



Computer Based Information System Journal

ISSN (Print): 2337-8794 | E- ISSN : 2621-5292
 web jurnal : <http://ejournal.upbatam.ac.id/index.php/cbis>



PERFORMANCE DAN STRESS TESTING DALAM MENGOPTIMASI WEBSITE

Doni Andriansyah

STMIK Nusa Mandiri Jakarta, Indonesia.

INFORMASI ARTIKEL

Diterima Redaksi: Februari 2019
 Diterbitkan Online: Maret 2019

KATA KUNCI

Performance test, stress test, optimasi web, pengujian web, wapt

KORESPONDENSI

E-mail: andrean.devz@gmail.com

A B S T R A C T

Kendala yang sering dihadapi pengguna ketika mengakses informasi dari suatu website adalah waktu respon yang terkadang lama dari website tersebut dalam menampilkan informasi. Hal ini terjadi karena faktor elemen yang ada didalam konten website tersebut, seperti gambar dengan resolusi tinggi, atau penggunaan struktur script yang tidak mengikuti aturan penulisan program yang baik. Untuk itu diperlukan optimasi untuk meningkatkan performa website agar menjadi lebih baik. Metode yang digunakan dalam penelitian ini adalah menguji performa dan tingkat stress pada sebuah website menggunakan tools PageSpeed Insights, GTmetrix, dan Web Application Tools (WAPT). Hasil dari penelitian ini berupa dokumentasi pengujian yang telah dilakukan dan diimplementasikan terhadap website terkait. Terdapat perbedaan score yang signifikan terhadap hasil pengujian sebelum dilakukannya optimasi dengan setelah dilakukannya optimasi website.

I. Latar Belakang

Website merupakan salah satu media yang dapat menyajikan informasi dengan cepat dan baik keseluruh dunia. Sebuah website harus memiliki ketepatan dan kecepatan dalam menghadirkan sebuah informasi ke hadapan pengguna.

Kecepatan sebuah website dalam menghadirkan informasi dipengaruhi oleh beberapa faktor yang umum ditemukan diantaranya penggunaan gambar dengan resolusi

tinggi pada halaman website, dan penggunaan atau struktur *script* yang tidak efisien sehingga menyebabkan *response time* yang lama ketika akan menampilkan halaman website. Optimasi diperlukan untuk meningkatkan kembali performa dengan cara melakukan serangkaian pengujian pada halaman website.

Pengujian merupakan tahapan yang penting didalam daur hidup pengembangan perangkat lunak/*Software Development Life Cycle* (SDLC) [1]. Menurut [2] pengujian adalah tahapan yang

harus dilalui untuk menghasilkan produk sistem informasi yang baik dan berkualitas. Pada tahapan ini pengembang dituntut apakah sistem telah dibangun dengan benar, atau sudahkah membangun sebuah sistem yang benar.

Pengujian dilakukan untuk mengoptimasi website dengan menemukan kesalahan atau kekurangan dalam proses pengembangan untuk kemudian dianalisa dan diperbaiki.

Glen Myers dalam [3] menetapkan beberapa aturan sebagai tujuan dari uji coba, yaitu:

- a. Uji coba merupakan proses eksekusi program dengan tujuan untuk menemukan kesalahan.
- b. Sebuah uji coba kasus yang baik adalah yang memiliki probabilitas yang tinggi dalam menemukan kesalahan-kesalahan yang belum terungkap.
- c. Uji coba yang berhasil adalah yang mengungkap kesalahan yang belum ditemukan.

Seperti yang disampaikan oleh [4] bahwa dalam membangun sebuah produk sistem informasi yang sukses adalah bergantung pada dua bahan dasar, yaitu fungsionalitas dan performa. Fungsionalitas mengacu pada apa yang dapat dilakukan oleh sistem. Sedangkan performa mengacu pada kemampuan sistem dalam memberikan informasi dengan cepat dan akurat meskipun banyak pengguna berinteraksi atau dalam keterbatasan sumber daya.

Penelitian ini berfokus pada pengujian performa dan stress sebuah website untuk mengetahui tingkat performa website.

II. Kajian Literatur

A. Definisi Pengujian

Pengujian adalah satu set aktifitas yang direncanakan dan sistematis untuk menguji dan mengevaluasi kebenaran yang diinginkan [5]. Aktifitas pengujian terdiri dari satu set atau sekumpulan langkah dimana dapat menempatkan desain kasus uji yang spesifik dan metode pengujian.

Dalam lingkup yang lebih besar, terdapat pengujian perangkat lunak yang memiliki cakupan luas dan sering dikaitkan dengan verifikasi (*verification*) dan validasi (*validation*) (V&V). Verifikasi mengacu pada serangkaian aktifitas yang menjamin bahwa perangkat lunak

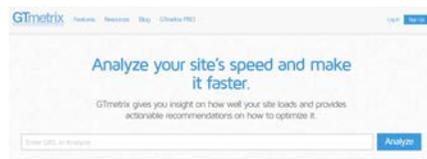
mengimplementasikan fungsi dengan benar. Sedangkan validasi mengacu pada serangkaian aktifitas yang berbeda yang menjamin bahwa perangkat lunak sudah sesuai dengan yang diinginkan.

B. Performance Testing

Performance testing atau pengujian performa bertujuan untuk memverifikasi performa sistem secara spesifik [6] seperti waktu respon, ketersediaan layanan, dan jumlah halaman yang diakses. Pengujian dilakukan dengan cara simulasi oleh banyak pengguna secara serentak dengan rentang waktu yang telah ditentukan. Pada aplikasi berbasis web, performa sistem merupakan masalah yang kritis. Menurut [1] ada beberapa jenis *performance test*, yaitu *stress test*, *load test*, *strength test*, dan *volume test*.

C. GTmetrix

GTmetrix dapat diakses melalui <https://gtmetrix.com>, digunakan untuk menganalisa kecepatan dalam menampilkan halaman website. Fitur GTmetrix terdiri dari PageSpeed dan YSlow *scores*, rekomendasi optimasi, *Page Load Details* yang terdiri dari waktu, ukuran, dan jumlah permintaan akan suatu halaman website. Cara kerja GTmetrix adalah mengumpulkan informasi dari halaman website, menganalisa YSlow *score*, menganalisa PageSpeed Score, kemudian *generate report* hasil analisa.

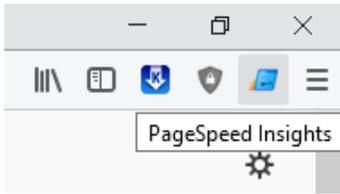


Sumber: Dokumen pribadi (2019).

Gambar 1. GTmetrix.

D. PageSpeed Insights

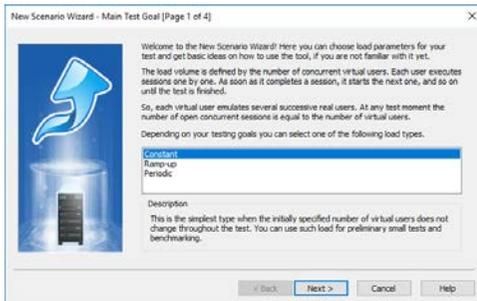
Tools ini disediakan untuk pengembang web, digunakan untuk mengecek dan menganalisa performa halaman website, dapat dipasang pada browser Chrome maupun Mozilla Firefox. Cara kerja PageSpeed Insights adalah menganalisa konten pada halaman website lalu *generate* saran yang dapat diikuti oleh pengembang untuk mengoptimasi website.



Sumber: Dokumen pribadi (2019).
Gambar 2. PageSpeed Insights.

E. Web Application Testing (WAPT)

Web Application Testing (WAPT) merupakan *tools* yang dapat digunakan untuk melakukan pengujian, diantaranya pengujian performa (*performance test*), pengujian tingkat stress (*stress test*), dan pengujian daya tahan (*endurance test*). Cara kerja WAPT adalah dengan menentukan terlebih dahulu banyak pengguna yang akan terlibat dalam proses simulasi pengujian dengan rentang waktu yang telah ditentukan. Kemudian menentukan berapa lama proses pengujian akan berlangsung.



Sumber: Dokumen pribadi (2019).
Gambar 3. WAPT.

III. Metodologi

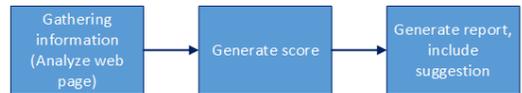
Metode penelitian yang digunakan adalah dengan menguji website baik secara langsung maupun simulasi menggunakan *tools* diantaranya GTmetrix, PageSpeed Insights, dan WAPT untuk mengetahui performa dan tingkat stress dalam memberikan informasi yang diakses oleh banyak pengguna yang berinteraksi pada rentang waktu yang hampir bersamaan.

PageSpeed Insights menerapkan score dengan skala yang terbagi kedalam tiga kategori, yaitu 90-100 (*fast*), 50-89 (*average*), dan 0-49 (*slow*). Sedangkan GTmetrix menggunakan grade dengan huruf A-F disertai angka. Semakin

<http://ejournal.upbatam.ac.id/index.php/cbis>

mendekati huruf A, maka grade akan semakin baik.

Secara umum, cara kerja GTmetrix dan PageSpeed Insights dapat dilihat pada gambar berikut:



Sumber: Dokumen pribadi (2019).
Gambar 4. Model Kerja Performance Tools.

IV. Pembahasan

Pengujian pertama menggunakan PageSpeed Insights yang terpasang pada browser Mozilla Firefox dengan obyek website <https://jas-aero.com> sebelum dilakukan optimasi dengan hasil sebagai berikut:



Sumber: Hasil penelitian (2019).
Gambar 5. PageSpeed Score Sebelum Optimasi.

Pada pengujian menggunakan PageSpeed sebelum optimasi menghasilkan score 45 dan masuk kedalam kategori *slow* (lambat). Rekomendasi yang diperoleh sebagai berikut:

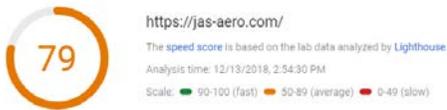
Opportunity	Estimated Savings
1. Serve images in next-gen formats	6.6%
2. Properly size images	3.48%
3. Enable text compression	3.35%
4. Eliminate render-blocking resources	2.01%
5. Defer unused CSS	1.95%
6. Efficiently encode images	1.2%
7. Reduce server response times (TTFB)	0.47%
8. Minify CSS	0.33%
9. Minify JavaScript	0.33%

Diagnostics	Issues
1. Serve static assets with an efficient cache policy	69 resources found
2. Use Timing marks and measures	1 issue
3. Ensure text remains visible during media load	1 issue
4. Minimize main-thread work	2.9s
5. Avoid excessive network payloads	Total size was 3,152.9 KB
6. Minimize Critical Request Depth	23 chains found

Passed audits	Count
7 audits	7 audits

Sumber: Hasil penelitian (2019).
Gambar 6. Rekomendasi Sebelum Optimasi.

Setelah dilakukan optimasi dan dilakukan pengujian kembali, maka terlihat perubahan yang cukup signifikan, dari score 45 sebelumnya meningkat menjadi 79 dan masuk kedalam kategori *average*.



Sumber: Hasil penelitian (2019).
 Gambar 7. PageSpeed Score Setelah Optimasi.

Banyaknya jumlah rekomendasi setelah optimasi berkurang. Jumlah yang berhasil di audit sebelumnya hanya 7, setelah optimasi meningkat menjadi 13.



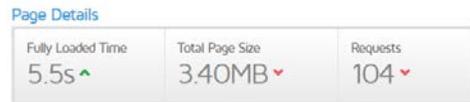
Sumber: Hasil penelitian (2019).
 Gambar 8. Rekomendasi Setelah Optimasi.

Selanjutnya pengujian dengan GTmetrix sebelum dilakukan optimasi menghasilkan *performance score* untuk PageSpeed sebesar F(45%) dan YSlow E(54%).



Sumber: Hasil penelitian (2019).
 Gambar 9. GTmetrix Grade Sebelum Optimasi.

Untuk Page Details sebelum optimasi menghasilkan Total Page Size sebesar 3.40MB.



Sumber: Hasil penelitian (2019).
 Gambar 10. GTmetrix Page Details Sebelum Optimasi.

Rekomendasi sebelum optimasi masih terdapat grade F(0) pada empat baris teratas. Grade F merupakan grade terendah dalam pengujian menggunakan GTmetrix.

RECOMMENDATION	GRADE	TYPE	PRIORITY
Enable gzip compression	F (0)	SERVER	HIGH
Leverage browser caching	F (0)	SERVER	HIGH
Serve scaled images	F (0)	IMAGES	HIGH
Optimize images	F (0)	IMAGES	HIGH
Minify JavaScript	D (80)	JS	HIGH
Enable Keep-Alive	C (78)	SERVER	HIGH
Minify CSS	B (80)	CSS	HIGH
Minify HTML	B (80)	CONTENT	LOW
Specify image dimensions	A (97)	IMAGES	MEDIUM
Inline small CSS	A (90)	CSS	HIGH
Inline small JavaScript	A (80)	JS	HIGH
Optimize the order of styles and scripts	A (80)	CSS/JS	HIGH
Avoid CSS @import	A (80)	CSS	MEDIUM
Specify a cache validator	A (90)	SERVER	HIGH
Defer parsing of JavaScript	A (90)	JS	HIGH
Specify a Vary: Accept-Encoding header	B (80)	SERVER	LOW
Avoid bad requests	A (100)	CONTENT	HIGH
Avoid landing page redirects	A (100)	SERVER	HIGH
Minimize redirects	A (100)	CONTENT	HIGH
Minimize request size	A (100)	CONTENT	HIGH
Put CSS in the document head	A (100)	CSS	HIGH
Remove query strings from static resources	A (80)	CONTENT	LOW
Serve resources from a consistent URL	A (100)	CONTENT	HIGH
Combine images using CSS sprites	A (100)	IMAGES	HIGH
Prefer asynchronous resources	A (100)	JS	MEDIUM
Specify a character set early	A (100)	CONTENT	MEDIUM
Avoid a character set in the meta tag	A (100)	CONTENT	LOW

Sumber: Hasil penelitian (2019).
 Gambar 11. Rekomendasi GTmetrix Sebelum Optimasi.

Setelah dilakukan optimasi dan dilakukan pengujian kembali, terlihat perubahan yang cukup signifikan. Pada pengujian sebelumnya PageSpeed score menunjukkan F(45%), kini menjadi B(80%). YSlow score dari E(54%) meningkat menjadi C(71%).

Performance Scores



Sumber: Hasil penelitian (2019).
Gambar 12. Performance Scores GTmetrix Setelah Optimasi.

Perubahan signifikan terlihat jelas pada *Total Page Size*, yaitu dari 3.40MB pada pengujian sebelumnya meningkat menjadi 2.78MB pada pengujian setelah optimasi.

Page Details



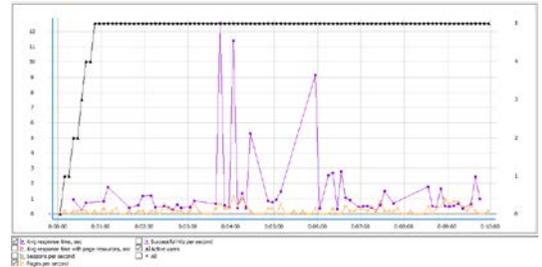
Sumber: Hasil penelitian (2019).
Gambar 13. GTmetrix Page Details.

Peningkatan grade pun terjadi pada rekomendasi setelah optimasi dilakukan.

RECOMMENDATION	GRADE	TYPE	PRIORITY
Optimize images	F	IMAGES	HIGH
Enable Keep-Alive	E	SERVER	HIGH
Enable gzip compression	E	SERVER	HIGH
Leverage browser caching	D	SERVER	HIGH
Mimify HTML	D	CONTENT	LOW
Specify image dimensions	D	IMAGES	MEDIUM
Inline small CSS	D	CSS	HIGH
Inline small JavaScript	D	JS	HIGH
Optimize the order of styles and scripts	D	CSS/JS	HIGH
Avoid CSS @import	D	CSS	MEDIUM
Mimify CSS	D	CSS	HIGH
Specify a cache validator	D	SERVER	HIGH
Mimify JavaScript	D	JS	HIGH
Serve scaled images	D	IMAGES	HIGH
Defer parsing of JavaScript	D	JS	HIGH
Avoid bad requests	D	CONTENT	HIGH
Avoid landing page redirects	D	SERVER	HIGH
Minimize redirects	D	CONTENT	HIGH
Minimize request size	D	CONTENT	HIGH
Put CSS in the document head	D	CSS	HIGH
Remove query strings from static resources	D	CONTENT	LOW
Serve resources from a consistent URL	D	CONTENT	HIGH
Combine images using CSS sprites	D	IMAGES	HIGH
Prefer asynchronous resources	D	JS	MEDIUM
Specify a character set early	D	CONTENT	MEDIUM
Avoid a character set in the meta tag	D	CONTENT	LOW
Specify a Vary: Accept-Encoding header	D	SERVER	LOW

Sumber: Hasil penelitian (2019).
Gambar 14. Rekomendasi GTmetrix Setelah Optimasi.

Selanjutnya hasil pengujian performa dan stress menggunakan Web Application Testing (WAPT) dengan metode Ramp-up sebagai berikut:



Sumber: Hasil penelitian (2019).
Gambar 15. Pengujian Performa dan Stress WAPT.

Dari grafik diatas menghasilkan ringkasan pengujian sebagai berikut:

Tabel 1. Ringkasan Pengujian WAPT.

Test result	SUCCESS
Profile	Profile1
Successful sessions	10
Failed sessions	0
Successful pages	138
Failed pages	0
Successful hits	1378
Failed hits	3
Other errors	0
Total Kbytes sent	1159
Total Kbytes received	63188
Avg response time, sec (with page resources)	1.71(9.95)

Sumber: Hasil penelitian (2019).

V. Kesimpulan

Dari hasil pengujian performa dan stress menggunakan *tools* PageSpeed Insights dan GTmetrix dapat disimpulkan bahwa *performance scores* keduanya mengalami peningkatan yang signifikan setelah dilakukannya optimasi terhadap website.

Hasil pengujian menggunakan Web Application Tools (WAPT) menunjukkan rata-rata waktu respon dalam menampilkan halaman

web adalah 1.71 detik per halaman, dengan hasil pengujian SUKSES.

Optimasi terhadap website sangat penting dilakukan untuk memberikan kepuasan terhadap pengguna ketika mengakses informasi dari suatu website.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada UPBatam yang telah memberikan kesempatan untuk mempublikasikan karya tulis ilmiah ini di CBIS.

Daftar Pustaka

- [1] S. Dhiman and P. Sharma, "Performance Testing: A Comparative Study and Analysis of Web Service Testing Tools," *Int. J. Comput. Sci. Mob. Comput.*, vol. 5, no. 6, pp. 507–512, 2016.
- [2] D. Andriansyah, "Pengujian Kotak Hitam Boundary Value Analysis Pada Sistem Informasi Manajemen Konseling Tugas Akhir," *Indones. J. Netw. Secur.*, vol. 7, no. 1, pp. 20–25, 2018.
- [3] D. Gea, "ANALISA PENGUJIAN OPTIMALISASI KINERJA WEBSITE," *Semin. Nas. Apl. Teknol. Inf.*, vol. 2011, no. Snati, pp. 17–18, 2011.
- [4] H. Sarojadevi, "Performance Testing: Methodologies and Tools," *J. Inf. Eng. Appl.*, vol. 1, no. 5, pp. 5–13, 2011.
- [5] A. S. Rosa and M. Shalahuddin, *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Penerbit Informatika, 2014.
- [6] M. S. Sharmila and E. Ramadevi, "Analysis of Performance Testing on Web Applications," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 3, no. 3, pp. 5258–5260, 2014.