

# **Laporan UTS *Take-Home* Mata Kuliah Aplikasi Data Scientist**

**Dosen Pengampu: Ledy Elsera Astrianty, S.Kom., M.Kom.**



**Disusun oleh:**

**Lathif Ramadhan (5231811022)**

**PROGRAM STUDI SAINS DATA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS TEKNOLOGI YOGYAKARTA  
YOGYAKARTA**

**2025**

# Pendahuluan

Laporan ini disusun sebagai pemenuhan tugas Ujian Tengah Semester (UTS) mata kuliah Aplikasi Data Scientist, yang diampu oleh Ibu Ledy Elsera Astrianty, S.Kom., M.Kom. Tugas UTS ini melibatkan analisis dan visualisasi data untuk menjawab serangkaian studi kasus yang diberikan, yang mencakup penerapan berbagai konsep dan teknik ilmu data.

Fokus utama dari laporan ini adalah demonstrasi kemampuan dalam menggunakan *library* Python seperti Pandas, NumPy, Matplotlib, dan Seaborn untuk melakukan manipulasi data, visualisasi, dan analisis statistik. Setiap studi kasus dalam soal ujian menuntut pendekatan yang berbeda dalam pemilihan jenis grafik yang tepat dan interpretasi hasil analisis yang relevan.

Analisis ini diimplementasikan dalam lingkungan Google Colaboratory, yang memungkinkan penulisan dan eksekusi kode Python secara interaktif melalui *browser*. Untuk memastikan transparansi dan *reproducibility*, kode sumber Python yang digunakan dalam analisis ini tersedia di repositori GitHub pribadi saya berikut:

- [https://github.com/LatiefDataVisionary/data-science-application-college-task/blob/main/mid\\_test/UTS\\_data\\_science\\_application\\_college\\_subject.ipynb](https://github.com/LatiefDataVisionary/data-science-application-college-task/blob/main/mid_test/UTS_data_science_application_college_subject.ipynb)

Atau bisa di akses dengan QR Code berikut:



Laporan ini mencakup analisis untuk menjawab soal-soal UTS berikut:

1. **Soal 1 (Harga Bitcoin):** Analisis perbandingan harga Bitcoin antara tahun 2018 dan 2019 untuk menentukan tahun dengan potensi pengembalian investasi yang lebih baik.
2. **Soal 2 (Lima Jenis Permen):** Perhitungan dan visualisasi peluang pengambilan permen Snickers dari sekantong permen campuran.
3. **Soal 3 (Menu Makanan Penutup):** Identifikasi tiga menu makanan penutup yang kurang populer untuk direkomendasikan penghapusannya dari menu restoran.
4. **Soal 4 (Penggunaan CPU):** Visualisasi dan analisis pola penggunaan CPU komputer pekerja selama seminggu.
5. **Soal 5 (Pertumbuhan Jamur):** Estimasi lokasi pusat pertumbuhan jamur berdasarkan data koordinat.

Setiap bagian dari laporan ini akan menyajikan langkah-langkah implementasi kode Python, visualisasi data yang relevan, dan penjelasan mendalam terkait analisis untuk menjawab pertanyaan-pertanyaan yang diajukan dalam soal UTS.



# Daftar Isi

<b>Pendahuluan .....</b>	<b>II</b>
<b>Daftar Isi .....</b>	<b>IV</b>
<b>1. Soal No. 1: Harga Bitcoin .....</b>	<b>1</b>
1.1. Langkah-langkah Mengerjakan dengan Python .....	1
1.1.1 Import Libraries .....	1
1.1.2. Load Dataset .....	1
1.1.3. Data Preprocessing .....	1
1.1.4. Visualisasi Data: Line Plot .....	2
1.1.4.1. Grafik return/persentase perubahan mingguan .....	2
1.1.5. Analisis dan Interpretasi .....	3
1.2. Analisis Pengembalian Bitcoin: 2018 atau 2019? .....	4
1.3. Analisis Pemilihan Chart .....	5
1.4. Analisis Keuntungan Bitcoin: 2018 atau 2019? .....	5
<b>Soal No. 2: Lima Jenis Permen .....</b>	<b>6</b>
2.1 Langkah-langkah Mengerjakan dengan Python .....	6
2.1.1 Import Libraries .....	6
2.1.2. Load Dataset .....	6
2.1.3. Data Preprocessing .....	6
2.1.4. Visualisasi Data: Pie Chart .....	6
2.1.5. Analisis dan Interpretasi .....	7
2.2. Analisis Pemilihan Chart .....	8
2.3. Peluang Memilih Snickers .....	9
<b>3. Soal No. 3: Menu Makanan Penutup .....</b>	<b>10</b>
3.1 Langkah-langkah Mengerjakan dengan Python .....	10
3.1.1 Import Libraries .....	10
3.1.2. Load Dataset .....	10
3.1.3. Data Preprocessing .....	10
3.1.4. Visualisasi Data: Bar Chart .....	10
3.1.5. Analisis dan Interpretasi .....	11
3.2. Analisis Pemilihan Chart .....	11
3.3. Tiga Menu Penutup yang Perlu Dihapus .....	12
<b>Soal No. 4: Penggunaan CPU .....</b>	<b>14</b>
4.1 Langkah-langkah Mengerjakan dengan Python .....	14
4.1.1 Import Libraries .....	14
4.1.2. Load Dataset .....	14
4.1.3. Data Preprocessing .....	14
4.1.4. Visualisasi Data: Heat Map .....	15

4.2. Analisis Pemilihan Chart .....	16
4.3. Waktu Para Pekerja Makan Siang .....	17
4.4. Apakah Para Pekerja Bekerja di Akhir Pekan .....	18
4.5. Hari Para Pekerja Mulai Bekerja di Malam Hari.....	18
<b>Soal No. 5: Pertumbuhan Jamur .....</b>	<b>19</b>
5.1 Langkah-langkah Mengerjakan dengan Python .....	19
5.1.1 Import Libraries.....	19
5.1.2. Load Dataset .....	19
5.1.3. Data Preprocessing .....	19
5.1.4. Visualisasi Data: Scatter Plot & Analisis serta Interpretasi.....	20
5.2. Analisis Pemilihan Chart .....	20
5.3. Koordinat (x, y) Pusat Pertumbuhan Jamur.....	22

# 1. Soal No. 1: Harga Bitcoin

(Poin: 20) Kita memiliki daftar harga Bitcoin yang dicatat setiap akhir minggu (Minggu) di 2018 dan 2019. Buat visualisasi yang memungkinkan Anda menjawab pertanyaan:

- Tahun apa, 2018 atau 2019, yang cenderung memberikan pengembalian yang lebih baik bagi pemegang Bitcoin?
- Chart apa yang anda pilih untuk problem diatas dan mengapa anda memilih chart tersebut?
- Tahun berapa pemegang bitcoin memiliki keuntungan yang lebih banyak?

## 1.1. Langkah-langkah Mengerjakan dengan Python

### 1.1.1 Import Libraries

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

### 1.1.2. Load Dataset

```
prices = [14292.2, 12858.9, 11467.5, 9241.1, 8559.6, 11073.5, 9704.3, 11402.3,
8762.0, 7874.9, 8547.4, 6938.2, 6905.7, 8004.4, 8923.1, 9352.4,
9853.5, 8459.5, 8245.1, 7361.3, 7646.6, 7515.8, 6505.8, 6167.3,
6398.9, 6765.5, 6254.8, 7408.7, 8234.1, 7014.3, 6231.6, 6379.1,
6734.8, 7189.6, 6184.3, 6519.0, 6729.6, 6603.9, 6596.3, 6321.7,
6572.2, 6494.2, 6386.2, 6427.1, 5621.8, 3920.4, 4196.2, 3430.4,
3228.7, 3964.4, 3706.8, 3785.4, # Akhir tahun 2018
3597.2, 3677.8, 3570.9, 3502.5, # Awal tahun 2019
3661.4, 3616.8, 4120.4, 3823.1, 3944.3, 4006.4, 4002.5, 4111.8,
5046.2, 5051.8, 5290.2, 5265.9, 5830.9, 7190.3, 7262.6, 8027.4,
8545.7, 7901.4, 8812.5, 10721.7, 11906.5, 11268.0, 11364.9, 10826.7,
9492.1, 10815.7, 11314.5, 10218.1, 10131.0, 9594.4, 10461.1, 10337.3,
9993.0, 8208.5, 8127.3, 8304.4, 7957.3, 9230.6, 9300.6, 8804.5,
8497.3, 7324.1, 7546.6, 7510.9, 7080.8, 7156.2, 7321.5, 7376.8]
```

### 1.1.3. Data Preprocessing

#### 1.1.3.1. Membagi data menjadi 2018 dan 2019 (masing masing 52 minggu)

```
prices_2018 = prices[:52]
prices_2019 = prices[52:]
```

#### 1.1.3.2. Menghitung ROI (Return on Investment) untuk analisis kuantitatif

```
roi_2018 = ((prices_2018[-1] - prices_2018[0]) / prices_2018[0]) * 100
roi_2019 = ((prices_2019[-1] - prices_2019[0]) / prices_2019[0]) * 100
```

#### 1.1.3.3. Membuat urutan minggu (1-52) untuk sumbu x

```
weeks = list(range(1, 53))
```

#### 1.1.3.4. Menghitung rata-rata harga

```
avg_price_2018 = np.mean(prices_2018)
avg_price_2019 = np.mean(prices_2019)
```

### 1.1.4. Visualisasi Data: Line Plot

```
# Visualisasi dengan styling dengan seaborn
# Use the 'whitegrid' style from seaborn
sns.set_style("whitegrid")
fig, ax = plt.subplots(figsize=(12, 6))

# Plotting harga Bitcoin untuk 2018 dan 2019 dengan marker titik penting
ax.plot(weeks, prices_2018, marker='o', linestyle='-',
        color='#1f77b4', markersize=5, linewidth=2, label='2018')
ax.plot(weeks, prices_2019, marker='o', linestyle='-',
        color='#ff7f0e', markersize=5, linewidth=2, label='2019')

# Menambahkan anotasi ROI
ax.annotate(f'ROI 2018: {roi_2018:.1f}%', xy=(52, prices_2018[-1]),
           xytext=(40, 14000), arrowprops=dict(arrowstyle='->'), fontweight='bold',
           fontsize=10)
ax.annotate(f'ROI 2019: {roi_2019:.1f}%', xy=(52, prices_2019[-1]),
           xytext=(40, 13000), arrowprops=dict(arrowstyle='->'), fontweight='bold',
           fontsize=10)

# Formatting sumbu dan grid
ax.set_title('Perbandingan Harga Bitcoin Mingguan (2018 vs 2019)',
            fontsize=20, fontweight='bold', pad=20)
ax.set_xlabel('Minggu ke-', fontsize=15, fontweight='bold')
ax.set_ylabel('Harga (USD)', fontsize=15, fontweight='bold')
ax.set_xticks(range(1, 53, 4))
ax.grid(True, linestyle='--', alpha=0.7)

# Formatting sumbu Y sebagai mata uang
ax.yaxis.set_major_formatter('${x:,.0f}')

# Menambahkan legenda dan layout
legend = ax.legend(title='Tahun:', fontsize=10, title_fontsize=13,
                  loc='upper left', frameon=True)
# Mengakses objek legend title dan mengubah fontweight menjadi bold
legend.get_title().set_fontweight('bold')
plt.tight_layout()

# Menampilkan plot
plt.show()
```

*(Ouput kode berupa visualisasi Line Chart yang saya tampilkan di 1.2. Analisis Pemilihan Chart*

#### 1.1.4.1. Grafik return/persentase perubahan mingguan

```
returns_2018 = pd.Series(prices_2018).pct_change().fillna(0)
returns_2019 = pd.Series(prices_2019).pct_change().fillna(0)
```

```

fig, ax = plt.subplots(figsize=(10, 4))
ax.plot(weeks, returns_2018 * 100, label='2018', color='dodgerblue')
ax.plot(weeks, returns_2019 * 100, label='2019', color='orangered')
ax.set_title('Persentase Perubahan Harga Mingguan Bitcoin', fontsize=20,
fontweight='bold')
ax.set_xlabel('Minggu ke-', fontsize=14, fontweight='bold')
ax.set_ylabel('Return (%)', fontsize=14, fontweight='bold')
ax.get_yaxis().set_major_formatter(plt.FuncFormatter(lambda x, loc: f'{x:,.0f}'))
ax.legend(title='Tahun', fontsize=12, title_fontsize=13)
ax.grid(True, linestyle=':', alpha=0.7)
plt.xticks(fontsize=10)
plt.yticks(fontsize=10)
plt.tight_layout()
plt.show()

```



### 1.1.5. Analisis dan Interpretasi

```

print(f'--- Analisis Harga Bitcoin 2018 vs 2019 ---')
print(f'Harga Awal 2018 \t\t: ${prices_2018[0]:,.2f}')
print(f'Harga Akhir 2018 \t\t: ${prices_2018[-1]:,.2f}')
print(f'Rata-rata Harga 2018 \t\t: ${avg_price_2018:,.2f}')
print(f'Pengembalian Tahunan 2018 \t: {roi_2018:,.2f}%')
print('=' * 30)
print(f'Harga Awal 2019 \t\t: ${prices_2019[0]:,.2f}')
print(f'Harga Akhir 2019 \t\t: ${prices_2019[-1]:,.2f}')
print(f'Rata-rata Harga 2019 \t\t: ${avg_price_2019:,.2f}')
print(f'Pengembalian Tahunan 2019 \t: {roi_2019:,.2f}%')
print('=' * 30)
if roi_2019 > roi_2018:
    print('Kesimpulan: Tahun 2019 cenderung memberikan pengembalian yang lebih baik secara
persentase dari awal hingga akhir tahun.')
elif roi_2018 > roi_2019:
    print('Kesimpulan Tahun 2018 cenderung memberikan yang lebih baik secara persentase
dari awla hingga akhir tahun.')
else:
    print('Kesimpulan: Pengembalian persentase dari awla hingga akhir tahun untuk 2018 dan
2019 relatif lama.')

if avg_price_2018 > avg_price_2019:
    print('Namun. harga rata-rata Bitcoin secara keseluruhan lebih tinggi pada tahun
2018.')

```



```
else:
    print('Harga rata-rata Bitcoin secara keseluruhan lebih tinggi pada tahun 2019.')
```

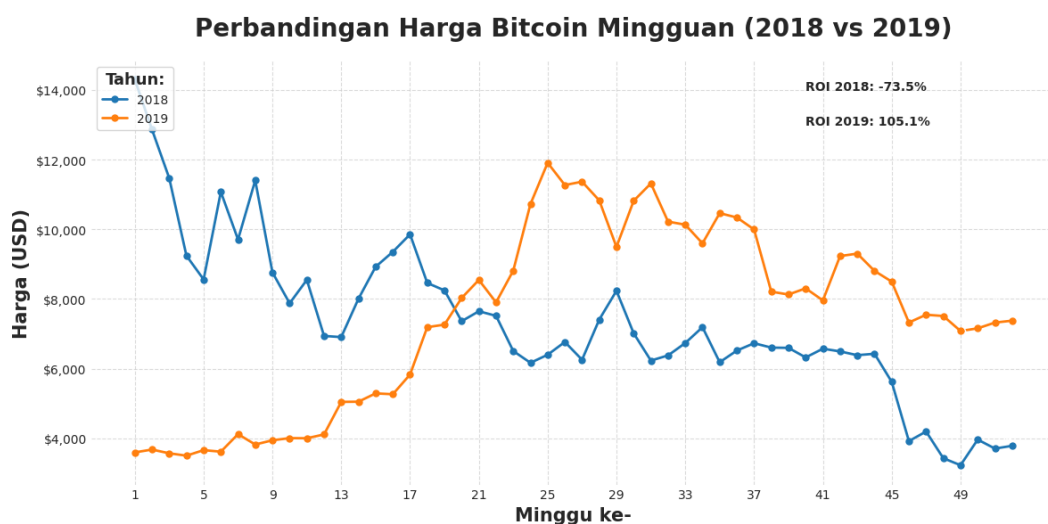
```

--- Analisis Harga Bitcoin 2018 vs 2019 ---
Harga Awal 2018      : $14,292.20
Harga Akhir 2018      : $3,785.40
Rata-rata Harga 2018   : $7,326.20
Pengembalian Tahunan 2018 : -73.51%
=====
Harga Awal 2019      : $3,597.20
Harga Akhir 2019      : $7,462.55
Rata-rata Harga 2019   : $7,462.55
Pengembalian Tahunan 2019 : 105.07%
=====
Kesimpulan: Tahun 2019 cenderung memberikan pengembalian yang lebih
            baik secara persentase dari awal hingga akhir tahun.
            Harga rata-rata Bitcoin secara keseluruhan lebih tinggi pada tahun 2019.

```

## 1.2. Analisis Pengembalian Bitcoin: 2018 atau 2019?

*Tahun apa, 2018 atau 2019, yang cenderung memberikan pengembalian yang lebih baik bagi pemegang Bitcoin?*



Berdasarkan visualisasi line chart yang dihasilkan, dapat diamati bahwa tahun **2019** cenderung memberikan pengembalian yang lebih baik bagi pemegang Bitcoin dibandingkan tahun 2018.

- **Tahun 2018 (ROI= -72,5%)**
  - Harga Bitcoin mengalami tren penurunan yang signifikan sepanjang tahun. Dimulai dengan harga yang relatif tinggi di awal tahun, terlihat adanya fluktuasi namun secara keseluruhan harga terus menurun hingga mencapai titik terendah di akhir tahun. Hal ini mengindikasikan bahwa pemegang Bitcoin pada tahun 2018 akan mengalami kerugian jika membeli di awal tahun dan menjual di akhir tahun.
- **Tahun 2019 (ROI= 105,1%)**
  - Terlihat adanya pembalikan tren dimana harga Bitcoin cenderung mengalami kenaikan. Meskipun terdapat fluktuasi harga, secara umum harga Bitcoin meningkat dari awal hingga akhir tahun. Ini menunjukkan bahwa pemegang Bitcoin yang membeli di awal 2019 dan menjual di akhir tahun berpotensi mendapatkan keuntungan.

Secara keseluruhan, tahun 2019 menunjukkan performa yang **lebih positif** dengan tren kenaikan harga, sementara tahun 2018 didominasi oleh tren penurunan.

### 1.3. Analisis Pemilihan Chart

*Chart apa yang anda pilih untuk problem diatas dan mengapa anda memilih chart tersebut?*

Chart yang dipilih untuk memvisualisasikan data harga Bitcoin adalah **Line Chart**.

Alasan pemilihan line chart:

- **Menampilkan Tren**

Line chart sangat efektif dalam menampilkan perubahan nilai suatu variabel (dalam hal ini, harga Bitcoin) terhadap waktu (minggu). Dengan menghubungkan titik-titik data, line chart dengan jelas menunjukkan tren naik, turun, atau stagnan dari harga Bitcoin selama periode waktu tertentu.

- **Perbandingan yang Mudah**

Penggunaan dua line chart dalam satu grafik (satu untuk 2018 dan satu untuk 2019) memudahkan perbandingan langsung antara pergerakan harga Bitcoin di kedua tahun tersebut. Pola dan tren dari masing-masing tahun dapat dengan mudah dibandingkan secara visual.

- **Kejelasan Informasi**

Line chart memberikan gambaran yang jelas dan ringkas tentang fluktuasi harga. Pembaca dapat dengan cepat memahami bagaimana harga Bitcoin berubah dari minggu ke minggu dan bagaimana pola perubahan tersebut berbeda antara tahun 2018 dan 2019.

- **Kontinuitas Data**

Harga Bitcoin direkam setiap minggu, menciptakan kumpulan data yang berkesinambungan. Line chart cocok untuk data berkesinambungan karena menekankan kesinambungan antara titik data, sehingga menyoroti tren dan pola dalam data.

### 1.4. Analisis Keuntungan Bitcoin: 2018 atau 2019?

*Tahun berapa pemegang bitcoin memiliki keuntungan yang lebih banyak?*

Berdasarkan analisis tren harga yang divisualisasikan, pemegang Bitcoin cenderung memiliki keuntungan yang lebih banyak pada tahun **2019**.

Penjelasan:

- Pada tahun 2018, harga Bitcoin secara konsisten menurun sepanjang tahun, yang berarti bahwa investor yang membeli Bitcoin di awal tahun akan mengalami kerugian jika mereka menjualnya di akhir tahun.
- Sebaliknya, pada tahun 2019, harga Bitcoin menunjukkan tren kenaikan secara keseluruhan. Meskipun ada fluktuasi, harga cenderung meningkat dari awal hingga akhir tahun, yang berpotensi memberikan keuntungan bagi investor.

Oleh karena itu, tahun 2019 lebih menguntungkan bagi pemegang Bitcoin dibandingkan dengan tahun 2018, dengan asumsi strategi beli dan tahan (*buy and hold*) dari awal hingga akhir tahun

## Soal No. 2: Lima Jenis Permen

**(Poin: 20)** Kita memiliki sekantong permen. Terdapat lima jenis permen, masing-masing diberi nama di bawah ini. Buat diagram yang menunjukkan persentase peluang bahwa kita akan mengeluarkan permen Snickers dari kantong jika kita melakukan pengambilan acak. Sebutkan peluang memilih permen Snickers.

- Chart apa yang anda pilih untuk problem diatas dan mengapa anda memilih chart tersebut?
- Berapa persen kemungkinan Anda akan memilih Snickers saat mengeluarkan permen dari tas secara acak?

## 2.1 Langkah-langkah Mengerjakan dengan Python

### 2.1.1 Import Libraries

```
import matplotlib.pyplot as plt
import numpy as np
```

### 2.1.2. Load Dataset

```
candy_names = ['Kit Kat', 'Snickers', 'Milky Way', 'Toblerone', 'Twix']
candy_counts = [52, 39, 78, 13, 78]
```

### 2.1.3. Data Preprocessing

#### 2.1.3.1. Hitung total jumlah permen

```
total_candies = np.sum(candy_counts)
```

#### 2.1.3.2. Hitung probabilitas mengambil snickers

```
snickers_count = candy_counts[candy_names.index('Snickers')]
probability_snickers = snickers_count / total_candies
percentage_snickers = probability_snickers * 100
```

#### 2.1.3.3. Hitung persentase setiap jenis permen (untuk diagram)

```
percentages = (candy_counts / total_candies) * 100
```

### 2.1.4. Visualisasi Data: Pie Chart

```
# Siapkan explode dan warna
explode = [0] * len(candy_names)
snickers_index = candy_names.index('Snickers')
explode[snickers_index] = 0.1

# Gunakan warna yang berbeda dan menonjolkan Snickers
colors = ['#8dd3c7', '#ffffb3', '#bebada', '#fb8072', '#80b1d3']

# Ganti warna Snickers secara spesifik
colors[snickers_index] = '#f032e6' # Magenta yang mencolok
```

```

# Pie chart
fig, ax = plt.subplots(figsize=(10, 8))
wedges, texts, autotexts = ax.pie(
    candy_counts,
    labels=candy_names,
    autopct=lambda p: f'{p:.1f}%',
    startangle=90,
    explode=explode,
    colors=colors,
    shadow=True,
    wedgeprops={'edgecolor': 'white', 'linewidth': 1}
)

# Judul dan anotasi
ax.set_title('Distribusi Permen dalam Kantong (Total: 260 buah)', fontsize=16, pad=20,
fontweight='bold')
ax.annotate(
    f'Peluang Snickers: {percentage_snickers:.1f}%',
    xy=(0.5, -0.1), xytext=(0, -20),
    textcoords='offset points', ha='center',
    arrowprops=dict(arrowstyle='->'), fontsize=12,
    bbox=dict(boxstyle='round,pad=0.5', fc='white', alpha=0.8)
)

# Legenda
ax.legend(wedges, candy_names, title="Jenis Permen", loc='upper right')

plt.tight_layout()
plt.show()

```

*(Ouput kode berupa visualisasi Pie Chart yang saya tampilkan di 2.2. Analisis Pemilihan Chart)*

### 2.1.5. Analisis dan Interpretasi

```

print('--- Analisis Probabilitas Permen dalam Kantong ---')
print(f'Data Permen:')
for name, count in zip(candy_names, candy_counts):
    print(f'\t- {name:<10}: {count} buah')
print('-' * 40)
print(f'Total Jumlah Permen\t\t\t: {total_candies} buah')
print(f'Jumlah Permen Snickers\t\t\t: {snickers_count} buah')
print(f'Probabilitas Mengambil Snickers {snickers_count}/{total_candies} :
{probability_snickers:.3f}')
print(f'Persentase Peluang Mengambil Snickers : {percentage_snickers:.1f}%)
print('-' * 40)

```

--- Analisis Probabilitas Permen dalam Kantong ---	
Data Permen:	
- Kit Kat	: 52 buah
- Snickers	: 39 buah
- Milky Way	: 78 buah
- Toblerone	: 13 buah
- Twix	: 78 buah
-----	
Total Jumlah Permen	: 260 buah
Jumlah Permen Snickers	: 39 buah
Probabilitas Mengambil Snickers $39/260$	: 0.150
Persentase Peluang Mengambil Snickers	: 15.0%

## 2.2. Analisis Pemilihan Chart

*Chart apa yang anda pilih untuk problem diatas dan mengapa anda memilih chart tersebut?*

Chart yang dipilih adalah **Pie Chart**.

**Distribusi Permen dalam Kantong (Total: 260 buah)**

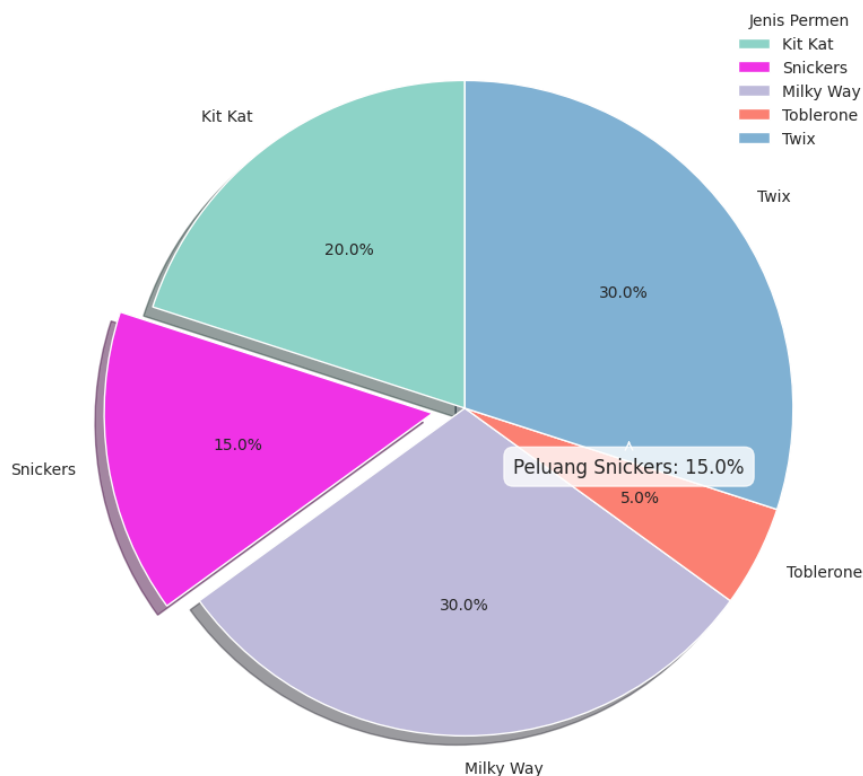


Diagram saya pilih adalah **pie chart** yang secara visual merepresentasikan distribusi persentase dari setiap jenis permen dalam kantong. Setiap slice pada pie chart mewakili satu jenis permen, dengan ukuran slice yang proporsional terhadap jumlah permen jenis tersebut dibandingkan dengan total permen dalam kantong. Slice yang secara spesifik merepresentasikan permen Snickers diberi penekanan visual melalui efek "explode" (sedikit terpisah dari pusat pie) dan warna yang berbeda dan mencolok (magenta). Anotasi dalam bentuk persentase (misalnya, 15.0%) ditambahkan pada setiap slice untuk memberikan informasi kuantitatif yang tepat tentang proporsi setiap jenis permen. Selain itu, diagram dilengkapi dengan judul deskriptif ("Distribusi Permen dalam Kantong (Total: 260 buah)") dan legenda yang jelas, yang memungkinkan interpretasi yang mudah dan akurat. Sebuah anotasi tambahan dengan panah mengarah ke slice Snickers secara spesifik menyebutkan persentase peluang Snickers, memperkuat fokus pada informasi yang relevan dengan pertanyaan.

Alasan pemilihan:

- **Komposisi Proporsional**

Pie chart sangat efektif dalam memvisualisasikan bagaimana suatu keseluruhan dibagi menjadi bagian-bagian atau kategori. Dalam konteks ini, keseluruhan adalah total jumlah permen dalam kantong, dan bagian-bagiannya adalah jenis-jenis permen yang berbeda. Pie chart secara intuitif menunjukkan proporsi relatif setiap jenis permen terhadap total.

- **Fokus pada Persentase**

Pie chart secara langsung menampilkan persentase, yang merupakan metrik kunci dalam menjawab pertanyaan tentang peluang. Setiap slice secara visual mewakili persentase dari total, membuat perbandingan antar kategori menjadi mudah.

- **Kemudahan Interpretasi**

Pie chart mudah dipahami oleh audiens yang luas. Representasi visual dari "bagian dari keseluruhan" sangat jelas, bahkan bagi orang-orang yang tidak memiliki latar belakang teknis yang kuat.

- **Penekanan pada Satu Kategori**

Dalam soal ini, ada penekanan khusus pada permen Snickers. Pie chart memungkinkan penekanan visual yang efektif pada kategori tertentu melalui teknik seperti "explode" dan penggunaan warna yang berbeda, menarik perhatian pada informasi yang paling relevan.

## **2.3. Peluang Memilih Snickers**

*Berapa persen kemungkinan Anda akan memilih Snickers saat mengeluarkan permen dari tas secara acak?*

Kemungkinan memilih permen Snickers saat mengeluarkan permen dari tas secara acak adalah **0,15** atau **15.0%**. Ini secara langsung ditunjukkan oleh persentase yang dihitung dan juga divisualisasikan dalam pie chart.

### 3. Soal No. 3: Menu Makanan Penutup

(Poin: 20) Restoran memiliki menu makanan penutup yang terlalu besar. Mereka ingin memotong beberapa item dari menu. Untuk membuat sebagian besar pelanggan mereka senang, mereka ingin menghapus hanya tiga makanan penutup yang paling tidak populer dari menu. Kita memiliki daftar makanan penutup yang disajikan restoran, serta hitungan berapa kali makanan penutup tersebut dijual dalam seminggu terakhir. Buat visualisasi yang menunjukkan popularitas relatif dari makanan penutup. Sebutkan tiga makanan penutup yang harus disingkirkan.

- Chart apa yang anda pilih untuk problem di atas dan mengapa anda memilih chart tersebut?
- Makanan penutup apa saja yang perlu anda sarankan untuk dikeluarkan dari menu?

### 3.1 Langkah-langkah Mengerjakan dengan Python

#### 3.1.1 Import Libraries

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

#### 3.1.2. Load Dataset

```
dessert_sales = {
    'Lava Cake': 14,
    'Mousse': 5,
    'Chocolate Cake': 12,
    'Ice Cream': 19,
    'Truffles': 6,
    'Brownie': 8,
    'Chocolate Chip Cookie': 12,
    'Chocolate Pudding': 9,
    'Souffle': 10,
    'Chocolate Cheesecake': 17,
    'Chocolate Chips': 2,
    'Fudge': 9,
    'Mochi': 13,
}
```

#### 3.1.3. Data Preprocessing

```
dessert_sales_df = pd.DataFrame(list(dessert_sales.items()), columns=['Dessert',
'Sales'])
```

#### 3.1.4. Visualisasi Data: Bar Chart

```
# Mengurutkan data berdasarkan penjualan
dessert_sales_df = dessert_sales_df.sort_values(by='Sales')

# Membuat kolom warna untuk highlight 3 terendah
colors = ['red' if x in dessert_sales_df['Dessert'].head(3).values else 'blue' for x in
dessert_sales_df['Dessert']]

plt.figure(figsize=(12, 6))
sns.barplot(x='Dessert', y='Sales', data=dessert_sales_df, palette=colors) # Menggunakan
palette untuk warna
plt.title('Popularitas Makanan Penutup', fontsize=19, fontweight='bold') # Memperbesar
ukuran judul
```

```
plt.xlabel('Makanan Penutup', fontsize=12, fontweight='bold')
plt.ylabel('Jumlah Penjualan', fontsize=12, fontweight='bold')
plt.xticks(rotation=45, ha='right', fontsize=10) # Rotasi dan ukuran font label sumbu x
plt.yticks(fontsize=10) # Ukuran font label sumbu y
plt.tight_layout()
sns.despine(left=True, bottom=False) # Menghilangkan spine (garis tepi) atas dan kanan
plt.grid(axis='y', linestyle='--', alpha=0.7) # Menambahkan grid horizontal

# Menambahkan anotasi untuk nilai penjualan
for index, row in dessert_sales_df.iterrows():
    plt.text(index, row['Sales'] + 0.5, f"{row['Sales']}", ha='center', fontsize=8)
plt.show()
```

(Ouput kode berupa visualisasi Bar Chart yang saya tampilkan di 3.2. Analisis Pemilihan Chart)

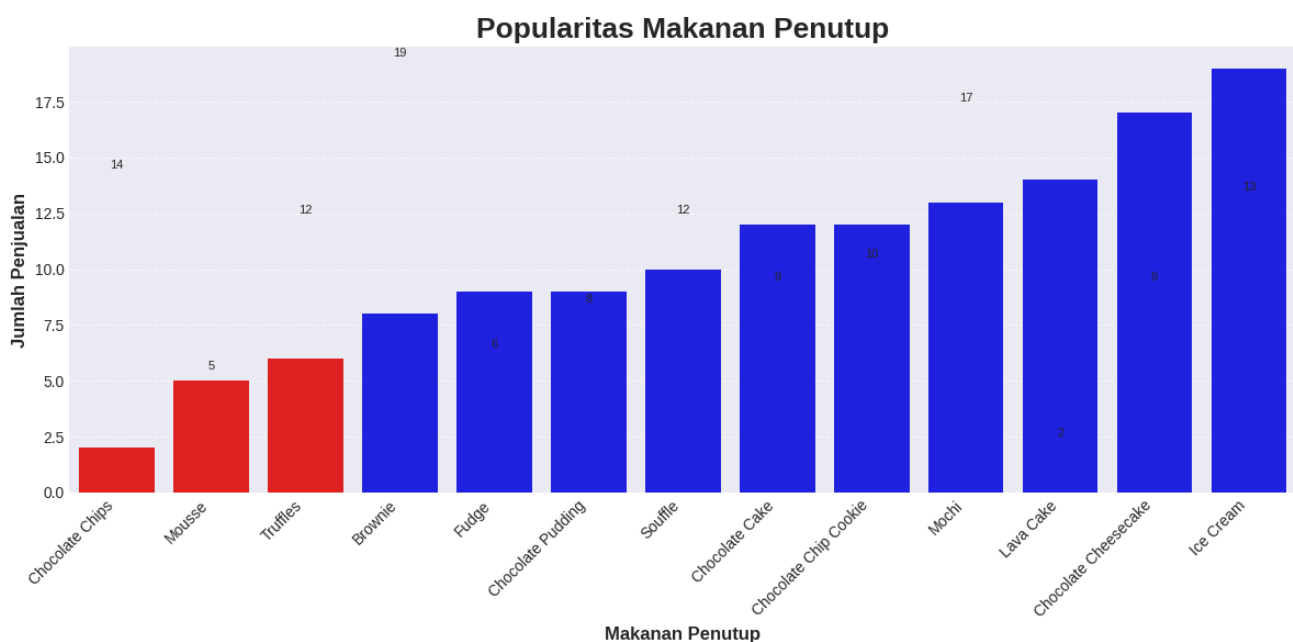
### 3.1.5. Analisis dan Interpretasi

```
# mengurutkan data berdasarkan penjualan terendah
least_popular_dessert = dessert_sales_df.sort_values(by='Sales').head(3)
print(f'\nTiga Makanan Penutup yang Paling Tidak Populer: \n{least_popular_dessert}')
```

```
Tiga Makanan Penutup yang Paling Tidak Populer:
      Dessert  Sales
10  Chocolate Chips    2
1      Mousse         5
4      Truffles        6
```

## 3.2. Analisis Pemilihan Chart

Chart apa yang anda pilih untuk problem di atas dan mengapa anda memilih chart tersebut?



Untuk memvisualisasikan popularitas relatif dari berbagai makanan penutup berdasarkan jumlah penjualan, jenis grafik yang paling tepat dan informatif adalah **Bar Chart** (Grafik Batang). Berikut adalah beberapa alasan mengapa Bar Chart menjadi pilihan yang sangat baik dalam konteks ini:

1. **Representasi Perbandingan yang Jelas dan Intuitif**



Bar Chart secara visual merepresentasikan besaran nilai (dalam kasus ini, jumlah penjualan) melalui ketinggian atau panjang batang. Ini memungkinkan kita untuk dengan mudah dan cepat membandingkan popularitas antara satu jenis makanan penutup dengan yang lainnya. Perbedaan ketinggian batang secara langsung mencerminkan perbedaan jumlah penjualan, sehingga audiens dapat langsung memahami peringkat popularitas setiap item.

## 2. Fokus pada Nilai Diskret

Data penjualan makanan penutup bersifat diskret, di mana setiap jenis makanan penutup memiliki jumlah penjualan yang spesifik dan terpisah. Bar Chart dirancang untuk memvisualisasikan data diskret seperti ini dengan sangat efektif, di mana setiap batang merepresentasikan satu kategori (jenis makanan penutup) dan tingginya merepresentasikan nilai (jumlah penjualan).

## 3. Kemudahan Interpretasi

Grafik batang sangat mudah diinterpretasikan oleh berbagai kalangan audiens, bahkan mereka yang tidak memiliki latar belakang statistik yang kuat. Pembaca dapat dengan cepat mengidentifikasi makanan penutup mana yang paling populer (batang tertinggi) dan mana yang paling tidak populer (batang terpendek).

## 4. Penekanan pada Peringkat dan Urutan

Dengan mengurutkan batang berdasarkan jumlah penjualan (seperti yang telah dilakukan dalam visualisasi yang diberikan), Bar Chart secara efektif menyoroti peringkat popularitas makanan penutup. Ini sangat relevan dengan tujuan restoran untuk mengidentifikasi tiga item dengan penjualan terendah.

## 5. Fleksibilitas Penambahan Informasi Tambahan

Bar Chart memungkinkan penambahan elemen visual yang memperkaya informasi, seperti label nilai di atas setiap batang (seperti yang ada pada visualisasi ini). Ini memberikan pembaca nilai penjualan yang tepat untuk setiap makanan penutup, melengkapi perbandingan visual dengan data numerik yang akurat. Penggunaan warna yang berbeda untuk menyoroti kelompok tertentu (dalam hal ini, tiga makanan penutup terkurang populer) juga merupakan keunggulan Bar Chart dalam menyampaikan pesan spesifik.

Meskipun jenis grafik lain seperti Pie Chart mungkin tampak relevan karena menunjukkan proporsi, Pie Chart kurang efektif dalam membandingkan nilai yang relatif mirip dan bisa menjadi berantakan jika terdapat banyak kategori. Line Chart lebih cocok untuk menunjukkan tren dari waktu ke waktu, yang tidak menjadi fokus dalam analisis popularitas statis ini. Oleh karena itu, Bar Chart adalah pilihan yang paling solid dan efektif untuk menjawab pertanyaan mengenai popularitas relatif makanan penutup.

### 3.3. Tiga Menu Penutup yang Perlu Dihapus

*Makanan penutup apa saja yang perlu anda sarankan untuk dikeluarkan dari menu?*

Berdasarkan visualisasi Bar Chart dan data penjualan, kita dapat secara jelas mengidentifikasi tiga makanan penutup yang memiliki jumlah penjualan paling rendah dalam seminggu terakhir. Makanan penutup ini adalah kandidat utama untuk dipertimbangkan penghapusannya dari menu, dengan asumsi bahwa biaya produksi dan faktor-faktor lain relatif serupa di antara semua item.

Dari data dan visualisasi, tiga makanan penutup yang menunjukkan popularitas terendah (jumlah penjualan terkecil) adalah:

1. **Chocolate Chips:** Hanya terjual sebanyak **2** kali dalam seminggu terakhir. Ini menunjukkan permintaan yang sangat rendah dari pelanggan.
2. **Mousse:** Tercatat hanya **5** penjualan dalam seminggu. Angka ini juga jauh di bawah rata-rata penjualan makanan penutup lainnya.

3. **Truffles:** Dengan total penjualan sebanyak **6** kali, Truffles menjadi makanan penutup dengan popularitas ketiga terendah dalam daftar ini.

#### **Rekomendasi:**

Sebagai seorang data scientist, berdasarkan analisis data penjualan ini, saya akan merekomendasikan kepada pihak restoran untuk mempertimbangkan **menghapus "Chocolate Chips", "Mousse", dan "Truffles" dari menu makanan penutup mereka**. Alasan rekomendasi:

- **Penjualan Terendah:** Ketiga makanan penutup ini secara konsisten menunjukkan jumlah penjualan yang paling rendah dibandingkan dengan pilihan makanan penutup lainnya. Ini mengindikasikan kurangnya minat atau preferensi pelanggan terhadap item-item ini.
- **Potensi Optimalisasi Menu:** Dengan menghilangkan item yang kurang populer, restoran dapat menyederhanakan operasional, mengurangi potensi pemborosan bahan baku, dan fokus pada promosi makanan penutup yang lebih diminati oleh pelanggan.
- **Efisiensi Sumber Daya:** Sumber daya seperti ruang penyimpanan, waktu persiapan staf, dan biaya inventaris yang saat ini dialokasikan untuk ketiga item ini dapat dialihkan untuk mendukung produksi dan promosi item yang lebih laku.
- **Peningkatan Kepuasan Pelanggan Secara Keseluruhan:** Meskipun ada sebagian kecil pelanggan yang mungkin menyukai ketiga makanan penutup ini, menghilangkan item yang tidak populer dapat memungkinkan restoran untuk lebih fokus pada kualitas dan variasi pilihan yang lebih luas dan diminati oleh mayoritas pelanggan.

#### **Pertimbangan Tambahan (Meskipun Tidak Diminta):**

Perlu untuk diingat bahwa keputusan akhir penghapusan menu mungkin juga dipengaruhi oleh faktor lain di luar data penjualan mingguan ini. Restoran mungkin ingin mempertimbangkan:

- **Margin Keuntungan:** Apakah ketiga item ini memiliki margin keuntungan yang sangat tinggi yang mungkin membenarkan keberadaannya meskipun penjualannya rendah?
- **Biaya Bahan Baku:** Apakah biaya bahan baku untuk ketiga item ini jauh lebih rendah dibandingkan yang lain?
- **Niche Market:** Apakah ketiga item ini melayani segmen pelanggan tertentu yang penting bagi restoran?
- **Strategi Menu Keseluruhan:** Apakah ketiga item ini memainkan peran strategis dalam keseluruhan penawaran menu (misalnya, sebagai satu-satunya pilihan berbasis cokelat putih)?
- **Umpan Balik Pelanggan Langsung:** Apakah ada keluhan atau permintaan khusus terkait dengan ketersediaan atau kualitas ketiga item ini?

Namun, berdasarkan data popularitas penjualan saat ini, **"Chocolate Chips", "Mousse", dan "Truffles"** adalah kandidat yang paling kuat untuk dievaluasi lebih lanjut dan dipertimbangkan untuk dihapus dari menu makanan penutup guna mengoptimalkan penawaran dan meningkatkan efisiensi.

## Soal No. 4: Penggunaan CPU

(Poin: 20) Kita memiliki penggunaan CPU rata-rata per jam untuk komputer pekerja selama seminggu. Setiap baris data mewakili satu hari dalam seminggu yang dimulai dengan Senin. Setiap kolom data adalah satu jam dalam sehari dimulai dengan 0 menjadi tengah malam. Buat bagan yang menunjukkan penggunaan CPU selama seminggu. Anda harus dapat menjawab pertanyaan-pertanyaan berikut menggunakan bagan:

- Chart apa yang anda pilih untuk problem diatas dan mengapa anda memilih chart tersebut?
- Jam berapa pekerja biasanya makan siang?
- Apakah pekerja tersebut bekerja pada akhir pekan?
- Pada hari apa pekerja mulai bekerja pada komputer mereka pada malam hari?

## 4.1 Langkah-langkah Mengerjakan dengan Python

### 4.1.1 Import Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

### 4.1.2. Load Dataset

```
cpu_usage = [
    [2, 2, 4, 2, 4, 1, 1, 4, 4, 12, 22, 23, 45, 9, 33, 56, 23, 40, 21, 6, 6, 2, 2, 3], #
    Monday
    [1, 2, 3, 2, 3, 2, 3, 2, 7, 22, 45, 44, 33, 9, 23, 19, 33, 56, 12, 2, 3, 1, 2, 2], #
    Tuesday
    [2, 3, 1, 2, 4, 4, 2, 2, 1, 2, 5, 31, 54, 7, 6, 34, 68, 34, 49, 6, 6, 2, 2, 3], #
    Wednesday
    [1, 2, 3, 2, 4, 1, 2, 4, 1, 17, 24, 18, 41, 3, 44, 42, 12, 36, 41, 2, 2, 4, 2, 4], #
    Thursday
    [4, 1, 2, 2, 3, 2, 5, 1, 2, 12, 33, 27, 43, 8, 38, 53, 29, 45, 39, 3, 1, 1, 3, 4], #
    Friday
    [2, 3, 1, 2, 2, 5, 2, 8, 4, 2, 3, 1, 5, 1, 2, 3, 2, 6, 1, 2, 2, 1, 4, 3], # Saturday
    [1, 2, 3, 1, 1, 3, 4, 2, 3, 1, 2, 2, 5, 3, 2, 1, 4, 2, 45, 26, 33, 2, 2, 1], # Sunday
]
```

### 4.1.3. Data Preprocessing

```
cpu_usage_df = pd.DataFrame(cpu_usage)

days = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
hours = [f'{i:02d}:00' for i in range(24)]
cpu_usage_df.index = days
cpu_usage_df.columns = hours
cpu_usage_df
```

```
Out[ ]:
```

	00:00	01:00	02:00	03:00	04:00	05:00	06:00	07:00	08:00	09:00	...	14:00	15:00	16:00	17:00
<b>Monday</b>	2	2	4	2	4	1	1	4	4	12	...	33	56	23	40
<b>Tuesday</b>	1	2	3	2	3	2	3	2	7	22	...	23	19	33	56
<b>Wednesday</b>	2	3	1	2	4	4	2	2	1	2	...	6	34	68	34
<b>Thursday</b>	1	2	3	2	4	1	2	4	1	17	...	44	42	12	36
<b>Friday</b>	4	1	2	2	3	2	5	1	2	12	...	38	53	29	45
<b>Saturday</b>	2	3	1	2	2	5	2	8	4	2	...	2	3	2	6
<b>Sunday</b>	1	2	3	1	1	3	4	2	3	1	...	2	1	4	2

```
# Melting dataset untuk seaborn
cpu_usage_melted = cpu_usage_df.reset_index().melt(id_vars='index', var_name='Hour',
value_name='CPU Usage')
cpu_usage_melted.rename(columns={'index': 'Day'}, inplace=True)
cpu_usage_melted
```

```
Out[ ]:
```

	Day	Hour	CPU Usage
<b>0</b>	Monday	00:00	2
<b>1</b>	Tuesday	00:00	1
<b>2</b>	Wednesday	00:00	2
<b>3</b>	Thursday	00:00	1
<b>4</b>	Friday	00:00	4
<b>...</b>	...	...	...
<b>163</b>	Wednesday	23:00	3
<b>164</b>	Thursday	23:00	4
<b>165</b>	Friday	23:00	4
<b>166</b>	Saturday	23:00	3
<b>167</b>	Sunday	23:00	1

#### 4.1.4. Visualisasi Data: Heat Map

```
plt.figure(figsize=(12, 8))
sns.heatmap(cpu_usage_df, cmap='YlGnBu', annot=True, fmt='d', linewidths=.5) # fmt='d'
for integers
plt.title('Penggunaan CPU Rata-rata per Jam Selama Perminggu', fontweight='bold',
fontsize=20)
plt.xlabel('Jam', fontweight='bold', fontsize=15)
plt.ylabel('Hari', fontweight='bold', fontsize=15)
plt.yticks(rotation=0)
plt.tight_layout()
plt.show()
```

*(Ouput kode berupa visualisasi Heat Map yang saya tampilkan di 4.2. Analisis Pemilihan Chart)*

#### 4.1.5. Analisis dan Interpretasi

```
print("--- Pertanyaan 1: Jam berapa pekerja biasanya makan siang? ---")
lunch_hours = cpu_usage_df.iloc[:, 12:14].mean(axis=0) # Jam 12:00 - 13:00
lunch_hour = lunch_hours.idxmin()
avg_usage = lunch_hours.min()
print(f'Jam makan siang pekerja adalah sekitar {lunch_hour} dengan rata-rata penggunaan
CPU { avg_usage:.2f}')
```

```

print('\n--- Pertanyaan 2: Apakah pekerja tersebut bekerja pada akhir pekan? --')
weekend_usage = cpu_usage_df.loc[['Saturday', 'Sunday']].mean().mean()
print(f'Rata-rata penggunaan CPU pada akhir pekan adalah: {weekend_usage:.2f}')
if weekend_usage > 5:
    print('Pekerja tersebut bekerja pada akhir pekan.')
else:
    print('Pekerja tersebut tidak bekerja pada akhir pekan.')

print('\n--- Pertanyaan 3: Pada hari apa pekerja mulai bekerja pada komputer mereka pada malam hari? ---')
night_start_days = cpu_usage_df.loc[:, '00:00':'06:00'].mean(axis=1).idxmax()
print(f'Pekerja mulai bekerja pada komputer mereka pada malam hari {night_start_days}')

--- Pertanyaan 1: Jam berapa pekerja biasanya makan siang? ---
Jam makan siang pekerja adalah sekitar 13:00 dengan rata-rata penggunaan CPU 5.71

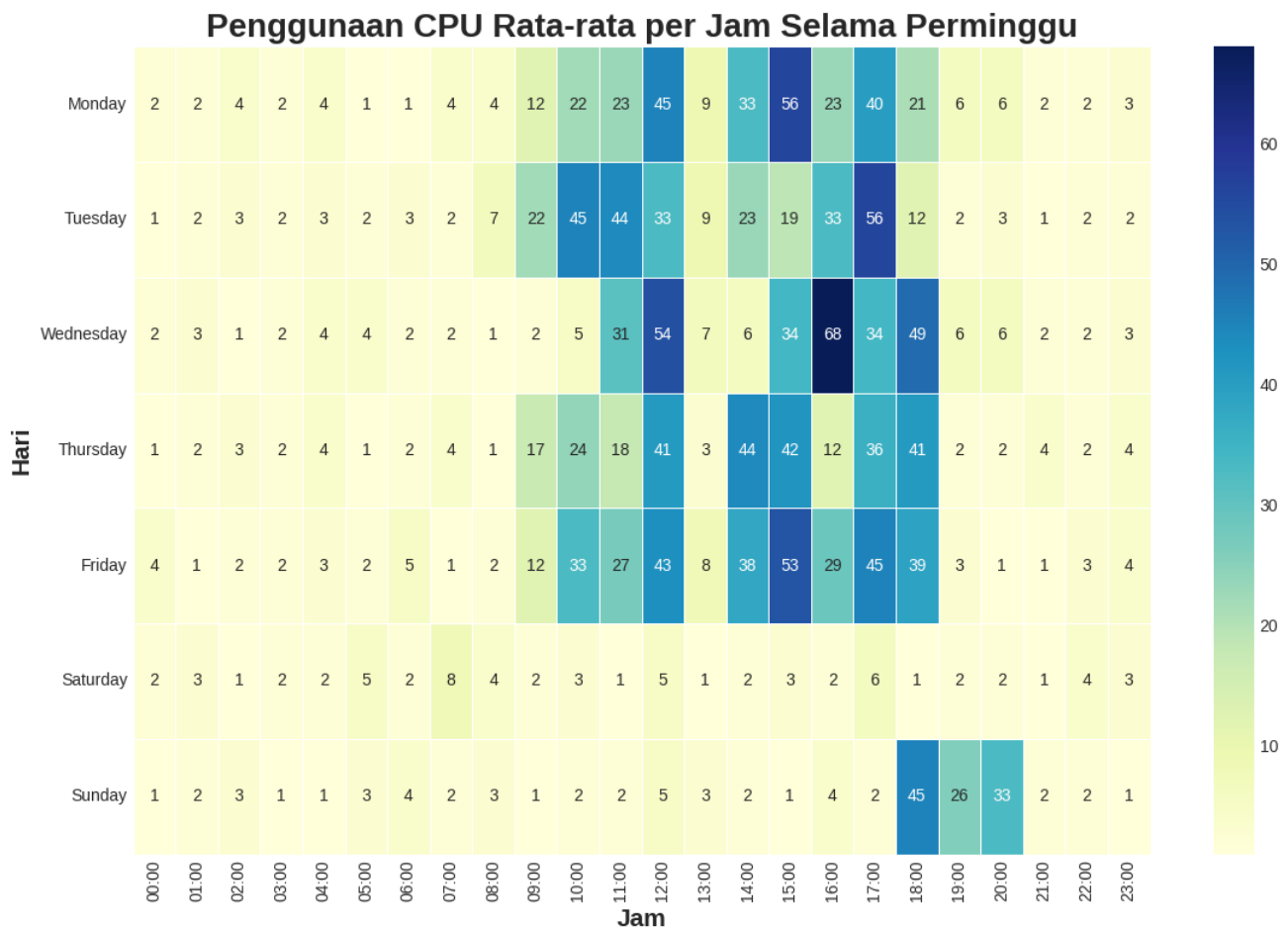
--- Pertanyaan 2: Apakah pekerja tersebut bekerja pada akhir pekan? --
Rata-rata penggunaan CPU pada akhir pekan adalah: 4.54
Pekerja tersebut tidak bekerja pada akhir pekan.

--- Pertanyaan 3: Pada hari apa pekerja mulai bekerja pada komputer mereka pada malam hari? ---
Pekerja mulai bekerja pada komputer mereka pada malam hari Friday

```

## 4.2. Analisis Pemilihan Chart

*Chart apa yang anda pilih untuk problem diatas dan mengapa anda memilih chart tersebut?*



Dalam kasus visualisasi data penggunaan CPU rata-rata per jam selama seminggu ini, jenis grafik yang paling efektif dan informatif adalah **Heatmap**. Pemilihan Heatmap didasarkan pada kemampuannya yang unggul dalam merepresentasikan matriks data dua dimensi dengan intensitas warna sebagai indikator nilai. Berikut

adalah alasan mendasar mengapa Heatmap menjadi pilihan yang sangat tepat untuk menganalisis pola penggunaan CPU dalam konteks waktu (jam) dan hari:

#### 1. Visualisasi Intensitas Data yang Efisien

Data penggunaan CPU yang kita miliki terstruktur dalam format tabel, di mana baris merepresentasikan hari dalam seminggu dan kolom merepresentasikan jam dalam sehari. Heatmap secara elegan mengubah tabel numerik ini menjadi representasi visual di mana setiap sel diwarnai berdasarkan nilai penggunaan CPU. Semakin tinggi penggunaan CPU, semakin intens warna yang ditampilkan (dalam visualisasi ini, menuju warna biru yang lebih gelap), dan sebaliknya, penggunaan CPU yang rendah ditunjukkan dengan warna yang lebih terang (menuju warna kuning muda). Ini memungkinkan kita untuk dengan cepat mengidentifikasi area dengan aktivitas CPU tinggi dan rendah secara keseluruhan.

#### 2. Identifikasi Pola Temporal dan Harian

Heatmap sangat efektif dalam mengungkapkan pola yang mungkin tersembunyi dalam tabel data mentah. Dengan melihat distribusi warna di seluruh grafik, kita dapat dengan mudah mengidentifikasi jam-jam sibuk (penggunaan CPU tinggi) dan jam-jam sepi (penggunaan CPU rendah) untuk setiap hari dalam seminggu. Kita juga dapat membandingkan pola penggunaan CPU antar hari, misalnya, apakah ada perbedaan signifikan antara hari kerja dan akhir pekan.

#### 3. Kepadatan Informasi

Dalam satu tampilan visual, Heatmap mampu menyajikan informasi yang sangat padat. Kita dapat melihat penggunaan CPU untuk setiap jam dalam setiap hari secara simultan, memberikan gambaran komprehensif tentang pola kerja pekerja selama seminggu penuh. Ini jauh lebih efisien daripada mencoba menganalisis serangkaian grafik garis terpisah untuk setiap hari atau setiap jam.

#### 4. Intuisi Visual

Otak manusia sangat baik dalam memproses informasi visual, terutama yang berkaitan dengan perbedaan warna dan intensitas. Heatmap memanfaatkan kemampuan ini untuk menyampaikan informasi secara intuitif. Area dengan warna yang mencolok akan secara otomatis menarik perhatian kita, mengarahkan kita ke titik-titik penting dalam data.

#### 5. Kemampuan untuk Menyoroti Anomali

Pola warna yang tidak biasa atau intensitas yang berbeda dari sekitarnya dapat dengan mudah terdeteksi dalam Heatmap. Misalnya, jika ada lonjakan penggunaan CPU yang tidak terduga pada jam atau hari tertentu, ini akan terlihat jelas sebagai area dengan warna yang lebih intens.

Dalam konteks pertanyaan yang diajukan, Heatmap memungkinkan kita untuk menjawabnya secara visual: jam makan siang kemungkinan akan ditandai dengan penurunan intensitas warna di sekitar jam-jam tertentu, pekerjaan di akhir pekan akan terlihat dari intensitas warna pada baris Sabtu dan Minggu, dan awal kerja di malam hari akan terlihat dari transisi peningkatan intensitas warna pada kolom-kolom jam malam.

Meskipun jenis grafik lain seperti Line Chart atau Bar Chart dapat digunakan untuk memvisualisasikan penggunaan CPU per jam untuk setiap hari secara terpisah, mereka akan menghasilkan banyak grafik individual yang sulit untuk dibandingkan dan dianalisis secara keseluruhan. Heatmap merangkum semua informasi ini dalam satu visualisasi yang koheren dan informatif, menjadikannya pilihan yang superior untuk memahami pola penggunaan CPU lintas waktu dan hari.

### 4.3. Waktu Para Pekerja Makan Siang

*Jam berapa pekerja biasanya makan siang?*

Untuk mengidentifikasi jam makan siang, kita perlu mencari periode waktu di mana terjadi penurunan signifikan dalam penggunaan CPU secara konsisten di sebagian besar hari kerja (Senin hingga Jumat). Dari Heatmap, kita dapat mengamati bahwa terdapat penurunan intensitas warna (menuju kuning muda) di sekitar

**jam 12:00 hingga 13:00.** Ini mengindikasikan bahwa pekerja kemungkinan besar mengambil **istirahat makan siang pada pukul 13:00**, di mana aktivitas komputer mereka cenderung menurun. Meskipun ada beberapa variasi di hari tertentu, pola penurunan penggunaan CPU yang paling konsisten terlihat pada rentang waktu ini.

#### 4.4. Apakah Para Pekerja Bekerja di Akhir Pekan

*Apakah pekerja tersebut bekerja pada akhir pekan?*

Untuk menjawab pertanyaan ini, kita perlu melihat intensitas warna pada baris yang merepresentasikan hari **Sabtu** dan **Minggu**. Secara keseluruhan, intensitas warna pada kedua hari akhir pekan ini cenderung **lebih terang (menuju kuning muda)** dibandingkan dengan hari kerja. Ini menunjukkan bahwa **penggunaan CPU secara signifikan lebih rendah pada akhir pekan**. Meskipun ada sedikit aktivitas sesekali di beberapa jam, tingkat penggunaan CPU secara umum sangat rendah, mengindikasikan bahwa **pekerja tersebut kemungkinan besar tidak bekerja secara aktif atau memiliki beban kerja yang sangat ringan pada akhir pekan**.

#### 4.5. Hari Para Pekerja Mulai Bekerja di Malam Hari

*Pada hari apa pekerja mulai bekerja pada komputer mereka pada malam hari?*

Untuk menentukan hari di mana pekerja mulai bekerja pada malam hari, kita perlu mencari hari di mana terjadi peningkatan signifikan dalam penggunaan CPU pada jam-jam malam (misalnya, setelah pukul 18:00). Dengan mengamati kolom-kolom yang merepresentasikan jam-jam malam (18:00 hingga 23:00), kita melihat bahwa pada hari **Jumat**, terdapat peningkatan yang cukup signifikan dalam intensitas warna (menuju biru) pada jam **18:00 dan seterusnya**. Sementara pada hari-hari kerja lainnya (Senin hingga Kamis), penggunaan CPU cenderung menurun setelah jam kerja reguler, pada hari Jumat terlihat adanya aktivitas yang berlanjut hingga malam hari. Oleh karena itu, dapat disimpulkan bahwa **pekerja tersebut mulai bekerja atau memiliki aktivitas signifikan pada komputer mereka pada malam hari, terutama pada hari Jumat**.

#### Kesimpulan Analisis

Secara keseluruhan, Heatmap ini memberikan visualisasi yang jelas dan ringkas mengenai pola penggunaan CPU pekerja selama seminggu. Kita dapat melihat ritme kerja harian dengan periode aktivitas tinggi di siang hari kerja dan penurunan di malam hari, indikasi istirahat makan siang di sekitar pukul 13:00, aktivitas yang sangat minim di akhir pekan, dan kecenderungan untuk bekerja hingga larut malam pada hari Jumat. Informasi ini sangat berharga bagi manajemen untuk memahami pola kerja, mengidentifikasi potensi waktu istirahat, dan mengevaluasi distribusi beban kerja sepanjang minggu.

## Soal No. 5: Pertumbuhan Jamur

(Poin: 20) Seorang peneliti sedang mempelajari jamur. Mereka telah menemukan cincin jamur dan memberi label koordinat. Biasanya jamur menyebar keluar dari pusat jamur awal. Dengan koordinat di bawah ini, peneliti ingin menjawab pertanyaan: Kira-kira di manakah letak pusat pertumbuhan jamur? Buat bagan yang memungkinkan peneliti memperkirakan pusat pertumbuhan.

- Chart apa yang anda pilih untuk problem diatas dan mengapa anda memilih chart tersebut?
- Koordinat pusat (x,y) pusat pertumbuhan jamur berada di?

## 5.1 Langkah-langkah Mengerjakan dengan Python

### 5.1.1 Import Libraries

```
import pandas as pd
import matplotlib.pyplot as plt
```

### 5.1.2. Load Dataset

```
x = [4.61, 5.08, 5.18, 7.82, 10.46, 7.66, 7.6, 9.32, 14.04, 9.95, 4.95, 7.23,
      5.21, 8.64, 10.08, 8.32, 12.83, 7.51, 7.82, 6.29, 0.04, 6.62, 13.16, 6.34,
      0.09, 10.04, 13.06, 9.54, 11.32, 7.12, -0.67, 10.5, 8.37, 7.24,
      9.18, 10.12, 12.29, 8.53, 11.11, 9.65, 9.42, 8.61, -0.67, 5.94, 6.49,
      7.57, 3.11, 8.7, 5.28, 8.28, 9.55, 8.33, 13.7, 6.65, 2.4, 3.54, 9.19, 7.51,
      -0.68,
      8.47, 14.82, 5.31, 14.01, 8.75, -0.57, 5.35, 10.51, 3.11, -0.26
      , 5.74,
      8.33, 6.5, 13.85, 9.78, 4.91, 4.19, 14.8, 10.04, 13.47, 3.28];
y = [-2.36, -3.41, 13.01, -2.91, -2.28, 12.83, 13.13, 11.94, 0.93, -2.76, 13.31,
      -3.57, -2.33, 12.43, -1.83, 12.32, -0.42, -3.08, -2.98, 12.46,
      8.34, -3.19,
      -0.47, 12.78, 2.12, -2.72, 10.64, 11.98, 12.21, 12.52, 5.53, 11.72, 12.91,
      12.56, -2.49, 12.08, -1.09, -2.89, -1.78, -2.47, 12.77, 12.41,
      5.33, -3.23,
      13.45, -3.41, 12.46, 12.1, -2.56, 12.51, -2.37, 12.76, 9.69, 12.59, -1.12,
      -2.8, 12.94, -3.55, 7.33, 12.59, 2.92, 12.7, 0.5, 12.57, 6.39,
      12.84,
      -1.95, 11.76, 6.82, 12.44, 13.28, -3.46, 0.7, -2.55, -2.37, 12.48, 7.26,
      -2.45, 0.31, -2.51]
```

### 5.1.3. Data Preprocessing

```
data = {'x': x, 'y': y}
mushroom_df = pd.DataFrame(data)
mushroom_df
```

Out[ ]:	x	y	...	...	...
0	4.61	-2.36	75	4.19	12.48
1	5.08	-3.41	76	14.80	7.26
2	5.18	13.01	77	10.04	-2.45
3	7.82	-2.91	78	13.47	0.31
4	10.46	-2.28	79	3.28	-2.51
...	...	...	80 rows × 2 columns		



### 5.1.4. Visualisasi Data: Scatter Plot & Analisis serta Interpretasi

```
# Visualisasi Data
plt.figure(figsize=(8, 8))
plt.scatter(mushroom_df['x'], mushroom_df['y'], alpha=0.7, label='Lokasi Jamur')

# Estimasi Pusat Pertumbuhan
center_x = mushroom_df['x'].mean()
center_y = mushroom_df['y'].mean()

# Menandai pusat pertumbuhan pada plot
plt.scatter(center_x, center_y, color='r', marker='x', s=100, label='Perkiraan Pusat
Pertumbuhan')

# Analisis dan Interpretasi
plt.title('Penyerbaran Jamur dan Perkiraan Pusat Pertumbuhan', fontsize=17,
fontweight='bold')
plt.xlabel('Koordinat X', fontsize=12, fontweight='bold')
plt.ylabel('Koordinat Y', fontsize=12, fontweight='bold')
plt.legend()
plt.grid(True, linestyle=':', alpha=0.7)
plt.tight_layout()
plt.show()

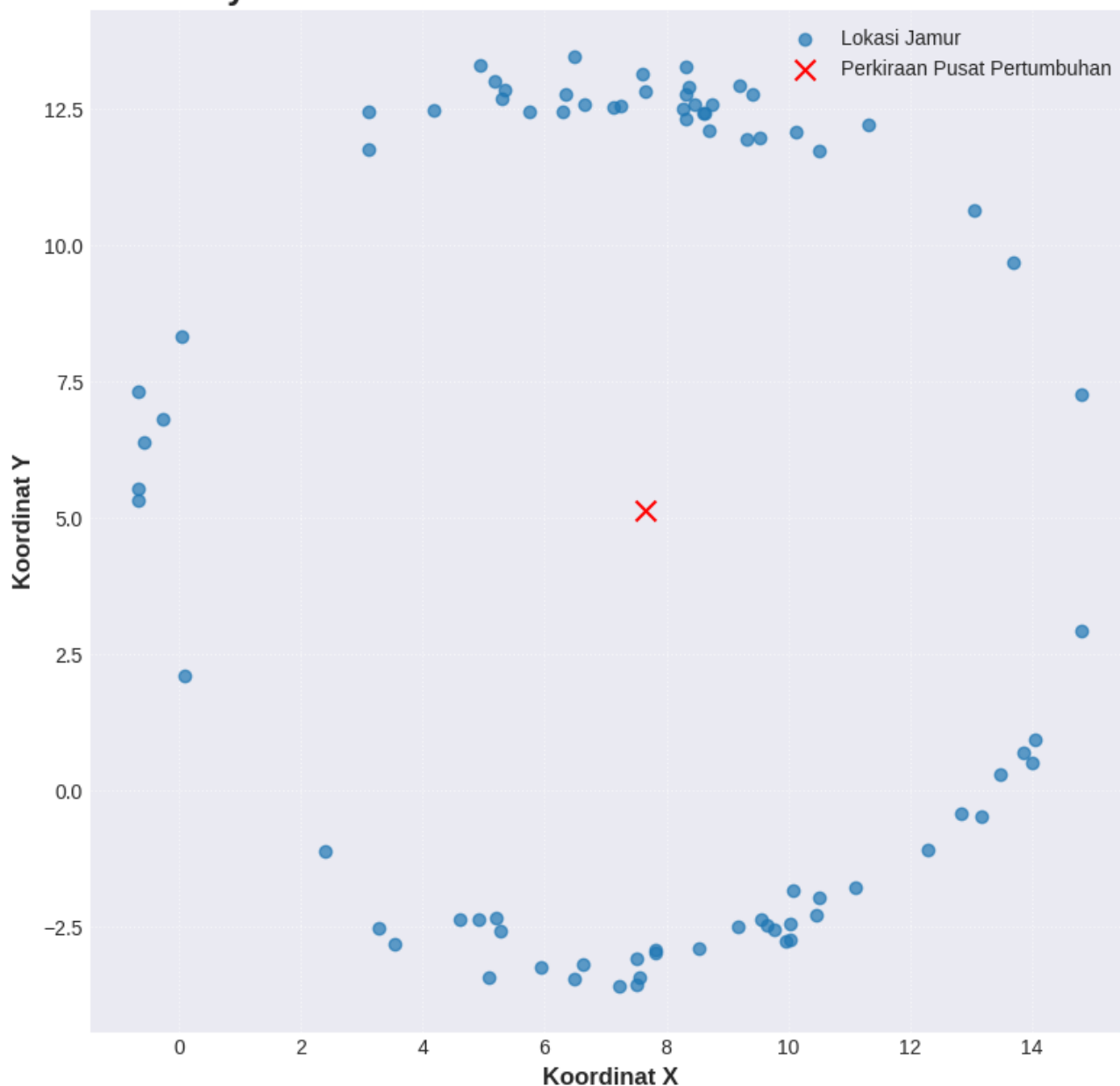
print(f'\nPerkiraan Koordinat Pusat Pertumbuhan Jamur: (x: {center_x:.2f}, y:
{center_y:.2f})')
```

Perkiraan Koordinat Pusat Pertumbuhan Jamur: (x: 7.64, y: 5.13)

## 5.2. Analisis Pemilihan Chart

*Chart apa yang anda pilih untuk problem diatas dan mengapa anda memilih chart tersebut?*

## Penyerbaran Jamur dan Perkiraan Pusat Pertumbuhan



Untuk memvisualisasikan lokasi jamur yang ditemukan dan memperkirakan pusat pertumbuhannya, jenis grafik yang paling tepat dan fundamental adalah **Scatter Plot**. Pemilihan Scatter Plot didasarkan pada kemampuannya yang inheren dalam merepresentasikan hubungan antara dua variabel numerik (dalam kasus ini, koordinat X dan Y) untuk setiap titik data (lokasi jamur). Berikut adalah alasan mendasar mengapa Scatter Plot menjadi pilihan yang sangat efektif untuk menjawab pertanyaan peneliti:

### 1. Representasi Distribusi Spasial

Scatter Plot secara langsung memetakan setiap pasangan koordinat (X, Y) sebagai sebuah titik pada bidang kartesian. Ini memberikan representasi visual yang intuitif tentang bagaimana jamur-jamur tersebut tersebar secara geografis. Pola-pola seperti pembentukan cincin yang disebutkan dalam deskripsi masalah akan menjadi jelas terlihat melalui visualisasi ini.

### 2. Identifikasi Pola dan Struktur

Dengan melihat sebaran titik-titik pada Scatter Plot, kita dapat mengidentifikasi adanya pola atau struktur tertentu dalam penyebaran jamur. Dalam kasus ini, deskripsi "cincin jamur" mengimplikasikan adanya pola melingkar atau elips. Scatter Plot memungkinkan kita untuk mengonfirmasi dan memvisualisasikan pola ini secara eksplisit.

### 3. Estimasi Pusat Secara Visual

Meskipun Scatter Plot itu sendiri tidak secara langsung memberikan koordinat pusat, visualisasi sebaran titik-titik yang membentuk cincin memungkinkan kita untuk secara intuitif memperkirakan area di mana pusat pertumbuhan jamur kemungkinan besar berada. Area dengan kepadatan titik yang lebih rendah di tengah cincin akan menjadi kandidat visual untuk pusat pertumbuhan.

### 4. Kemudahan Penambahan Informasi Tambahan

Scatter Plot sangat fleksibel untuk ditambahkan informasi visual lainnya. Dalam konteks ini, kita dapat dengan mudah menambahkan titik yang merepresentasikan perkiraan pusat pertumbuhan (seperti yang telah dilakukan dalam visualisasi ini) dengan warna dan simbol yang berbeda untuk membedakannya dari lokasi jamur sebenarnya.

### 5. Tidak Ada Asumsi Hubungan Fungsional

Tidak seperti grafik garis yang menyiratkan adanya hubungan fungsional atau tren antara variabel, Scatter Plot hanya menampilkan korelasi atau pola asosiasi antara dua variabel tanpa memaksakan asumsi tentang jenis hubungan matematis di antara mereka. Ini sangat sesuai untuk data lokasi spasial di mana kita tertarik pada kedekatan dan distribusi, bukan tren temporal atau hubungan sebab-akibat langsung.

Dalam kasus ini, karena kita memiliki data berupa koordinat lokasi dan ingin memahami penyebarannya serta memperkirakan pusatnya, Scatter Plot adalah pilihan yang paling mendasar dan efektif. Jenis grafik lain seperti Histogram atau Bar Chart tidak relevan karena kita tidak tertarik pada distribusi frekuensi satu variabel atau perbandingan nilai antar kategori

## 5.3. Koordinat (x, y) Pusat Pertumbuhan Jamur

*Koordinat pusat (x,y) pusat pertumbuhan jamur berada di?*

Untuk memperkirakan koordinat pusat pertumbuhan jamur berdasarkan data lokasi yang diberikan, pendekatan yang paling sederhana dan umum digunakan dalam analisis spasial awal adalah dengan menghitung **rata-rata (mean)** dari semua koordinat X dan rata-rata dari semua koordinat Y. Asumsi di balik metode ini adalah bahwa jika jamur menyebar secara radial dari satu titik pusat, maka rata-rata dari semua koordinat lokasi akan memberikan perkiraan yang cukup baik untuk pusat awal tersebut.

```
center_x = mushroom_df['x'].mean()
center_y = mushroom_df['y'].mean()
print(f'\nPerkiraan Koordinat Pusat Pertumbuhan Jamur: (x: {center_x:.2f}, y: {center_y:.2f})')
```

**Perkiraan Koordinat Pusat Pertumbuhan Jamur: (x: 7.64, y: 5.13)**

Oleh karena itu, berdasarkan metode rata-rata koordinat, **perkiraan koordinat pusat (x, y) pusat pertumbuhan jamur adalah sekitar (7.64, 5.13).**

#### Interpretasi:

Titik (7.64, 5.13) ini merupakan perkiraan lokasi pusat dari mana penyebaran jamur kemungkinan besar berasal. Pada Scatter Plot diatas, titik ini ditandai dengan simbol 'x' berwarna merah dan terletak di tengah pola cincin jamur yang terbentuk.

Meskipun metode rata-rata ini memberikan perkiraan awal yang baik, ada beberapa pertimbangan dan metode yang lebih canggih yang dapat digunakan untuk memperkirakan pusat pertumbuhan dengan lebih akurat, terutama jika penyebaran jamur tidak benar-benar radial sempurna atau jika ada faktor-faktor lain yang memengaruhi pertumbuhan:

- **Metode Geometris:** Jika pola cincin sangat jelas, kita dapat mencoba mencari pusat geometris dari bentuk tersebut, misalnya dengan mengidentifikasi titik tengah dari diameter yang berbeda dan mengambil rata-ratanya.
- **Algoritma Clustering:** Algoritma seperti K-Means dapat digunakan untuk mencoba mengidentifikasi kelompok-kelompok jamur dan mungkin memberikan indikasi pusat pertumbuhan jika kita berasumsi bahwa kepadatan jamur akan lebih tinggi di sekitar pusat awal.
- **Analisis Jarak:** Kita dapat menghitung jarak setiap titik jamur ke berbagai kandidat pusat dan mencari titik yang meminimalkan total jarak atau rata-rata jarak ke semua titik jamur.
- **Model Pertumbuhan:** Jika kita memiliki pemahaman yang lebih mendalam tentang model pertumbuhan jamur (misalnya, tingkat penyebaran per waktu), kita dapat mencoba memodelkan proses pertumbuhan mundur untuk memperkirakan pusat awal.

Namun, untuk tujuan soal ini dan dengan visualisasi **Scatter Plot** yang jelas menunjukkan pola cincin, perkiraan pusat pertumbuhan berdasarkan rata-rata koordinat memberikan jawaban yang masuk akal dan mudah diinterpretasikan. Koordinat **(7.64, 5.13)** adalah perkiraan terbaik kita untuk pusat pertumbuhan jamur berdasarkan data yang tersedia.