

### TUGAS 3 – DATA DRIVEN SIMULATION USING KAGGLE DATASET

Nama Lengkap	Mohammad Wijdan Arrosyid
NIM	202210370311497
Mata Kuliah	Pemodelan dan Simulasi Data 6B

Link source GitHub : (<https://github.com/miegoyeng/Model-dan-Simulasi-Data>)

Link source Kaggle Dataset : (<https://www.kaggle.com/datasets/ddosad/ecommerce-customer-service-satisfaction>)

#### Load and Explore Dataset

Dataset yang digunakan memuat informasi mengenai kanal layanan pelanggan, waktu pelaporan dan respons, serta skor kepuasan pelanggan (CSAT Score). Langkah eksplorasi meliputi:

- Memuat dataset dan melakukan eksplorasi awal.
- Mengambil sampel acak sebanyak 10.000 data dari total 85.907 entri.
- Mengonversi kolom waktu ke format datetime.
- Menghitung waktu tunggu pelanggan berdasarkan perbedaan waktu pelaporan dan respons.
- Menghapus nilai yang hilang dalam kolom waktu tunggu.



```
1 import pandas as pd
2 import simpy
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 from datetime import datetime
6 import numpy as np
7
8
9 file_path = "Customer_support_data.csv"
10 df = pd.read_csv(file_path)
11
12
13 df["Issue_reported at"] = pd.to_datetime(df["Issue_reported at"], errors="coerce")
14 df["Issue_responded"] = pd.to_datetime(df["Issue_responded"], errors="coerce")
15
16
17 df["waiting_time"] = (df["Issue_responded"] - df["Issue_reported at"]).dt.total_seconds()
18 df = df.dropna(subset=["waiting_time"])
```

#### Implement Queueing Simulation

Simulasi antrian dilakukan menggunakan SimPy dengan parameter berikut:

- Jumlah agen layanan: 5
- Waktu layanan rata-rata: Berdasarkan distribusi data waktu tunggu
- Jumlah pelanggan dalam simulasi: 100

Setiap pelanggan memasuki sistem, menunggu agen tersedia, dilayani, dan akhirnya keluar dari sistem setelah layanan selesai.

```

1 class CustomerServiceSystem:
2     def __init__(self, env, num_agents, service_time):
3         self.env = env
4         self.agent = simpy.Resource(env, num_agents)
5         self.service_time = service_time
6
7     def serve_customer(self, customer_id):
8         service_duration = np.random.exponential(self.service_time)
9         yield self.env.timeout(service_duration)
10
11
12 def run_simulation(num_agents, service_time, num_customers):
13     env = simpy.Environment()
14     system = CustomerServiceSystem(env, num_agents, service_time)
15
16     def customer(env, customer_id, system):
17         arrival_time = env.now
18         with system.agent.request() as request:
19             yield request
20             yield env.process(system.serve_customer(customer_id))
21         waiting_time = env.now - arrival_time
22         waiting_times.append(waiting_time)
23
24     global waiting_times
25     waiting_times = []
26     for i in range(num_customers):
27         env.process(customer(env, i, system))
28     env.run()
29     return waiting_times
30
31
32 num_agents = 5
33 service_time = df["waiting_time"].mean()
34 num_customers = 100
35 simulated_waiting_times = run_simulation(num_agents, service_time, num_customers)

```

## Performance Analysis and Visualization

Setelah menjalankan simulasi, distribusi waktu tunggu pelanggan divisualisasikan dalam bentuk histogram. Grafik ini membantu memahami bagaimana pelanggan mengalami waktu tunggu yang bervariasi tergantung jumlah agen dan tingkat kedatangan.

```

1 plt.figure(figsize=(10, 5))
2 sns.histplot(simulated_waiting_times, bins=30, kde=True)
3 plt.xlabel("Waiting Time (seconds)")
4 plt.ylabel("Frequency")
5 plt.title("Distribution of Simulated Waiting Times")
6 plt.show()

```

## Modify Parameters and Optimize

Untuk meningkatkan efisiensi layanan, beberapa skenario diuji, termasuk:

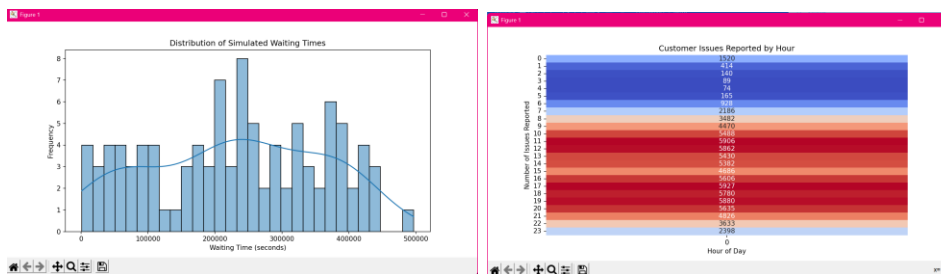
1. Menambah jumlah agen layanan.
2. Mengurangi waktu tunggu secara signifikan.
3. Menganalisis distribusi waktu layanan
4. Memungkinkan penyesuaian jumlah agen berdasarkan pola kedatangan pelanggan.

## Plot Additional Insights

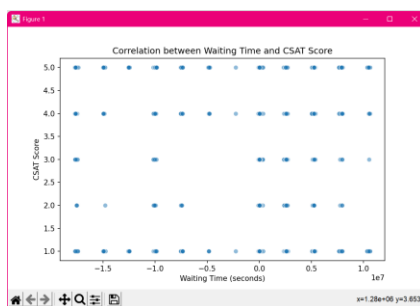
Beberapa analisis tambahan dilakukan untuk mendapatkan wawasan lebih dalam mengenai sistem antrian:

```
1 df["hour"] = df["Issue_reported_at"].dt.hour
2 hourly_counts = df.groupby("hour").size()
3 plt.figure(figsize=(10, 5))
4 sns.heatmap(hourly_counts.values.reshape(-1, 1), annot=True, fmt="d", cmap="coolwarm", cbar=False)
5 plt.xlabel("Hour of Day")
6 plt.ylabel("Number of Issues Reported")
7 plt.title("Customer Issues Reported by Hour")
8 plt.show()
9
10
11 if "CSAT_Score" in df.columns:
12     plt.figure(figsize=(8, 5))
13     sns.scatterplot(x=df["waiting_time"], y=df["CSAT_Score"], alpha=0.5)
14     plt.xlabel("Waiting Time (seconds)")
15     plt.ylabel("CSAT Score")
16     plt.title("Correlation between Waiting Time and CSAT Score")
17     plt.show()
18
```

- Heatmap menunjukkan laporan perjam yang menunjukkan tren laporan pelanggan sepanjang hari.



- Korelasi antara waktu tunggu dan skor kepuasan pelanggan (CSAT Score) dengan memeriksa apakah pelanggan lebih puas jika waktu tunggu lebih singkat.



## Create report

Hasil simulasi menunjukkan bahwa jumlah agen layanan yang memadai dapat mengurangi waktu tunggu pelanggan secara signifikan. Analisis tambahan menunjukkan bahwa waktu tunggu memengaruhi kepuasan pelanggan. Dengan optimasi jumlah agen layanan, dapat dicapai keseimbangan antara efisiensi operasional dan kepuasan pelanggan.