

Implementasi Heuristic Miner Untuk Menganalisis Pola Perilaku Mahasiswa Dalam Mengerjakan Ujian Tengah Semester Pada Learning Management System

1st Asbahid Rifqi Bilgiflah
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia

asbahidr@students.telkomuniversity.ac.id

2nd Anisa Herdiani
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia

anisaherdiani@telkomuniversity.ac.id

3rd Rosa Reska Riskiana
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia

rosareska@telkomuniversity.ac.id

Abstrak — Untuk memastikan kualitas dari aplikasi dibutuhkan adanya proses pengujian baik dari fungsional maupun non fungsional. Pengujian ini sangatlah penting untuk memastikan bahwa aplikasi tersebut siap untuk digunakan. Salah satu jenis pengujian non fungsional untuk aplikasi yang penting adalah pengujian performansi. Pengujian performansi terdiri dari Load testing dan Stress testing. Melalui pengujian Performance testing menggunakan Load Testing dan Stress Testing pada aplikasi DMAX oleh PT. Telkom Indonesia Tbk. dapat menilai hasil performansi yang sesuai dengan requirement yang dibutuhkan dan apakah masih terdapatnya gejala bottleneck yang dapat mengurangi performansi sehingga dapat ditingkatkan. Hasilnya performansi aplikasi DMAX sudah sangat baik dan memenuhi nilai acceptance criteria yang ditentukan dan bisa diterapkan di 7 regional Telkom Indonesia.

Kata kunci— Performance testing, Load Testing, Stress Testing

I. PENDAHULUAN

A. Latar Belakang

PT. Telkom Indonesia Tbk, sedang mengembangkan aplikasi DMAX berbasis web untuk pengelolaan unserved demand untuk diintegrasikan dengan proses planning, deployment, dan fulfillment sehingga dapat membantu proses permintaan calon pelanggan Indihome. Saat ini aplikasi DMAX sudah digunakan pada 2 dari total 7 regional dan ditargetkan dapat diperluas sehingga mencakup seluruh regional Telkom se-Indonesia.

Sebelum aplikasi DMAX diterapkan di regional lainnya aplikasi DMAX harus dipastikan memenuhi standar yang sudah ditetapkan oleh IT Strategy & Governance -NITS PT. Telkom Indonesia pada tahun 2020. Dalam standar tersebut terdapat proses Testing yang terdiri dari 3 bagian yaitu System Testing, User Acceptance Testing, dan Installation on Staging Environment. Dari hasil wawancara yang dilakukan, Performance Efficiency merupakan bagian yang belum termasuk dalam standar NITS yang ditetapkan oleh PT. Telkom Indonesia. sehingga, belum memenuhi standar ISO 25010 secara keseluruhan.

Penelitian ini berfokus pada pengujian Performance Testing yaitu Load Testing dan Stress Testing untuk

memastikan aplikasi DMAX dapat memenuhi standar ISO 25010. Output yang didapat berupa nilai response time, throughput, dan error rate aplikasi DMAX saat diakses oleh beberapa user secara bersamaan menggunakan tools open source Apache JMeter. Melalui penilaian terhadap hasil output yang didapat dari pengujian, diharapkan dapat membantu PT. Telkom Indonesia selaku pengembang DMAX untuk bahan evaluasi sebelum digunakan diseluruh 7 regional Telkom se-Indonesia.

B. Topik dan Batasannya.

Beberapa rumusan masalah yang diangkat pada penelitian ini, yaitu:

1. Bagaimana performansi aplikasi DMAX berdasarkan hasil dari performance testing yang dilakukan ?
2. Apakah aplikasi DMAX sudah dapat diterapkan pada regional lain berdasarkan hasil dari performance testing yang dilakukan?

Batasan masalah agar penelitian tugas akhir ini dapat lebih terfokus, sebagai berikut:

Pengujian performansi yang dilakukan terhadap DMAX berfokus kepada sub-karakteristik time behavior dikarenakan tidak mendapatkan perizinan untuk mengakses dan mengamati performansi server saat melakukan pengujian, sehingga tidak memungkinkannya untuk dapat mengukur performansi dari sub-karakteristik dari performance efficiency yaitu resource utilization dan capacity.

C. Tujuan

Tujuan dari tugas akhir ini adalah Melakukan performance testing pada aplikasi DMAX dan Melakukan analisa dari hasil pengujian performance testing untuk mengetahui apakah aplikasi DMAX sudah dapat digunakan di 7 regional Telkom.

D. Organisasi Tulisan

Pada Bagian Studi Terkait berisi teori/studi/literatur pendukung TA ini. Bagian Metodologi menjelaskan Metodologi Penelitian yang digunakan pada TA ini. Bagian Evaluasi berisi pengujian, hasil pengujian, dan analisis

pengujian. Terakhir Bagian Kesimpulan berisi kesimpulan akhir dari seluruh TA ini

II. KAJIAN TEORI

A. Performance Testing

Dikutip pada [3]. Performance Testing adalah proses yang penting dalam siklus hidup pengembangan perangkat lunak atau systems development life cycle. Performance Testing selalu berkaitan dengan pengembangan sistem perangkat lunak. Pada buku [3] juga dijelaskan bahwa pengembangan sistem perangkat lunak tanpa pengujian performa akan menghasilkan sistem atau aplikasi yang tidak berkinerja dengan baik. Itu dapat mengakibatkan penambahan beban biaya, waktu dan uang serta pendapat buruk dari pengguna sistem.

Berdasarkan ISO/IEC 25010 [4], matriks terkait kualitas sebuah software yang mengacu pada efisiensi performansi adalah:

1. Time behaviour: Tingkatan dari response, waktu proses, dan throughput rate sistem.
2. Resource utilization: Tingkatan dari penggunaan resource yang dibutuhkan sistem, resource yang dimaksud adalah Processor, Memory, Disk, dan Network.
3. Capacity: Total workload maksimum yang dapat dihandle oleh sistem.

Dijelaskan pada [3], manfaat performance testing dari pengujian sistem antara lain: Mengukur performansi sistem, Menemukan permasalahan yang terkait dengan performansi sistem, an- tara lain: lack of server-side resource, bad network bandwidth, dan lain sebagainya.

Untuk mencegah terjadinya gejala bottleneck dan melihat sejauh mana sistem mampu menangani beban dari pengguna, maka perlu dilakukan performance testing. Semakin besar bottleneck maka semakin kecil tingkat performansi sistem yang berhasil dicapai.

Berdasarkan [1], Untuk menguji efisiensi performansi berdasarkan ISO/IEC 25010 pada matriks Time Behaviour, Resource Utilization, dan Capacity diperlukannya pengujian pada sistem yang didasarkan pada pengujian beban, terdapat dua jenis pengujian yaitu:

a. Stress Testing

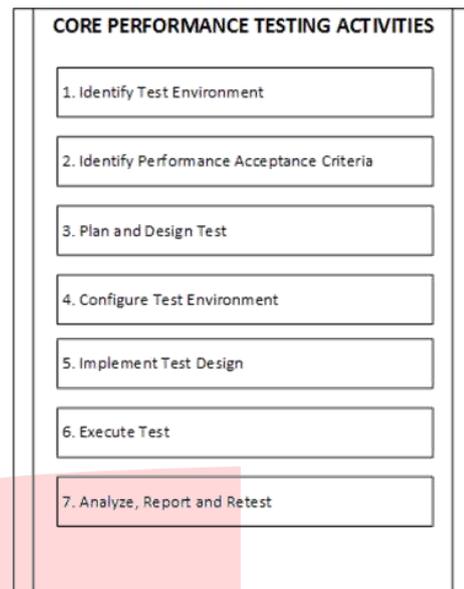
Stress Testing adalah jenis pengujian performansi yang berfokus pada pemberian beban melebihi kapasitas dari spesifikasi sistem dan melakukan proses sistem secara terus menerus [1][2].

b. Load Testing

Load Testing adalah salah satu teknik pengujian sistem dimana pengujian difokuskan pada pemberian beban yang disesuaikan dengan beban yang sesungguhnya di lingkungan pengguna [1].

B. Performance Testing Core Activities

Pada Performance Testing Core Activities terdapat tujuh proses yang harus dilakukan saat melakukan Performance Load Test [1]. Diagram pada gambar 2-1 berikut menggambarkan tujuh proses yang harus dilalui untuk melakukan performance test:



GAMBAR 2-1
Tujuh proses utama performance test [1]

Sesuai pada gambar 2-1 sebelum melakukan pengujian, perlu adanya proses identifikasi terhadap lingkungan pengujian dari sistem yang akan diuji dengan melakukan identifikasi *resource hardware* maupun *resource software* yang digunakan oleh sistem.

Setelah melakukan proses identifikasi lingkungan sistem, langkah selanjutnya adalah menentukan bagian apa saja dari sistem yang akan diuji dan menentukan nilai performansi minimal yang harus diraih oleh sistem yang akan diuji. Pada tahap *Plan and Design Test*, Hasil dari pengidentifikasian lingkungan sistem dan menentukan nilai performansi yang harus diraih oleh sistem digunakan untuk melakukan perancangan desain pengujian yang tepat agar bisa didapatkan hasil yang baik dan sesuai dengan kebutuhan.

Setelah selesai membuat desain pengujian, saatnya untuk mempersiapkan dan mengkonfigurasi setiap komponen yang akan digunakan dan juga mengimplementasi desain pengujian yang sudah dibuat untuk melakukan *Performance Testing* agar siap untuk dilakukan pengujian.

Setelah semua sudah siap barulah dapat dilakukan pengujian berdasarkan tahap-tahap sebelumnya. lalu hasil dari pengujian tersebut dapat dianalisis dan didokumentasikan atau melakukan pengujian ulang apabila hasil dari pengujian tersebut masih kurang memuaskan.

C. Load Testing Tools

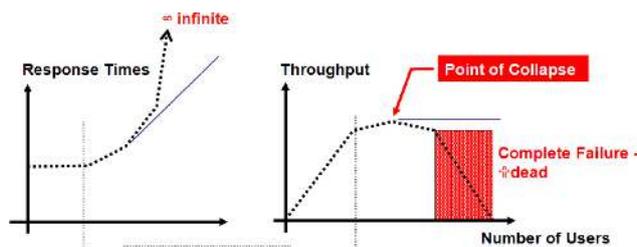
Load testing tool adalah sebuah alat yang digunakan untuk melakukan pengujian performansi pada suatu sistem dengan menggunakan virtual user untuk melihat bagaimana suatu sistem dapat menangani beban yang diberikan. Penguji dapat memasukkan parameter yang ingin diuji kedalam tools tersebut dan mensimulasikan skenario yang akan diuji. Setelah proses pengujian dilakukan, tools akan memberikan informasi mengenai hasil dari proses pengujian tersebut sesuai dengan parameter yang sudah ditentukan diawal. Hasil dari pengujian tersebut dapat dianalisis dan disimpulkan oleh penguji untuk menentukan kesimpulan performansi dari sistem [6].

Beberapa contoh alat yang dapat digunakan untuk melakukan Load Testing diantaranya seperti Apache JMeter,

HP Loadrunner, Siege, Microsoft Visual Studio, dan lain sebagainya. Pada tugas akhir ini digunakan Load Testing Tool Apache JMeter. Apache JMeter adalah salah satu Load Testing Tools yang *open source*, *user-friendly*, dan mempunyai komunitas *user* dan *developer* yang besar. JMeter adalah alat pengujian Load testing yang kuat dan fleksibel, JMeter dapat membantu *developer* untuk mengidentifikasi *bottleneck* pada sistem. JMeter juga dapat digunakan untuk menguji *web service*, *database*, atau *FTP server* dengan bantuan plugin yang tersedia. JMeter juga dapat mensimulasikan banyak *user* dengan beban yang berat pada aplikasi dengan mudah. Sehingga pengujian dapat lebih mudah untuk mensimulasikan pengujian dan menganalisis hasil dari pengujian [6].

D. Mendeteksi Gejala Bottleneck pada Sistem

Bottleneck adalah suatu kondisi dimana sistem tidak dapat memenuhi permintaan pengguna. Kinerja sistem dapat berkurang secara signifikan, dan bahkan hilang. Ini mungkin karena sistem tidak mampu menangani lonjakan beban yang datang dari pengguna. Lonjakan beban yang besar dalam jangka waktu yang panjang dapat menyebabkan kelebihan beban sistem (*overload*). Oleh karena itu, pentingnya melakukan testing untuk mengukur kinerja sistem yang sedang dibangun [7].

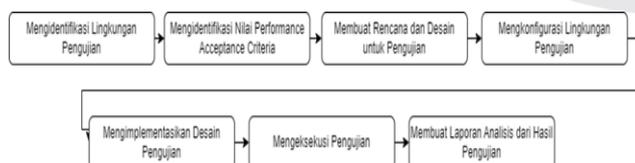


GAMBAR 2-2

Hubungan kenaikan *response time* dengan *throughput* terhadap gejala *bottleneck*[7]

Untuk mengidentifikasi gejala *bottleneck* memerlukan pengujian performansi sistem untuk dapat menentukan perilaku dan kapasitas dari sistem. Ada beberapa parameter yang dapat diamati untuk mendeteksi gejala *bottleneck*, seperti menurunnya nilai *throughput*, meningkatnya waktu *response time*, dan penurunan kapasitas CPU, dan lain hal sebagainya.

III. METODE



GAMBAR 3.1

Metodologi Penelitian

A. Mengidentifikasi Lingkungan Pengujian

Proses identifikasi lingkungan pengujian fisik dan lingkungan produksi serta alat dan sumber daya yang tersedia untuk tim pengujian sehingga lebih terukur sebagai langkah awal dalam melakukan penelitian.

B. Mengidentifikasi Nilai Performance Acceptance Criteria

Mengidentifikasi nilai performansi dari *Response time*, *Throughput*, dan *Resource Utilization* untuk dijadikan parameter standar bahwa nilai dari hasil pengujian dapat dinyatakan baik.

C. Membuat Rencana dan Desain untuk Pengujian

Membuat rencana dan desain untuk menentukan arah pengujian sehingga pengujian dilakukan sesuai dengan kebutuhan.

D. Mengkonfigurasi Lingkungan Pengujian

Mempersiapkan lingkungan pengujian, alat, dan sumber daya yang diperlukan untuk menjalankan setiap rencana dari desain pengujian yang sudah dibuat.

E. Mengimplementasikan Desain Pengujian

Mengimplementasikan semua rencana dan desain pengujian yang sudah dibuat untuk digunakan saat melakukan pengujian.

F. Mengeksekusi Pengujian

Menjalankan dan memantau pengujian yang dilakukan. Pada proses ini dilakukan pengujian terhadap aplikasi web sesuai dengan desain dan parameter yang sudah ditentukan sebelumnya. Dari hasil pengujian tersebut didapatkan nilai *response time*, *throughput*, dan *error rate* yang digunakan pada tahap selanjutnya.

G. Membuat Laporan Analisis dari Hasil Pengujian

Proses ini bertujuan untuk menganalisa nilai *response time*, *throughput*, dan *error rate* yang didapat pada proses pengujian. Hasil analisa yang didapat diambil kesimpulan dan dilanjutkan ke proses berikutnya, yaitu pembuatan laporan sesuai hasil analisa yang sudah disimpulkan dalam bentuk tugas akhir sesuai kaidah penulisan tata tulis ilmiah.

IV. HASIL DAN PEMBAHASAN

A. Mengidentifikasi Lingkungan Pengujian

Melakukan identifikasi pada lingkungan pengujian fisik, lingkungan produksi, alat, dan sumber daya. sebagai data untuk melakukan pengujian nanti. berikut adalah hal-hal yang diidentifikasi:

1. Spesifikasi Perangkat Keras

Perangkat Keras yang digunakan adalah sebagai berikut

- a. Processor Laptop 11th Gen Intel(R) Core(TM) i5-11400H 2.7GHz
- b. Memory 16.0 GB
- c. Hard Drive SSD 512

2. Spesifikasi Perangkat Lunak

Perangkat Lunak yang digunakan adalah sebagai berikut

- a. Windows 11
- b. Apache JMeter 5.4.3
- c. VPN Global Protect
3. Spesifikasi Jaringan Internet
Jaringan internet yang digunakan adalah jaringan internet telkom dengan kecepatan internet Up to 30 Mbps.
4. Data Pengguna

Dalam penelitian ini digunakan data yang didapat dari hasil rekaman jumlah pengguna DMAX harian pada rentan waktu bulan November 2022 yang disimpan dalam file format ‘.csv’. Data jumlah pengguna pada bulan November 2022 dapat dilihat pada lampiran IV. data bulan November adalah data dengan jumlah pengguna terbanyak yang diberikan oleh *developer* DMAX pada tahun 2022.

B. Mengidentifikasi Nilai Performance Acceptance Criteria

Untuk menentukan nilai *performance acceptance criteria* penulis menggunakan rule *Eight-Second Rule* yang ditemukan dan diformulasikan pada tahun 1970an dimana pada penelitian tersebut dibuat suatu metode transaksi yang selesai dalam waktu sepersekian detik hingga 20 detik. Kesimpulan dari penelitian tersebut didapat pada tahun 1990an dimana saat menggunakan internet produktivitas pengguna berkurang secara cepat jika *response time* lebih dari 8 detik per transaksi. *Response time* yang lebih dari 8 detik membuat pengguna lupa dengan apa yang sebenarnya sedang sistem lakukan sehingga *Eight-Second Rule* sangat relevan untuk menentukan kebutuhan *response time* pada produk digital. [8]

Berdasarkan penelitian *Eight-Second Rule*, 8 detik adalah nilai maksimal yang harus diraih DMAX pada setiap transaksi oleh karena itu akan dilakukannya penyesuaian berdasarkan tingkat kompleksitas pada setiap transaksi aplikasi DMAX berdasarkan jumlah data yang harus diisi, jumlah data yang harus dikirim, dan jumlah data yang diterima dan akan ditampilkan saat transaksi selesai. Berikut adalah tabel nilai *performance acceptance criteria* yang sudah disesuaikan dengan kompleksitas setiap transaksi untuk aplikasi web DMAX:

TABEL 4-1
Nilai Acceptance Criteria aplikasi web DMAX

Proses Transaksi	Data Isian	Data yang dikirim	Data yang diterima	Response Time
Login	3	450 bytes	4741 bytes	Maksimal 2 detik
Unservred MAP	4	595 bytes	5321 bytes	Maksimal 3 detik
Unservred Demand	5	868 bytes	6285 bytes	Maksimal 5 detik
Clustering	3	836 bytes	6278 bytes	Maksimal 5 detik
Demand	5	835 bytes	6285 bytes	Maksimal 5 detik

C. Membuat Rencana dan Desain Pengujian

Mengidentifikasi Jumlah User 7 Regional untuk Load Test

Berdasarkan data yang sudah didapat dari hasil rekaman jumlah pengguna DMAX harian pada rentan waktu bulan November 2022 untuk 2 regional Telkom didapatkan bahwa:

TABEL 4-2
Hasil Analisis jumlah pengguna DMAX November 2022

	Jumlah User
Minimal	0
Rata-rata	22
Maksimal	40

Karena adanya kemungkinan dari 2 regional yang ada saat mengakses DMAX ada salah satu regional yang tidak mengakses DMAX sama sekali pada hari itu maka jumlah user yang digunakan adalah jumlah user maksimal untuk memastikan saat melakukan load testing untuk 7 regional DMAX dapat menampung jumlah user semaksimal mungkin yaitu $40(\text{user maksimal}) / 2(\text{jumlah regional hasil data}) \times 7(\text{target regional load testing}) = 140 \text{ user}$.

1. Skenario Pengujian Load Testing

Pengujian load testing dilakukan dengan menggunakan target yang sudah ditentukan sebelumnya yaitu 140 user secara bersamaan mengakses DMAX.

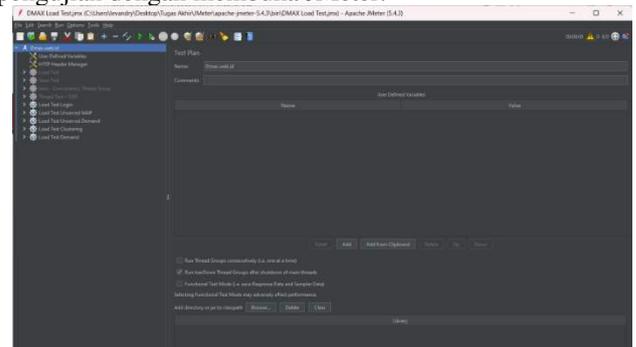
Setelah dilakukan pengujian menggunakan 140 user, penulis menganalisis hasil *response time* dari hasil pengujian, bila ditemukan adanya hasil pengujian yang *response time* nya lebih tinggi dari target yang sudah ditentukan sebelumnya penulis melakukan ulang load testing pada transaksi tersebut dengan mengurangi jumlah user sampai hasil dari *response time* lebih rendah dari target yang sudah ditentukan.

2. Skenario Pengujian Stress Testing

Pengujian stress testing dilakukan dengan menggunakan kelipatan 100 user dari user yang sudah digunakan pada load testing yaitu 140 menjadi 200, 300, 400, dst. sampai didapatkannya error saat mengakses setiap transaksi yang terdapat pada web DMAX.

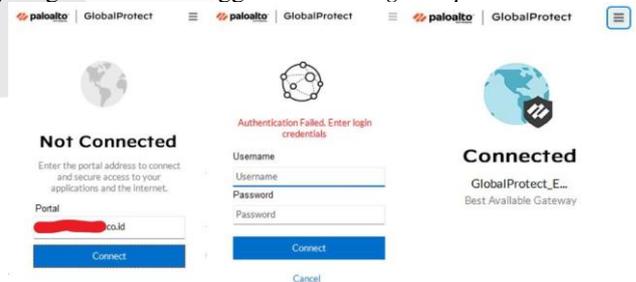
D. Mengkonfigurasi Lingkungan Pengujian Aplikasi

Langkah pertama adalah mempersiapkan lingkungan pengujian dengan membuka JMeter.



GAMBAR 4-1
Aplikasi JMeter

Kedua adalah mengkoneksikan Komputer/Laptop dengan jaringan telkom menggunakan VPN *globalprotect*.



GAMBAR 4-2
VPN *globalprotect*

lalu membuka halaman DMAX untuk memastikan sudah terkoneksi dengan jaringan telkom, karena DMAX hanya bisa diakses apabila sudah terkoneksi dengan baik.

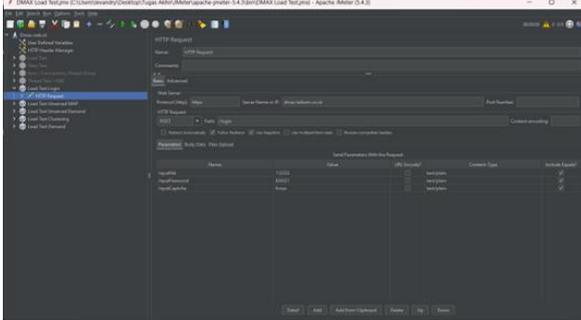


GAMBAR 4-3
Tampilan utama DMAX

E. Mengimplementasikan Desain Pengujian

1. Mengimplementasikan Test Plan Dan Target Pengujian
Langkah pertama adalah membuat test plan dan selanjutnya dilakukan testing terhadap aplikasi. Skenario pengujian aplikasi adalah:
 - a. Login
 - b. Underserved MAP
 - c. Underserved Demand
 - d. Clustering
 - e. Demand

Rancangan skenario testing yang dilakukan sebagai berikut:



GAMBAR 4-4
Rancangan skenario testing

Saat melakukan set konfigurasi tahapannya adalah sebagai berikut:

- a. Set Number of Thread (user) / jumlah user yang mengakses modul/fitur tersebut.
- b. Set Ramp-Up Period (in seconds) / jumlah detik yang dibutuhkan untuk semua user yang telah di set sebelumnya. Misalkan pada contoh diatas, jumlah user di set sebesar 140 dan jumlah detik untuk semua user adalah 1 detik. Maka dapat dihitung bahwa setiap user mengakses fitur tersebut dalam waktu 1 detik secara bersamaan.
- c. Set Loop Count / berapa kali thread dijalankan 1 sampai infinite.
- d. Set HTTP Request / untuk menentukan proses transaksi yang diuji. Contoh untuk transaksi login maka Path di isi dengan /login sesuai dengan web DMAX bagian login
- e. Set Parameter / menentukan parameter sesuai dengan proses transaksi yang diuji dengan mensimulasikan parameter yang diminta saat melakukan transaksi tersebut. Contoh login membutuhkan NIK, Password, dan Captcha.

F. Mengeksekusi Pengujian

1. Pengujian Load Testing
Melakukan eksekusi kepada semua bagian DMAX, sebagai berikut:
 - a. Pengujian Load Testing Login

Load Test pada bagian Login dengan memasukkan number of threads = 140, ramp-up period = 1, dan loop count = 1, itu artinya ada 140 user yang dijalankan dalam waktu 1 detik dengan tanpa adanya pengulangan.

b. Pengujian Load Testing Underserved MAP

Load Test pada bagian Underserved MAP dengan memasukkan number of threads = 140, ramp-up period = 1, dan loop count = 1, itu artinya ada 140 user yang dijalankan dalam waktu 1 detik dengan tanpa adanya pengulangan.

c. Pengujian Load Testing Underserved Demand

Load Test pada bagian Underserved Demand dengan memasukkan number of threads = 140, ramp-up period = 1, dan loop count = 1, itu artinya ada 140 user yang dijalankan dalam waktu 1 detik dengan tanpa adanya pengulangan.

d. Pengujian Load Testing Clustering

Load Test pada bagian Clustering dengan memasukkan number of threads = 140, ramp-up period = 1, dan loop count = 1, itu artinya ada 140 user yang dijalankan dalam waktu 1 detik dengan tanpa adanya pengulangan.

e. Pengujian Load Testing Demand

Load Test pada bagian Demand dengan memasukkan number of threads = 140, ramp-up period = 1, dan loop count = 1, itu artinya ada 140 user yang dijalankan dalam waktu 1 detik dengan tanpa adanya pengulangan.

2. Hasil Pengujian Load Testing

Berdasarkan hasil dari pengujian diatas maka dibuatkan data tabel sebagai berikut:

TABEL 4-3
Tabel Hasil Pengujian Load Testing

Parameter	Login	Underserved MAP	Underserved Demand	Clustering	Demand
Response Time	2000 ms	3000 ms	5000 ms	5000 ms	5000 ms
Nilai Acceptance Criteria					
Jumlah User	140 User				
Min Response Time	63 ms	279 ms	244 ms	128 ms	154 ms
Average Response Time	347 ms	518 ms	1197 ms	1062 ms	630 ms
Max Response Time	1125 ms	1384 ms	1664 ms	1653 ms	1637 ms
Throughput	67.0/sec	90.9/sec	56.6/sec	60.0/sec	76.1/sec
Error	0%	0%	0%	0%	0%

3. Pengujian Stress Testing

Pengujian stress testing dilakukan dengan menggunakan skenario dan konfigurasi yang sama seperti load testing dengan adanya perbedaan pada jumlah user yang menggunakan kelipatan 100 user dari user pertama yang sudah digunakan pada load testing yaitu 140 menjadi 200, 300, 400, dst. pengujian stress testing ini dilakukan untuk mengetahui sejauh mana kekuatan web DMAX dalam menangani jumlah user yang banyak dalam satu waktu sampai didapatkannya error saat melakukan stress testing pada web DMAX.

4. Hasil Pengujian Stress Testing

Berdasarkan hasil dari pengujian stress testing didapatkan data yang banyak terkait nilai response time dan throughput,

hasil dari pengujian tersebut dijadikan diagram agar lebih mudah untuk dibaca dan dipahami dapat dilihat di lampiran III.

Berdasarkan diagram diatas dapat disimpulkan hasil dari Stress Testing dengan nilai max response time masih dibawah nilai *acceptance criteria* :

TABEL 4-4

Tabel Hasil Pengujian Stress Testing dibawah nilai *acceptance criteria*

Parameter	Login	Unserv MAP	Unserv Demand	Clusterin g	Demand
Response Time Nilai Acceptance Criteria	2000 ms	3000 ms	5000 ms	5000 ms	5000 ms
Jumlah User	300	400	500	400	500
Min Response Time	76 ms	129 ms	202 ms	102 ms	153 ms
Average Response Time	600 ms	1390 ms	2486 ms	1787 ms	2111 ms
Max Response Time	1290 ms	2610 ms	4430 ms	2996 ms	3710 ms
Throughput	163.8/sec	127.7/sec	101.8/sec	98.4/sec	106.7/sec
Error	0%	0%	0%	0%	0%

Hasil Stress Testing sebelum terdapatnya *error* dan saat terdapatnya *error* :

TABEL 4-5

Tabel Hasil Pengujian Stress Testing sebelum terjadinya *error*

Parameter	Login	Unserve MAP	Unserve Demand	Clusteri ng	Demand
Jumlah User	1900	800	700	500	900
Min Response Time	457 ms	2391 ms	955 ms	555 ms	1972 ms
Average Response Time	7295 ms	8247 ms	8820 ms	4738 ms	8599 ms
Max Response Time	12203 ms	13482 ms	16642 ms	8653 ms	14259 ms
Throughput	138.3/sec	56.3/sec	41.7/sec	54.0/sec	59.8/sec
Error	0%	0%	0%	0%	0%

TABEL 4-6

Tabel Hasil Pengujian Stress Testing sesudah terjadinya *error*

Parameter	Login	Unserv MAP	Unserv Demand	Clusterin g	Demand
Jumlah User	2000	900	800	600	1000
Min Response Time	156 ms	1401 ms	383 ms	387 ms	488 ms
Average Response Time	8182 ms	6712 ms	5707 ms	5260 ms	9204 ms
Max Response Time	13355 ms	13918 ms	9775 ms	9038 ms	14443 ms
Throughput	141.1/sec	62.7/sec	76.5/sec	58.7/sec	65.4/sec
Error	10.45%	15.44%	16.38%	0.50%	28.40%

G. Analisis Hasil Pengujian

1. Analisis Hasil Pengujian Load Testing

Berdasarkan Hasil dari pengujian Load Testing Aplikasi DMAX sudah memenuhi standar nilai *acceptance criteria* untuk digunakan di 7 regional PT. Telkom di Indonesia dimana dapat dilihat pada tabel 4-3 bahwa hasil pengujian

140 user secara bersamaan nilai max *response time* dari setiap bagian yang ada pada Aplikasi DMAX berada dibawah nilai *acceptance criteria* yang sudah ditentukan.

Sehingga dapat disimpulkan bahwa aplikasi DMAX sudah siap untuk diluncurkan ke 5 regional lainnya tanpa adanya masalah di *response time* saat menggunakan aplikasi tersebut.

H. Analisis Hasil Pengujian Stress Testing

Sedangkan dari hasil pengujian Stress Testing dapat disimpulkan bahwa aplikasi DMAX sebenarnya sudah dapat digunakan lebih dari target 140 user yang ditentukan.

Pada Tabel 4-4 aplikasi DMAX dapat dilihat bahwa Login bisa menampung sampai 300 lebih user, Unserved MAP dapat menampung 400 user, Unserved Demand dapat menampung 500 user, Clustering dapat menampung 400 user, dan Demand dapat menampung 500 user.

Dari data tersebut dapat disimpulkan bahwa aplikasi DMAX saat ini dapat digunakan secara bersamaan dengan mengambil jumlah user terkecil yaitu login dengan jumlah user sebanyak 300 user secara bersamaan dengan *max response time* masih dibawah nilai *acceptance criteria*, hasil tersebut adalah dua kali lipat dari target 140 user yang ditetapkan untuk 7 regional PT. Telkom saat ini.

1. Analisis Hasil Pengujian Stress Testing dengan *Eight-Second Rule*

Dari hasil pengujian Stress Testing juga dapat terlihat jumlah user yang dapat mengakses aplikasi DMAX secara bersamaan saat menggunakan *Eight-Second Rule* sebagai nilai *acceptance criteria*.

TABEL 4-7

Tabel Hasil Pengujian Stress Testing dibawah nilai *Eight-Second Rule*

Parameter	Login	Unserve MAP	Unserve Demand	Clusteri ng	Demand
Response Time Nilai Eight-Second Rule	8000 ms	8000 ms	8000 ms	8000 ms	8000 ms
Jumlah User	1000	600	500	400	500
Min Response Time	299	161	202	102	153
Average Response Time	2672	2577	2486	1787	2111
Max Response Time	5316	4087	4430	2996	3710
Throughput	165.4/sec	114.8/sec	101.8/sec	98.4/sec	106.7/sec
Error	0%	0%	0%	0%	0%

Berdasarkan Tabel 4-7 terlihat adanya peningkatan jumlah user maksimal yang mengakses secara bersamaan pada Login dari 300 user menjadi 1000 user dan Unserved Map dari 400 user menjadi 600 user karena menggunakan *Eight-Second Rule* sebagai nilai *acceptance criteria*. Hasil tersebut membuat jumlah user yang dapat mengakses aplikasi DMAX secara bersamaan menjadi 400 user, dengan mengambil jumlah user terkecil yaitu Clustering dan masih memenuhi standar *Eight-Second Rule* dimana produktivitas user berkurang jika *response time* dari aplikasi DMAX diatas 8 detik (8000 ms).

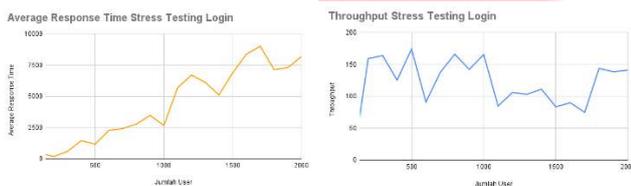
2. Analisis Hasil Pengujian Stress Testing dan Error

Dari hasil pengujian Stress Testing juga dapat menentukan jumlah maksimal user yang mengakses aplikasi DMAX secara bersamaan tanpa adanya user yang terkena *error*.

Berdasarkan Tabel 4-5 dan Tabel 4-6 dapat disimpulkan bahwa aplikasi DMAX memiliki jumlah user maksimal yaitu 500 user dengan mengambil jumlah terkecil user maksimal sebelum terjadinya *error* yaitu pada bagian Clustering, karena pada saat mensimulasikan transaksi Clustering dengan 600 user, didapatkan 3 user mengalami *error* atau sama sekali tidak dapat mengakses Clustering.

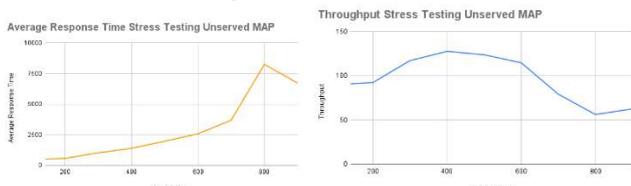
3. Analisis Bottleneck dari Hasil Pengujian Stress Testing

Berdasarkan data yang didapat dari hasil Stress Testing gejala *bottleneck* dapat diidentifikasi dengan melihat grafik *response time* dan *throughput* seiring saat dilakukannya testing, dengan melihat lonjakan naiknya waktu *response time* dan menurunnya nilai *throughput* disaat yang bersamaan.



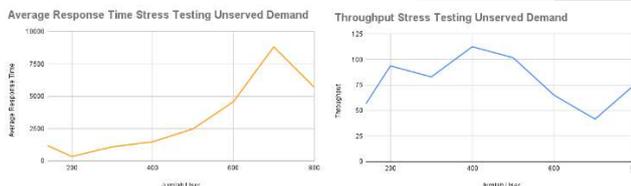
GAMBAR 4-5
Diagram Average Response Time dan Throughput Login

Pada gambar 4-5 dapat dilihat adanya lonjakan waktu *response time* pada 1000 user dan 1400 user, dan dapat dilihat juga adanya penurunan *throughput* yang signifikan pada 500 user dan 1000 user. sehingga dapat disimpulkan bahwa adanya kemungkinan terjadi *bottleneck* di bagian Login ketika 1000 user mengakses secara bersamaan.



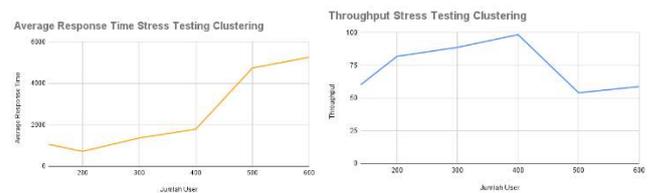
GAMBAR 4-6
Diagram Average Response Time dan Throughput Unserved MAP

Pada gambar 4-6 dapat dilihat adanya lonjakan waktu *response time* pada 700 user, dan dapat dilihat juga adanya penurunan *throughput* yang signifikan pada 600 user dan 700 user. sehingga dapat disimpulkan bahwa adanya kemungkinan terjadi *bottleneck* di bagian Unserved MAP ketika 700 user mengakses secara bersamaan.



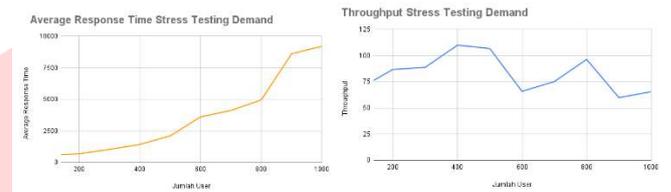
GAMBAR 4-7
Diagram Average Response Time dan Throughput Unserved Demand

Pada gambar 4-7 dapat dilihat adanya lonjakan waktu *response time* pada 500 user dan 600 user, dan dapat dilihat juga adanya penurunan *throughput* yang signifikan pada 500 user dan 600 user. sehingga dapat disimpulkan bahwa adanya kemungkinan terjadi *bottleneck* di bagian Unserved Demand ketika 500 user mengakses secara bersamaan.



GAMBAR 4-8
Diagram Average Response Time dan Throughput Clustering

Pada gambar 4-8 dapat dilihat adanya lonjakan waktu *response time* pada 400 user, dan dapat dilihat juga adanya penurunan *throughput* yang signifikan pada jumlah user yang sama yaitu 400 user. sehingga dapat disimpulkan bahwa adanya kemungkinan terjadi *bottleneck* di bagian Clustering ketika 400 user mengakses secara bersamaan.



GAMBAR 4-9
Diagram Average Response Time dan Throughput Demand

Pada gambar 4-9 dapat dilihat adanya lonjakan waktu *response time* pada 800 user, dan dapat dilihat juga adanya penurunan *throughput* yang signifikan pada 500 user dan 800 user. sehingga dapat disimpulkan bahwa adanya kemungkinan terjadi *bottleneck* di bagian Demand ketika 800 user mengakses secara bersamaan.

Berikut tabel kesimpulan berdasarkan uraian analisis kemungkinan terjadinya *bottleneck* pada setiap proses:

TABEL 4-8
Hasil Analisis Kemungkinan Terjadinya Bottleneck

Nama Proses	Kemungkinan Terjadinya Bottleneck
Login	1000 user
Unserved MAP	700 user
Unserved Demand	500 user
Clustering	400 user
Demand	800 user

V. KESIMPULAN

Berdasarkan hasil dari performance testing, performansi aplikasi DMAX sudah sangat baik dan memenuhi nilai acceptance criteria yang ditentukan. Hasil dari melakukan load testing terlihat bahwa aplikasi DMAX sudah dapat diterapkan di 7 regional, dengan 140 user mengakses secara bersamaan tanpa adanya masalah pada response time. Dan dari hasil stress testing aplikasi DMAX diketahui sebenarnya DMAX sudah dapat menampung dua kali lipat user yang mengakses secara bersamaan yaitu 300 user dan masih memenuhi nilai acceptance criteria yang ditentukan.

Disarankan agar kedepannya DMAX bisa melakukan performance testing untuk bagian sub karakteristik resource utilization dan capacity, agar dapat mengoptimalkan penggunaan sumber daya server seperti prosesor, memori, disk, jaringan dan dapat menjaga kinerja sistem kedepannya.

REFERENSI

- [1] J Meier, Carlos Farre, Prashant Bansode, Scott Barber, and Dennis Rea. 2007. Performance testing guidance for web applications: patterns & practices. Microsoft press.
- [2] Roger S Pressman. 2005. Software engineering: a practitioner's approach. Palgrave macmillan.
- [3] Ian Molyneaux. 2014. The art of application performance testing: from strategy to tools. "O'Reilly Media, Inc."
- [4] ISO 25000, "ISO/IEC 25010 - System and software quality models,". 2019. available at <https://iso25000.com/index.php/en/iso-25000-standards/iso-25010> [Retrieved July, 2022].
- [5] D.A. Menascé, "Load testing of Web sites". 2002. in IEEE Internet Computing 6(4). 70-74.
- [6] Abbas, R., Sultan, Z., & Bhatti, S. N. ". 2017. Comparative analysis of automated load testing tools: Apache JMeter, Microsoft Visual Studio (TFS), LoadRunner, Siege."
- [7] Duttagupta, Subhasri, Rupinder Virk, and Manoj Nambiar. 2015. "Software bottleneck analysis during performance testing." In *2015 International Conference and Workshop on Computing and Communication (IEMCON)*. IEEE. 1-7. Vuković, V., Đurković, J., Trinić, J. 2014. Defining Performance Criteria and Planning Performance Tests for the Exam Registration Software. *International Scientific Journal of Management Information Systems*, 9 (2), 15-19.