

**SKRIPSI**

**RANCANG BANGUN SIMULATOR WEB SERVER GUNA  
MENGUJI KINERJA SERVER MENGGUNAKAN *DISCRETE  
EVENT SIMULATION***

**Disusun dan Diajukan oleh :**

**RISCHA NURUL HIDAYATI  
D121 20 1068**



**PROGRAM STUDI SARJANA TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS HASANUDDIN  
GOWA  
2024**



Optimized using  
trial version  
[www.balesio.com](http://www.balesio.com)

# LEMBAR PENGESAHAN SKRIPSI

## Rancang Bangun Simulator Web Server Guna Menguji Kinerja Server Menggunakan *Discrete Event Simulation*

Disusun dan diajukan oleh

**Rischa Nurul Hidayati**  
**D121 20 1068**

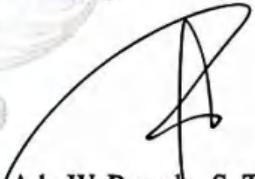
Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin Pada tanggal 12 Juni 2024 dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

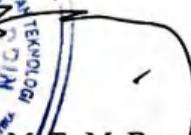
Pembimbing Utama,

Pembimbing Pendamping,

  
Mukarramah Yusuf, B.Sc., M.Sc., Ph.D  
NIP 19831008 201212 2 003

  
Dr. Eng. Ady W. Paundu, S. T., M. T.  
NIP 19750313 200912 1 003

Ketua Program Studi,

  
Prof. Dr. Ir. Indrabayuni, S. T., M. T., M. Bus., Sys., IPM, ASEAN. Eng.  
NIP 19750716 200212 1 004



## PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini :

Nama : Rischa Nurul Hidayati  
NIM : D121201068  
Program Studi : Teknik Informatika  
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

Rancang Bangun Simulator Web Server Guna Menguji Kinerja Server  
Menggunakan Discrete Event Simulation

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 07 Mei 2024

Yang Menyatakan



Rischa Nurul Hidayati



## ABSTRAK

**RISCHA NURUL HIDAYATI.** Rancang Bangun Simulator Web Server guna Menguji Kinerja Server Menggunakan Metode *Discrete Event Simulation* (dibimbing oleh Mukarramah Yusuf dan Ady W. Paundu)

Web server adalah perangkat lunak yang memberikan layanan data yang mempunyai fungsi untuk menerima permintaan HTTP atau HTTPS yang dikirim oleh klien melalui web browser dan mengirimkan kembali hasilnya dalam bentuk halaman web. Beban kerja web server bervariasi secara signifikan, dipengaruhi oleh faktor seperti distribusi waktu permintaan, jumlah pengguna, jenis permintaan pengguna, hingga kompleksitas halaman web.

Penelitian ini bertujuan menghasilkan simulator yang dapat digunakan dalam mensimulasikan permintaan yang masuk ke web server serta sebagai bantu untuk analisis lebih lanjut dalam penelitian dan pengembangan web server.

Metode penelitian yang digunakan untuk merancang dan mengembangkan simulator web server yang dapat menguji kinerja web server yaitu metode *Discrete Event Simulation* (DES). Adapun dataset yang digunakan dalam perancangan model yaitu data primer yang berjumlah 30 hari log akses dengan total data 1.178.884 data untuk domain utama dan 450.846 untuk sub-domain website. Simulator ini dibangun menggunakan pustaka SimPy dalam Python. Model simulator dibangun berdasarkan distribusi data dan modul generator bilangan acak (*Random Number Generator*) yang diperoleh dari log akses web server yang diteliti.

Hasil penelitian menunjukkan bahwa nilai parameter lambda memiliki pengaruh terhadap jumlah permintaan yang dihasilkan. Dari hasil skenario percobaan simulator dapat diketahui bahwa dari keempat skenario percobaan, nilai *lambda* 2.0 dapat menghasilkan jumlah permintaan yang paling mendekati data primer yaitu sebesar 90.83% (simulator domain) dan 90.67% (simulator sub-domain). Simulator menghasilkan metrik-metrik seperti rata-rata permintaan, distribusi interval kedatangan, distribusi kode status, dan distribusi *referrer*.

Kata Kunci: Web server, Simulasi, *Discrete Event Simulation*, SimPy



## ABSTRACT

**RISCHA NURUL HIDAYATI.** Designing a Web Server Simulator to Test Server Performance Using the Discrete Event Simulation Method (supervised by Mukarramah Yusuf and Ady W. Paundu)

A web server is a software that provides data services that receive HTTP or HTTPS requests sent by clients through a web browser and send back the results in the form of web pages. The workload of the web server varies significantly, influenced by factors such as the distribution of request time, the number of users, the type of user request, to the complexity of the web page.

This research aims to produce a simulator that can be used in simulating incoming requests to the web server as well as a tool for further analysis in web server research and development.

The research method used to design and develop a web server simulator that can test web server performance is the Discrete Event Simulation (DES) method. The dataset used in the model design is primary data totaling 30 days of access logs with a total of 1,178,884 data for the main domain and 450,846 for the website sub-domain. The simulator was built using the SimPy library in Python. The simulator model was built based on the data distribution and random number generator module obtained from the web server access logs studied.

The results show that the lambda parameter value influences the number of requests generated. From the results of the simulator experiment scenarios, it can be seen that of the four experimental scenarios, the lambda value of 2.0 can produce the number of requests closest to the primary data, namely 90.83% (domain simulator) and 90.67% (sub-domain simulator). The simulator generates metrics such as average requests, arrival interval distribution, status code distribution, and referrer distribution.

Keywords: Web server, Simulation, *Discrete Event Simulation*, SimPy



## DAFTAR ISI

LEMBAR PENGESAHAN SKRIPSI.....	i
PERNYATAAN KEASLIAN.....	ii
ABSTRAK.....	iii
ABSTRACT.....	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	vii
DAFTAR TABEL.....	x
DAFTAR SINGKATAN DAN ARTI SIMBOL.....	xi
DAFTAR LAMPIRAN.....	xii
KATA PENGANTAR.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian.....	2
1.4 Manfaat Penelitian.....	3
1.5 Ruang Lingkup.....	3
BAB II TINJAUAN PUSTAKA.....	4
2.1 Web Server.....	4
2.1.1 NGINX Web Server.....	5
2.1.2 Apache Web Server.....	7
2.2 Website.....	8
2.3 Log.....	9
2.3.1 IP Address.....	9
2.3.2 <i>Timestamp</i> .....	10
2.3.3 <i>Request Method</i> .....	10
2.3.4 <i>Path</i> .....	11
2.3.5 <i>Status Code</i> .....	11
2.3.6 <i>Response Size</i> .....	13
2.3.7 <i>Referrer</i> .....	13
2.3.8 <i>User Agent</i> .....	13
2.4 Pemodelan dan Simulasi.....	13
2.4.1 Pemodelan.....	13
2.4.2 Simulasi.....	14
2.4.3 Validasi dan Verifikasi.....	14
2.5 Metode <i>Discrete Event Simulation</i> .....	15
2.5.1 Antrian.....	15
2.5.2 Proses.....	16
2.5.3 Lingkungan.....	16
2.5.4 Sumber Daya.....	16
2.6 Random Number Generator.....	17
tribusi Poisson.....	17
tribusi Eksponensial.....	18
tPy.....	19
Hipotesis.....	20
T.....	21



2.8.2 Uji <i>Chi Square</i> .....	22
2.9 <i>Load Balancing</i> dan Pemanfaatannya .....	22
2.9.1 <i>Round Robin</i> .....	23
2.9.2 Pemanfaatan <i>Load Balancing</i> .....	23
2.10 Penelitian Terkait .....	24
BAB III METODE PENELITIAN .....	27
3.1 Tahapan Penelitian .....	27
3.2 Waktu dan Lokasi Penelitian .....	28
3.3 Instrumen Penelitian .....	28
3.4 Teknik Pengambilan Data .....	29
3.4.1 Pengambilan Data .....	29
3.4.2 Karakteristik Data .....	30
3.5 Perancangan dan Implementasi Model .....	31
3.5.1 Proses Pengembangan Model .....	31
3.5.2 Pembuatan Model .....	39
3.5.3 Perbedaan Penggunaan Pustaka SimPy .....	45
3.5.4 Implementasi DES .....	46
3.6 Skenario Pengujian .....	47
3.6.1 Pengujian Simulator Domain .....	47
3.6.2 Pengujian Simulator Sub-domain .....	48
3.6.3 Pengujian Lambda .....	49
3.6.4 Uji Hipotesis .....	49
BAB IV HASIL DAN PEMBAHASAN .....	51
4.1 Pengujian Simulator Domain .....	51
4.1.1 Skenario pertama .....	51
4.1.2 Skenario kedua .....	56
4.1.3 Skenario ketiga .....	61
4.1.4 Skenario keempat .....	65
4.2 Pengujian Simulator Sub-domain .....	70
4.2.1 Skenario pertama .....	70
4.2.2 Skenario kedua .....	76
4.2.3 Skenario ketiga .....	82
4.2.4 Skenario keempat .....	87
4.3 Uji Hipotesis .....	93
4.3.1 Simulator Domain .....	93
4.3.2 Simulator Sub-domain .....	96
4.3.3 Kesimpulan Uji Hipotesis .....	98
4.4 Pengujian Lambda .....	98
4.4.1 Kesimpulan Grafik Pengujian Lambda .....	100
4.4.2 Pengaruh Perangkat Dalam Pengujian Lambda .....	100
4.5 Visualisasi Simulator .....	100
BAB V PENUTUP .....	102
5.1 Kesimpulan .....	102
5.2 Kesimpulan dan Saran .....	102
DAFTAR PUSTAKA .....	104
DAFTAR LAMPIRAN .....	109



## DAFTAR GAMBAR

Gambar 1. Cara kerja web server .....	4
Gambar 2. Mekanisme kerja NGINX .....	6
Gambar 3. Mekanisme kerja apache .....	7
Gambar 4. Website resmi universitas hasanuddin.....	8
Gambar 5. Struktur log akses .....	9
Gambar 6. Distribusi poisson.....	18
Gambar 7. Distribusi eksponensial .....	19
Gambar 8. Tahapan penelitian.....	27
Gambar 9. Cuplikan data log akses yang diperoleh.....	30
Gambar 10. Flowchart perancangan sistem .....	31
Gambar 11. (a) Cuplikan log akses domain; (b) Cuplikan log akses sub-domain	32
Gambar 12. Pembersihan data.....	32
Gambar 13. (a) Interval permintaan domain; (b) Interval permintaan domain dalam persen.....	33
Gambar 14. (a) Interval permintaan sub-domain; (b) Interval permintaan sub- domain dalam persen.....	34
Gambar 15. (a) Distribusi <i>request method</i> domain; (b) Distribusi <i>request method</i> sub-domain.....	35
Gambar 16. (a) Distribusi <i>status code</i> domain; (b) Distribusi <i>status code</i> sub- domain.....	36
Gambar 17. (a) Distribusi <i>referrer</i> domain; (b) Distribusi <i>referrer</i> sub-domain ..	37
Gambar 18. <i>Random number generator</i> untuk jumlah permintan .....	38
Gambar 19. <i>Random number generator</i> untuk jarak kedatangan permintaan .....	38
Gambar 20. <i>Random number generator</i> untuk kode status.....	38
Gambar 21. <i>Random number generator</i> untuk metode permintaan.....	39
Gambar 22. <i>Random number generator</i> untuk <i>referrer</i> .....	39
Gambar 23. <i>Flowchart</i> program.....	40
Gambar 24. (a) Rata-rata permintaan skenario pertama domain (hasil simulasi); (b) Rata-rata permintaan domain (data primer) .....	52
Gambar 25. (a) Interval permintaan skenario pertama domain (hasil simulasi); (b) Interval permintaan domain (data primer) .....	53
Gambar 26. (a) Kode status skenario pertama domain (hasil simulasi); (b) Kode status domain (data primer).....	53
Gambar 27. (a) <i>Referrer</i> domain skenario pertama (hasil simulasi); (b) <i>Referrer</i> domain (data primer).....	54
Gambar 28. (a) Metode permintaan skenario pertama domain (hasil simulasi); (b) Metode permintaan (data primer).....	54
Gambar 29. (a) Rata-rata permintaan skenario kedua domain (hasil simulasi); (b) permintaan domain (data primer) .....	57
Gambar 30. (a) Interval permintaan skenario kedua domain (hasil simulasi); (b) permintaan domain (data primer) .....	57



Gambar 31. (a) Kode status skenario kedua domain (hasil simulasi); (b) Kode status domain (data primer).....	58
Gambar 32. (a) <i>Referrer</i> skenario kedua domain (hasil simulasi); (b) <i>Referrer</i> domain (data primer).....	58
Gambar 33. (a) Metode permintaan skenario kedua domain (hasil simulasi); (b) Metode permintaan domain (data primer).....	59
Gambar 34. (a) Rata-rata permintaan skenario ketiga domain (hasil simulasi); (b) Rata-rata permintaan domain (data primer) .....	61
Gambar 35. (a) Interval permintaan skenario ketiga domain (hasil simulasi); (b) Interval permintaan domain (data primer) .....	62
Gambar 36. (a) Kode status skenario ketiga domain (hasil simulasi); (b) Kode status domain (data primer).....	62
Gambar 37. (a) <i>Referrer</i> skenario ketiga domain (hasil simulasi); (b) <i>Referrer</i> domain (data primer).....	63
Gambar 38. (a) Metode permintaan skenario ketiga domain (hasil simulasi); (b) Metode permintaan (data primer).....	63
Gambar 39. (a) Rata-rata permintaan skenario keempat domain (hasil simulasi); (b) Rata-rata permintaan domain (data primer) .....	66
Gambar 40. (a) Interval permintaan skenario keempat domain (hasil simulasi); (b) Interval permintaan domain (data primer) .....	66
Gambar 41. (a) Kode status skenario keempat domain (hasil simulasi); (b) Kode status domain (data primer).....	67
Gambar 42. (a) <i>Referrer</i> skenario keempat domain (hasil simulasi); (b) <i>Referrer</i> domain (data primer).....	67
Gambar 43. (a) Metode permintaan skenario keempat domain (hasil simulasi); (b) Metode permintaan domain (data primer).....	68
Gambar 44. (a) Rata-rata permintaan skenario pertama sub-domain (hasil simulasi); (b) Rata-rata permintaan (data primer).....	71
Gambar 45. (a) Interval permintaan skenario pertama sub-domain (hasil simulasi); (b) Interval permintaan sub-domain (data primer).....	72
Gambar 46. (a) Kode status skenario pertama sub-domain (hasil simulasi); (b) Kode status sub-domain (data primer) .....	72
Gambar 47. (a) <i>Referrer</i> skenario pertama sub-domain (hasil simulasi); (b) <i>Referrer</i> sub-domain (data primer) .....	73
Gambar 48. (a) Metode permintaan skenario pertama sub-domain (hasil simulasi); (b) Metode permintaan (data primer).....	73
Gambar 49. (a) Rata-rata permintaan skenario kedua sub-domain (hasil simulasi); (b) Rata-rata permintaan sub-domain (data primer).....	77
Gambar 50. (a) Interval permintaan skenario kedua sub-domain (hasil simulasi); (b) Interval permintaan sub-domain (data primer).....	77
51. (a) Kode status skenario kedua sub-domain (hasil simulasi); (b) Kode sub-domain (data primer).....	78
52. <i>Referrer</i> skenario kedua sub-domain (hasil simulasi); (b) <i>Referrer</i> sub-domain (data primer).....	78



Gambar 53. (a) Metode permintaan skenario kedua sub-domain (hasil simulasi); (b) Metode permintaan (data primer).....	79
Gambar 54. (a) Rata-rata permintaan skenario ketiga sub-domain (hasil simulasi); (b) Rata-rata permintaan sub-domain (data primer).....	82
Gambar 55. (a) Interval permintaan skenario ketiga sub-domain (hasil simulasi); (b) Interval permintaan sub-domain (data primer).....	83
Gambar 56. (a) Kode status skenario ketiga sub-domain (hasil simulasi); (b) Kode status sub-domain (data primer).....	83
Gambar 57. (a) <i>Referrer</i> skenario ketiga sub-domain (hasil simulasi); (b) <i>Referrer</i> sub-domain (data primer).....	84
Gambar 58. (a) Metode permintaan skenario ketiga sub-domain (hasil simulasi); (b) Metode permintaan sub-domain (data primer).....	84
Gambar 59. (a) Rata-rata permintaan skenario keempat sub-domain (hasil simulasi); (b) Rata-rata permintaan sub-domain (data primer).....	88
Gambar 60. (a) Interval permintaan skenario keempat sub-domain (hasil simulasi); (b) Interval permintaan sub-domain (data primer).....	88
Gambar 61. (a) Kode status skenario keempat sub-domain (hasil simulasi); (b) Kode status sub-domain (data primer).....	89
Gambar 62. (a) <i>Referrer</i> skenario keempat sub-domain (hasil simulasi); (b) <i>Referrer</i> sub-domain (data primer).....	89
Gambar 63. (a) Metode permintaan skenario keempat sub-domain (hasil simulasi); (b) Metode permintaan sub-domain (data primer).....	90
Gambar 64. Hasil pengujian lambda.....	99
Gambar 65. Visualisasi simulator .....	101



## DAFTAR TABEL

Tabel 1. Atribut dalam kelas WebServer .....	41
Tabel 2. Metode dalam kelas WebServer .....	42
Tabel 3. Atribut dalam kelas Customer .....	43
Tabel 4. Metode dalam kelas Customer .....	44
Tabel 5. Perbedaan penggunaan pustaka Simpy .....	45
Tabel 6. Uji hipotesis.....	49
Tabel 7. Parameter skenario pertama domain .....	51
Tabel 8. Perbandingan hasil simulator skenario pertama dan data primer.....	54
Tabel 9. Parameter skenario kedua domain.....	56
Tabel 10. Perbandingan hasil simulator skenario kedua dan data primer .....	59
Tabel 11. Parameter skenario ketiga domain.....	61
Tabel 12. Perbandingan hasil simulator skenario ketiga dan data primer.....	63
Tabel 13. Parameter skenario keempat domain.....	65
Tabel 14. Perbandingan hasil simulator skenario keempat dan data primer .....	68
Tabel 15. Perbandingan hasil skenario domain.....	70
Tabel 16. Parameter skenario pertama sub-domain .....	70
Tabel 17. Perbandingan hasil simulator skenario pertama dan data primer.....	73
Tabel 18. Parameter skenario kedua sub-domain.....	76
Tabel 19. Perbandingan hasil simulator skenario kedua dan data primer .....	79
Tabel 20. Parameter skenario ketiga sub-domain .....	82
Tabel 21. Perbandingan hasil simulator skenario ketiga dan data primer.....	84
Tabel 22. Parameter skenario keempat sub-domain .....	87
Tabel 23. Perbandingan hasil simulator skenario keempat dan data primer .....	90
Tabel 24. Perbandingan hasil skenario sub-domain.....	93
Tabel 25. Nilai yang digunakan dalam uji T simulator domain .....	93
Tabel 26. Hasil uji T domain.....	94
Tabel 27. Nilai yang digunakan dalam uji <i>chi-square</i> simulator domain .....	94
Tabel 28. Hasil uji <i>chi-square</i> domain .....	95
Tabel 29. Nilai yang digunakan dalam uji T simulator sub-domain .....	96
Tabel 30. Hasil uji T sub-domain .....	96
Tabel 31. Nilai yang digunakan dalam uji <i>chi-square</i> simulator sub-domain.....	97
Tabel 32. Hasil uji <i>chi-square</i> sub-domain .....	97



## DAFTAR SINGKATAN DAN ARTI SIMBOL

Lambang/Singkatan	Arti dan Keterangan
$\lambda$	Laju rata-rata ( <i>lambda</i> )
$\alpha$	Tingkat signifikansi ( <i>alpha</i> )
DES	<i>Discrete Event Simulation</i>
H0	Hipotesis nol/netral
H1	Hipotesis alternatif
p-value	Ukuran probabilitas (nilai p)
RNG	<i>Random Number Generator</i>



## DAFTAR LAMPIRAN

Lampiran 1. Dataset .....	110
Lampiran 2. Pra-pemrosesan data (data preprocessing) .....	111
Lampiran 3. Mencari distribusi data .....	113
Lampiran 4. Uji parameter .....	118
Lampiran 5. Uji hipotesis .....	120
Lampiran 6. Daftar hadir dan berita acara seminar hasil .....	122
Lampiran 7. Daftar hadir dan berita acara ujian skripsi .....	125
Lampiran 8. Lembar perbaikan skripsi .....	128



## KATA PENGANTAR

Segala puji hanya milik Allah *Subhanallahu Wa Ta'ala* atas segala limpahan rahmat dan hidayah-Nya. Shalawat dan salam senantiasa tercurahkan kepada baginda Rasulullah *Shallallahu 'Alaihi Wa sallam. Alhamdulillahirabbil'aalamiin*, berkat *ridha'* dan kemudahan yang diberikan oleh Allah *Subhanallahu Wa Ta'ala*, sehingga penulis dapat menyelesaikan tugas akhir dengan judul **“Rancang Bangun Simulator Web Server guna Menguji Kinerja Server Menggunakan Metode *Discrete Event Simulation*”** sebagai salah satu syarat dalam menyelesaikan jenjang Strata-1 di Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin.

Penulis menyadari bahwa dalam penyusunan dan penulisan tugas akhir ini tidak dapat terselesaikan dengan baik tanpa adanya bimbingan dan bantuan baik materi maupun non-materi dari berbagai pihak. Bimbingan dan bantuan tersebut adalah alasan penulisan dan penyusunan tugas akhir ini dapat terselesaikan dengan lancar. Oleh karena itu sebagai salah bentuk penghargaan yang setinggi-tingginya, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Allah *Subhanallahu Wa Ta'ala*, atas izin rahmat, nikmat dan kasih sayang-Nya sehingga skripsi ini dapat terselesaikan.
2. Keluarga penulis, (Almarhum) Bapak H. Jumadil Tunggu dan Ibu Hj. Nurmiati sebagai orang tua penulis yang selalu mendoakan untuk kebaikan penulis, selalu memberikan kasih sayang, cinta, dukungan dan motivasi. Terima kasih kepada Bapak dan Ibu yang selalu sabar dan semangat tiada henti dalam menghadapi dan mendidik penulis. Resty Prazilia Afzarita sebagai kakak penulis yang senantiasa mendoakan, memberikan semangat, dan bantuan materi maupun non-materi kepada penulis.
3. Ibu Mukarramah Yusuf, B.Sc., M.Sc., Ph.D. selaku pembimbing I dan Bapak Dr. Eng. Ady W. Paundu, S.T., M.T. selaku pembimbing II, yang senantiasa menyempatkan waktu untuk mengarahkan, membimbing, dan berbagi ilmu kepada penulis untuk menyelesaikan tugas akhir.



4. Segenap dosen dan staf Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin yang telah membantu penulis selama masa perkuliahan.
5. Segenap staf Direktorat Sistem Informasi dan Transformasi Digital Universitas Hasanuddin yang telah membantu penulis selama proses pengumpulan data.
6. Agung, Calvin, Ray, Kisana, Shekinah, Tasya, Nuni, Nabila, Mutiah, Avil, Rizal, May, Thesa, Firman, dan teman-teman dekat penulis yang tidak dapat penulis sebutkan satu persatu.
7. Teman-teman Rezolver 20 yang telah kebersamai penulis selama masa perkuliahan, atas waktu bersama dan hiburan-hiburan yang telah diberikan.
8. Mochi selaku kucing kesayangan penulis yang senantiasa menghibur penulis dalam menyelesaikan tugas akhir.
9. BTS, LE SSERAFIM, dan ENHYPEN yang memberikan hiburan dan semangat kepada penulis melalui musik mereka.
10. Seluruh pihak yang tidak disebutkan dan tanpa sadar telah menjadi inspirasi dan membantu penulis dalam menyelesaikan tugas akhir.
11. Kepada diri sendiri yang sudah bertahan dan berjuang sejauh ini.

Penulis berharap semoga Allah membalas segala kebaikan yang telah diterima oleh penulis dari berbagai pihak yang telah membantu mempermudah penulis dalam penyelesaian tugas akhir ini. Penulis menyadari bahwa tugas akhir ini masih jauh dari kata sempurna. Oleh karena itu, penulis mengharapkan segala bentuk saran serta masukan yang membangun dari berbagai pihak. Semoga tugas akhir ini dapat berguna bagi para pembaca.

Gowa, 07 Mei 2024

Rischa Nurul Hidayati



# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Perkembangan teknologi yang semakin canggih memberikan kemudahan dalam memperoleh berbagai informasi. Hampir semua informasi dapat diketahui melalui media web. Dengan meningkatnya pengunjung yang mengakses suatu web akan berdampak pada beban web server tersebut, sehingga waktu aksesnya menjadi lebih lambat bahkan dapat menjadikan web server mati dan tidak dapat melayani permintaan akses. Hal tersebut sangat mungkin terjadi jika web server merupakan server tunggal dalam melayani permintaan akses (Akbar, 2023).

Web server merupakan sistem kompleks yang terdiri dari berbagai komponen dan subsistem. Komponen-komponen tersebut berinteraksi satu sama lain dengan cara yang kompleks, sehingga sulit untuk memprediksi bagaimana perubahan pada satu komponen akan memengaruhi kinerja sistem secara keseluruhan (Akbar, 2023).

Performa sistem sangat bervariasi berdasarkan arsitektur, logika aplikasi, dan kapasitasnya. Manajemen kinerja sistem yang efisien dan efektif diperlukan untuk memenuhi dan mempertahankan layanan pelanggan yang memuaskan, serta meminimalkan biaya operasional dan modal. Untuk sistem baru, penting untuk menganalisis kinerja sebelum penerapan untuk memastikan bahwa sistem bekerja sesuai tingkat yang diharapkan. Untuk sistem yang sudah ada, penting untuk menganalisis bagaimana kinerja sistem dalam kondisi yang berubah (Alam, 2012).

Simulasi adalah tiruan bagaimana sistem dunia nyata beroperasi sepanjang waktu. Hal ini dapat digunakan untuk mengidentifikasi titik-titik kritis hambatan sistem dan untuk menjawab pertanyaan “*what-if*” tentang skenario dunia nyata. Salah satu metode dalam pemodelan dan simulasi yaitu metode *Discrete Event Simulation* (DES). *Discrete Event simulation* (DES) adalah teknik riset operasi berbasis komputer yang memodelkan sistem yang berbeda sebagai jaringan antrian untuk menilai, memprediksi, dan mengoptimalkan sistem yang ada atau yang sudah ada, di mana perubahan terjadi pada periode diskrit dari waktu (Isaac et al., 2022).



Beban kerja web server bervariasi secara signifikan. Variasi ini dapat disebabkan oleh berbagai faktor, seperti distribusi waktu permintaan, jumlah pengguna, jenis permintaan pengguna, hingga kompleksitas halaman web. Hal ini dapat menyulitkan untuk memilih algoritma yang optimal untuk menguji semua kondisi. Dengan adanya masalah ini, salah satu pendekatan yang dapat digunakan untuk mengatasinya adalah dengan menerapkan metode simulasi. Simulasi dapat digunakan untuk menguji kinerja web server dalam berbagai kondisi beban kerja, sehingga dapat dipilih algoritma peningkatan kinerja yang optimal untuk semua kondisi (Isaac et al., 2022).

Dengan melihat permasalahan tersebut, penulis mengusulkan judul “Rancang Bangun Simulator Web Server Guna Menguji Kinerja Server dengan Metode *Discrete Event Simulation*” untuk membangun simulator yang dapat digunakan dalam mengukur kinerja server dengan menggunakan metode *Discrete Event Simulation*.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan, maka rumusan masalah pada penelitian ini yaitu:

1. Bagaimana merancang simulator web server yang dapat menguji kinerja server menggunakan *Discrete Event Simulation*?
2. Bagaimana *Discrete Event Simulation* dapat digunakan untuk memodelkan dan mensimulasikan kinerja web server?
3. Apa saja metrik kinerja yang penting untuk diuji dalam simulator web server untuk merefleksikan kondisi web server nyata?

## 1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah diuraikan dapat diperoleh tujuan dari penelitian ini yaitu:

1. Mengembangkan simulator web server yang dapat digunakan untuk menguji kinerja server menggunakan *Discrete Event Simulation*.  
nerapkan model simulasi *Discrete Event Simulation* sebagai pendekatan ma dalam simulator untuk mencerminkan kinerja web server.



3. Menentukan dan mengintegrasikan metrik kinerja yang relevan dalam simulator web server.

#### 1.4 Manfaat Penelitian

Melalui penelitian ini, diharapkan dapat memberikan manfaat antara lain:

1. Menghasilkan simulator web server yang dapat digunakan peneliti dan pengembang untuk menguji kinerja server.
2. Memberikan pemahaman tentang perilaku server di berbagai kondisi beban kerja melalui simulasi.
3. Membantu dalam mengambil keputusan terkait peningkatan infrastruktur dan penyesuaian sumber daya server berdasarkan data simulasi.

#### 1.5 Ruang Lingkup

Ruang lingkup dalam penelitian ini adalah sebagai berikut:

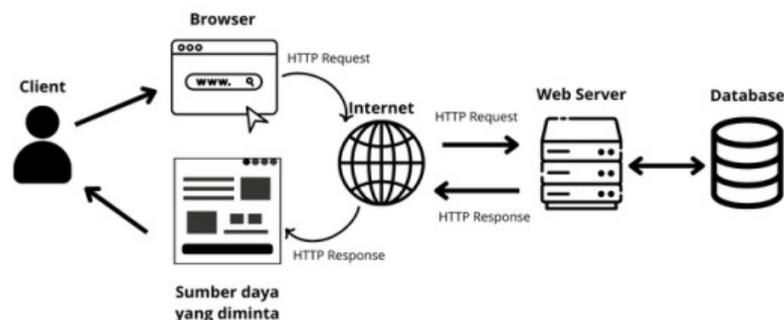
1. Simulator yang dibuat hanya untuk menguji web server yang menjadi objek penelitian yaitu domain utama Universitas Hasanuddin ([www.unhas.ac.id](http://www.unhas.ac.id)) dan sub-domain Universitas Hasanuddin yang berisi web hosting beberapa instansi di Universitas Hasanuddin.
2. Data primer yang digunakan dalam pembuatan model bersumber dari log akses (*access log*).
3. Metode yang digunakan dalam pembuatan model yaitu metode *Discrete Event Simulation* dengan menggunakan pustaka SimPy dan bahasa Python.
4. Simulator hanya bisa dijalankan per hari.
5. Simulator hanya dapat mensimulasikan pola permintaan yang normal (mengikuti log akses) dan tidak dapat menirukan pola permintaan yang berulang (*requests looping*).
6. Jumlah server yang digunakan yaitu satu server (*default*).



## BAB II TINJAUAN PUSTAKA

### 2.1 Web Server

Web server adalah perangkat lunak yang memberikan layanan data yang mempunyai fungsi untuk menerima permintaan HTTP (*HyperText Transfer Protocol*) atau HTTPS (*HyperText Transfer Protocol Secure*) yang dikirim oleh klien melalui web browser dan mengirimkan kembali hasilnya dalam bentuk halaman web yang umumnya berbentuk dokumen HTML (*Hyper Text Markup Language*). Web Server berfungsi pula untuk mengirim seluruh aspek pemberkasan dalam sebuah halaman web yang terkait, termasuk di dalamnya teks, gambar, video, atau lainnya (Dimas & Asmunin, 2018).



Gambar 1. Cara kerja web server

Web server menggunakan model klien-server. Dalam struktur ini, klien akan meminta sumber daya atau layanan dari server. Ketika pengguna web ingin mengakses konten website, browser akan meminta akses melalui internet yang disebut *HTTP request*. Kemudian browser mencari alamat IP website yang diminta dengan menerjemahkan URL (*Uniform Resource Locator*) halaman web melalui sistem domain atau pencarian dalam *cache*. Proses ini akan menemukan web server yang menghosting file website tersebut. Web server menerima permintaan HTTP, lalu memprosesnya melalui server HTTP. Ketika permintaan diterima, server HTTP akan menelusuri file server untuk mengambil data yang relevan. Setelah itu, web server mengembalikan hasil dalam bentuk file website ke browser yang tadi akan permintaan. Kemudian, pengguna bisa melihat konten website (Matz et al., 2018).



Perangkat lunak web server yang umum digunakan yaitu Apache, NGINX, Microsoft IIS (*Internet Information Services*), dan LiteSpeed. Setiap perangkat lunak memiliki karakteristik unik dalam hal konfigurasi, kinerja, dan fitur keamanan. Pilihan web server biasanya didasarkan pada persyaratan spesifik aplikasi web, seperti skalabilitas, keamanan, dan kemampuan menangani koneksi bersamaan.

### 2.1.1 NGINX Web Server

Berdasarkan situs resmi nginx, NGINX Web Server adalah perangkat lunak server web sumber terbuka yang berfungsi sebagai reverse proxy, penyeimbang beban HTTP, serta proxy email untuk IMAP, POP3, dan SMTP. Nginx dapat menjalankan *multi-thread* dengan menggunakan pendekatan asinkron, *event-driven*, dan *non-blocking*.

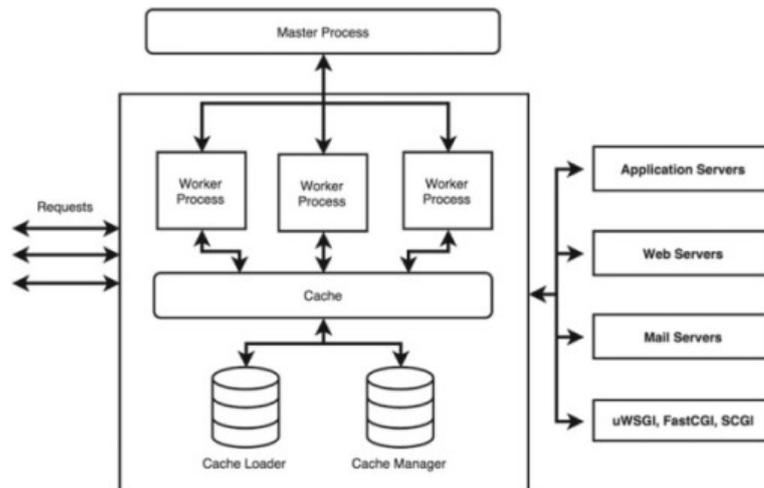
#### 1. *Event-driven*

NGINX mengadopsi model pemrosesan berbasis *event* di mana web server beroperasi berdasarkan peristiwa (*events*), seperti permintaan masuk, koneksi baru, atau selesainya sebuah operasi I/O. Model *event-driven* memungkinkan NGINX untuk menangani banyak koneksi secara simultan dengan menggunakan mekanisme yang lebih ringan. Ketika sebuah *event* terjadi, NGINX akan segera menanganinya tanpa harus menunggu *event* lain selesai (Floyd & Jacobson, 1993).

#### 2. *Non-blocking*

Pada NGINX, operasi input/output (I/O) dilakukan secara *non-blocking*. Ketika melakukan operasi I/O (seperti membaca atau menulis data), NGINX tidak akan menunggu hingga operasi tersebut selesai. Sebaliknya, NGINX akan melanjutkan untuk melakukan tugas lain yang tersedia pada saat itu (Thau, 1996).





Sumber: goldenfast.net (2017)

Gambar 2. Mekanisme kerja NGINX

Pada Gambar 2 dapat dilihat bahwa pada umumnya NGINX memiliki arsitektur *master-slave* yang memiliki beberapa bagian yaitu:

1. Master adalah proses utama yang mengendalikan proses-proses lainnya. Master bertugas mengendalikan semua proses anak (*worker processes*), mengelola, dan memvalidasi konfigurasi baru.
2. *Worker/slave* adalah proses yang menangani permintaan dari klien. *Worker* bertugas menerima permintaan baru yang dialokasikan dari master.
3. *Cache* adalah mekanisme penyimpanan sementara yang digunakan untuk menyimpan Salinan dari konten yang sering diakses oleh klien. *Cache* bertugas mengurangi waktu yang dibutuhkan untuk menghasilkan dan mengirim ulang konten kepada klien.

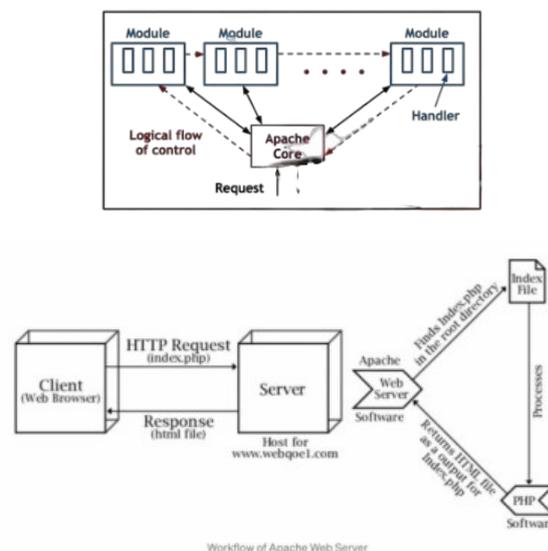
Dengan menggunakan NGINX, satu proses utama dapat mengendalikan berbagai proses lainnya sehingga proses utama tetap dapat menjalankan tugasnya dengan mengontrol proses lain yang berada di bawahnya (Wang & Zhao, 2021). NGINX menggunakan pendekatan asinkron, sehingga setiap permintaan web dapat dieksekusi oleh proses lain tanpa mengganggu permintaan web lainnya. Ketika ada permintaan baru, mintaan tersebut akan dibuat dan dijalankan dalam proses yang sudah ada.



## 2.1.2 Apache Web Server

Apache adalah perangkat lunak web server yang dibuat oleh *Apache Software Foundation* yang dapat berjalan di Linux, Windows, dan macOS (Apache.org). Apache merupakan perangkat lunak sumber terbuka yang tersedia secara gratis dan digunakan untuk menyajikan halaman web dan konten lainnya melalui internet.

Apache mampu menyajikan konten statis dan dinamis serta dapat dikonfigurasi untuk menangani berbagai protokol, seperti HTTP, HTTPS, dan FTP. Apache menggunakan model pemrosesan yang disebut MPM (*Multi Processing Module*) yang memungkinkan Apache untuk memproses permintaan secara bersamaan dengan menggunakan beberapa proses atau *thread* (Kunda & Sinyinda, 2017).



Sumber: itbox.id (2023)

Gambar 3. Mekanisme kerja apache

Pada Gambar 3 dapat dilihat arsitektur Apache yaitu:

### 1. Modul

Apache dibangun dengan arsitektur modular yang memungkinkan administrator server untuk menyesuaikan fungsionalitas server sesuai kebutuhan. Modul dapat mengendalikan berbagai aspek operasi server, seperti pengolahan permintaan, autentikasi pengguna, pengiriman konten, *logging*, dan lain sebagainya (Apache.org).



## 2. Multi Processing Module (MPM)

MPM adalah komponen utama Apache yang bertanggung jawab untuk menangani koneksi klien dan memproses permintaan klien. Apache mendukung MPM yang berbeda, seperti *Prefork*, *Worker*, dan *Event*. MPM membuat *thread* untuk menangani permintaan masuk. Setiap proses atau *thread* anak memiliki ruang memorinya sendiri dan tidak bergantung pada proses atau *thread* lainnya. Hal ini memastikan bahwa jika salah satu *thread* mengalami kegagalan, tidak akan memengaruhi proses atau *thread* lainnya (Kunda & Sinyinda, 2017).

## 2.2 Website

Website menjadi salah satu aspek penting dalam dunia bisnis dan pendidikan yang memungkinkan organisasi untuk berinteraksi dengan konsumen dan menyediakan berbagai layanan secara online. Tarafdar dan Zhang (2005) menekankan pentingnya desain dan kegunaan website yang baik. Website dengan navigasi yang mudah, konten yang relevan dan informatif, serta tampilan yang menarik cenderung lebih efektif dalam menarik dan mempertahankan pengunjung. Website yang dirancang dengan mempertimbangkan aspek kegunaan dapat meningkatkan pengalaman pengguna, memfasilitasi interaksi yang lancar, dan meningkatkan kepuasan pengguna (Statnik, 2023).



unhas.ac.id (2024)

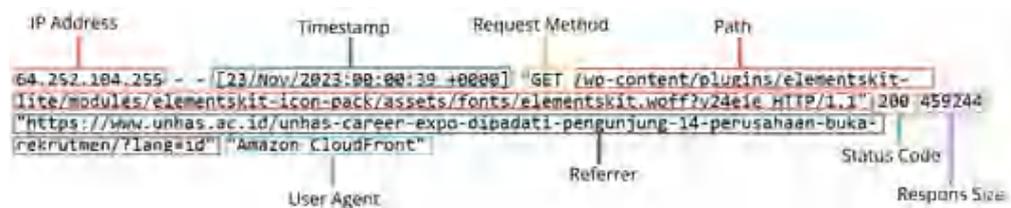
Gambar 4. Website resmi universitas hasanuddin



Website yang memiliki tingkat usability yang tinggi memberikan kesan positif kepada pengunjung. Strategi penting untuk memastikan navigasi yang mudah di sebuah website adalah memberikan kemudahan bagi pengunjung untuk menemukan informasi yang mereka butuhkan. Tanpa kemampuan ini, pengunjung mungkin akan beralih ke alternatif lain untuk memenuhi kebutuhan mereka yang dapat menyebabkan kehilangan pelanggan potensial (Statnik, 2023).

## 2.3 Log

Log adalah file yang dihasilkan perangkat lunak yang berisi informasi mengenai aktivitas, operasi, dan pola penggunaan aplikasi, server, atau sistem di dalam komputer (Khaire et al., 2022). File log merekam dan mencatat aktivitas, kejadian, atau perubahan yang terjadi dalam lingkungan komputer. Log web server adalah file yang berisi catatan tentang semua aktivitas yang terjadi pada web server. Informasi yang terkandung dalam log sangat beragam, termasuk alamat IP pengguna, jenis permintaan HTTP yang dilakukan, status respons dari server, waktu permintaan, dan detail lainnya yang berkaitan dengan interaksi pengguna dengan situs web. Analisis log memungkinkan pemahaman mendalam tentang perilaku pengunjung situs, efisiensi server, dan pengoptimalan kinerja server (Calzarossa & Massari, 2010). Pada penelitian ini, peneliti mengambil data dari log akses. Data yang dapat diperoleh dari log akses adalah sebagai berikut:



Gambar 5. Struktur log akses

### 2.3.1 IP Address

Alamat IP (*IP Address*) merujuk pada nomor unik yang digunakan untuk mengidentifikasi perangkat dalam suatu jaringan internet. Alamat IP liri dari deretan angka biner antara 32 bit sampai 128 bit dan terbagi menjadi dua versi yaitu IPv4 dan IPv6 (Kurose & Ross, 2017). Dalam web



server, alamat IP digunakan untuk menentukan sumber permintaan yang masuk ke server.

### 2.3.2 *Timestamp*

*Timestamp* adalah catatan waktu ketika permintaan dibuat ke server. Setiap kali pengguna atau perangkat mengunjungi situs web, pengguna meminta web server untuk menyajikan halaman web atau sumber daya lainnya, seperti gambar atau skrip (Kiruthika & Selvi, 2019). Ketika permintaan ini diterima oleh web server, rinciannya dicatat dalam log akses. Informasi yang dikumpulkan yaitu waktu permintaan diterima yang mencakup informasi seperti tanggal, waktu, zona waktu, dan terkadang informasi tambahan seperti offset zona waktu.

### 2.3.3 *Request Method*

Jenis permintaan web server mencakup protokol dan metode yang digunakan oleh klien saat berkomunikasi dengan web server (Cahyanto & Prayudi, 2014). Dalam data log akses yang digunakan diperoleh jenis permintaan yang terdapat di web server yaitu:

a. GET

GET digunakan untuk mengambil data dari server. Metode GET dianggap aman karena hanya mengambil data dan tidak mengubah apapun di server.

b. POST

POST digunakan untuk mengirimkan data ke server untuk membuat atau memperbarui sumber daya.

c. HEAD

HEAD digunakan untuk mengambil informasi *header* sumber daya tanpa mengunduh konten sebenarnya.

d. OPTIONS

OPTIONS digunakan untuk mendapatkan informasi tentang kemampuan komunikasi server dan opsi lain yang tersedia untuk sumber daya.



### 2.3.4 Path

*Path* dalam akses log merujuk pada file atau halaman tertentu yang diminta oleh klien dari web server. Sumber daya ini dapat berupa berbagai jenis file, seperti halaman HTML (*Hypertext Markup Language*), gambar, dokumen CSS (*Cascading Style Sheet*), JavaScript, video, audio, atau file lain yang disajikan oleh server (Kiruthika & Selvi, 2019).

### 2.3.5 Status Code

Status permintaan dalam log akses web server merujuk pada kode status HTTP yang dikembalikan oleh server sebagai tanggapan terhadap permintaan yang diterima, Kode status HTTP memberikan informasi tentang keberhasilan atau kegagalan suatu permintaan (Khaire et al., 2022). Dalam data log akses yang digunakan diperoleh status permintaan yang terdapat di web server yaitu:

a. 200 OK

Kode status 200 OK adalah kode status yang menunjukkan bahwa permintaan telah berhasil dan server telah memberikan respons yang tepat.

b. 201 *Created*

Kode status 201 *Created* adalah kode status yang menunjukkan bahwa permintaan telah berhasil dan server telah membuat sumber daya baru.

c. 206 *Partial Content*

Kode status 206 *Partial Content* adalah kode status yang menunjukkan bahwa server hanya mengirimkan sebagian dari sumber daya ketika *range header* yang dikirim dari klien untuk meminta hanya sebagian dari sumber daya.

d. 301 *Moved Permanently*

Kode status 301 *Moved Permanently* adalah kode status yang menunjukkan bahwa sumber daya yang diminta telah dipindahkan secara permanen ke lokasi baru.



e. *302 Found*

Kode status *302 Found* adalah kode status yang menunjukkan bahwa sumber daya yang diminta telah ditemukan di lokasi yang berbeda secara sementara. Klien harus menggunakan URL yang diberikan untuk permintaan selanjutnya, tetapi pengalihan ini hanya bersifat sementara.

f. *400 Bad Request*

Kode status *400 Bad Request* adalah kode status yang menunjukkan bahwa server tidak dapat memahami permintaan klien karena sesuatu yang dianggap sebagai kesalahan klien.

g. *401 Unauthorized*

Kode status *401 Unauthorized* adalah kode status yang menunjukkan bahwa akses ke sumber daya yang diminta dilarang karena klien tidak memiliki otorisasi yang tepat. Klien mungkin perlu menyediakan kredensial yang valid untuk mengakses sumber daya.

h. *403 Forbidden*

Kode status *403 Forbidden* adalah kode status yang menunjukkan bahwa klien tidak memiliki hak akses yang sah ke sumber daya sehingga server menolak memberikan sumber daya yang diminta.

i. *404 Not Found*

Kode status *404 Not Found* adalah kode status yang menunjukkan bahwa sumber daya yang diminta oleh klien tidak dapat ditemukan oleh server.

j. *499 Client Closed Request*

Kode status *499 Client Closed Request* adalah kode status yang menunjukkan bahwa klien telah menutup koneksi sebelum server selesai mengirimkan respons. Hal ini sering terjadi ketika klien membatalkan permintaan sebelum server selesai mengirim sumber daya yang diminta.

k. *500 Internal Server Error*

Kode status *500 Internal Server Error* adalah kode status yang menunjukkan bahwa terjadi kesalahan internal di server saat mencoba



memproses permintaan dan server tidak dapat menyelesaikan permintaan tersebut.

#### 1. 503 *Service Unavailable*

Kode status 503 *Service Unavailable* adalah kode status yang menunjukkan bahwa server tidak dapat menanggapi permintaan karena sementara tidak tersedia atau sedang mengalami peningkatan beban.

### 2.3.6 *Response Size*

Ukuran respons pada log akses web server adalah jumlah data yang dikirimkan oleh server ke klien sebagai tanggapan atas permintaan tertentu. Ukuran respons diukur dalam *byte* dan mencakup total data yang terkirim (Morville & Rosenfeld, 2006).

### 2.3.7 *Referrer*

*Referrer* adalah URL (*Uniform Resource Locator*) yang memberikan informasi tentang sumber lalu lintas yang mengarahkan pengguna ke halaman web (Haidar & Dhiya, 2008). Dalam log web server, *referrer* adalah informasi tentang halaman web yang mengarahkan pengguna ke halaman web yang sedang dikunjungi.

### 2.3.8 *User Agent*

*User agent* merujuk pada string yang mengidentifikasi perangkat lunak klien yang mengakses server. *User agent* berupa sebuah web browser atau bot (Morville & Rosenfeld, 2006). Dalam log web server, *user agent* dicatat untuk setiap permintaan dan dapat digunakan untuk mengetahui browser dan sistem operasi yang paling banyak digunakan klien.

## 2.4 **Pemodelan dan Simulasi**

### 2.4.1 **Pemodelan**



Model adalah suatu deskripsi atau analogi yang digunakan untuk membantu menggambarkan sesuatu yang tidak dapat diamati secara

langsung. Dalam KBBI (Kamus Besar Bahasa Indonesia), model adalah pola (contoh, acuan, ragam, dan sebagainya) dari sesuatu yang akan dibuat atau dihasilkan. Pada umumnya model didefinisikan sebagai suatu representasi sistem nyata. Pemodelan dapat didefinisikan sebagai proses pembentukan model dari sistem tersebut.

#### 2.4.2 Simulasi

Dalam KBBI (Kamus Besar Bahasa Indonesia), simulasi adalah metode pelatihan yang meragakan sesuatu dalam bentuk tiruan yang mirip dengan keadaan yang sesungguhnya. Berdasarkan *The Oxford American Dictionary* (1980) yang dikutip oleh Harrell, C., Ghosh, B. K., & Bowden, R (2004) di dalam buku *Simulation Using Promodel* simulasi didefinisikan sebagai cara untuk mereproduksi kondisi situasi dengan menggunakan model untuk pembelajaran, pengujian, atau pelatihan. Model yang digunakan adalah model komputer. Simulasi juga dapat didefinisikan sebagai imitasi dari sistem dinamis menggunakan model komputer untuk mengevaluasi dan meningkatkan kinerja sistem.

#### 2.4.3 Validasi dan Verifikasi

Validasi dan verifikasi merupakan fundamental dalam pemodelan dan simulasi. Validasi bertujuan untuk memastikan bahwa model merefleksikan kenyataan secara akurat. Validasi dilakukan dengan membandingkan output simulasi dengan data yang diamati dari sistem nyata (Sargent, 2005). Hal ini bisa melibatkan penerapan metode statistik untuk mengevaluasi kemiripan antara model dan sistem nyata, seperti menilai kesesuaian distribusi waktu antara kedatangan *event* dalam simulasi dengan data historis.

Verifikasi melibatkan proses pengecekan kesesuaian implementasi model dengan rancangan awal tanpa kesalahan dalam pengkodean atau logika proses. Tahapan ini bertujuan untuk memastikan bahwa model beroperasi sesuai dengan yang diharapkan. Verifikasi sering dilakukan melalui pengujian t, di mana komponen individu dari model diuji secara terpisah untuk memastikan konsistensi dan keakuratan (Sargent, 2005).



Validasi dan verifikasi merupakan proses yang berkelanjutan di mana setiap perubahan dalam model memerlukan tinjauan ulang untuk memastikan bahwa modifikasi tidak menimbulkan masalah baru dan model masih merepresentasikan sistem dengan tepat (Balci, 1998).

## 2.5 Metode *Discrete Event Simulation*

*Discrete Event Simulation (DES)* atau simulasi kejadian diskrit merupakan suatu pendekatan simulasi berdasarkan kejadian yang ditunjuk sebagai kondisi yang memiliki variabel yang dapat berubah seketika pada waktu yang terpisah. Pendekatan simulasi ini dapat menggunakan model matematika atau logika fisik dalam memberikan solusi dari permasalahan (Lestari, 2018). Dalam konteks *Discrete Event Simulation*, kejadian (*event*) menggambarkan sebuah aliran proses yang merupakan suatu penggambaran sistem yang dipengaruhi oleh perubahan di luar sistem. Aliran proses ini juga melibatkan entitas yang merupakan objek dalam sistem dan atribut yang merupakan suatu sifat dari entitas tersebut.

Aliran proses memberikan urutan peristiwa yang akan disimulasikan. Peristiwa yang umum terjadi dalam simulasi ini meliputi kedatangan entitas ke sebuah stasiun kerja (*work station*), kegagalan sumber daya, dan aplikasi lainnya. Dalam peristiwa tersebut, terjadi keterlambatan akibat perubahan kondisi sistem saat menjalankan simulasi untuk mereplikasi suatu rentang waktu tertentu. Perubahan kondisi sistem ini akan menghasilkan keputusan akhir dalam pencapaian simulasi (Vázquez-Serrano et al., 2021).

### 2.5.1 Antrian

Antrian adalah suatu garis tunggu dari pelanggan yang memerlukan layanan dari satu atau lebih fasilitas pelayanan. Sedangkan, sistem antrian merupakan kedatangan pelanggan untuk mendapatkan pelayanan, menunggu untuk dilayani jika fasilitas pelayanan (*server*) masih sibuk, mendapatkan pelayanan dan kemudian meninggalkan sistem setelah dilayani. Pada umumnya, sistem antrian dapat diklasifikasikan menjadi sistem yang berbeda-beda di mana teori antrian dan simulasi sering diterapkan secara luas (Goswami, 2008).



### 2.5.2 Proses

Dalam *Discrete Event Simulation*, proses tidak hanya merupakan elemen yang berjalan sendiri, tetapi juga berperan sebagai representasi dari entitas atau tugas yang harus dikerjakan dalam sistem yang disimulasikan (Vázquez-Serrano et al., 2021). Pentingnya interaksi antar-proses memungkinkan simulasi untuk merepresentasikan perilaku dinamis sistem, di mana proses-proses dapat saling berkomunikasi, mengambil keputusan berdasarkan informasi yang diterima, dan bereaksi terhadap peristiwa-peristiwa dalam lingkungan mereka.

### 2.5.3 Lingkungan

Jerry (2015) menyatakan bahwa lingkungan (*environment*) dalam simulasi merupakan kerangka kerja di mana proses-proses berjalan dan peristiwa-peristiwa sistem dipicu dan dikelola. Secara lebih rinci, lingkungan (*environment*) memberikan kerangka kerja untuk mengatur proses-proses simulasi. Hal ini mencakup parameter-parameter dasar seperti waktu simulasi yang menentukan berapa lama simulasi berlangsung dan peristiwa-peristiwa diskrit yang relevan dengan sistem yang sedang disimulasikan. Dalam simulasi web server, peristiwa-peristiwa diskrit dapat meliputi kedatangan permintaan baru dari klien, kegagalan perangkat keras, atau perubahan keadaan sistem lainnya yang dapat memengaruhi kinerja web server.

### 2.5.4 Sumber Daya

Sumber daya (*resource*) dalam konteks simulasi merujuk pada representasi dari entitas fisik atau logis dalam sistem yang diperlukan atau digunakan oleh proses-proses dalam menjalankan aktivitas. Dalam simulasi sistem, sumber daya digunakan untuk merepresentasikan sumber daya yang terbatas atau dapat digunakan oleh proses-proses dalam sistem. Sumber daya ini dapat berupa berbagai hal, mulai dari perangkat keras fisik seperti server komputer, mesin produksi, hingga sumber daya logis seperti antrian atau asi penyimpanan. Setiap *resource* memiliki atribut yang menggambarkan kapasitas, ketersediaan, waktu pemrosesan, dan karakteristik lainnya yang



relevan dengan fungsinya dalam sistem. Penggunaan *resource* dalam simulasi memungkinkan pengguna untuk memodelkan sifat-sifat kritis dari sumber daya dalam sistem, seperti keterbatasan, persaingan untuk penggunaan, atau kebijakan alokasi (Averill, 2015).

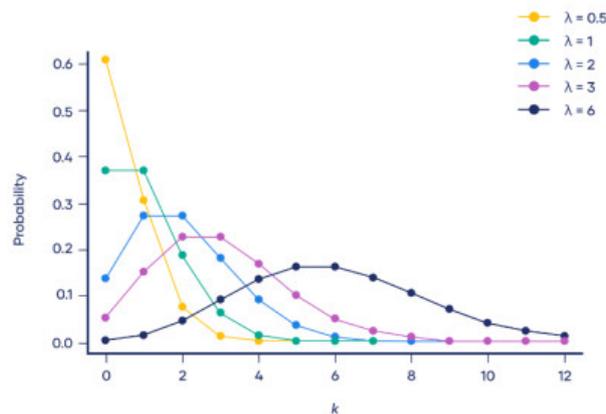
## 2.6 Random Number Generator

*Random Number Generator* (RNG) merupakan sebuah algoritma yang menghasilkan deret angka yang tampaknya acak dalam suatu rentang tertentu. Algoritma ini memungkinkan simulasi untuk memodelkan ketidakpastian atau variasi dalam sistem (Averill, 2015). Tujuan *Discrete Event Simulation* adalah untuk mengembangkan metode menggunakan *Random Number Generator* untuk mengubah program simulasi kejadian diskrit yang digerakkan oleh jejak menjadi program simulasi kejadian diskrit yang tidak bergantung pada data eksternal. RNG penting dalam DES karena sering digunakan untuk mensimulasikan perilaku stokastik dalam sistem, seperti waktu kedatangan permintaan, waktu pemrosesan, atau distribusi variabel lainnya yang bersifat acak.

### 2.6.1 Distribusi Poisson

Distribusi Poisson adalah distribusi probabilitas yang menggambarkan jumlah kejadian yang terjadi dalam interval waktu atau ruang tertentu, dengan asumsi bahwa kejadian tersebut terjadi secara acak dan independen dalam interval tersebut (Pahdian dkk., 2021). Dalam praktiknya, distribusi poisson digunakan ketika peristiwa-peristiwa yang diamati jarang terjadi dalam interval waktu yang diberikan, tetapi memiliki tingkat kedatangan yang relatif stabil. Dalam simulasi ini, distribusi poisson digunakan untuk memodelkan jumlah permintaan yang masuk ke server.





sumber: scribbr.com (2022)

Gambar 6. Distribusi poisson

$$P(X = k) = \frac{e^{-\lambda} \lambda^k}{k!} \quad (1)$$

$$P(x) = \frac{e^{-\lambda} \lambda^x}{x!} \quad (2)$$

dimana,

$X$  : variabel acak yang mengikuti distribusi poisson,

$k$  : berapa kali suatu peristiwa terjadi,

$e$  : konstanta Euler (2,718),

$\lambda$  : rata-rata berapa kali suatu peristiwa terjadi,

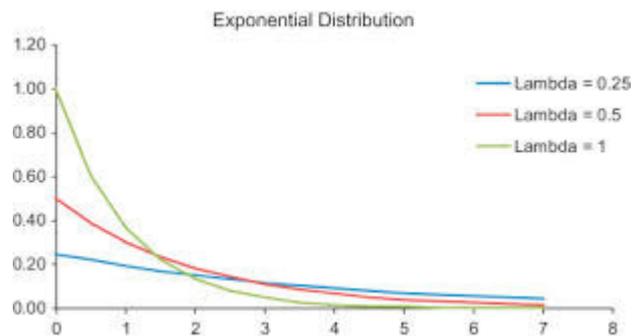
! : fungsi faktorial

### 2.6.2 Distribusi Eksponensial

Distribusi eksponensial adalah distribusi probabilitas kontinu yang sering digunakan untuk memodelkan waktu tunggu antara kejadian-kejadian yang terjadi secara independen dan acak (Walpole et al., 2020). Distribusi ini sering digunakan untuk memodelkan waktu hingga suatu peristiwa terjadi. Distribusi eksponensial dicirikan oleh sifatnya yang tidak memiliki ingatan, di mana probabilitas suatu peristiwa yang terjadi pada interval waktu berikutnya tidak bergantung pada berapa banyak waktu yang telah berlalu (syad et al., 2019). Dalam simulasi ini, distribusi eksponensial digunakan



untuk memodelkan waktu kedatangan permintaan ke server dan waktu layanan permintaan yang masuk ke server.



sumber: sciencedirect.com (2017)

Gambar 7. Distribusi eksponensial

$$f(x; \lambda) = \lambda e^{-\lambda x} \text{ untuk } x \geq 0 \quad (3)$$

dimana,

$x$  = variabel acak (waktu antara kedatangan),

$\lambda$  = laju rata-rata peristiwa per unit waktu,

$e$  = basis logaritma natural

## 2.7 SimPy

SimPy adalah sebuah pustaka (*library*) Python yang dirancang untuk simulasi berbasis diskrit-event. Pustaka ini digunakan untuk memodelkan dan mensimulasikan peristiwa diskrit dan proses paralel dalam konteks penelitian ilmiah, teknis, dan operasional. SimPy secara khusus dirancang untuk menangani simulasi berbasis peristiwa, di mana waktu bergerak maju berdasarkan kejadian (*events*) dan bukan pada langkah waktu yang tetap. SimPy memungkinkan pengguna untuk menentukan proses yang berinteraksi selama waktu simulasi dan memfasilitasi pembuatan model kompleks melalui pendekatan yang sederhana dan intuitif (Matloff, 2008).

SimPy memungkinkan pembuatan proses yang dapat berjalan secara mandiri

iteraksi satu sama lain menggunakan generator Python dan mekanisme

Keunggulan ini membedakan SimPy dari alat simulasi lainnya dengan

kan fleksibilitas yang lebih besar dalam pembuatan model dan



menjalankan simulasi. Manfaat lain dari SimPy termasuk dukungan untuk pengembangan modular, penggunaan sistem kontrol, dan fungsionalitas yang dapat diperluas sepenuhnya serta menyediakan pendekatan simulasi yang dapat beradaptasi dan interaktif (Castillo, 2006).

## 2.8 Uji Hipotesis

Hipotesis berasal dari bahasa Yunani yang terdiri dari dua kata yaitu “*hupo*” yang berarti lemah atau kurang dan “*thesis*” yang berarti teori atau pernyataan yang disajikan sebagai bukti. Dengan demikian, Hipotesis dapat diartikan sebagai pernyataan yang kebenarannya masih perlu dibuktikan (Fay & Proschan, 2010). Uji hipotesis adalah metode pengambilan keputusan berdasarkan hasil analisa data eksperimen. Uji hipotesis berguna untuk memvalidasi suatu pernyataan atau asumsi yang telah didefinisikan. Proses pengujian ini bertujuan mencari bukti yang cukup untuk mendukung hipotesis alternatif atau untuk menolak hipotesis nol.

### 1. Hipotesis nol ( $H_0$ )

Uji hipotesis dilakukan dengan membuat asumsi yang disebut hipotesis nol ( $H_0$ ). Hipotesis nol menyatakan bahwa tidak terdapat efek atau perbedaan yang signifikan dalam situasi yang sedang diuji (Cohen et al., 2003).

### 2. Hipotesis alternatif ( $H_1$ )

Hipotesis alternatif ( $H_1$ ) merupakan hipotesis yang menunjukkan adanya perbedaan atau efek nyata (Cohen et al., 2003).

Dalam pengujian hipotesis terdapat dua jenis kesalahan pengambilan keputusan yaitu:

#### 1. Kesalahan tipe I ( $\alpha$ )

Kesalahan tipe I merupakan kesalahan yang menolak  $H_0$ . Kesalahan ini menyimpulkan adanya perbedaan yang terjadi padahal sesungguhnya tidak ada perbedaan. Peluang kesalahan tipe I adalah  $\alpha$  atau disebut tingkat signifikansi (*significiance level*). Tingkat signifikansi ( $\alpha$ ) merupakan nilai yang menunjukkan besarnya peluang salah dalam menolak  $H_0$ . Nilai  $\alpha$  dapat diartikan juga sebagai nilai batas maksimal kesalahan menolak  $H_0$ . Dengan demikian, peluang untuk tidak membuat kesalahan tipe I adalah sebesar  $1 - \alpha$ .



$\alpha$  yang disebut sebagai tingkat kepercayaan (*confidence level*) (Peck & Devore, 2015).

## 2. Kesalahan tipe II (*beta*)

Kesalahan tipe II merupakan kesalahan yang tidak menolak  $H_0$ . Kesalahan ini menyimpulkan tidak ada perbedaan yang terjadi padahal sesungguhnya ada perbedaan. Peluang kesalahan tipe II adalah sebesar  $\beta$ . Sedangkan, peluang untuk tidak membuat kesalahan tipe II adalah sebesar  $1 - \beta$  yang disebut sebagai tingkat kekuatan (*power of the test*) (Peck & Devore, 2015).

### 2.8.1 Uji T

Uji T merupakan salah satu jenis uji parametrik dalam uji hipotesis. Uji T sering digunakan untuk menguji hipotesis dengan membandingkan rata-rata antara dua kelompok yang dianggap memiliki distribusi normal. Metode ini bisa digunakan baik untuk sampel yang berpasangan maupun yang tidak berpasangan, dengan asumsi bahwa kedua kelompok memiliki varians yang sama (Montgomery, 2012).

Uji T sangat berguna ketika ukuran sampel relatif kecil dan standar deviasi dari populasi tidak diketahui, situasi yang sering terjadi dalam penelitian praktis. Metode ini dapat diterapkan dalam berbagai skenario, seperti membandingkan rata-rata dari sebuah sampel dengan nilai teoretis atau membandingkan dua sampel yang independen atau yang berpasangan. Uji T cocok digunakan ketika data diketahui atau diasumsikan mengikuti distribusi normal (Armitage & Matthews, 2002).

$$t = \frac{\bar{x}_1 - \bar{x}_2}{s_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \quad (4)$$

$$s_p = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} \quad (5)$$

dimana,

$\bar{x}$  = rata-rata sampel,

$n$  = ukuran sampel,

$s_p$  = standar deviasi gabungan,

$s^2$  = variansi sampel



### 2.8.2 Uji *Chi Square*

Uji *Chi-square* adalah teknik statistik yang digunakan untuk membandingkan frekuensi yang diamati dengan frekuensi yang diharapkan dalam berbagai kategori dari satu atau lebih variabel. Tujuannya adalah untuk mengidentifikasi perbedaan antara data aktual dan data prediksi merupakan hasil dari kebetulan atau disebabkan oleh hubungan antara variabel yang dianalisis (Rana & Sigal, 2015).

Uji *Chi-square* merupakan jenis uji non-parametrik yang digunakan untuk menguji hipotesis tentang distribusi variabel kategori. Variabel kategori adalah jenis variabel yang dibagi ke dalam kategori-kategori tertentu dan tidak memiliki distribusi normal karena hanya memiliki sejumlah nilai yang spesifik. Variabel kategori bisa berbentuk nominal atau ordinal. Variabel nominal adalah tipe variabel kategori yang tidak mempunyai urutan alami. Sementara itu, variabel ordinal adalah tipe variabel yang kategorinya dapat diurutkan (Franke et al., 2011).

$$\chi^2 = \frac{\sum(O_i - E_i)^2}{E_i} \quad (6)$$

dimana,

O = nilai yang diamati,

E = nilai yang diharapkan

## 2.9 *Load Balancing* dan Pemanfaatannya

*Load balancing* adalah teknik yang digunakan untuk mendistribusikan beban kerja secara merata ke berbagai sumber daya komputasi, seperti server, jaringan, atau perangkat penyimpanan untuk mengoptimalkan penggunaan sumber daya, memaksimalkan *throughput*, meminimalkan waktu respons, dan menghindari kelebihan beban pada server tunggal (Bryhni et al., 2000).

Menurut Cardellini et al. (1999), *load balancing* dinamis pada sistem web server terdistribusi memungkinkan situs web untuk mengakomodasi variasi dalam



kerja dengan menyesuaikan jumlah server yang beroperasi secara *real-time*. Ini memastikan bahwa setiap server menjalankan beban kerja optimal tanpa kelebihan beban, sementara pengguna akhir mempertahankan pengalaman yang

konsisten dan responsif. Cara kerja *load balancing* bervariasi tergantung pada lingkungan sistemnya. Secara umum terdapat beberapa metode yang digunakan yaitu:

### 2.9.1 Round Robin

*Round robin* adalah algoritma penjadwalan yang banyak digunakan untuk mengelola waktu prosesor di antara beberapa proses yang berjalan secara simultan (Eldahshan et al, 2017). Algoritma ini menetapkan sebuah interval waktu yang tetap ke setiap proses dalam antrian. Jika proses tidak selesai dalam satu waktu, ia akan ditempatkan kembali di akhir antrian. Pendekatan ini memastikan bahwa semua proses mendapat kesempatan yang sama dalam menggunakan sumber daya yang tersedia (Purnomo & Putra, 2022).

Salah satu keunggulan utama dari *Round robin* adalah keadilan dalam penanganan proses. Setiap proses diberikan alokasi waktu yang sama sehingga tidak ada proses yang mendominasi atau diabaikan oleh penjadwal (Nayak & Padhy, 2014). Penulis menggunakan algoritma *round-robin* dalam perancangan simulator web server dalam penelitian ini.

### 2.9.2 Pemanfaatan Load Balancing

Simulator menyediakan platform yang efektif untuk menguji berbagai algoritma *load balancing* yang bertujuan mendistribusikan beban kerja secara optimal di antara server-server dalam sebuah kluster. Penelitian yang dilakukan Bryhni (2000) memperlihatkan bagaimana berbagai strategi *load balancing* dapat berdampak pada kinerja web server dalam skala besar.

Pembuatan simulator web server yang memanfaatkan log akses dari server nyata adalah cara efektif untuk menguji strategi *load balancing* dalam situasi yang menyerupai kondisi server sebenarnya (Ngamsuriyaroj et al., 2011). Simulator web server dalam penelitian ini menggunakan data dari log akses untuk mensimulasikan trafik pengguna. Dengan cara ini, pengujian bisa dilakukan dalam kondisi yang mirip dengan operasi nyata tanpa mengganggu sistem yang sedang aktif.



Dalam konteks *load balancing*, penggunaan simulator yang berbasis SimPy dan memanfaatkan log akses dapat diaplikasikan untuk menguji beragam algoritma *load balancing*. Mulai dari metode sederhana seperti *round-robin* hingga metode yang lebih kompleks seperti yang berbasis prioritas atau kapasitas (Yang et al., 2006).

## 2.10 Penelitian Terkait

Berikut penelitian terkait yang dijadikan bahan perbandingan dalam penelitian ini.

Judul	Penulis	Metode	Hasil
Towards a DEVS Model Management System for Decision-Making Web Applications	Laurent Capocchi dan Jean-Francois Santucci	Dalam penelitian ini, metode yang digunakan adalah pendekatan DEVS ( <i>Discrete Event System Specification</i> ) untuk memodelkan dan mensimulasikan sistem acara diskrit. Objek penelitian yang diteliti adalah kemampuan DEVS dalam melakukan pemodelan dan simulasi sistem acara diskrit melalui platform web kolaboratif.	Platform manajer aplikasi web berbasis DEVS-MMS yang dapat digunakan melalui <i>web-front-office</i> yang dihasilkan menggunakan mekanisme kuasi-otomatis.
A web-based simulation of a discrete-event system of mobile application DEVSIMPy-mob	Laurent Capocchi dan Jean-Francois Santucci	Menggunakan lingkungan DEVSIMPy untuk mengimplementasikan model simulasi DEVSIMPy (dalam format .yaml) yang akan mereka simpan di server file perpustakaan. Model simulasi ini, yang mungkin bergantung pada pustaka dan plugin model DEVS, dapat	Menyajikan aplikasi seluler DEVSIMPy-mob yang memungkinkan simulasi DES berbasis web dari SoS. Aplikasi seluler ini meningkatkan penggunaan model simulasi dan membuatnya lebih tersedia untuk pengguna non-



Judul	Penulis	Metode	Hasil
		disimulasikan melalui web menggunakan perangkat lunak DEVSimPy (DEVSimPy-nogui) versi baris perintah yang dapat diakses melalui REST antarmuka server DEVSimPy-Rest.	spesialis dalam skala besar.
Discrete Event Simulation and Digital Twins: Review and Challenges for Logistics	K. Agalianos, S. T. Ponis, E. Aretoulaki, G. Plakas, dan O. Efthymiou	Menggunakan <i>Discrete event Simulation</i> (DES) dalam proses logistik di gudang serta mengintegrasikan <i>Discrete Event Simulation</i> dan <i>Digital Twin</i> .	<i>Digital Twin</i> (DT) dapat digunakan sebagai struktur awal membangun model simulasi <i>Discrete Event Simulation</i> (DES). Simulasi dengan metode <i>Discrete Event Simulation</i> (DES) dapat meningkatkan kinerja proses manufaktur dan logistik.
Modeling and Simulation of Perfomance analysis for a cluster-based Web Server	Jianhua Yang, Di Jin, Ye Li, Kai Steffen H., & Reinhart German	Metode yang digunakan adalah <i>Discrete-Event Simulation</i> (DES) untuk system web server berbasis cluster. Penelitian ini menggunakan Automod dan R untuk menganalisis karakteristik kinerja server web cluster. Dalam proses simulasi, digunakan teknik pemodelan input, teori distribusi acak, dan analisis output.	Hasil dari penelitian ini menunjukkan bahwa jumlah server nyata dapat memengaruhi kinerja sistem dan menyarankan penyempurnaan untuk model simulasi di masa depan. Studi ini juga mengidentifikasi kemampuan proses maksimum dan menganalisis kemungkinan hambatan kinerja. Berbagai analisis statistik dan



Judul	Penulis	Metode	Hasil
			histogram data keterlambatan disajikan untuk memberikan informasi dalam proses pemodelan.

