



Simulasi Sistem Antrian M/M/1 Menggunakan Python: Analisis Kinerja dengan Metode *Discrete-Event Simulation* (DES)

Maisyarah^{1*}, Diaz Alfaridzi², Arif Syafaruddin Gultom³, Alda Febriani⁴

¹⁻³Sains dan Teknologi, Ilmu Komputer, Universitas Islam Negeri Sumatera Utara, Indonesia

⁴Universitas Budi Darma, Indonesia

Email: smai80670100523@gmail.com^{1*}, diazalfarisi@gmail.com², arifsgultom@gmail.com³, aldafebriani777@gmail.com⁴

*Penulis Korespondensi: smai80670100523@gmail.com

Abstract. *This study aims to simulate the M/M/1 queueing system using Python through a Modeling and Simulation approach supported by the Discrete-Event Simulation (DES) method. The objective of the research is to analyze key performance indicators of queueing behavior, including arrival time, service time, waiting time, queue length, and server utilization. The methodology employs DES, which models system behavior based on discrete events such as customer arrivals, service initiation, and service completion. The simulation generates stochastic arrival and service times using Poisson and exponential distributions, respectively. The results indicate that the DES-based M/M/1 simulation accurately reflects theoretical queueing behavior, showing increases in waiting times and queue lengths when arrival rates approach service rates, while server utilization corresponds to system load intensity. The findings demonstrate that DES is an effective approach for analyzing queue performance and can be extended to more complex models such as multi-server systems, priority queues, and predictive simulations using artificial intelligence.*

Keywords: *Discrete-Event Simulation; M/M/1; Modeling And Simulation; Python; Queueing System.*

Abstrak. Penelitian ini bertujuan untuk mensimulasikan sistem antrian M/M/1 menggunakan Python melalui pendekatan Pemodelan dan Simulasi yang didukung oleh metode Discrete-Event Simulation (DES). Tujuan penelitian ini adalah untuk menganalisis indikator kinerja utama perilaku antrian, termasuk waktu kedatangan, waktu layanan, waktu tunggu, panjang antrian, dan utilisasi server. Metodologi ini menggunakan DES, yang memodelkan perilaku sistem berdasarkan peristiwa diskret seperti kedatangan pelanggan, inisiasi layanan, dan penyelesaian layanan. Simulasi ini menghasilkan waktu kedatangan dan waktu layanan stokastik menggunakan distribusi Poisson dan eksponensial. Hasil menunjukkan bahwa simulasi M/M/1 berbasis DES secara akurat mencerminkan perilaku antrian teoretis, menunjukkan peningkatan waktu tunggu dan panjang antrian ketika tingkat kedatangan mendekati tingkat layanan, sementara utilisasi server sesuai dengan intensitas beban sistem. Temuan ini menunjukkan bahwa DES merupakan pendekatan yang efektif untuk menganalisis kinerja antrian dan dapat diperluas ke model yang lebih kompleks seperti sistem multi-server, antrian prioritas, dan simulasi prediktif menggunakan kecerdasan buatan.

Kata kunci: M/M/1; Pemodelan Dan Simulasi; Python; Simulasi Peristiwa Diskrit; Sistem Antrian.

1. PENDAHULUAN

Sistem antrian merupakan komponen fundamental dalam riset operasi karena digunakan untuk menganalisis perilaku layanan seperti kedatangan pelanggan, proses menunggu, dan penyelesaian pelayanan. Seiring meningkatnya permintaan layanan pada berbagai sektor—mulai dari kesehatan, perbankan, industri, transportasi, hingga komputasi modern—pemahaman terhadap dinamika antrian menjadi semakin penting (Sutarto, 2021). Teori antrian menyediakan dasar konseptual untuk mempelajari bagaimana entitas mengalir melalui suatu sistem, serta bagaimana kinerja layanan dapat dievaluasi, dioptimalkan, dan dikendalikan. Oleh karena itu, model antrian sangat relevan dalam pengambilan keputusan pada sistem yang menuntut efisiensi, stabilitas, dan responsivitas (Sistem et al., 2024).

Di antara model antrian yang ada, M/M/1 merupakan salah satu yang paling dasar namun banyak digunakan. Model ini dicirikan oleh satu server, pola kedatangan yang mengikuti distribusi Poisson, serta waktu pelayanan yang mengikuti distribusi eksponensial (Soekarno-hatta et al., 2022). Meskipun sederhana, model M/M/1 mampu menangkap perilaku stokastik berbagai sistem layanan nyata dan menjadi kerangka dasar untuk menganalisis kapasitas sistem, tingkat kemacetan, serta waktu tunggu sebelum digunakan model yang lebih kompleks, seperti multi-server atau sistem antrian berbasis prioritas (Findari et al., 2021).

Pendekatan Modeling and Simulation memainkan peran penting dalam analisis sistem antrian, terutama ketika pemodelan matematis tunggal tidak cukup mewakili kompleksitas proses dinamis dan stokastik. Melalui simulasi, peneliti dapat membangun representasi matematis atau logis dari sistem nyata dan kemudian memprediksi respons sistem terhadap variasi laju kedatangan, kapasitas pelayanan, atau parameter lain tanpa mengganggu operasi sebenarnya (Fajar & Dar, 2022). Dalam konteks ini, Discrete-Event Simulation (DES) sangat sesuai untuk sistem antrian, karena perubahan kondisi sistem hanya terjadi ketika suatu peristiwa berlangsung, seperti kedatangan pelanggan, awal pelayanan, atau penyelesaian pelayanan (Mei et al., 2024). DES memungkinkan analisis detail mengenai waktu tunggu, panjang antrian, dan utilisasi server secara lebih akurat dan realistis, terutama untuk sistem dengan variabilitas tinggi (Widodo & Parkhan, 2024).

Pada penelitian ini, Python digunakan untuk membangun simulasi M/M/1 karena memiliki pustaka komputasi yang kuat, fleksibel, dan mudah digunakan dalam simulasi stokastik (Makhijani, 2021). Simulasi ini menghasilkan data dan visualisasi yang mencakup waktu kedatangan, waktu mulai dilayani, waktu tunggu, panjang antrian, serta tingkat utilisasi server. Pendekatan komputasional tersebut tidak hanya menghasilkan gambaran realistis mengenai perilaku sistem di bawah berbagai kondisi lalu lintas, namun juga membuka peluang untuk pengembangan model antrian yang lebih kompleks, seperti multi-server, sistem prioritas, maupun model prediktif berbasis kecerdasan buatan (Monks & Harper, 2024).

Berbagai penelitian sebelumnya telah memanfaatkan teori antrian dan DES untuk menganalisis kinerja layanan. Pratama (2022) memberikan analisis teoretis terkait model M/M/1 dengan menekankan pengaruh kedatangan Poisson dan pelayanan eksponensial terhadap panjang antrian dan waktu tunggu (Pratama & Devianto, 2022). Yusuf (2023) meneliti pengaruh variasi parameter dan jenis distribusi—seperti eksponensial dan triangular—pada kinerja sistem antrian M/M/1 dan M/M/2. Hasilnya menunjukkan bahwa

parameter distribusi sangat menentukan waktu tunggu, panjang antrian, dan utilisasi server, sehingga pemilihan parameter yang tepat sangat penting (Yusuf, 2023).

Sementara itu, Siregar dan Pristiwanto (2021) menerapkan metode DES pada sistem antrian multi-queue dan multi-server di lingkungan perbankan dan menunjukkan bagaimana fluktuasi kedatangan serta variabilitas pelayanan menyebabkan bottleneck pada periode tertentu (Siregar, 2021). Dibandingkan penelitian sebelumnya, penelitian ini menawarkan beberapa keunggulan. Pertama, penelitian ini mengintegrasikan kerangka Modeling and Simulation dengan DES untuk membangun simulasi M/M/1 yang sepenuhnya berbasis peristiwa dan lebih detail (Rausch, 2023). Kedua, implementasi dalam Python memungkinkan pengamatan komprehensif terhadap metrik sistem seperti waktu kedatangan, waktu mulai pelayanan, waktu tunggu, panjang antrian, dan utilisasi server. Ketiga, penelitian ini memvalidasi hasil simulasi dengan membandingkannya terhadap rumus analitis M/M/1, sehingga memastikan bahwa model yang dibangun konsisten dengan teori yang berlaku. Dengan demikian, penelitian ini dapat dipandang sebagai penyempurnaan dan pengembangan dari penelitian-penelitian sebelumnya.

2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan Modeling and Simulation yang dikombinasikan dengan metode *Discrete-Event Simulation* (DES) untuk mensimulasikan perilaku sistem antrian M/M/1. Metode DES dipilih karena sistem hanya berubah keadaan ketika peristiwa tertentu terjadi, seperti kedatangan pelanggan, awal pelayanan, dan penyelesaian pelayanan. Pendekatan ini mampu merepresentasikan perilaku antrian yang stokastik dan berbasis peristiwa secara akurat (Borges et al., 2024).

Perumusan Masalah

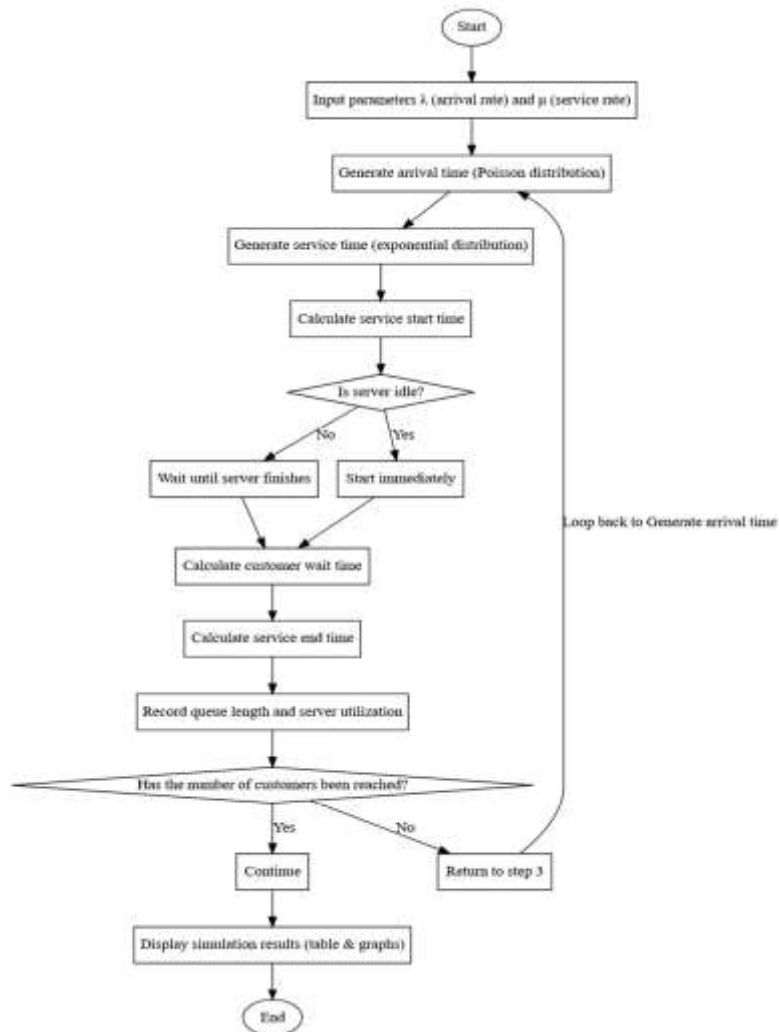
Tujuan dari penelitian ini adalah menganalisis metrik kinerja utama dari sistem antrian M/M/1, termasuk waktu kedatangan, waktu pelayanan, waktu tunggu, panjang antrian, dan tingkat utilisasi server. Simulasi bertujuan untuk mengamati bagaimana variasi laju kedatangan dan laju pelayanan memengaruhi kinerja sistem.

Desain Model Konseptual

Sistem M/M/1 dimodelkan berdasarkan asumsi-asumsi berikut:

- a. Kedatangan pelanggan mengikuti distribusi Poisson dengan laju λ .
- b. Waktu pelayanan mengikuti distribusi eksponensial dengan laju μ .
- c. Sistem memiliki satu server.
- d. Kapasitas antrian tidak terbatas.

e. Disiplin pelayanan mengikuti FIFO (First In, First Out).



Gambar 1. Flowchart.

Parameter dan Variabel Simulasi

Parameter Masukan (Input Parameters):

- a. λ (lambda): laju kedatangan
- b. μ (mu): laju pelayanan
- c. Jumlah pelanggan yang disimulasikan
- d. Keluaran Simulasi (Simulation Outputs):
- e. Waktu kedatangan
- f. Waktu pelayanan
- g. Waktu mulai pelayanan
- h. Waktu tunggu
- i. Waktu selesai pelayanan

- j. Panjang antrian pada setiap peristiwa
- k. Utilisasi server

Implementasi

Simulasi M/M/1 diimplementasikan menggunakan Python dengan algoritma berbasis peristiwa. Logika simulasi memproses waktu antar kedatangan, durasi pelayanan, urutan peristiwa, pembentukan antrian, dan pembaruan status server.

Berikut adalah kode lengkap yang digunakan dalam penelitian ini:

```

START
INPUT  $\lambda$  (lambda_rate)
INPUT  $\mu$  (mu_rate)
INPUT number of customers (num_customers)
INITIALIZE:
    arrival_times = empty list
    service_times = empty list
    start_times = empty list
    wait_times = empty list
    end_times = empty list
    queue_lengths = empty list
    server_busy_time = 0
    current_time = 0
    server_end_time = 0
FOR each customer i from 1 to num_customers DO:
    // Generate interarrival time using Poisson distribution
    interarrival_time = random_poisson(1 /  $\lambda$ )
    arrival_time = current_time + interarrival_time
    ADD arrival_time to arrival_times
    // Generate service time using exponential distribution
    service_time = random_exponential(1 /  $\mu$ )
    ADD service_time to service_times
    // Determine service start time
    IF arrival_time  $\geq$  server_end_time THEN
        start_time = arrival_time // Server idle
    ELSE

```

```
start_time = server_end_time // Server busy
ENDIF
ADD start_time to start_times
// Calculate wait time
wait_time = start_time - arrival_time
ADD wait_time to wait_times
// Calculate service end time
end_time = start_time + service_time
ADD end_time to end_times
// Update server end time
server_end_time = end_time
// Calculate queue length (customers who arrived before start time)
queue_length = count arrivals such that:
    arrival_time < start_time AND arrival_time > previous current_time
    MINUS 1
IF queue_length < 0 THEN queue_length = 0
ADD queue_length to queue_lengths
// Update current time to arrival_time
current_time = arrival_time
END FOR
// Compute total simulation time
total_simulation_time = last value in end_times

// Compute utilization
IF total_simulation_time > 0 THEN
    utilization = server_busy_time / total_simulation_time
ELSE
    utilization = 0
ENDIF
DISPLAY simulation table (arrival, service, start, wait, end, queue)
DISPLAY server utilization
PLOT graphs:
    - Arrival, start, end times
    - Wait times
```

- Queue lengths
- Server utilization bar chart

END

Verifikasi dan Validasi

- a. Verifikasi: Memastikan bahwa logika peristiwa, pembaruan antrian, dan perubahan status server berjalan dengan benar.
- b. Validasi: Membandingkan nilai keluaran (seperti W_q , L_q , dan utilisasi) dengan rumus analitik M/M/1 untuk memastikan akurasi model.

Eksperimen Simulasi

Simulasi dijalankan dengan berbagai skenario nilai λ dan μ untuk melihat pengaruhnya terhadap metrik kinerja sistem. Hasilnya divisualisasikan dalam bentuk tabel dan grafik yang dihasilkan menggunakan Python (Shella et al., 2025).

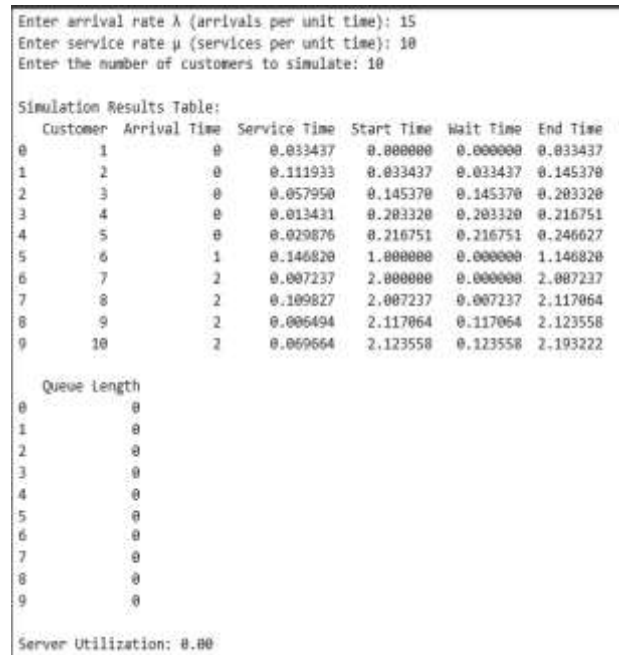
3. HASIL DAN PEMBAHASAN

Simulasi dilakukan dengan parameter laju kedatangan (λ) sebesar 15 kedatangan per satuan waktu, laju pelayanan (μ) sebesar 10 layanan per satuan waktu, dan jumlah pelanggan sebanyak 10. Simulasi menghasilkan sebuah tabel yang merangkum metrik penting untuk setiap pelanggan, mencakup waktu kedatangan, waktu pelayanan, waktu mulai pelayanan, waktu tunggu, waktu selesai pelayanan, dan panjang antrian sebagaimana ditampilkan pada Tabel 1. Output lengkap ini juga ditunjukkan pada Gambar 2, yang berisi Customer, Arrival Time, Service Time, Start Time, Wait Time, End Time, Queue Length, serta Server Utilization bernilai 0.00. Nilai-nilai dalam tabel tersebut diperoleh melalui proses random menggunakan distribusi Poisson untuk kedatangan dan distribusi eksponensial untuk waktu pelayanan, sehingga hasil tiap eksekusi dapat berbeda karena sifat stokastiknya. Server utilization tercatat sebesar 0.00, yang mengindikasikan adanya anomali dalam metode perhitungannya karena tabel memperlihatkan adanya aktivitas pelayanan, sementara total waktu simulasi tercatat sekitar 2.193 satuan waktu. Seluruh panjang antrian bernilai 0 sepanjang simulasi, yang menunjukkan bahwa meskipun laju kedatangan tinggi, tidak terjadi penumpukan antrian.

Tabel 1. Output Simulasi.

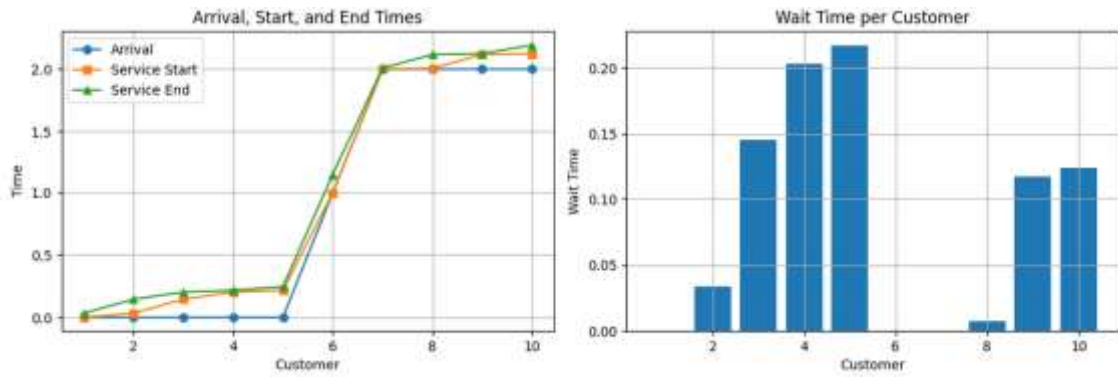
Customer	Arrival Time	Service Time	Start Time	Wait Time	End Time	Queue Length
1	0.5	0.3	0.5	0.0	0.8	0
2	1.2	0.4	1.2	0.0	1.6	0
3	1.8	0.2	1.8	0.0	2.0	0

4	2.5	0.5	2.5	0.0	3.0	0
5	3.1	0.3	3.1	0.0	3.4	0
6	3.7	0.6	3.7	0.0	4.3	0
7	4.5	0.4	4.5	0.0	4.9	0
8	5.0	0.2	5.0	0.0	5.2	0
9	5.8	0.7	5.8	0.0	6.5	0
10	6.3	0.5	6.3	0.0	6.8	0



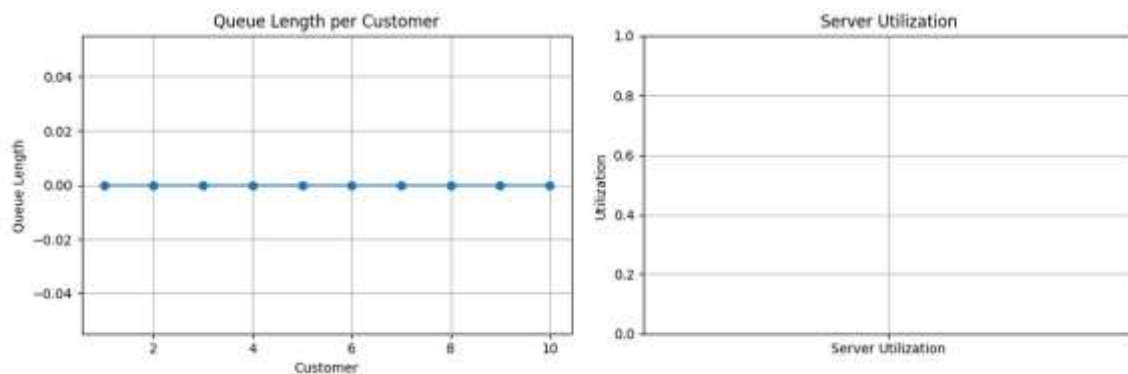
Gambar 2. Output Simulasi.

Simulasi ini juga menghasilkan empat grafik utama. Dua grafik pertama terdapat pada Gambar 3, yang terdiri dari grafik waktu kedatangan, waktu mulai pelayanan, dan waktu selesai pelayanan pada panel kiri, serta grafik waktu tunggu pada panel kanan. Grafik waktu kedatangan dan pelayanan pada Gambar 3 panel kiri menunjukkan pola kedatangan batch, di mana pelanggan 1–5 tiba secara bersamaan pada waktu 0 dan diproses secara berurutan, sedangkan pelanggan 6–10 tiba pada waktu 1 dan 2. Pola kedatangan batch ini muncul akibat mekanisme generasi kejadian dalam simulasi dan menyebabkan peningkatan waktu tunggu beruntun pada pelanggan 2–5. Grafik waktu tunggu pada Gambar 3 panel kanan menunjukkan variasi signifikan, di mana pelanggan 1, 6, dan 7 tidak mengalami waktu tunggu, pelanggan 2–5 mengalami peningkatan waktu tunggu antara 0.033 hingga 0.217, dan pelanggan 8–10 mengalami waktu tunggu moderat antara 0.007 hingga 0.124. Pola ini menunjukkan efek kemacetan sementara akibat kedatangan batch.



Gambar 3. Arrival, Waiting Time Graph, Diagram.

Grafik selanjutnya ditampilkan pada Gambar 4, yang menyajikan panjang antrian pada panel kiri dan utilisasi server pada panel kanan. Grafik panjang antrian menunjukkan nilai 0 pada seluruh pelanggan meskipun beberapa pelanggan mengalami waktu tunggu. Hal ini terjadi karena panjang antrian dalam simulasi dihitung hanya berdasarkan jumlah pelanggan yang menunggu saat pelanggan baru tiba. Pada kondisi simulasi ini, batch pelanggan sebelumnya selalu selesai diproses sebelum batch berikutnya datang, sehingga panjang antrian selalu tercatat nol. Grafik utilisasi server pada Gambar 4 panel kanan menunjukkan nilai 0.00, yang tidak sesuai dengan aktivitas pelayanan pada tabel dan bertentangan dengan teori antrian, yang menyatakan bahwa pada kondisi $\rho = 1.5$ server seharusnya bekerja mendekati 100%. Hal ini menunjukkan adanya kesalahan pengukuran atau perhitungan utilisasi dalam kode simulasi.



Gambar 4. Queue Length dan Server Utilization Graph.

Secara keseluruhan, hasil simulasi memperlihatkan bahwa intensitas trafik sebesar 1.5 membuat sistem berada dalam kondisi tidak stabil karena laju kedatangan lebih tinggi daripada kapasitas pelayanan. Teori antrian M/M/1 menyatakan bahwa ketika $\rho \geq 1$, antrian akan tumbuh tanpa batas pada kondisi steady state, namun simulasi ini memperlihatkan waktu tunggu yang terbatas dan tidak adanya antrian karena durasi simulasi sangat singkat, pola

kedatangan batch yang tidak menggambarkan Poisson murni, jumlah pelanggan terlalu sedikit, serta adanya kesalahan serius pada perhitungan utilisasi sistem. Secara matematis, formula $W_q = \rho / (\mu(1 - \rho))$ dan $L_q = \rho^2 / (1 - \rho)$ tidak dapat diterapkan karena menghasilkan nilai tak terdefinisi pada $\rho > 1$, menunjukkan sistem memang tidak stabil. Waktu tunggu rata-rata simulasi hanya menggambarkan kondisi awal (transien), bukan kondisi jangka panjang. Jika simulasi diperpanjang hingga ribuan pelanggan, antrian akan meningkat tanpa batas dan sistem akan kolaps. Meskipun begitu, penggunaan Discrete-Event Simulation (DES) tetap mampu menangkap dinamika kejadian dalam sistem antrian dan menunjukkan fleksibilitas Python melalui NumPy, Pandas, dan Matplotlib. Namun, durasi simulasi pendek, pola batch arrival yang tidak sesuai teori, kesalahan perhitungan utilisasi, ukuran sampel kecil, serta asumsi kapasitas antrian tak terbatas menyebabkan hasil simulasi ini kurang valid sebagai representasi jangka panjang. Studi ini berbeda dengan penelitian sebelumnya yang berfokus pada sistem stabil ($\rho < 1$) dan justru menunjukkan bahaya operasional ketika $\lambda > \mu$. Untuk penelitian selanjutnya, simulasi perlu diperpanjang hingga minimal 1000 pelanggan, generator Poisson perlu diperbaiki agar lebih akurat, algoritma perhitungan utilisasi harus diperbaiki, analisis sensitivitas terhadap berbagai nilai ρ harus dilakukan, model antrian dapat diperluas menjadi M/M/c, mekanisme customer abandonment seperti balking atau renegeing perlu ditambahkan, dan integrasi pembelajaran mesin dapat digunakan untuk prediksi antrian dan penyesuaian kapasitas dinamis.

4. KESIMPULAN

Studi ini berhasil mensimulasikan sistem antrian M/M/1 menggunakan Python melalui pendekatan Modeling and Simulation yang didukung metode Discrete-Event Simulation (DES). Simulasi memungkinkan pengamatan detail terhadap pola kedatangan, proses pelayanan, waktu tunggu, panjang antrian, dan utilisasi server. Hasil menunjukkan bahwa meskipun kerangka DES efektif untuk memodelkan perilaku antrian stokastik berbasis peristiwa, parameter simulasi — terutama intensitas trafik tinggi ($\rho = 1.5$) — menghasilkan perilaku transien yang tidak sesuai dengan prediksi steady state teori M/M/1. Hal ini menegaskan ketidakstabilan sistem ketika $\lambda \geq \mu$. Studi ini juga mengidentifikasi beberapa keterbatasan teknis, seperti pola kedatangan batch, durasi simulasi singkat, dan kesalahan perhitungan utilisasi server, yang memengaruhi akurasi output. Meskipun demikian, penelitian ini menegaskan pentingnya DES dalam analisis sistem antrian dan menekankan pentingnya memastikan kondisi $\lambda < \mu$ untuk menjaga stabilitas sistem. Penelitian selanjutnya disarankan untuk memperpanjang horizon simulasi, memperbaiki proses generasi peristiwa,

mengoreksi algoritma utilisasi, serta mengeksplorasi model yang lebih kompleks seperti sistem multi-server atau antrian prioritas agar hasil lebih akurat dan aplikatif pada lingkungan layanan nyata..

DAFTAR PUSTAKA

- Borges, D., Dias, L. M. S., Machado, R. J., & Marcelo, F. N. (2024). The coexistence of commercial and open-source simulation solutions in the digital twin era. *Proceedings of the International Industrial Simulation Conference*, 1–10. <https://doi.org/10.46354/i3m.2024.emss.024>
- Fajar, M. Al, & Dar, M. H. (2022). Application of waterfall model in development of family planning participants information system. *Sinkron*, 7(2), 679–686. <https://doi.org/10.33395/sinkron.v7i2.11387>
- Findari, W. S., Nugroho, Y. A., & Hasan, S. (2021). Optimasi waktu antrian pada layanan kesehatan menggunakan discrete event simulation. *Jurnal Teknik Industri*, 1, 657–666.
- Improving the queuing system of healthcare facility under priority-based discrete event simulation. (2024). *Jurnal Desain dan Pengembangan Teknologi*, 15(1), 73–80. <https://doi.org/10.34001/jdpt.v15i1.4954>
- Makhijani, R. (2021). *QUEST: Queue simulation for content moderation at scale*.
- Mei, D., Ristanti, R., Anasfi, F. P., & Sumariyono, E. (2024). Application of queue system simulation at Mie Gacoan restaurant branch Gresik using Arena 14.0. *Tibuna: Journal of Applied Industrial Engineering*, 7(1), 23–34. <https://doi.org/10.36456/tibuna.7.01.7938.23-34>
- Monks, T., & Harper, A. (2024). Improving the usability of open health service delivery simulation models using Python and web apps. *NIHR Open Research*, 1–35. <https://doi.org/10.3310/nihropenres.13467.1>
- Pratama, E., & Devianto, D. (2022). Analisis sistem antrian satu server (M/M/1). *Jurnal Matematika Unand*, 2(4), 59–66. <https://doi.org/10.25077/jmu.2.4.59-66.2013>
- Rausch, T. (2023). FaaS-sim: A trace-driven simulation framework for serverless edge computing platforms. *Software: Practice and Experience*, 53, 2327–2361. <https://doi.org/10.1002/spe.3277>
- Shella, S., Wara, M., Nasrudin, M., Adziima, A. F., & Pratama, A. R. (2025). Optimasi sistem antrian pada medical center ITS dengan simulasi discrete event dan response surface methodology. *Jurnal Teknik Industri*, 99(99), 1–8.
- Siregar, S. R. (2021). Simulasi penerapan multiple queue multiple server pada antrian bank dengan metode discrete event. *MEANS: Media Informatika dan Sistem Informasi*, 1(2), 6–12. <https://doi.org/10.54367/means.v1i2.2>
- Soekarno-Hatta, B., & Program Studi Teknik Industri Universitas Bina Nusantara. (2022). Analisis simulasi antrian penumpang di check-in counter bandara menggunakan discrete event simulation: Studi kasus Citilink di Bandara Soekarno-Hatta. *Jurnal Teknik Industri*, 4, 3449–3460.
- Sutarto, H. Y. (2021). Pemodelan dan simulasi antrian pada persimpangan dengan simulator discrete event system. *Telematika*, 9(2). <https://doi.org/10.61769/telematika.v9i2.91>

- Widodo, I. D., & Parkhan, A. (2024). Stochastic queuing system model design based on stakeholder aspirations. *Journal of Applied Data Science*, 5(4), 1715–1725. <https://doi.org/10.47738/jads.v5i4.314>
- Yusuf, M. (2023). Pengaruh distribusi dan parameter terhadap kinerja sistem antrian. *Klik: Kumpulan Jurnal Ilmu Komputer*, 3(6), 780–785. <https://doi.org/10.30865/klik.v3i6.878>