

IMPLEMENTASI MODULE GOOGLE PAGESPEED PADA APACHE WEB SERVER UNTUK MENINGKATKAN PERFORMA APLIKASI WEB

Muhammad Faizal Kurniawan¹⁾, Wahyu Setianto²⁾, Agus Ilyas³⁾

STMik Widya Pratama Jl. Patriot No. 25 Pekalongan

faizal@stmik-wp.ac.id, kian@stmik-wp.ac.id, agusilyas@stmik-wp.ac.id

Abstrak

Performa aplikasi web salah satunya dapat dipengaruhi dari konfigurasi web server yang tepat. Meskipun memiliki hardware dengan spesifikasi yang tinggi namun jika konfigurasinya tidak tepat maka performa aplikasi web yang ada di dalam web server tersebut tidak akan meningkat. Salah satu cara agar aplikasi web dapat diakses dengan cepat adalah dengan mengecilkan ukuran file yang di unduh dari web server ke browser pengguna. Selama ini developer web akan melakukan kompresi terhadap file-file gambar, js dan css sebelum diupload ke web server. Google Page Speed adalah sebuah modul eksternal yang dapat dikonfigurasi ke dalam Apache web server untuk menggantikan tugas developer web dalam mengkompresi file-file sebelum diunggah. Fungsi dari modul ini adalah secara otomatis akan melakukan kompresi terhadap file-file gambar, css dan js secara realtime. Penelitian ini bertujuan untuk menguji performa aplikasi web sebelum dan setelah mengimplementasikan modul Google Page Speed pada web server yang dalam penelitian ini menggunakan Apache Web Server.

Kata kunci: optimasi web, kompresi konten, web server

1. Pendahuluan

Seiring dengan meningkatnya layanan akses ke jaringan internet, maka semakin banyak aplikasi-aplikasi berbasis web yang dibuat. Aplikasi berbasis web memiliki kelebihan dibandingkan dengan aplikasi desktop antara lain karena tidak perlu diinstall di perangkat pengguna dan proses update dapat perlu dilakukan. Portal berita, media sosial, game, aplikasi perkantoran hingga e-commerce sebagian besar tersedia dalam bentuk aplikasi web. Berbeda dengan aplikasi desktop yang terinstall langsung pada perangkat pengguna, aplikasi web disimpan didalam web server. Web server bertugas menjembatani kebutuhan pengguna dan aplikasi. Salah satu web server yang banyak digunakan adalah Apache.

Apache banyak digunakan karena kemudahan instalasi dan konfigurasinya. Seluruh file pada aplikasi web tersimpan di dalam web server dan pengguna aplikasi dapat mengakses melalui jaringan baik itu intranet maupun internet. Karena file-file pada aplikasi web tersimpan di dalam web server, maka cepat atau tidaknya loading sebuah aplikasi web dapat dipengaruhi oleh beberapa hal antara lain: bandwidth jaringan yang tersedia, spesifikasi dan konfigurasi web server dan performa dari aplikasi web itu sendiri.

Berbagai penelitian telah dilakukan untuk meningkatkan performa aplikasi web baik dari sisi jaringan, web server maupun teknik pemrogramannya. Salah satu cara meningkatkan kecepatan aplikasi web adalah dengan cara mengusahakan agar file-file assets (file gambar, css dan js) yang diakses oleh pengguna berukuran sekecil mungkin, karena semakin kecil filenya maka makin kecil bandwidth yang dibutuhkan untuk mengaksesnya dan makin cepat pula loading aplikasinya. File-file dapat dikecilkan dengan cara melakukan kompresi sebelum diunggah di webserver, namun demikian cara ini tentunya memakan waktu dan tenaga bagi pada developer web.

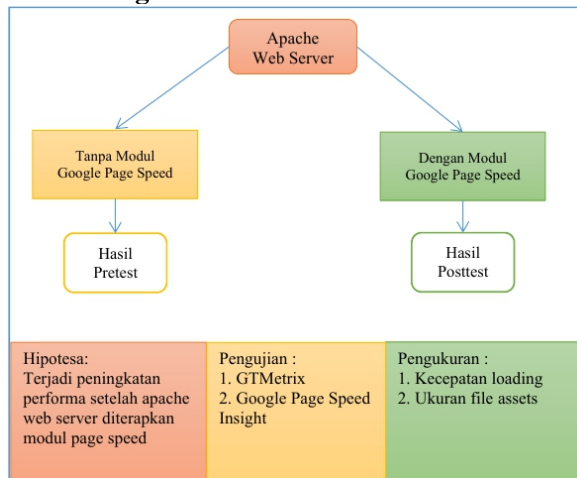
Modul google page speed adalah sebuah module tambahan yang dapat diinstall pada web server. Saat ini modul page speed tersedia untuk Apache dan Nginx. Modul ini berfungsi untuk mengotomatisasi proses kompresi file assets. Modul page speed secara otomatis membuat potongan gambar sesuai dengan ukuran yang didefinisikan. Module page speed juga mampu melakukan kompresi dan caching terhadap file css dan js secara otomatis. Penelitian ini bertujuan mengimplementasikan modul google page speed pada Apache web server dan untuk menguji

performa aplikasi web sebelum dan setelah web server dikonfigurasi dengan modul ini.

Hipotesa dari penelitian ini adalah performa aplikasi web akan meningkat setelah google page speed diaktifkan. Namun kemungkinan peningkatan performa akan terjadi setelah web diakses pada kali kedua, sedangkan pada saat pertama diakses kemungkinan akan terjadi perlambatan karena ada proses kompresi dan caching yang dilakukan oleh module page speed. Proses pengujian akan dilakukan menggunakan metode pretest-posttest dengan tool pengujian yaitu: GTMetrix dan Google PageSpeed Insight.

2. Metode Penelitian

2.1 Kerangka Pemikiran



Gambar 1. Kerangka Pemikiran

2.2 Tahapan Eksperimen

Beberapa tahapan dilakukan dilakukan melakukan eksperimen, yaitu: (1) Persiapan (2) Instalasi dan Konfigurasi VPS (3) Pembuatan Halaman Uji (4) Pengujian

2.2.1 Persiapan

Pada tahap persiapan, dilakukan dengan mempersiapkan lingkungan server. Server yang digunakan memiliki spesifikasi sebagai berikut:

- CPU 1 Core
- Memory 1024 MB
- SSD 20 GB
- OS : Debian 10

2.2.2 Instalasi dan konfigurasi VPS

Tahapan berikutnya adalah dengan melakukan instalasi dan konfigurasi VPS. Tahapan instalasi adalah:

Instalasi Apache 2.4, perintah untuk melakukan instalasi Apache 2.4 adalah sebagai berikut:

```
sudo apt-get install apache2
```

Instalasi Module PageSpeed, modul pagespeed dapat diunduh dilaman resmi Google Developer, yaitu dialamat url berikut: <https://developers.google.com/speed/pagespeed/module>. Perintah yang digunakan untuk melakukan tahapan instalasi modul pagespeed adalah

```
wget https://dl-ssl.google.com/dl/linux/direct/mod-pagespeed-stable_current_amd64.deb
dpkg -i mod-pagespeed-stable_current_amd64.deb
```

2.2.2.1 Pembuatan Halaman Uji

Langkah berikutnya adalah dengan membuat halaman web sebagai bahan uji coba. Halaman yang dibuat memuat assets berupa gambar, css dan javascript.

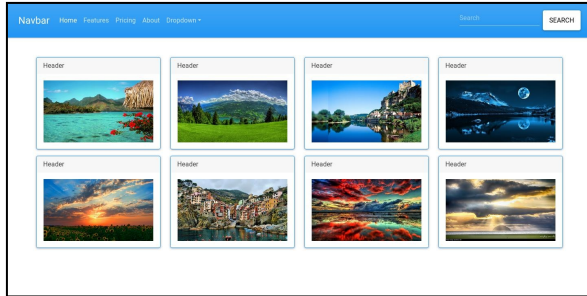
2.2.2.2 Pengujian

Tahap terakhir adalah dengan melakukan pengujian pre test dan post test. Alat bantu pengujian menggunakan GTMetrix dan Google PageSpeed Insight, dimana akan dibanding performa halaman yang telah dibuat sebelumnya pada saat sebelum mengaktifkan modul pagespeed dan sesudah mengaktifkan modul pagespeed.

3. Hasil Sistem

Halaman uji coba yang dibuat pada penelitian ini adalah sebuah halaman web yang dibuat dengan format HTML yang memuat:

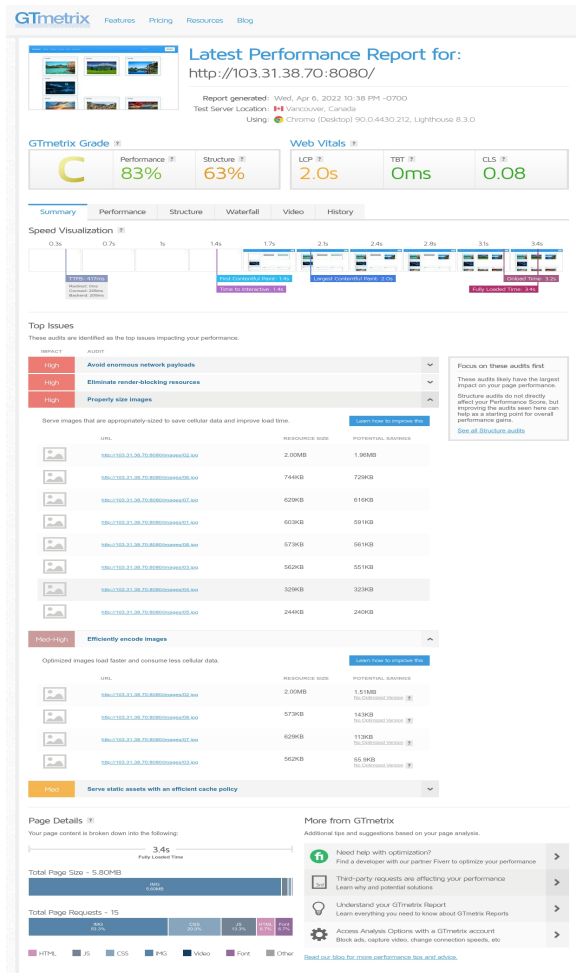
- Gambar yang berukuran antara 244 KB hingga 2 MB
- Non-compressed file css yang merupakan file bawaan dari Bootstrap dan juga menggunakan tema dari Bootwatch.
- Non-compressed file javascript yang merupakan file bawaan Bootstrap dan juga tambahan JQuery.



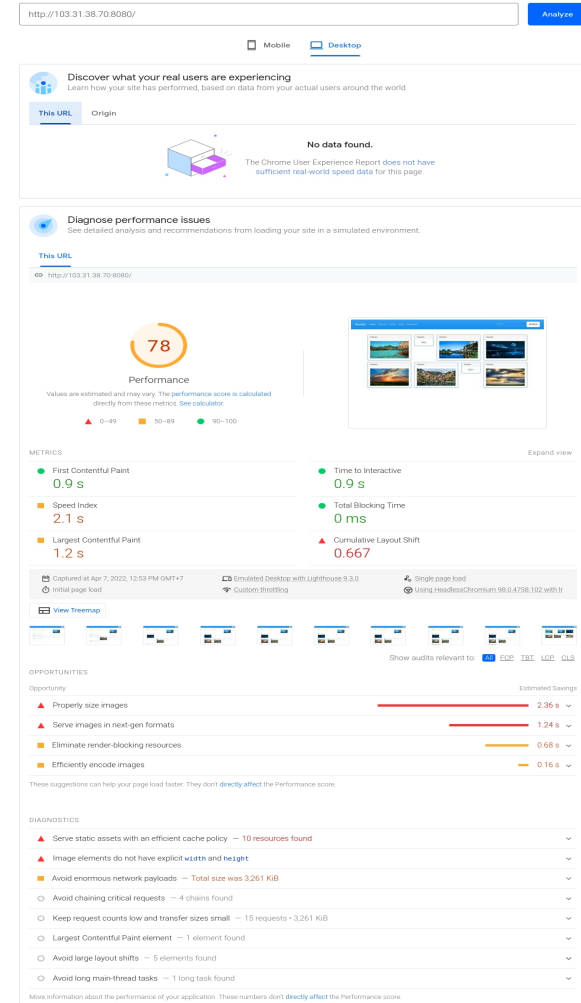
Gambar 2. Halaman Uji

4. Pengujian 4.1. Uji Pre Test

Uji pre test dilakukan dengan mengukur halaman uji coba pada Apache 2.4 sebelum mengaktifkan modul PageSpeed. Berikut ini adalah hasil tangkapan layar dari GTmetrix dan Google PageSpeed Insight.



Gambar 3. Hasil Pre Test dengan GTmetrix



Gambar 4. Hasil Pre Test dengan Google PageSpeed Insight

Melihat hasil pre test dari kedua tools (GTmetrix dan Google PageSpeed Insight), terlihat ada salah 1 indikator yang direkomendasikan oleh keduanya yang berkaitan dengan penelitian ini yaitu properly size images. Hal disebabkan karena website uji coba menggunakan gambar dengan ukuran 1920x1080 tetapi hanya menampilkannya dengan ukuran lebar 270. Hal ini tentunya akan membuat loading halaman akan lambat, karena file gambar yang diunduh oleh browser tetap file gambar yang berukuran 1090x080. Idealnya jika jika gambar yang akan ditampilkan berukuran lebar 270 maka seharusnya gambar harus terlebih dulu di resize dengan ukuran lebar 270 juga. Harapannya dengan menggunakan modul PageSpeed gambar-gambar akan diresize terlebih secara

otomatis sesuai dengan ukuran yang didefinisikan sebelum ditampilkan ke browser.

4.2. Uji Post Test

Tahap Post Test dilakukan dengan terlebih dahulu mengaktifkan modul PageSpeed pada Apache dan mengaktifkan fungsi-fungsi yang berkaitan dengan optimasi file assets

```

IfModule pagespeed_module>
# Turn on mod_pagespeed. To completely disable mod_pagespeed, you
# can set this to "off".
ModPagespeed on

# Direct Apache to send all HTML output to the mod_pagespeed
# output handler.
AddOutputFilterByType MOD_PAGESPEED_OUTPUT_FILTER text/html

# If you want mod_pagespeed process XHTML as well, please uncomment this
# line.
# AddOutputFilterByType MOD_PAGESPEED_OUTPUT_FILTER application/xhtml+xml

```

Gambar 5. Mengaktifkan Modul PageSpeed

```

# Explicitly enables specific filters. This is useful in
# conjunction with ModPagespeedRewriteLevel. For instance, filters
# not included in the CoreFilters may be enabled using this
# directive. This directive contains a comma-separated list of
# filter names, and can be repeated.
#
ModPagespeedEnableFilters rewrite_images
ModPagespeedEnableFilters rewrite_javascript,rewrite_css
ModPagespeedEnableFilters collapse_whitespace,elide_attributes
ModPagespeedEnableFilters inline_images,resize_images
ModPagespeedEnableFilters recompress_images
ModPagespeedEnableFilters combine_javascript,combine_css

```

Gambar 6. Mengaktifkan Filter Modul PageSpeed

Setelah modul PageSpeed diaktifkan, ada beberapa perubahan yang terjadi ketika melihat source kode dari halaman uji yang dihasilkan di browser. Nama file javascript, css dan file gambar berubah. Hal ini dikarenakan sebelum ditampilkan di browser, modul PageSpeed akan mengoptimalkan ukuran gambar, css, javascript serta html.

```

<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="vendor/twbs/bootstrap/dist/css/A.bootstrap.css.pagespeed.cf.0703W6KAW.css">
<link rel="stylesheet" href="assets/A.bootstrap.css.pagespeed.cf.K07V9-8AwR.css">
<script src="assets/jquery-3.6.0.js.pagespeed.jm.L1AL7wZaL.js"></script>
<script src="vendor/twbs/bootstrap/dist/js/bootstrap.bundle.js.pagespeed.jm.0a5I2tA4AL.js"></script>

```

Gambar 7. File CSS dan Javascript dimampatkan oleh modul PageSpeed

```

<div class="container">
<div class="row">
<div class="col-sm">
<div class="card border-primary mb-3" style="max-width: 20rem;">
<div class="card-header">Header</div>
<div class="card-body text-center">

</div>
</div>
<div class="col-sm">
<div class="card border-primary mb-3" style="max-width: 20rem;">
<div class="card-header">Header</div>
<div class="card-body text-center">

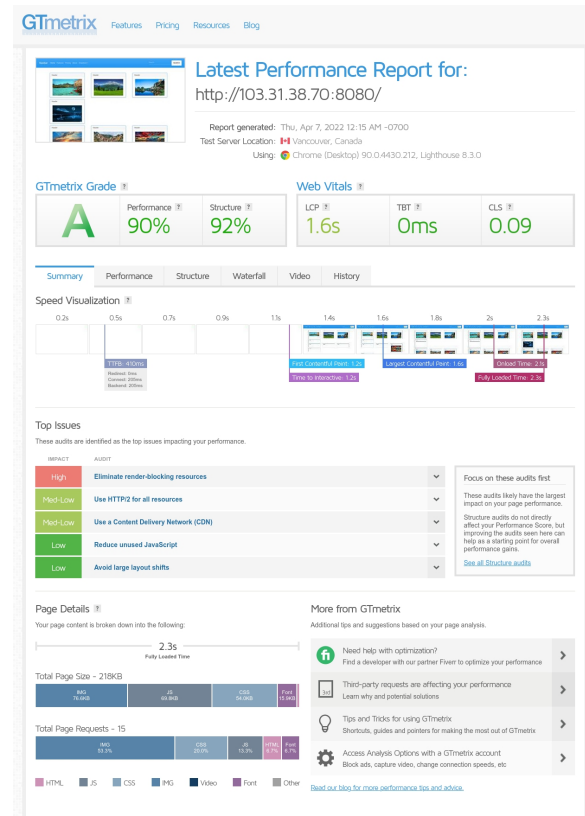
</div>
</div>
<div class="col-sm">
<div class="card border-primary mb-3" style="max-width: 20rem;">
<div class="card-header">Header</div>
<div class="card-body text-center">

</div>
</div>
</div>

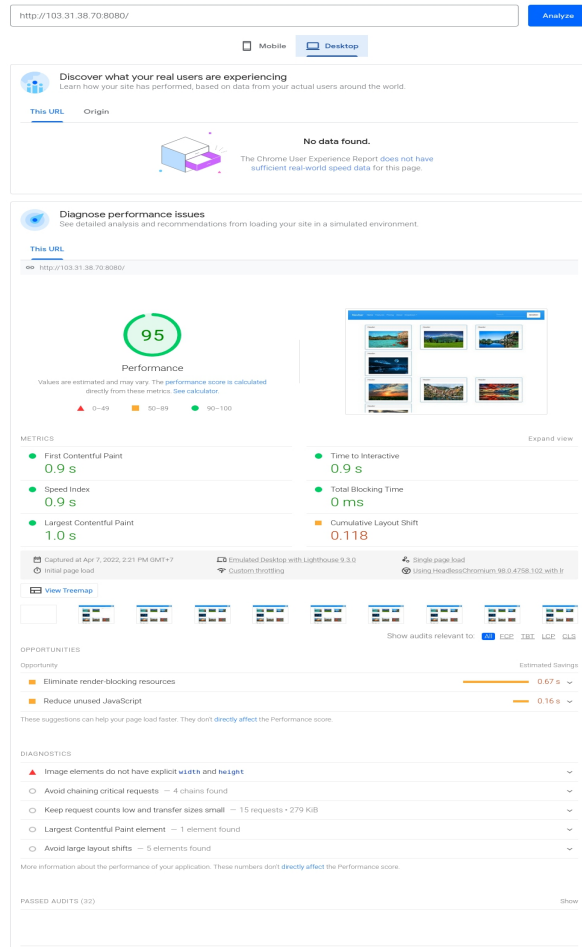
```

Gambar 8. File gambar diresize oleh modul PageSpeed sesuai ukuran yang didefinisikan

Perubahan juga terjadi ketika halaman uji di test menggunakan GTmetrix dan Google PageSpeed Insight. Terlihat bahwa sebelum mengaktifkan modul PageSpeed, halaman uji coba mendapat skor C pada GTmetrix dan 75 pada Google PageSpeed Insight. Namun setelah modul PageSpeed diaktifkan, halaman uji mendapatkan skor A pada GTmetrix dan 95 pada Google PageSpeed Insight. Selain itu rekomendasi properly size images juga sudah muncul lagi. Hal ini dapat disimpulkan bahwa gambar yang di tampilkan di browser sudah optimal.



Gambar 9. Hasil pengujian GTmetrix setelah mengaktifkan modul PageSpeed



Gambar 10. Hasil pengujian Google PageSpeed Insight setelah mengaktifkan modul PageSpeed

5. Kesimpulan dan Saran

5.1. Kesimpulan

Dari hasil penelitian yang telah dilakukan terdapat beberapa hal yang menjadi kesimpulan antara lain:

- (1) Modul PageSpeed dapat diterapkan dengan mudah dengan menggunakan Apache web server.
- (2) Modul PageSpeed terbukti dapat mengoptimalkan file-file assets pada halaman web.
- (3) Hasil pengujian menggunakan GTmetrix dan Google PageSpeed Insight menunjukkan bahwa setelah menerapkan module PageSpeed perfoma halaman web uji coba menjadi meningkat dan sangat baik.

5.2. Saran

Penerapan modul PageSpeed pada Apache web server telah terbukti meningkatkan perfoma halaman web. Namun demikian penelitian hanya terbatas membahas penggunaan modul tersebut pada Apache web server. Saran untuk peneliti selanjutnya adalah dengan menerapkan modul ini untuk web server lain seperti Nginx, Open Lite Speed dan lain sebagainya.

DAFTAR PUSTAKA

- Garry. (2012). Apa Kegunaan Folder Assets dalam Yii Framework? Retrieved September 28, 2021, from <http://www.computesta.com/blog/2012/03/apa-kegunaan-folder-assets-dalam-yii-framework/#.YVXfLbxBzeM>
- Nurmiati, E. (2012). Analisis dan Perancangan Web Server Pada Handphone. *Studia Informatika: Jurnal Sistem Informasi*, 5(2), 1–17. Retrieved from <http://download.portalgaruda.org/article.php?article=2481&val=329&title=A>
- NALISIS DAN PERANCANGAN WEB SERVER PADA HANDPHONE
- Dewanto, I. Joko. (2006). *Web Desain (Metode Aplikasi dan Implementasi)*. Yogyakarta: Graha Ilmu.
- Saraswati, I., Praptodiyono, S., Pramudyo, A. S., & Kurniawan. (2019). Increasing web server performance using the web balancing method. *IOP Conference Series: Materials Science and Engineering*, 673(1). <https://doi.org/10.1088/1757-899X/673/1/012065>
- N. Suradkar and S. Lomte, "VMware ESXi: Virtual Web Server performance evaluation with weighttp Benchmark," 2020 IEEE International Conference on Advent Trends in Multidisciplinary Research and Innovation (ICATMRI), 2020, pp. 1-4, doi: 10.1109/ICATMRI51801.2020.9398491.
- Google. (2018). PageSpeed Modules. <https://developers.google.com/speed/pagespeed/module>