

Tujuan Praktikum

- Mahasiswa memahami cara membuat dan menampilkan grafik dengan *library* Matplotlib
- Mahasiswa mengetahui fitur-fitur utama yang tersedia di *library* Matplotlib

Jangan lupa untuk selalu import *library* Matplotlib ketika memulai sesi baru (*run time*) atau kode Python yang memerlukan penggunaan *library* Matplotlib

```
# Import Matplotlib
import matplotlib.pyplot as plt
```

1. Inisialisasi dan Labeling pada Figure (Bobot Nilai: 15)

a. Buat *figure* dengan ukuran 10x8 inci dengan resolusi 72 DPI.

Tip: gunakan `plt.figure()`

```
# Inisialisasi figure dengan ukuran dan dpi yang sudah ditentukan
fig = plt.figure(figsize=(10,8), dpi=72)

<Figure size 720x576 with 0 Axes>
```

Test case program

Input

Output

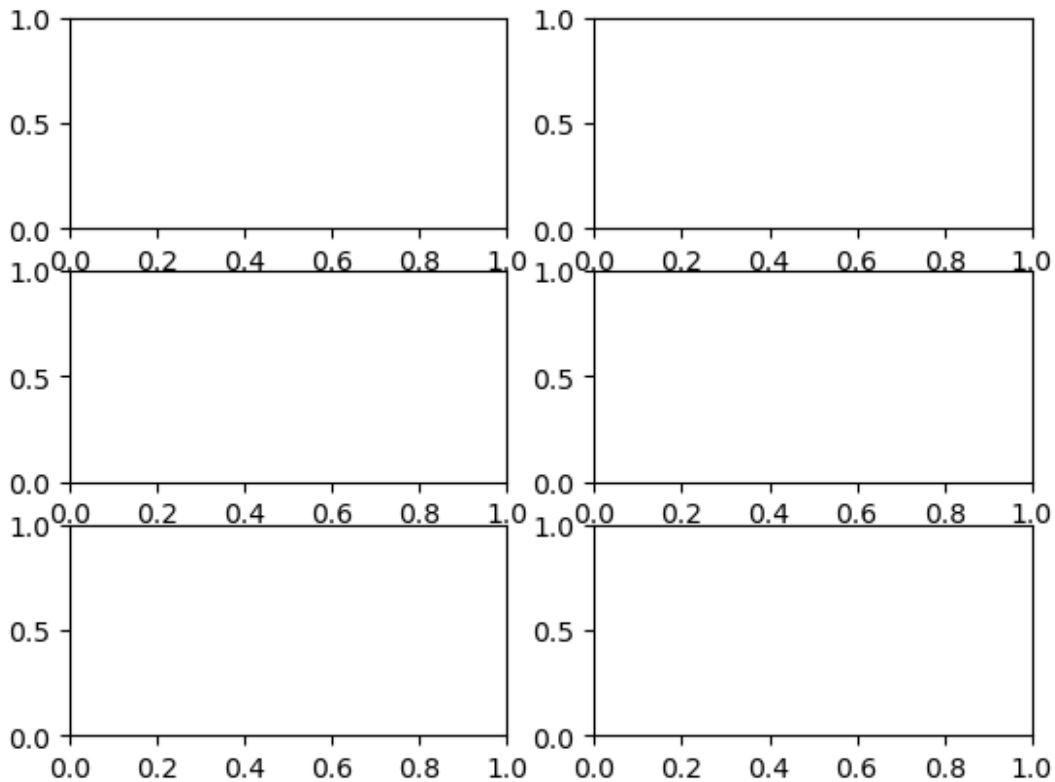
<Figure size 720x576 with 0 Axes>

b. Buat *figure* dengan isi *sub-figure* sebanyak 6 dengan 2 kolom dan 3 baris

Tip: gunakan `plt.subplots()`

```
# Inisialisasi figure dengan 6 subplot (2 kolom dan 3 baris)
fig, ax = plt.subplots(3,2)

# Tampilkan plot yang sudah dibuat
plt.show()
```



Contoh *output*:

Contoh output subfigure 3x2

c. Buat *figure* dengan ukuran 10x6 dan DPI sebesar 150 dengan isi *sub-figure* 2x1. Kemudian berikan judul utama pada *figure* dengan "Latihan Matplotlib DKA" dengan ukuran *font* 16, kemudian judul *sub-figure* pertama dengan "Figure Pertama", dan judul *sub-figure* kedua dengan "Figure Kedua". Setelah itu, berikan label pada sumbu *y* pada *sub-figure* pertama dengan "Nilai Pertama" dan *sub-figure* kedua dengan "Nilai Kedua".

Tip: Gunakan `plt.subplots()`.

```
# Inisialisasi figure dengan jumlah dan ukuran yang sesuai
fig, ax = plt.subplots(2,1, figsize=(10,8), dpi=72)

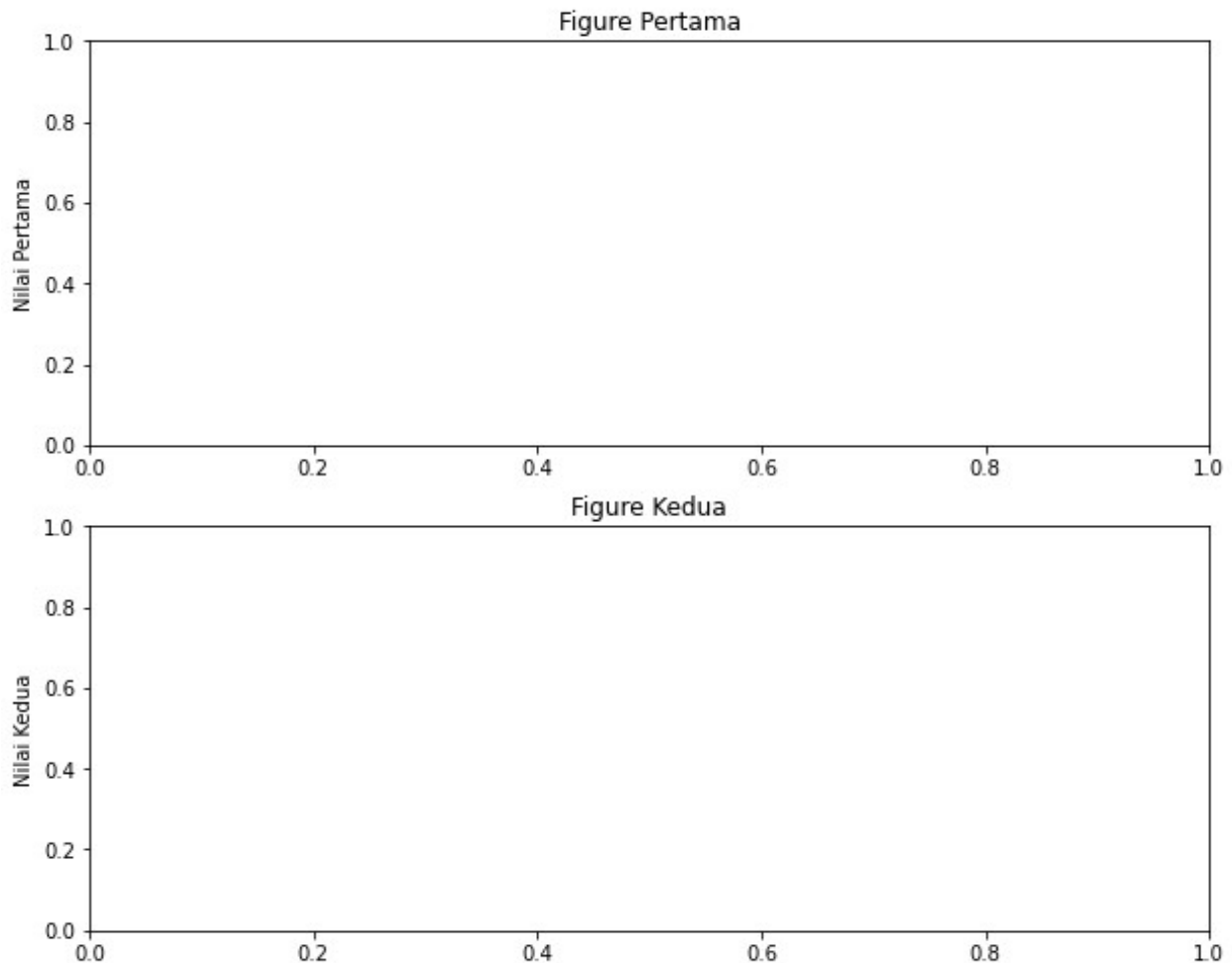
# Menambahkan judul dan label untuk subplot pertama
ax[0].set_title("Figure Pertama")
ax[0].set_ylabel("Nilai Pertama")

# Menambahkan judul dan label untuk subplot kedua
ax[1].set_title("Figure Kedua")
ax[1].set_ylabel("Nilai Kedua")

# Menambahkan judul dari figure
plt.suptitle("Latihan Matplotlib DKA", fontsize=16)
```

```
# Tampilkan plot yang sudah dibuat  
plt.show()
```

Latihan Matplotlib DKA



Contoh *output*:

Contoh figure dengan judul

2. Membuat Plot Pairwise pada Matplotlib (Bobot Nilai: 20)

a. Sebuah toko sedang memantau penjualan dua produk unggulan mereka, **Barang A** dan **Barang B**, selama lima hari terakhir. Manajer penjualan ingin mengetahui pola penjualan dari kedua produk ini dengan membandingkan kinerja keduanya dari hari ke hari. Buat visualisasi data penjualan dengan menggunakan *line chart* untuk mempermudah analisis tren.

Bantu manajer penjualan dengan membuat grafik *line chart* berdasarkan penjualan berikut:

- Menampilkan tren penjualan kedua **Barang A** dan **Barang B** dalam satu graf yang sama.
- Ukuran *figure* sebesar 7x5 dengan resolusi 100 dpi.

- Beri judul grafik dengan : "Tren Penjualan Barang A dan B Selama 5 Hari".
- Sumbu x mewakili hari ke berapa, dengan label "Tanggal".
- Sumbu y mewakili jumlah barang yang terjual, dengan label "Jumlah Terjual".
- Tampilkan *legend* di pojok kiri bawah untuk membedakan penjualan Barang A dan Barang B.

Data Penjualan: |Hari|Barang A|Barang B| |-----|-----|-----| |1|10|8| |2|7|8| |3|8|5| |4|12|7| |5|6|11|

Tip :

Gunakan `plt.figure()`

```
# Inisialisasi figure dengan ketentuan yang sesuai
fig = plt.figure(figsize=(7,5), dpi=100)

# List data penjualan
hari = [1,2,3,4,5]
barang_a = [10, 7, 8, 12, 6]
barang_b = [8, 8, 5, 7, 11]

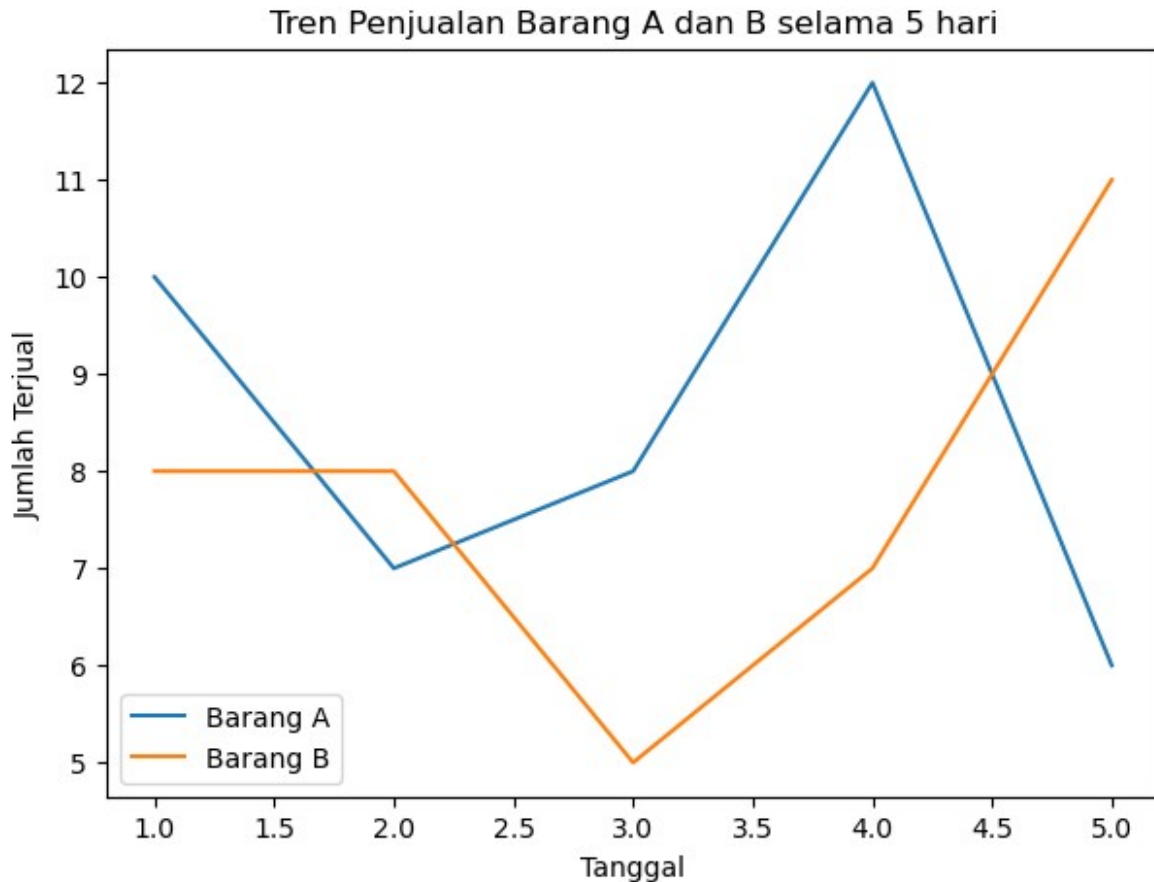
# Plot linechart Barang A pada figure
plt.plot(hari, barang_a, label="Barang A") # Beri label "Barang A"

# Plot linechart Barang B pada figure
plt.plot(hari, barang_b, label="Barang B") # Beri label "Barang B"

# Menambahkan judul, dan label untuk x dan y
plt.title("Tren Penjualan Barang A dan B selama 5 hari")
plt.xlabel("Tanggal")
plt.ylabel("Jumlah Terjual")

# Menambahkan legend dengan lokasi yang ditentukan
plt.legend(loc='lower left')

# Tampilkan linechart yang sudah dibuat
plt.show()
```



Contoh *output*:

Linechart Penjualan

b. Seorang ilmuwan pertanian sedang melakukan penelitian tentang dua spesies tanaman eksotis, **Tanaman A** dan **Tanaman B**, yang berasal dari daerah tropis. Peneliti ingin mengetahui bagaimana kedua spesies tanaman ini tumbuh selama delapan minggu di bawah kondisi lingkungan yang sama. Data pertumbuhan diukur setiap minggu dengan satuan sentimeter. Berdasarkan data yang dikumpulkan, peneliti ingin memvisualisasikan tren pertumbuhan ini dengan *scatter plot*.

Bantu ilmuwan tersebut dengan membuat **scatter plot** dari data berikut:

- **Tanaman A:** Titik data berwarna merah dengan ukuran titik 75 dan tingkat transparansi (*alpha*) sebesar 0.65
- **Tanaman B:** Titik data berwarna biru dengan ukuran titik 50 dan tingkan transparansi (*alpha*) sebesar 0.50
- Beri judul grafik dengan: "Pertumbuhan Tanaman Eksotis A dan B Selama 8 Minggu"
- Sumbu x menunjukkan minggu ke berapa, dengan label "Minggu"
- Sumbu y menunjukkan tinggi tanaman dalam cm, dengan label "Tinggi Tanaman (cm)"
- Tampilkan *legend* untuk membedakan titik data dari kedua tanaman tersebut.

Data pertumbuhan tanaman |Minggu|Tinggi Tanaman A (cm)|Tinggi Tanaman B (cm)| |---|---|---|
|1|10|12| |2|21|20| |3|25|26| |4|30|29| |5|34|33| |6|41|37| |7|47|41| |8|50|43|

```
# List data tanaman
minggu = [1, 2, 3, 4, 5, 6, 7, 8]
tanaman_a = [10, 21, 25, 30, 34, 41, 47, 50]
tanaman_b = [12, 20, 26, 29, 33, 37, 41, 43]
# Inisialisasi figure dengan ukuran yang sesuai
fig = plt.figure(figsize=(9,5), dpi = 100)# Tidak ada ketentuan ukuran figure

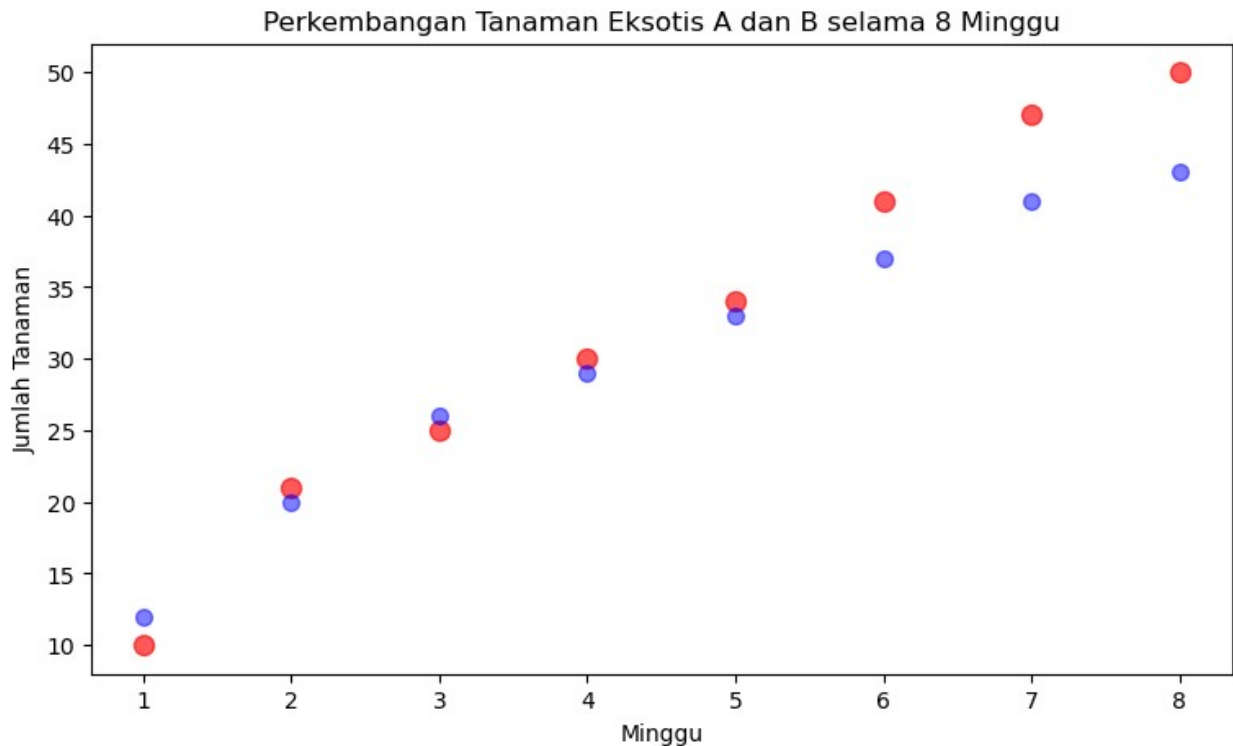
# Plot scatterplot untuk data Tanaman A
plt.scatter(minggu, tanaman_a, s = 75, c='red', alpha=0.65,
label="Tanaman A") # Beri Warna dan Label yang sesuai

# Plot scatterplot untuk data Tanaman B
plt.scatter(minggu, tanaman_b, s = 50, c='blue', alpha= 0.50, label =
"Tanaman B") # Beri Warna dan Label yang sesuai

# Menambahkan judul dan label untuk x dan y
plt.xlabel("Minggu")
plt.ylabel("Jumlah Tanaman")
plt.title("Perkembangan Tanaman Eksotis A dan B selama 8 Minggu")

# Menambahkan legend dan tentukan posisinya
plt.legend # Posisi dibebaskan. Pastikan tidak menghalangi data

# Tampilkan scatterplot yang sudah dibuat
plt.show()
```



Contoh *output*:

Contoh scatterplot tanaman

c. Sebuah toko buku ingin menganalisis total penjualan lima kategori buku terlaris selama sebulan terakhir. Kategori buku yang ingin dianalisis adalah **Fiksi**, **Non-Fiksi**, **Komik**, **Majalah**, dan **Sains**. Bantu toko buku tersebut dengan membuat *bar plot* untuk memvisualisasikan data tersebut. Gunakan atribut berikut pada *figure* visualisasi yang dibuat.

- Ukuran *figure* adalah 8x6 dengan 75 dpi.
- Judul grafik yang dibuat: "Penjualan Buku Berdasarkan Kategori Selama Sebulan"
- Label sumbu x: "Kategori Buku"
- Label sumbu y: "Jumlah Terjual"
- Gunakan warna yang berbeda untuk setiap bar

Data penjualan: |Kategori Buku|Jumlah Terjual| ---|---| |Fiksi|120| |Non-Fiksi|90| |Komik|150| |Majalah|80| |Sains|110|

Tip:

Gunakan `plt.subplots()`

```
# List data penjualan buku
kategori = ['Fiksi', 'Non-Fiksi', 'Komik', 'Majalah', 'Sains']
jumlah_terjual = [120, 90, 150, 80, 110]
```

```

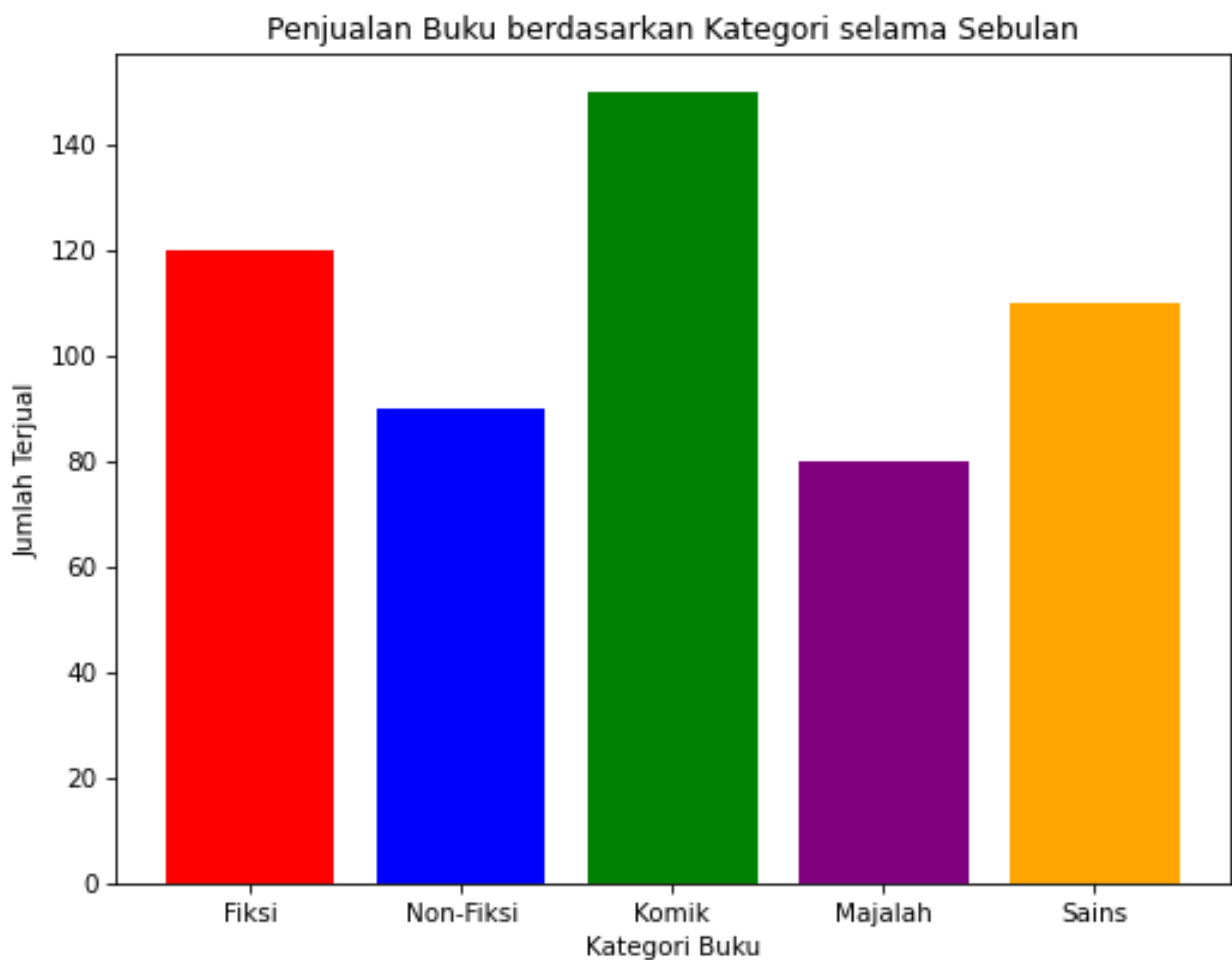
# Inisialisasi figure dengan ukuran dan dpi yang ditentukan
fig, ax= plt.subplots(figsize=(8,6), dpi = 75)

# List warna pada barchart
list_color = ['red', 'blue', 'green', 'purple', 'orange']
# Plot barchart untuk data penjualan buku
ax.bar(kategori, jumlah_terjual, color=list_color)

# Menambahkan judul dan label untuk x dan y
ax.set_title("Penjualan Buku berdasarkan Kategori selama Sebulan")
ax.set_xlabel("Kategori Buku")
ax.set_ylabel("Jumlah Terjual")

# Tampilkan barchart yang sudah dibuat
plt.show()

```



Contoh *output*:

Contoh barchart buku

3. Membuat Plot Distribusi Statistik pada Matplotlib (Bobot Nilai: 25)

a. Seorang peneliti ingin menganalisis distribusi waktu belajar harian (dalam jam) dari sekelompok mahasiswa selama satu minggu. Peneliti telah mengumpulkan data dari 50 mahasiswa, di mana setiap mahasiswa mencatat waktu belajar mereka setiap hari selama seminggu. Berdasarkan data yang dikumpulkan, peneliti ingin mengetahui distribusi waktu belajar tersebut dalam bentuk histogram.

Buatlah histogram untuk membantu peneliti memvisualisasikan distribusi waktu belajar mahasiswa. Gunakan atribut berikut:

- Ukuran *figure* adalah 10x5 dengan resolusi 100dpi.
- Judul grafik: "Distribusi Waktu Belajar Mahasiswa Selama Seminggu".
- Label sumbu x: "Waktu Belajar (Jam)".
- Label sumbu y: "Jumlah Mahasiswa".
- Karena nilai minimum pada data adalah 1 dan nilai maksimum adalah 7, maka jumlah *bins* menjadi 7 (1 bin merepresentasikan 1 jam).

Berikut list data dari 50 mahasiswa

```
[2, 3, 5, 4, 6, 2, 1, 3, 2, 4, 5, 2, 3, 1, 6, 3, 5, 4, 4, 2, 2, 3, 5, 1, 3, 5, 6, 7, 4, 5, 2, 3, 4, 1, 2, 3, 4, 6, 2, 5, 4, 3, 5, 2, 3, 6, 7, 5, 4, 3]
```

Tip:

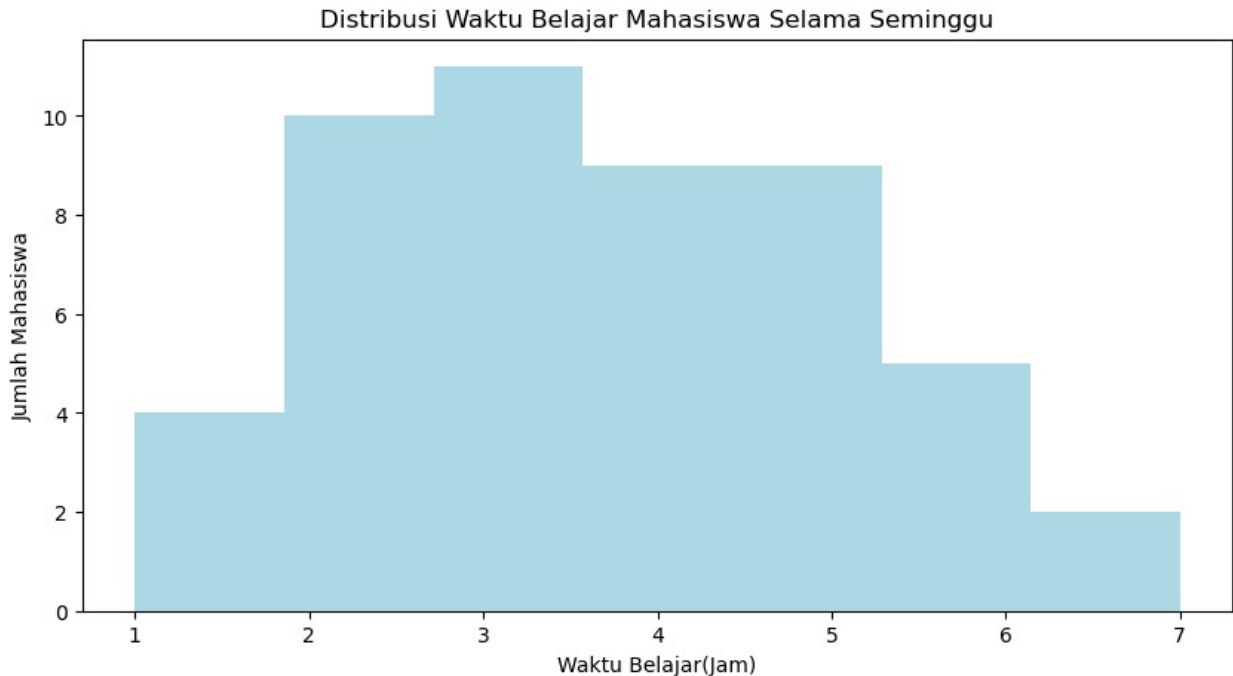
Gunakan `plt.subplots()`

```
# Data waktu belajar harian mahasiswa
data_waktu_belajar = [2, 3, 5, 4, 6, 2, 1, 3, 2, 4, 5, 2, 3, 1, 6, 3, 5, 4, 4, 2, 2, 3, 5, 1, 3, 5, 6, 7, 4, 5, 2, 3, 4, 1, 2, 3, 4, 6, 2, 5, 4, 3, 5, 2, 3, 6, 7, 5, 4, 3]

# Inisialisasi figure dengan ukuran dan dpi yang ditentukan
fig, ax = plt.subplots(figsize=(10,5), dpi=100)
# Plot histogram waktu belajar
ax.hist(data_waktu_belajar, bins=7, color='lightblue') # Tambahkan warna bebas

# Menambahkan judul dan label untuk x dan y
ax.set_title("Distribusi Waktu Belajar Mahasiswa Selama Seminggu")
ax.set_xlabel("Waktu Belajar(Jam)")
ax.set_ylabel("Jumlah Mahasiswa")

# Tampilkan histogram yang sudah dibuat
plt.show()
```



Contoh *output*:

Contoh histogram belajar

b. Sebuah perusahaan *e-commerce* ingin menganalisis distribusi kategori produk yang paling sering dibeli oleh pelanggan selama bulan terakhir. Data yang dikumpulkan menunjukkan bahwa terdapat lima kategori produk yang paling sering dibeli: **Elektronik**, **Pakaian**, **Peralatan Rumah Tangga**, **Buku**, dan **Kosmetik**. Perusahaan ingin melihat presentase kontribusi dari setiap kategori produk dalam bentuk *pie chart*.

Bantu tim analisis perusahaan dengan membuat *pie chart* untuk memvisualisasikan distribusi pembelian produk berdasarkan kategori berikut.

- **Elektronik**: 1500 pembelian.
- **Pakaian**: 1200 pembelian.
- **Peralatan Rumah Tangga**: 900 pembelian.
- **Buku**: 800 pembelian.
- **Kosmetik**: 600 pembelian.

Gunakan atribut berikut:

- Ukuran *figure* adalah 8x8 dengan resolusi 100 dpi.
- Judul grafik: "Distribusi Pembelian Produk Berdasarkan Kategori".
- Tambahkan persentase pada setiap bagian *pie chart*.
- Pisahkan potongan *pie chart* untuk kategori dengan jumlah pembelian tertinggi.
- Gunakan warna yang berbeda untuk setiap kategori.

```
# List data kategori produk dan jumlah pembelian
kategori = ['Elektronik', 'Pakaian', 'Peralatan Rumah Tangga', 'Buku',
```

```
'Kosmetik']
jumlah_pembelian = [1500, 1200, 900, 800, 600]

# Inisialisasi figure dengan ukuran dan resolusi yang sudah ditentukan
fig, ax = plt.subplots(figsize=(8,8), dpi=100)

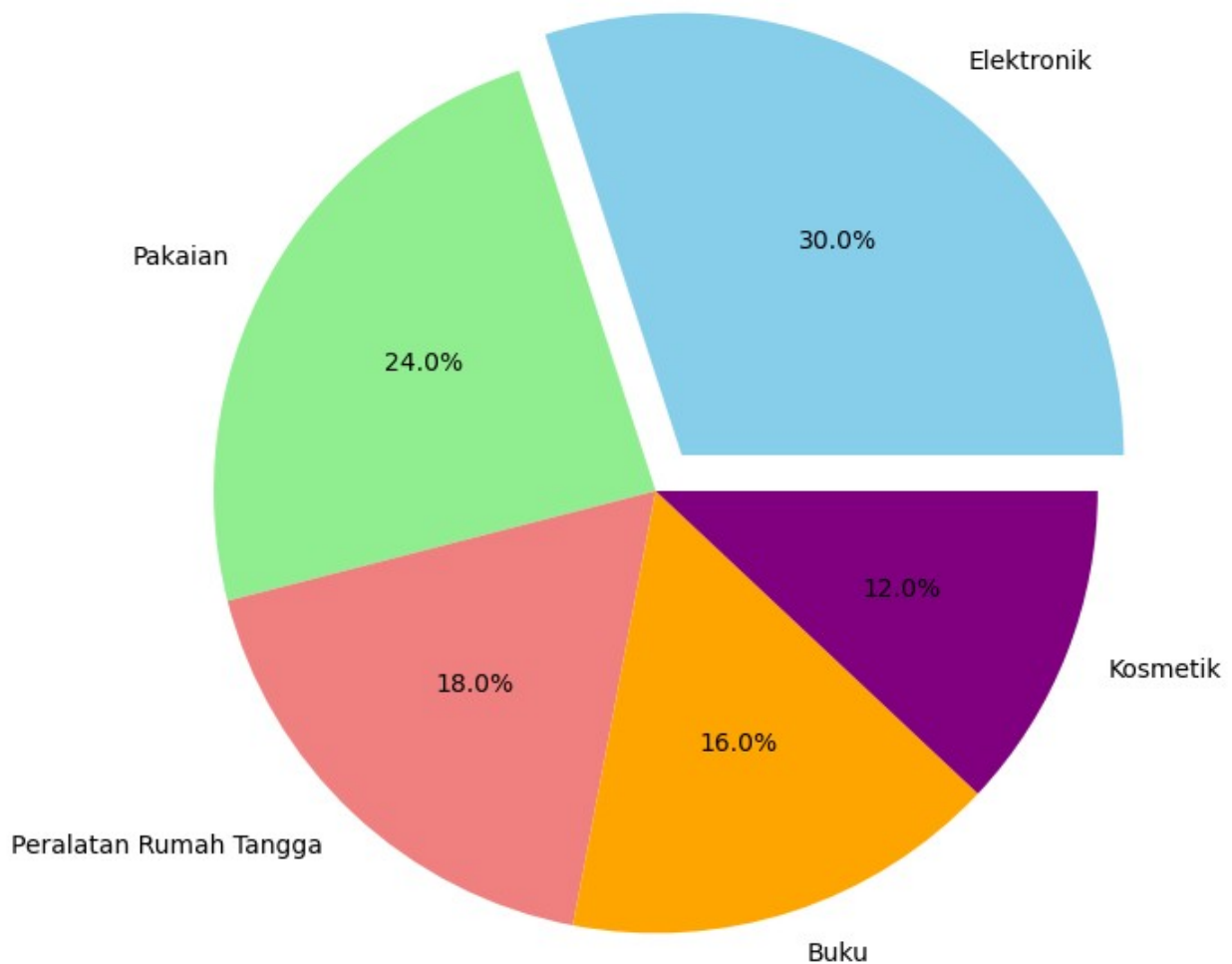
# Membuat explode untuk memisahkan kategori dengan pembelian tertinggi
explode = [0.1, 0, 0, 0, 0]

# List warna pada barchart
color_list = ['skyblue', 'lightgreen', 'lightcoral', 'orange',
'purple']
# Plot piechart untuk data prnjualan produk
ax.pie(jumlah_pembelian, labels=kategori, autopct='%1.1f%%',
explode=explode, colors=color_list)

# Menambahkan judul
ax.set_title("Diagram Pie Penjualan Produk")

# Tampilan piechart yang sudah dibuat
plt.show()
```

Diagram Pie Penjualan Produk



c. Sebuah sekolah sedang menganalisis nilai ujian akhir dari tiga kelas yang berbeda: **Kelas A**, **Kelas B**, **Kelas C**. Setiap kelas diikuti oleh sejumlah siswa, dan mereka telah menyelesaikan ujian akhir dalam mata pelajaran Matematika. Sekolah ingin memvisualisasikan sebaran nilai ujian dari masing-masing kelas untuk membandingkan performa antar kelas tersebut.

Bantu pihak sekolah dengan membuat *boxplot* yang menampilkan distribusi nilai ujian dari ketiga kelas tersebut. Data nilai ujian dari masing-masing kelas adalah sebagai berikut:

- **Kelas A:** [85, 78, 92, 88, 76, 90, 85, 91, 89, 95]
- **Kelas B:** [72, 75, 78, 74, 73, 80, 76, 77, 79, 81]
- **Kelas C:** [88, 90, 85, 93, 87, 92, 91, 86, 84, 94]

Gunakan atribut berikut dalam visualisasi:

- Ukuran *figure* adalah 8x6 dengan resolusi 100 dpi.

- Judul grafik: "Distribusi Nilai Ujian Akhir Matematika di Kelas A, B, dan C".
- Label sumbu x: "Kelas".
- Label sumbu y: "Nilai Ujian".
- Gunakan warna yang berbeda untuk setiap *boxplot*.

```
# List data nilai pada ketiga kelas
nilai_kelas_a = [85, 78, 92, 88, 76, 90, 85, 91, 89, 95]
nilai_kelas_b = [72, 75, 78, 74, 73, 80, 76, 77, 79, 81]
nilai_kelas_c = [88, 90, 85, 93, 87, 92, 91, 86, 84, 94]

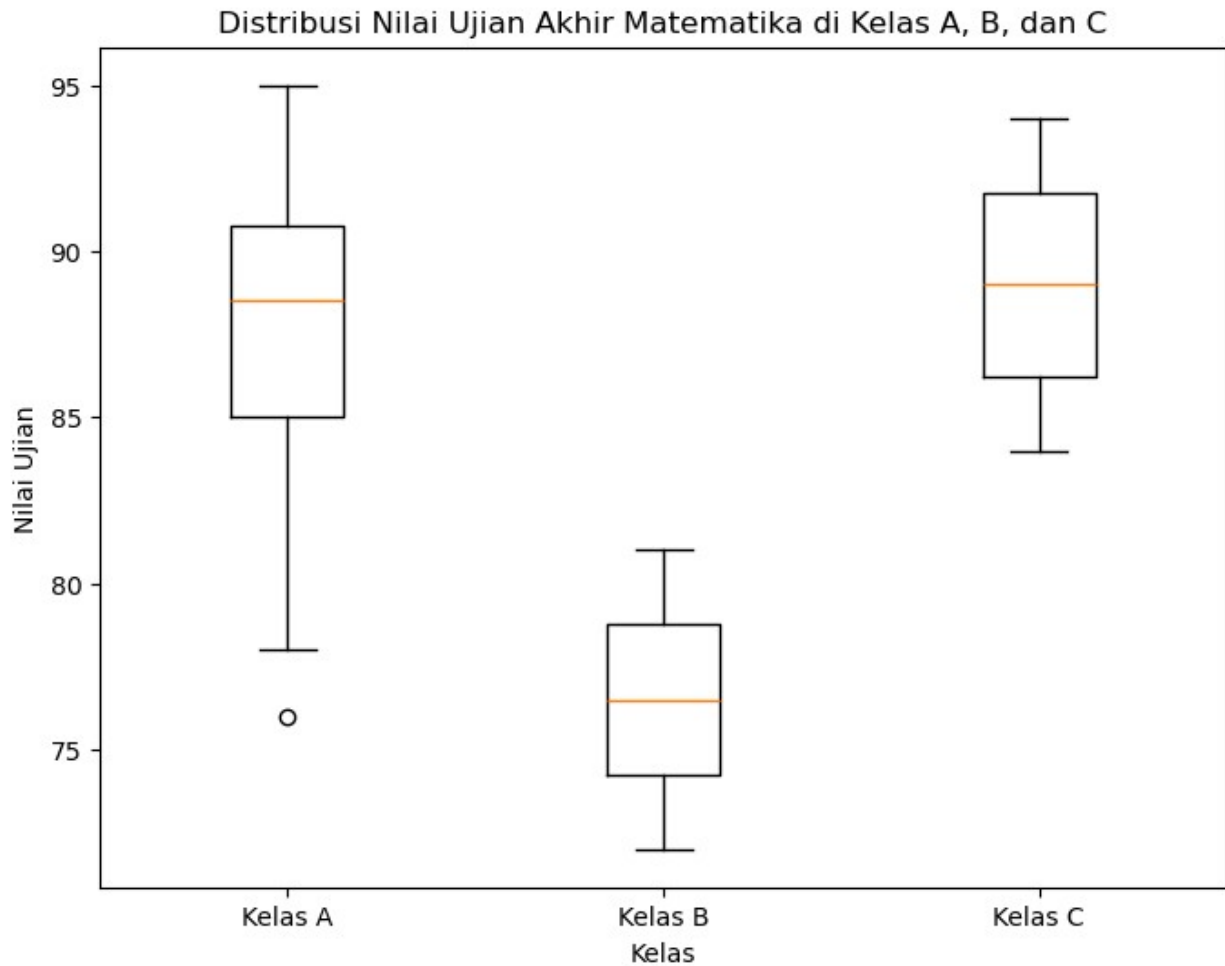
# Menyiapkan data untuk boxplot
data = [nilai_kelas_a, nilai_kelas_b, nilai_kelas_c]
labels = ['Kelas A', 'Kelas B', 'Kelas C']

# Inisialisasi figure dengan ukuran dan dpi yang sudah ditentukan
fig, ax = plt.subplots(figsize=(8,6), dpi=100)

# Plot boxplot untuk data nilai ujian
ax.boxplot(data, vert=True, labels=labels)

# Menambahkan judul dan label untuk x dan y
ax.set_title("Distribusi Nilai Ujian Akhir Matematika di Kelas A, B, dan C")
ax.set_xlabel("Kelas")
ax.set_ylabel("Nilai Ujian")
# Menampilkan plot
plt.show()
```

```
C:\Users\Asus\AppData\Local\Temp\ipykernel_6096\1453669773.py:14:
MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has
been renamed 'tick_labels' since Matplotlib 3.9; support for the old
name will be dropped in 3.11.
    ax.boxplot(data, vert=True, labels=labels)
```



Contoh *output*:

Contoh boxplot nilai

4. Soal Mandiri (Bobot Nilai: 40)

Sebuah universitas sedang melakukan analisis performa akademik mahasiswa di empat jurusan: **Informatika**, **Teknologi Informasi**, **Rekayasa Perangkat Lunak**, dan **Sains Data**. Universitas ingin menganalisis beberapa aspek berikut :

1. **Distribusi Nilai Akhir Mahasiswa** Setiap jurusan mengumpulkan nilai akhir mahasiswa dalam tiga mata kuliah inti selama satu semester. Universitas ingin melihat distribusi nilai dari masing-masing jurusan.
 - **Informatika**: [85, 78, 92, 88, 76, 90, 85, 91, 89, 95].
 - **Teknologi Informasi**: [72, 75, 78, 74, 73, 80, 76, 77, 79, 81].
 - **Rekayasa Perangkat Lunak**: [88, 90, 85, 93, 87, 92, 91, 86, 84, 94].
 - **Sains Data**: [80, 82, 78, 85, 84, 83, 82, 88, 86, 87].

Buatlah *boxplot* untuk memvisualisasikan distribusi nilai akhir dari keempat jurusan tersebut.

2. **Rata-rata Kehadiran Mahasiswa** Universitas juga ingin mengetahui rata-rata kehadiran mahasiswa per jurusan selama semester ini. Data kehadiran dalam persen adalah sebagai berikut:

- **Informatika:** 85 %.
- **Teknologi Informasi:** 90 %.
- **Rekayasa Perangkat Lunak:** 88 %.
- **Sains Data:** 87 %.

Buatlah *barplot* untuk memvisualisasikan rata-rata kehadiran mahasiswa per jurusan.

3. **Proporsi Mahasiswa Berdasarkan Jurusan** Universitas ingin melihat proporsi jumlah mahasiswa di setiap jurusan untuk memahami distribusi populasi. Data jumlah mahasiswa per jurusan adalah sebagai berikut:

- **Informatika:** 300 Mahasiswa.
- **Teknologi Informasi:** 250 Mahasiswa.
- **Rekayasa Perangkat Lunak:** 200 Mahasiswa.
- **Sains Data:** 150 Mahasiswa.

4. **Distribusi Jam Belajar Mahasiswa** Universitas juga melakukan survei terkait jam belajar harian dari 50 mahasiswa secara acak. Data waktu belajar harian (dalam jam) yang diukur dari mahasiswa adalah sebagai berikut

```
[2, 3, 5, 4, 6, 2, 1, 3, 2, 4, 5, 2, 3, 1, 6, 3, 5, 4, 4, 2, 2, 3, 5, 1, 3, 5, 6, 7, 4, 5, 2, 3, 4, 1, 2, 3, 4, 6, 2, 5, 4, 3, 5, 2, 3, 6, 7, 5, 4, 3]
```

Buatlah histogram untuk menampilkan distribusi jam belajar harian dari mahasiswa.

Gunakan atribut berikut pada setiap visualisasi:

- Ukuran *figure* untuk semua grafik adalah 8x6 dengan resolusi 100 dpi.
- Beri judul yang sesuai untuk setiap grafik.
- Berikan label sumbu pada *barplot*, *boxplot*, dan *histogram*.
- Tambahkan presentase pada *pie chart*.

```
# 1. Data nilai akhir mahasiswa
nilai_informatika = [85, 78, 92, 88, 76, 90, 85, 91, 89, 95]
nilai_teknologi_informasi = [72, 75, 78, 74, 73, 80, 76, 77, 79, 81]
nilai_rekayasa_perangkat_lunak = [88, 90, 85, 93, 87, 92, 91, 86, 84, 94]
nilai_sains_data = [80, 82, 78, 85, 84, 83, 82, 88, 86, 87]

fig = plt.figure(figsize=(8,6), dpi=100)

plt.boxplot([nilai_informatika, nilai_teknologi_informasi,
nilai_rekayasa_perangkat_lunak, nilai_sains_data], patch_artist=True,
```

```

labels=['Informatika', 'Teknologi Informasi', 'RPL', 'Sains Data'])
plt.xlabel("Jurusan")
plt.ylabel("Nilai Akhir")

plt.title("Distribusi Nilai Akhir Mahasiswa di Empat Jurusan")

plt.show()

# 2. Data rata-rata kehadiran mahasiswa
jurusan = ['Informatika', 'Teknologi Informasi', 'RPL', 'Sains Data']
kehadiran = [85, 90, 88, 87]

fig = plt.figure(figsize=(8,6), dpi=100)
color_list = ['blue', 'green', 'orange', 'purple']
plt.bar(jurusan, kehadiran, color=color_list)
plt.xlabel("Jurusan")
plt.ylabel("Kehadiran (%)")

plt.title("Rata-rata Kehadiran Mahasiswa per Jurusan")

plt.show()

# 3. Data jumlah mahasiswa berdasarkan jurusan
jurusan = ['Informatika', 'Teknologi Informasi', 'RPL', 'Sains Data']
mahasiswa = [300, 250, 200, 150]

fig = plt.figure(figsize=(8,6), dpi=100)
color_list = ["lightblue", "lightgreen", "lightcoral", "lightyellow"]
explode = [0.1, 0, 0, 0]

plt.pie(mahasiswa, labels=jurusan, autopct='%1.1f%%',
        colors=color_list, explode=explode)
plt.title("Jumlah Mahasiswa per Jurusan")

plt.show()

# 4. Data jam belajar harian mahasiswa
data_waktu_belajar = [2, 3, 5, 4, 6, 2, 1, 3, 2, 4, 5, 2, 3, 1, 6, 3,
5, 4, 4, 2,
                        2, 3, 5, 1, 3, 5, 6, 7, 4, 5, 2, 3, 4, 1, 2, 3,
4, 6, 2, 5,
                        4, 3, 5, 2, 3, 6, 7, 5, 4, 3]

fig = plt.figure(figsize=(8,6), dpi=100)

plt.hist(data_waktu_belajar, bins=7, color='blue', alpha = 0.4)
plt.xlabel("Jam Belajar")
plt.ylabel("Jumlah Mahasiswa")

plt.title("Distribusi Waktu Belajar Mahasiswa")

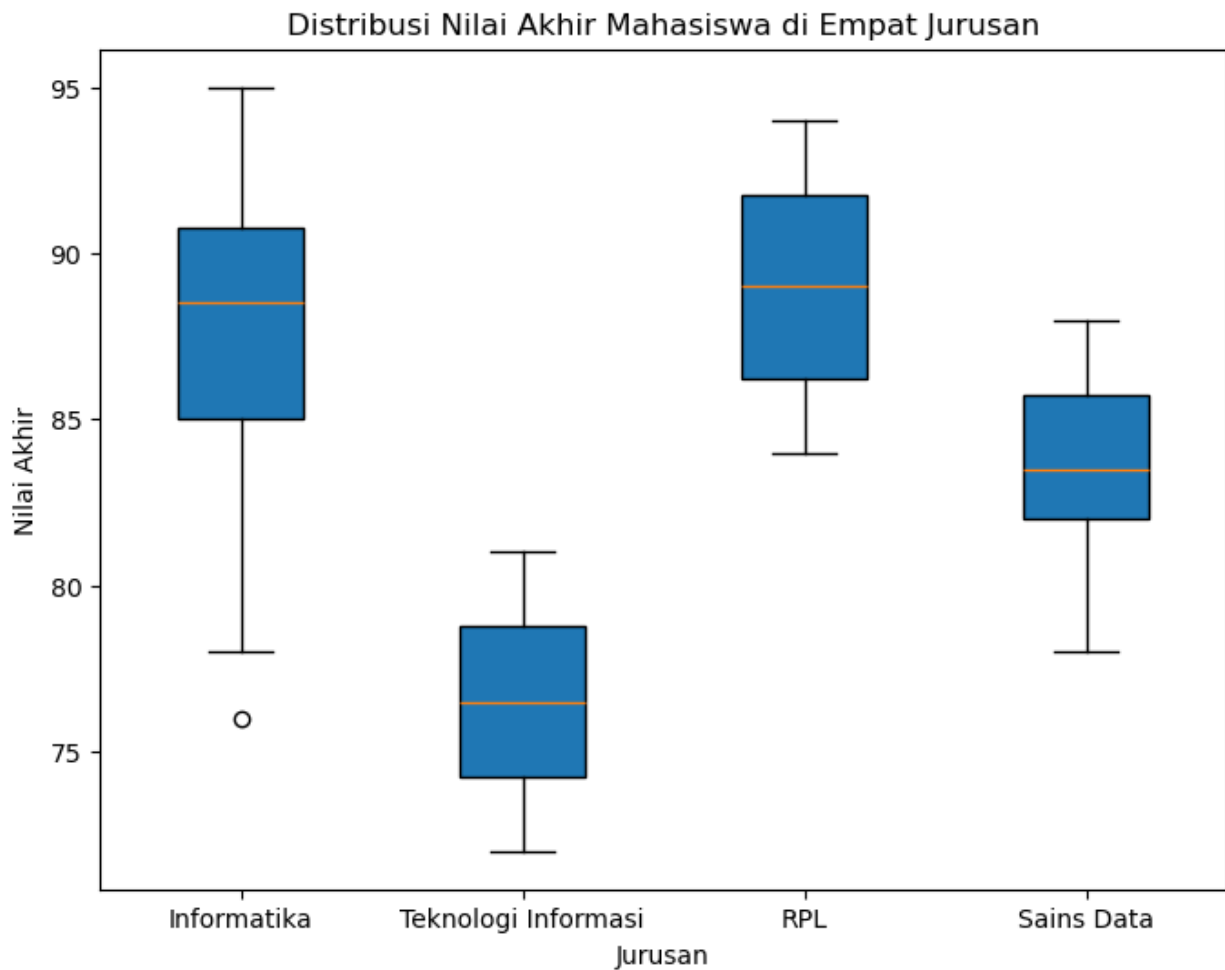
```



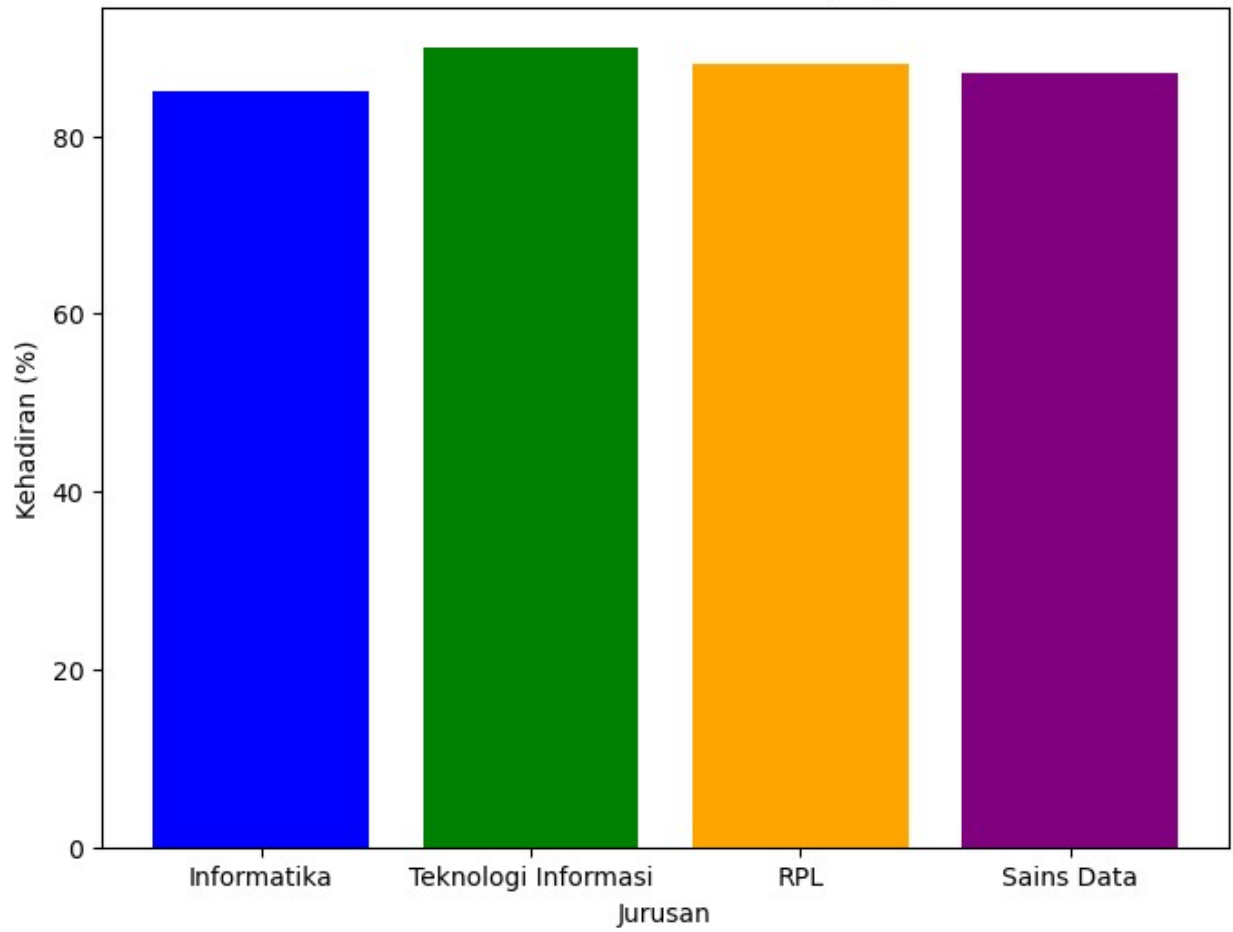
```
plt.show()
```

```
C:\Users\Asus\AppData\Local\Temp\ipykernel_6096\1746826003.py:9:  
MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has  
been renamed 'tick_labels' since Matplotlib 3.9; support for the old  
name will be dropped in 3.11.
```

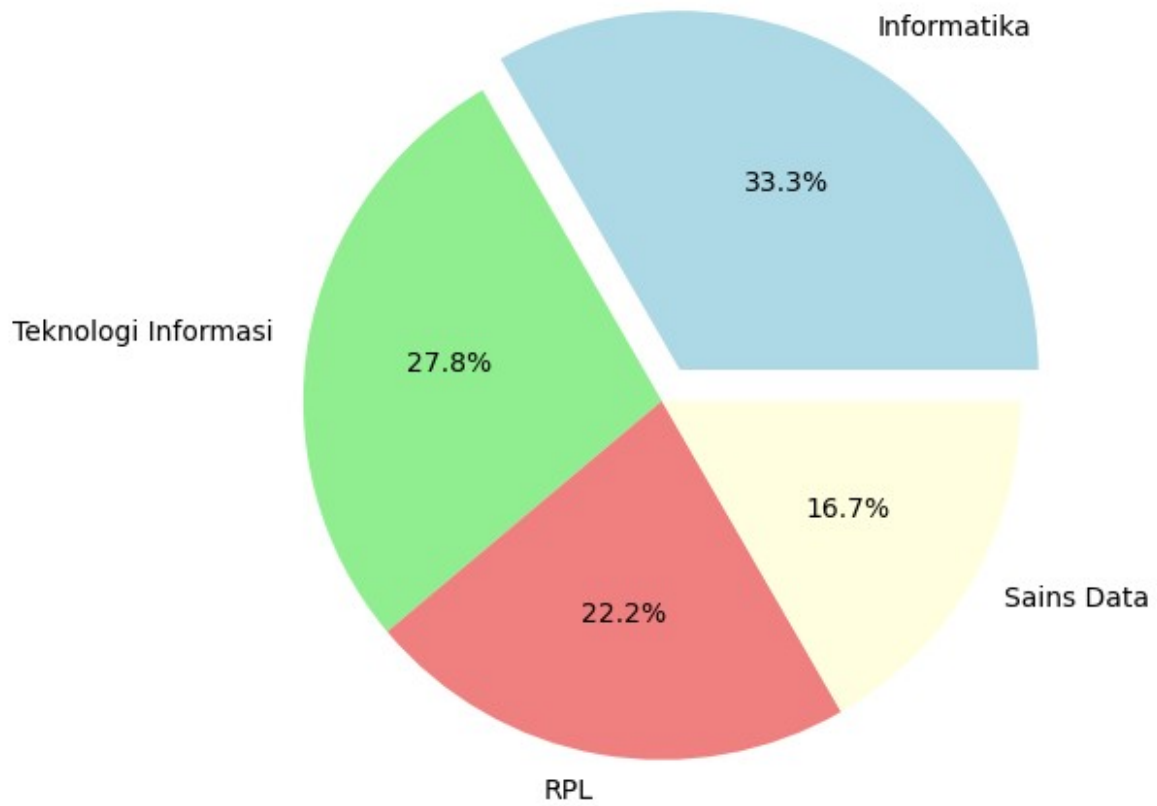
```
plt.boxplot([nilai_informatika, nilai_teknologi_informasi,  
nilai_rekayasa_perangkat_lunak, nilai_sains_data], patch_artist=True,  
labels=['Informatika', 'Teknologi Informasi', 'RPL', 'Sains Data'])
```

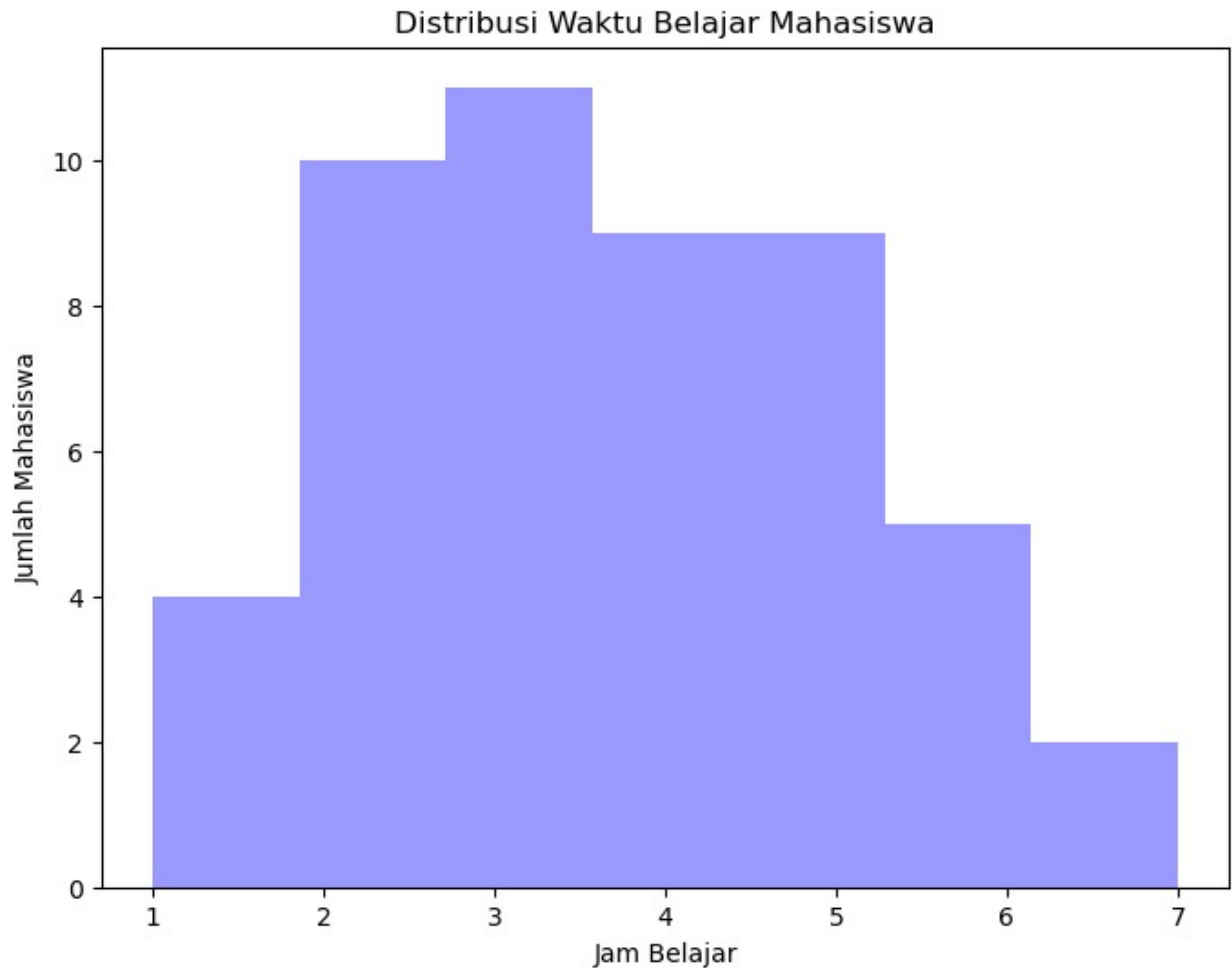


Rata-rata Kehadiran Mahasiswa per Jurusan



Jumlah Mahasiswa per Jurusan





Contoh *output*:

Boxplot nilai

Barchart kehadiran

Pie Chart Proporsi Mahasiswa

Histogram Distribusi Belajar