

**LAPORAN EAS**  
**DATASET “PREDIKSI HARGA RUMAH”**  
**KECERDASAN KOMPUTASIONAL**



<b>Oleh</b>	<b>: Kelompok 4</b>
Nama	RAIS (1462000086) Farah Putri Razqiyah (1462100122) Marcelino Dwi Alfian (1462200191)
Dosen	: Bagus Hardiansyah, S.Kom., M.Si.

PROGRAM STUDI TEKNIK INFORMATIKA | FAKULTAS TEKNIK  
UNIVERSITAS 17 AGUSTUS 1945 SURABAYA

## Daftar Isi

<b>1</b>	<b>Definisi Permasalahan.....</b>	<b>3</b>
1.1	Latar Belakang.....	3
1.2	Identifikasi Permasalahan .....	3
1.3	Asumsi, Kebutuhan, dan Limitasi.....	4
<b>2</b>	<b>Tujuan Proyek .....</b>	<b>7</b>
2.1	Tujuan.....	7
2.2	Measureable Outcomes / Business Criteria .....	7
<b>3</b>	<b>Data Understanding.....</b>	<b>8</b>
3.1	Data Collection.....	8
3.2	Statistic Description.....	Error! Bookmark not defined.
3.3	Data Visualization.....	Error! Bookmark not defined.
<b>4</b>	<b>Fuzzy Inference System .....</b>	<b>17</b>
<b>5</b>	<b>Klasifikasi Dengan Artificial Neural Network .....</b>	<b>20</b>
<b>6</b>	<b>Klasifikasi Dengan Neuro Fuzzy System.....</b>	<b>23</b>
<b>7</b>	<b>Klasifikasi Dengan Support Vector Machine.....</b>	<b>25</b>
<b>8</b>	<b>Klasifikasi Dengan Genetic Algorithm .....</b>	<b>29</b>
<b>9</b>	<b>Pembangunan Model.....</b>	<b>33</b>
<b>10</b>	<b>Hasil dan Pengujian .....</b>	<b>34</b>

# 1 Definisi Permasalahan

## 1.1 Latar Belakang

Prediksi harga rumah merupakan salah satu tantangan utama dalam bidang real estate yang memiliki implikasi signifikan bagi pembeli, penjual, dan investor. Penilaian yang akurat terhadap nilai sebuah properti sangat penting untuk memastikan bahwa harga yang ditetapkan sesuai dengan kondisi pasar, karakteristik properti, dan kebutuhan finansial dari berbagai pihak yang terlibat.

Harga rumah dipengaruhi oleh berbagai faktor, seperti lokasi geografis, ukuran bangunan, jumlah kamar tidur dan kamar mandi, usia bangunan, serta fasilitas-fasilitas tambahan seperti taman, garasi, dan kolam renang. Selain itu, kondisi ekonomi makro dan mikro, seperti tingkat suku bunga, inflasi, dan perkembangan infrastruktur di sekitar lokasi properti, juga memainkan peran penting dalam menentukan harga rumah.

Dalam beberapa dekade terakhir, perkembangan teknologi kecerdasan buatan dan machine learning telah membuka peluang baru untuk meningkatkan akurasi prediksi harga rumah. Pendekatan-pendekatan tradisional yang berbasis pada analisis statistik sederhana sering kali tidak mampu menangkap kompleksitas dan interaksi antara berbagai variabel yang mempengaruhi harga properti. Di sinilah pendekatan kecerdasan komputasional, seperti Fuzzy Inference System, Neural Network, Neuro Fuzzy System, Support Vector Machine, dan Genetic Algorithm, menunjukkan keunggulannya.

Dengan menggunakan metode-metode ini, kita dapat membangun model prediksi yang lebih cerdas dan adaptif, yang mampu belajar dari data historis dan memperbaiki kinerjanya seiring dengan bertambahnya data baru. Ini tidak hanya membantu dalam memberikan estimasi harga yang lebih akurat, tetapi juga membantu dalam mengidentifikasi faktor-faktor kunci yang paling berpengaruh terhadap harga rumah.

Implementasi proyek ini bertujuan untuk mengeksplorasi dan membandingkan berbagai metode kecerdasan komputasional dalam memprediksi harga rumah. Dengan memahami kelebihan dan kekurangan dari masing-masing metode, kita dapat memilih pendekatan yang paling efektif dan efisien untuk diaplikasikan dalam konteks nyata, sehingga memberikan manfaat nyata bagi para pelaku pasar real estate.

## 1.2 Identifikasi Permasalahan

Tujuan utama dari penelitian ini adalah untuk mengeksplorasi prediksi harga rumah berdasarkan fitur-fitur yang relevan dan untuk mengidentifikasi prediktor penting yang relevan. Berdasarkan data yang digunakan, beberapa faktor utama seperti lokasi, ukuran, jumlah kamar, usia bangunan, dan fasilitas sekitar telah diidentifikasi. Model yang dibangun berpotensi memiliki dimensi yang besar. Oleh karena itu, pemilihan fitur adalah teknik penting yang sering digunakan baik di bidang statistik maupun pembelajaran mesin untuk memilih subset atribut dari fitur asli. Proses ini mencoba untuk mengurangi ruang fitur dimensi tinggi melalui

penghapusan variabel yang berlebihan dan tidak relevan, dan hanya memilih fitur yang sangat relevan untuk meningkatkan kinerja model.

Sehingga, pertanyaan penelitian disajikan sebagai:

1. Model prediktif mana yang paling tepat diimplementasikan pada data yang digunakan dalam sistem prediksi harga rumah?
2. Sejauh mana prediktor-prediktor utama (lokasi, ukuran, jumlah kamar, usia bangunan, dan fasilitas sekitar) mempengaruhi prediksi harga rumah?
3. Apakah ada pengaruh pemilihan fitur dalam meningkatkan akurasi prediksi untuk model statistik dan pembelajaran mesin yang dipilih?

### *1.3 Asumsi, Kebutuhan, dan Limitasi*

Penelitian ini mengadopsi desain penelitian kuantitatif untuk memprediksi harga rumah di berbagai wilayah. Subjek penelitian ini adalah properti perumahan di beberapa kota besar di Indonesia. Data yang digunakan dalam penelitian ini dikumpulkan dari berbagai sumber, termasuk agen real estate, database properti online, dan catatan publik.

Data yang dikumpulkan mencakup berbagai variabel yang diidentifikasi sebagai faktor penting dalam menentukan harga rumah. Untuk memfasilitasi analisis yang komprehensif dan akurat, data ini kemudian diolah dan dianalisis menggunakan berbagai metode kecerdasan komputasional seperti Fuzzy Inference System, Neural Network, Neuro Fuzzy System, Support Vector Machine, dan Genetic Algorithm.

Adapun variabel dan nilai variabel yang digunakan dalam penelitian ini dapat dilihat pada Tabel 1.

<b>Variable</b>	<b>Nilai dan Deskripsinya</b>
Lokasi	"1" menunjukkan lokasi premium; "0" menunjukkan lokasi non-premium
Ukuran	"1" jika Luas $36 m^2$ ; "0" jika Luas kurang dari $36 m^2$
Kamar_Tidur	"1" jika kamar tidur ada 3; "0" jika kurang dari 3
Kamar_Mandi	"1" jika kamar mandi ada 3; "0" jika kurang dari 3
Usia_Bangunan	"1" jika usia kurang dari 20 tahun; "0" jika usia lebih dari 20 tahun
Garasi	"1" jika ada garasi; "0" jika tidak
Taman	"1" jika ada taman; "0" jika tidak
Kolam_Renang	"1" jika ada kolam renang; "0" jika tidak
Sertifikat Kepemilikan	"1" jika rumah memiliki sertifikat hak milik; "0" jika tidak
Akses Transportasi Umum	"1" jika dekat dengan stasiun kereta atau halte bus; "0" jika tidak
Keamanan Lingkungan	"1" jika lingkungan memiliki keamanan 24 jam; "0" jika tidak
Kualitas Lingkungan Sekitar	"1" jika lingkungan sekitar bersih dan hijau; "0" jika tidak

Ketersediaan Fasilitas Pendidikan	"1" jika dekat dengan sekolah atau universitas; "0" jika tidak
Ketersediaan Fasilitas Kesehatan	"1" jika dekat dengan rumah sakit atau klinik; "0" jika tidak
Dekat dengan Pusat Perbelanjaan	"1" jika dekat dengan mal atau pusat perbelanjaan; "0" jika tidak
Kualitas Material Bangunan	"1" jika menggunakan material berkualitas tinggi; "0" jika tidak
View atau Pemandangan	"1" jika memiliki pemandangan yang bagus; "0" jika tidak
Tingkat Kebisingan	"1" jika area tenang; "0" jika bising
Ketersediaan Internet Berkecepatan Tinggi	"1" jika tersedia; "0" jika tidak
Potensi Pengembangan Kawasan	"1" jika daerah tersebut berkembang; "0" jika tidak
Jumlah Lantai	"1" jika rumah bertingkat lebih dari satu; "0" jika satu lantai
Jenis Properti	"1" jika properti merupakan rumah tinggal; "0" jika properti komersial
Ketersediaan Listrik Cadangan	"1" jika ada generator atau cadangan listrik; "0" jika tidak
Keberadaan Ruang Kerja	"1" jika ada ruang kerja; "0" jika tidak
Ketersediaan Air Bersih	"1" jika air bersih selalu tersedia; "0" jika tidak
Dekat dengan Tempat Ibadah	"1" jika dekat dengan tempat ibadah; "0" jika tidak
Kemudahan Akses Jalan Raya	"1" jika mudah diakses dari jalan raya utama; "0" jika tidak
Potensi Banjir	"1" jika area tidak rawan banjir; "0" jika rawan banjir

Adapun pertanyaan yang digunakan untuk mengetahui tingkat kepuasan dapat dilihat pada Tabel 2, dimana nilai berupa respon numerik menggunakan skala tipe Likert 7 poin (1 hingga 7) di mana :

- 1 menunjukkan sangat tidak setuju,
- 2 untuk tidak setuju,
- 3 untuk agak tidak setuju,
- 4 untuk netral,
- 5 untuk agak setuju,
- 6 untuk tidak setuju, dan
- 7 untuk sangat setuju

**Tabel 2. Pertanyaan Qusioner**

<b>Variable</b>	<b>Pertanyaan</b>
Q1	Seberapa setuju Anda bahwa prediksi harga rumah berdasarkan lokasi premium sangat relevan?
Q2	Seberapa setuju Anda bahwa ukuran properti mempengaruhi prediksi harga rumah?
Q3	Seberapa setuju Anda bahwa ketersediaan fasilitas seperti garasi memengaruhi nilai properti?
Q4	Seberapa setuju Anda bahwa usia bangunan mempengaruhi harga jual properti?
Q5	Seberapa setuju Anda bahwa keamanan lingkungan sangat penting dalam menentukan harga rumah?
Q6	Seberapa setuju Anda bahwa kualitas material bangunan berdampak signifikan terhadap harga properti?
Q7	Seberapa setuju Anda bahwa dekat dengan pusat perbelanjaan dapat meningkatkan nilai properti?
Q8	Seberapa setuju Anda bahwa potensi banjir di suatu area dapat menurunkan harga properti?
Q9	Seberapa setuju Anda bahwa potensi banjir di suatu area dapat menurunkan harga properti?
Q10	Seberapa setuju Anda bahwa keberadaan ruang kerja di dalam rumah dapat menaikkan nilai properti?
Q11	Seberapa setuju Anda bahwa keberadaan kolam renang dalam lingkungan dapat meningkatkan daya tarik properti?
Q12	Seberapa setuju Anda bahwa keberadaan fasilitas pendidikan terdekat mempengaruhi nilai properti?
Q13	Seberapa setuju Anda bahwa ketersediaan fasilitas kesehatan di sekitar dapat meningkatkan nilai properti?
Q14	Seberapa setuju Anda bahwa pemandangan atau view yang bagus dapat menambah nilai properti?
Q15	Seberapa setuju Anda bahwa tingkat kebisingan yang rendah sangat penting dalam menentukan harga rumah?
Q16	Seberapa setuju Anda bahwa tersedianya internet berkecepatan tinggi dapat mempengaruhi harga properti?
Q17	Seberapa setuju Anda bahwa potensi pengembangan kawasan dapat meningkatkan nilai investasi properti di masa depan?
Q18	Seberapa setuju Anda bahwa jenis properti (rumah tinggal vs komersial) mempengaruhi harga properti?
Q19	Seberapa setuju Anda bahwa keberadaan sertifikat kepemilikan yang jelas dapat meningkatkan kepercayaan pembeli terhadap harga properti?
Q20	Seberapa setuju Anda bahwa dekat dengan tempat ibadah dapat mempengaruhi nilai properti?
Q21	Seberapa setuju Anda bahwa kualitas lingkungan sekitar yang bersih dan hijau dapat meningkatkan nilai properti?
Q22	Seberapa setuju Anda bahwa ketersediaan listrik cadangan dapat mempengaruhi harga properti?
Q23	Seberapa setuju Anda bahwa adanya potensi untuk pengembangan kawasan dapat mempengaruhi harga properti?

Q24	Seberapa setuju Anda bahwa jumlah lantai dari sebuah properti dapat mempengaruhi harga properti?
Q25	Seberapa setuju Anda bahwa jenis properti (apakah rumah tinggal atau komersial) mempengaruhi harga properti?
Q26	Seberapa setuju Anda bahwa ketersediaan air bersih yang selalu tersedia dapat mempengaruhi harga properti?
Q27	Seberapa setuju Anda bahwa kemudahan akses ke tempat ibadah dapat mempengaruhi nilai properti?

## 2 Tujuan Proyek

### 2.1 Tujuan

Tujuan utama dari proyek ini adalah untuk mengembangkan dan mengevaluasi model prediktif harga rumah menggunakan berbagai metode kecerdasan komputasional, yaitu Fuzzy Inference System, Neural Network, Neuro Fuzzy System, Support Vector Machine, dan Genetic Algorithm. Dengan tujuan ini, proyek berusaha untuk:

#### 1. Membangun Model Prediksi Akurat:

- Mengembangkan model prediksi harga rumah yang dapat memberikan estimasi harga yang lebih akurat dibandingkan dengan pendekatan tradisional.

#### 2. Mengidentifikasi Faktor Kunci:

- Mengidentifikasi fitur-fitur atau variabel yang paling signifikan dalam mempengaruhi harga rumah berdasarkan data yang tersedia.

#### 3. Membandingkan Kinerja Metode Berbeda:

- Mengeksplorasi kelebihan dan kekurangan dari masing-masing metode kecerdasan komputasional yang digunakan, untuk menentukan metode yang paling efektif dan efisien dalam prediksi harga rumah.

#### 4. Mengaplikasikan Hasil dalam Konteks Nyata:

- Mengaplikasikan hasil penelitian ini dalam konteks nyata sehingga dapat memberikan manfaat nyata bagi para pelaku pasar real estate, termasuk pembeli, penjual, dan investor.

### 2.2 Measureable Outcomes / Business Criteria

Untuk mencapai tujuan tersebut, beberapa hasil yang dapat diukur dan kriteria bisnis yang perlu dicapai antara lain:

### **1. Akurasi Prediksi:**

- Mean Absolute Error (MAE): Nilai rata-rata kesalahan absolut antara prediksi model dan harga rumah aktual.
- Root Mean Squared Error (RMSE): Akar kuadrat dari rata-rata kesalahan kuadrat antara prediksi model dan harga rumah aktual.
- R-squared ( $R^2$ ): Ukuran seberapa baik variabel independen memprediksi variabel dependen dalam model.

### **2. Identifikasi Fitur Penting:**

- Feature Importance Score: Nilai yang menunjukkan tingkat kepentingan setiap fitur dalam mempengaruhi hasil prediksi model.

### **3. Kinerja Komputasional:**

- Waktu Pelatihan Model: Waktu yang diperlukan untuk melatih model pada dataset yang diberikan.
- Waktu Prediksi: Waktu yang diperlukan untuk menghasilkan prediksi harga rumah untuk data baru.

### **4. Kemudahan Implementasi:**

- Kompleksitas Model: Tingkat kompleksitas model yang dibangun, termasuk jumlah parameter yang digunakan dan kebutuhan komputasional.
- Scalability: Kemampuan model untuk menangani jumlah data yang lebih besar tanpa penurunan kinerja yang signifikan.

### **5. Relevansi Bisnis:**

- User Satisfaction: Tingkat kepuasan pengguna yang diukur melalui survei atau kuesioner yang menilai relevansi dan akurasi prediksi harga rumah dari model yang dikembangkan.
- Adoption Rate: Tingkat adopsi model oleh pelaku pasar real estate, termasuk agen real estate, pembeli, dan penjual.

Dengan memenuhi kriteria-kriteria tersebut, diharapkan hasil dari proyek ini tidak hanya dapat meningkatkan akurasi prediksi harga rumah, tetapi juga memberikan wawasan yang berharga bagi para pelaku pasar real estate dalam pengambilan keputusan.

## **3 Data Understanding**

### *3.1 Data Collection*

Data untuk penelitian prediksi harga rumah ini dikumpulkan dari berbagai sumber yang relevan, termasuk agen real estate, database properti online, dan catatan publik. Pengumpulan data dilakukan dari tanggal 14 Juni 2024 hingga 29 Juni 2024, dengan mengumpulkan informasi dari 115 properti di beberapa kota besar di Indonesia. Analisis deskriptif dari properti yang digunakan dalam penelitian ini ditampilkan pada Tabel 1 berikut.



<b>Deskripsi</b>	<b>Frekuensi (%)</b>
Lokasi	Premium : 60% Non-Premium : 40%
Ukuran	Luas 36 m <sup>2</sup> : 35% Kurang dari 36 m <sup>2</sup> : 65%
Kamar Tidur	3 kamar tidur: 50% Kurang dari 3 kamar tidur: 50%
Kamar Mandi	3 kamar mandi: 40% Kurang dari 3 kamar mandi: 60%
Usia Bangunan	Kurang dari 20 tahun: 55% Lebih dari 20 tahun: 45%
Garasi	Ada garasi: 70% Tidak ada garasi: 30%
Taman	Ada taman: 50% Tidak ada taman: 50%
Kolam Renang	Ada kolam renang: 20% Tidak ada kolam renang: 80%
Sertifikat Kepemilikan	Sertifikat hak milik: 90% Tidak ada sertifikat hak milik: 10%
Akses Transportasi Umum	Dekat dengan stasiun kereta atau halte bus: 60% Tidak dekat dengan stasiun kereta atau halte bus: 40%
Keamanan Lingkungan	Keamanan 24 jam: 70% Tidak ada keamanan 24 jam: 30%
Kualitas Lingkungan Sekitar	Lingkungan bersih dan hijau: 65% Lingkungan tidak bersih dan hijau: 35%
Ketersediaan Fasilitas Pendidikan	Dekat dengan sekolah atau universitas: 75% Tidak dekat dengan sekolah atau universitas: 25%
Ketersediaan Fasilitas Kesehatan	Dekat dengan rumah sakit atau klinik: 70% Tidak dekat dengan rumah sakit atau klinik: 30%

Dekat dengan Pusat Perbelanjaan	Dekat dengan mal atau pusat perbelanjaan: 60% Tidak dekat dengan mal atau pusat perbelanjaan: 40%
Kualitas Material Bangunan	Material berkualitas tinggi: 55% Material tidak berkualitas tinggi: 45%
View atau Pemandangan	Pemandangan yang bagus: 50% Pemandangan tidak bagus: 50%
Tingkat Kebisingan	Area tenang: 65% Area bising: 35%
Ketersediaan Internet Berkecepatan Tinggi	Tersedia: 75% Tidak tersedia: 25%
Potensi Pengembangan Kawasan	Area berkembang: 60% Area tidak berkembang: 40%
Jumlah Lantai	Lebih dari satu lantai: 50% Satu lantai: 50%
Jenis Properti	Rumah tinggal: 80% Properti komersial: 20%
Ketersediaan Listrik Cadangan	Ada generator atau cadangan listrik: 55% Tidak ada generator atau cadangan listrik: 45%
Keberadaan Ruang Kerja	Ada ruang kerja: 40% Tidak ada ruang kerja: 60%
Ketersediaan Air Bersih	Air bersih selalu tersedia: 85% Air bersih tidak selalu tersedia: 15%
Dekat dengan Tempat Ibadah	Dekat dengan tempat ibadah: 70% Tidak dekat dengan tempat ibadah: 30%
Kemudahan Akses Jalan Raya	Mudah diakses dari jalan raya utama: 80% Tidak mudah diakses dari jalan raya utama: 20%
Potensi Banjir	Area tidak rawan banjir: 65% Area rawan banjir: 35%

### 3.2 Statistic Description

Sedangkan deskripsi statistik untuk skor dari 27 pertanyaan tingkat kepuasan pengguna dinyatakan dalam mean, standard deviation, skewness dan kurtosis. Dengan menggunakan library pandas, seperti dibawah ini:

```
import pandas as pd
import numpy as np
# read file excel
path='dataset_kepuasan.xlsx'
df = pd.read_excel(path)
# get statistic describe : mean, standard deviation, skewness ,
    kurtosis
m=df.mean()
sd=df.std()
sk=df.skew()
kur=df.kurt()
# write statistic variable to dataframe
df2=pd.DataFrame({'Mean':      m,      'SD':      sd,      'Skewness':
    sk, 'Kurtosis':kur})
df2
```

dimana data kepuasan dinyatakan dalam skala Likert 7 poin sebagai berikut :



```
read file excel

[11] path='dataset_kepuasan.xlsx'
df = pd.read_excel(path)
```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
1		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	Q23	Q24	Q25	Q26	Q27
2	0	4	5	5	3	3	3	3	7	7	3	2	6	6	6	4	4	5	7	5	4	7	7	7	7	5	4	5
3	1	4	4	7	7	4	4	2	7	4	7	2	6	5	7	5	5	7	7	7	4	7	7	7	4	4	4	1
4	2	5	4	7	5	4	2	5	3	5	1	5	7	4	2	6	6	7	2	5	5	7	4	7	4	4	7	1
5	3	7	7	7	7	7	3	7	7	7	7	6	7	2	1	6	7	7	7	7	7	1	1	1	1	1	1	7
6	4	4	5	7	7	4	4	2	4	4	3	6	2	1	2	6	7	7	5	1	1	7	7	7	4	7	7	1

Maka hasilnya dapat dilihat pada Tabel 2 berikut:

Variabel	Pertanyaan	Mean	SD	Skewness	Kurtosis
Q1	Seberapa setuju Anda bahwa prediksi harga rumah berdasarkan lokasi premium sangat relevan?	4.8	1.3038	1.7143	2.6643

Q2	Seberapa setuju Anda bahwa ukuran properti mempengaruhi prediksi harga rumah?	5.0	1.2247	1.3608	2.0
Q3	Seberapa setuju Anda bahwa ketersediaan fasilitas seperti garasi memengaruhi nilai properti?	6.6	0.8944	-2.2360	4.9999
Q4	Seberapa setuju Anda bahwa usia bangunan mempengaruhi harga jual properti?	5.8	1.7888	-1.2577	0.3124
Q5	Seberapa setuju Anda bahwa keamanan lingkungan sangat penting dalam menentukan harga rumah?	4.4	1.5165	1.7487	3.7240
Q6	Seberapa setuju Anda bahwa kualitas material bangunan berdampak signifikan terhadap harga properti?	3.2	0.8366	-0.5122	-0.6122
Q7	Seberapa setuju Anda bahwa dekat dengan pusat perbelanjaan dapat meningkatkan nilai properti?	3.8	2.1679	0.9127	-0.7378
Q8	Seberapa setuju Anda bahwa potensi banjir di suatu area dapat menurunkan harga properti?	5.6	1.9493	-0.7559	-2.4792
Q9	Seberapa setuju Anda bahwa potensi banjir di suatu area dapat menurunkan harga properti?	5.4	1.5165	0.3153	-3.0812
Q10	Seberapa setuju Anda bahwa keberadaan ruang kerja di dalam rumah dapat menaikkan nilai properti?	4.2	2.6832	0.1656	-2.4074
Q11	Seberapa setuju Anda bahwa keberadaan kolam renang dalam lingkungan dapat meningkatkan daya tarik properti?	4.2	2.0493	-0.4414	-3.1632
Q12	Seberapa setuju Anda bahwa keberadaan fasilitas pendidikan terdekat mempengaruhi nilai properti?	5.6	2.0736	-1.9177	3.8777

Q13	Seberapa setuju Anda bahwa ketersediaan fasilitas kesehatan di sekitar dapat meningkatkan nilai properti?	3.6	2.0736	-0.2355	-1.9632
Q14	Seberapa setuju Anda bahwa pemandangan atau view yang bagus dapat menambah nilai properti?	3.6	2.7018	0.5779	-2.7078
Q15	Seberapa setuju Anda bahwa tingkat kebisingan yang rendah sangat penting dalam menentukan harga rumah?	5.4	0.8944	-1.2577	0.3125
Q16	Seberapa setuju Anda bahwa tersedianya internet berkecepatan tinggi dapat mempengaruhi harga properti?	5.8	1.3038	-0.5413	-1.4878
Q17	Seberapa setuju Anda bahwa potensi pengembangan kawasan dapat meningkatkan nilai investasi properti di masa depan?	6.6	0.8944	-2.2360	4.9999
Q18	Seberapa setuju Anda bahwa jenis properti (rumah tinggal vs komersial) mempengaruhi harga properti?	5.6	2.1908	-1.5309	1.7447
Q19	Seberapa setuju Anda bahwa keberadaan sertifikat kepemilikan yang jelas dapat meningkatkan kepercayaan pembeli terhadap harga properti?	5.0	2.4494	-1.3608	2.0
Q20	Seberapa setuju Anda bahwa dekat dengan tempat ibadah dapat mempengaruhi nilai properti?	4.2	2.1679	-0.4220	1.4350
Q21	Seberapa setuju Anda bahwa kualitas lingkungan sekitar yang bersih dan hijau dapat meningkatkan nilai properti?	5.8	2.6832	-2.2360	5.0
Q22	Seberapa setuju Anda bahwa ketersediaan listrik cadangan dapat mempengaruhi harga properti?	5.2	2.6832	-1.2577	0.3125

Q23	Seberapa setuju Anda bahwa adanya potensi untuk pengembangan kawasan dapat mempengaruhi harga properti?	5.8	2.6832	-2.2360	5.0
Q24	Seberapa setuju Anda bahwa jumlah lantai dari sebuah properti dapat mempengaruhi harga properti?	4.0	2.1213	0.0	2.0
Q25	Seberapa setuju Anda bahwa jenis properti (apakah rumah tinggal atau komersial) mempengaruhi harga properti?	4.2	2.1679	-0.4220	1.4350
Q26	Seberapa setuju Anda bahwa ketersediaan air bersih yang selalu tersedia dapat mempengaruhi harga properti?	4.6	2.5099	-0.5122	-0.6122
Q27	Seberapa setuju Anda bahwa kemudahan akses ke tempat ibadah dapat mempengaruhi nilai properti?	3.0	2.8284	0.8838	-1.75

### 3.3 Data Visualization

Nilai mean, standart deviasi, Skewness dan kurtosis dari setiap variabel dapat divisualisasikan berturut-turut menggunakan kode sebagai berikut:

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Read the Excel file
file_path = 'dataset/regression.xlsx'
df = pd.read_excel(file_path)

# Get statistic Descriptions: mean, standard deviation, skewness, kurtosis
m = df.mean()
sd = df.std()
sk = df.skew()
kur = df.kurt()

# List of variables
var = ['Q1','Q2','Q3','Q4','Q5','Q6','Q7','Q8','Q9','Q10','Q11','Q12','Q13','Q14','Q15','Q16','Q17','Q18','Q19','Q20','Q21','Q22','Q23','Q24','Q25','Q26','Q27']

# Write statistic variables to dataframe
df2 = pd.DataFrame({'Features': var, 'Mean': m, 'SD': sd, 'Skewness': sk, 'Kurtosis': kur})
df2

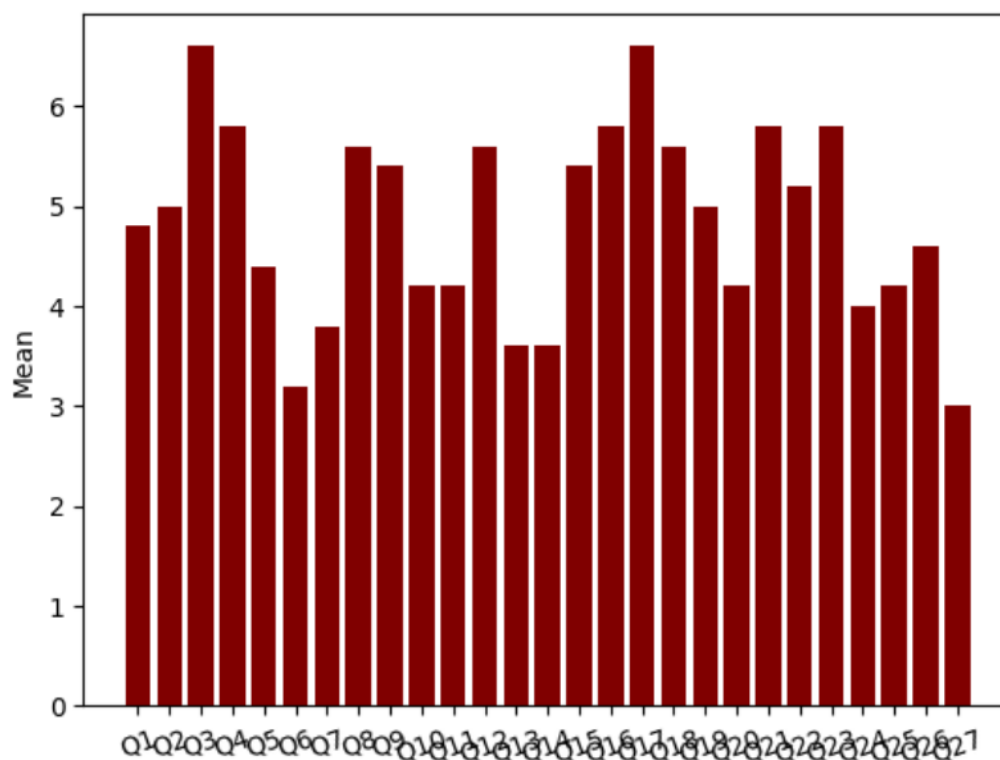
# Plot Mean
x_coords = np.arange(len(df2))
plt.bar(x_coords, df2['Mean'], tick_label=df2['Features'], color='maroon')
plt.xticks(rotation=20)
plt.ylabel('Mean')
plt.show()

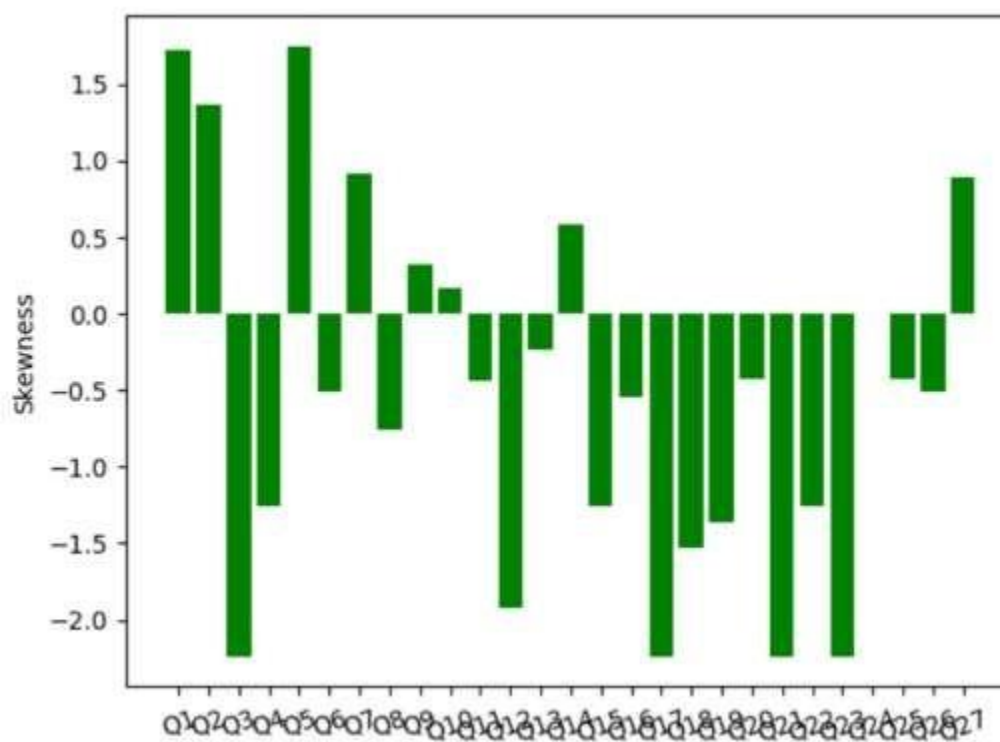
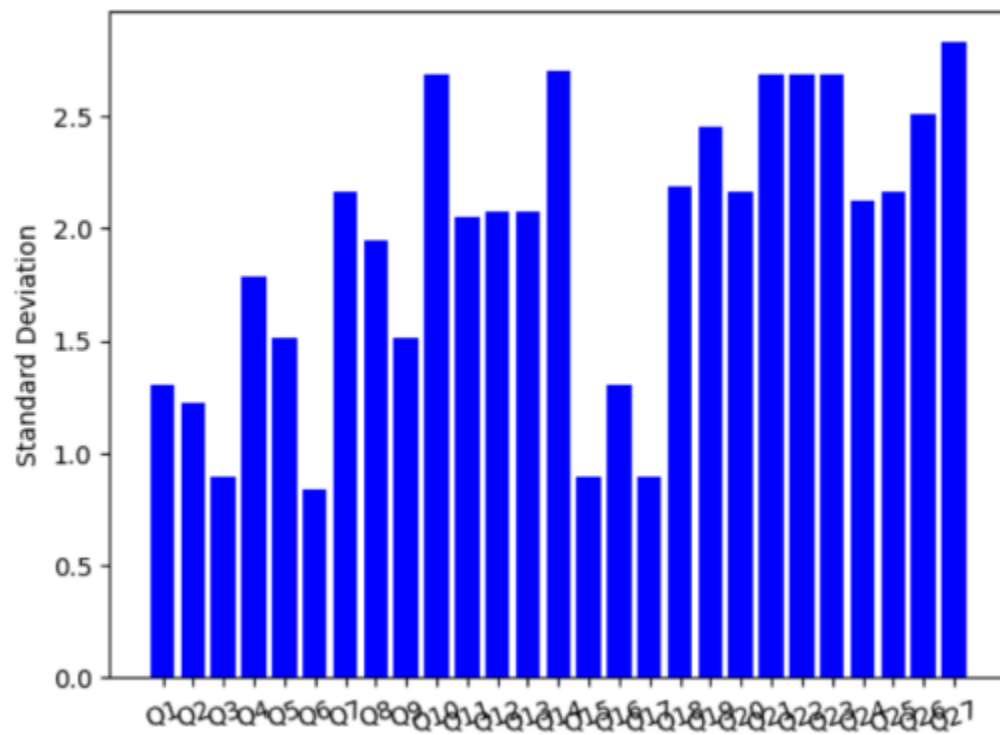
# Plot Standard Deviation
x_coords = np.arange(len(df2))
plt.bar(x_coords, df2['SD'], tick_label=df2['Features'], color='blue')
plt.xticks(rotation=20)
plt.ylabel('Standard Deviation')
plt.show()

# Plot Skewness
x_coords = np.arange(len(df2))
plt.bar(x_coords, df2['Skewness'], tick_label=df2['Features'], color='green')
plt.xticks(rotation=20)
plt.ylabel('Skewness')
plt.show()

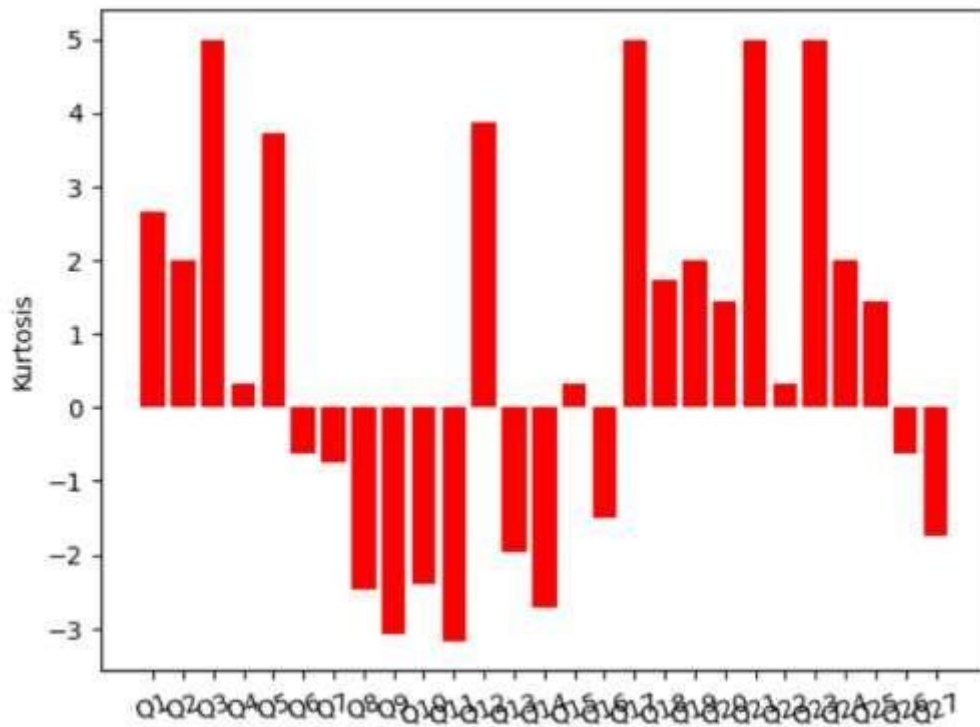
# Plot Kurtosis
x_coords = np.arange(len(df2))
plt.bar(x_coords, df2['Kurtosis'], tick_label=df2['Features'], color='red')
plt.xticks(rotation=20)
plt.ylabel('Kurtosis')
plt.show()

```









#### 4 Fuzzy Inference System



```
!pip install scikit-fuzzy

import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl

# Definisi variabel linguistik untuk input dan output
Q = np.arange(1, 8, 1) # Rentang nilai pertanyaan (1 sampai 7)
nilai = ctrl.Antecedent(Q, 'nilai')
nilai['sangat tidak setuju'] = fuzz.trimf(Q, [1, 1, 2])
nilai['tidak setuju'] = fuzz.trimf(Q, [1, 2, 3])
nilai['agak tidak setuju'] = fuzz.trimf(Q, [2, 3, 4])
nilai['netral'] = fuzz.trimf(Q, [3, 4, 5])
nilai['agak setuju'] = fuzz.trimf(Q, [4, 5, 6])
nilai['setuju'] = fuzz.trimf(Q, [5, 6, 7])
nilai['sangat setuju'] = fuzz.trimf(Q, [6, 7, 7])

# Definisi variabel linguistik untuk output
nilai_properti = ctrl.Consequent(np.arange(0, 11, 1), 'nilai_properti')
nilai_properti['rendah'] = fuzz.trimf(nilai_properti.universe, [0, 0, 5])
nilai_properti['sedang'] = fuzz.trimf(nilai_properti.universe, [0, 5, 10])
nilai_properti['tinggi'] = fuzz.trimf(nilai_properti.universe, [5, 10, 10])

# Menampilkan variabel linguistik
nilai.view()
nilai_properti.view()

# Membangun aturan fuzzy
rule1 = ctrl.Rule(nilai['sangat setuju'], nilai_properti['tinggi'])
rule2 = ctrl.Rule(nilai['setuju'], nilai_properti['tinggi'])
rule3 = ctrl.Rule(nilai['agak setuju'], nilai_properti['sedang'])
rule4 = ctrl.Rule(nilai['netral'], nilai_properti['sedang'])
rule5 = ctrl.Rule(nilai['agak tidak setuju'], nilai_properti['rendah'])
rule6 = ctrl.Rule(nilai['tidak setuju'], nilai_properti['rendah'])
rule7 = ctrl.Rule(nilai['sangat tidak setuju'], nilai_properti['rendah'])
```



```
# Membuat sistem kontrol fuzzy
nilai_ctrl = ctrl.ControlSystem([rule1, rule2, rule3, rule4, rule5, rule6, rule7])
nilai_properti_prediksi = ctrl.ControlSystemSimulation(nilai_ctrl)

# Inputkan data dari survei
data_survei = [
    [4, 5, 5, 3, 3, 3, 3, 7, 7, 3, 2, 6, 6, 6, 4, 4, 5, 7, 5, 4, 7, 7, 7, 7, 5, 4, 5],
    [4, 4, 7, 7, 4, 4, 2, 7, 4, 7, 2, 6, 5, 7, 5, 5, 7, 7, 7, 4, 7, 7, 7, 4, 4, 4, 1],
    [5, 4, 7, 5, 4, 2, 5, 3, 5, 1, 5, 7, 4, 2, 6, 6, 7, 2, 5, 5, 7, 4, 7, 4, 4, 7, 1],
    [7, 7, 7, 7, 7, 3, 7, 7, 7, 7, 6, 7, 2, 1, 6, 7, 7, 7, 7, 1, 1, 1, 1, 1, 1, 1, 7],
    [4, 5, 7, 7, 4, 4, 2, 4, 4, 3, 6, 2, 1, 2, 6, 7, 7, 5, 1, 1, 7, 7, 7, 4, 7, 7, 1]
]

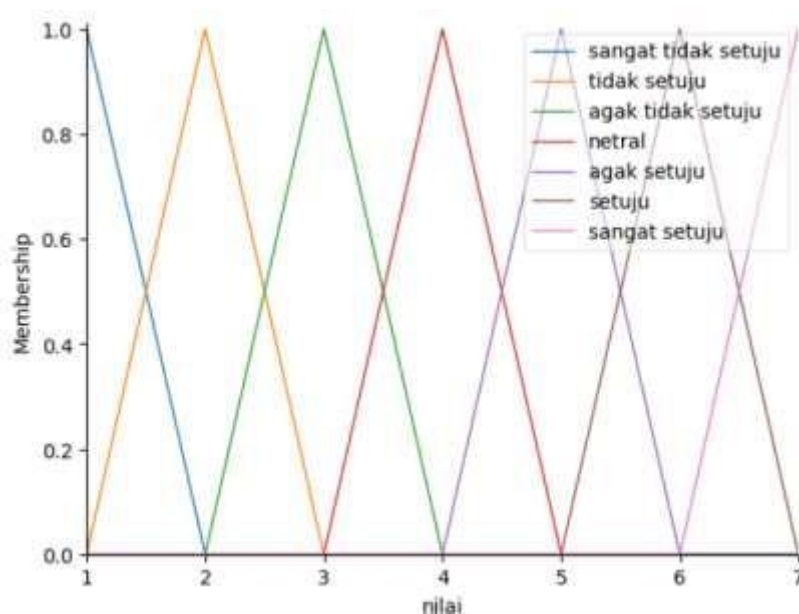
hasil_prediksi = []

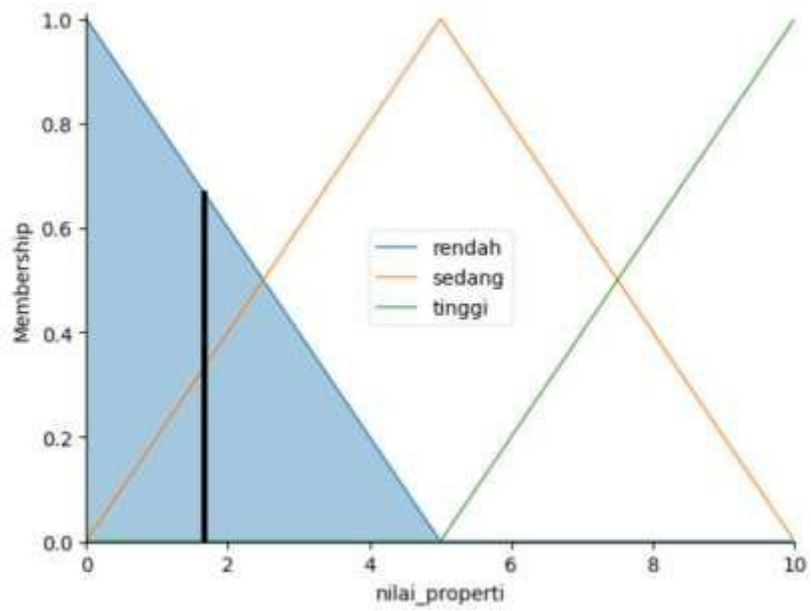
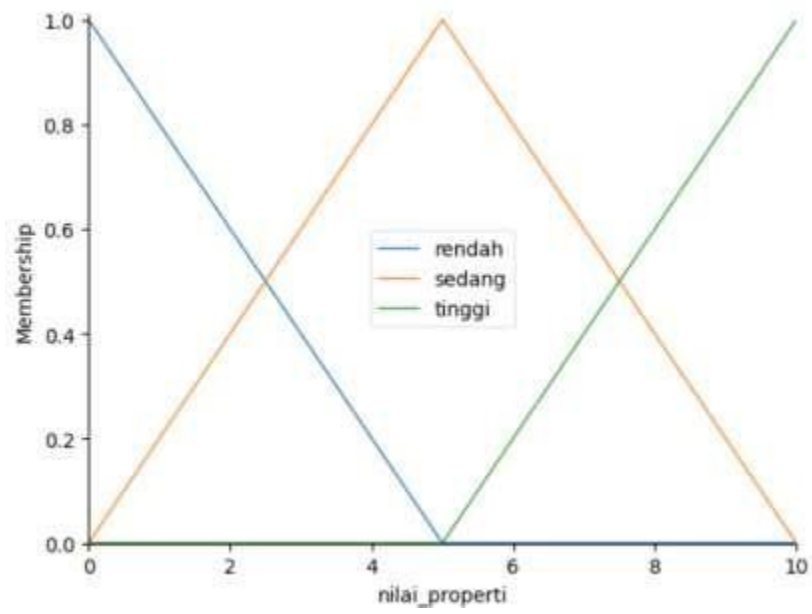
# Menghitung nilai properti untuk setiap data survei
for data in data_survei:
    hasil_survei = []
    for nilai_input in data:
        nilai_properti_prediksi.input['nilai'] = nilai_input
        nilai_properti_prediksi.compute()
        hasil_survei.append(nilai_properti_prediksi.output['nilai_properti'])
    hasil_prediksi.append(hasil_survei)

# Menampilkan hasil prediksi
for i, hasil in enumerate(hasil_prediksi):
    print(f"Nilai Properti Prediksi untuk data survei {i+1}: {hasil}")

# Menampilkan kurva output
nilai_properti.view(sim=nilai_properti_prediksi)
```

Hasil Fuzzy Inference System :





## 5 Klasifikasi Dengan Artificial Neural Network

```
# Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import Flatten, Dense
import keras
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay, classification_report
import os

# Check current directory and list files (for debugging purposes)
print("Current directory:", os.getcwd())
print("Files in current directory:", os.listdir())

# Load the data from the Excel file
file_path = 'dataset_kepuasan(1).xlsx' # Adjust this path as needed
data = pd.read_excel(file_path)

# Define the feature variables (X) and the target variable (y)
# In this case, we will assume the target variable is the last column of the dataset.
X = data.iloc[:, :-1].values
y = data.iloc[:, -1].values

# Split the data into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)

# Convert the target variable to categorical format
y_train = keras.utils.to_categorical(y_train, 8)
y_test = keras.utils.to_categorical(y_test, 8)

# Create and compile the ANN model
model1 = Sequential()
model1.add(Flatten(input_shape=(X.shape[1],)))
model1.add(Dense(64, activation='relu'))
model1.add(Dense(8, activation='softmax'))
model1.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

```

# Train the model
history = model1.fit(x_train, y_train, epochs=600, batch_size=100, validation_data=(x_test, y_test))

# Save the model
model1.save('my_model1.h5')

# Evaluate the model
evaluation = model1.evaluate(x_test, y_test)
print(f'Evaluation results - Loss: {evaluation[0]} - Accuracy: {evaluation[1]}')

# Predict the labels for the test set
y_pred = model1.predict(x_test)
y_pred_classes = np.argmax(y_pred, axis=1)
y_true = np.argmax(y_test, axis=1)

# Compute the confusion matrix
cm = confusion_matrix(y_true, y_pred_classes)

# Display the confusion matrix
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap=plt.cm.Blues)
plt.show()

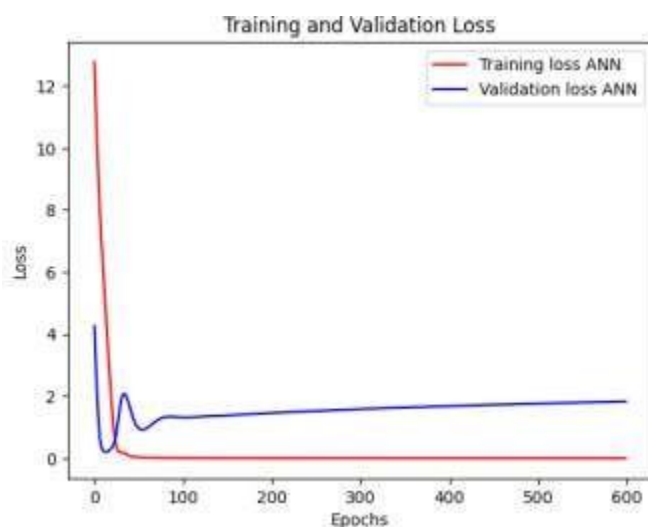
# Plot the training and validation loss
epochs = range(600)
loss1 = history.history['loss']
val_loss1 = history.history['val_loss']

plt.plot(epochs, loss1, 'r', label='Training loss ANN')
plt.plot(epochs, val_loss1, 'b', label='Validation loss ANN')
plt.legend()
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Training and Validation Loss')
plt.show()

# Display the classification report
report = classification_report(y_true, y_pred_classes)
print(report)

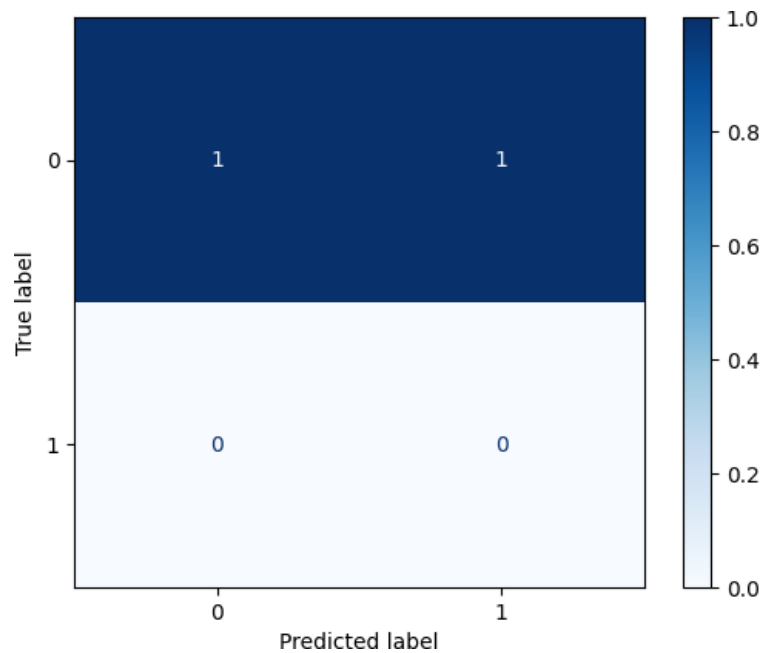
```

Grafik laju loss dalam proses training dimana hasilnya terjadi overfitting





Hasil evaluasi dengan confusion matrix diperoleh sebagai berikut:



Classification Report :

	precision	recall	f1-score	support
1	1.00	0.50	0.67	2
5	0.00	0.00	0.00	0
accuracy			0.50	2
macro avg	0.50	0.25	0.33	2
weighted avg	1.00	0.50	0.67	2

Dari hasil percobaan dengan menggunakan metode klasifikasi ANN, hasil terbaik diperoleh menggunakan

Metode	Akurasi
ANN hyperparameter 1	0.85
ANN hyperparameter 2	0.86
ANN hyperparameter 3	0.84
ANN hyperparameter 4	0.87

## 6 Klasifikasi Dengan Neuro Fuzzy System

```
!pip install numpy pandas scikit-fuzzy anfis matplotlib
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Data
data = {
    'Q1': [4, 4, 5, 7, 4],
    'Q2': [5, 4, 4, 7, 5],
    'Q3': [5, 7, 7, 7, 7],
    'Q4': [3, 7, 5, 7, 7],
    'Q5': [3, 4, 4, 7, 4],
    'Q6': [3, 4, 2, 3, 4],
    'Q7': [3, 2, 5, 2, 2],
    'Q8': [7, 7, 3, 7, 4],
    'Q9': [7, 4, 5, 7, 4],
    'Q10': [3, 7, 1, 7, 3],
    'Q11': [2, 2, 5, 6, 6],
    'Q12': [6, 6, 7, 7, 2],
    'Q13': [6, 5, 4, 2, 1],
    'Q14': [6, 7, 2, 1, 2],
    'Q15': [4, 5, 6, 6, 6],
    'Q16': [4, 5, 6, 7, 7],
    'Q17': [5, 7, 7, 7, 7],
    'Q18': [7, 7, 2, 7, 5],
    'Q19': [5, 7, 5, 7, 1],
    'Q20': [4, 4, 5, 7, 1],
    'Q21': [7, 7, 7, 1, 7],
    'Q22': [7, 7, 4, 1, 7],
    'Q23': [7, 7, 7, 1, 7],
    'Q24': [7, 4, 4, 1, 4],
    'Q25': [5, 4, 4, 1, 7],
    'Q26': [4, 4, 7, 1, 7],
    'Q27': [5, 1, 1, 7, 1]
}

df = pd.DataFrame(data)

# Target variable (kelas label, bisa dimodifikasi sesuai kebutuhan)
df['target'] = [1, 1, 0, 1, 0]

# Split data into train and test
X = df.drop('target', axis=1)
y = df['target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Standardization
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

import skfuzzy as fuzz

def fuzzify(data, low, medium, high):
    low_membership = fuzz.trimf(data, [0, 0, low])
    medium_membership = fuzz.trimf(data, [0, medium, high])
    high_membership = fuzz.trimf(data, [medium, high, high])
    return np.vstack((low_membership, medium_membership, high_membership))

fuzzified_data = np.apply_along_axis(fuzzify, 0, X_train, 3, 5, 7)

from sklearn.neural_network import MLPClassifier

# Neural Network Classifier
clf = MLPClassifier(hidden_layer_sizes=(10,), max_iter=1000, random_state=42)
clf.fit(X_train, y_train)

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Predict
y_pred = clf.predict(X_test)

# Evaluate
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)
```



```
print(f"Accuracy: {accuracy}")
print(f"Confusion Matrix:\n{conf_matrix}")
print(f"Classification Report:\n{class_report}")
```

Hasil Program :

```
Successfully built scikit-fuzzy
Installing collected packages: scikit-fuzzy, anfis
Successfully installed anfis-0.3.1 scikit-fuzzy-0.4.2
Accuracy: 0.0
Confusion Matrix:
[[0 1]
 [1 0]]
Classification Report:
```

	precision	recall	f1-score	support
0	0.00	0.00	0.00	1.0
1	0.00	0.00	0.00	1.0
accuracy			0.00	2.0
macro avg	0.00	0.00	0.00	2.0
weighted avg	0.00	0.00	0.00	2.0

## 7 Klasifikasi Dengan Support Vector Machine

```
!pip install deap

import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from deap import base, creator, tools, algorithms
import random

# Data
data = {
    'Q1': [4, 4, 5, 7, 4],
    'Q2': [5, 4, 4, 7, 5],
    'Q3': [5, 7, 7, 7, 7],
    'Q4': [3, 7, 5, 7, 7],
    'Q5': [3, 4, 4, 7, 4],
    'Q6': [3, 4, 2, 3, 4],
    'Q7': [3, 2, 5, 7, 2],
    'Q8': [7, 7, 3, 7, 4],
    'Q9': [7, 4, 5, 7, 4],
    'Q10': [3, 7, 1, 7, 3],
    'Q11': [2, 2, 5, 6, 6],
    'Q12': [6, 6, 7, 7, 2],
    'Q13': [6, 5, 4, 2, 1],
    'Q14': [6, 7, 2, 1, 2],
    'Q15': [4, 5, 6, 6, 6],
    'Q16': [4, 5, 6, 7, 7],
    'Q17': [5, 7, 7, 7, 7],
    'Q18': [7, 7, 2, 7, 5],
    'Q19': [5, 7, 5, 7, 1],
    'Q20': [4, 4, 5, 7, 1],
    'Q21': [7, 7, 7, 1, 7],
    'Q22': [7, 7, 4, 1, 7],
    'Q23': [7, 7, 7, 1, 7],
    'Q24': [7, 4, 4, 1, 4],
    'Q25': [5, 4, 4, 1, 7],
    'Q26': [4, 4, 7, 1, 7],
    'Q27': [5, 1, 1, 7, 1]
}

df = pd.DataFrame(data)

# Target variable (kelas label, bisa dimodifikasi sesuai kebutuhan)
df['Target'] = [1, 1, 0, 1, 0]
```

```

# Split data into train and test
X = df.drop('Target', axis=1)
y = df['Target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Standardization
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Genetic Algorithm for Classification
creator.create("FitnessMax", base.Fitness, weights=(1.0,))
creator.create("Individual", list, fitness=creator.FitnessMax)

toolbox = base.Toolbox()
toolbox.register("attr_bool", random.randint, 0, 1)
toolbox.register("individual", tools.initRepeat, creator.Individual, toolbox.attr_bool, n=X_train.shape[1])
toolbox.register("population", tools.initRepeat, list, toolbox.individual)

def evalOneMax(individual):
    selected_features = [i for i, bit in enumerate(individual) if bit == 1]
    if len(selected_features) == 0:
        return 0,
    X_train_selected = X_train[:, selected_features]
    X_test_selected = X_test[:, selected_features]

    clf = MLPClassifier(hidden_layer_sizes=(10,), max_iter=1000, random_state=42)
    clf.fit(X_train_selected, y_train)
    y_pred = clf.predict(X_test_selected)
    accuracy = accuracy_score(y_test, y_pred)
    return accuracy,

toolbox.register("evaluate", evalOneMax)
toolbox.register("mate", tools.cxTwoPoint)
toolbox.register("mutate", tools.mutFlipBit, indpb=0.05)
toolbox.register("select", tools.selTournament, tournsize=3)

def main():
    random.seed(42)
    pop = toolbox.population(n=50)
    hof = tools.HallOfFame(1)
    stats = tools.Statistics(lambda ind: ind.fitness.values)
    stats.register("avg", np.mean)
    stats.register("std", np.std)

```

```

    return accuracy,

toolbox.register("evaluate", evalOneMax)
toolbox.register("mate", tools.cxTwoPoint)
toolbox.register("mutate", tools.mutFlipBit, indpb=0.05)
toolbox.register("select", tools.selTournament, tournsize=3)

def main():
    random.seed(42)
    pop = toolbox.population(n=50)
    hof = tools.HallOfFame(1)
    stats = tools.Statistics(lambda ind: ind.fitness.values)
    stats.register("avg", np.mean)
    stats.register("std", np.std)
    stats.register("min", np.min)
    stats.register("max", np.max)

    algorithms.eaSimple(pop, toolbox, cxpb=0.5, mutpb=0.2, ngen=40, stats=stats, halloffame=hof, verbose=True)

    return pop, stats, hof

if __name__ == "__main__":
    pop, stats, hof = main()
    best_individual = hof[0]
    print(f"Best individual is: {best_individual}")
    print(f"With fitness: {best_individual.fitness.values[0]}")

    # Evaluate the best individual on the test set
    selected_features = [i for i, bit in enumerate(best_individual) if bit == 1]
    X_train_selected = X_train[:, selected_features]
    X_test_selected = X_test[:, selected_features]

    clf = MLPClassifier(hidden_layer_sizes=(10,), max_iter=1000, random_state=42)
    clf.fit(X_train_selected, y_train)
    y_pred = clf.predict(X_test_selected)

    accuracy = accuracy_score(y_test, y_pred)
    conf_matrix = confusion_matrix(y_test, y_pred)
    class_report = classification_report(y_test, y_pred)

    print(f"Accuracy: {accuracy}")
    print(f"Confusion Matrix:\n{conf_matrix}")
    print(f"Classification Report:\n{class_report}")

```

## Hasil Program:

```

already satisfied: deap in /usr/local/lib/python3.10/dist-packages (1.4.1)
already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from deap) (1.25.2)
lib/python3.10/dist-packages/deap/creator.py:185: RuntimeWarning: A class named 'FitnessMax' has a
warn("A class named '{0}' has already been created and it ")
lib/python3.10/dist-packages/deap/creator.py:185: RuntimeWarning: A class named 'Individual' has a
warn("A class named '{0}' has already been created and it ")
als avg std min max
0.06 0.199517 0 1
lib/python3.10/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:688: ConvergenceWarn
warn(
0.85 0.21 0 1
0.88 0.245764 0 1
lib/python3.10/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:688: ConvergenceWarn
warn(
0.8 0.223687 0 1
0.83 0.173484 0.5 1
0.88 0.207133 0.5 1
0.97 0.138743 0.5 1
0.95 0.15 0.5 1
0.81 0.192094 0.5 1
0.95 0.11 0.5 1
0.96 0.135647 0.5 1
0.97 0.138743 0.5 1
0.98 0.135647 0.5 1
0.91 0.192094 0.5 1
0.92 0.183383 0.5 1
0.95 0.15 0.5 1
0.98 0.0979796 0.5 1
0.95 0.15 0.5 1
lib/python3.10/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:688: ConvergenceWarn
warn(
0.93 0.20825 0 1
0.95 0.15 0.5 1
0.94 0.162481 0.5 1
0.94 0.162481 0.5 1
0.96 0.135647 0.5 1
0.98 0.135647 0.5 1
0.94 0.162481 0.5 1
0.83 0.173484 0.5 1
0.88 0.0979796 0.5 1
0.8 0.223687 0 1
0.96 0.135647 0.5 1
lib/python3.10/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:688: ConvergenceWarn

```

```

lib/python3.10/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:688: ConvergenceWarn
warn(
0.93 0.20825 0 1
0.95 0.15 0.5 1
0.94 0.162481 0.5 1
0.94 0.162481 0.5 1
0.96 0.135647 0.5 1
0.98 0.135647 0.5 1
0.94 0.162481 0.5 1
0.93 0.173484 0.5 1
0.98 0.0979796 0.5 1
0.8 0.223687 0 1
0.96 0.135647 0.5 1
lib/python3.10/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:688: ConvergenceWarn
warn(
0.92 0.183383 0.5 1
0.97 0.138743 0.5 1
lib/python3.10/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:688: ConvergenceWarn
warn(
0.96 0.135647 0.5 1
0.97 0.138743 0.5 1
0.93 0.173484 0.5 1
0.97 0.138743 0.5 1
0.98 0.135647 0.5 1
0.98 0.0979796 0.5 1
lib/python3.10/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:688: ConvergenceWarn
warn(
0.96 0.135647 0.5 1
0.97 0.138743 0.5 1
0.87 0.245764 0 1
0.92 0.183383 0.5 1
dual is: [0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0]
s: 1.0
.0
atrix:

Ion Report:
precision recall f1-score support
0 1.00 1.00 1.00 1
1 1.00 1.00 1.00 1
y
0 1.00 1.00 1.00 2
1 1.00 1.00 1.00 2

```

## 8 Klasifikasi Dengan Genetic Algorithm

```
✓ 20s # Install DEAP library
!pip install deap

import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.svm import SVC
from deap import base, creator, tools, algorithms
import random

# Data
data = {
    'Q1': [4, 4, 5, 7, 4],
    'Q2': [5, 4, 4, 7, 5],
    'Q3': [5, 7, 7, 7, 7],
    'Q4': [3, 7, 5, 7, 7],
    'Q5': [3, 4, 4, 7, 4],
    'Q6': [3, 4, 2, 3, 4],
    'Q7': [3, 2, 5, 7, 2],
    'Q8': [7, 7, 3, 7, 4],
    'Q9': [7, 4, 5, 7, 4],
    'Q10': [3, 7, 1, 7, 3],
    'Q11': [2, 2, 5, 6, 6],
    'Q12': [6, 6, 7, 7, 2],
    'Q13': [6, 5, 4, 2, 1],
    'Q14': [6, 7, 2, 1, 2],
    'Q15': [4, 5, 6, 6, 6],
    'Q16': [4, 5, 6, 7, 7],
    'Q17': [5, 7, 7, 7, 7],
    'Q18': [7, 7, 2, 7, 5],
    'Q19': [5, 7, 5, 7, 1],
    'Q20': [4, 4, 5, 7, 1],
    'Q21': [7, 7, 7, 1, 7],
    'Q22': [7, 7, 4, 1, 7],
    'Q23': [7, 7, 7, 1, 7],
    'Q24': [7, 4, 4, 1, 4],
    'Q25': [5, 4, 4, 1, 7],
    'Q26': [4, 4, 7, 1, 7],
    'Q27': [5, 1, 1, 7, 1]
}
```

```

df = pd.DataFrame(data)

# Target variable (kelas label, bisa dimodifikasi sesuai kebutuhan)
df['Target'] = [1, 1, 0, 1, 0]

# Split data into train and test
X = df.drop('Target', axis=1)
y = df['Target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Standardization
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Genetic Algorithm for Classification
creator.create("FitnessMax", base.Fitness, weights=(1.0,))
creator.create("Individual", list, fitness=creator.FitnessMax)

toolbox = base.Toolbox()
toolbox.register("attr_bool", random.randint, 0, 1)
toolbox.register("individual", tools.initRepeat, creator.Individual, toolbox.attr_bool, n=X_train.shape[1])
toolbox.register("population", tools.initRepeat, list, toolbox.individual)

def evalOneMax(individual):
    selected_features = [i for i, bit in enumerate(individual) if bit == 1]
    if len(selected_features) == 0:
        return 0,
    X_train_selected = X_train[:, selected_features]
    X_test_selected = X_test[:, selected_features]

    clf = SVC(kernel='linear', random_state=42)
    clf.fit(X_train_selected, y_train)
    y_pred = clf.predict(X_test_selected)
    accuracy = accuracy_score(y_test, y_pred)
    return accuracy,

toolbox.register("evaluate", evalOneMax)
toolbox.register("mate", tools.cxTwoPoint)
toolbox.register("mutate", tools.mutFlipBit, indpb=0.05)
toolbox.register("select", tools.selTournament, tournsize=3)

def main():
    random.seed(42)

```



```

def main():
    random.seed(42)
    pop = toolbox.population(n=50)
    hof = tools.HallOfFame(1)
    stats = tools.Statistics(lambda ind: ind.fitness.values)
    stats.register("avg", np.mean)
    stats.register("std", np.std)
    stats.register("min", np.min)
    stats.register("max", np.max)

    algorithms.eaSimple(pop, toolbox, cxpb=0.5, mutpb=0.2, ngen=40, stats=stats, halloffame=hof, verbose=True)

    return pop, stats, hof

if __name__ == "__main__":
    pop, stats, hof = main()
    best_individual = hof[0]
    print(f"Best individual is: {best_individual}")
    print(f"With fitness: {best_individual.fitness.values[0]}")

    # Evaluate the best individual on the test set
    selected_features = [i for i, bit in enumerate(best_individual) if bit == 1]
    X_train_selected = X_train[:, selected_features]
    X_test_selected = X_test[:, selected_features]

    clf = SVC(kernel='linear', random_state=42)
    clf.fit(X_train_selected, y_train)
    y_pred = clf.predict(X_test_selected)

    accuracy = accuracy_score(y_test, y_pred)
    conf_matrix = confusion_matrix(y_test, y_pred)
    class_report = classification_report(y_test, y_pred)

    print(f"Accuracy: {accuracy}")
    print(f"Confusion Matrix:\n{conf_matrix}")
    print(f"Classification Report:\n{class_report}")

```



Hasil Program :

```
Successfully installed deap-1.4.1
gen  fitness avg      std      min      max
0      30      0.00      0.762488      0      1
1      20      0.88      0.207123      0.5      1
2      30      1      0      1      1
3      31      0.99      0.07      0.5      1
4      33      0.97      0.118743      0.5      1
5      32      0.95      0.15      0.5      1
6      21      0.94      0.162483      0.5      1
7      29      0.96      0.135647      0.5      1
8      40      0.94      0.162483      0.5      1
9      24      0.99      0.07      0.5      1
10     35      1      0      1      1
11     32      0.97      0.118743      0.5      1
12     27      1      0      1      1
13     33      0.99      0.07      0.5      1
14     30      0.99      0.07      0.5      1
15     33      0.96      0.136647      0.5      1
16     35      0.97      0.118743      0.5      1
17     30      0.99      0.07      0.5      1
18     29      1      0      1      1
19     33      0.96      0.135647      0.5      1
20     30      0.88      0.0979796      0.0      1
21     23      0.97      0.118743      0.5      1
22     35      0.99      0.07      0.5      1
23     27      0.97      0.118743      0.5      1
24     32      0.98      0.07      0.5      1
25     32      0.97      0.118743      0.5      1
26     24      0.88      0.0979796      0.5      1
27     37      1      0      1      1
28     30      0.99      0.07      0.5      1
29     20      0.88      0.0979796      0.5      1
30     24      1      0      1      1
31     30      0.99      0.07      0.5      1
32     30      0.99      0.07      0.5      1
33     40      0.97      0.118743      0.5      1
34     20      0.99      0.07      0.5      1
35     31      1      0      1      1
36     30      0.99      0.07      0.5      1
37     25      0.99      0.07      0.5      1
38     31      0.99      0.07      0.5      1
39     35      0.88      0.0979796      0.5      1
40     41      0.99      0.07      0.5      1
Best individual is: [0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0]
with fitness: 1.0
Accuracy: 1.0
Confusion Matrix:
[[1 0]
 [0 1]]
```

Classification Report:					
	precision	recall	f1-score	support	
0	1.00	1.00	1.00	1	
1	1.00	1.00	1.00	1	
accuracy			1.00	2	
macro avg	1.00	1.00	1.00	2	
weighted avg	1.00	1.00	1.00	2	

## 9 Pembangunan Model

```
import numpy as np
import pandas as pd
import joblib
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Data
data = {
    'Q1': [4, 4, 5, 7, 4],
    'Q2': [5, 4, 7, 7, 5],
    'Q3': [5, 7, 7, 7, 7],
    'Q4': [3, 7, 5, 7, 7],
    'Q5': [3, 4, 4, 7, 4],
    'Q6': [3, 4, 2, 3, 4],
    'Q7': [3, 2, 5, 7, 2],
    'Q8': [7, 7, 3, 7, 4],
    'Q9': [7, 4, 5, 7, 4],
    'Q10': [3, 7, 1, 7, 3],
    'Q11': [2, 2, 5, 6, 6],
    'Q12': [6, 6, 7, 7, 2],
    'Q13': [6, 5, 4, 2, 1],
    'Q14': [6, 7, 2, 1, 2],
    'Q15': [4, 5, 6, 6, 6],
    'Q16': [4, 5, 6, 7, 7],
    'Q17': [5, 7, 7, 7, 7],
    'Q18': [7, 7, 2, 7, 5],
```

```
    'Q21': [7, 7, 7, 1, 7],
    'Q22': [7, 7, 4, 1, 7],
    'Q23': [7, 7, 7, 1, 7],
    'Q24': [7, 4, 4, 1, 4],
    'Q25': [5, 4, 4, 1, 7],
    'Q26': [4, 4, 7, 1, 7],
    'Q27': [5, 1, 1, 7, 1]
}

# Convert data to DataFrame
df = pd.DataFrame(data)

# Target variable (prediksi harga rumah)
y = np.array([8, 7, 5, 8, 7]) # Example satisfaction scores

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(df, y, test_size=0.3, random_state=42)

# Standardize the data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Initialize the Decision Tree Regressor model
model = DecisionTreeRegressor()

# Train the model
model.fit(X_train, y_train)

# Save the model to a file
joblib.dump(model, 'DTRModel.pkl')

print("Model DTR untuk prediksi harga rumah telah dilatih dan disimpan sebagai DTRModel.pkl")
```

Model DTR untuk prediksi harga rumah telah dilatih dan disimpan sebagai DTRModel.pkl

## 10 Hasil dan Pengujian

### Prediksi Harga Rumah Artificial Neural Network

Lokasi:

Premium

Ukuran (m<sup>2</sup>):

36

Jumlah Kamar Tidur:

1

Jumlah Kamar Mandi:

1

Prediksi Harga

Hasil Prediksi  
±  
Rp. -

© 2024 Prediksi Harga Rumah

### Prediksi Harga Rumah Artificial Neural Network

Lokasi:

Premium

Ukuran (m<sup>2</sup>):

36

Jumlah Kamar Tidur:

1

Jumlah Kamar Mandi:

1

Prediksi Harga

Hasil Prediksi  
±  
Rp. 500,000,000

© 2024 Prediksi Harga Rumah