jasiah, J., Kusumawati, I. R., & Febrina, W. (2023). Pelatihan Sistematika Penulisan Skripsi bagi Mahasiswa. *Masyarakat Berdaya Dan Inovasi*, 4(1), 58-64.

<http://mayadani.org/index.php/MAYADANI/article/view/112>

Andaria, A. C., & ST, M. P. (2024). Teknik Penulisan Latar Belakang Penelitian. *Metode Penelitian Kualitatif: Konsep dan Teori, Agama: Yayasan Tri Edukasi Ilmiah*, 15-33.

[https://www.researchgate.net/profile/Alex-Copernikus-](https://www.researchgate.net/profile/Alex-Copernikus-Andaria/publication/383787533_Teknik_Penulisan_Latar_Belakang_Penelitian/links/66d9ddd2fa5e11512c77a86a/Teknik-Penulisan-Latar-Belakang-Penelitian.pdf)

[Andaria/publication/383787533 Teknik Penulisan Latar Belakang Penelitian/links/66d9ddd2fa5e11512c77a86a/Teknik-Penulisan-Latar-Belakang-Penelitian.pdf](https://www.researchgate.net/profile/Alex-Copernikus-Andaria/publication/383787533_Teknik_Penulisan_Latar_Belakang_Penelitian/links/66d9ddd2fa5e11512c77a86a/Teknik-Penulisan-Latar-Belakang-Penelitian.pdf)

Mandira, I. M. C., & Suputra, I. D. G. W. D. (2023). Analisis Perilaku Konsumen Dalam Keputusan Pembelian Item Virtual Pada Game Online Valorant. *Jurnal Ilmu Ekonomi & Sosial*, 14(1), 74-82.

<https://www.ejournal.unmus.ac.id/index.php/ekosos/article/view/5090>

Syaefullah, W., & Anggapuspa, M. L. (2023). Analisis Visual pada Karakter Agen Sage dalam Game Valorant. *BARIK-Jurnal S1 Desain Komunikasi Visual*, 4(3), 130-140.

<https://ejournal.unesa.ac.id/index.php/JDKV/article/view/51005>

Bustomi, A. (2023). Sistem Rekomendasi Pemilihan Kinerja Koperasi Terbaik Menggunakan Simple Multi Attribute Rating Technique Method. *Journal of Information Technology, Software Engineering and Computer Science*, 1(4), 135-142.

<https://ejournal.techcart-press.com/index.php/itsecs/article/view/72>

Ardiansyah, R., Bianto, M. A., & Saputra, B. D. (2023). Sistem Rekomendasi Buku Perpustakaan Sekolah menggunakan Metode Content-Based Filtering. *Jurnal CoSciTech (Computer Science and Information Technology)*, 4(2), 510-518.

<https://ejurnal.umri.ac.id/index.php/coscitech/article/view/5131>

Sembiring, F. G., Nst, S. P. S., Gaol, A. S. L., & Harliana, P. (2025). IMPLEMENTASI ALGORITMA GREEDY DALAM PENENTUAN JALUR STRATEGIS DARI SIMPANG POS KE LAPANGAN MERDEKA KOTA MEDAN. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 9(1), 423-428.

<https://ejournal.itn.ac.id/index.php/jati/article/view/12343>

Cahyani, L., Sephiana, N., Tahir, M., & Aisyiah, J. (2024). Sistem Rekomendasi Wisata Kuliner Madura Menggunakan Content Based Filtering. *Explore IT: Jurnal Keilmuan dan Aplikasi Teknik Informatika*, 16(1), 31-38.

<https://jurnal.yudharta.ac.id/v2/index.php/EXPLORE-IT/article/view/5366>

Putranto, A., Azizah, N. L., & Astutik, I. R. I. (2023). Sistem Prediksi Penyakit Jantung Berbasis Web Menggunakan Metode SVM dan Framework Streamlit. *Kesatria: Jurnal Penerapan Sistem Informasi (Komputer dan Manajemen)*, 4(2), 442-452.

<http://www.pkm.tunasbangsa.ac.id/index.php/kesatria/article/view/180>

Widyaningrum, N. (2024). *SEGMENTASI NASABAH BANK MENGGUNAKAN ALGORITMA K-MEANS CLUSTERING DAN VISUALISASI DINAMIS DENGAN STREAMLIT* (Doctoral dissertation, UNIVERSITAS PGRI SEMARANG).

<https://eprints3.upgris.ac.id/id/eprint/4628/>

THOLIB, A. (2023). Buku Refrensi Implementasi Algoritma Machine Learning Berbasis Web dengan Framework Streamlit.

<https://repository.unuja.ac.id/id/eprint/1666/1/Buku%20Implementasi%20Machine%20Learning%20berbasis%20Web%20dengan%20Framework%20Streamlit%20.pdf>

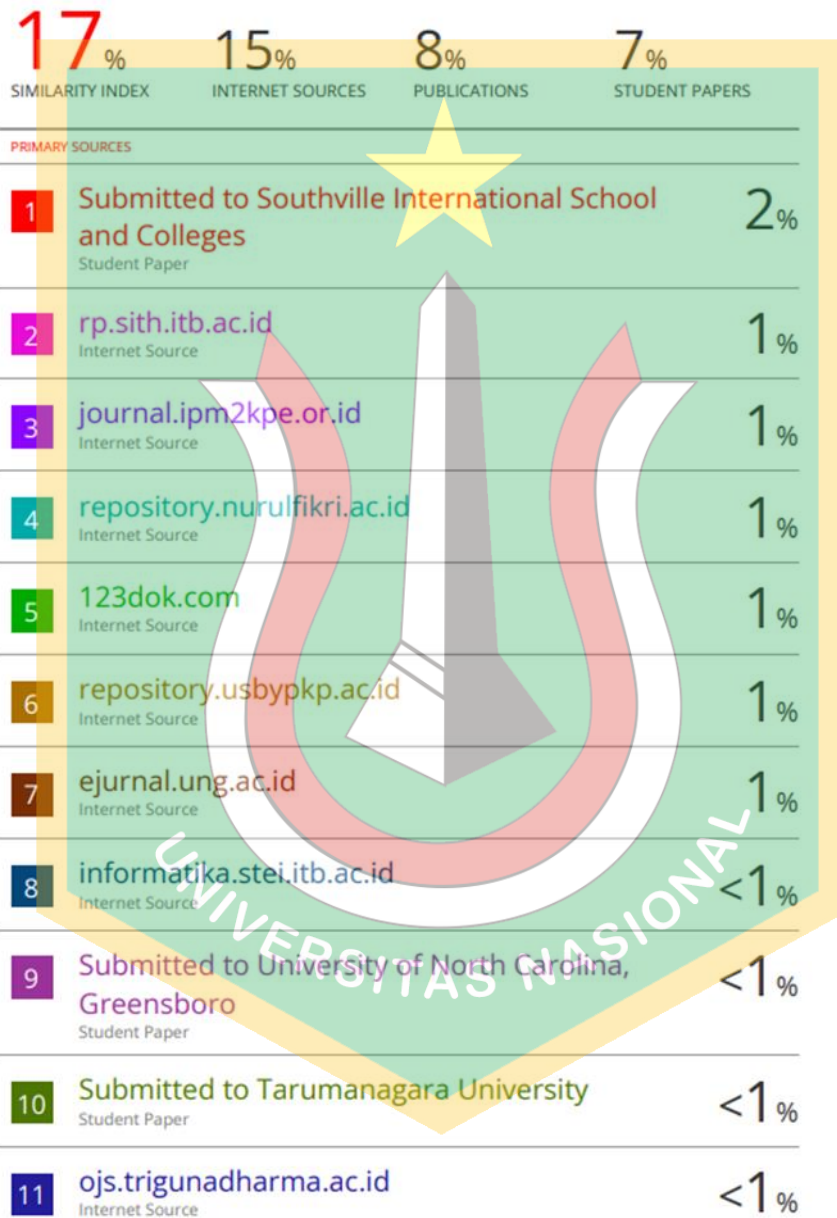
Hartati, T., Sohadi, R. T., Tohidi, E., & Wahyudin, E. (2024). Penerapan Algoritma Naive Bayes pada Analisis Sentimen Ulasan Aplikasi Whoosh–Kereta Cepat Di Google Play Store. *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 6(1), 244-249.
<https://publikasiilmiah.unwahas.ac.id/JINRPL/article/view/10307>



HASIL TURNITIN

Skripsi_Final_Muhammad_Sh Salman_Alfarisi-1740063820056

ORIGINALITY REPORT





UNIVERSITAS NASIONAL
FAKULTAS TEKNOLOGI KOMUNIKASI DAN
INFORMATIKA

S1 - Sistem Informasi, S1- Informatika, S2 - Pasca Sarjana Teknologi Infomasi

JL. Sawo Manila No. 61 Pejaten, Pasar Minggu, Jakarta 12520 Telp. (021) 78833307, 7806700 (Hunting) Fax. 7802718, 7802719

P.O. Box 4741 Jakarta 12047 Homepage : <http://www.unas.ac.id>. E-mail : febunas49@gmail.com

KONSULTASI BIMBINGAN

Npm : 217064516050
Nama : MUHAMMAD SHALMAN ALFARISI
Program Studi : Informatika
Konsentrasi : Teknologi Perangkat Lunak

KONSULTASI PEMBIMBING PROPOSAL

Tanggal	Materi Konsultasi	Status
30 November, 2024	melakukan bimbingan terkait judul yang ingin di pilih	Sudah Ditanggapi
30 November, 2024	melakukan bimbingan terkait perubahan judul yang ingin dipilih	Sudah Ditanggapi
30 November, 2024	melakukan bimbingan terkait penulisan bab 1 - bab 4	Sudah Ditanggapi
30 November, 2024	melakukan bimbingan untuk merevisi penulisan skripsi	Sudah Ditanggapi

KONSULTASI PEMBIMBING TUGAS AKHIR

Tanggal	Materi Konsultasi	Status
17 February, 2025	perubahan judul skripsi sesuai dengan arahan dosen pembimbing dan dosen penguji	Sudah Ditanggapi
17 February, 2025	perubahan flowchart sesuai dengan arahan dosen penguji	Sudah Ditanggapi
17 February, 2025	perubahan rumus algoritma menggunakan equation sesuai dengan arahan dosen penguji.	Sudah Ditanggapi
17 February, 2025	penambahan preprocessing data untuk menghapus tanda koma pada kolom matches	Sudah Ditanggapi
17 February, 2025	penambahan visualisasi algoritma greedy.	Sudah Ditanggapi

Tanggal	Materi Konsultasi	Status
17 February, 2025	penambahan evaluasi matriks algoritma k-means.	Sudah Ditanggapi
17 February, 2025	penambahan evaluasi matriks algoritma greedy.	Sudah Ditanggapi
17 February, 2025	bab v kesimpulan dan saran	Sudah Ditanggapi



LAMPIRAN

Turnitin Skripsi:

Skripsi_Final_Muhammad_Shalman_Alfarisi-1740063820056

ORIGINALITY REPORT

17%	15%	8%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS
PRIMARY SOURCES			
1	Submitted to Southville International School and Colleges		2%
	Student Paper		
2	rp.sith.itb.ac.id		1%
	Internet Source		
3	journal.ipm2kpe.or.id		1%
	Internet Source		
4	repository.nurulfikri.ac.id		1%
	Internet Source		
5	123dok.com		1%
	Internet Source		
6	repository.usbypkp.ac.id		1%
	Internet Source		
7	ejurnal.ung.ac.id		1%
	Internet Source		
8	informatika.stel.itb.ac.id		<1%
	Internet Source		
9	Submitted to University of North Carolina, Greensboro		<1%
	Student Paper		
10	Submitted to Tarumanagara University		<1%
	Student Paper		
11	ojs.trigunadharma.ac.id		<1%
	Internet Source		

Source Code:

```
import numpy as np
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, davies_bouldin_score
import seaborn as sns
import streamlit as st
import matplotlib.pyplot as plt

# Membaca dataset
df = pd.read_csv('Dataset.csv')

# Membersihkan kolom yang mengandung angka dengan tanda koma (',')
df['Matches'] = df['Matches'].replace({' ': ''},
regex=True).astype(float)
df['Win %'] = df['Win %'].astype(float)
df['Pick %'] = df['Pick %'].astype(float)
df['Avg. Score'] = df['Avg. Score'].astype(float)

df_clean = df.loc[:, ~df.columns.str.contains('^Unnamed')]

# 1. Menampilkan jumlah data sebelum duplikat
st.subheader("Jumlah Data Sebelum Menghapus Duplikat:")
st.write(f"Jumlah Data: {df_clean.shape[0]}")

# 2. Menghapus Data Duplikat
df_no_duplicates = df_clean.drop_duplicates()

# Menampilkan hasil setelah menghapus duplikat
st.subheader("Jumlah Data Setelah Menghapus Duplikat:")
st.write(f"Jumlah Data: {df_no_duplicates.shape[0]}")

# Visualisasi Data Sebelum dan Setelah Menghapus Duplikat
fig, ax = plt.subplots(1, 2, figsize=(12, 6))

# Visualisasi Jumlah Baris sebelum duplikat
ax[0].bar(['Data'], [df_clean.shape[0]], color='blue')
ax[0].set_title("Jumlah Baris Sebelum Menghapus Duplikat")
ax[0].set_ylabel("Jumlah Baris")

# Visualisasi Jumlah Baris setelah duplikat
ax[1].bar(['Data'], [df_no_duplicates.shape[0]], color='green')
ax[1].set_title("Jumlah Data Setelah Menghapus Duplikat")
```

```

ax[1].set_ylabel("Jumlah Data")

# Menampilkan plot
plt.tight_layout()
st.pyplot(fig)

# Menghapus baris dengan missing values
df_no_missing = df_no_duplicates.dropna()

# Menampilkan jumlah missing values setelah penanganan
st.subheader("Jumlah Missing Values")
st.write(df_no_missing.isnull().sum()) # Menampilkan jumlah missing values setelah penanganan

# Pilihan map dan tipe agen
map_options = ['Pilih Map', 'Abyss', 'Ascent', 'Bind', 'Pearl', 'Split', 'Sunset']
agent_types = ['Pilih Agent', 'Controller', 'Initiator', 'Sentinel', 'Duelist']

st.title(f'Sistem Rekomendasi Agen Valorant')

# Terapkan KMeans terlebih dahulu pada seluruh data (tanpa filter)
X = df[['Win %', 'Pick %', 'Avg. Score', 'Matches']]
kmeans = KMeans(n_clusters=3, random_state=42)
df['Cluster'] = kmeans.fit_predict(X)

# Pilih map dan tipe agen
selected_map = st.selectbox('Pilih Map:', map_options)
selected_agent_type = st.selectbox('Pilih Tipe Agen:', agent_types)

# Filter data berdasarkan pilihan map dan tipe agen setelah penerapan KMeans
filtered_df = df[(df['Map'] == selected_map) & (df['Role'] == selected_agent_type)]

# Menghapus kolom yang tidak relevan (kolom "Unnamed")
filtered_df = filtered_df.loc[:, ~filtered_df.columns.str.contains('^Unnamed')]

# Pastikan ada cukup data untuk KMeans (min 3 sampel)
if len(filtered_df) >= 3:
    cluster_summary = filtered_df.groupby('Cluster')['Matches'].mean()
    best_cluster = cluster_summary.idxmax()

```

```

# Filter agen terbaik di cluster terbaik
best_cluster_df = filtered_df[filtered_df['Cluster'] ==
best_cluster]

best_agent_greedy = best_cluster_df.sort_values(
    by=['Matches', 'Win %'],
    ascending=[False, False]
).iloc[0] # Pilih agen terbaik berdasarkan urutan kriteria

# Tampilkan hasil dalam tabel
st.title(f'Rekomendasi Agen {selected_agent_type} Terbaik untuk
Map {selected_map}')
if selected_map == "Pearl":
    st.image("https://cdn.oneesports.id/cdn-
data/sites/2/2022/06/Valorant_Pearl_Map_Overview-1024x576-1.png",
width=420)
    st.write("Tim Penyerang push ke area Tim Pertahanan di peta
dua lokasi, berlatar di kota bawah laut yang meriah. Pearl adalah
peta berbasis geografi tanpa mekanika. Lancarkan aksi melalui jalur
mid yang padat atau sisi sayap yang lebih jauh di peta pertama yang
berlatar di Omega Earth.")
elif selected_map == "Bind":
    st.image("https://cmsassets.rgpub.io/sanity/images/dsfx7636/
news/e9b923ef40c945736bb200d54faef5375cbc476f-
914x514.jpg?auto=format&fit=fill&q=80&w=914", width=420) # Ganti
dengan URL gambar untuk Bind
    st.write("Bind sangat populer di Valorant karena
teleporternya yang khas, sebuah fitur yang tidak ditemukan di peta
lain mana pun. Teleporter ini menawarkan pemain kemampuan untuk
menavigasi medan, menuntut kewaspadaan terus-menerus dan kemampuan
beradaptasi dengan cepat. Ini adalah peta yang membuat pemain tetap
waspada, selalu siap menghadapi perubahan tak terduga di medan
perang.")
elif selected_map == "Ascent":
    st.image("https://beebom.com/wp-
content/uploads/2023/09/Ascent-Valorant-Cover.jpg", width=420) #
Ganti dengan URL gambar untuk Ascent
    st.write("Ascent dianggap sebagai peta favorit penggemar,
menarik pemain dengan desain Italia yang terinspirasi dari Venesia.
Tata letak dan garis pandangnya yang unik membuatnya disukai oleh
para gamer yang menyukai pertempuran jarak jauh dan pertempuran
jarak dekat.")
elif selected_map == "Abyss":

```

```

st.image("https://cdn.oneesports.gg/cdn-
data/2024/06/Valorant_AbyssMap_OfficialImage_4-1536x864.jpg",
width=420) # Ganti dengan URL gambar untuk Ascent

st.write("Abyss adalah peta ke-11 Valorant dan menampilkan
desain unik tanpa batas yang menantang pemain dengan batas terbuka.
Satu kesalahan langkah dapat membuat Anda terjatuh, sehingga
memerlukan gerakan hati-hati dan kesadaran.")

elif selected_map == "Split":
    st.image("https://cmsassets.rgpub.io/sanity/images/dsfx7636/
news/a90e310c8023ac6ed718ab9616939fc137ca0507-
914x514.jpg?auto=format&fit=fill&q=80&w=914", width=420) # Ganti
dengan URL gambar untuk Ascent
    st.write("Jika Anda ingin pergi jauh, Anda harus naik.
Sepasang situs yang dipisahkan oleh pusat yang ditinggikan
memungkinkan pergerakan cepat menggunakan dua tali ascender. Setiap
situs dibangun dengan menara menjulang yang penting untuk kendali.
Ingatlah untuk mengawasi di atas sebelum semuanya meledak setinggi
langit.")

elif selected_map == "Sunset":
    st.image("https://cmsassets.rgpub.io/sanity/images/dsfx7636/
news/0749044db042a158fc2e24849ca5adfedadc71-
915x515.png?auto=format&fit=fill&q=80&w=915", width=420) # Ganti
dengan URL gambar untuk Ascent
    st.write("Sunset menawarkan tata letak klasik dengan dua
situs lonjakan dan tiga jalur berbeda, mengadopsi desain peta
tradisional. Peta bertema Los Angeles juga menyertakan pintu mekanis
yang mengingatkan kita pada Ascent, memperkenalkan elemen dinamis
pada gameplay.")

# Menampilkan agen terbaik setelah algoritma greedy
st.title(f"Agen Terbaik: {best_agent_greedy['Agent']}")
result_df = pd.DataFrame({
    'Tipe Agen': [best_agent_greedy['Role']],
    'Win %': [best_agent_greedy['Win %']],
    'Pick %': [best_agent_greedy['Pick %']],
    'Avg. Score': [best_agent_greedy['Avg. Score']],
    'Matches': [best_agent_greedy['Matches']]
})

result_df = result_df.applymap(lambda x: f"{x:.1f}" if
isinstance(x, float) else x)
st.table(result_df) # Menampilkan tabel hasil rekomendasi agen
terbaik

```

```

st.title(f"Hasil Clustering K-Means")
df = df.loc[:, ~df.columns.str.contains('^Unnamed')]
df_cluster = df.applymap(lambda x: f"{x:.1f}" if isinstance(x,
float) else x)
st.dataframe(df_cluster)

st.title(f"Hasil Filtering Berdasarkan Map dan Role")
filtered_df_table = filtered_df.applymap(lambda x: f"{x:.1f}" if
isinstance(x, float) else x)
st.dataframe(filtered_df_table)

# Visualisasi Cluster menggunakan scatter plot
plt.figure(figsize=(10, 6))
scatter = plt.scatter(
    filtered_df['Win %'], filtered_df['Matches'],
    c=filtered_df['Cluster'], cmap='viridis', s=100,
    edgecolors='k', alpha=0.7
)

# Menambahkan judul, label sumbu X dan Y
plt.title('Clustering of Agents', fontsize=16)
plt.xlabel('Win %', fontsize=12)
plt.ylabel('Matches', fontsize=12)
plt.colorbar(scatter, label='Cluster')

# Menambahkan nama agen sebagai anotasi pada scatter plot
for i in range(len(filtered_df)):
    plt.text(
        filtered_df['Win %'].iloc[i],
        filtered_df['Matches'].iloc[i],
        filtered_df['Agent'].iloc[i], fontsize=9, ha='right',
        color='black'
    )

st.pyplot(plt)

# Mencoba berbagai jumlah cluster untuk evaluasi
k_values = range(2, 8)
evaluation_results = []

# Ambil seluruh data tanpa filter
X = df[['Win %', 'Pick %', 'Avg. Score', 'Matches']] # Pastikan
Anda menggunakan seluruh data

for k in k_values:

```

```

kmeans = KMeans(n_clusters=k, random_state=42)
labels = kmeans.fit_predict(X)
silhouette_avg = silhouette_score(X, labels)
davies_bouldin_avg = davies_bouldin_score(X, labels)

evaluation_results.append([k, davies_bouldin_avg,
silhouette_avg])

# Konversi hasil evaluasi ke dalam DataFrame
eval_df = pd.DataFrame(evaluation_results, columns=['Cluster',
'Davies-Bouldin Index', 'Silhouette Score'])

# Menampilkan hasil evaluasi dalam tabel
st.write("### Evaluasi KMeans untuk Berbagai Cluster")
st.table(eval_df.style.format({'Davies-Bouldin Index': '{:.3f}',
'Silhouette Score': '{:.3f}'}))

# **Visualisasi Algoritma Greedy**: Bar Chart untuk
memperlihatkan kriteria sebelum dan setelah pengurutan
st.write("### Visualisasi Algoritma Greedy")

# **Sebelum Pengurutan (Greedy)**: Agen sebelum diurutkan
st.write("#### Sebelum penerapan (Greedy)")
fig, ax = plt.subplots(figsize=(10, 6))

# Visualisasi kriteria agen sebelum pengurutan
ax.bar(best_cluster_df['Agent'], best_cluster_df['Matches'],
color='orange', alpha=0.6, label='Matches')

ax.set_xlabel('Agen')
ax.set_ylabel('Nilai')
ax.legend()

st.pyplot(fig)

st.write("#### Setelah penerapan (Greedy)")
fig, ax = plt.subplots(figsize=(10, 6))

ax.bar(best_agent_greedy['Agent'], best_agent_greedy['Matches'],
color='orange', alpha=0.6, label='Matches')
ax.set_xlabel('Agen')
ax.set_ylabel('Nilai')
ax.legend()

st.pyplot(fig)

```

```

# Prediksi greedy: memilih cluster dengan Win % tertinggi
greedy_predictions = filtered_df['Cluster'] == best_cluster

# 1. Evaluasi berdasarkan jumlah Matches
best_agent_matches = filtered_df.sort_values(by=['Matches', 'Win
%'], ascending=False).iloc[0] # Agen dengan Matches terbanyak
st.write(f"Agen dengan Matches Terbanyak:
{best_agent_matches['Agent']}")

# 2. Top-k Accuracy: Apakah agen dengan Matches terbanyak ada di
antara Top-k agen terbanyak
top_k = 5
top_agents_matches = filtered_df.sort_values(by=['Matches', 'Win
%'], ascending=False).head(top_k)
greedy_selected = best_agent_matches['Agent']
is_greedy_top_k = greedy_selected in
top_agents_matches['Agent'].values
st.write(f"Apakah agen yang dipilih oleh Greedy ada di antara
{top_k} agen dengan Matches dan Win% terbanyak?: {'Yes' if
is_greedy_top_k else 'No'}")

# 3. Visualisasi perbandingan agen berdasarkan Matches
fig1, ax1 = plt.subplots(figsize=(10, 6))
ax1.bar(filtered_df['Agent'], filtered_df['Matches'],
color='green', alpha=0.6, label='Matches')
ax1.set_xlabel('Agen')
ax1.set_ylabel('Matches')
ax1.legend()
st.pyplot(fig1)

else:
st.write("Tidak cukup data untuk melakukan klasterisasi.")

```