

Analysis Report

Requests		Executions			Response Times (ms)							Throughput	Network (KB/sec)	
Label	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent	
Total	40	0	0.00%	1158.50	53	5340	169.00	4516.80	4633.55	5340.00	3.31	95.80	3.36	
GET My Ticket	5	0	0.00%	92.40	71	148	78.00	148.00	148.00	148.00	1.26	5.37	1.11	
Get Ticket Active	5	0	0.00%	264.40	219	329	244.00	329.00	329.00	329.00	1.23	183.66	1.35	
POST Create Ticket	5	0	0.00%	4525.00	4023	5340	4527.00	5340.00	5340.00	5340.00	0.59	1.06	0.51	
POST Create User	5	0	0.00%	3932.20	3073	4635	4342.00	4635.00	4635.00	4635.00	0.80	1.06	0.77	
POST Login	5	0	0.00%	227.40	190	279	207.00	279.00	279.00	279.00	1.49	55.88	3.23	
POST Login-0	5	0	0.00%	89.80	68	129	84.00	129.00	129.00	129.00	1.55	1.08	1.23	
POST Login-1	5	0	0.00%	78.40	60	117	71.00	117.00	117.00	117.00	1.56	1.10	1.05	
POST Login-2	5	0	0.00%	58.40	53	64	60.00	64.00	64.00	64.00	1.59	57.47	1.12	

Statistik 5 Users

Pada pengujian load testing ini, saya menjalankan beberapa skenario beban yaitu **5 users**, **25 users**, dan **50 users** untuk beberapa endpoint utama seperti **Create User**, **Login**, **Create Ticket**, **Get My Ticket**, dan **Get Ticket Active**. Hasilnya menunjukkan bahwa **pada 5 users aplikasi masih stabil karena tidak ada error** dan semua request berhasil diproses. Namun, dari sisi performa sudah terlihat bahwa endpoint **POST Create User** dan **POST Create Ticket** memiliki waktu respon paling tinggi (dalam hitungan beberapa detik), sehingga endpoint “create” ini terlihat lebih berat dibanding endpoint “get”.

Requests		Executions			Response Times (ms)							Throughput	Network (KB/sec)	
Label	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent	
Total	200	86	43.00%	229.58	44	6624	74.00	214.00	312.95	4467.09	15.78	208.06	16.58	
GET My Ticket	25	22	88.00%	81.60	61	215	72.00	124.80	189.20	215.00	2.12	4.40	1.94	
Get Ticket Active	25	21	84.00%	108.16	61	405	73.00	236.60	377.70	405.00	2.10	53.36	2.38	
POST Create Ticket	25	22	88.00%	437.28	60	3684	71.00	2754.80	3540.60	3684.00	2.13	3.20	1.88	
POST Create User	25	21	84.00%	782.36	115	6624	135.00	3496.40	5979.30	6624.00	2.95	4.25	2.93	
POST Login	25	0	0.00%	213.84	186	363	206.00	251.00	342.00	363.00	3.02	113.33	6.76	
POST Login-0	25	0	0.00%	74.12	62	134	71.00	85.00	120.20	134.00	3.07	2.14	2.45	
POST Login-1	25	0	0.00%	71.92	57	93	71.00	85.60	91.50	93.00	3.07	2.16	2.17	
POST Login-2	25	0	0.00%	67.40	44	223	60.00	74.40	178.60	223.00	3.08	111.28	2.26	

Statistik 25 Users

Saat beban dinaikkan menjadi **25 users**, mulai muncul masalah besar yaitu **banyak request gagal** dan error didominasi oleh **HTTP 500 Internal Server Error**. Endpoint yang paling sering gagal adalah endpoint yang berhubungan dengan proses create dan ticket. Ini menandakan bahwa ketika jumlah user bersamaan meningkat, **server/back-end atau database tidak sanggup menampung request** sehingga request banyak yang gagal, bukan hanya melambat.

Requests	Executions				Response Times (ms)							Throughput	Network (KB/sec)	
Label	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent	
Total	400	172	43.00%	710.31	46	10959	102.00	2661.80	3307.95	9286.89	22.89	301.74	24.04	
GET My Ticket	50	44	88.00%	417.62	62	3623	92.00	2792.40	3275.35	3623.00	2.98	6.04	2.73	
Get Ticket Active	50	45	90.00%	393.90	61	3501	99.50	759.70	3231.05	3501.00	3.01	76.31	3.41	
POST Create Ticket	50	39	78.00%	1649.92	63	10959	104.00	6483.60	8929.75	10959.00	2.99	4.59	2.64	
POST Create User	50	44	88.00%	1299.00	113	10850	148.50	4115.30	9520.80	10850.00	4.37	6.30	4.34	
POST Login	50	0	0.00%	961.18	184	2791	242.50	2731.80	2768.90	2791.00	4.41	165.52	9.87	
POST Login-0	50	0	0.00%	563.76	60	2629	86.00	2529.50	2571.40	2629.00	4.46	3.11	3.55	
POST Login-1	50	0	0.00%	338.74	57	2648	81.00	2331.50	2586.50	2648.00	4.46	3.15	3.15	
POST Login-2	50	0	0.00%	58.38	46	96	56.00	70.50	80.40	96.00	4.46	161.25	3.28	

Statistik 50 Users

Pada skenario **50 users**, kondisi makin terlihat jelas karena error tetap tinggi dan beberapa endpoint mengalami kegagalan yang sangat besar. Selain itu, nilai response time juga semakin “melebar” (P95 menjadi beberapa detik dan max sampai belasan detik), yang berarti sebagian user masih dapat respon cepat, tetapi banyak juga yang mengalami respon lambat atau gagal. Secara keseluruhan, hasil ini menunjukkan bahwa aplikasi **belum stabil untuk beban concurrent menengah–tinggi**, terutama pada endpoint yang melakukan proses tulis data (create) dan proses ticket.

Rekomendasi dan Saran

Top 5 Errors by sampler													
Sample	#Samples	#Errors	Error	#Errors	Error	#Errors	Error	#Errors	Error	#Errors	Error	#Errors	Error
Total	200	86	500/Internal Server Error	85	Test failed: code expected to equal /n/n***** received : 20[[[7]]]n/n***** comparison: 20[[[0]]]n/n/	1							
GET My Ticket	25	22	500/Internal Server Error	21	Test failed: code expected to equal /n/n***** received : 20[[[7]]]n/n***** comparison: 20[[[0]]]n/n/	1							
Get Ticket Active	25	21	500/Internal Server Error	21									
POST Create Ticket	25	22	500/Internal Server Error	22									
POST Create User	25	21	500/Internal Server Error	21									

Errors 25 Users

Top 5 Errors by sampler													
Sample	#Samples	#Errors	Error	#Errors	Error	#Errors	Error	#Errors	Error	#Errors	Error	#Errors	Error
Total	400	172	500/Internal Server Error	149	Test failed: code expected to equal /n/n***** received : 20[[[7]]]n/n***** comparison: 20[[[0]]]n/n/	23							
GET My Ticket	50	44	500/Internal Server Error	33	Test failed: code expected to equal /n/n***** received : 20[[[7]]]n/n***** comparison: 20[[[0]]]n/n/	11							
Get Ticket Active	50	45	500/Internal Server Error	33	Test failed: code expected to equal /n/n***** received : 20[[[7]]]n/n***** comparison: 20[[[0]]]n/n/	12							
POST Create Ticket	50	39	500/Internal Server Error	39									
POST Create User	50	44	500/Internal Server Error	44									

Errors 50 Users

Berdasarkan hasil load testing, rekomendasi utama saya adalah melakukan perbaikan pada sisi **resource dan handling database** karena error yang muncul banyak berupa **500 Internal Server Error** ketika concurrent user meningkat. Tim developer perlu mengecek kemungkinan bottleneck seperti **database connection limit/connection pool**, query yang berat, atau proses write yang memakan waktu lama. Selain itu, endpoint yang paling perlu diprioritaskan untuk optimasi adalah **POST Create User** dan **POST Create Ticket**, karena sejak skenario 5 users pun sudah terlihat sebagai endpoint yang paling berat.

Dari sisi optimasi performa, beberapa hal yang bisa dilakukan adalah melakukan **optimasi query database**, menambahkan **index** yang sesuai, serta memastikan proses berat (misalnya pengiriman notifikasi/email atau proses validasi kompleks) tidak membuat request utama menjadi blocking terlalu lama. Jika memungkinkan, proses berat dapat dipindahkan ke mekanisme asynchronous/job queue.

Dari sisi load testing menggunakan JMeter, saya juga menyarankan untuk menyesuaikan **assertion status code** agar sesuai dengan response yang valid (misalnya jika ada endpoint yang mengembalikan kode selain 200 namun masih dianggap sukses oleh sistem). Selain itu, untuk membuat hasil test lebih realistis, data untuk request create sebaiknya dibuat **unik/dinamis** dan penambahan **ramp-up yang lebih halus** dapat membantu melihat titik mulai sistem menurun secara bertahap, bukan langsung “kaget” ketika beban naik.

Secara umum, aplikasi masih stabil pada skenario **5 users**, namun pada skenario **25 dan 50 users** mulai muncul error tinggi dan performa menurun terutama di endpoint yang melakukan proses create dan ticket. Rekomendasi utama adalah peningkatan kapasitas/konfigurasi database connection dan optimasi endpoint yang berat agar aplikasi lebih siap menghadapi concurrent user yang lebih besar.