

✓ Import Library

```
import pandas as pd
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objs as go
```

✓ Visualisasi dengan Matplotlib

✓ Menampilkan Histogram

```
# HISTOGRAM DAN KURVA DISTRIBUSI NORMAL

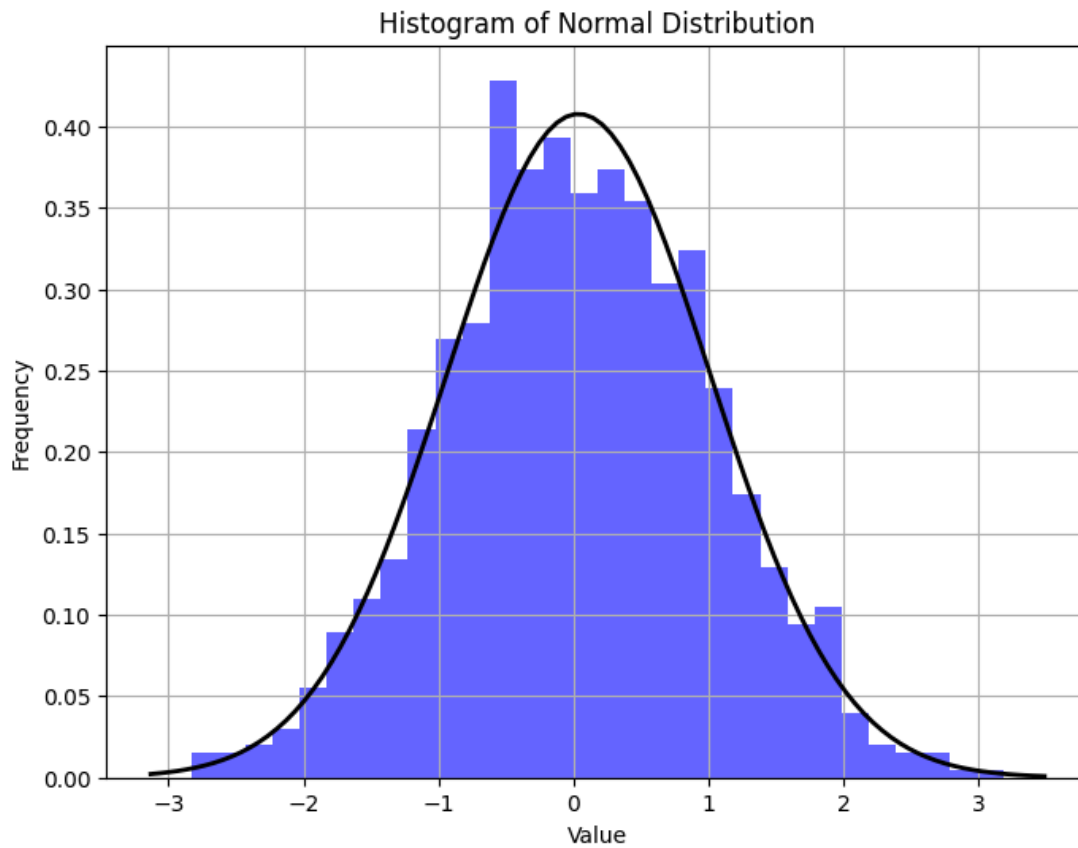
# Generate random data
data = pd.Series(np.random.randn(1000))

# Plot histogram
plt.figure(figsize=(8, 6))
plt.hist(data, bins=30, density=True, alpha=0.6, color='b')

# Add a normal distribution curve
mu, std = data.mean(), data.std()
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
p = norm.pdf(x, mu, std)
plt.plot(x, p, 'k', linewidth=2)

# Customize plot
plt.title('Histogram of Normal Distribution')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.grid(True)

# Show plot
plt.show()
```



▼ Menampilkan *Boxplot*

```
# BOXPLOT
```

```
# Generate random data
```

```
data = pd.DataFrame(np.random.rand(10, 5), columns=['A', 'B', 'C', 'D', 'E'])
```

```
# Plot boxplot
```

```
plt.figure(figsize=(8, 6))
```

```
data.boxplot()
```

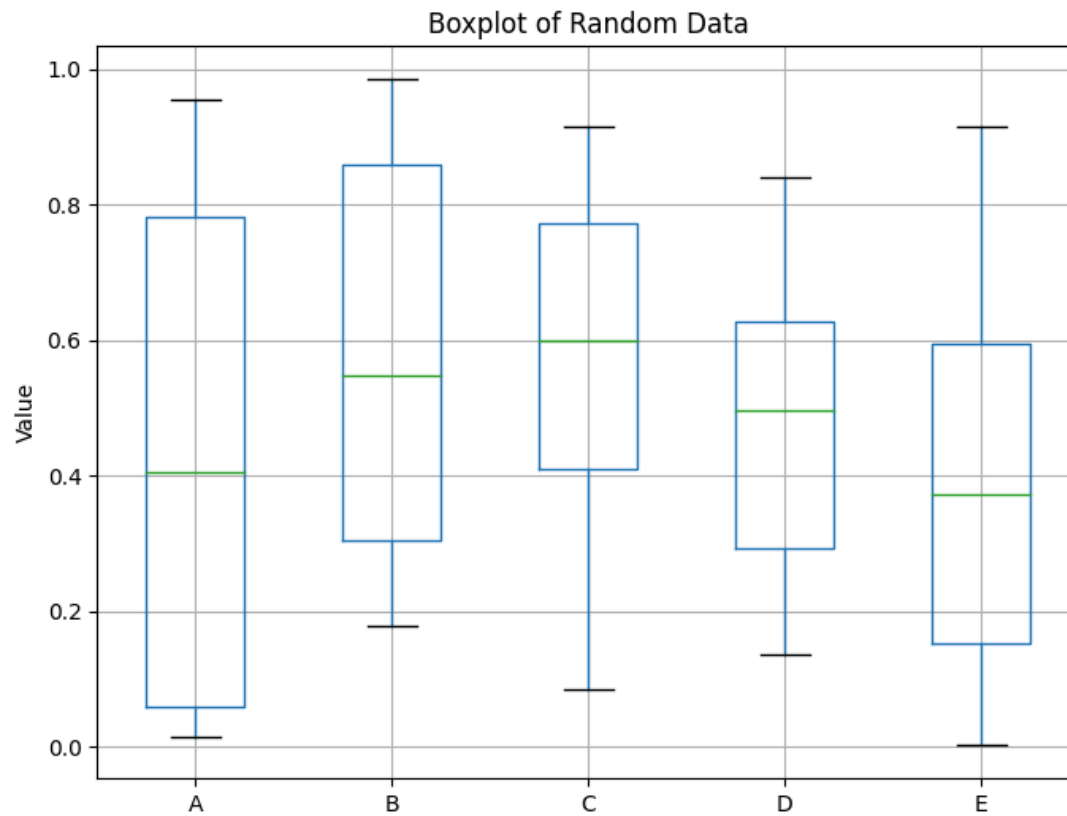
```
# Customize plot
```

```
plt.title('Boxplot of Random Data')
```

```
plt.ylabel('Value')
```

```
# Show plot
```

```
plt.show()
```



✓ Menampilkan *Bar Chart*

```
# BAR CHART

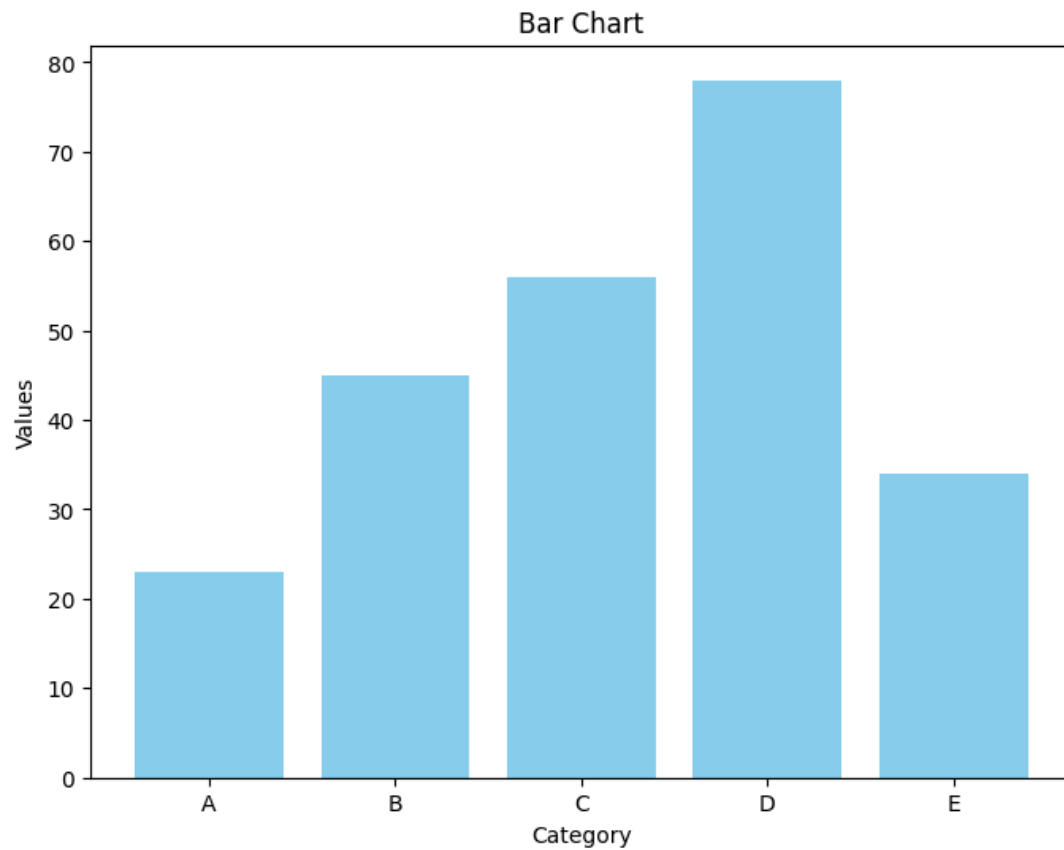
# Data
data = {'Category': ['A', 'B', 'C', 'D', 'E'],
        'Values': [23, 45, 56, 78, 34]}

# Konversi data menjadi DataFrame
df = pd.DataFrame(data)

# Plot bar chart
plt.figure(figsize=(8, 6))
plt.bar(df['Category'], df['Values'], color='skyblue')

# Tambahkan label dan judul
plt.xlabel('Category')
plt.ylabel('Values')
plt.title('Bar Chart')

# Tampilkan plot
plt.show()
```



✓ Menampilkan *Line Chart*

```
# LINE CHART
```

```
# Contoh data dalam bentuk DataFrame pandas
```

```
data = {  
    'Year': [2010, 2011, 2012, 2013, 2014, 2015, 2016],  
    'Sales': [1000, 1200, 1500, 1800, 2000, 2100, 2300]  
}
```

```
df = pd.DataFrame(data)
```

```
# Plot line chart
```

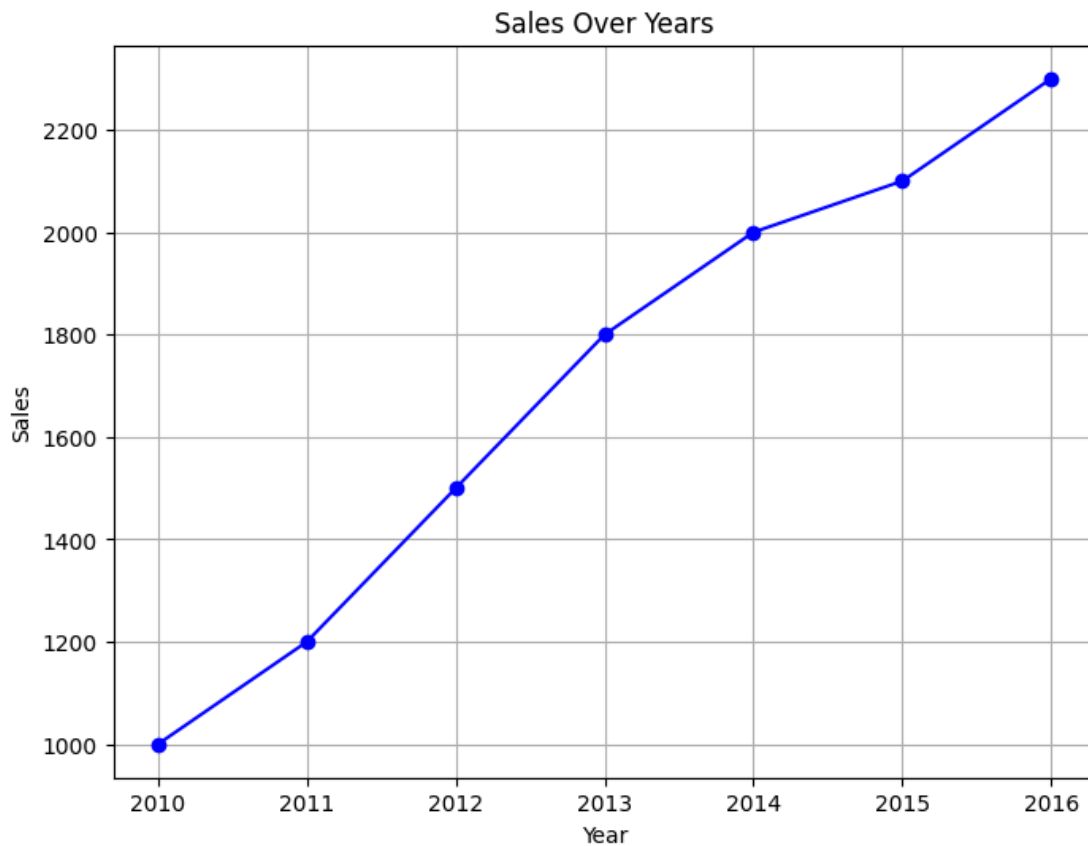
```
plt.figure(figsize=(8, 6))  
plt.plot(df['Year'], df['Sales'], marker='o', color='b', linestyle='-')
```

```
# Customize plot
```

```
plt.title('Sales Over Years')  
plt.xlabel('Year')  
plt.ylabel('Sales')  
plt.grid(True)
```

```
# Show plot
```

```
plt.show()
```



✓ Menampilkan *Pie Chart*

```
# PIE CHART

# Data
data = {'Category': ['A', 'B', 'C', 'D', 'E'],
        'Values': [30, 20, 15, 10, 25]}

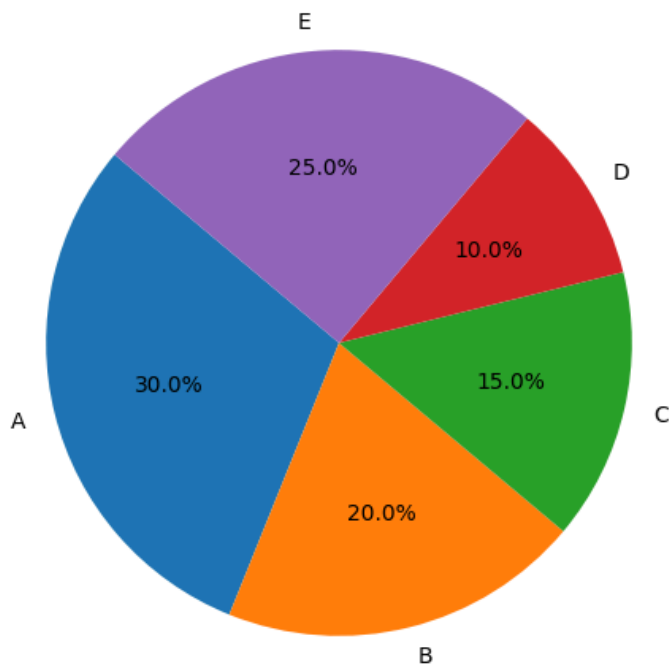
# Konversi data ke dalam DataFrame pandas
df = pd.DataFrame(data)

# Plot pie chart
plt.figure(figsize=(8, 6))
plt.pie(df['Values'], labels=df['Category'], autopct='%1.1f%%', startangle=140)

# Customisasi plot
plt.title('Pie Chart of Categories')

# Tampilkan plot
plt.show()
```

Pie Chart of Categories



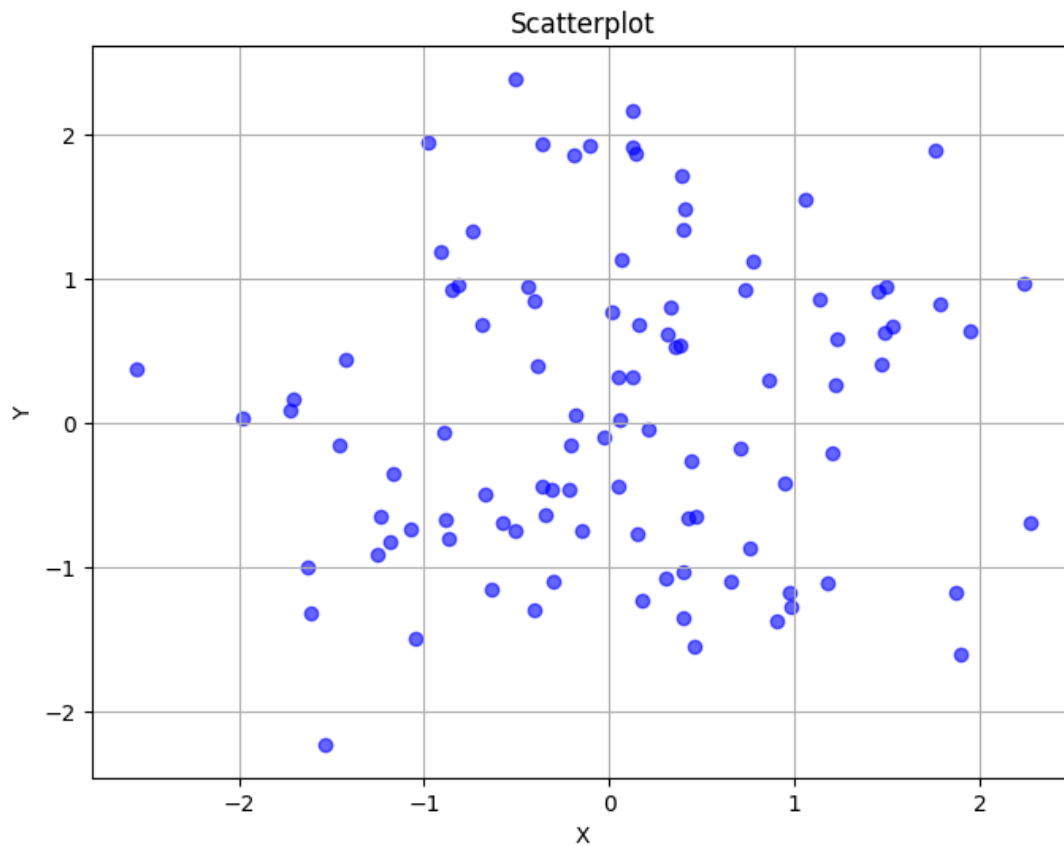
✓ Menampilkan *Scatter Plot*

```
# SCATTER PLOT

# Generate random data
np.random.seed(0)
data = pd.DataFrame({
    'x': np.random.randn(100),
    'y': np.random.randn(100)
})

# Plot scatterplot
plt.figure(figsize=(8, 6))
plt.scatter(data['x'], data['y'], color='b', alpha=0.6)

# Customize plot
plt.title('Scatterplot')
plt.xlabel('X')
plt.ylabel('Y')
plt.grid(True)
# Show plot
plt.show()
```



✓ Visualisasi dengan Seaborn

✓ Menampilkan Histogram

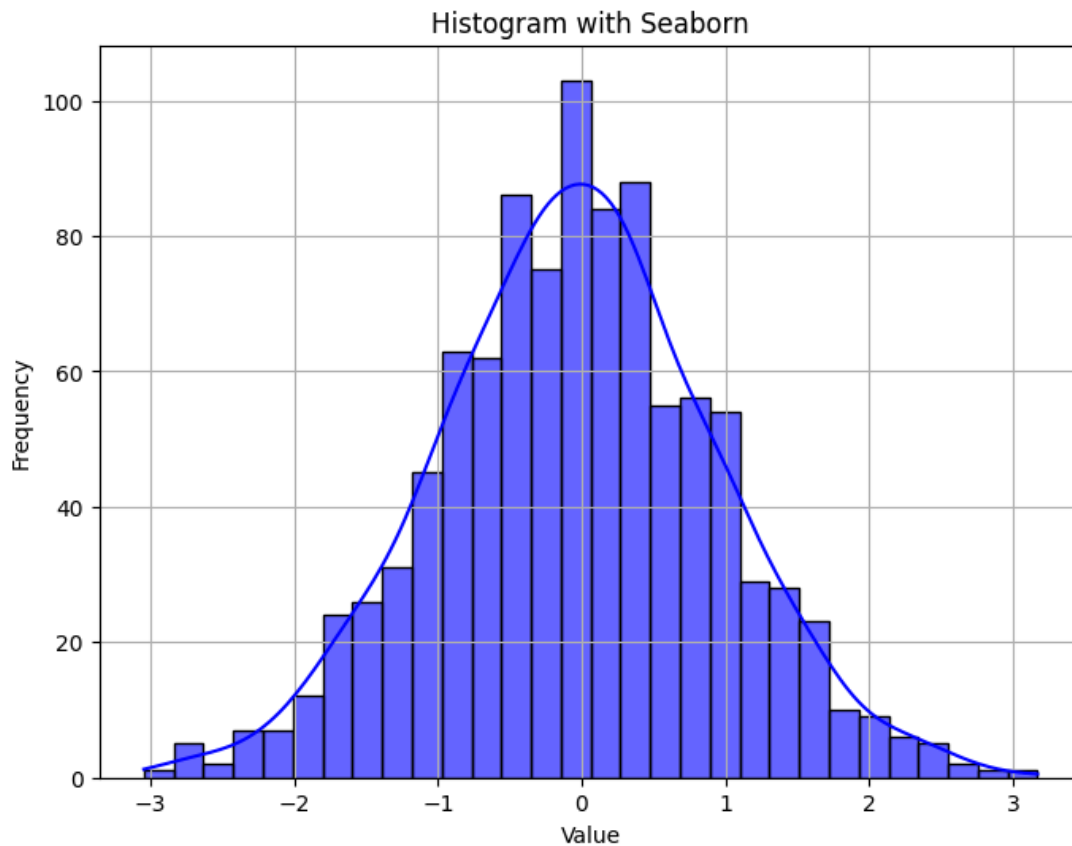
```
# SEABORN, HISTOGRAM

# Generate random data
data = pd.Series(np.random.randn(1000))

# Plot histogram with seaborn
plt.figure(figsize=(8, 6))
sns.histplot(data, bins=30, kde=True, color='b', alpha=0.6)

# Customize plot
plt.title('Histogram with Seaborn')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.grid(True)

# Show plot
plt.show()
```



✓ Menampilkan *Boxplot*

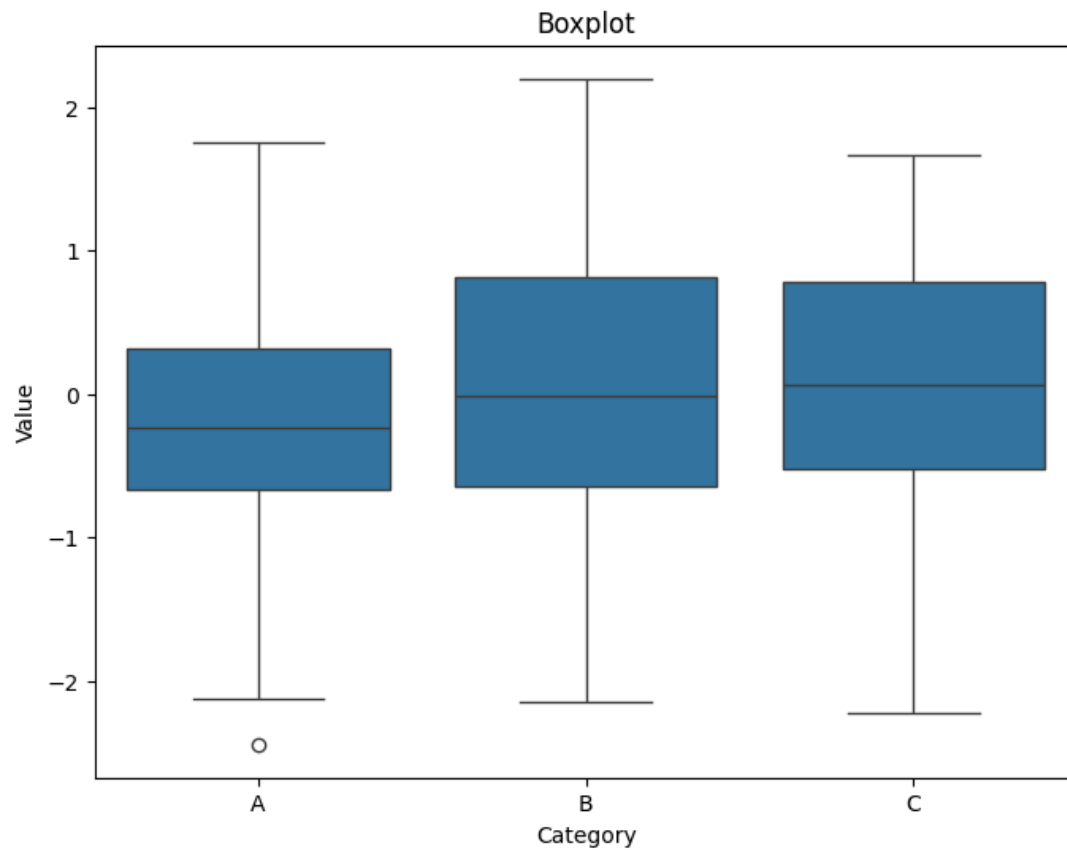
```
# Boxplot

# Generate random data
np.random.seed(0)
data = pd.DataFrame({
    'Category': np.random.choice(['A', 'B', 'C'], size=100),
    'Value': np.random.randn(100)
})

# Plot boxplot
plt.figure(figsize=(8, 6))
sns.boxplot(x='Category', y='Value', data=data)

# Customize plot
plt.title('Boxplot')
plt.xlabel('Category')
plt.ylabel('Value')

# Show plot
plt.show()
```



✓ Menampilkan *Bar Chart*

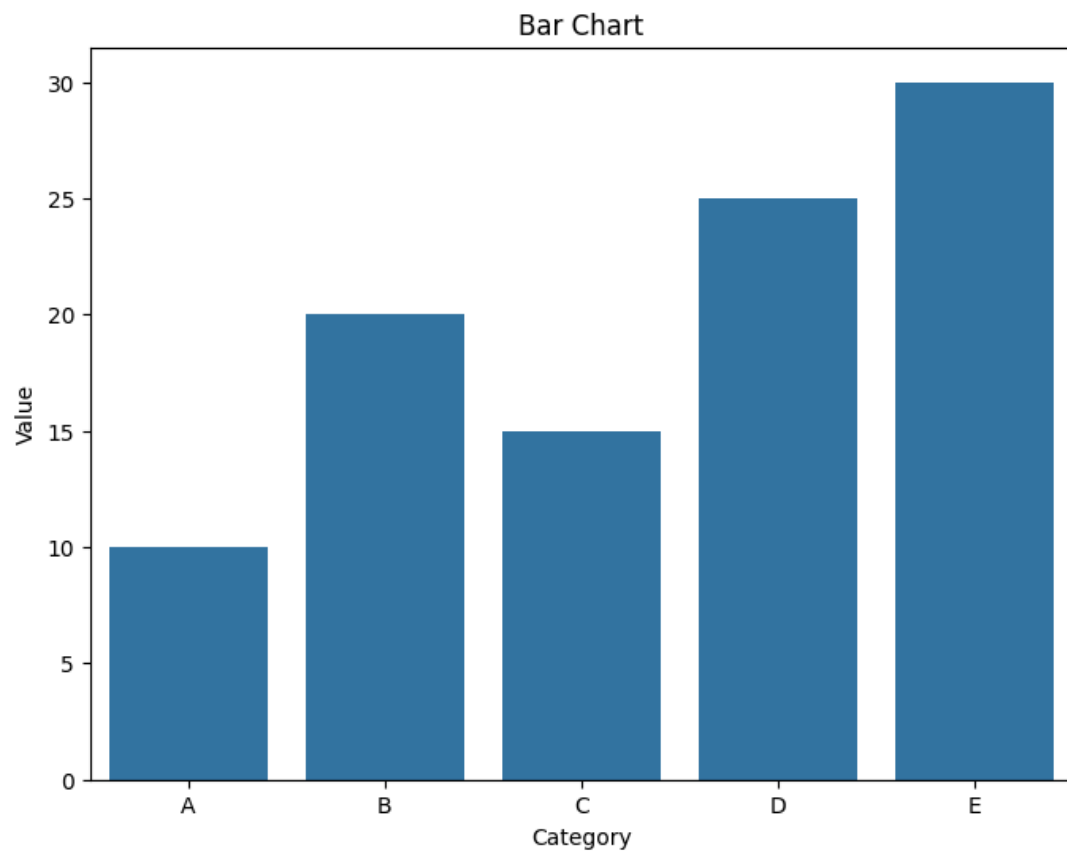
```
# BAR CHART

# Generate sample data
data = pd.DataFrame({
    'Category': ['A', 'B', 'C', 'D', 'E'],
    'Value': [10, 20, 15, 25, 30]
})

# Plot bar chart
plt.figure(figsize=(8, 6))
sns.barplot(x='Category', y='Value', data=data)

# Customize plot
plt.title('Bar Chart')
plt.xlabel('Category')
plt.ylabel('Value')

# Show plot
plt.show()
```



✓ Menampilkan *Line Chart*

```

# LINE PLOT

# Generate random time series data
np.random.seed(0)
dates = pd.date_range('2022-01-01', periods=100)
data = pd.DataFrame({
    'Date': dates,
    'Value': np.random.randn(100).cumsum()
})

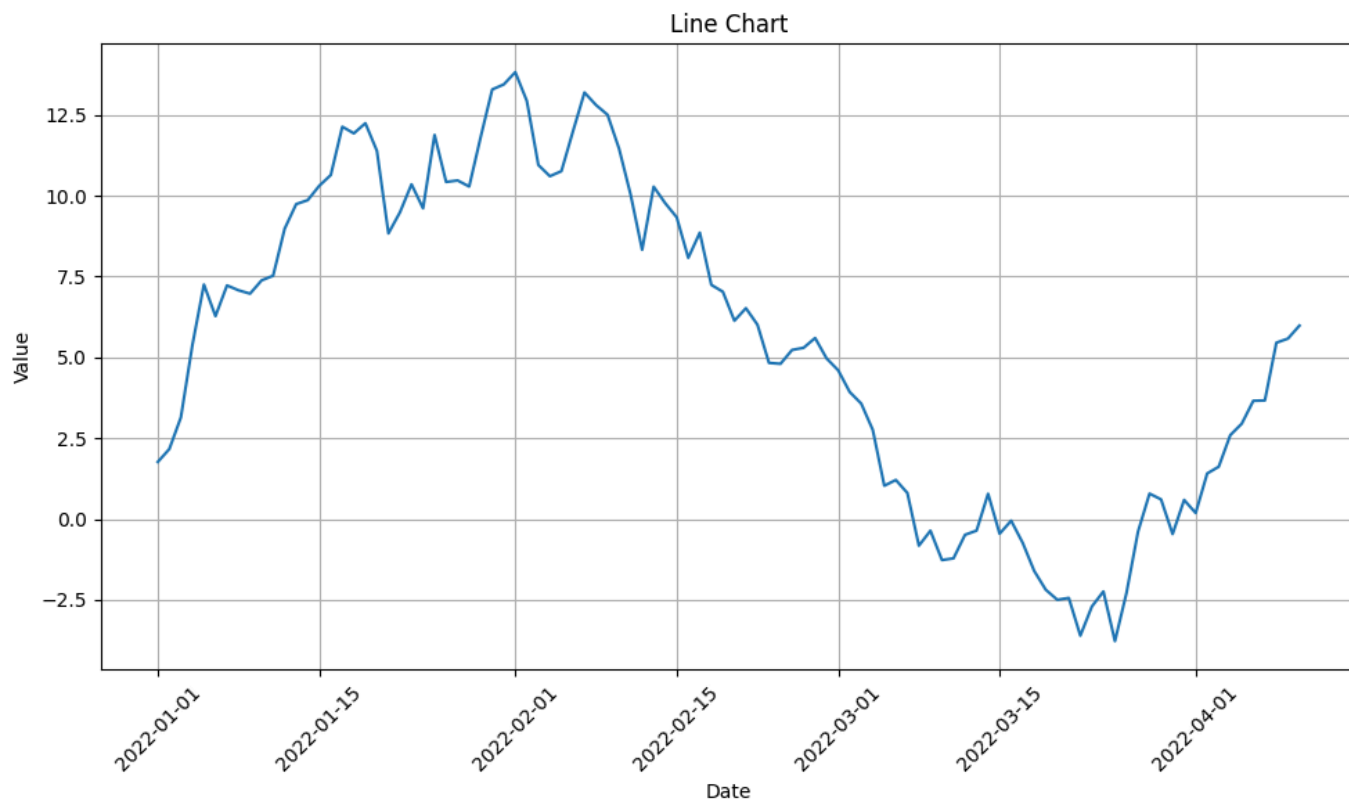
# Plot line chart using Seaborn
plt.figure(figsize=(10, 6))
sns.lineplot(x='Date', y='Value', data=data)

# Customize plot
plt.title('Line Chart')
plt.xlabel('Date')
plt.ylabel('Value')
plt.grid(True)

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Show plot
plt.tight_layout()
plt.show()

```



✓ Menampilkan *Pie Chart*

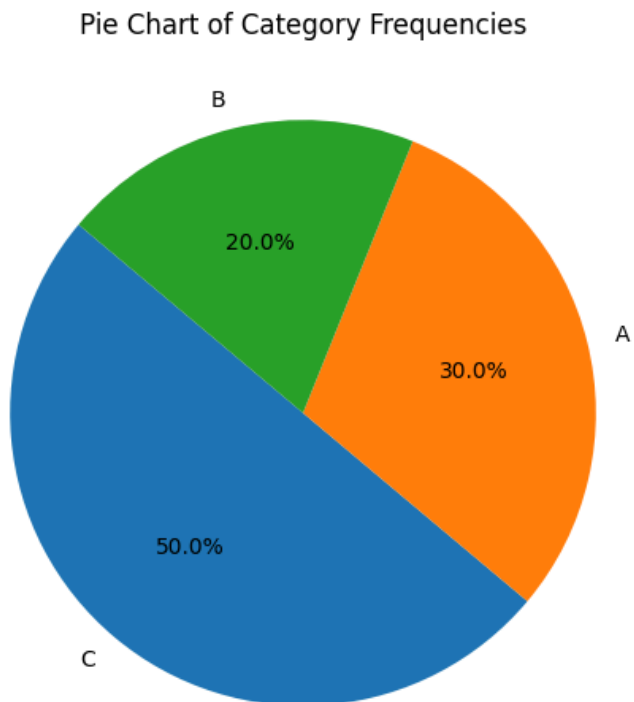
```
# PIE CHART

# Contoh data
data = pd.Series(['A', 'A', 'B', 'C', 'A', 'B', 'C', 'C', 'C', 'C'])

# Hitung frekuensi kategori
counts = data.value_counts()

# Plot pie chart
plt.figure(figsize=(8, 6))
plt.pie(counts, labels=counts.index, autopct='%1.1f%%', startangle=140)
plt.title('Pie Chart of Category Frequencies')

# Tampilkan plot
plt.show()
```



✓ Menampilkan *Scatter Plot*

```

# SCATTER PLOT

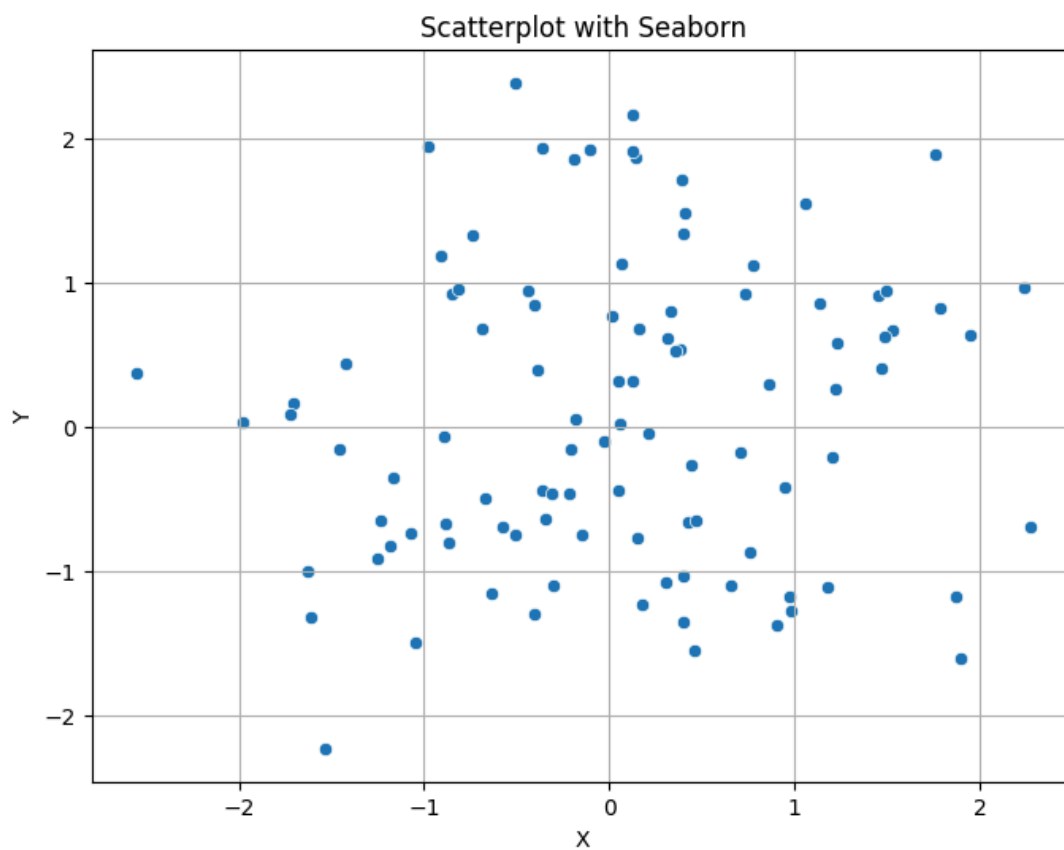
# Generate random data
np.random.seed(0)
data = pd.DataFrame({
    'x': np.random.randn(100),
    'y': np.random.randn(100)
})

# Plot scatterplot with seaborn
plt.figure(figsize=(8, 6))
sns.scatterplot(data=data, x='x', y='y')

# Customize plot
plt.title('Scatterplot with Seaborn')
plt.xlabel('X')
plt.ylabel('Y')
plt.grid(True)

# Show plot
plt.show()

```



- ✓ Visualisasi dengan Plotly
- ✓ Menampilkan Histogram

```

# Generate data using NumPy
np.random.seed(0) # untuk hasil yang konsisten
n_data_points = 1000
data = {
    'id': range(1, n_data_points + 1),
    'tinggi_badan': np.random.normal(loc=170, scale=10, size=n_data_points)
}
# Create DataFrame using Pandas
df = pd.DataFrame(data)

# Create histogram using Plotly
histogram = go.Histogram(x=df['tinggi_badan'],
                        marker_color='skyblue',
                        xbins=dict(start=np.min(df['tinggi_badan']),
                                end=np.max(df['tinggi_badan']),
                                size=10))

layout = go.Layout(title='Histogram Tinggi Badan',
                  xaxis=dict(title='Tinggi Badan'),
                  yaxis=dict(title='Frekuensi'))

fig = go.Figure(data=[histogram], layout=layout)
fig.show()

```



▼ Menampilkan *Boxplot*

```

# Generate data using NumPy
np.random.seed(0) # untuk hasil yang konsisten
n_data_points = 1000
tinggi_badan = np.random.normal(loc=170, scale=10, size=n_data_points)

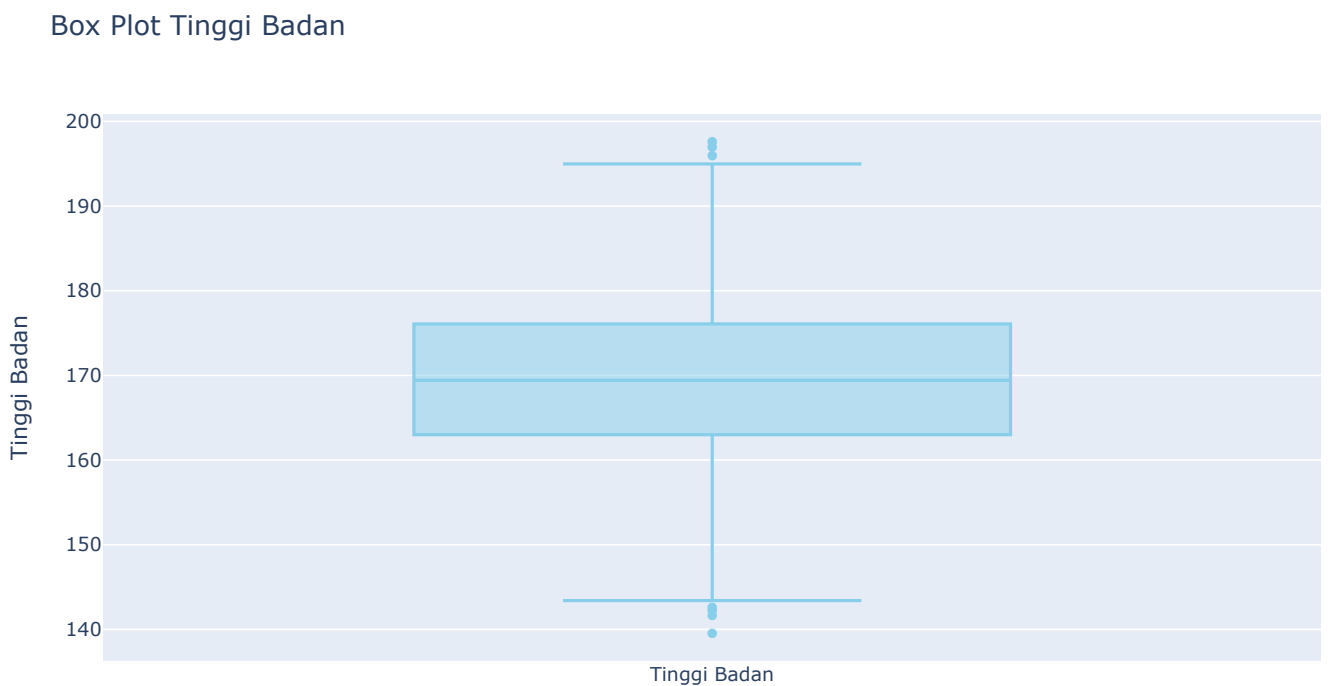
# Create DataFrame using Pandas
df = pd.DataFrame({'id': range(1, n_data_points + 1), 'tinggi_badan': tinggi_badan})

# Create box plot using Plotly
fig = go.Figure(data=[go.Box(y=df['tinggi_badan'],
                             name='Tinggi Badan',
                             marker_color='skyblue')])

# Set plot layout
fig.update_layout(title='Box Plot Tinggi Badan',
                  yaxis_title='Tinggi Badan')

# Show plot
fig.show()

```



✓ Menampilkan *Bar Chart*

```
# Contoh DataFrame
data = {'Nama': ['John', 'Anna', 'Peter', 'Linda'],
        'Usia': [28, 35, 40, 32]}

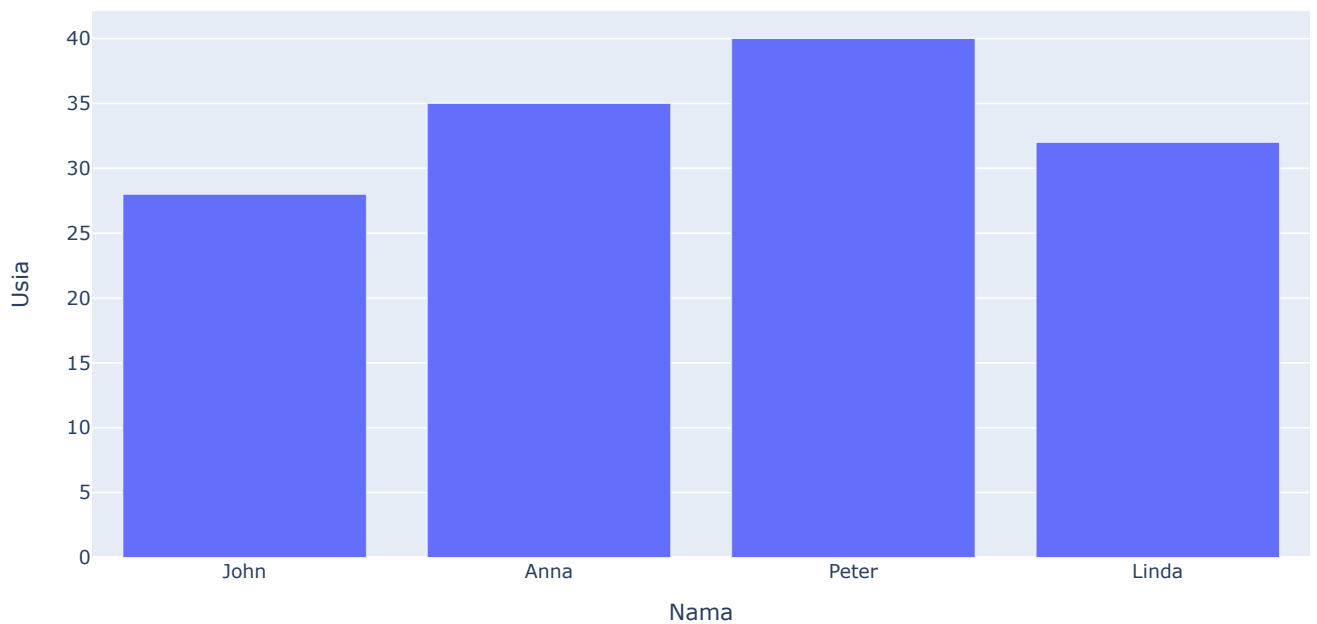
df = pd.DataFrame(data)

# Membuat bar chart dengan Plotly
fig = go.Figure([go.Bar(x=df['Nama'], y=df['Usia'])])

# Menambahkan judul dan label sumbu
fig.update_layout(title='Diagram Batang Usia',
                  xaxis_title='Nama',
                  yaxis_title='Usia')

# Menampilkan plot
fig.show()
```

Diagram Batang Usia



✓ Menampilkan *Line Chart*

```

# Contoh DataFrame
data = {
    'tahun': [2015, 2016, 2017, 2018, 2019],
    'penjualan': [1000, 1500, 2000, 1800, 2200]
}
df = pd.DataFrame(data)

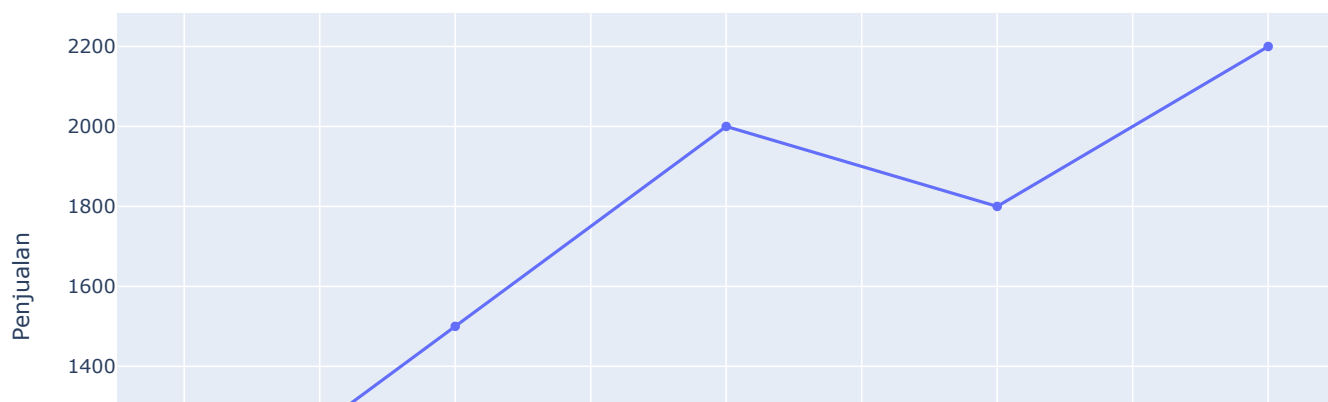
# Membuat line chart menggunakan Plotly
fig = go.Figure(data=go.Scatter(x=df['tahun'], y=df['penjualan'], mode='lines+markers', name='Penjualan'))

# Menyesuaikan layout plot
fig.update_layout(title='Grafik Penjualan Tahunan',
                  xaxis_title='Tahun',
                  yaxis_title='Penjualan')

# Menampilkan plot
fig.show()

```

Grafik Penjualan Tahunan



▼ Menampilkan *Pie Chart*

