

Analysis Report

Performance testing dilakukan menggunakan **JMeter** pada website **resonance.dibimbing.id** dengan tiga skenario beban, yaitu **5 users**, **25 users**, dan **50 users** untuk beberapa endpoint utama: **Create User**, **Login**, **Create Ticket**, **Get My Ticket**, dan **Get Ticket Active**.

Komponen yang dilihat pada report performance testing mencakup **Response Time**, **Latency**, **Error Rate**, dan **Throughput** serta **nilai maksimum** untuk melihat lonjakan performa saat beban meningkat.

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	754	0	0.00%	788.83	47	7015	145.00	3588.50	4571.50	6014.55	6.25	179.31	6.35	
GET My Ticket	93	0	0.00%	87.47	67	317	78.00	111.40	151.00	317.00	0.82	3.51	0.73	
Get Ticket Active	93	0	0.00%	178.97	132	333	166.00	232.80	265.20	333.00	0.82	122.68	0.90	
POST Create Ticket	94	0	0.00%	2677.01	868	7015	2491.00	4998.00	5672.00	7015.00	0.80	1.43	0.68	
POST Create User	98	0	0.00%	2774.18	882	6941	2851.00	5258.50	5978.90	6941.00	0.81	1.07	0.78	
POST Login	94	0	0.00%	247.60	172	2750	203.00	266.50	365.00	2750.00	0.81	30.40	1.76	
POST Login-0	94	0	0.00%	108.49	58	2582	71.00	103.00	148.25	2582.00	0.81	0.57	0.64	
POST Login-1	94	0	0.00%	79.94	59	327	69.00	105.50	139.00	327.00	0.81	0.57	0.55	
POST Login-2	94	0	0.00%	58.59	47	90	56.00	71.00	78.25	90.00	0.81	29.30	0.57	

Statistik 5 Users

Pada skenario **5 users**, sistem masih terlihat **stabil** karena **error rate total 0.00%** dari **754 samples**. Namun, terdapat indikasi bottleneck pada endpoint tertentu karena nilai **response time maksimum** masih bisa tinggi.

Pada skenario ini, total pengujian menghasilkan **754 samples** dengan **error 0 (0.00%)**, serta **Average Response Time 788.83 ms**, **Min 47 ms**, dan **Max 7015 ms** dengan **throughput 6.25 transaksi/detik**. Untuk endpoint **GET My Ticket**, performanya tergolong cepat dengan **Average 87.47 ms**, **Median 78 ms**, **90th percentile 111.40 ms**, dan **Max 317 ms**. Endpoint yang paling berat justru berada pada proses **POST Create Ticket** dan **POST Create User**, dengan **Average 2677.01 ms** (Max **7015 ms**) untuk Create Ticket dan **Average 2774.18 ms** (Max **6941 ms**) untuk Create User, sehingga dua endpoint ini menjadi prioritas utama untuk optimasi karena menyumbang latency terbesar pada hasil total.

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	35702	15756	44.13%	137.16	43	8769	68.00	245.00	472.85	1275.87	197.10	2477.39	206.97
GET My Ticket	4453	3943	88.55%	74.73	53	3285	66.00	86.00	105.30	291.00	25.07	51.99	22.97
Get Ticket Active	4451	3888	87.35%	116.57	52	3686	67.00	195.00	307.40	1042.56	25.07	513.13	28.40
POST Create Ticket	4459	3944	88.45%	205.65	53	8422	66.00	875.00	896.00	1529.80	24.83	37.30	21.94
POST Create User	4471	3981	89.04%	197.52	52	8769	66.00	880.00	898.40	1537.56	24.73	35.76	24.56
POST Login	4467	0	0.00%	251.33	102	2801	206.00	348.00	512.00	932.88	24.94	935.91	55.80
POST Login-0	4467	0	0.00%	72.40	52	2536	64.00	84.00	104.00	272.32	24.96	17.43	19.89
POST Login-1	4467	0	0.00%	68.84	51	2431	63.00	79.00	93.00	196.64	24.99	17.62	17.65
POST Login-2	4467	0	0.00%	110.03	43	2677	72.00	169.00	362.00	717.00	24.99	902.84	18.35

Statistik 25 Users

Errors			
Type of error	Number of errors	% in errors	% in all samples
500/Internal Server Error	15756	100.00%	44.13%

Errors 25 Users

Pada skenario **25 users**, performa aplikasi mulai bermasalah karena **error meningkat sangat signifikan**. Hasil total menunjukkan **35.702 samples** dengan **15.756 fail**, sehingga **error rate total 44,13%**. Nilai **throughput total 197,10 transactions/s** terlihat tinggi, namun throughput ini **tidak bisa dianggap sehat** karena hampir setengah request berakhir gagal.

Beberapa endpoint yang mengalami error dominan adalah sebagai berikut (sesuai data di tabel):

- **GET My Ticket: 4.453 samples, 3.943 fail (error rate 88,55%), average 74,73 ms, max 3.285 ms, throughput 25,07 transactions/s.**
Ini menunjukkan request sering gagal walaupun rata-rata respons terlihat cepat (karena banyak response error yang balik cepat).
- **Get Ticket Active: 4.451 samples, 3.888 fail (error rate 87,35%), average 116,57 ms, max 3.686 ms, throughput 25,07 transactions/s.**
Error tinggi di endpoint ini mengindikasikan proses ambil data tiket aktif kemungkinan berat di server/DB saat beban naik.
POST Create Ticket: 4.459 samples, 3.944 fail (error rate 88,45%), average 205,65 ms, max 8.422 ms, throughput 24,83 transactions/s.
Selain error tinggi, nilai max sangat besar, menandakan ada request yang “tersendat” cukup lama sebelum gagal/selesai.
- **POST Create User: 4.471 samples, 3.981 fail (error rate 89,04%), average 197,52 ms, max 8.769 ms, throughput 24,73 transactions/s.**
Endpoint ini menjadi yang paling tinggi error rate-nya, sehingga proses create user belum stabil saat concurrency tinggi.

Sementara itu, endpoint login terlihat paling stabil:

- **POST Login: 4.467 samples, fail 0 (error rate 0,00%), average 251,33 ms, max 2.801 ms, throughput 24,94 transactions/s.**
- **POST Login-0 / Login-1 / Login-2 juga error rate 0,00% pada 4.467 samples masing-masing.**

Kesimpulan untuk beban 25 users: walaupun banyak response time rata-rata terlihat kecil (puluhan–ratusan ms), **error rate yang sangat tinggi (44,13% total, bahkan 87–89% di endpoint ticket & create)** menunjukkan sistem **tidak sanggup memproses request dengan benar** pada beban ini. Fokus perbaikan sebaiknya diarahkan ke endpoint ticket (GET My Ticket & Get Ticket Active) dan proses create (POST Create Ticket & POST Create User) karena menjadi sumber kegagalan terbesar.

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	36333	16643	45.81%	235.63	43	13433	73.00	875.00	1139.95	2318.97	200.13	2284.69	210.17	
Get My Ticket	4518	4234	93.71%	89.69	52	4229	68.00	111.00	281.05	329.00	26.13	50.60	23.94	
Get Ticket Active	4512	4231	93.77%	205.67	52	7656	68.00	287.00	761.55	3637.28	26.72	301.25	30.27	
POST Create Ticket	4529	4091	90.33%	228.69	52	13433	67.00	318.00	894.00	1734.50	25.66	38.38	22.67	
POST Create User	4558	4087	89.67%	257.97	52	11274	67.00	876.00	903.05	6070.62	25.11	36.32	24.93	
POST Login	4554	0	0.00%	550.82	152	6005	344.00	1119.00	1534.50	2566.20	25.79	967.88	57.70	
POST Login-0	4554	0	0.00%	76.02	52	2375	65.00	90.00	114.00	285.45	25.85	18.05	20.59	
POST Login-1	4554	0	0.00%	73.38	50	878	64.00	84.00	108.00	285.00	25.83	18.22	18.24	
POST Login-2	4554	0	0.00%	401.29	43	5868	200.00	964.50	1374.25	2395.50	25.81	932.47	18.96	

Statistik 50 Users

Type of error	Number of errors	% in errors	% in all samples
500/Internal Server Error	16326	98.10%	44.93%
Test failed: code expected to equal /n/n***** received : 20[[7]]n/n***** comparison: 20[[0]]n/n/	317	1.90%	0.87%

Errors 50 Users

Pada skenario **50 users**, kondisi sistem semakin berat. Tabel total menunjukkan **36.333 samples** dengan **16.643 request gagal**, sehingga **error rate total 45,81%**. Throughput total **200,13 transaksi/detik** masih tinggi, namun kembali tidak sehat karena **proporsi error meningkat** dan **nilai max response time naik sangat tinggi**.

Detail per endpoint pada skenario 50 users:

- GET My Ticket: 4.518 samples, 4.234 fail (93,71% error), average 89,69 ms, max 4.229 ms, throughput 26,13 req/detik.
- Get Ticket Active: 4.512 samples, 4.231 fail (93,77% error), average 205,67 ms, max 7.656 ms, throughput 26,72 req/detik.
- POST Create Ticket: 4.529 samples, 4.091 fail (90,33% error), average 228,69 ms, max 13.433 ms, throughput 25,66 req/detik.
- POST Create User: 4.558 samples, 4.087 fail (89,67% error), average 257,97 ms, max 11.274 ms, throughput 25,11 req/detik.
- POST Login: 4.554 samples, 0 fail (0,00% error), average 550,82 ms, max 6.005 ms, throughput 25,79 req/detik.

Kesimpulan skenario 50 users: sistem sudah masuk kondisi **overload**, terlihat dari dominasi error **500**, serta **max response time** yang melonjak sampai **13.433 ms** pada beberapa endpoint.

Penjelasan Error yang Muncul

Top 5 Errors by sampler												
Sample	#Samples	#Errors	Error	#Errors	Error	#Errors	Error	#Errors	Error	#Errors	Error	#Errors
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								

Sampler Errors 25 Users

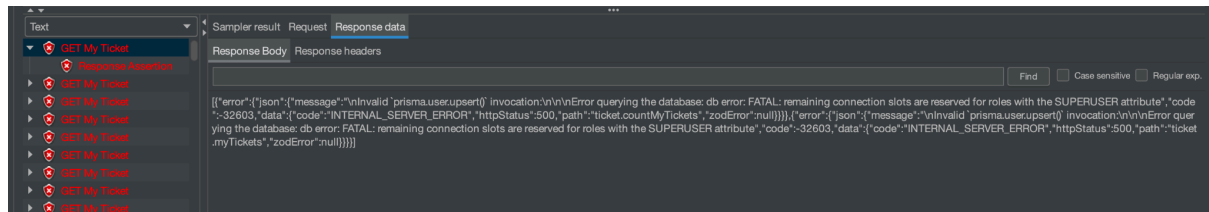
Top 5 Errors by sampler												
Sample	#Samples	#Errors	Error	#Errors	Error	#Errors	Error	#Errors	Error	#Errors	Error	#Errors
Total	400	172	500/Internal Server Error	149	Test failed: code expected to equal /n/n***** received : 20[[[7]]]n/n***** comparison: 20[[[0]]]n/n/	23						
GET My Ticket	50	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								

Sample Errors 50 Users

Error pada **GET My Ticket** dan **Get Ticket Active** umumnya terjadi karena endpoint GET membaca data lebih sering dan lebih “berat”, sehingga pada beban tinggi request menumpuk dan backend kewalahan. Dari error yang muncul sebelumnya seperti **500/Internal Server Error** dan pesan terkait database (contoh kasus: koneksi database habis), penyebab paling masuk akal adalah **koneksi database mencapai batas maksimum** atau **query yang berat tidak ditangani dengan pooling yang baik**, sehingga request gagal di sisi server.

Selain itu, endpoint GET biasanya dipanggil berulang pada tiap iterasi test sehingga frekuensinya tinggi. Jika hasil GET juga besar (banyak data tanpa pagination/filter), beban CPU/DB akan lebih tinggi dan peluang error meningkat.

Error paling sering yang muncul adalah **HTTP 500 Internal Server Error**, yang berarti proses pada sisi backend gagal memproses request. Pada hasil pengujian, error ini dominan muncul pada beban 25 dan 50 users, sehingga indikasinya kuat bahwa backend tidak mampu menangani request concurrent dalam jumlah besar.



Selain itu, terdapat error database dengan pesan **“remaining connection slots are reserved for roles with the SUPERUSER attribute”**. Pesan ini menunjukkan bahwa koneksi database sudah mencapai batas maksimal, sehingga request baru tidak memperoleh koneksi untuk diproses. Masalah ini umumnya berkaitan dengan keterbatasan **max connection database**, pengaturan **connection pooling** yang kurang tepat, adanya potensi **connection leak**, atau adanya query/transaksi yang menahan koneksi terlalu lama. Error ini muncul pada proses yang berkaitan dengan operasi database seperti `prisma.user.upsert()` dan proses ticket/user.

Rekomendasi dan Saran

Rekomendasi berikut disusun berdasarkan pola utama: **error 500**, endpoint GET dominan error, serta response time max yang naik sampai beberapa detik.

1. Perbaiki koneksi database dan connection pooling

Perlu dilakukan pengecekan batas koneksi database (misal max connections) dan konfigurasi pooling dari aplikasi/ORM. Jika koneksi cepat habis saat load tinggi, sebaiknya diterapkan pooling yang lebih aman (misalnya pengaturan pool size, timeout, atau penggunaan pooler seperti PgBouncer jika relevan).

2. Optimasi query untuk endpoint GET (My Ticket & Ticket Active)

Endpoint GET perlu dicek apakah mengambil data terlalu besar tanpa pagination/filter. Jika data besar, response bisa berat dan mempercepat overload. Pagination, indexing, dan optimasi query dapat menurunkan beban DB dan mengurangi error.

3. Tambahkan caching untuk data yang sering dibaca

Jika endpoint GET memuat data yang sering dipanggil dan tidak selalu berubah, caching (server-side atau layer tertentu) dapat mengurangi beban query berulang ke database.

4. Perbaiki error handling dan observability

Error 500 perlu dipetakan lebih detail dengan logging dan monitoring (contoh: APM, metrics CPU/RAM, DB connections, slow query log). Dengan ini akar m

Kesimpulannya, sistem masih cukup stabil pada beban kecil (5 users), namun belum stabil untuk beban menengah \approx tinggi (25–50 users) karena error rate tinggi dan terdapat indikasi bottleneck pada koneksi database. Fokus perbaikan disarankan pada peningkatan kapasitas serta pengelolaan koneksi database, dan optimasi endpoint create/ticket agar sistem lebih siap menangani concurrent user yang lebih besar.