

Laporan Tugas Besar Dasar Kecerdasan Artifisial Penerapan Fuzzy System dalam Klasifikasi Pemain NBA Musim 2023-2024



Oleh:

Ezra Mangasi Andika Sibuea	103012330117
Muhammad Arya Dwi Kesuma	103012300165

**PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM
BANDUNG
2025**

A. Pendahuluan

1. Latar Belakang

Mengutip dari *website* milik Kementerian Koordinator Bidang Pembangunan Manusia dan Kebudayaan Republik Indonesia atau Kemenko PMK tentang Desain Besar Olahraga Nasional yang mana DBON ini mencakup olahraga rekreasi, olahraga pendidikan, olahraga prestasi, dan industri olahraga atau wisata olahraga. Target jangka menengah dan panjang desain besar olahraga 2021-2045.

“Dengan memperhatikan bahwa selama 20 tahun terakhir partisipasi masyarakat berolahraga masih rendah, lanjut Wapres, yaitu di sekitar 30 persen, maka perlu ditingkatkan kesadaran masyarakat dalam berolahraga agar tercapai angka partisipasi sebesar 40 persen pada tahun 2024, dan perlu terus ditingkatkan sehingga olahraga menjadi bagian dari gaya hidup masyarakat”.

Sehubungan dengan hal tersebut, kegiatan olahraga yang diadakan di setiap sekolah bisa menjadi fondasi untuk memenuhi rencana atau program pemerintah yang satu ini. Salah satu hal yang dapat dilakukan untuk melaksanakannya adalah mengadakan program scouting. Dalam dunia olahraga, terutama bola basket, scouting pemain merupakan aspek krusial yang dapat memengaruhi strategi dan pemilihan pemain dalam sebuah tim. Data statistik pemain menjadi dasar utama dalam menilai performa individu, termasuk menentukan kemampuan spesifik seperti menembak. Seiring dengan perkembangan zaman, orang-orang akan lebih memilih cara scouting yang lebih modern, teknologi yang menawarkan penilaian yang lebih objektif dan akurat serta lebih memfasilitasi pengambilan keputusan secara strategis dan meningkatkan kinerja tim (Yunus et al., dikutip dalam Logan et al. 2018).

Kemampuan menembak, khususnya kemampuan mencetak poin dari luar garis tiga poin (3-point shooting), merupakan salah satu elemen penting dalam permainan bola basket modern. Pemain dengan kemampuan menembak yang baik dapat memberikan keunggulan kompetitif bagi sebuah tim. Namun, analisis data pemain secara manual pastinya membutuhkan waktu dan sangat mungkin dipengaruhi subjektivitas. Oleh karena itu, diperlukan sistem berbasis teknologi yang dapat mengelompokkan pemain berdasarkan kemampuan menembaknya secara otomatis dan akurat.

Pendekatan sistem fuzzy, seperti Mamdani dan Sugeno, menawarkan solusi yang efektif untuk menangani ketidakpastian dalam penilaian kualitatif dan kuantitatif. Sistem ini dapat diterapkan untuk mengelompokkan pemain ke dalam dua kategori utama, yaitu "Pemain Pencetak 3 Poin Aktif" dan "Pemain Bukan Pencetak 3 Poin", berdasarkan parameter statistik seperti percobaan dan keberhasilan tembakan dua dan tiga poin, serta efisiensi tembakan.

Dengan implementasi sistem fuzzy ini, program scouting di Indonesia dapat lebih terstruktur, objektif, dan berbasis data. Ini dapat membantu mempercepat identifikasi pemain berbakat, meningkatkan keputusan strategis, dan mengurangi ketergantungan pada pengamatan subjektif.

2. Rumusan Masalah

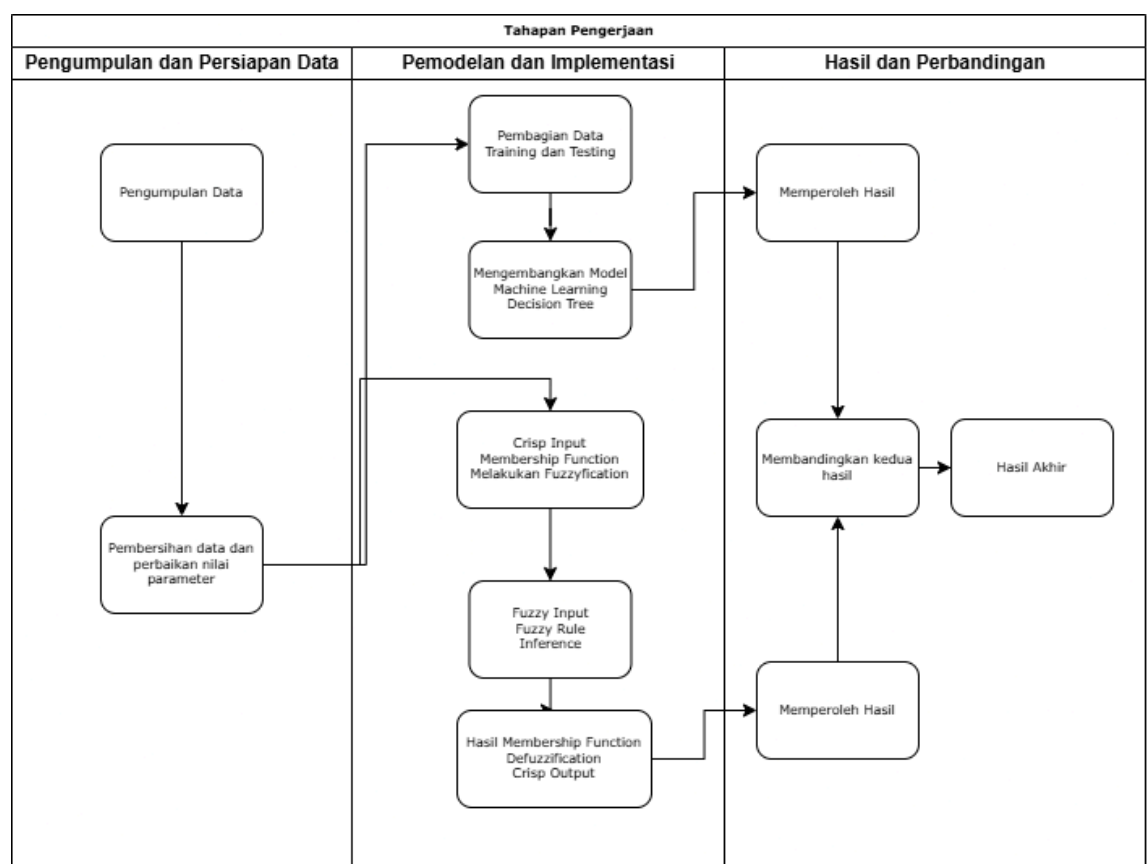
- Bagaimana membangun sistem yang dapat mengklasifikasikan pemain bola basket berdasarkan kemampuan menembak dengan menggunakan sistem fuzzy Mamdani?
- Bagaimana sistem fuzzy dapat menangani parameter statistik yang kompleks dan tidak pasti untuk menghasilkan kategori pemain secara akurat?

- Bagaimana membandingkan efektivitas metode fuzzy Mamdani dengan model Decision Tree?

3. Tujuan

- Mengembangkan sistem berbasis fuzzy (Mamdani dan Sugeno) untuk mengelompokkan pemain bola basket berdasarkan kemampuan menembak.
- Mengimplementasikan dan menguji sistem menggunakan data statistik pemain NBA musim 2023-2024.
- Membandingkan kinerja sistem fuzzy dengan model pembelajaran mesin Decision Tree untuk menilai keunggulan dan kelemahan masing-masing pendekatan.

4. Solusi



a. Pengumpulan dan Persiapan Data

Tahapan ini mencakup semua aktivitas yang diperlukan untuk mengumpulkan, mempersiapkan, dan memvalidasi data statistik pemain NBA. Langkah-Langkah yang dilakukan diantaranya:

- Pengumpulan data statistik pemain NBA musim 2023-2024.
- Pembersihan data (menghapus data yang tidak lengkap atau inkonsisten).
- Normalisasi nilai parameter seperti jumlah permainan, persentase tiga poin, dan efisiensi.

b. Pemodelan dan Implementasi

Tahapan ini melibatkan pengembangan model klasifikasi berbasis fuzzy Mamdani, fuzzy Sugeno, dan algoritma pembelajaran mesin untuk mengelompokkan pemain.

Langkah-Langkah yang dilakukan diantaranya:

- Membuat aturan fuzzy (Mamdani dan Sugeno).
- Mengimplementasikan model fuzzy untuk klasifikasi pemain.
- Mengembangkan model pembelajaran mesin Decision Tree untuk perbandingan.
- Mengintegrasikan semua model untuk pengujian.

c. Pengujian, Evaluasi, dan Perbandingan

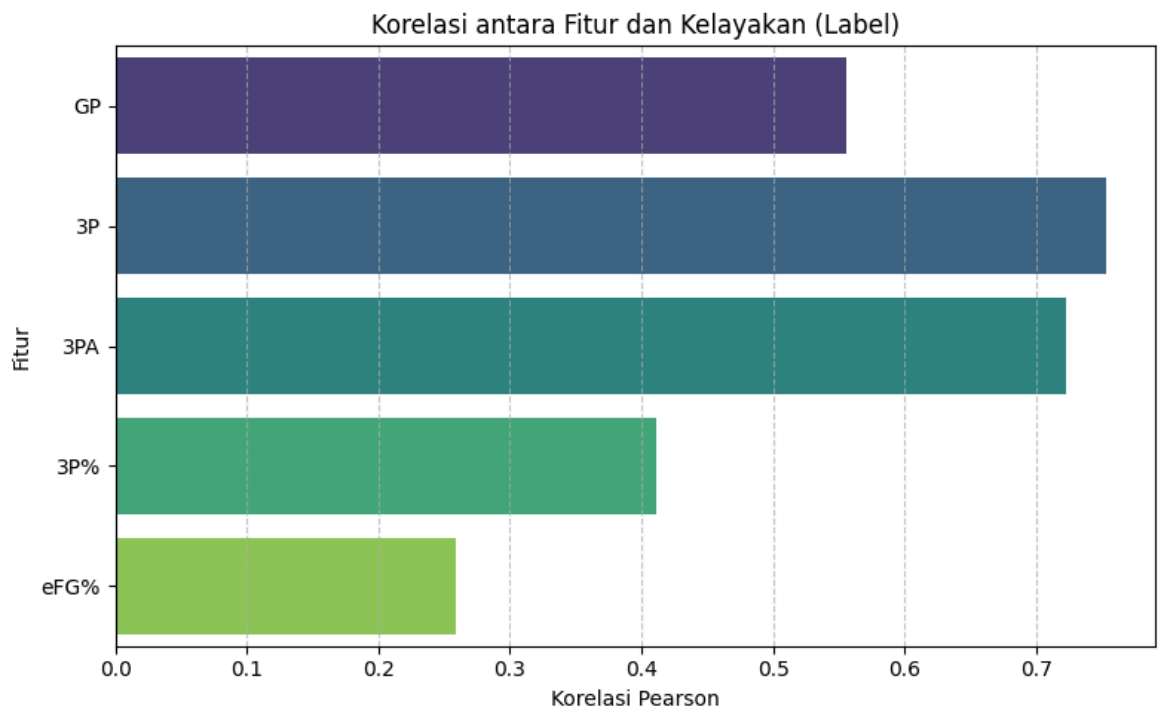
Tahapan akhir yang mencakup pengujian performa model, analisis hasil, dan memberikan rekomendasi berdasarkan evaluasi.

Langkah-Langkah:

- Melakukan validasi hasil klasifikasi dengan data uji.
- Mengevaluasi performa model berdasarkan akurasi.
- Membandingkan hasil fuzzy logic dengan metode pembelajaran mesin decision tree

B. Metode

1. Paparan, statistik, dan sumber dari dataset yang digunakan



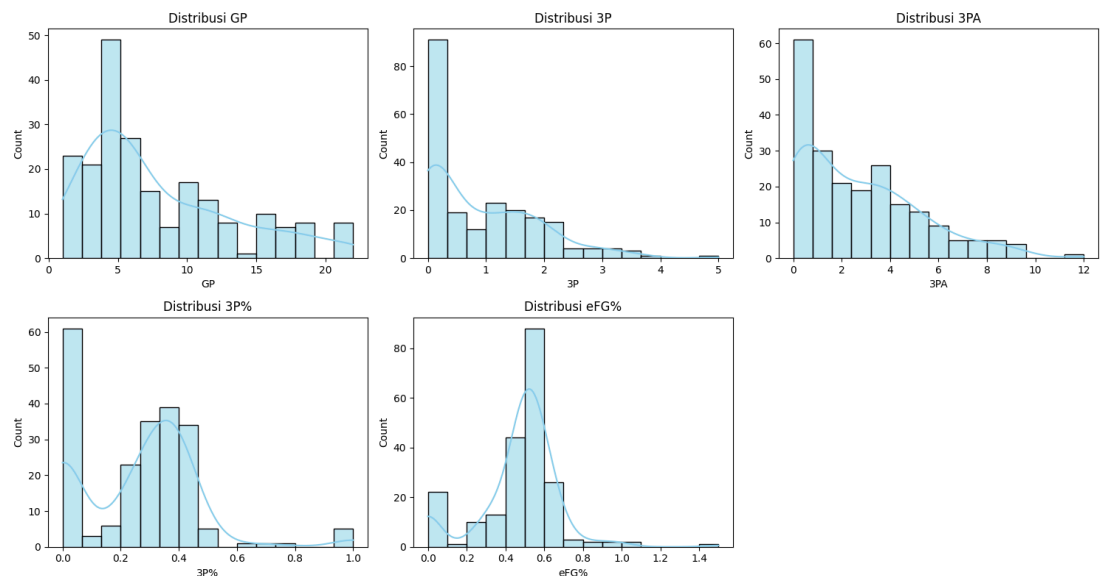
3P (Jumlah Tembakan 3 Poin yang Masuk) memiliki korelasi positif tertinggi terhadap kelayakan. Ini menunjukkan bahwa pemain yang memiliki jumlah tembakan 3 Poin yang masuk dengan jumlah yang banyak cenderung lebih sering diklasifikasikan sebagai layak.

3PA (Percobaan Tembakan 3 Poin) juga berkorelasi positif cukup kuat. Banyaknya percobaan dalam tembakan 3 poin menjadi indikator performa yang relevan untuk klasifikasi pemain.

GP (Jumlah Pertandingan) memiliki korelasi yang cukup positif. Menunjukkan bahwa bukan hanya jumlah tembakan 3 poin yang penting, tetapi juga jumlah pertandingannya.

3P% (Persentase Tembakan 3 Poin) Korelasinya sedang. Ini menunjukkan bahwa persentase menembak yang tinggi tidak selalu berarti pemain tersebut berkualitas efisiensi lebih penting.

eFG% (Effective Field Goal Percentage) korelasinya lemah terhadap kelayakan. Walaupun menit bermain penting, model tidak menganggap eFG% sebagai indikator utama dalam menentukan kelayakan.



Untuk data GP (Games Played), histogram menunjukkan bahwa sebagian besar pemain bermain sekitar 4–6 pertandingan. Sebaran cenderung ke kiri, menunjukkan hanya sedikit pemain yang bermain dalam jumlah banyak.

Pada histogram 3P (Tembakan 3 Poin yang Masuk), mayoritas pemain memiliki kurang dari 2 tembakan 3 poin per game, dengan puncak distribusi pada kisaran 0–1. Hanya sedikit pemain dengan nilai yang benar-benar tinggi.

Pada histogram 3PA (Percobaan 3 Poin), distribusi lebih menyebar dibanding 3P, namun puncak tetap pada kisaran 0 tembakan per game. Hal ini menunjukkan bahwa meskipun banyak pemain mencoba menembak tiga poin, hanya sebagian yang sering berhasil.

Histogram 3P% (Persentase Tembakan 3 Poin) membentuk distribusi yang bisa dibilang tidak terlalu teratur, dengan konsentrasi pada 0–10% dan naik lagi pada 20–45%. Ini cukup mencerminkan akurasi yang umum atau tidak spesial oleh pemain NBA dalam menembak dari jarak jauh.

Untuk eFG% (Efektif Field Goal %) distribusinya agak simetris dan terpusat di sekitar 40–70%, dengan sebagian kecil pemain sangat efisien (>65%) dan sebagian besarnya kurang efisien (<40%).

Sumber dataset:

<https://www.kaggle.com/datasets/vivovinco/2023-2024-nba-player-stats>

2. Paparan preprocessing dataset

a. Import Dataset dan pemeriksaan data

Dataset ini berisi berbagai fitur statistik untuk setiap pemain NBA yang berpartisipasi dalam babak playoff, termasuk kolom target Label yang menyatakan apakah pemain termasuk pemain yang menambak 3 poin secara aktif (1) atau tidak (0).

b. Pemeriksaan Missing Values

Langkah ini dilakukan untuk mengecek kelengkapan data: Dataset diperiksa apakah ada nilai kosong. Dalam kasus ini, tidak ditemukan missing values, sehingga tidak dilakukan penghapusan.

c. Pemeriksaan dan Pembersihan Nilai Tidak Valid

Beberapa logika validasi digunakan untuk menjaga kualitas data: Nilai 3P% dan eFG% diperiksa agar berada dalam rentang [0, 1] Validasi inkonsistensi logis, seperti: jika 3PA = 0 maka 3P seharusnya juga 0. Jika ditemukan kasus seperti ini, dapat dilakukan koreksi atau pengecualian.

d. Konversi Tipe Data (Jika Diperlukan)

Label klasifikasi harus dipastikan dalam bentuk numerik. Hal ini diperlukan agar model Decision Tree maupun evaluasi dapat berjalan dengan benar.

e. Pemilihan dan Penyaringan Kolom

Untuk fokus pada fitur yang relevan secara statistik berfokus pada topik tembakan 3 poin, hanya kolom berikut yang digunakan:

[GP, 3P, 3PA, 3P%, eFG%, Label]

Kolom seperti Player dihapus karena bersifat identitas dan tidak memberikan informasi prediktif.

f. Skalasi atau Normalisasi (Jika Perlu)

Dalam proyek ini tidak dilakukan normalisasi atau standarisasi, karena: Decision Tree tidak sensitif terhadap skala. Fuzzy Logic menggunakan nilai absolut dan fungsi keanggotaan yang telah disesuaikan dengan rentang aktual fitur

g. Pembuatan Label

Pada dataset ini, kolom Label sudah tersedia, dan diperoleh melalui perhitungan yang dilakukan secara subjektif.

h. Pemisahan Dataset (Opsional)

Langkah ini dilakukan untuk model Decision Tree. Namun untuk model Fuzzy Logic, seluruh data digunakan langsung sebagai input sistem inferensi, karena sifatnya bukan supervised learning dalam bentuk tradisional.

3. Penjelasan mengenai rancangan dua metode yang digunakan

a. Nilai Linguistik

Atribut	Jumlah Nilai Linguistik	Jenis Linguistik
GP	3	Sedikit, Sedang Banyak
3P	3	Rendah, Sedang, tinggi
3PA	3	Rendah, Sedang, tinggi
3P%	3	Rendah, Sedang, tinggi
eFG%	3	Rendah, Sedang, tinggi
Kelayakan	2	Layak, Tidak Layak

b. Fungsi Keanggotaan dan Batas-batas nilai linguistik

```

GP = ctrl.Antecedent(np.arange(1, 29, 1), 'GP')
threeP = ctrl.Antecedent(np.arange(0.0, 10.1, 0.1), 'threeP')
threePA = ctrl.Antecedent(np.arange(0.0, 15.1, 0.1), 'threePA')
threeP_pct = ctrl.Antecedent(np.arange(0.0, 1.01, 0.01), 'threeP_pct')
efg = ctrl.Antecedent(np.arange(0.00, 1.51, 0.01), 'efg')
kelayakan = ctrl.Consequent(np.arange(0, 1.1, 0.01), 'kelayakan')

GP['sedikit'] = fuzz.trimf(GP.universe, [1, 1, 8])
GP['cukup'] = fuzz.trimf(GP.universe, [1, 8, 14])
GP['banyak'] = fuzz.trimf(GP.universe, [8, 28, 28])
GP.view()

threeP['rendah'] = fuzz.trimf(threeP.universe, [0.0, 0.0, 5.0])
threeP['sedang'] = fuzz.trimf(threeP.universe, [0.0, 5.0, 10.0])
threeP['tinggi'] = fuzz.trimf(threeP.universe, [5.0, 10.0, 10.0])
threeP.view()

threePA['sedikit'] = fuzz.trimf(threePA.universe, [0.0, 0.0, 7.5])
threePA['sedang'] = fuzz.trimf(threePA.universe, [0.0, 7.5, 15.0])
threePA['banyak'] = fuzz.trimf(threePA.universe, [7.5, 15.0, 15.0])
threePA.view()

threeP_pct['rendah'] = fuzz.trimf(threeP_pct.universe, [0.0, 0.0, 0.5])
threeP_pct['cukup'] = fuzz.trimf(threeP_pct.universe, [0.0, 0.5, 1.0])
threeP_pct['tinggi'] = fuzz.trimf(threeP_pct.universe, [0.5, 1.0, 1.0])
threeP_pct.view()

efg['rendah'] = fuzz.trimf(efg.universe, [0.0, 0.0, 0.75])
efg['cukup'] = fuzz.trimf(efg.universe, [0.0, 0.75, 1.5])
efg['tinggi'] = fuzz.trimf(efg.universe, [0.75, 1.5, 1.5])
efg.view()

kelayakan['tidak'] = fuzz.trimf(kelayakan.universe, [0.0, 0.0, 0.5])
kelayakan['layak'] = fuzz.trimf(kelayakan.universe, [0.5, 1.0, 1.0])
kelayakan.view()

```

1. Variabel Input: threeP(Mewakili jumlah tembakan 3 poin per pertandingan)

Rendah : [0.0, 0.0, 5.0]

Sedang : [0.0, 5.0, 10.0]

Tinggi : [5.0, 5.0, 10.0]

2. Variabel Input: threePA(Jumlah percobaan tembakan 3 poin)

Sedikit : [0.0, 0.0, 7.5]

Sedang : [0.0, 7.5, 15.0]

Banyak : [7.5, 7.5, 15.0]

3. Variabel Input: threeP_pct(Persentase keberhasilan tembakan 3 poin)

Rendah : [0.0, 0.0, 0.5]

Cukup : [0.0, 0.5, 1.0]

Tinggi : [0.5, 0.5, 1.0]

4. Variabel Input: efg(Effective Field Goal Percentage)

Rendah : [0.0, 0.0, 0.5]

Cukup : [0.0, 0.5, 1.0]

Tinggi : [0.5, 0.5, 1.0]

5. Variabel Input: GP(Jumlah pertandingan / games played)

Sedikit : [1, 1, 8]

Cukup : [1, 8, 14]

Banyak : [8, 28, 28]

6. Variabel Output: kelayakan (Mewakili keputusan apakah pemain layak atau tidak layak digolongkan kedalam pemain penembak 3 poin)

Tidak Layak : [0.0, 0.0, 0.5]

Layak : [0.5, 1.0, 1.0]

(Biner)

c. Fuzzy Rule

```
rules = [  
    ctrl.Rule(threeP['rendah'], kelayakan['tidak']),  
    ctrl.Rule(threeP['sedang'] & GP['sedikit'], kelayakan['tidak']),  
    ctrl.Rule(threeP['sedang'] & GP['banyak'], kelayakan['layak']),  
    ctrl.Rule(threeP['tinggi'] & GP['sedikit'] & threeP['rendah'] & threePA['sedikit'], kelayakan['tidak']),  
    ctrl.Rule(threeP['tinggi'] & GP['sedikit'] & threeP['rendah'] & threePA['sedang'], kelayakan['layak']),  
    ctrl.Rule(threeP['tinggi'] & GP['sedikit'] & threeP['rendah'] & threePA['banyak'], kelayakan['tidak']),  
    ctrl.Rule(threeP['tinggi'] & GP['sedikit'] & threeP['tinggi'], kelayakan['layak']),  
    ctrl.Rule(threeP['tinggi'] & GP['banyak'] & efg['rendah'], kelayakan['tidak']),  
    ctrl.Rule(threeP['tinggi'] & GP['banyak'] & efg['cukup'] & GP['sedikit'], kelayakan['tidak']),  
    ctrl.Rule(threeP['tinggi'] & GP['banyak'] & efg['cukup'] & GP['banyak'] & threeP_pct['rendah'], kelayakan['tidak']),  
    ctrl.Rule(threeP['tinggi'] & GP['banyak'] & efg['cukup'] & GP['banyak'] & threeP_pct['cukup'], kelayakan['layak']),  
    ctrl.Rule(threeP['tinggi'] & GP['banyak'] & efg['tinggi'], kelayakan['layak'])  
]
```

Aturan Dasar

IF threeP rendah THEN kelayakan tidak layak

IF threeP sedang AND GP sedikit THEN kelayakan tidak layak

IF threeP sedang AND GP banyak THEN kelayakan layak

Aturan Kombinasi Tembakan 3 Poin dan Percobaan

IF threeP tinggi AND GP sedikit AND threeP rendah AND threePA sedikit THEN kelayakan tidak layak

IF threeP tinggi AND GP sedikit AND threeP rendah AND threePA sedang THEN kelayakan layak

IF threeP tinggi AND GP sedikit AND threeP rendah AND threePA banyak THEN kelayakan tidak layak

IF threeP tinggi AND GP sedikit AND threeP tinggi THEN kelayakan layak

Aturan Kombinasi Tembakan dan Efektivitas

IF threeP tinggi AND GP banyak AND efg rendah THEN kelayakan tidak layak

IF threeP tinggi AND GP banyak AND efg cukup AND GP sedikit
 THEN kelayakan tidak layak
 IF threeP tinggi AND GP banyak AND efg cukup AND GP banyak
 AND threeP_pct rendah THEN kelayakan tidak layak
 IF threeP tinggi AND GP banyak AND efg cukup AND GP banyak
 AND threeP_pct cukup THEN kelayakan layak
 IF threeP tinggi AND GP banyak AND efg tinggi THEN kelayakan
 layak

d. Decision Tree

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, classification_report
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier

data = data.drop(columns=["Player"])

X = data.drop(columns=["Label"])
y = data["Label"]

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

dt_model = DecisionTreeClassifier(random_state=42)
dt_model.fit(X_train, y_train)

y_pred = dt_model.predict(X_test)
print("Akurasi:", accuracy_score(y_test, y_pred))
print("Presisi:", precision_score(y_test, y_pred))
print("Recall:", recall_score(y_test, y_pred))
print("F1:", f1_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

Pada Model Decision Tree, yang akan dilakukan adalah menghapus dan memisahkan kolom yang tidak digunakan seperti kolom 'Player' yang berisi nama pemain NBA yang ada dalam data set, dimana kolom tidak tersebut tidak dapat memengaruhi prediksi yang akan dilakukan. Setelah itu, akan dilakukan pemisahan atau pembagian jumlah data antara data yang ingin ditraining dengan data yang akan dites nantinya, untuk perbandingannya adalah 80% data akan digunakan untuk melakukan training dan sisanya akan diterapkan pada saat testing untuk melihat hasil akhirnya.

C. Hasil dan Analisis

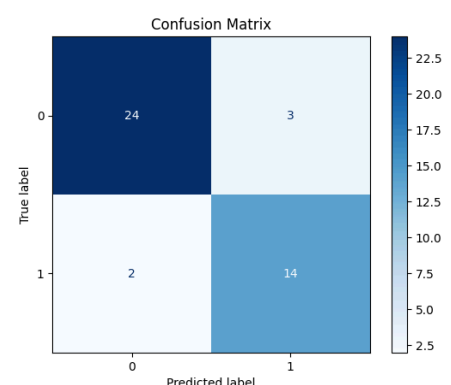
1. Decision Tree

```
Akurasi: 0.8837209302325582
Presisi: 0.8235294117647058
Recall: 0.875
F1: 0.8484848484848485

Classification Report:
              precision    recall  f1-score   support

     0       0.92         0.89         0.91         27
     1       0.82         0.88         0.85         16

   accuracy          0.88         0.88         0.88         43
  macro avg          0.87         0.88         0.88         43
 weighted avg          0.89         0.88         0.88         43
```



Kelas 0 (Label: Bukan Penembak 3 Poin)

- Precision: 0.92
Artinya: Dari semua yang diprediksi sebagai tidak layak, 92% benar-benar tidak layak. Model sangat jarang salah mengklasifikasikan pemain layak sebagai tidak layak.
- Recall: 0.89
Artinya: Dari semua pemain yang sebenarnya tidak layak, model berhasil menangkap 89% di antaranya. Masih ada sedikit yang lolos (false negatives), tapi sebagian besar tertangkap.
- F1-score: 0.91
Artinya: Gabungan harmonis precision dan recall kelas 0 cukup tinggi, menunjukkan keseimbangan performa.
- Support: 27
Ada 27 pemain dengan label sebenarnya = 0 (tidak layak).

Kelas 1 (Label: Penembak 3 Poin)

- Precision: 0.82
Dari semua yang diprediksi sebagai layak, 82% memang benar-benar layak. Masih ada kesalahan model yang salah menandai pemain tidak layak sebagai layak (false positives).
- Recall: 0.88
Dari semua pemain yang sebenarnya layak, model mampu mendeteksi 88% di antaranya.
- F1-score: 0.85
Kombinasi precision dan recall cukup solid, menunjukkan model cukup akurat dalam menangani pemain yang layak.
- Support: 16
Hanya ada 16 pemain yang benar-benar layak dalam dataset ini.

Keseluruhan Model

Accuracy: 0.88

Secara keseluruhan, 88% dari semua prediksi yang dilakukan oleh model adalah benar.

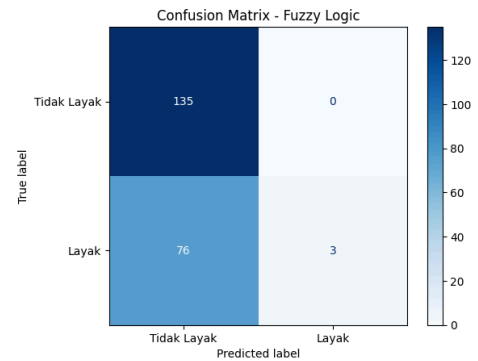
Model Decision Tree yang dibangun menunjukkan performa yang baik, dengan akurasi keseluruhan sebesar 88%. Metrik precision, recall, dan F1-score juga relatif tinggi untuk kedua kelas, menandakan bahwa model mampu mengenali pemain layak dan tidak layak dengan cukup andal.

Kelas 0 (Label: Bukan Penembak 3 Poin)

- Precision: 0.64

2. Fuzzy Logic

	precision	recall	f1-score	support
0	0.64	1.00	0.78	135
1	1.00	0.04	0.07	79
accuracy			0.64	214
macro avg	0.82	0.52	0.43	214
weighted avg	0.77	0.64	0.52	214
Akurasi Fuzzy Mamdani: 64.49 %				
Presisi Fuzzy Mamdani: 100.0 %				
Recall Score Fuzzy Mamdani: 3.8 %				
F1 Score Fuzzy Mamdani: 7.32 %				



Dari semua pemain yang diprediksi tidak layak, 64% benar-benar tidak layak. Masih terdapat 36% pemain yang sebenarnya layak namun diklasifikasikan sebagai tidak layak (false positives).

- Recall: 1.00
Model mendeteksi semua pemain yang sebenarnya tidak layak dengan tepat. Tidak ada pemain tidak layak yang lolos (false negatives = 0).
- F1-Score: 0.78
Kombinasi precision dan recall menunjukkan bahwa meskipun model agak banyak salah klasifikasi terhadap kelas 1, performa pada kelas 0 sangat baik.
- Support: 135
Jumlah data sebenarnya untuk kelas 0 sebanyak 135 pemain.

Kelas 1 (Label: Penembak 3 Poin)

- Precision: 1.00
Semua pemain yang diprediksi layak, memang benar-benar layak. Tidak ada false positive (tidak layak yang salah diklasifikasikan sebagai layak).
- Recall: 0.04
Hanya 4% dari pemain yang sebenarnya layak berhasil dideteksi oleh model. Sebanyak 96% pemain layak tidak dikenali dan salah diklasifikasikan sebagai tidak layak (false negatives tinggi).
- F1-Score: 0.07 Sangat rendah karena ketidakseimbangan antara recall dan precision. Meskipun precision sempurna, recall yang sangat rendah menurunkan nilai F1.
- Support: 79
Jumlah data sebenarnya untuk kelas 1 sebanyak 79 pemain.

Akurasi Keseluruhan

Accuracy: 0.64 menandakan Model berhasil mengklasifikasikan 64% dari total 214 data dengan benar.

D. Kesimpulan

Metrik	Decision Tree	Fuzzy Logic
Accuracy	88%	64%
Precision 0/Tidak Layak	0.92	0.64
Recall 0/Tidak Layak	0.89	1.00
F1 0/Tidak Layak	0.91	0.78
Precision 1/Layak	0.82	1.00
Recall 1/Layak	0.88	0.04
F1 1/Layak	0.85	0.07

Secara perbandingan, Decision Tree unggul secara keseluruhan dimana, Model Decision Tree menunjukkan performa yang konsisten tinggi di semua metrik, termasuk pada kedua kelas labelingnya(layak dan tidak layak). Selain itu Fuzzy Logic hanya unggul pada Recall kelas 0. Fuzzy Logic mampu mengenali semua pemain tidak layak dengan benar, yang berarti tidak ada false negative untuk kelas ini. Namun, ia gagal mendeteksi sebagian besar pemain layak, terlihat dari recall kelas 1 yang hanya 0.04. Terdapat Ketimpangan sensitivitas pada Fuzzy Logic dilihat dari Precision untuk kelas layak sangat tinggi (1.00), namun terlalu konservatif (hanya menandai pemain sebagai layak jika sangat yakin). Hal ini menghasilkan banyak false negatives, yang menurunkan F1-score kelas 1 secara drastis (hanya 0.07). Hal-hal tersebut bisa terjadi karena Decision Tree lebih akurat karena dia data-driven, otomatis, dan mengoptimalkan pemisahan kelas. Fuzzy logic lebih bersifat penalaran manusia dan tidak mengoptimalkan aturan dari data secara langsung.