

LAPORAN UJIAN AKHIR SEMESTER

Clustering Pemain Sepak Bola FIFA24 Berdasarkan Kemampuan dan Value

Disusun untuk Memenuhi Tugas Kuliah

Mata Kuliah: Kecerdasan Artifisial

Dosen Pengampu:

Riskyana Dewi Intan Puspitasari, M.Kom



Disusun Oleh:

Muhammad Fabyan Putroagung	23031554029
Muhammad Taufiqulhakim	23031554111
Aufatir Diaul Haq	23031554127
Akhmad Alviantio	23031554238

Kelas: Sains Data 2023 D
PROGRAM STUDI SAINS DATA UNIVERSITAS NEGERI SURABAYA TAHUN 2024

1. Ide Project

- a. Judul Project : Clustering Pemain Sepak Bola FIFA24 Berdasarkan Kemampuan dan Value
- b. Latar Belakang, Manfaat, dan Tujuan :

Latar Belakang: Dalam era sepak bola modern, analisis data telah menjadi bagian integral dalam pengambilan keputusan. Data statistik pemain tidak hanya membantu pelatih dalam memilih strategi terbaik tetapi juga memberikan wawasan kepada manajer tim, analis olahraga, dan pengembang gim. Dataset FIFA24 Players Stats adalah salah satu sumber data yang kaya akan informasi, mencakup atribut fisik, teknis, dan nilai ekonomi pemain dari berbagai negara.

Dengan meningkatnya kompleksitas dalam dunia sepak bola, pengelompokan pemain berdasarkan kemampuan mereka dapat memberikan wawasan yang lebih dalam untuk memahami kekuatan dan kelemahan setiap pemain. Selain itu, analisis berbasis data dapat digunakan untuk mengidentifikasi pemain muda berbakat, merancang strategi transfer, atau mengoptimalkan komposisi tim. Oleh karena itu, proyek ini bertujuan untuk mengaplikasikan metode clustering dalam mengelompokkan pemain berdasarkan kemampuan dan nilai mereka.

Proses clustering ini tidak hanya relevan untuk tim sepak bola profesional tetapi juga dapat diterapkan dalam pengembangan fitur gim sepak bola, seperti mode karier atau simulasi manajemen tim. Dengan cara ini, data pemain dapat dimanfaatkan secara maksimal untuk menciptakan solusi yang lebih strategis dan berbasis bukti.

Manfaat: Proyek ini memiliki berbagai manfaat yang signifikan. Pertama, hasil clustering dapat membantu manajer tim dalam mengidentifikasi pemain yang sesuai dengan kebutuhan taktis mereka. Misalnya, pemain dalam cluster tertentu mungkin lebih cocok untuk peran playmaker atau striker. Kedua, analisis ini memberikan wawasan tentang potensi pemain muda dengan nilai rendah yang dapat dikembangkan menjadi pemain bintang. Selain itu, data ini dapat digunakan oleh pengembang gim FIFA untuk menciptakan pengalaman bermain yang lebih realistis dan berbasis data.

Manfaat lain adalah membantu analis olahraga dalam membuat laporan yang lebih informatif tentang karakteristik pemain dan pola permainan. Dengan menggunakan algoritma clustering, informasi yang dihasilkan menjadi lebih terorganisir dan mudah dipahami oleh berbagai pihak yang terlibat dalam industri sepak bola.

Tujuan: Proyek ini bertujuan untuk mengembangkan sistem clustering yang efektif dalam mengelompokkan pemain sepak bola FIFA24 berdasarkan kemampuan teknis, fisik, dan nilai mereka. Secara khusus, proyek ini memiliki tujuan berikut:

1. Mengidentifikasi tipe pemain sepak bola berdasarkan kombinasi atribut seperti kemampuan teknis dan fisik.
2. Menyediakan alat bantu analisis yang dapat digunakan oleh manajer tim, pengembang gim, dan analis olahraga untuk membuat keputusan yang lebih informatif.
3. Mengeksplorasi berbagai kombinasi atribut untuk menentukan fitur yang paling

relevan dalam clustering pemain.

4. Memberikan rekomendasi pemain terbaik berdasarkan kebutuhan tim dan identifikasi pemain muda berbakat.

Dengan mencapai tujuan-tujuan ini, proyek ini diharapkan dapat memberikan kontribusi yang signifikan dalam pengambilan keputusan berbasis data dalam industri sepak bola.

- c. Referensi/publikasi acuan yang digunakan (prosiding/jurnal internasional) :

Jain, A. K. (2010). Data Clustering: 50 Years Beyond K-Means. Pattern Recognition Letters.

FIFA24 Dataset - Kaggle Link

Kohonen, T. (1982). "Self-Organized Formation of Topologically Correct Feature Maps."

Biological Cybernetics. Blog: "Clustering FIFA Players for Fun and Insight" - Kaggle Notebook. Scikit-learn Documentation: K-Means Clustering.

2. Dataset

- a. Dataset yang digunakan: Dataset FIFA24 Players Stats yang mencakup atribut fisik, teknis, dan ekonomis dari ribuan pemain sepak bola. Dataset ini berisi kolom seperti:

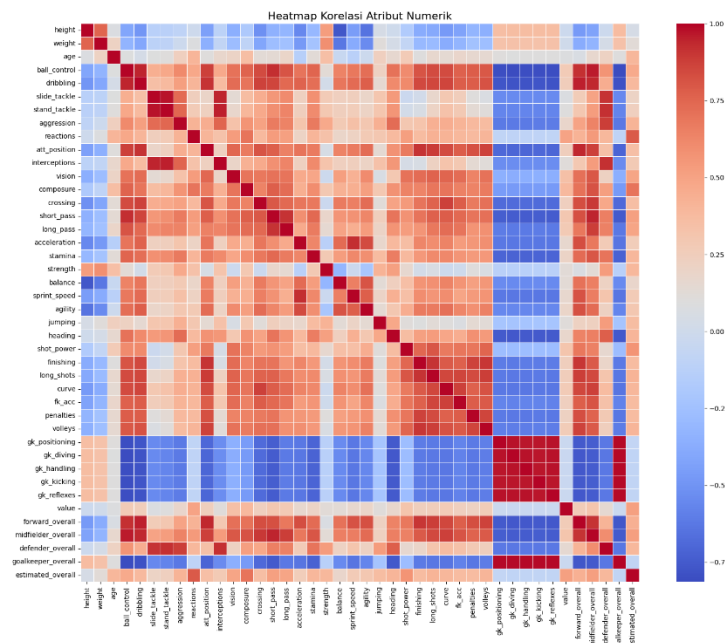
player, country, height, weight, age, club, ball_control, dribbling, slide_tackle, acceleration, stamina, value, dll.

Link dataset: <https://www.kaggle.com/datasets/rehandl23/fifa-24-player-stats-dataset>

- b. Exploratory Data Analysis:

1. Heatmap Korelasi:

- Analisis korelasi antar atribut menggunakan heatmap untuk mengevaluasi hubungan antar variabel seperti ball_control, dribbling, value, dan atribut teknis lainnya.
- Korelasi ini membantu menentukan atribut mana yang paling relevan untuk clustering.



Berdasarkan heatmap korelasi, berikut adalah analisisnya:

- Korelasi Positif Kuat:

Kotak merah tua menunjukkan korelasi positif yang kuat antara atribut, artinya nilai satu atribut meningkat seiring peningkatan nilai atribut lain. Contoh yang mungkin terlihat:

- ball_control dan dribbling memiliki korelasi tinggi karena keterampilan ini sering saling mendukung dalam performa pemain.
- short_pass dan long_pass memiliki korelasi kuat, yang masuk akal karena kemampuan passing sering saling berkaitan.
- gk_positioning, gk_diving, gk_reflexes menunjukkan korelasi kuat di antara atribut kiper.

- Korelasi Negatif Kuat:

Kotak biru tua menunjukkan korelasi negatif yang kuat. Ini menunjukkan bahwa ketika nilai satu atribut meningkat, nilai atribut lain cenderung menurun. Misalnya:

Atribut fisik seperti height dan kecepatan (misalnya sprint_speed, agility) mungkin menunjukkan korelasi negatif. Pemain yang lebih tinggi cenderung kurang gesit atau lebih lambat.

- Hubungan Atribut Spesifik:

- Atribut menyerang seperti finishing, att_position, shot_power cenderung memiliki korelasi tinggi satu sama lain, karena atribut ini mendukung performa pemain dalam posisi menyerang.
- Atribut bertahan seperti stand_tackle, slide_tackle, interceptions juga menunjukkan hubungan yang kuat satu sama lain.

- Korelasi Atribut Kiper:

Atribut kiper seperti gk_positioning, gk_diving, gk_handling, gk_kicking, gk_reflexes memiliki korelasi yang sangat tinggi, yang menunjukkan bahwa pemain yang unggul di satu kemampuan kiper cenderung unggul di kemampuan lainnya.

- Korelasi dengan value:

Korelasi antara value (nilai pemain) dan atribut lain menunjukkan atribut yang paling memengaruhi nilai pasar pemain. Biasanya, atribut seperti finishing, short_pass, dribbling, ball_control, dan kemampuan fisik seperti sprint_speed akan menunjukkan korelasi tinggi dengan nilai pemain.

- Zona Putih (Korelasi Rendah):

Area dengan warna mendekati putih menunjukkan korelasi lemah antara atribut. Misalnya, mungkin terdapat korelasi rendah antara atribut menyerang seperti finishing dan atribut bertahan seperti slide_tackle, yang menunjukkan peran spesifik pemain.

- Kesimpulan:

Clustering Pemain:

- Pemain dengan atribut yang saling berkorelasi kuat dapat dikelompokkan bersama, misalnya:
- Striker: finishing, shot_power, att_position.
- Playmaker: short_pass, vision, composure.
- Defender: interceptions, stand_tackle, slide_tackle.
- Goalkeeper: gk_positioning, gk_reflexes, gk_diving.
- Atribut dengan korelasi kuat terhadap value dapat menjadi indikator untuk memilih pemain berkualitas tinggi dan bernilai pasar tinggi.

2. Normalisasi Data:

- Menskalakan atribut numerik menggunakan Standard Scaler untuk memastikan atribut dengan rentang yang berbeda tidak mendominasi hasil clustering.

3. Statistik Deskriptif:

- Setelah clustering data, dilakukan analisis statistik deskriptif untuk melihat distribusi atribut seperti mean, median, minimum, dan maksimum pada setiap cluster. Hal ini memberikan gambaran karakteristik utama dari masing-masing cluster.

3. Metode

a. Penentuan Metode:

K-Means adalah salah satu algoritma clustering yang paling populer. Algoritma ini mengelompokkan data menjadi k cluster, di mana k adalah jumlah cluster yang telah ditentukan sebelumnya. Prinsip dasar dari K-Means adalah meminimalkan jarak antara setiap data dengan centroid cluster yang relevan. Berikut adalah langkah-langkah dasar dalam algoritma K-Means:

- i. Inisialisasi Centroid: Tentukan jumlah cluster k dan pilih titik acak sebagai centroid awal.
- ii. Penugasan Data ke Cluster: Setiap titik data akan dialokasikan ke cluster yang memiliki centroid terdekat berdasarkan jarak Euclidean.
- iii. Pembaruan Centroid: Setelah semua data dialokasikan, centroid cluster dihitung ulang berdasarkan rata-rata posisi data dalam cluster tersebut.
- iv. Iterasi: Langkah penugasan data dan pembaruan centroid diulang sampai tidak ada perubahan signifikan pada posisi centroid atau jumlah iterasi mencapai batas yang ditentukan.

Salah satu tantangan dalam K-Means adalah pemilihan jumlah cluster k yang optimal. Untuk itu, digunakan metode seperti Elbow Method yang membantu dalam menentukan nilai k yang memberikan pemisahan terbaik antar cluster.

Metode Elbow digunakan untuk menentukan jumlah cluster yang optimal dalam algoritma K-Means. Metode ini memplot jumlah cluster k terhadap inertia (jumlah jarak total antara titik data dan centroidnya). Inertia biasanya menurun seiring bertambahnya jumlah cluster, tetapi ada titik di mana penurunan inertia mulai melambat. Titik ini disebut sebagai "elbow", dan biasanya dianggap sebagai jumlah cluster optimal. Dengan menggunakan Elbow Method, kita dapat menghindari pemilihan jumlah cluster yang terlalu banyak atau terlalu sedikit, yang bisa mengurangi kualitas clustering.

b. Kesesuaian metode dengan penyelesaian masalah dalam project:

- K-Means cocok untuk mengelompokkan pemain berdasarkan kombinasi atribut numerik seperti kemampuan teknis dan value. Algoritma ini efektif untuk dataset FIFA24 karena memiliki kompleksitas waktu yang terukur dan bekerja baik pada data dengan jumlah fitur yang relatif besar.

c. Penentuan Metrik Evaluasi

Silhouette Score adalah metrik evaluasi yang digunakan untuk mengukur kualitas hasil clustering. Skor ini mengukur seberapa mirip data dalam satu cluster dibandingkan dengan data di cluster lain. Nilai Silhouette Score berkisar antara -1 dan 1, di mana nilai yang lebih tinggi menunjukkan bahwa data lebih baik terkelompok dalam cluster yang sesuai. Jika

skor Silhouette negatif atau rendah, itu menunjukkan bahwa data mungkin telah dikelompokkan dengan buruk atau ada banyak tumpang tindih antar cluster.

Silhouette Score dihitung dengan mempertimbangkan dua nilai penting untuk setiap titik data:

- $a(i)$: Rata-rata jarak antara titik i dan semua titik dalam cluster yang sama.
- $b(i)$: Rata-rata jarak antara titik i dan titik terdekat di cluster lain.

Skor Silhouette untuk titik data i dihitung sebagai:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Semakin tinggi nilai $s(i)$, semakin baik data tersebut dikelompokkan dalam cluster yang sesuai.

Lalu ada Davies-Bouldin Index, yaitu metrik evaluasi untuk mengukur kualitas clustering berdasarkan rasio antara jarak dalam cluster (intra-cluster) dan jarak antar cluster (inter-cluster). Semakin rendah nilai Davies-Bouldin Index, semakin baik hasil clustering. Indeks ini dihitung sebagai rata-rata dari nilai maksimum rasio jarak untuk setiap cluster, di mana nilai tersebut menggambarkan seberapa mirip cluster tertentu dengan cluster lainnya. Davies-Bouldin Index memberikan informasi penting tentang seberapa terpisah dan homogen cluster yang dihasilkan.

4. Hasil dan Analisis

a. Hasil Eksperimen/Permodelan

Dilakukan serangkaian eksperimen untuk membandingkan hasil clustering berdasarkan kombinasi fitur yang digunakan. Semua fitur yang diuji dibandingkan dengan dan tanpa Standard Scaler. Berikut adalah beberapa konfigurasi fitur yang diuji:

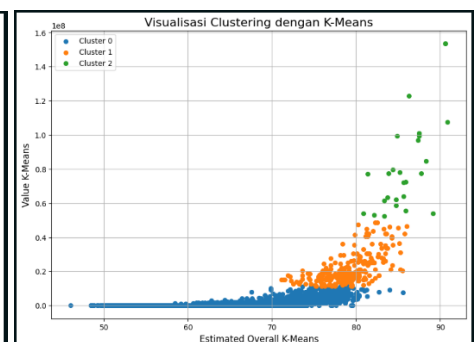
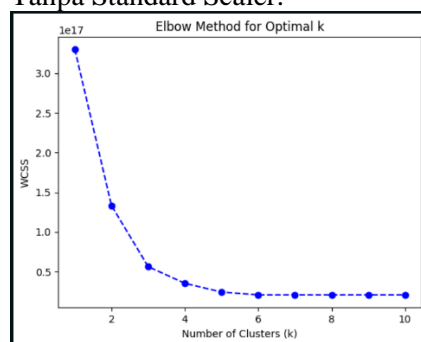
1. Hanya `estimated_overall` dan `value`

Kombinasi ini digunakan untuk mengelompokkan pemain berdasarkan kemampuan keseluruhan (`estimated_overall`) dan nilai pasar mereka (`value`).

Tujuan: Fokus pada pengelompokan sederhana untuk analisis performa dasar pemain.

Hasil:

- Tanpa Standard Scaler:



	estimated_overall				value			
	mean	median	min	max	mean	median	min	max
Cluster								
0	65.943737	66.250	46.000	85.571429	9.728482e+05	725.0	4.0	11000000.0
1	79.027259	79.000	71.125	86.000000	2.146735e+07	18500000.0	11500000.0	48500000.0
2	85.892989	85.625	80.875	90.875000	8.268519e+07	77500000.0	52500000.0	153500000.0

```
new_player = [[90, 11220097]]

def predict_cluster(new_data, loaded_centroids):
    predictions = []
    for point in new_data:
        distances = [euclidean_distance(point, centroid) for centroid in loaded_centroids]
        predictions.append(distances.index(min(distances)))
    return predictions
predicted_cluster = predict_cluster(new_player, loaded_centroids)
print(f"Pemain baru dengan data {new_player} masuk ke cluster: {predicted_cluster[0]}")
```

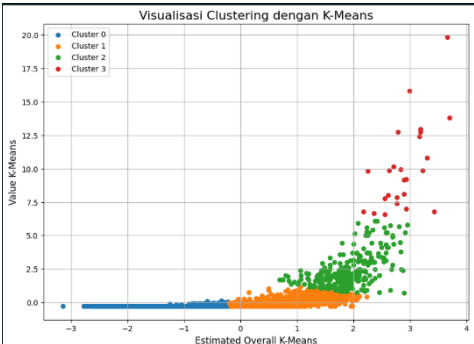
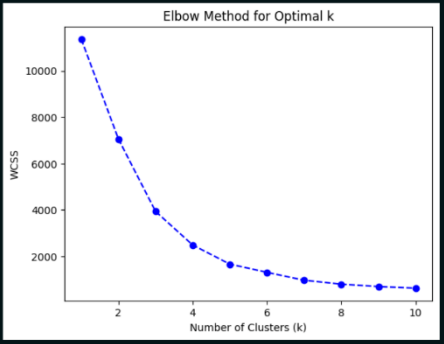
✓ 0.0s
Pemain baru dengan data [[90, 11220097]] masuk ke cluster: 0

```
new_player = [[40, 11220098]]

def predict_cluster(new_data, loaded_centroids):
    predictions = (parameter) new_data: Any
    for point in new_data:
        distances = [euclidean_distance(point, centroid) for centroid in loaded_centroids]
        predictions.append(distances.index(min(distances)))
    return predictions
predicted_cluster = predict_cluster(new_player, loaded_centroids)
print(f"Pemain baru dengan data {new_player} masuk ke cluster: {predicted_cluster[0]}")
```

✓ 0.0s
Pemain baru dengan data [[40, 11220098]] masuk ke cluster: 1

• Dengan Standard Scaler:



	estimated_overall				value			
	mean	median	min	max	mean	median	min	max
Cluster								
0	60.590394	61.500000	46.000000	65.428571	6.214544e+04	325.0	4.0	3300000.0
1	70.163634	69.625000	65.285714	81.285714	1.606699e+06	1200000.0	25.0	11000000.0
2	78.959612	78.828571	71.125000	86.000000	1.993972e+07	17500000.0	7500000.0	48500000.0
3	85.892989	85.625000	80.875000	90.875000	8.268519e+07	77500000.0	52500000.0	153500000.0

```
new_player = [[98, 49000000]]

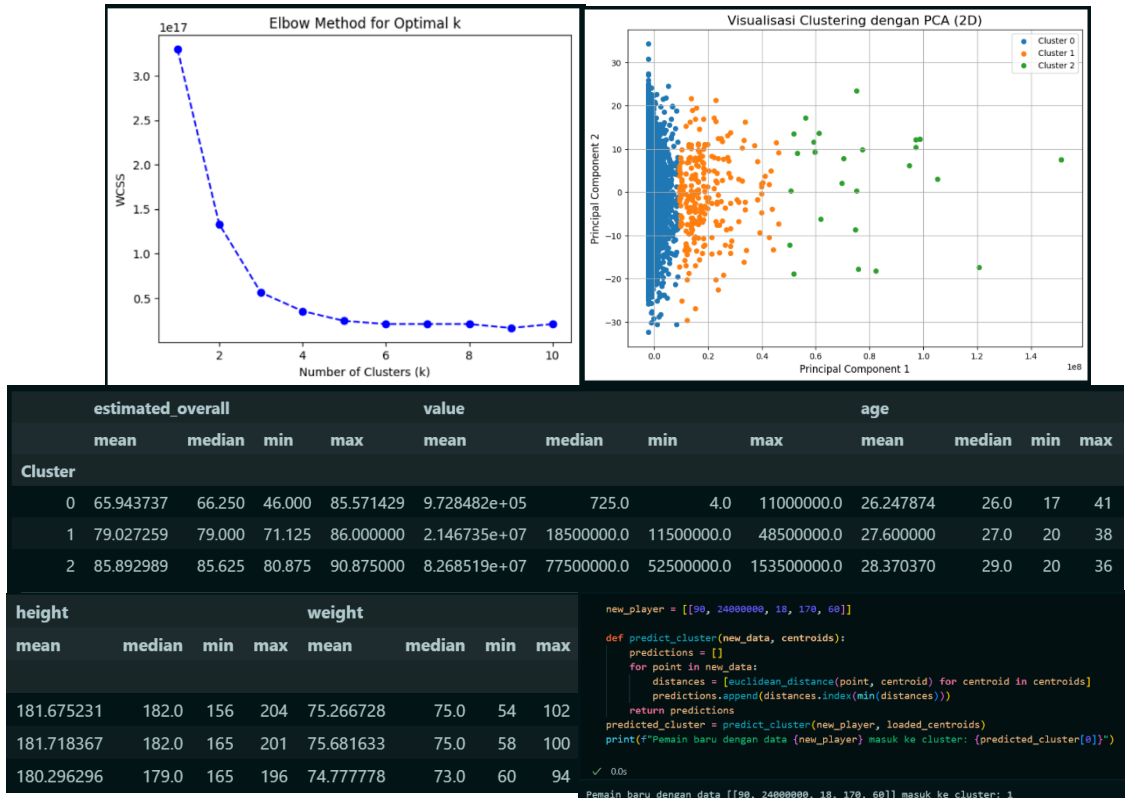
def predict_cluster(new_data, centroids):
    predictions = []
    for point in new_data:
        distances = [euclidean_distance(point, centroid) for centroid in centroids]
        predictions.append(distances.index(min(distances)))
    return predictions
predicted_cluster = predict_cluster(new_player, loaded_centroids)
print(f"Pemain baru dengan data {new_player} masuk ke cluster: {predicted_cluster[0]}")
```

✓ 0.0s
Pemain baru dengan data [[98, 49000000]] masuk ke cluster: 3

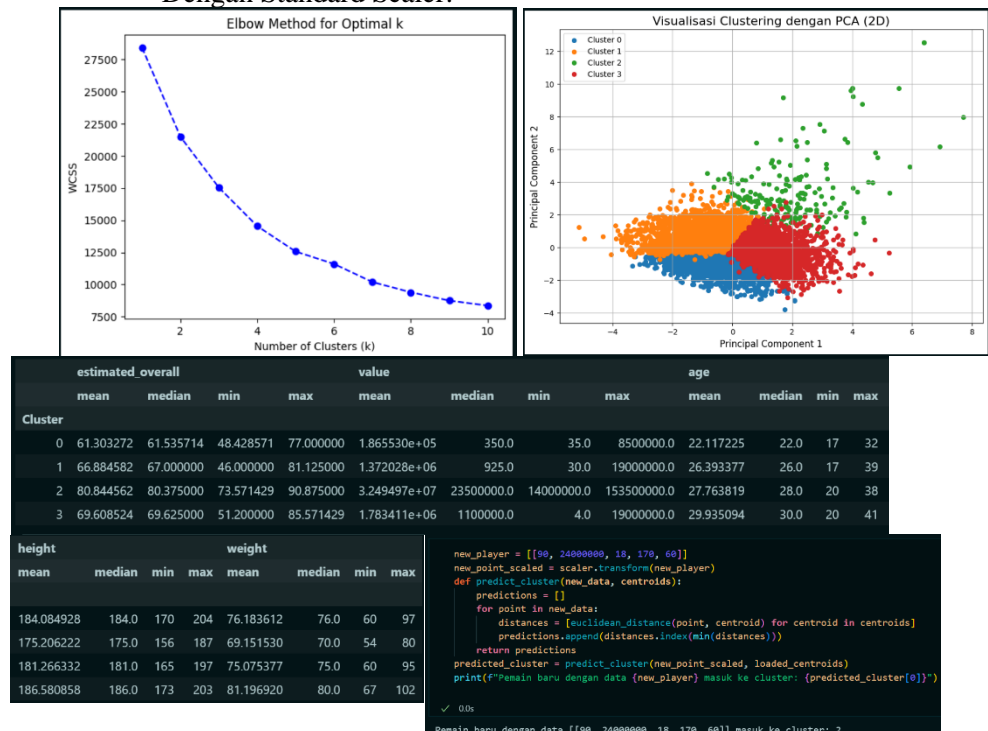
2. estimated_overall, value, age, height, dan weight
Kombinasi ini menambahkan faktor fisik dan usia pemain sebagai parameter clustering.
Tujuan: Mengidentifikasi apakah faktor usia dan fisik memengaruhi pembentukan cluster.

Hasil:

- Tanpa Standard Scaler:



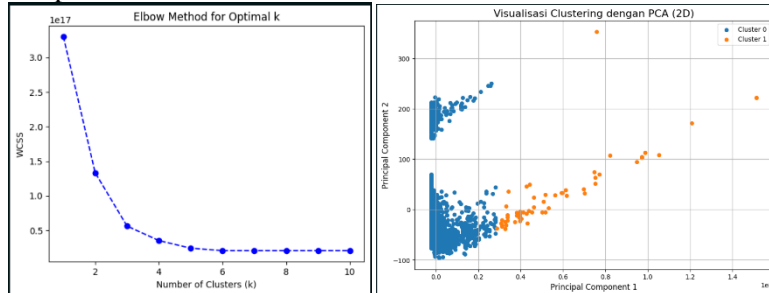
- Dengan Standard Scaler:



- Mengganti estimated_overall dengan atribut teknis pemain.
Atribut teknis (seperti dribbling, ball_control, stamina, strenght, dll) diuji secara bersama dengan value.
Tujuan: Melihat pengaruh jika atribut pemain tidak dijadikan sebagai overall terhadap hasil clustering.

Hasil:

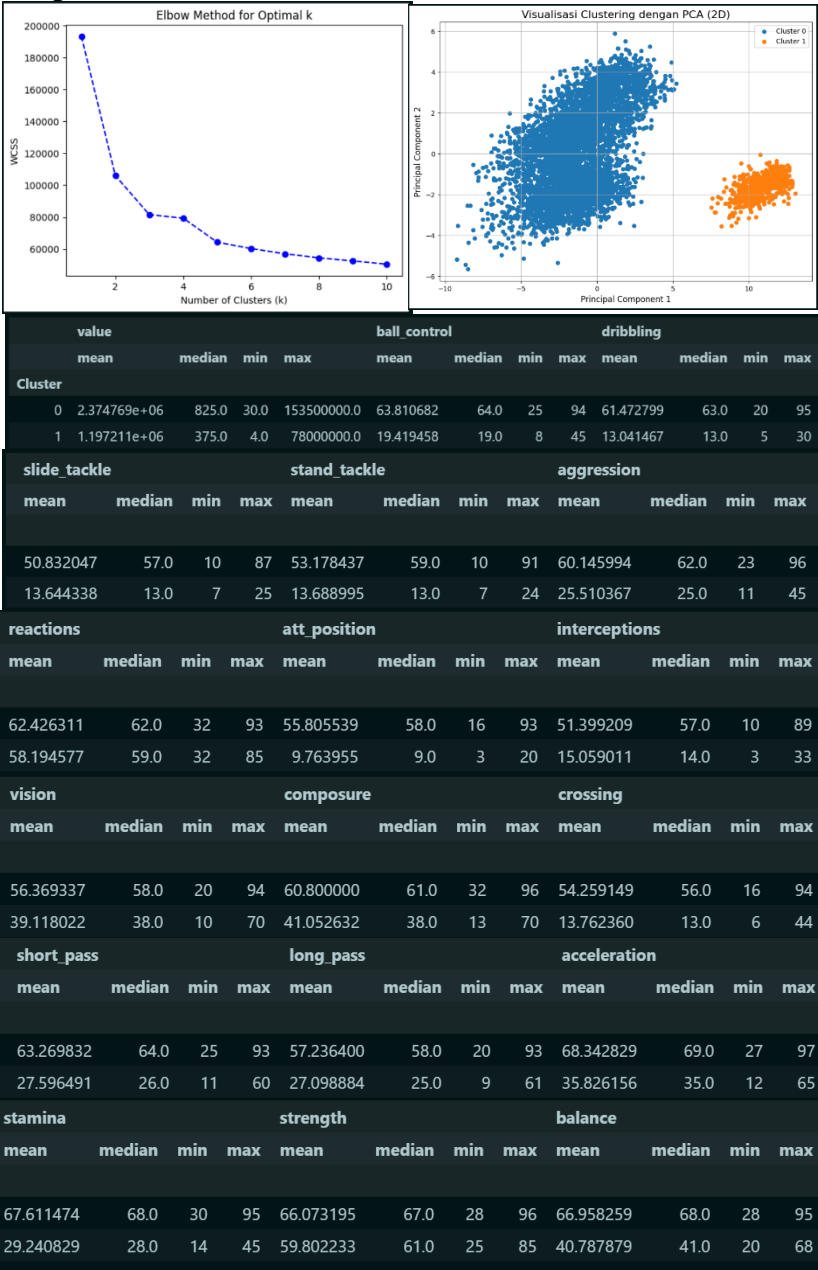
- Tanpa Standard Scaler



Cluster	value				ball_control				dribbling			
	mean	median	min	max	mean	median	min	max	mean	median	min	max
0	1.460793e+06	750.0	4.0	23500000.0	58.526240	63.0	8	91	55.719685	61.0	5	90
1	4.645960e+07	36000000.0	24500000.0	153500000.0	80.676768	84.0	12	94	79.181818	84.0	14	95
slide_tackle				stand_tackle				aggression				
mean	median	min	max	mean	median	min	max	mean	median	min	max	
46.660219	54.0	7	85	48.724342	57.0	7	86	56.115171	59.0	11	96	
50.575758	44.0	15	87	54.262626	48.0	13	91	68.101010	69.0	20	90	
reactions				att_position				interceptions				
mean	median	min	max	mean	median	min	max	mean	median	min	max	
61.587498	62.0	32	87	50.309690	56.0	3	91	47.263658	54.0	3	85	
82.929293	83.0	74	93	74.141414	81.0	10	93	54.464646	50.0	12	89	
vision				composure				crossing				
mean	median	min	max	mean	median	min	max	mean	median	min	max	
54.055167	56.0	10	89	58.225506	60.0	13	90	49.418234	54.0	6	88	
77.616162	80.0	34	94	80.919192	82.0	40	96	70.777778	75.0	12	94	
short_pass				long_pass				acceleration				
mean	median	min	max	mean	median	min	max	mean	median	min	max	
58.979223	62.0	11	85	53.572273	57.0	9	86	64.524091	67.0	12	96	
79.303030	80.0	29	93	73.000000	74.0	18	93	77.757576	80.0	35	97	
stamina				strength				balance				
mean	median	min	max	mean	median	min	max	mean	median	min	max	
63.123948	66.0	14	94	65.284077	66.0	25	96	63.893426	66.0	20	95	
77.666667	78.0	34	95	70.858586	71.0	35	95	74.050505	78.0	29	95	
sprint_speed				agility				jumping				
mean	median	min	max	mean	median	min	max	mean	median	min	max	
64.734372	68.0	13	95	63.529464	66.0	21	93	65.005553	66.0	23	93	
77.626263	80.0	34	97	77.939394	80.0	36	93	69.565657	72.0	39	95	
heading				shot_power				finishing				
mean	median	min	max	mean	median	min	max	mean	median	min	max	
52.024539	55.0	6	93	57.849901	59.0	20	90	45.919936	50.0	4	86	
64.696970	67.0	12	90	76.949495	80.0	37	94	70.848485	77.0	12	94	
long_shots				curve				fk_acc				
mean	median	min	max	mean	median	min	max	mean	median	min	max	
46.745477	51.0	5	87	47.673294	50.0	7	92	42.918144	43.0	4	89	
70.777778	77.0	18	91	72.151515	78.0	12	93	64.808081	69.0	13	94	

penalties				volleys				gk positioning							
mean	median	min	max	mean	median	min	max	mean	median	min	max	mean	median	min	max
47.842200	49.0	8	92	42.501343	44.0	4	87	16.242522	11.0	2	90				
66.939394	71.0	15	91	67.212121	73.0	8	90	13.090909	10.0	4	85				
gk diving				gk handling				gk kicking				gk reflexes			
mean	median	min	max	mean	median	min	max	mean	median	min	max	mean	median	min	max
16.500627	11.0	2	87	16.248254	11.0	2	84	16.121977	11.0	3	90	16.626545	11.0	3	87
12.909091	10.0	4	90	13.010101	10.0	5	87	13.252525	10.0	2	85	13.393939	11.0	3	89
<pre>new_player = [[24000000, 70, 60, 40, 80, 90, 77, 83, 91, 71, 62, 52, 53, 54, 55, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 9, 10, 11, 12, 13, 14]] def predict_cluster(new_data, centroids): predictions = [] for point in new_data: distances = [euclidean_distance(point, centroid) for centroid in centroids] predictions.append(distances.index(min(distances))) return predictions predicted_cluster = predict_cluster(new_player, loaded_centroids) print(f'Player baru dengan data {new_player} masuk ke cluster: {predicted_cluster[0]}')</pre>															
Python															
dengan data [[24000000, 70, 60, 40, 80, 90, 77, 83, 91, 71, 62, 52, 53, 54, 55, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 9, 10, 11, 12, 13, 14]] masuk ke cluster: 1															

- Dengan Standard Scaler

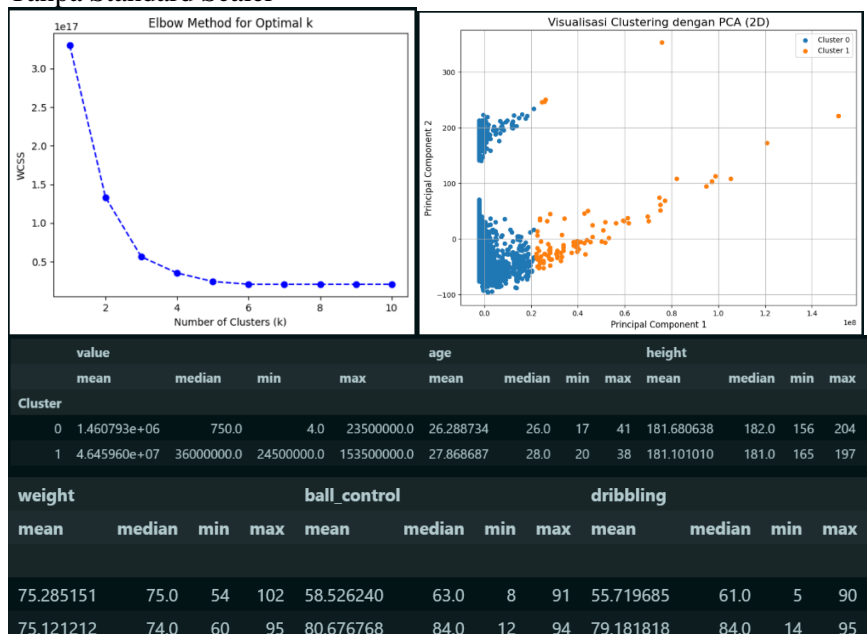


sprint speed				agility				jumping							
mean	median	min	max	mean	median	min	max	mean	median	min	max				
68.505045	69.0	30	97	66.881306	68.0	28	93	66.171909	67.0	31	95				
36.370016	36.0	13	65	38.781499	37.0	21	70	56.322169	58.0	23	84				
gk diving				gk handling				gk kicking				gk reflexes			
mean	median	min	max	mean	median	min	max	mean	median	min	max	mean	median	min	max
10.400396	10.0	2	71	10.370920	10.0	2	65	10.335707	10.0	2	75	10.429871	10.0	3	75
65.114833	65.0	46	90	63.121212	63.0	46	87	62.318979	62.0	42	90	66.074960	66.0	45	89
heading				shot power				finishing							
mean	median	min	max	mean	median	min	max	mean	median	min	max				
57.029674	58.0	22	93	59.585559	61.0	20	94	50.800989	54.0	14	94				
13.673046	13.0	6	39	46.872408	47.0	32	68	10.503987	9.0	4	23				
long shots				curve				fk_acc							
mean	median	min	max	mean	median	min	max	mean	median	min	max				
51.639960	54.0	12	91	52.278932	53.0	15	93	46.938675	45.0	12	94				
11.079745	10.0	5	45	14.406699	13.0	7	69	13.960128	13.0	4	74				
penalties				volleys				gk_positioning							
mean	median	min	max	mean	median	min	max	mean	median	min	max				
51.815232	51.0	17	92	46.894164	47.0	11	90	10.330959	10.0	2	71				
18.826156	18.0	8	70	10.987241	10.0	4	22	63.405104	64.0	43	90				
<pre>new_player = [[24000000, 70, 60, 40, 80, 90, 77, 88, 91, 71, 62, 52, 53, 54, 55, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 9, 10, 11, 12, 13, 14]] new_point_scaled = scaler.transform(new_player) def predict_cluster(new_data, centroids): predictions = [] for point in new_data: distances = [euclidean_distance(point, centroid) for centroid in centroids] predictions.append(distances.index(min(distances))) return predictions predicted_cluster = predict_cluster(new_point_scaled, loaded_centroids) print(f"Pemain baru dengan data {new_player} masuk ke cluster: {predicted_cluster[0]}")</pre>															
Python															

- Mengganti estimated_overall dengan atribut teknis pemain.
Atribut teknis (seperti dribbling, ball_control, stamina, strenght, dll) dan ditambahkan age, height, weight diuji secara bersama dengan value. Semua fitur penting digabungkan untuk mengeksplorasi pengaruh kombinasi variabel fisik, teknis, dan nilai terhadap clustering.
Tujuan: Menyediakan analisis komprehensif tentang pola pengelompokan pemain.

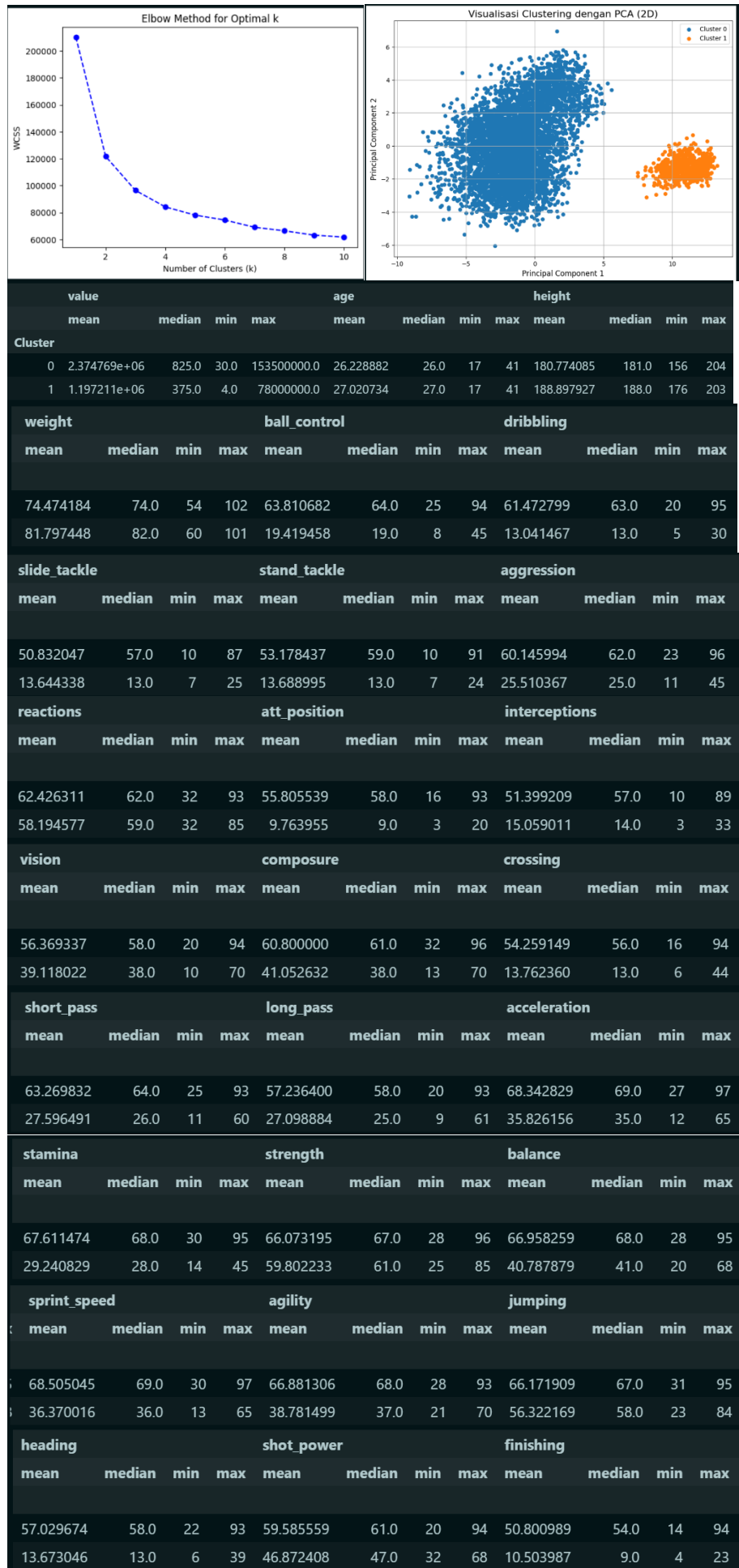
Hasil:

- Tanpa Standard Scaler



slide_tackle				stand_tackle				aggression							
mean	median	min	max	mean	median	min	max	mean	median	min	max				
46.660219	54.0	7	85	48.724342	57.0	7	86	56.115171	59.0	11	96				
50.575758	44.0	15	87	54.262626	48.0	13	91	68.101010	69.0	20	90				
reactions				att_position				interceptions							
mean	median	min	max	mean	median	min	max	mean	median	min	max				
61.587498	62.0	32	87	50.309690	56.0	3	91	47.263658	54.0	3	85				
82.929293	83.0	74	93	74.141414	81.0	10	93	54.464646	50.0	12	89				
vision				composure				crossing							
mean	median	min	max	mean	median	min	max	mean	median	min	max				
54.055167	56.0	10	89	58.225506	60.0	13	90	49.418234	54.0	6	88				
77.616162	80.0	34	94	80.919192	82.0	40	96	70.777778	75.0	12	94				
short_pass				long_pass				acceleration							
mean	median	min	max	mean	median	min	max	mean	median	min	max				
58.979223	62.0	11	85	53.572273	57.0	9	86	64.524091	67.0	12	96				
79.303030	80.0	29	93	73.000000	74.0	18	93	77.757576	80.0	35	97				
stamina				strength				balance							
mean	median	min	max	mean	median	min	max	mean	median	min	max				
63.123948	66.0	14	94	65.284077	66.0	25	96	63.893426	66.0	20	95				
77.666667	78.0	34	95	70.858586	71.0	35	95	74.050505	78.0	29	95				
sprint_speed				agility				jumping							
mean	median	min	max	mean	median	min	max	mean	median	min	max				
64.734372	68.0	13	95	63.529464	66.0	21	93	65.005553	66.0	23	93				
77.626263	80.0	34	97	77.939394	80.0	36	93	69.565657	72.0	39	95				
heading				shot_power				finishing							
mean	median	min	max	mean	median	min	max	mean	median	min	max				
52.024539	55.0	6	93	57.849901	59.0	20	90	45.919936	50.0	4	86				
64.696970	67.0	12	90	76.949495	80.0	37	94	70.848485	77.0	12	94				
long_shots				curve				fk_acc							
mean	median	min	max	mean	median	min	max	mean	median	min	max				
46.745477	51.0	5	87	47.673294	50.0	7	92	42.918144	43.0	4	89				
70.777778	77.0	18	91	72.151515	78.0	12	93	64.808081	69.0	13	94				
penalties				volleys				gk_positioning							
mean	median	min	max	mean	median	min	max	mean	median	min	max				
47.842200	49.0	8	92	42.501343	44.0	4	87	16.242522	11.0	2	90				
66.939394	71.0	15	91	67.212121	73.0	8	90	13.090909	10.0	4	85				
gk_diving				gk_handling				gk_kicking				gk_reflexes			
mean	median	min	max	mean	median	min	max	mean	median	min	max	mean	median	min	max
16.500627	11.0	2	87	16.248254	11.0	2	84	16.121977	11.0	3	90	16.626545	11.0	3	87
12.909091	10.0	4	90	13.010101	10.0	5	87	13.252525	10.0	2	85	13.393939	11.0	3	89
player = [[24000000, 18, 170, 60, 70, 60, 40, 80, 90, 77, 88, 91, 71, 62, 52, 53, 54, 55, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 9, 10, 11, 12, 13, 14]]															
predict_cluster(new_data, centroids): predictions = [] for point in new_data: distances = [euclidean_distance(point, centroid) for centroid in centroids] predictions.append(distances.index(min(distances))) return predictions icted_cluster = predict_cluster(new_player, loaded_centroids) t("Pemain baru dengan data {new_player} masuk ke cluster: {predicted_cluster[0]}")															
✓ Oke															
Python															

- Dengan Standard Scaler



long_shots				curve				fk_acc							
mean	median	min	max	mean	median	min	max	mean	median	min	max				
51.639960	54.0	12	91	52.278932	53.0	15	93	46.938675	45.0	12	94				
11.079745	10.0	5	45	14.406699	13.0	7	69	13.960128	13.0	4	74				
penalties				volleys				gk_positioning							
mean	median	min	max	mean	median	min	max	mean	median	min	max				
51.815232	51.0	17	92	46.894164	47.0	11	90	10.330959	10.0	2	71				
18.826156	18.0	8	70	10.987241	10.0	4	22	63.405104	64.0	43	90				
gk_diving				gk_handling				gk_kicking				gk_reflexes			
mean	median	min	max	mean	median	min	max	mean	median	min	max	mean	median	min	max
10.400396	10.0	2	71	10.370920	10.0	2	65	10.335707	10.0	2	75	10.429871	10.0	3	75
65.114833	65.0	46	90	63.121212	63.0	46	87	62.318979	62.0	42	90	66.074960	66.0	45	89

```
new_player = [[24000000, 18, 170, 60, 70, 60, 40, 80, 90, 77, 88, 91, 71, 62, 52, 53, 54, 55, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 9, 10, 11, 12, 13, 14]]
new_point_scaled = scaler.transform(new_player)
def predict_cluster(new_data, centroids):
    predictions = []
    for point in new_data:
        distances = [euclidean_distance(point, centroid) for centroid in centroids]
        predictions.append(distances.index(min(distances)))
    return predictions
predicted_cluster = predict_cluster(new_point_scaled, loaded_centroids)
print(f"Pemain baru dengan data {new_player} masuk ke cluster: {predicted_cluster[0]}")
```

✓ 0.0s

Python

[24000000, 18, 170, 60, 70, 60, 40, 80, 90, 77, 88, 91, 71, 62, 52, 53, 54, 55, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 9, 10, 11, 12, 13, 14] masuk ke cluster:

Tabel Perbandingan Hasil:

Fitur	Standard Scaler	Optimal Cluster	Silhouette Score	Davies-Bouldin Index
estimated_overall dan value	Tidak	3	0.90	0.441
estimated_overall dan value	Ya	4	0.525	0.57
estimated_overall, value, age, height, weight	Tidak	3	0.90	0.441
estimated_overall, value, age, height, weight	Ya	4	0.241	1.186
Atribut teknis, value	Tidak	2	0.931	0.482
Atribut teknis, value	Ya	2	0.568	0.589
Atribut teknis, value, age, height, weight	Tidak	2	0.931	0.482
Atribut teknis, value, age, height, weight	Ya	2	0.544	0.638

b. Penentuan Metode Terbaik

Berdasarkan tujuan proyek dan hasil eksperimen:

Fitur Terbaik:

- Atribut Teknis dan Value atau Atribut Teknis, Value, Age, Height, dan Weight adalah pilihan terbaik. Kombinasi ini mencakup atribut yang relevan untuk mengidentifikasi tipe pemain dan nilai mereka.
- Silhouette Score dan DBI menunjukkan performa optimal dengan cluster

yang kompak dan terpisah.

Penggunaan Scaler:

- Tidak menggunakan scaler memberikan hasil lebih baik secara konsisten.
- Hal ini masuk akal karena perbedaan skala antara atribut mungkin mencerminkan karakteristik asli pemain (misalnya, tinggi badan vs kemampuan teknis).

Optimal Cluster:

- 2 cluster dari eksperimen "Atribut Teknis dan Value" dan "Atribut teknis, value, age, height, dan weight" memberikan Silhouette Score tertinggi (0.931) dan DBI terendah (0.482).
- Alternatif: 3 cluster pada eksperimen "Estimated Overall, Value, Age, Height, dan Weight tanpa scaler" juga menunjukkan kinerja baik (Silhouette Score = 0.90, DBI = 0.441).

c. Analisis atau Penjelasan Hasil Eksperimen

Kami memutuskan menggunakan fitur `estimated_overall` dan `value` dengan Standard Scaler. Alasan pemilihan ini adalah meskipun hasil evaluasi clustering menggunakan Silhouette Score (0.525) dan Davies-Bouldin Index (DBI) (0.57) tidak terlalu bagus dibandingkan eksperimen lainnya, namun hasil clusteringnya lebih variatif dan memberikan lebih banyak wawasan.

Hasil Analisis:

- Penentuan Optimal Cluster (Elbow Method):
 - Dari grafik Elbow Method, jumlah cluster optimal adalah 4.
 - Pemilihan 4 cluster memungkinkan variasi yang lebih luas untuk interpretasi data pemain.
- Distribusi Cluster
 - Cluster 0: Pemain dengan kemampuan rendah hingga menengah, memiliki nilai pasar kecil (mean value ~ 62.1K).
 - Cluster 1: Pemain dengan kemampuan menengah, nilai pasar mulai meningkat (mean value ~ 1.6M).
 - Cluster 2: Pemain dengan kemampuan di atas rata-rata (mean `estimated_overall` ~ 78.95) dan nilai pasar tinggi (mean value ~ 19.9M).
 - Cluster 3: Pemain elite/top-tier, kemampuan sangat tinggi (mean `estimated_overall` ~ 85.89) dan nilai pasar sangat besar (mean value ~ 82.7M).

- Prediksi Pemain Baru
 - Untuk pemain baru dengan data `estimated_overall` = 98 dan `value` = 49 juta, hasil prediksi menunjukkan pemain ini masuk ke Cluster 3.
 - Hal ini sesuai dengan karakteristik Cluster 3 sebagai pemain top-tier dengan kemampuan dan nilai pasar yang sangat tinggi.

5. Kesimpulan dan Saran

a. Kesimpulan

Proyek ini bertujuan untuk mengelompokkan pemain sepak bola berdasarkan kemampuan dan nilai pasar menggunakan data dari FIFA24 Players Stats. Dengan fokus pada fitur `estimated_overall` dan `value`, serta pemrosesan menggunakan Standard Scaler, eksperimen menghasilkan 4 cluster optimal yang memberikan variasi signifikan dalam analisis. Cluster ini mampu menggambarkan perbedaan pemain dari segi kualitas dan nilai ekonomi, mulai dari pemain dengan kemampuan rendah dan nilai kecil hingga pemain elite dengan nilai pasar tinggi. Meskipun metrik evaluasi, seperti Silhouette Score (0.525) dan Davies-Bouldin Index (0.57), tidak menunjukkan performa clustering yang sempurna, hasil ini tetap bermanfaat untuk mencapai tujuan utama proyek, yakni memberikan rekomendasi pemain berbakat dan memahami karakteristik setiap cluster.

b. Kendala Berdasarkan Hasil Eksperimen

Salah satu kendala utama yang dihadapi adalah cluster dengan nilai Silhouette Score dan DBI yang baik cenderung menghasilkan hanya 2 cluster. Hal ini mengurangi variasi dalam hasil clustering, sehingga sulit untuk membedakan kelompok pemain dengan lebih spesifik, seperti pemain muda berbakat atau pemain murah potensial. Selain itu, kendala lainnya adalah keterbatasan informasi dalam dataset, yang lebih berfokus pada atribut teknis dan nilai pasar tetapi kurang mencakup aspek penting lainnya, seperti pengalaman, posisi utama, atau perkembangan karier pemain dari waktu ke waktu. Hal ini menyebabkan analisis yang dihasilkan menjadi kurang kaya dan tidak sepenuhnya mencerminkan kebutuhan nyata dalam pengambilan keputusan strategis, seperti perekrutan pemain atau penilaian investasi. Selain itu, beberapa eksperimen menunjukkan adanya overlap antar cluster, yang mengindikasikan bahwa metode scaling atau pemilihan fitur mungkin belum optimal.

c. Rencana Lanjut Hasil Eksperimen

Untuk pengembangan lebih lanjut, proyek ini akan fokus pada penambahan data pendukung seperti statistik perkembangan pemain, posisi utama, waktu bermain, dan performa pertandingan untuk memperkaya analisis. Algoritma clustering lanjutan seperti Gaussian Mixture Models (GMM) atau Spectral Clustering akan digunakan untuk menangani cluster yang lebih kompleks. Hasil clustering juga akan diintegrasikan ke dalam sistem rekomendasi yang dinamis, memberikan saran berbasis kebutuhan spesifik, seperti anggaran atau tipe pemain yang diinginkan. Visualisasi interaktif akan disiapkan untuk mempermudah interpretasi hasil. Selain itu, proyek ini akan dievaluasi dengan data terbaru untuk memastikan relevansi dan manfaat nyata, seperti mengidentifikasi pemain berbakat potensial atau mengelola anggaran transfer secara efektif. Hasil ini juga akan diintegrasikan dalam sistem rekomendasi pemain berdasarkan kebutuhan klub, seperti pencarian Striker, Playmaker, atau Defender, yang mendukung pengembangan strategi tim di masa depan.