Pengembangan Test Script untuk Load Testing Web dengan metode Software Testing Life Cycle

I Gusti Ngurah Putu Devtian Dicky Diastama[1], I Made Sukarsa[2], Ni Kadek Ayu Wirdiani[3]
Program Studi Teknologi Informasi, Fakultas Teknik, Universitas Udayana Bukit Jimbaran, Bali, Indonesia Telp. (0361) 701806
e-mail: [1]devtian_dicky@student.unud.ac.id, [2]sukarsa@unud.ac.id, [3]ayuwirdiani@unud.ac.id

Abstrak

Web Performance adalah variabel yang mengindikasikan kemampuan performa suatu website dalam menerima, memproses dan merespons request dari pengguna, dengan performance testing pengguna dapat mengetahui performa dari suatu website yang dapat digunakan untuk meningkatkan performa atau optimalisasi sebuah website. Permasalahan yang sering teradi adalah kurangnya pemahaman oleh developer tentang performa dari sebuah website, sehingga website memiliki performa atau optimalisasi rendah yang berdampak pada pengalaman pengguna. Performacne testing dapat digunakan untuk memahami performa dari suatu website, namun kesalahan dalam pembuatan script dapat berdampak pada informasi yang salah pada hasil dari pengujian performance testing. Untuk mencegah kesalahan dalam pengujian, pengembangan test script perlu mengikuti alur dari Software Testing Life Cycle, dengan mengikuti Software Testing Life Cycle developer dapat mengetahui requirement sehingga meminimalisir kesalahan pada test script. Requirement dari pengujian performa suatu website memerlukan hasil pengujian dari operasi CRUD yang akan di gunakan oleh penguna. Dengan mengetahui performa CRUD, developer dapat mengembangkan optimalisasi pada frontend dan backend.

Kata kunci: *Performa Web, Analisa Performa, Load Test, JMeter, Cloud.*

Abstract

Web Performance is a variable that shows the performance ability of a website in receiving, processing, and responding to requests from users, with performance testing users can find out the performance of a website which can be used to improve the performance or optimization of a website. The problem that often occurs is a lack of understanding by developers about the performance of a website, so that the website has low performance or optimization which has an impact on the user experience. Performance testing can be utilized to gain knowledge of a website performance, but errors in scripting can have an impact on incorrect information on the results of performance testing. To prevent errors in testing, test script development needs to follow the flow of the Software Testing Life Cycle, by following the Software Testing Life Cycle, developers can find out the requirements to minimize errors in the test script. The requirements for testing the performance of a website require the test results of the CRUD operation that the user will use. By knowing the performance of the CRUD, the developer can develop optimizations on the frontend and backend.

Keywords: *Web Performance, Performance Analysis, Load Test, JMeter, Cloud.*

1. Introduction

Performance testing adalah tipe dari software testing yang menguji performa sebuah software dan memantau behaviour sebuah software ketika di berikan beban tertentu, performance testing dapat di lakukan dengan menggunakan metode load testing, stress testing, capability testing, spike testing, dan endurance testing [1]. Kecepatan akses situs web yang lebih cepat telah terbukti meningkatkan retensi dan loyalitas pengunjung dan kepuasan pengguna, terutama bagi pengguna dengan koneksi dan menggunakan perangkat mobile.

Situs web modern dan aplikasi berbasis web sering kali lambat dalam memuat dan berinteraksi, terutama pada perangkat seluler dan jaringan seluler. Kinerja yang kurang optimal

ini mengganggu pengguna, bahkan sedikit peningkatan waktu buka halaman dapat memiliki dampak yang konsisten dan terukur pada retensi pengguna [2]. Masalah pada performa web adalah ukuran dari halaman web yang di tampilkan dan kompleksitas, dan optimalisasi dari web tersebut [3].

Pengujian performa dari suatu software ataupun website dapat dilakukan dengan menggunakan tools JMeter, namun pengembangan test script yang salah dapat mengakibatkan informasi yang di dapatkan dari hasil pengujian tidak akurat, untuk mencegah hasil dari pengujian yang tidak akurat proses pengujian dilakukan dengan metode Software Testing Life Cycle (STLC). Diagram STLC yang mencakup lima langkah. Pertama, Requirement Analysis, di mana dilakukan analisis tentang seberapa banyak web sumber daya yang akan digunakan dalam keadaan normal. Test Planning dimana perencanaan tentang pendekatan apa yang digunakan pada pengujian, Test Case Development adalah pembuatan script pengujian yang akan digunakan JMeter untuk melakukan pengujian. Environment Setup adalah langkah di mana server yang akan diterapkan untuk pengujian disiapkan. Test Execution di mana web dan database diuji di bawah beban dengan skrip yang dibuat sebelumnya dan Test Reporting adalah hasil dari analisis pengujian dan dilaporkan [4], [5] Hasil dari pengembangan test script yang di kembangkan dengan mengikuti alur STLC menghasilkan test script yang akan mengambil/mendapatkan informasi yang kita perlukan untuk optimalisasi web yang di ujikan.

2. Research Method / Proposed Method

Pengembangan Test Script pada pengujian performa dengan JMeter dilakukan dengan menggunakan metode STLC atau Software Testing Life Cycle.

Gambar 1 Diagram Software Testing Life Cycle (STLC)

adalah diagram STLC yang meliputi 5 langkah yaitu yang pertama Requirement Analysis, yang kedua Test Planning, ketiga Test Case Development, keempat Environment Setup, kelima Test Execution, dan yang terakhir Test Reporting. Berikut merupakan penjelasan dari masing-masing langkah metode Software Testing Life Cycle (STLC) [4].

2.1 Requirement Analysis

Analisa kebutuhan adalah langkah pertama dari Software Testing Life Cycle (STLC). Pada tahap ini penguji harus mengetahui kebutuhan dalam bidang apa yang akan di uji dan mengetahui persyaratan dari pengujian. Jika terjadi konflik, tidak ditemukan atau tidak diketahuinya kebutuhan, penguji dapat menanyakan variabel tersebut ke System Architecture, Programmer, dan Project Manajer, untuk mengetahui lebih detail persyaratan dari pengujian [5].

Tahap pertama yang melibatkan STLC dapat mencegah adanya test yang cacat pada saat pengujian. Kebutuhan dapat berupa fungsional atau non-fungsional, seperti performa yang dibahas pada penelitian ini.

2.2 Test Planning

Perencanaan test adalah tahap paling penting pada software testing life cycle dimana strategi pengujian di rencanakan. Tahap ini juga disebut Test Strategy phase. Pada tahap ini perencanaan estimasi upaya dan biaya dari penelitian. Tahap ini akan dimulai ketika tahap

pengumpulan kebutuhan telah selesai, dan kemudian dimulainya persiapan test plan. Setelah tahap perencanaan selesai penelitian dapat mengembangkan test cases.

2.3 Test Case Development

Pengembangan kasus tes dimulai saat proses perencanaan selesai. Ini merupakan tahapan dari STLC dimana skrip pengujian dikembangkan, bersama dengan tes skrip data yang dibutuhkan saat pengujian juga diisapkan.

2.4 Environment Setup

Environment Setup adalah langkah vital dari STLC. Test Environment adalah kondisi dari pengujian yang dilakukan pada software. Aktivitas ini bersifat independen dan dapat dijalankan secara bersamaan dengan Test Case Development. Proses mempersiapkan Environment berdasarkan permintaan dari pengembang sofware atau klien.

2.5 Test Execution

Saat persiapan Test Case Development dan Environment setup selesai berikutnya proses pengujian dapat dilakukan. Pada tahap ini pengujian dari software dalam kasus ini web dan database dapat dilakukan mengikuti tes plan yang telah dibuat pada bagian Test Case Development.

2.6 Test Reporting

Test Reporting merupakan bagian terakhir dari metode Software Testing Life Cycle (STLC). Diskusi dari hasil pengujian dan area yang dapat ditingkatkan dari STLC sebagai masukan untuk test berikutnya yang akan membantu untuk meningkatkan performa sistem, Test case dan bug report akan di analisa untuk menemukan cacat pada sistem.

3. Literature Study

Kajian pustaka memuat materi referensi dalam penelitian yang dibuat. Referensi penelitian Pengembangan Test Script untuk Load Testing Web merupakan landasan teori yang digunakan dalam membangun penelitian ini. Referensi penelitian Pengembangan Test Script untuk Load Testing Web yang dimuat terkait dengan Performance testing, dan JMeter,

3.1 Performance Test

Tujuan dari pengujian performa adalah untuk memvalidasi "kecepatan" perangkat lunak terhadap kebutuhan bisnis untuk "Kecepatan" sebagaimana didokumentasikan dalam persyaratan perangkat lunak. "Kecepatan" perangkat lunak pada umumnya didefinisikan sebagai kombinasi waktu respons dan beban kerja selama masa beban puncak [6].

Matriks performa dalam performance testting dapat dibagi menjadi 2 yaitu response time dan troughput. Response time adalah waktu yang di perlukan untuk sebuah permintaan dari client di response oleh server. Pengujian performa akan menganalisa kecepatan response dari server pada saat user dalam jumlah tertentu mengirimkan request secara bersamaan. Troughput dihitung berdasarkan request/waktu. Throughput menampilkan jumlah request yang ditampilkan pada rentang waktu tertetu selama jalannya pengujian yang befungsi menampilkan jumlah beban pada server.

3.2 JMeter

Apache JMeter adalah Apache project yang dikembangkan dengan bahasa Java dan dapat dijalankan pada semua environment. JMeter dapat digunakan untuk load testing, menganalisis dan mengukur performa dari suatu software [1], [7].

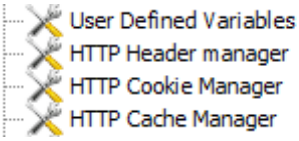
Fitur utama dari JMeter adalah platform Java yang dapat dijalankan pada semua sistem. JMeter dapat digunakan dengan distributed mode, JMeter juga mendukung protokol HTTP, protokol SMTP, protokol POP3, protokol LDAP protokol JDBC, protokol FTP, protokol JMS, protokol SOAP, dan protokol TCP tanpa memerlukan plugin. Pre-processor dan post-processor pada sampler dapat dikostum sesuai dengan kebutuhan, external plugin dari pihak ketiga dapat digunakan untuk memvisualisasi hasil test [8].

4. Result and Discussion

Hasil dari penelitian menghasilkan tes skrip yang digunakan untuk menguji performa dari website yang akan dijalankan pada JMeter. Berikut adalah penjelasan detail mengenai tes skrip yang telah dikembangkan.

4.1 Standard web config elements

Jmeter secara standar tidak dapat berfungsi seperti web browser karena Jmeter hanya melakukan http request menuju server dan tidak menimpan session, cache, cookie, dan header, agar membuat jmeter bertindak seperti web browser dibutuhkan 4 config elements seperti pada Gambar.



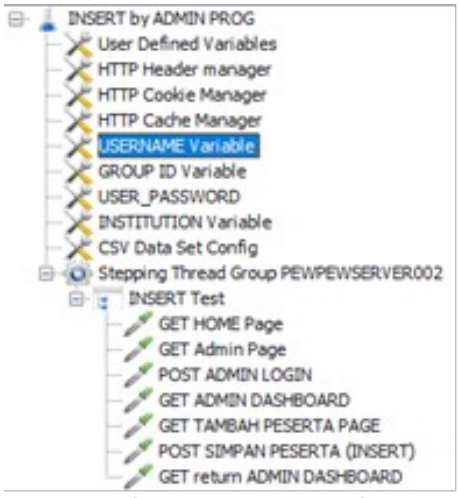
Gambar 2 Config element

config elements yang diperlukan untuk pengujian web diantara lain, User defined variable dalam pengujian web ini berfungsi untuk menyimpan ip dari server yang akan di uji pada saat membuat konfigurasi pengujian. HTTP Header Manager memungkinkan pengujian untuk menambahkan atau mengesampingkan HTTP header pada request dalam pengujian http request. HTTP header adalah komponen dari bagian header dari request atau respon pada HTTP yang mendefinisikan parameter operasi pada operasi http.

HTTP cookie manager, konfigurasi cookie manager dibiarkan secara default, fungsi utama dari HTTP cookie manager adalah untuk menyimpan cookie dari halaman web yang memiliki cookie seperti pada web browser, cookie manager akan menimpan cookie dan akan menggunakannya untuk request selanjutnya. HTTP cache manager berfungsi sebagai mensimulasikan fungsi cache pada browser setiap virtual user memiliki cache mereka sendiri secara default cache manager dapat menyimpan hingga 5000 item cache per virtual user thread.

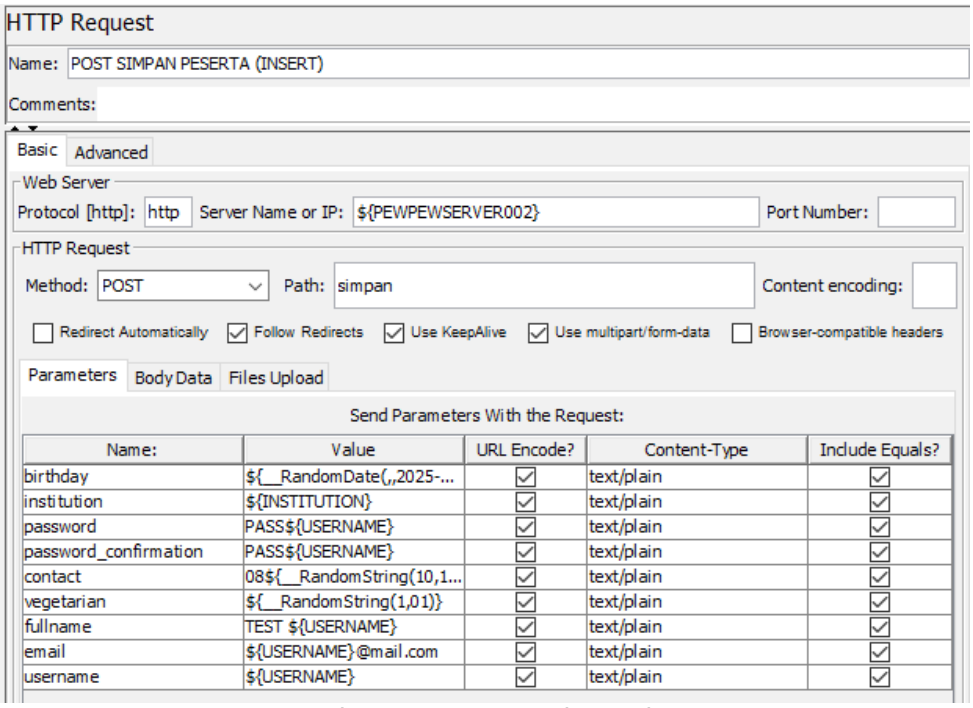
4.2. Insert script

Konfigurasi pengujian INSERT pada web dilakukan menggunakan akun Admin alur dari transaksi insert adalah pertama user akan membuka home page dari web kemudan admin akan melakukan login melalui halaman login admin, admin kemudian akan melakukan post data peserta yang akan ditambahkan dan akan di kembalikan ke halaman dashboard admin.



Gambar 3 Insert test script

HTTP request untuk menambahkan peserta pada halaman web, konfigurasi dari HTTP request pada Jmeter terdiri dari server name yang didapatkan dari variable yang disimpan pada user defined variable, method request menjadi POST dan path yang di tuju menjadi /simpan.

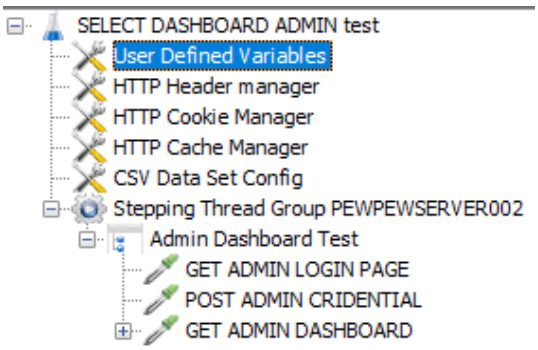


Gambar 4 Post request insert data

Send Parameter terdiri dari isi form atau data pengguna yang akan ditambahkan melalui web, parameter yang dikirim berupa birthday yang didapat menggunakan fungsi random date pada Jmeter, institution uang didapat menggunakan fungsi random variable, password dan password confirmation yang didapatkan melalui fungsi random variable username yang ditambah string “PASS”, contact yang didapatkan dengan menggunakan fungsi random string, vegetarian yang ditandai dengan 0 dan 1 sebagai falls dan true didapatkan menggunakan fungsi random string, fullname yang didapatkan dengan menggunakan fungsi username dengan penambahan string “TEST”, email yang didapatkan dengan menambahkan value dari fungsi username dengan domain email, dan username yang didapatkan dengan menggunakan random variable.

4.3. Select script

Konfigurasi pengujian SELECT pada web menggunakan halaman dashboard yang akan dibebani dengan request dari admin, pada halaman dashboard admin terdiri dari beberapa fungsi, pertama fungsi select data peserta yang berfungsi menampilkan data peserta dari cabang kompetisi admin yang mengakses dashboard, fungsi count yang terdiri dari count Jumlah Peserta yang menghitung jumlah peserta perorangan pada setiap kompetisi, count Jumlah Tim yang berfungsi menghitung jumlah tim yang berpartisipasi, dan count peserta veget/non veget yang berfungsi untuk menghitung jumlah peserta yang vegetarian dan tidak vegetarian.

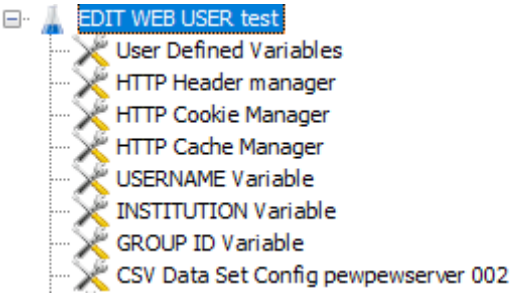


Gambar 5 Select test script

Pengujian Select menggunakan 3 http request yang mensimulasikan pengguna mengakses halaman dashboard admin, http request pertama berperan untuk mengakses halaman login admin, http request kedua berperan untuk login dari admin dengan method post, dan http request yang terakhir berfungsi untuk mengakses halaman dashboard admin.

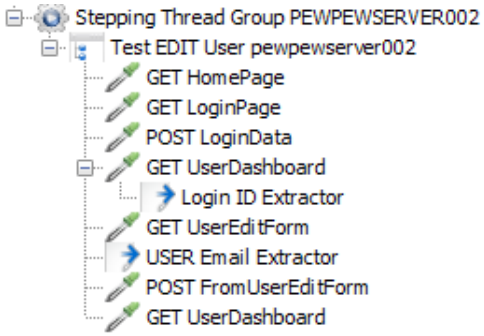
4.4. Update script

Pengujian fungsi Update pada web menggunakan fungsi edit pada group peserta dimana ketua group dapat menambahkan,mengedit dan menghapus anggota group yang terdaftar, fungsi ini membutuhkan login sebagai peserta dan mengubah data anggota yang terdaftar.



Gambar 6 Update test config element

config elements dari pengujian update web konfigurasi yang digunakan adalah user defined variable, 1 http header manager, 1 http cookie manager, 1 http cache manager, 3 random variable dan csv data set config yang digunakan sama seperti pada pengujian insert pada bagian 3.4.2 Web Insert Test.



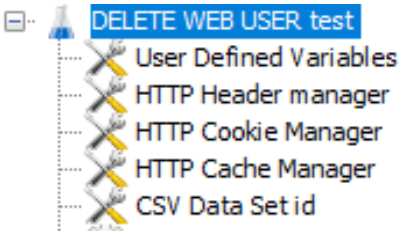
Gambar 7 Insert test script

konfigurasi pengujian fungsi update pada web pengujian diawali dengan user request home page kemudian melakukan login untuk dapat mengakses dashboard peserta kemudian user dapat melakukan perubahan data peserta.

4.4. Delete script

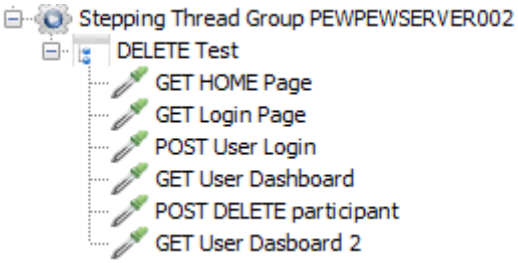
Pengujian fungsi delete pada web menggunakan fungsi delete pada group peserta dimana ketua group dapat menambahkan,mengedit dan menghapus anggota group yang terdaftar, fungsi ini membutuhkan login sebagai peserta dan menghapus data anggota yang terdaftar.

Konfigurasi dari pengujian delete melalui web menggunakan standart config element yang telah dibahas pada halaman 80, dan dengan konfigurasi csv dataset yang menyimpan id pengguna yang akan dihapus.



Gambar 8 Delete test config element

Skenario pengujian dimulai dengan request home page yang kemudian dialihkan ke halaman login peserta, setelah login kemudian peserta akan dialihkan ke halaman dashboard dan dapat menghapus data anggota tim dengan menggunakan fungsi delete pada halaman web.



Gambar 9 Delete test script

Konfigurasi pada file JMX menggunakan 6 http request sampler yang akan mengirimkan request ke server dimulai dengan get halaman home, get login page, post login, get halaman dashboard, post delete request, dan get dashboard.

Waktu untuk menyelesaikan semua request dihitung dengan menggunakan transaction controller, setap http request akan memiliki response time tersendiri dan fungsi dari transaction controller adalah untuk mengghitung waktu yang diperlukan untuk menghapus suatu akun dimulai dengan request home page hingga re-derect kembali ke halaman dashboard peserta.

5. Conclusion

Penulis mengusulkan pengembangan test script dengan menggunakan metode Software Testing Life Cycle dapat dapat mempermudah developer dalam melakukan pengembangan test script yang akan di gunakan pada load testing, dengan mengikuti alur STCL developer akan mendapatkan informasi seputar website yang akan di uji dan dapat mengembangkan test script yang dapat mencangkup load test pada fungsi krusial seperti CRUD yang akan di gunakan oleh pengguna.

References

[1] M. Niranjanamurthy, K. K. S, A. Saha, and D. Chahar, “Comparative Study on Performance Testing with JMeter,” 2016, doi: 10.17148/IJARCCE.

[2] R. Marx, “Web Performance Automation for the People,” *Companion Proceedings of the The Web Conference 2018*, no. April, pp. 825–829, 2018, doi: 10.1145/3184558.3186570.

[3] N. SyamimiSaid, R. Alsaqour, H. Shaker, M. Abdelhaq, O. Alsaqour, and M. Uddin, “Review on web performance,” *Journal of Engineering and Applied Sciences*, vol. 9, no. 1, pp. 18–23, 2014, doi: 10.3923/jeasci.2014.18.23.

[4] M. A. Jamil, M. Arif, N. S. A. Abubakar, and A. Ahmad, “Software testing techniques: A literature review,” *Proceedings - 6th International Conference on Information and Communication Technology for the Muslim World, ICT4M 2016*, no. November 2017, pp. 177–182, 2017, doi: 10.1109/ICT4M.2016.40.

[5] A. M. Kale, V. v Bandal, and K. Chaudhari, “A Review Paper on Software Testing,” *International Research Journal of Engineering and Technology*, p. 1268, 2019, [Online]. Available: www.irjet.net.

[6] S. S. Gokhale and J. Lu, “Performance and Availability Analysis of an E-Commerce Site,” in *30th Annual International Computer Software and Applications Conference (COMPSAC’06)*, Sep. 2006, vol. 1, no. July, pp. 495–502, doi: 10.1109/COMPSAC.2006.65.

[7] S. Nachiyappan and S. Justus, “Cloud testing tools and its challenges: A comparative study,” *Procedia Computer Science*, vol. 50, pp. 482–489, 2015, doi: 10.1016/j.procs.2015.04.018.

[8] A. S. Monika Sharma, Vaishnavi S. Iyer, Sugandhi Subramanian, “A Comparative Study on Load Testing Tools,” *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 4, no. 2, pp. 1906–1912, 2016, doi: 10.15680/IJIRCCE.2016.0402201.