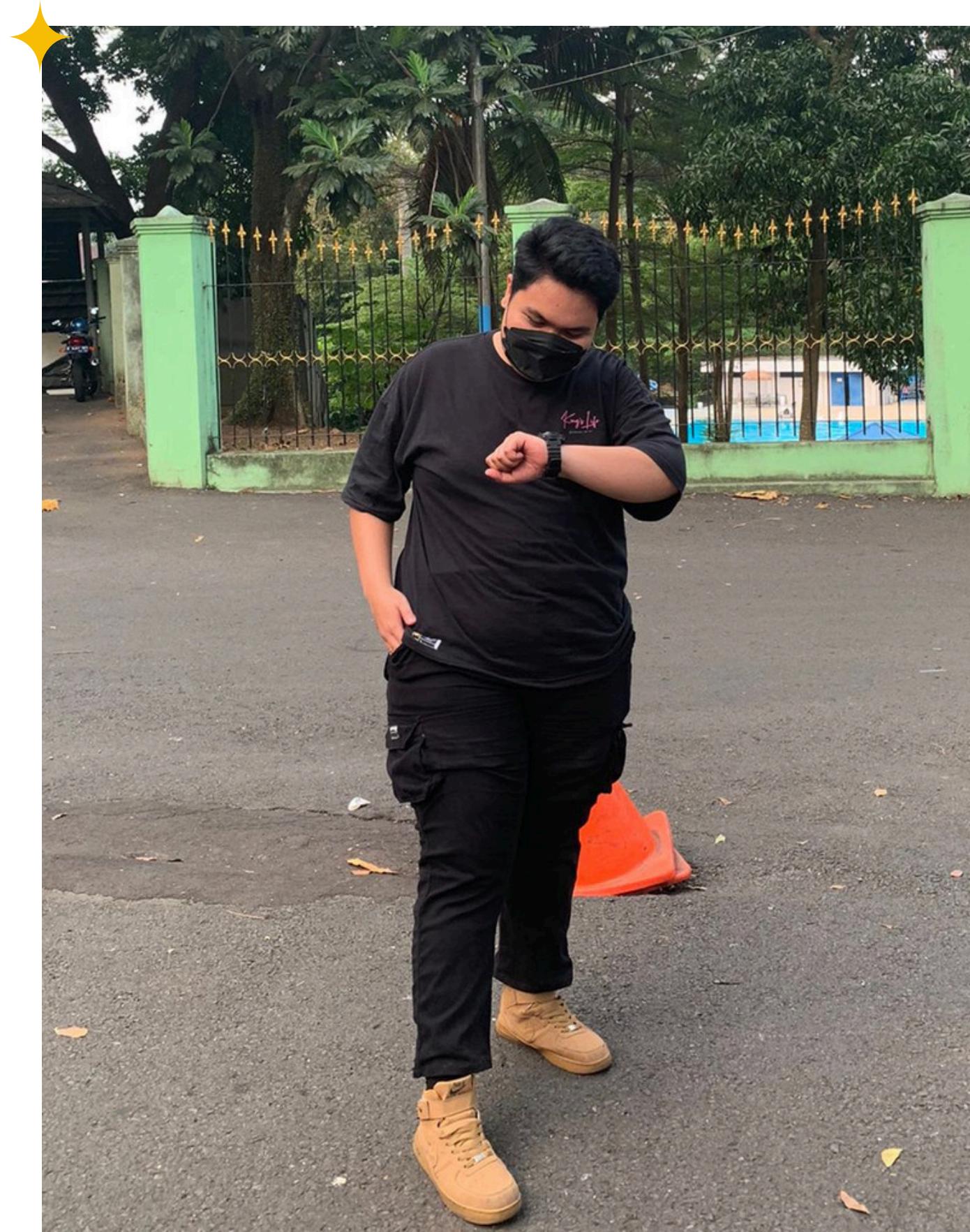

Portofolio.

Hello.





About Me.

Hi! I'm Adam Maulana, an Data enthusiast with a strong passion for analysis, digital content, and visualization. Experienced in data management and storytelling to support decisions. Adaptable, communicative, and quick to learn new tools.

Education & Experience

Feb - 2023



Oct 2021 – Nov 2022



Oct 2023 – Des 2024

FREELANCE

Jun 2025



Graduate From
Universitas Nasional
B.Sc. in
Agrotechnology

Join
PT. Harcoselaras
Sentosajaya as
Administrative Staff

Handled
Content creation
and management
for Instagram
platform

Finished
Full Stack
Data Analyst
Program Batch 26
[Certificate click here](#)

★ Data
Analyst.

```

SELECT
    c.city,
    AVG(basket_size) AS average_basket_size
FROM (
    SELECT
        o.order_number,
        SUM(od.quantity_ordered) AS basket_size
    FROM orders o
    JOIN orderdetails od ON o.order_number = od.order_number
    GROUP BY o.order_number
) AS transaction_sizes
JOIN customers c ON c.customer_number = (
    SELECT customer_number
    FROM orders
    WHERE orders.order_number = transaction_sizes.order_number
)
GROUP BY c.city
ORDER BY average_basket_size DESC;

```

Output:

customers 1

SELECT c.city, AVG(basket_size) AS average_basket_size

	Grid	A-z city	123 average_basket_size
Grid	1	Manchester	592.66666666667
Text	2	Bergamo	550
Text	3	Genève	539
Text	4	San Diego	477
Text	5	Lyon	476

Objective:

To calculate the average basket size per city based on customer transactions.

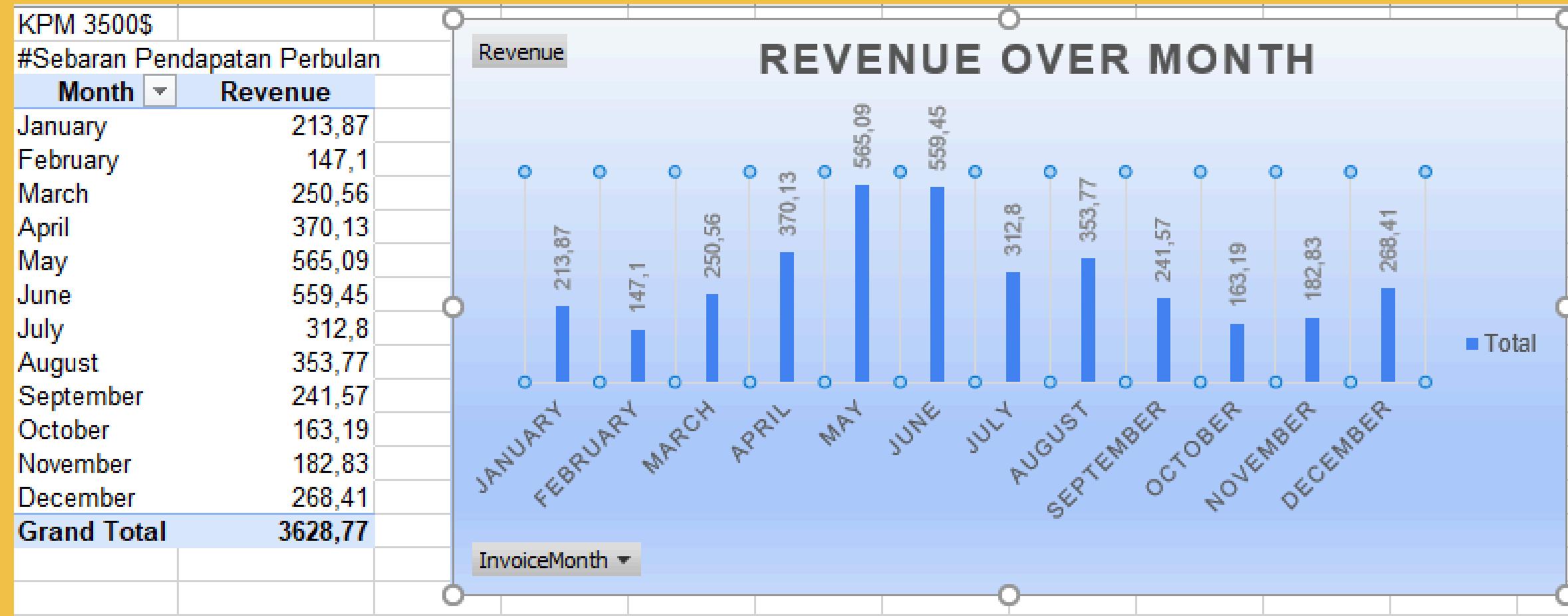
Output Insights:

- The table shows that Manchester has the highest average basket size: ~592.67
- Other high-ranking cities include Bergamo, Genève, and San Diego

Use Case:

- Useful for understanding regional purchasing behavior.
- Can inform targeted marketing, inventory distribution, and sales strategies based on city-level demand.

Pivot Table



Key Insights

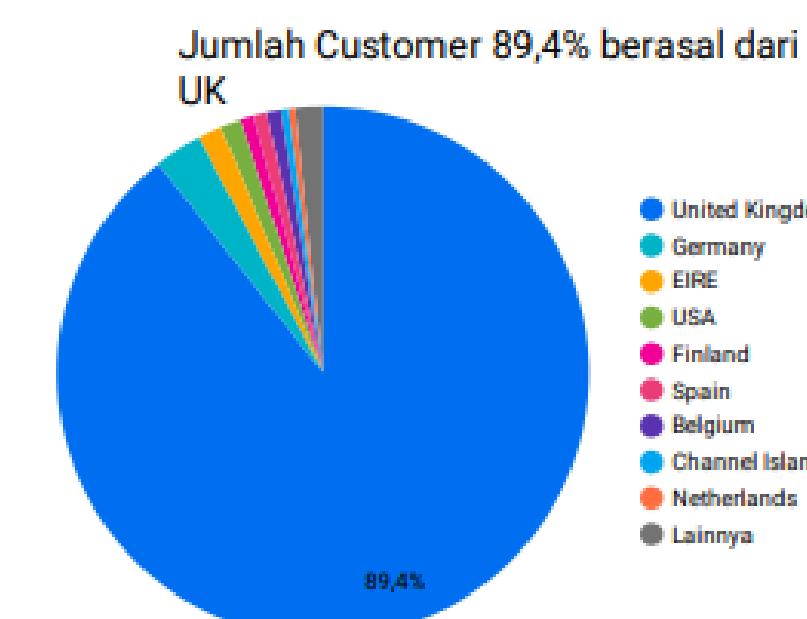
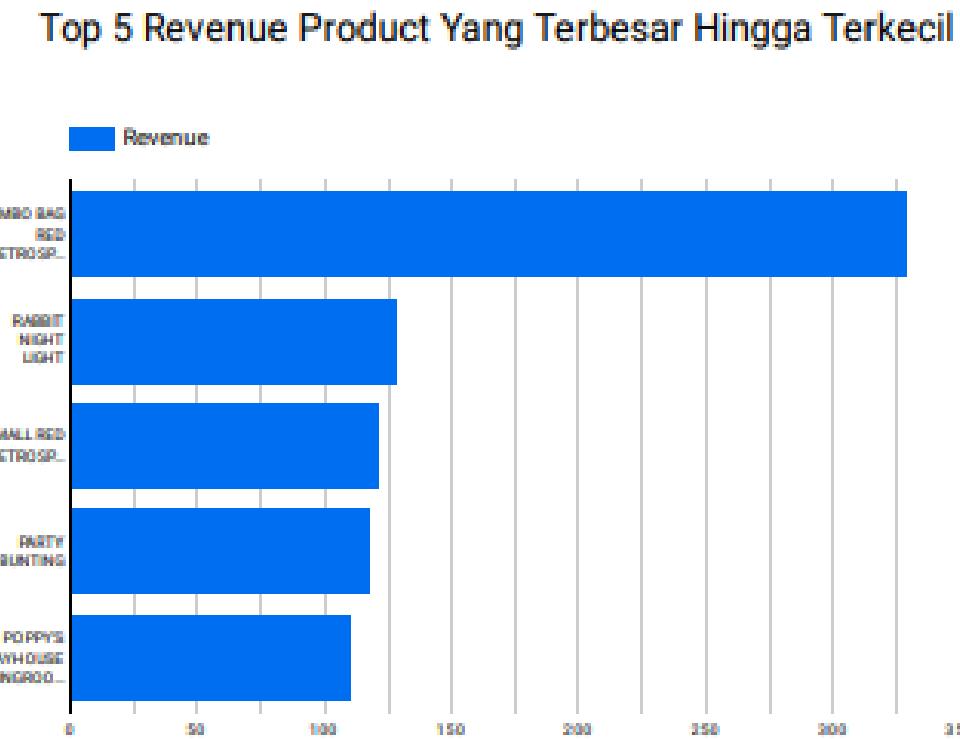
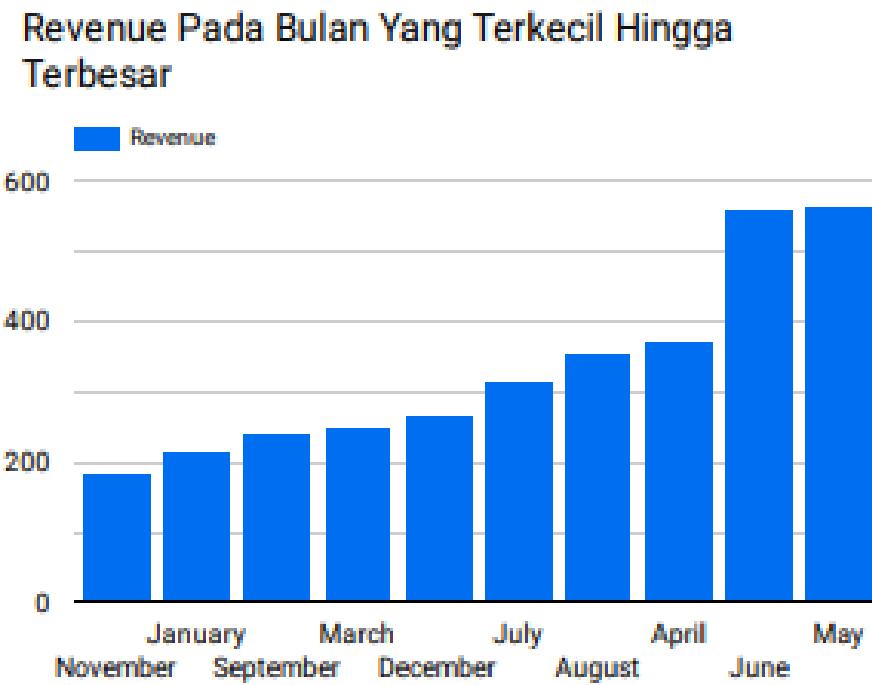
- Highest revenue in May (565.09) and June (559.45) – likely due to seasonal or promotional factors.
- Lowest revenue in February (147.1) and October (163.19) – potential for improvement.
- Revenue peaks mid-year, then declines steadily.
- Total annual revenue: 3,628.77.

Recommendations

- Replicate successful strategies from May–June in other months.
- Run promotions during February and October to boost sales.
- Prepare resources for Q2, when demand is highest.

DASHBOARD

Adam Maulana



Pivot Table Dashboard

This dashboard summarizes key insights based on over 500,000 rows of data using PivotTables and visualizations.

This dashboard provides a clear overview of:

1. Top-performing months and products
2. Most effective marketing sources
3. Primary customer geography
4. It helps stakeholders focus on high-performing areas and identify opportunities for optimization and growth.

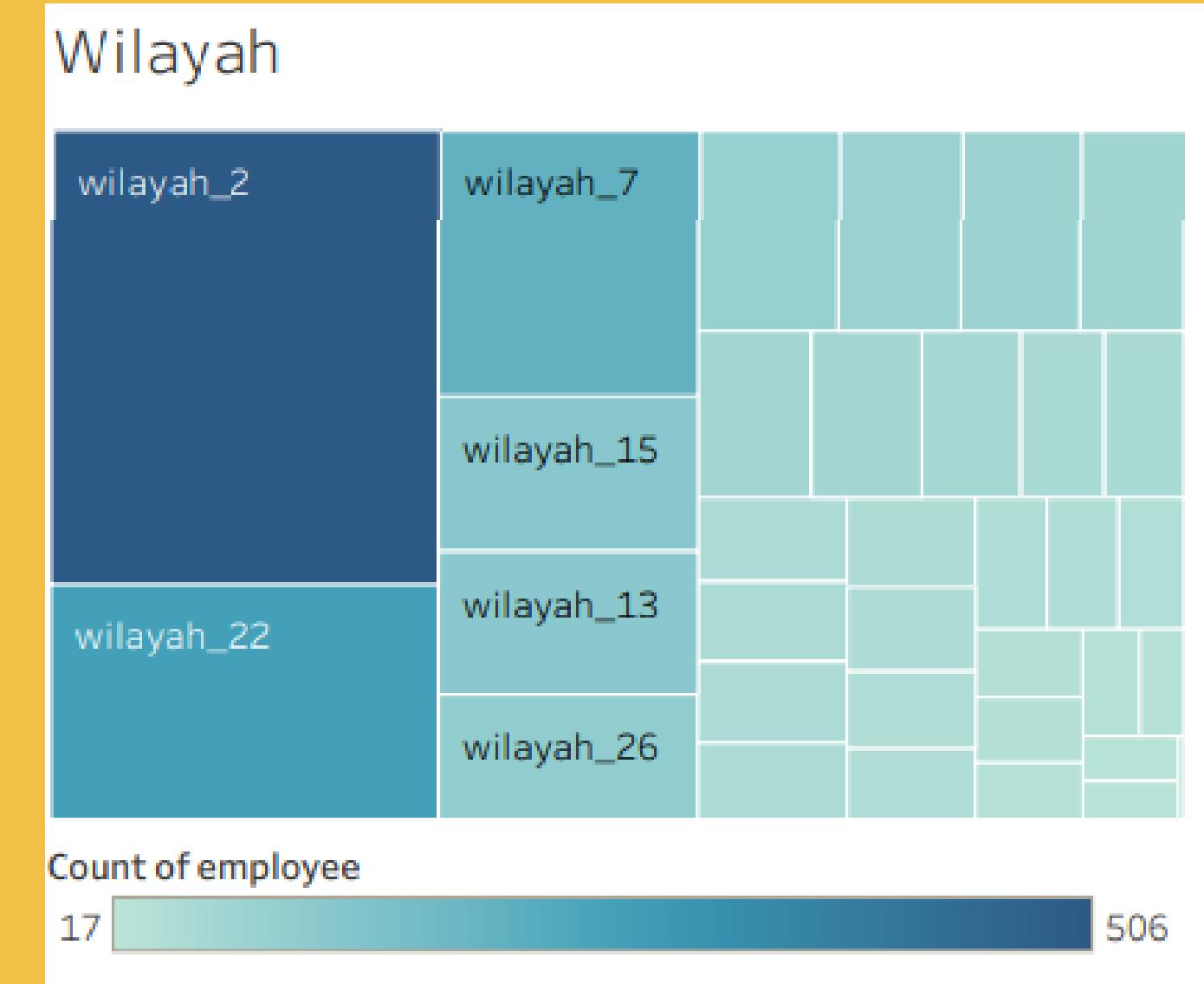
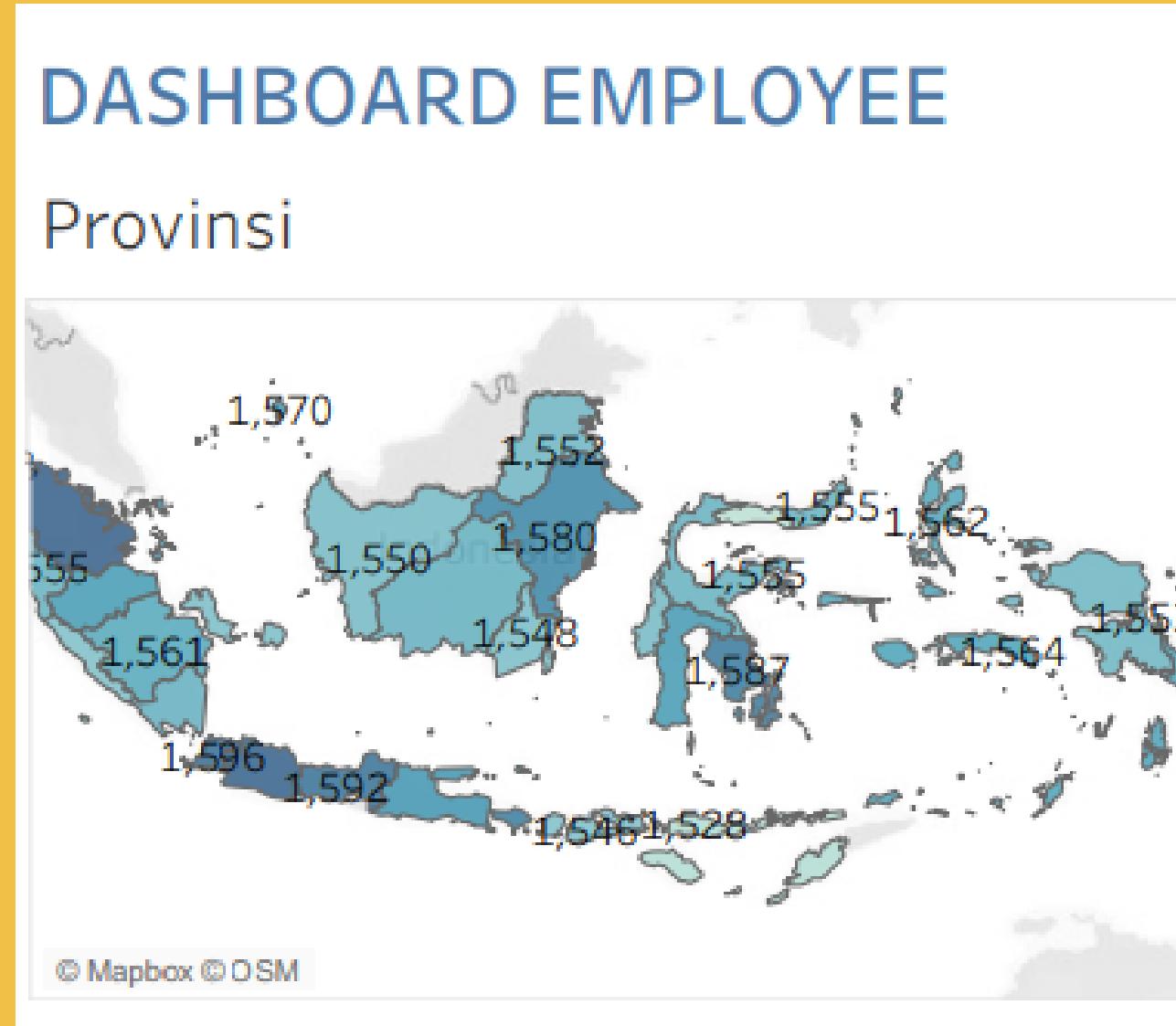


Tableau Dashboard

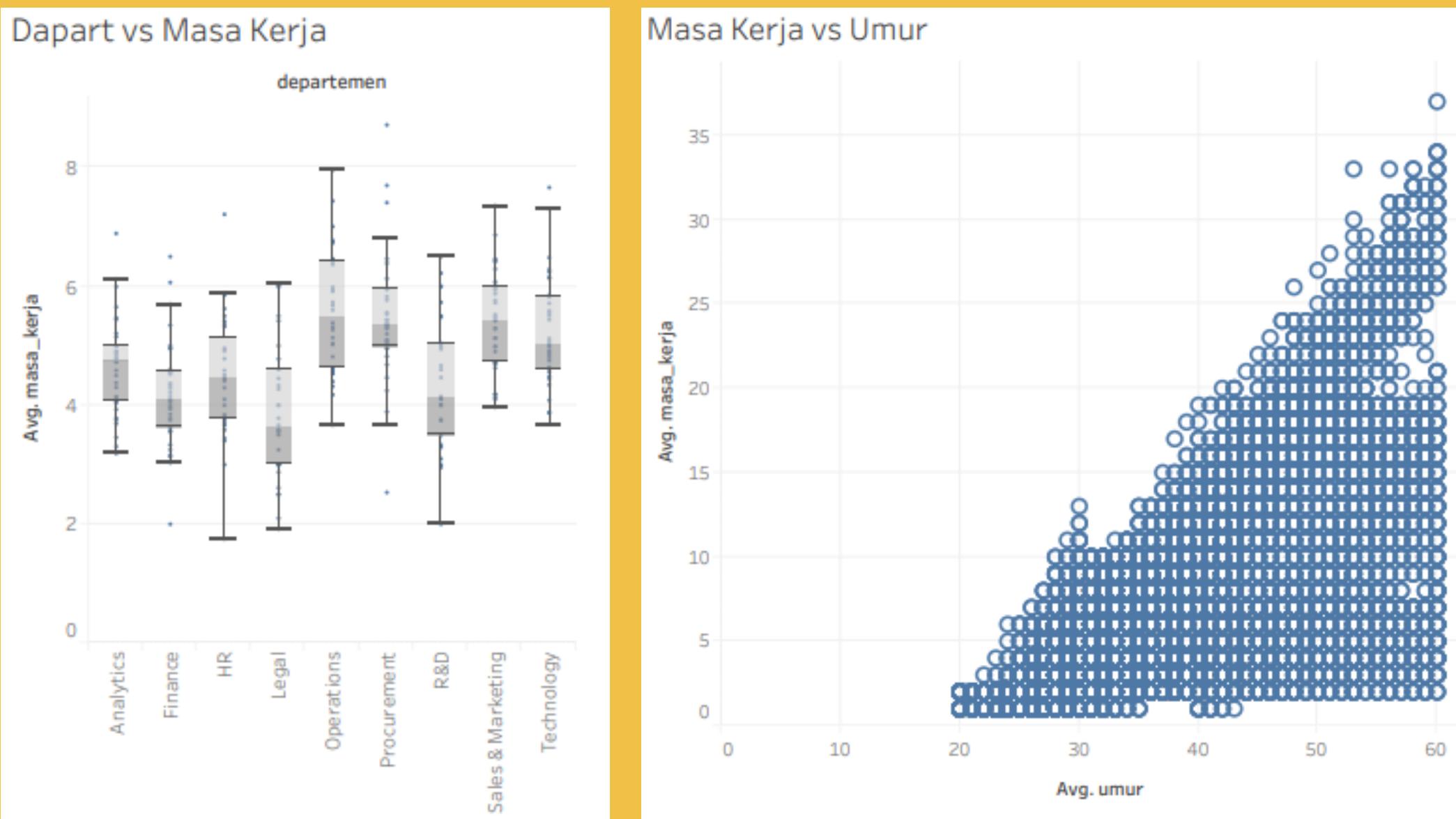
Built with Tableau, this dashboard provides a clear and interactive overview of employee concentration, helping stakeholders with regional workforce planning and resource allocation.

This dashboard, created using Tableau, visualizes the employee distribution across provinces and regions in Indonesia.

- The map view shows provinces such as Central Java (1,580 employees), West Java (1,570), and South Sulawesi (1,558) as having the highest number of employees.
- The treemap on the right highlights that Wilayah_2 and Wilayah_22 are the most employee-dense regions, with the top count reaching 506 employees.

Tableau Dashboard

Adam Maulana



Built with Tableau, this dashboard allows HR and management teams to monitor workforce maturity, analyze departmental retention, and review hiring patterns for strategic workforce planning.

Key Findings:

1. Tenure by Department (Left Chart):
 - Departments such as Legal, Procurement, and Technology show higher average years of service.
 - HR and Finance have the lowest average tenure among all departments.
2. Tenure vs. Age (Middle Chart):
 - There's a strong positive correlation between employee age and years of service.
 - Employees aged 50+ tend to have 25–35 years of experience.
3. Recruitment Trend (Right Chart):
 - A spike in recruitment occurred in Year 4 with over 9,000 employees hired, compared to a consistent ~4,982 in other years.
 - Indicates a major hiring phase or organizational expansion during that period.

[Click Here To See Full Coding.](#)

Python

```
[2]: df_sales = pd.read_csv('Sales Transaction v.4a.csv') # membaca file
```

▼ Data Cleansing

```
[3]: # 1. Ubah kolom Date menjadi tipe datetime
df_sales['Date']= pd.to_datetime(df_sales['Date'])

# 2. Buang semua transaksi yang memiliki quantity negative atau yang TransactionNo diawali dengan C
df_sales = df_sales[df_sales['TransactionNo'].str[0]!='C']
df_sales
```

This dataset was utilized during my Intermediate Data Analyst certification program at Growia, where I applied various analytical techniques as part of the final project.

Adam Maulana

	TransactionNo	Date	ProductNo	ProductName	Price	Quantity	CustomerNo	Country
0	581482	2019-12-09	22485	Set Of 2 Wooden Market Crates	21.47	12	17490.0	United Kingdom
1	581475	2019-12-09	22596	Christmas Star Wish List Chalkboard	10.65	36	13069.0	United Kingdom
2	581475	2019-12-09	23235	Storage Tin Vintage Leaf	11.53	12	13069.0	United Kingdom
3	581475	2019-12-09	23272	Tree T-Light Holder Willie Winkie	10.65	12	13069.0	United Kingdom
4	581475	2019-12-09	23239	Set Of 4 Knick Knack Tins Poppies	11.94	6	13069.0	United Kingdom
...
536320	536585	2018-12-01	37449	Ceramic Cake Stand + Hanging Cakes	20.45	2	17460.0	United Kingdom
536321	536590	2018-12-01	22776	Sweetheart 3 Tier Cake Stand	20.45	1	13065.0	United Kingdom
536322	536590	2018-12-01	22622	Box Of Vintage Alphabet Blocks	20.45	2	13065.0	United Kingdom
536323	536591	2018-12-01	37449	Ceramic Cake Stand + Hanging Cakes	20.45	1	14606.0	United Kingdom
536324	536597	2018-12-01	22220	Cake Stand Lovebird 2 Tier White	20.45	1	18011.0	United Kingdom

527765 rows × 8 columns

page 11 of 21

Click Here To See Full Coding

Adam Maulana

```

# Menghitung Volume Transaksi
volume_transaksi = df_sales.groupby("ProductNo")["TransactionNo"].nunique().reset_index()
volume_transaksi.columns = ["ProductNo", "Volume"]

# Menghitung Total Revenue (Price * Quantity)
df_sales["Revenue"] = df_sales["Price"] * df_sales["Quantity"]
total_revenue = df_sales.groupby("ProductNo")["Revenue"].sum().reset_index()
total_revenue.columns = ["ProductNo", "Total_Revenue"]

# Menggabungkan kedua metrik
df_segmented = volume_transaksi.merge(total_revenue, on="ProductNo")

# Mengambil nama produk dari df_sales
product_names = df_sales[["ProductNo", "ProductName"]].drop_duplicates()

# Menggabungkan ProductName ke df_segmented
df_segmented = df_segmented.merge(product_names, on="ProductNo", how="left")

# Menentukan batas segmentasi (20% teratas, 20-80%, dan 20% terbawah)
quantile_volume = df_segmented["Volume"].quantile([0.2, 0.8])
quantile_revenue = df_segmented["Total_Revenue"].quantile([0.2, 0.8])

```

In this project, I performed product segmentation based on:

- Transaction Volume: The number of transactions involving a product, regardless of the quantity purchased per transaction.
- Total Revenue: The total dollar revenue generated from the sales of each product.

The segmentation was done using the following criteria:

- Total revenue is calculated as the product of item price and quantity sold.
- Products were segmented by transaction volume as follows:
 - The top 20% highest in transaction volume were labeled "Popular"
 - The middle 60% (20%-80%) were labeled "Normal"
 - The bottom 20% were labeled "Low"
- A similar rule was applied to total revenue:
 - Top 20% in revenue → "Popular"
 - Middle 60% → "Normal"
 - Bottom 20% → "Low"

```

# Menentukan kategori Volume
def categorize_volume(volume):
    if volume >= quantile_volume[0.8]:
        return "Popular"
    elif volume >= quantile_volume[0.2]:
        return "Normal"
    else:
        return "Low"

df_segmented["Volume_Category"] = df_segmented["Volume"].apply(categorize_volume)

# Menentukan kategori Revenue
def categorize_revenue(revenue):
    if revenue >= quantile_revenue[0.8]:
        return "Popular"
    elif revenue >= quantile_revenue[0.2]:
        return "Normal"
    else:
        return "Low"

df_segmented["Revenue_Category"] = df_segmented["Total_Revenue"].apply(categorize_revenue)

# Menentukan kategori final berdasarkan kombinasi Volume dan Revenue
def categorize_final(row):
    if row["Volume_Category"] == "Popular" and row["Revenue_Category"] == "Popular":
        return "Super Popular"
    elif row["Volume_Category"] == "Popular" or row["Revenue_Category"] == "Popular":
        return "Popular"
    elif row["Volume_Category"] == "Low" and row["Revenue_Category"] == "Low":
        return "Low"
    else:
        return "Normal"

df_segmented["Final_Category"] = df_segmented.apply(categorize_final, axis=1)

# Menampilkan beberapa baris pertama hasil segmentasi
df_segmented

```

Then, based on the combination of both segmentations:

- Products that were "Popular" in both transaction volume and total revenue were labeled "Super Popular"
- Products that were "Popular" in one metric and "Normal" in the other were labeled "Popular"
- Products that were "Low" in both metrics were labeled "Low"
- All other combinations were categorized as "Normal"

[Click Here To See Full Coding.](#)

```
# Ambil 5 produk dengan total revenue terbesar
top5_products_november_2019 = super_popular_nov_sorted.head(5)

# Atur ukuran plot
plt.figure(figsize=(14, 6))

# Buat barplot horizontal dengan 'hue'
sns.barplot(
    data=top5_products_november_2019,
    x="Total_Revenue", # Corrected column name
    y="ProductName",
    hue="ProductName", # Gunakan ProductName sebagai hue
    dodge=False, # Pastikan batang tidak tumpang tindih
    palette="viridis" # Skema warna
)

# Tambahkan teks spesifikasi (jumlah revenue) di setiap batang
for i, row in enumerate(top5_products_november_2019.itertuples()):
    plt.text(row.Total_Revenue + 0.02 * row.Total_Revenue, i, f"${row.Total_Revenue:.2f}", va='center')

# Hapus Legend karena hanya untuk pewarnaan
plt.legend([], [], frameon=False)

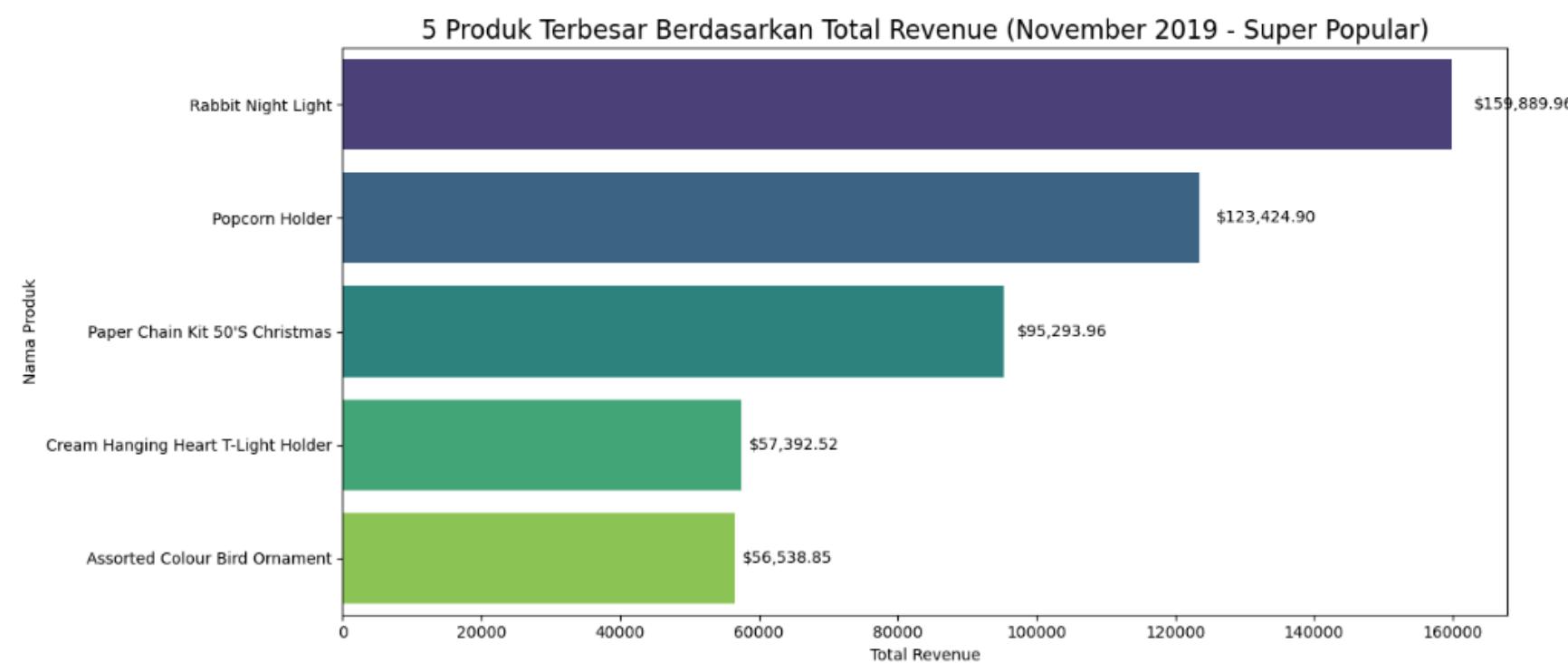
# Tambahkan judul dan Label
plt.title("5 Produk Terbesar Berdasarkan Total Revenue (November 2019 - Super Popular)", fontsize=16)
plt.xlabel("Total Revenue")
plt.ylabel("Nama Produk")

# Tampilkan plot
plt.tight_layout()
plt.show()
```

This script uses **Seaborn (sns)** for plotting a horizontal bar chart and **Matplotlib (plt)** for figure layout, annotations, and display.

It visualizes the top 5 products by total revenue for November 2019 from the "Super Popular" category, using a barplot with customized labels, colors, and annotations.

Python Data Visualitation

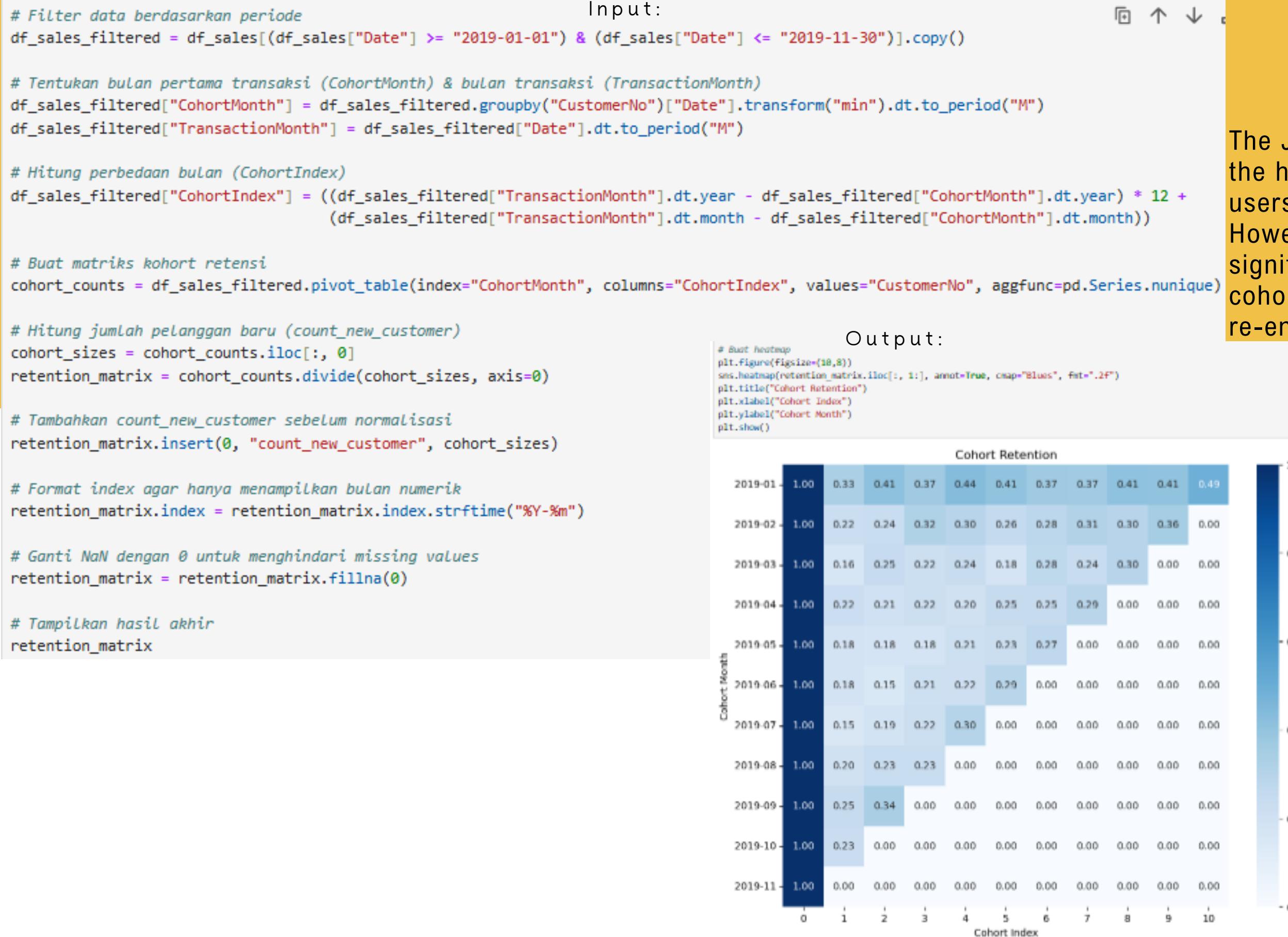


This chart can visually see which products made the most money in November 2019

- Rabbit Night Light is the product with the highest revenue: \$159,889.96 → the most successful product of the month.
- Followed by Popcor Holder and 50'S Christmas Paper Chain Kit, which also showed high sales performance.
- All products on the chart had revenue above \$57,000 → this is the informal threshold for "super popular" in this context.

[Click Here To See Full Coding.](#)

Python Retention Cohort Analysis



The January 2019 cohort showed the highest retention, with 64% of users remaining by month 2. However, retention dropped significantly after month 4 across all cohorts, indicating the need for a re-engagement strategy.

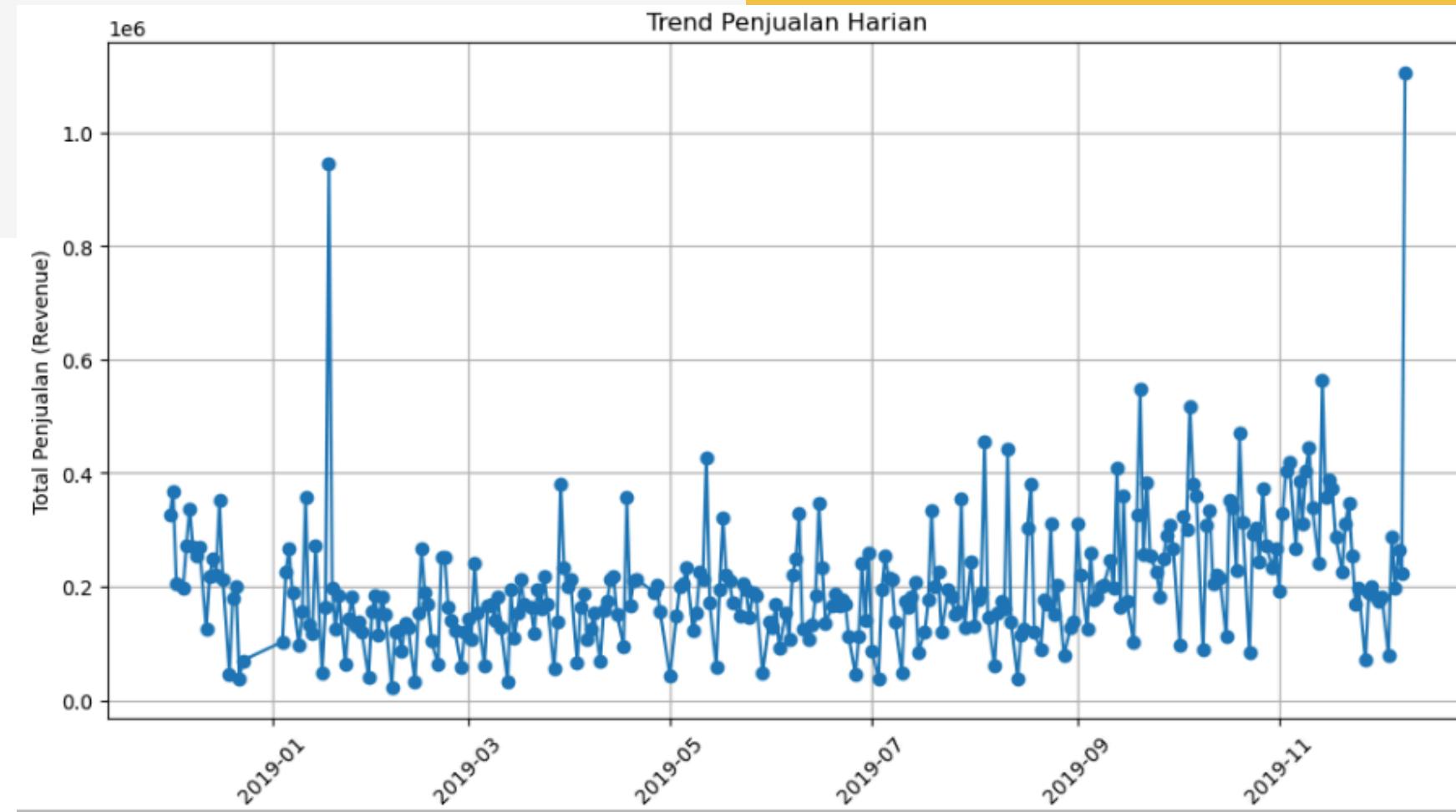
[Click Here To See Full Coding](#)

Python Time Series Analysis

Input:

```
# Nomor 1
# Agregasi total penjualan per hari
daily_sales = df_sales.groupby('Date')['Revenue'].sum()
# Plot trend penjualan harian
plt.figure(figsize=(12, 6))
plt.plot(daily_sales.index, daily_sales.values, marker='o', linestyle='-' )
plt.xlabel("Tanggal")
plt.ylabel("Total Penjualan (Revenue)")
plt.title("Trend Penjualan Harian")
plt.xticks(rotation=45)
plt.grid()
plt.show()
```

Output:



In overall terms, sales trends were seen to fluctuate with some sales peaks at certain times. There are some significant spikes, possibly related to certain promotions or campaigns. After a few spikes, the trend is seen to trend back down to a more stable level. Overall, sales trends do not show a consistent increase or decrease, but rather tend to fluctuate.

[Click Here To See Full Coding.](#)

Python Basket Size Analysis

```
# Tambahkan kolom total harga per baris produk
df_sales["TotalPrice"] = df_sales["Price"] * df_sales["Quantity"]

# Hitung total GMV per transaksi
gmv_per_transaction = df_sales.groupby(["DayOfWeek", "TransactionNo"]).agg({
    "TotalPrice": "sum",
    "Country": "first",
    "CustomerNo" : "first"
}).reset_index()

# Hitung GMV total dan jumlah transaksi per hari
basket_size_per_day = gmv_per_transaction.groupby("DayOfWeek").agg({
    "TotalPrice": ["sum", "mean", "count"]
}).reset_index()

basket_size_per_day.columns = ["Day", "Total_GMV", "Avg_Basket_Size", "Total_Transactions"]
```

Input:

```
ordered_days = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]

basket_size_per_day["Day"] = pd.Categorical(
    basket_size_per_day["Day"],
    categories=ordered_days,
    ordered=True
)

data = basket_size_per_day.set_index("Day").reindex(ordered_days)
y = data["Avg_Basket_Size"].values

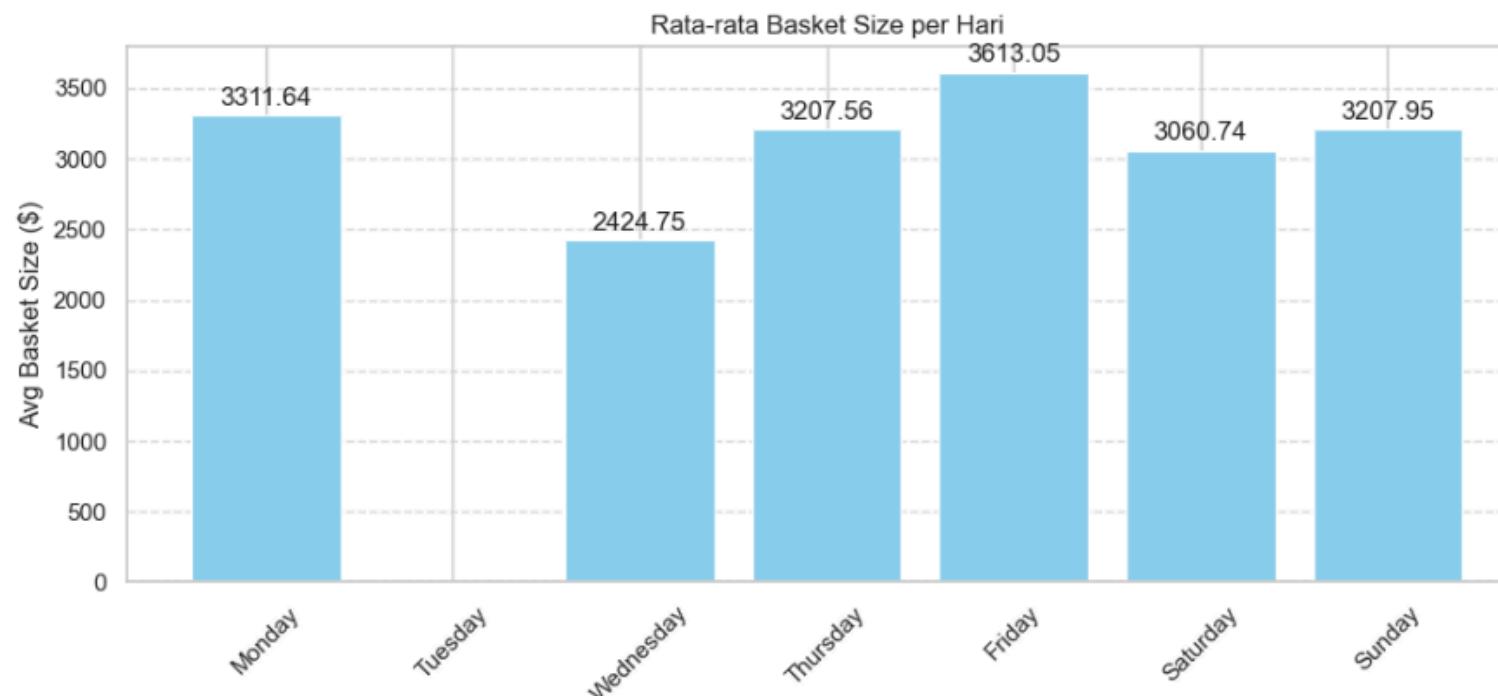
# Plotting seperti sebelumnya
import matplotlib.pyplot as plt
import numpy as np

plt.figure(figsize=(10, 5))
plt.bar(range(len(ordered_days)), y, color='skyblue')
plt.xlabel("Hari dalam Seminggu")
plt.ylabel("Avg Basket Size ($)")
plt.title("Rata-rata Basket Size per Hari")
plt.xticks(range(len(ordered_days)), ordered_days, rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)

for i, value in enumerate(y):
    if not np.isnan(value):
        plt.text(i, value + np.nanmax(y)*0.01, f"{value:.2f}", ha='center', va='bottom')

plt.tight_layout()
plt.show()
```

Output:



Consistent Weekday Pattern

Excluding Wednesday, weekdays—especially Monday, Thursday, and Friday—show high basket sizes (above \$3,200), indicating strong user activity at the start and end of the week.

Stable Weekend Performance

Saturday and Sunday maintain moderate basket sizes (\$3,060.74 and \$3,207.95), suggesting steady shopping activity during weekends.

★
**Content
Creator.**

Account

The image shows three Instagram account profiles arranged vertically:

- mobilelegendsverum**: 4.171 posts, 261 rb followers, 110 following. Bio: Verum Mobile Legends 🤴. Includes links to @kingsmlstore.woii, @markgamestore, and kingsmlstore.com.
- kingsmlstore.woii**: 1.162 posts, 69,4 rb followers, 139 following. Bio: TOPUP GENSIN | ML | FF | WUTHERING WAVES | HOK. Includes contact info: WA 081288115756, Order? Klik bawah!, and a link linktr.ee/kingsmlstore1.
- markgamestore**: 904 posts, 113 rb followers, 45 following. Bio: TOP #1 GLOBAL JOKI DAN TOPUP. Includes contact info: WA JOKI : 081311882316, WA TOPUP :082114904857, and a link linktr.ee/markgamestoreee.

Managed daily content creation for three social media accounts with a combined 440K+ followers, ensuring brand consistency and audience engagement. Utilized Canva and Instagram Insights to design, schedule, and optimize posts based on trends and performance analytics.



Content



Published 250+ content pieces across multiple platforms. Conducted weekly performance analysis using Instagram Insights to refine and enhance content strategy.

[check here to see all content](#)

Thank
you.

Contact Me



+ 62 87-880-925-800



[linkedin.com/in/adammaulana100](https://www.linkedin.com/in/adammaulana100)



adammaulana100@gmail.com