

Nama : Al Fitra Nur Ramadhani

NIM : 202210370311264

Mata Kuliah : Pemodelan dan Simulasi Data B

# Laporan Tugas 1

## Concert Ticket Booking System Simulation

### A. Deskripsi Proyek

Proyek ini bertujuan untuk mensimulasikan sistem antrian dalam platform pemesanan tiket konser menggunakan **SimPy**, sebuah pustaka Python untuk simulasi berbasis proses. Simulasi ini menganalisis kinerja sistem berdasarkan variasi jumlah server dan fase penjualan.

### B. Parameter Simulasi

- Jumlah Server : 2, 4, 6, dan 8 server
- Durasi Simulasi : 480 menit (8 jam).
- Fase Penjualan : Presale, Peak, Normal.
- Metrik Kinerja :

```
1 class ConcertTicketStats:
2     def __init__(self):
3         self.waiting_times = []
4         self.service_times = []
5         self.system_times = []
6         self.queue_lengths = []
7         self.time_stamps = []
8         self.server_busy_times = defaultdict(float)
9         self.server_utilization = defaultdict(float)
10        self.abandonment_count = 0
11        self.served_count = 0
12        self.total_customers = 0
```

Total Pelanggan	Rata-rata Waktu Layanan (menit)
Pelanggan Dilayani (%)	Rata-rata Waktu dalam Sistem (menit)
Pelanggan Batal (%)	Rata-rata Panjang Antrian (pelanggan)
Rata-rata Waktu Tunggu (menit)	Utilisasi Server (%)

### C. Ringkasan Hasil Simulasi

Simulation Results and Analysis Overview								
Analysis of customer service performance metrics								
Number of Servers	Total Customers	Customers Served (%)	Customers Canceled (%)	Avg. Wait Time (min)	Avg. Service Time (min)	Avg. Time in System (min)	Avg. Queue Length (customers)	Server Utilization (%)
2 Servers	1171	366 (31.3%)	788 (67.3%)	11.31	2.61	13.94	26.26	99.69
4 Servers	1122	683 (60.9%)	435 (38.8%)	4.87	2.62	7.50	17.73	93.27
6 Servers	1134	857 (75.6%)	272 (24.0%)	3.54	2.64	6.20	14.78	78.68
8 Servers	1143	949 (83.0%)	191 (16.7%)	2.27	2.72	4.99	9.89	67.12

## D. Analisis Hasil Simulasi

### 1. Pengaruh Jumlah Server

- **2 Server:**
  - Utilisasi server hampir 100%, menunjukkan beban kerja maksimum.
  - Tingkat pembatalan tinggi (67.3%) dan waktu tunggu rata-rata 11.31 menit, mengindikasikan antrean tidak efisien.
- **4 Server:**
  - Utilisasi turun menjadi 93.27%, dengan peningkatan pelanggan dilayani menjadi 60.9%.
  - Waktu tunggu turun signifikan (4.87 menit), tetapi antrean masih relatif panjang (17.73 pelanggan).
- **6 Server:**
  - Pembatalan berkurang drastis (24.0%) dengan waktu tunggu 3.54 menit.
  - Utilisasi server 78.68%, menunjukkan efisiensi yang lebih seimbang.
- **8 Server:**
  - Pelanggan dilayani mencapai 83.0% dengan waktu tunggu hanya 2.27 menit.
  - Utilisasi server 67.12%, mengindikasikan kapasitas berlebih jika permintaan tidak stabil.

### 2. Tren Utama

- **Peningkatan Jumlah Server:**
  - Mengurangi waktu tunggu, panjang antrian, dan tingkat pembatalan secara signifikan.
  - Menurunkan utilisasi server, tetapi meningkatkan kepuasan pelanggan.
- **Fase Peak:**
  - Transisi ke fase "peak" terjadi sekitar menit ke-60 di semua skenario, menunjukkan pola kedatangan yang konsisten.
- **Efisiensi Optimal:**
  - Pada 6 server, sistem mencapai keseimbangan antara utilisasi (78.68%) dan pelayanan (75.6% pelanggan dilayani).

## E. Kesimpulan

1. **2 Server tidak cukup** untuk menangani permintaan tinggi, menyebabkan antrean panjang dan pembatalan masif.
2. **4-6 Server** merupakan konfigurasi optimal untuk menyeimbangkan utilisasi dan kualitas layanan.
3. **8 Server** mungkin berlebihan untuk permintaan stabil, tetapi berguna jika terjadi lonjakan tak terduga.
4. **Penambahan server** secara linear meningkatkan kapasitas sistem, tetapi perlu dipertimbangkan biaya vs. manfaat.
5. **Fase peak** memerlukan alokasi sumber daya ekstra untuk meminimalkan pembatalan dan antrean.

## F. Rekomendasi

### 1. Jumlah Server yang Direkomendasikan

- **Baseline:** 4 server sebagai konfigurasi dasar untuk fase normal dan presale.
- **Fase Puncak:** Tambahkan server hingga 6 selama fase puncak untuk mengurangi pembatalan hingga 24.0% dan memangkas waktu tunggu ke 3.54 menit.
- **Fase Normal:** Kembalikan ke 4 server setelah puncak untuk optimasi biaya dan utilisasi (93.27%).

### 2. Strategi Pengelolaan Beban

- **Server Dinamis:** Implementasikan sistem yang menyesuaikan jumlah server otomatis berdasarkan trafik (misal: 6 server saat peak, 4 server saat normal).
- **Virtual Waiting Room:** Batasi akses pelanggan selama peak untuk menghindari lonjakan tiba-tiba. Contoh: 100 pelanggan/menit di fase peak.
- **Estimasi Waktu Tunggu:** Tampilkan perkiraan waktu tunggu di antarmuka pelanggan untuk mengurangi kecemasan dan pembatalan.

### 3. Optimasi Waktu Layanan

- **Sederhanakan Proses Pemesanan:** Kurangi langkah pemilihan kursi dengan opsi "random seat" untuk kategori ekonomi.
- **Quick Checkout:** Aktifkan fitur "1-click checkout" untuk pelanggan yang sudah terdaftar.
- **Pre-load Data:** Simpan informasi pelanggan sebelumnya (alamat, metode pembayaran) untuk mempercepat transaksi.

### 4. Mekanisme Antrian Khusus

- **Antrian Prioritas VIP:** Alokasikan 1-2 server khusus untuk pelanggan VIP agar waktu tunggu mereka <2 menit.
- **Batasi Pembelian per Transaksi:** Maksimal 4 tiket/transaksi selama fase peak untuk mencegah pembelian massal.
- **Waiting List:** Buka pendaftaran waiting list setelah tiket habis, lalu notifikasi pelanggan jika ada tiket batal.

### 5. Persiapan Infrastruktur

- **Scalable Cloud Server:** Gunakan layanan cloud yang dapat menambah kapasitas server secara instan selama peak (misal: AWS Auto Scaling).
  - **Uji Beban Berkala:** Lakukan simulasi rutin untuk memastikan sistem siap menghadapi lonjakan 2x lipat dari prediksi.
  - **Backup Database:** Pastikan replikasi database real-time untuk menghindari kegagalan sistem saat traffic tinggi.
- 

Link GitHub : [DATA-MINING/THEORY/ConcertTicketBookingSystem at main · alfitrannurr/DATA-MINING](https://github.com/alfitrnurr/DATA-MINING/THEORY/ConcertTicketBookingSystem)