

Performance Testing Menggunakan Metode Load Testing dan Stress Testing pada Sistem Core Banking PT. XYZ

1st Dzaki Madhani
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia

dzakimadhani@students.telkomuniversi
ty.ac.id

2nd Eko Darwiyanto
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia

ekodarwiyanto@telkomuniversity.ac.id,

3rd Arfive Gandhi
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia

arfivegandhi@telkomuniversity.ac.id

Abstrak — Core banking adalah aplikasi inti dari sistem perbankan yang berfungsi untuk mengelola aktivitas transaction, customer information file, dan berbagai layanan untuk nasabah perbankan lainnya. Oleh karena itu penting untuk dilakukan performance testing agar dapat dipastikan bahwa sistem core banking tetap dapat berjalan dengan baik ketika jumlah aktivitas nasabah di perbankan tersebut meningkat. Performance testing merupakan salah satu pengujian non fungsional pada perangkat lunak yang bertujuan untuk mengukur kehandalan suatu aplikasi seperti waktu respon, transaksi per detik, dan kestabilan akses dari sistem dan aplikasi dibawah beban kerja yang diberikan. Ada beberapa metode dalam performance testing yang akan digunakan yaitu seperti load testing dan stress testing. Load testing dilakukan untuk menguji aplikasi dibawah beban yang diharapkan. Sebelum dijalankan load testing akan dijalankan stress testing terlebih dahulu untuk mendapatkan nilai maksimum beban yang dapat ditangani oleh aplikasi yang kemudian akan digunakan untuk menjalankan load testing. Setelah dilakukan eksekusi performance test, ditemukan hasil yang dapat memenuhi semua objektifnya. Dari hasil yang didapatkan menunjukkan bahwa sistem core banking pada PT. XYZ cukup stabil untuk menangani beban yang diharapkan.

Kata kunci—performance test, load testing, stress testing, loadrunner, core banking

I. PENDAHULUAN

A. Latar Belakang

Setiap perbankan memiliki sistem core banking yang merupakan sebuah aplikasi inti dan kunci dari sebuah perbankan. Sistem tersebut berfungsi untuk mengelola aktivitas-aktivitas inti perbankan yang meliputi transaction, saving, loan, dan customer information file [1]. Aktivitas inti tersebut erat hubungannya dengan nasabah bank yang secara tidak langsung merupakan user yang mengakses core banking tersebut. Terdapat momen di setiap tahun dalam satu hari di mana jumlah pengguna yang mengakses core banking secara bersamaan meningkat berkali-kali lipat. Pada PT. XYZ sedang melakukan pengembangan mobile banking versi terbaru untuk menggantikan mobile banking versi lama mereka. Pengembangan mobile banking baru tersebut juga diharapkan dapat menangani kenaikan jumlah nasabah setiap tahunnya pada PT. XYZ. Akan menjadi hal yang berbahaya jika sistem tersebut menjadi down karena akan dapat

mengurangi tingkat kepercayaan nasabah terhadap bank nya dan dapat mengakibatkan penurunan jumlah nasabah pada bank tersebut. Hal tersebut mengharuskan sistem core banking tetap berjalan dengan baik di saat itu.

Untuk memastikan sistem core banking tersebut dapat berjalan dengan baik maka perlu dilakukan pengujian performa (performance testing). Performance testing merupakan pengujian yang berfungsi untuk mengetahui seberapa jauh performa dari sebuah aplikasi ketika beban di bawah beban kerja yang diharapkan [2]. Hasil performance testing dapat dikatakan baik apabila dapat mencapai ekspektasi dari pengguna aplikasi atau sistem yang diuji. Tidak ada patokan yang pasti untuk menentukan hasil yang baik pada performance test. Setiap aplikasi memiliki karakteristiknya masing-masing dan baik tidak hasilnya bergantung pada kebutuhan bisnis yang ada pada aplikasi tersebut.

Pada performance testing dapat dilakukan dengan bermacam metode pengujian, diantara yaitu stress testing, load testing, endurance testing, spike testing, dan lain sebagainya. Adapun metode yang akan digunakan untuk melakukan performance testing yaitu stress testing dan load testing. Metode tersebut dipilih karena pada load testing dapat digunakan untuk mengetahui apakah sistem core banking PT. XYZ dapat menangani trafik sesuai dengan ekspektasi untuk menghadapi momen kenaikan nasabah setiap tahunnya. Sedangkan dari hasil metode stress testing akan dapat membantu menemukan jumlah maksimum pengguna berjalan bersamaan (concurrent user) dan juga untuk menghindari terjadinya kegagalan sistem ketika didorong sampai kondisi puncaknya.

Dalam menjalankan pengujian tersebut perlu adanya performance testing tools salah satunya adalah Loadrunner. Loadrunner merupakan performance testing tools yang dapat mendukung stress testing maupun load testing dan menyertakan fungsionalitas untuk memudahkan dalam menganalisis hasil dari pengujian yang sudah dijalankan. Di Loadrunner beban kerja user akan ditentukan oleh virtual user yaitu concurrent user yang akan mengirimkan request ke aplikasi yang akan dilakukan pengujian [3].

Performance test pada core banking PT. XYZ sudah pernah dilakukan sebelumnya di environment dengan spesifikasi dan arsitektur server yang berbeda dengan environment

production. Environment production merupakan versi sistem core banking yang saat ini digunakan untuk keperluan aktivitas nasabah yang ada di PT. XYZ. Sehingga PT. XYZ menyediakan environment core banking baru yang disebut dengan environment QA untuk keperluan performance testing ini dengan spesifikasi dan arsitektur server yang lebih menyerupai dengan versi production. Karena jika dilakukan performance test di environment yang sama lagi maka yang terjadi adalah hasil dari performa aplikasi tidak akan menunjukkan hasil yang mencerminkan versi production. Hal tersebut akan membuat aplikasi tersebut belum diketahui hasil dari performanya dan tidak terlacak jika ada sumber permasalahan dalam performanya. Dengan spesifikasi dan arsitektur yang menyerupai versi production, maka hasil yang akan didapatkan juga akan menggambarkan performa core banking yang saat ini sedang digunakan untuk mengelola aktivitas nasabah pada bank tersebut, karena performance test yang dilakukan sebelumnya pada core banking tersebut memiliki kondisi spesifikasi dan arsitektur yang jauh berbeda dengan versi production.

Maka dari itu penelitian ini akan melakukan performance testing dengan menggunakan metode stress testing dan load testing pada sistem core banking sebagai tools yang akan dipakai.

B. Rumusan Masalah

Berdasarkan latar belakang yang telah disampaikan maka didapatkan rumusan masalah sebagai berikut.

Bagaimana cara penerapan performance testing menggunakan metode load testing dan stress testing pada sistem core banking PT. XYZ?

Bagaimana pencapaian hasil performance testing terhadap objektif yang telah ditentukan?

C. Batasan Masalah

Adapun ruang lingkup batasan masalah pada penelitian ini yaitu sebagai berikut.

Performance testing yang akan dilakukan menggunakan metode load testing dan stress testing.

Tools yang akan digunakan untuk pengujian ini adalah Loadrunner.

Parameter yang akan dinilai dalam pengujian ini yaitu response time, transactions per second (TPS), dan error rate.

D. Tujuan

Berdasarkan rumusan masalah yang ada di atas, maka tujuan dari penelitian ini sebagai berikut.

Melakukan load testing dan stress testing pada core banking menggunakan tools Loadrunner.

Menganalisis hasil performance testing yang sudah dilakukan.

II. KAJIAN TEORI

A. Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan proses untuk mengevaluasi dan memverifikasi perangkat lunak sudah sesuai dengan yang diharapkan, mencegah adanya *bug* ataupun *error*, dan juga memastikan bebas dari *defect*. Ada banyak jenis pengujian perangkat lunak, diantaranya adalah *functional testing* dan *non-functional testing*.

1. Functional Testing

Functional testing adalah pengujian yang menguji fungsi-fungsi yang ada pada sebuah perangkat lunak.

Perangkat lunak akan diuji dengan memberikan *input* data yang kemudian akan dicek hasil *output*-nya [4].

2. Non-Functional Testing

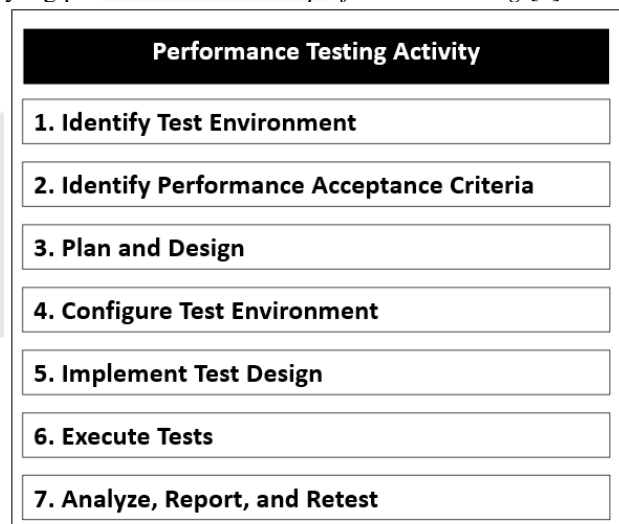
Non-functional seperti namanya, merupakan *requirement* yang tidak secara langsung terkait dengan spesifik *services* yang diberikan oleh sistem kepada penggunaannya. *Non-functional* biasanya tidak mementingkan fungsional individu perangkat lunak, tetapi menggambarkan seberapa bagus suatu perangkat lunak atau sistem terhadap *reliability* dan *response time*. Beberapa yang termasuk pengujian *non-functional* yaitu *performance*, *security*, *availability* dan lain-lainnya [5].

B. Performance Testing

Performance Test merupakan salah satu *non-functional testing* yang menentukan atau memverifikasi karakteristik dari *speed*, *scalability*, dan *stability* pada aplikasi yang sedang diuji di bawah beban kerja tertentu [4]. Berikut hal-hal yang dapat dicapai ketika dilakukan *performance testing*.

- Menilai kesiapan produksi
- Mengevaluasi terhadap kriteria performa
- Membandingkan karakteristik performa dari beberapa sistem atau konfigurasi sistem
- Menemukan sumber suatu permasalahan performa
- Support perbaikan peningkatan performa
- Menemukan level *throughput*

Jadi *performance testing* biasanya dilakukan untuk membantu mengidentifikasi *bottlenecks* dalam sebuah sistem, menjadi acuan dasar untuk pengujian di masa mendatang, mendukung perbaikan peningkatan performa, dan mengumpulkan data terkait performa untuk membantu membuat keputusan yang tepat terkait dengan kualitas keseluruhan dari aplikasi yang sedang diuji. Selain itu, hasil dari pengujian dan analisis *performace test* dapat membantu untuk memperkirakan konfigurasi *hardware* yang diperlukan untuk mendukung sebuah aplikasi yang berencana untuk “go live” ke versi *production*. Berikut beberapa proses aktivitas yang perlu dilakukan dalam *performance testing* [6].



GAMBAR Error! No text of specified style in document.-1
Aktivitas *Performance Testing* [6]

Hal pertama yang perlu dilakukan sebelum menjalankan *performance testing* yaitu mengidentifikasi lingkungan (*environment*) dari aplikasi yang akan diuji. Memahami lingkungan sistem yang akan diuji akan membantu perencanaan pengujian yang lebih efisien. Kemudian setelah

mengidentifikasi lingkungan sistem yang akan diuji, hal selanjutnya adalah menentukan kriteria-kriteria yang akan diuji untuk menjadi ketentuan penilaian yang perlu dicapai dalam pengujian ini. Kriteria yang dimaksud meliputi *response time*, *transactions per second (TPS)*, *error rate*.

Setelah selesai mengidentifikasi hal-hal di atas, maka bisa dimulai untuk membuat perencanaan dan desain pengujian (*plan and design*) dari hal-hal yang sudah diidentifikasi. Beberapa yang perlu diidentifikasi dari perencanaan tersebut adalah membuat *script*, skenario, dan menentukan data pengujian. Dari perencanaan tersebut akan diimplementasikan, dijalankan, dan dianalisis. Untuk mengimplementasikan maka perlu mulai dipersiapkan komponen-komponen untuk melakukan pengujian. Setelah setiap komponen siap, maka pengujian bisa mulai dieksekusi atau dijalankan.

Proses terakhir dalam pengujian ini adalah menganalisa hasil pengujian yang telah dieksekusi dan membuat laporan pengujian. Kemudian lakukan pengujian ulang jika diperlukan untuk memvalidasi hasil dari pengujian yang sebelumnya.

1. Stress Testing

Stress test adalah jenis dari *performance test* yang dirancang untuk mengevaluasi *behavior* aplikasi saat didorong melampaui kondisi normalnya atau kondisi puncaknya [6]. Tujuan dari *stress testing* adalah untuk menemukan *bug* aplikasi yang hanya muncul ketika beban aplikasi sedang dalam kondisi yang tinggi. *Bug* tersebut dapat mencakup hal-hal seperti masalah sinkronisasi, *race conditions*, dan kebocoran memori. *Stress test* juga memungkinkan untuk mengidentifikasi titik lemah dari sebuah aplikasi dan juga menunjukkan bagaimana aplikasi kondisi aplikasi di bawah tekanan beban yang ekstrem.

2. Load Testing

Load test merupakan salah satu jenis *performance test* di mana sistem disimulasikan untuk menangani beban yang sesuai dengan beban yang dapat ditangani oleh aplikasi yang diuji [6]. Untuk dapat menentukan beban yang sesuai salah satu caranya adalah dapat memanfaatkan hasil dari *stress test*. Dari hasil *stress test* akan dianalisa untuk mengetahui beban maksimal yang dapat ditangani aplikasi tersebut. Kemudian dari hasil tersebut dapat dipakai untuk kebutuhan *load test*.

C. Core Banking

Sistem *core banking* adalah aplikasi inti dan merupakan jantung dari sebuah perbankan [7]. *Core banking* berfungsi untuk memproses transaksi perbankan sehari-hari yang mencakup fungsi nasabah, simpanan, pinjaman, akuntansi, dan pelaporan. Aplikasi *core banking* pada PT. XYZ merupakan aplikasi berbasis *microservice*. *Microservice* merupakan salah satu contoh arsitektur perangkat lunak yang memecah aplikasi menjadi beberapa bagian kecil dan aplikasi kecil tersebut adalah *service*. Pada aplikasi *core banking* tersebut memiliki beberapa *service API* yang di mana di setiap *service* nya merupakan fungsionalitas dari aplikasi tersebut. Dari *service* yang ada tersebut akan dibagi menjadi 2 jenis fungsionalitas yaitu:

1. Transaksional

Fungsionalitas transaksional akan mewakili *service core banking* yang mengelola segala jenis transaksi yang ada di bank PT. XYZ seperti *service* untuk mengatur aktivitas penarikan uang, penyetoran yang, dan lain-lain. Beberapa *services* transaksional yang ada pada *core banking* PT. XYZ

meliputi *fund transfers*, *cash transaction*, *reversal funds transfer*, *web service SKN*, dan lain-lain.

2. Non-transaksional.

Pada fungsionalitas non-transaksional akan mewakili *service* selain yang berhubungan dengan transaksi yang ada di bank PT. XYZ seperti *service* yang mengelola informasi nasabah yang meliputi informasi saldo, portofolio, pembukaan rekening, dan lain-lain. Beberapa *services* non-transaksional yang ada pada *core banking* PT. XYZ meliputi *enquiry account*, *account details*, *customer portofolio*, *enquiry portofolio*, *open account*, dan lain-lain.

D. Loadrunner

Loadrunner adalah sebuah alat *performance test* yang dapat memperkirakan perilaku maupun properti sebuah sistem dan menemukan permasalahan melalui sejumlah simulasi pengguna yang akan melakukan hit ke lingkungan aplikasi yang akan diuji. Pada alat pengujian ini juga dapat melakukan *monitoring* untuk melihat utilisasi *server* dari aplikasi yang sedang diuji. Dengan menggunakan Loadrunner maka akan dapat mempersingkat waktu pengujian, mengoptimalkan kinerja pengujian, dan mempercepat siklus peluncuran sistem aplikasi yang telah selesai dibangun [8]. Ada 3 komponen yang ada pada Loadrunner, yaitu *Virtual Generator (VuGen)*, *Controller*, *Analysis*.

Pada Loadrunner memiliki langkah-langkah penggunaan yang dapat mendukung aktivitas *performance testing* yang sudah dijelaskan di atas. Berikut langkah-langkah pada Loadrunner.

- *Planning the test*
- *Creating the vuser script*
- *Creating the scenario*
- *Running the scenario*
- *Analyzing test results*

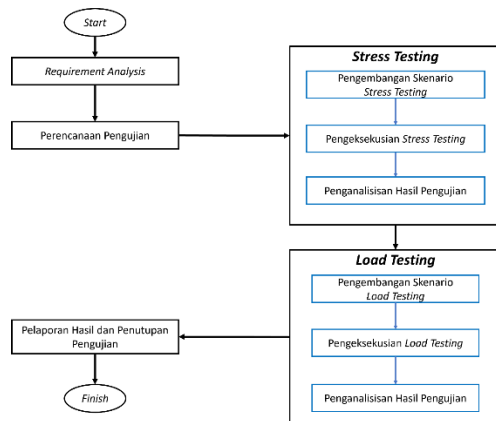
Selama bagian *planning* semua tujuan atau *objectives* pengujian perlu dipastikan dapat tercapai agar mendapatkan hasil yang lebih baik. Kemudian langkah selanjutnya adalah pembuatan *script testing*. Pembuatan *script testing* tersebut dapat dilakukan di komponen *Virtual Generator* di mana semua pengaturan *script* akan diatur di komponen tersebut. Setelah menyelesaikan pembuatan *script*, maka langkah selanjutnya adalah mempersiapkan skenario pengujian yang dapat diatur di komponen yang bernama *Controller*. Pada *Controller* dapat diatur skenario yang meliputi banyaknya concurrent user pengujian, durasi pengujian, dan lain-lain. Pada skenario yang telah dibuat tersebut akan menentukan apakah pengujian tersebut termasuk dalam metode *stress test* atau *load test*. Setelah skenario selesai dibuat maka pengujian dapat dijalankan. Selama pengujian berjalan, Loadrunner akan mengumpulkan data-data pengujian tersebut dan akan dijadikan hasil dari pengujian yang dapat dilihat pada komponen *Analysis*.

III. METODE

A. Alur Penelitian

Pada penelitian ini akan dilakukan *performance testing* pada *core banking* PT. XYZ untuk dapat diukur seberapa jauh aplikasi tersebut dapat berjalan di bawah beban kinerja yang diharapkan. Alur penelitian perlu dibuat agar aktivitas *performance testing* dapat berjalan dengan baik dan akan digunakan sebagai kerangka aktivitas penelitian ini. Pada alur

penelitian ini akan mengadaptasi *Software Testing Life Cycle* (STLC).



GAMBAR Error! No text of specified style in document.-2
Alur Penelitian

B. Requirement Analysis

Pada tahap pertama *Software Testing Life Cycle* (STLC) yaitu *requirement analysis* di mana pada tahap ini pengujian perlu mengetahui serta memahami kebutuhan bidang yang akan diuji dan persyaratan dari pengujian [9]. Tahap pertama perlu dilakukan koordinasi dengan tim *developer* maupun *user* dari perangkat lunak yang akan diuji.

1. Pengumpulan Data

Sebelum melakukan pengujian performa dalam penelitian ini diperlukan aktivitas pengumpulan data dari pihak PT. XYZ untuk menentukan perencanaan dalam pengujian ini. Data didapatkan dengan melakukan wawancara untuk memperoleh informasi langsung dari pihak PT. XYZ. Proses wawancara dilakukan dengan mengajukan pertanyaan yang dapat dilihat pada lampiran 1.

2. Identifikasi Bisnis Proses

Pada umumnya sistem *core banking* terdiri dari beberapa fungsi atau modul yang terintegrasi, antara lain yaitu modul kredit (pinjaman), modul dana (deposito), modul akuntansi (buku besar), modul pengiriman uang, dan sebagainya [10]. Fitur pada *core banking* PT. XYZ digambarkan nama *services* pada *core banking* itu sendiri. Setiap perbankan memiliki nama *service* yang berbeda di setiap *core banking*-nya, seperti *services* milik *core banking* PT. XYZ dapat dilihat pada lampiran 2. Setiap *service* pada *core banking* PT. XYZ dapat mengelola lebih dari satu fitur pada aplikasi *mobile banking* mereka.

3. Identifikasi Lingkungan Pengujian

Informasi *environment* aplikasi yang akan dijalankan diperlukan untuk menjalankan *performance test*. Jika tujuan pengujian ini adalah untuk melihat karakteristik kinerja aplikasi dalam *environment production* maka *environment* pengujian perlu dibuat untuk menyerupai versi *production*-nya. Faktor kunci dalam mengidentifikasi lingkungan pengujian adalah memahami sepenuhnya persamaan dan perbedaan antara lingkungan pengujian dan versi *production*. Berikut beberapa faktor yang perlu diperhatikan.

a. Hardware

- Konfigurasi
- Spesifikasi (CPU, RAM, dll.)

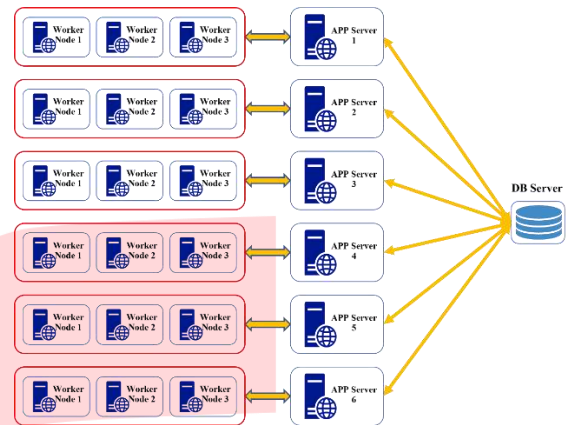
b. Software

- Tipe server yang digunakan

- Kapasitas Penyimpanan
- c. Faktor Eksternal
- Interaksi dengan sistem lain
- *Traffic* pada jaringan

4. Arsitektur Aplikasi

Berikut merupakan arsitektur dari *core banking* dari PT. XYZ berdasarkan data yang didapatkan.



GAMBAR Error! No text of specified style in document.-3
Arsitektur Aplikasi yang Akan Diuji

5. Spesifikasi Server Aplikasi

TABEL Error! No text of specified style in document.-1
Spesifikasi Server Aplikasi

No	Server	IP Address	Konfigurasi Hardware			OS (+version)
			CPU (Core)	Memory (GB)	Disk (GB)	
1	APP Server 1	10.0.19.153	80	1.536	556	Windows Server 2012 R2
2	APP Server 2	10.0.19.154	80	1.536	556	Windows Server 2012 R2
3	APP Server 3	10.0.19.156	44	512	556	Windows Server 2012 R2
4	APP Server 4	10.0.19.159	24	120	698	Windows Server 2012 R2
5	APP Server 5	10.0.19.160	24	120	698	Windows Server 2012 R2
6	APP Server 7	10.0.19.161	24	120	698	Windows Server 2012 R2

6. Identifikasi Acceptance Criteria

Untuk mengetahui performa dari aplikasi *core banking* ini dapat dengan mengamati perilaku aplikasi di bawah beban

pengujian. Berikut merupakan parameter objektif yang ingin dicapai dalam *performance test* kali ini.

TABEL Error! No text of specified style in document.-2

Parameter Objektif Pengujian

No.	Parameter
1.	Error rate
2.	Transactions Per Second (TPS)
3.	Response time

7. Pengembangan Script Pengujian

Berikut merupakan bisnis proses/ fitur yang diidentifikasi untuk *performance testing* berdasarkan pembagian 2 jenis fungsionalitas, transaksional dan non-transaksional. Proses bisnis ini telah mengambil bagian dalam skenario *performance test*.

TABEL Error! No text of specified style in document.-3

Bisnis Proses Core Banking

ID Fitur	Fitur	Deskripsi	Jenis Fungsionalitas
01	Enquiry Account	Pada fitur ini berfungsi untuk menampilkan informasi nasabah yang di antaranya adalah nama pemilik <i>account</i> , nomor <i>customer</i> , nomor rekening nasabah, kantor cabang <i>account</i> , isi saldo rekening, dan lain-lain.	Non-Transaksional
02	Fund Transfer	Pada fitur ini digunakan untuk mengatur transaksi transfer saldo nasabah sesama bank.	Transaksional

Dari masalah yang ada pada PT. XYZ yang sedang mengembangkan dan mempersiapkan peluncuran aplikasi *mobile banking* mereka, maka akan diambil kedua *services* pada *core banking* tersebut yaitu *fund transfer* dan *enquiry account*. Berdasarkan data transaksi harian pada *mobile banking* milik PT. XYZ yang dapat dilihat pada lampiran 3, transfer pinbuk (transfer antar bank PT. XYZ) merupakan transaksi terbanyak yang dilakukan oleh nasabah PT. XYZ. *Fund transfers* merupakan *service core banking* untuk mengatur transfer transfer antar bank PT. XYZ. Sedangkan pada fitur non-transaksional diambil *enquiry account* sebagai *service* yang mengatur beberapa fitur seperti cek saldo, nomor *customer* bank, dan fitur-fitur non-transaksional lainnya. Sehingga diambil 2 *service* atau fitur tersebut untuk dilakukan *performance testing*.

Untuk melakukan pengujian perlu dibuat *script* masing-masing dari 2 fitur di atas. Pengembangan *script* akan dilakukan di salah satu komponen yang dimiliki Loadrunner yaitu komponen VuGen (*Virtual Generator*). Untuk membuat *script* tersebut dibutuhkan *API Request* dari 2 fitur di atas yang dapat di lihat pada lampiran 3.

C. Perencanaan Pengujian

Bagian ini akan berisi desain skenario *stress test* dan *load test* yang akan digunakan untuk *performance testing core banking* pada PT. XYZ untuk menjalankan semua proses bisnis.

1. Desain Skenario Stress Test



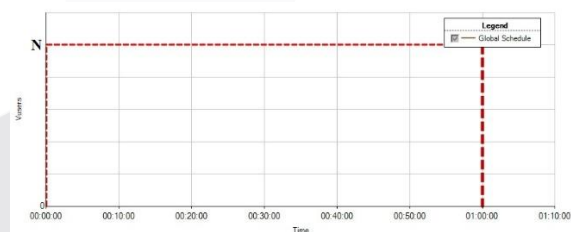
GAMBAR Error! No text of specified style in document.-4

Skenario Stress Test

- 2 Proses Bisnis/ Fitur akan digunakan untuk *stress test*.
- Meningkat N *virtual users* setiap 5 menit sampai NMax *virtual users*, lalu jalankan tes selama 5 menit. Kemudian hentikan semua *virtual users* secara bersamaan. Nmax = jumlah maksimal *virtual users*. N = jumlah kenaikan *virtual users* secara konstan.
- Hasil akan dikumpulkan pada setiap peningkatan N *virtual users* sampai NMax.
- Distribusi beban *virtual user* yang digunakan untuk semua skenario adalah sebagai berikut:

Fitur ID	Fitur	Distribusi beban (%)
01	Enquiry Account	50 %
02	Fund Transfer	50 %

2. Desain Skenario Load Test



Gambar Error! No text of specified style in document.-5

Skenario Load Test

- 2 Proses Bisnis/ Fitur akan digunakan untuk *load test*.
- Mulai N *virtual users* secara bersamaan lalu jalankan selama 2 jam, lalu hentikan semua vuser secara bersamaan. N = jumlah beban *virtual users* maksimal yang dapat ditangani oleh aplikasi.
- Hasil akan dikumpulkan dari awal sampai akhir pengujian.
- Distribusi beban *virtual user* yang digunakan untuk semua skenario adalah sebagai berikut:

Fitur ID	Fitur	Distribusi beban (%)
01	Enquiry Account	50 %
02	Fund Transfer	50

IV. HASIL DAN PEMBAHASAN

A. Objektif Pengujian

Untuk menentukan apakah sistem atau aplikasi yang akan diuji dapat dikatakan berjalan baik maka perlu diidentifikasi objektif nya dari sisi bisnis. Pada *core banking* PT. XZY didapatkan objektif yang sudah ditentukan sebagai berikut.

TABEL Error! No text of specified style in document.-4

Objektif Performance Test

No.	Parameter	Objektif
1.	Error rate	$\leq 1\%$
2.	Transactions Per Second (TPS)	> 1000 TPS
3.	Response time	$< 2,5$ detik

B. Implementasi Desain Pengujian

1. Script Pengujian Menggunakan Loadrunner

Terdapat 2 Fitur yang akan dibuat *script* untuk *performance testing* menggunakan komponen *Loadrunner* yang bernama VuGen (*Virtual Generator*). Fitur tersebut adalah *Enquiry Account* dan *Fund Transfer* didapatkan *API Request* dari masing-masing untuk diproses menjadi *script* pengujian. Berikut merupakan gambaran *script* yang sudah dibuat dari masing-masing fitur

a. BP01_EnquiryAccount

```

BP01_EnquiryAccount : Action.c
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
25
```

Load User	Fitur	Response Time (detik)			Jumlah Hit			Error Rate	TPS
		Min	Avg	Max	Pas sed	Fai led	Tot al		
	Fund Transfe rs	0,26	0,53	5,14	57.408	0	57.408	0,00%	
TOTAL					141.391	0	141.391	0,00%	
400	Enquiry Account	0,11	0,45	3,57	84.705	0	84.705	0,00%	506
	Fund Transfe rs	0,28	0,64	7,67	66.715	0	66.715	0,00%	
TOTAL					151.420	0	151.420	0,00%	
500	Enquiry Account	0,11	0,64	1,61	83.904	0	83.904	0,00%	525
	Fund Transfe rs	0,28	0,77	6,44	72.931	0	72.931	0,00%	
TOTAL					156.835	0	156.835	0,00%	
600	Enquiry Account	0,10	0,81	3,01	84.454	0	84.454	0,00%	533
	Fund Transfe rs	0,29	0,95	8,17	74.881	0	74.881	0,00%	
TOTAL					159.335	0	159.335	0,00%	
700	Enquiry Account	0,11	1,00	2,76	83.824	0	83.824	0,00%	537
	Fund Transfe rs	0,29	1,11	9,60	76.862	0	76.862	0,00%	
TOTAL					160.686	0	160.686	0,00%	
800	Enquiry Account	0,11	1,18	4,11	83.748	0	83.748	0,00%	542
	Fund Transfe rs	0,29	1,27	10,37	78.317	0	78.317	0,00%	
TOTAL					162.065	0	162.065	0,00%	
900	Enquiry Account	0,11	1,36	4,18	83.547	0	83.547	0,00%	544
	Fund Transfe rs	0,30	1,45	11,78	79.042	0	79.042	0,00%	
TOTAL					162.589	0	162.589	0,00%	

Load User	Fitur	Response Time (detik)			Jumlah Hit			Error Rate	TPS
		Min	Avg	Max	Pas sed	Fai led	Tot al		
1000	Enquiry Account	0,11	1,53	4,32	83.713	0	83.713	0,00%	545
	Fund Transfe rs	0,31	1,63	12,94	79.309	0	79.309	0,00%	
TOTAL					163.022	0	163.022	0,00%	
1100	Enquiry Account	0,11	1,72	4,98	83.241	0	83.241	0,00%	544
	Fund Transfe rs	0,35	1,81	13,99	79.509	0	79.509	0,00%	
TOTAL					162.750	0	162.750	0,00%	
1200	Enquiry Account	0,11	1,91	5,48	83.078	0	83.078	0,00%	538
	Fund Transfe rs	0,36	2,05	21,93	77.699	0	77.699	0,00%	
TOTAL					160.777	0	160.777	0,00%	
1300	Enquiry Account	0,11	2,09	6,18	82.930	0	82.930	0,00%	542
	Fund Transfe rs	0,38	2,20	24,25	79.175	0	79.175	0,00%	
TOTAL					162.105	0	162.105	0,00%	
1400	Enquiry Account	0,11	2,27	6,55	82.940	0	82.940	0,00%	546
	Fund Transfe rs	0,60	2,36	17,82	80.215	0	80.215	0,00%	
TOTAL					163.155	0	163.155	0,00%	
1500	Enquiry Account	0,11	2,45	7,09	83.010	0	83.010	0,00%	546
	Fund Transfe rs	0,65	2,54	18,77	80.340	0	80.340	0,00%	
TOTAL					163.350	0	163.350	0,00%	
1640	Enquiry Account	0,15	2,62	7,59	83.212	0	83.212	0,00%	548
	Fund Transfe rs	0,67	2,72	20,07	80.547	0	80.547	0,00%	

Load User	Fitur	Response Time (detik)			Jumlah Hit			Error Rate	TPS
		Min	Avg	Max	Pas sed	Fai led	Tot al		
TOTAL					163.759	0	163.759	0,00%	
1700	Enquiry Account	0,30	2,80	7,82	83.141	0	83.141	0,00%	547
	Fund Transfers	0,70	2,90	21,27	80.446	0	80.446	0,00%	
TOTAL					163.587	0	163.587	0,00%	
1800	Enquiry Account	0,41	3,02	8,20	82.354	0	82.354	0,00%	545
	Fund Transfers	0,87	3,09	22,42	80.542	0	80.542	0,00%	
TOTAL					162.896	0	162.896	0,00%	
1900	Enquiry Account	0,70	3,19	8,25	82.525	0	82.525	0,00%	547
	Fund Transfers	0,91	3,26	25,13	80.891	0	80.891	0,00%	
TOTAL					163.416	0	163.416	0,00%	
2000	Enquiry Account	0,58	3,37	8,39	82.649	0	82.649	0,00%	547
	Fund Transfers	0,97	3,44	25,17	80.930	0	80.930	0,00%	
TOTAL					163.579	0	163.579	0,00%	
2100	Enquiry Account	1,04	3,54	8,66	82.754	0	82.754	0,00%	548
	Fund Transfers	0,99	3,62	26,66	81.103	0	81.103	0,00%	
TOTAL					163.857	0	163.857	0,00%	
2200	Enquiry Account	1,22	3,73	8,79	82.566	0	82.566	0,00%	547
	Fund Transfers	1,05	3,80	27,57	81.082	0	81.082	0,00%	
TOTAL					163.648	0	163.648	0,00%	
2300	Enquiry Account	1,39	3,89	8,92	82.876	0	82.876	0,00%	519

Load User	Fitur	Response Time (detik)			Jumlah Hit			Error Rate	TPS
		Min	Avg	Max	Pas sed	Fai led	Tot al		
	Fund Transfers	1,18	4,18	42,68	71.897	0	71.897	0,00%	
TOTAL					154.773	0	154.773	0,00%	
2400	Enquiry Account	1,60	4,10	99,84	82.075	16	82.091	0,02%	280
	Fund Transfers	2,94	6,88	26,82	17.18	191.642	193.360	99,11%	
TOTAL					83.793	191.658	275.451	69,58%	
2500	Enquiry Account	1,81	4,33	9,30	81.612	0	81.612	0,00%	273
	Fund Transfers	0,00	0,00	0,00	0	287.976	287.976	100,00%	
TOTAL					81.612	287.976	369.588	77,92%	

D. Analisis Stress Test

Dari hasil *stress test* yang telah dilakukan, ditemukan *error rate* yang mulai muncul ketika di *hit* lebih dari 2.300 *user*. Pada 2.400 *user* didapatkan *error rate* sebanyak 69,58%. Berikut *error* yang berhasil ditangkap oleh Loadrunner pada saat mencapai 2.400 *user*.

TABEL Error! No text of specified style in document.-6
Error yang Ditemukan Selama Stress Test

No.	Error
1.	Failed to connect to server "10.0.19.156:8180" [10061] Connection refused
2.	Failed to connect to server "10.0.19.156:8280" [10061] Connection refused
3.	Failed to connect to server "10.0.19.156:8380" [10061] Connection refused
4.	Failed to connect to server "10.0.19.159:8180" [10061] Connection refused
5..	Failed to connect to server "10.0.19.159:8280" [10061] Connection refused
6.	Failed to connect to server "10.0.19.159:8380" [10061] Connection refused
7.	Failed to connect to server "10.0.19.160:8180" [10061] Connection refused
8.	Failed to connect to server "10.0.19.160:8280" [10061] Connection refused
9.	Failed to connect to server "10.0.19.160:8380" [10061] Connection refused
10.	Failed to connect to server "10.0.19.161:8180" [10060] Connection timed out
11.	Failed to connect to server "10.0.19.161:8180" [10061] Connection refused
12.	Failed to connect to server "10.0.19.161:8280" [10061] Connection refused
13.	Failed to connect to server "10.0.19.161:8380" [10060] Connection timed out

14.	Failed to connect to server "10.0.19.161:8380" [10061] Connection refused
-----	--

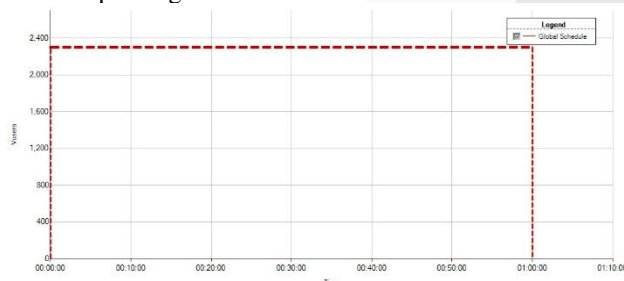
Dari *error* yang di atas tersebut, terlihat untuk *error* ketika *hit* pada server beralamatkan ip 10.0.19.156, 10.0.19.159, 10.0.19.160, 10.0.19.161. Mayoritas dari *error* tersebut terkait *connection refused*, yaitu di mana *request* yang dikirim oleh user ditolak oleh server sehingga *user* tidak menerima *response* dari *request* yang dikirimnya. *Error connection refused* tersebut tidak dapat disimpulkan begitu saja penyebabnya adalah karena koneksi internet dari *user*. Penyebab lain dari *error* tersebut adalah jumlah *request* yang kirim oleh *users* sudah mencapai maksimum yang dapat diterima oleh server, sehingga server menolak *request* yang dikirimkan oleh *user*. Dapat dilihat juga dari tabel 3-1 spesifikasi dari server yang mengalami *error* berbeda atau lebih kecil jika dibandingkan dengan server yang tidak mengalami *error*, di mana hal tersebut bisa menjadi salah satu indikasi yaitu untuk perlu dilakukan peningkatan spesifikasi pada server yang mengalami *error* dengan menyamakan dengan spesifikasi server yang tidak mengalami *error*.

Setelah dilakukannya *stress testing* juga diharapkan dapat mengidentifikasi perilaku aplikasi yang diuji ketika didorong pada kondisi puncak atau kondisi di atas normalnya. Dari hasil *stress test* yang telah dijabarkan, *core banking* PT. XYZ mengalami kondisi *error* di mana server dari *core banking* tersebut *down* sehingga *request* yang berikan oleh *user* banyak yang tidak dapat ditangani oleh server tersebut lagi. Untuk menghindari hal tersebut, perlu dilakukannya upgrade spesifikasi server ataupun juga bisa menambahkan jumlah server pada *core banking* PT. XYZ sehingga ketika terjadi kenaikan jumlah *concurrent user* di atas normalnya maka *core banking* ini masih dapat mengolah *request* yang diberikan oleh *user* tanpa mengalami *down*. Untuk dapat mengetahui ukuran upgrade spesifikasi ataupun penambahan server, diperlukan penelusuran lebih lanjut.

E. Load Testing

1. Skenario Pengujian

Pada hasil *stress test* yang sudah dilakukan, didapatkan bahwa titik kuat yang dapat ditangani oleh sistem *core banking* adalah di 2.300 user. Dari hasil tersebut akan diambil sebagai acuan untuk dilakukan *load test* sebanyak 2.300 *concurrent user* selama 1 jam untuk menguji kestabilan dari beban yang dapat ditangani oleh sistem ini. Maka dapat digambarkan skenario dari *load test* pada *core banking* ini. Berikut merupakan grafik skenario *load test*.



GAMBAR Error! No text of specified style in document.-9
Skenario Load Test 2300 User

2. Hasil Rangkuman Pengujian

Eksekusi *load test* pada *core banking* PT. XYZ berhasil dilakukan dengan skenario 2.300 *concurrent user* selama 1 jam. Berikut merupakan hasil *load testing* setelah dilakukan eksekusi.

TABEL Error! No text of specified style in document.-7
Hasil Load Testing 2300 Concurrent User

Load User	Fitur	Response Time (detik)			Jumlah Hit			Error Rate	TPS
		Min	Avg	Max	Passed	Failed	Total		
2300	Enquiry Account	0,08	1,17	35,47	2,917,888	4,603	2,922,491	0,16%	1.264
	Fund Transfers	0,36	2,29	86,90	1,629,384	75	1,629,459	0,00%	
TOTAL					4,547,272	4,678	4,551,950	0,10%	

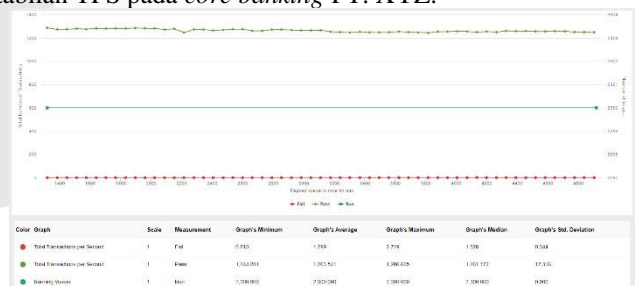
F. Analisis Load Test

Dari hasil *load testing* yang sudah dilakukan, didapatkan hasil yang memenuhi semua objektif yang ada. Adapun objektif yang telah didefinisikan pada tabel 4-1 yang meliputi sebagai berikut.

TABEL Error! No text of specified style in document.-8
Ringkasan Hasil Objektif

No.	Parameter	Objektif	Hasil
1.	Error rate	<= 1%	0,10%
2.	Transactions Per Second (TPS)	>= 1000 TPS	1,264 TPS
3.	Response time	<= 2,5 detik	Enquiry Account = 1,17 detik Fund Transfers = 2,29 detik

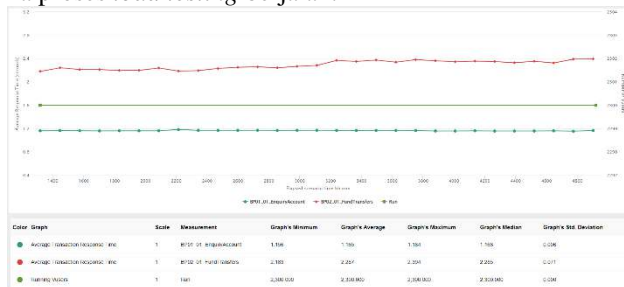
Dari gambar di bawah ini didapatkan *Transactions Per Second* (TPS) sebanyak 1.264 TPS di mana hasil tersebut sudah melampaui ekspektasi pada *core banking* tersebut. Grafik yang ditunjukkan pada gambar 4-5 juga menunjukkan kestabilan TPS pada *core banking* PT. XYZ.



GAMBAR Error! No text of specified style in document.-10
TPS Load Testing

Kemudian untuk *response time* dari kedua fitur juga mendapatkan rata-rata nilai di bawah yang diharapkan. Pada *Enquiry Account* mendapatkan rata-rata *response time* 1,17 detik, sedangkan *Fund Transfers* mendapatkan rata-rata *response time* yang sedikit lebih tinggi yaitu 2,29 detik. Dari hal tersebut bisa diartikan bahwa untuk fitur transaksional memiliki proses yang lebih berat dibandingkan dengan non-transaksional sehingga server memerlukan waktu lebih lama dalam memproses fitur transaksional dibandingkan dengan

non-transaksional. Berikut grafik *response time* pada kedua fitur yang dapat menunjukkan kestabilan untuk keduanya selama proses *load testing* berjalan.



GAMBAR Error! No text of specified style in document.-11
Rata-Rata *Response Time* Load Testing

Dalam eksekusi *load testing* ini, ditemukan total *request error* sebanyak 4.678 dari total *request hit* ke *core banking* sebanyak 4.551.950. Meskipun begitu, *error rate* tersebut masih memenuhi ekspektasi yang ditentukan dibawah 1% yaitu 0,10%. Berikut daftar *error* yang ditemukan selama *load testing* berjalan.

TABEL Error! No text of specified style in document.-9
Error yang Ditemukan Selama Load Testing

No.	Error
1.	Failed to connect to server "10.0.19.153:8180" [10061] Connection refused
2.	Failed to connect to server "10.0.19.153:8280" [10061] Connection refused
3.	Failed to connect to server "10.0.19.153:8380" [10061] Connection refused
4.	Failed to connect to server "10.0.19.154:8180" [10061] Connection refused
5..	Failed to connect to server "10.0.19.154:8280" [10061] Connection refused
6.	Failed to connect to server "10.0.19.154:8380" [10061] Connection refused
7.	Failed to connect to server "10.0.19.161:8280" [10060] Connection timed out

V. KESIMPULAN

A. Kesimpulan

Kesimpulan yang dapat diambil berdasarkan penelitian yang sudah dilakukan ini adalah:

1. Performance test pada core banking PT. XYZ dieksekusi dengan menjalankan metode stress testing terlebih dahulu sehingga mendapatkan hasil maksimum concurrent users. Kemudian dari hasil tersebut akan menjadi acuan dalam menjalankan load testing sesuai dengan kondisi normalnya.
2. Performance test telah dilakukan pada core banking PT. XYZ dan dengan menggunakan metode load testing dapat dipenuhi aspek pengujian speed dan stability di mana aspek speed pada core banking diukur dari hasil rata-rata response time 1,17 detik untuk non-transaksional dan 2,29 detik untuk transaksional, dan stability pada hasil grafik yang didapatkan tidak ada terjadi kenaikan atau penurunan signifikan sehingga menunjukkan kestabilan pada aplikasi.
3. Sedangkan menggunakan metode stress testing dapat memenuhi aspek pengujian scalability, di mana ketika diuji dengan menaikkan jumlah concurrent user perlahan pada durasi tertentu ditemukan bahwa maksimum beban yang dapat diterima oleh core banking berikut adalah 2.300 concurrent user. Selain didapatkan nilai maksimum beban, pada stress test juga ditemukan behavior dari core banking PT. XYZ yang mengalami server down atau tidak dapat lagi

menangani request yang dikirim oleh user ketika didorong pada kondisi melebihi normalnya sehingga perlu dilakukan upgrade spesifikasi atau penambahan server untuk mencegah adanya server down ketika terdapat kenaikan concurrent user ekstrem pada environment production.

4. 2.300 concurrent user hasil dari stress test tersebut diambil dan dijadikan patokan untuk menjalankan load testing. Hasil yang didapatkan setelah melakukan load testing menunjukkan bahwa semua objektif sudah tercapai sehingga dapat dikatakan bahwa sistem core banking tersebut telah stabil di 1.264 TPS, dengan rata-rata response time pada masing-masing fitur berada dibawah 2,5 detik yaitu transaksional di angka 2,29 detik dan non-transaksional berada di angka 1,17 detik, dan memiliki error rate dibawah 1% yaitu sebesar 0,10%.

5. Hasil rata-rata response time fitur transaksional di angka 2,29 detik, sedangkan untuk non-transaksional berada di angka 1,17 detik. Hasil tersebut menunjukkan proses fitur transaksional lebih lama dibandingkan dengan non-transaksional.

5.2 Saran

Berikut beberapa saran yang dapat dilakukan pada penelitian selanjutnya :

1. Pada penelitian performance test selanjutnya diharapkan dapat menggunakan metode yang lain dengan harapan ditemukan kondisi bottleneck pada saat dilakukan pengujian.
2. Melakukan monitoring pada semua server sistem yang dilakukan pengujian agar dapat melihat hasil utilisasi pada server-server tersebut ketika dilakukan pengujian.

REFERENSI

- [1] Andarwati M. (2016). Analisis Faktor Yang Mempengaruhi Kesuksesan Penggunaan Core Banking System (CBS) Dengan Menggunakan Model Delone Dan Mclean. Jurnal Keuangan dan Perbankan, 20(3), 458-467.
- [2] Permatasari D I, Ardani M, & Ma'ulfa A Y. (2020). Pengujian Aplikasi Menggunakan Metode Load Testing dengan Apache Jmeter pada Sistem Informasi Pertanian. JUSTIN (Jurnal Sistem dan Teknologi Informasi), 8(1), 135-139.
- [3] Nandal V, Solanki K, & Dhankhar A. (2018). Performance Testing on Web-based Application using LoadRunner. Jetir, 5(9) 37-42.
- [4] Gantini T, Djajalaksana Y M, Yefta S K. (2018). Pengujian Perangkat Lunak itworkforceindoensia.org. Jurnal Teknik Informatika dan Sistem Informasi, 4(3) 355-363.
- [5] Sianturi R A, Sinaga A M, Pratama Y. (2021). Perancangan Pengujian Fungsional Dan Non Fungsional Aplikasi Siappara Di Kabupaten Humbang Hasundutan. J-ICON, 9(2), 133-141.
- [6] Meier J D, Farre C, Bansode P, Barber S, & Rea D. (2007). Performance Testing Guidance for Web Applications. Microsoft Corporation.
- [7] Ari D P S. (2013). Pengaruh Technology Acceptance Model Dan Pengembangannya Dalam Perilaku Menggunakan Core Banking System. Jurnal Keuangan dan Perbankan, 17(2), 267-278.

- [8] Hui-li Z, Shu Z, Xiao-jie L, Pei Z, & Shao-bo L. (2012). Research of Load Testing and Result Application Based on LoadRunner. CITCS.
- [9] Diastama I G N P D D, Sukarsa I M, Wirdiani N K A. (2021). Pengembangan Test Script untuk Load Testing Web dengan metode Software Testing Life Cycle. JITTER. 2(1).
- [10] Ambodo B S, Suryanto R, & Sofyani H. (2017). Testing of Technology Acceptance Model on Core Banking System: A Perspective on Mandatory Use. Jurnal Dinamika Akuntansi, 9(1), 11-22.
- [11] Jamil M A, Arif M, Abubakar N S A. (2017). Software Testing Techniques: A Literature Review. Proceedings - 6th International Conference on Information and Communication Technology for the Muslim World. 177-182.
- [12] Sarojadevi H. (2011). Performance Testing: Methodologies and Tools. Journal of Information Engineering and Applications, 1(5), 5-13.
- [13] Fageria P & Kaushik M. (2014). Research of Load Testing and Result Based on Loadrunner. SSRG International Journal of Civil Engineering, 1(2), 1-4.
- [14] Chou C Y, Fang Y B, Wang S T, & Kuo F A. (2020). Smoke and Stress Tests for Travel Service Applications via LoadRunner. LNISA, 11894, 366-373.
- [15] Khan R & Amjad M. (2016). Web Application's Performance Testing Using HP LoadRunner and CA Wily Introscope Tools. ICCA2016, 802-806.
- [16] Ramakrishnan R, Shrawan V, & Singh P. (2017). Setting Realistic Think Times in Performance Testing - A Practitioner's Approach. ISEC '17: Proceedings of the 10th Innovations in Software Engineering Conference, 157-164. <https://dl.acm.org/doi/10.1145/3021460.3021479>
- [17] Delta E N & Asmunin. (2016). Performance Test Dan Stress Website Menggunakan Open Source Tools. Jurnal Manajemen Informatika, 6(1), 208-215.
- [18] Fansha D A, Setyawan M Y H, & Fauzan M N. (2021). Load Test pada Microservice yang menerapkan CQRS dan Event Sourcing. Jurnal Buana Informatika, 12(2), 126-134.
- [19] Pratama M A C, Widowati S, & Junaedi D. (2017). Automasi Performance Load Testing Aplikasi Berbasis Web (Studi Kasus: Sistem Informasi Production Enterprise PT. Bio Farma). Retrieved from <https://openlibrary.telkomuniversity.ac.id/home/catalog/id/138996/slug/automasi-performance-load-testing-aplikasi-berbasis-web-studi-kasus-sistem-informasi-production-enterprise-pt-bio-farma-.html>
- Andriansyah D. (2019). Performance Dan Stress Testing Dalam Mengoptimasi Website. CBIS Journal, 07(1), 23-28.