

PERANCANGAN *LITESPEED CACHE* MENGUNAKAN METODE PPDIOO DI PT. ABC

Imam Arief Rahman¹, Iskandar Ikbal²

^{1,2} Program Studi Teknik Informatika - Universitas Komputer Indonesia
Jl. Dipatiukur No. 222-226, Coblong, Bandung, Jawa Barat 40232
E-mail: imam@blast.co.id¹, iskandar.ikbal@email.unikom.ac.id²

ABSTRAK

PT ABC merupakan suatu perusahaan yang bergerak dibidang IT Infrastruktur dan *Software Development*. Perusahaan tersebut menggunakan sebuah situs tertentu dalam memasarkan *services*. Berdasarkan hasil audit yang dilakukan terhadap situs tersebut dengan menggunakan *Lighthouse Engine* dari *Google Pagespeed Insight*, diperoleh bahwa situs tersebut memiliki *performance metrics* yang cukup rendah. Selain itu juga, penggunaan *resource* pada *Web server* Apache berbasis arsitektur *process-based* dalam menangani *request* masih belum efisien. Hal ini disebabkan oleh situs tersebut belum dilakukan optimasi melalui teknologi *cache* yang efektif. Oleh karena itu, dibutuhkan perancangan *infrastruktur cache* dan *web server* berbasis *event driven* seperti *Litespeed Web server* dengan menggunakan metode PPDIOO. Sehingga diharapkan akan tercapainya pengoptimasian *Web server* pada *web* perusahaan tersebut. Tahapan yang dilakukan diantaranya adalah fase *prepare*, dan *plan* dialur PPDIOO, fase *design* pada PPDIOO, dan kemudian fase *implementation* pada alur PPDIOO dengan melakukan instalasi dan optimasi serta pengujian NRFU sebanyak 8 kali dalam *mode mobile* dan *desktop*.

Kata Kunci : *Web server*, *Litespeed*, PPDIOO, Optimasi.

1. PENDAHULUAN

PT. ABC adalah perusahaan yang bergerak dibidang *cloud service*. Perusahaan ini salah satu penyedia layanan *cloud hosting* bernama ABC Compute. Layanan ABC Compute dapat di *deploy* di lebih dari 8 *datacenter*. ABC Cloud ini memiliki situs dengan domain www.blast.co.id. Berdasarkan hasil wawancara, diketahui bahwa situs tersebut menggunakan *cloud server* dan Apache *web server*.

Diketahui bahwa performa pada Apache arsitektur *Event-Based* mengalahkan *Process-Based*. Selain itu *Process-Based* juga memiliki potensi menciptakan *performance bottleneck* untuk trafik *request* yang sangat besar [3]. Hal yang berbeda terjadi pada *Litespeed* yang menggunakan arsitektur *event-based*. Karena *Litespeed* tidak menggunakan banyak *thread* sehingga dapat melayani banyak *request* tanpa membebani *server resources* itu sendiri [3].

Pengujian berdasarkan *Google Pagespeed Insight* pada situs www.blast.co.id menunjukkan bahwa *performance metrics* sangat rendah [9]. Berdasarkan hasil audit tersebut diketahui bahwa belum dilakukan optimasi secara efektif dalam penggunaan teknologi *cache* pada situs www.blast.co.id. Rendahnya skor *performance metrics* yang dimiliki, akan menyebabkan *impact* yang besar didalam bisnis. *Impact* tersebut diantaranya adalah menurunkan *user experience*, tingkat *engagement*, retensi, dan konversi/penjualan serta peringkat SEO dari suatu situs. Pengurangan waktu pemuatan halaman situs sebesar 100 ms saja dapat meningkatkan konversi sebesar 1.55% [7].

Berdasarkan permasalahan yang telah dijelaskan sebelumnya, maka perlu dilakukan perancangan *infrastruktur cache* pada *web server* situs www.blast.co.id agar menghasilkan skor *performance metrics* yang dapat meningkatkan *user experience*, *engagement*, retensi, konversi, dan peringkat SEO di mesin pencarian [7].

2. ISI PENELITIAN

2.1. Landasan Teori

2.1.1. Cache

Cache merupakan perangkat keras atau perangkat lunak yang digunakan untuk menyimpan sesuatu, biasanya data, untuk sementara waktu di lingkungan komputasi [5]. Berikut adalah beberapa jenis *cache* :

1. Browser/HTTP cache

Browser cache membantu mempermudah dan membuat *loading* lebih cepat [7]. Dengan *browser cache*, *web server* tidak perlu lagi melakukan *request* dan transmisi data untuk menampilkan situs yang ingin dikunjungi di browser[8].

2. OpCode Cache

OpCode Cache adalah *cache* pada *script* PHP dengan menyimpan hasil kompilasi dari PHP. Sehingga proses untuk melakukan eksekusi PHP jadi lebih singkat tanpa harus melalui tahapan *parsing* dan kompilasi dan disimpan didalam *memory*[10].

3. Object Cache

Dengan *object cache*, data objek dapat disimpan secara lokal sehingga tidak perlu diambil secara konstan untuk permintaan tambahan. Dengan demikian, *object cache* membantu meningkatkan kecepatan dan kinerja aplikasi *web*[5].

4. Page Cache

Page cache memiliki kemiripan dengan *cache* lain. Salah satu manfaat melakukan *page cache* adalah untuk meningkatkan kecepatan waktu *loading* sebuah halaman *website* untuk memberikan *user experience* yang lebih baik [4]. *Page cache* menyimpan halaman *web* lengkap untuk ditampilkan di lain waktu kepada pengunjung [5].

5. Content Delivery Network Cache

CDN cache adalah bentuk penyimpanan data yang lebih luas. Dengan *cache* CDN, konten situs *web* statis ditambahkan ke *server proxy* yang didistribusikan secara global. Ini memungkinkan pengunjung dari seluruh dunia untuk mengunduh konten situs Anda lebih cepat sehingga mempercepat waktu buka situs [5].

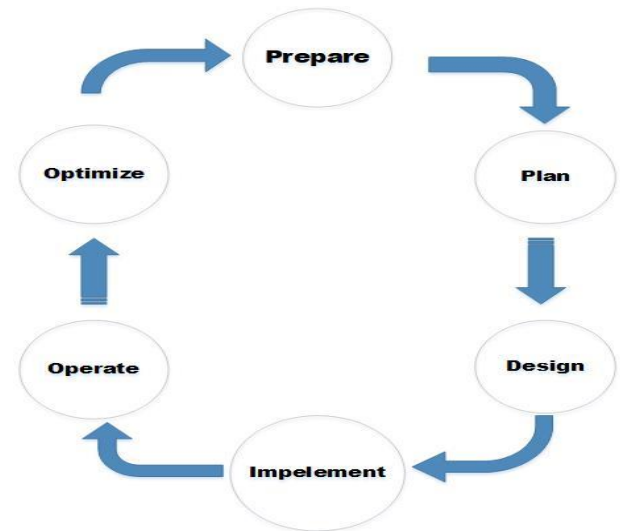
2.1.2. Litespeed

LiteSpeed Web server (LSWS), adalah perangkat lunak *web server*. LSWS dikembangkan oleh *LiteSpeed Technologies* milik pribadi. Perangkat lunak ini merupakan *Apache drop-in replacement*, artinya menggunakan format konfigurasi yang berbeda dengan *Apache*. [10]

Web server LiteSpeed meningkatkan kinerja dan skalabilitas *platform web hosting* melalui infrastruktur *cache* yang efisien dan efektif menggunakan *LSCache*. *LSCache* dapat *cache* halaman secara penuh. Ini. *LSCache* Plugin pada juga dapat dilakukan untuk mengkonfigurasi untuk *HTTP Cache*, *Object Cache*, *CDN Cache*, dan *OpCode Cache*. *Litespeed* memiliki kemampuan menangani lonjakan lalu lintas yang tiba-tiba serta membantu mengelola serangan *DDOS* tanpa perangkat keras mitigasi *DDOS*. [9]

2.2. Metode

Metode perancangan sistem akan dilakukan dengan mengikuti alur dari metode *PPDIOO*, dapat dilihat pada Gambar 1.1. Mulai dari *Prepare*, *Plan*, *Design*, *Implement*, dan *Optimize*[1]. Namun fase *Implement* dan *Optimize* tidak akan dilalui karena hanya dibatasi pada *Staging Environment*[2].



Gambar 1. Metode PPDIOO

2.2.1. Prepare

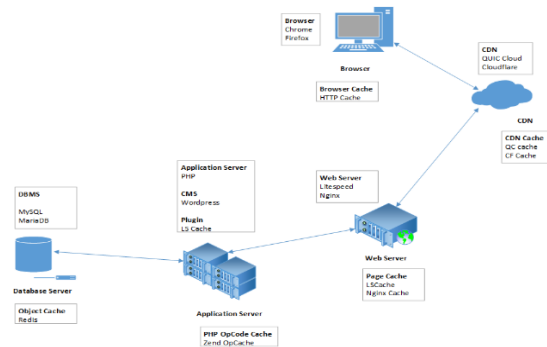
Merupakan tahap awal dalam perancangan sistem. Pada tahap ini akan dilakukan *Gathering Requirement* dan *overview* dari infrastruktur yang akan dirancang yakni *High Level Design*..

1. Gathering Requirement

Dari hasil wawancara dengan Divisi IT dari PT. ABC dapat disimpulkan diantaranya adalah menginginkan efisiensi *resources* pada server, menerapkan teknologi *cache* di *web server*, meningkatkan skor *performance metrics* dari *Lighthouse Engine* dan meningkatkan *user experience*, *engagement*, retensi, konversi dan peringkat *SEO* dari situs mereka melalui skor *performance metrics* yang tinggi.

2. High Level Design

Berikut adalah *overview* dari keseluruhan infrastruktur *cache* yang akan dirancang.



Gambar 2. High Level Design

2.2.2. Plan

Fase plan terdiri dari dua tahapan yakni Site Requirements Specifications dan tahapan Solution Test Plan.

1. Site Requirements Specifications

Tahap ini merupakan tahap Analisis Kebutuhan Perangkat Keras dan Analisis Kebutuhan Perangkat Lunak. Hasil analisis dapat dilihat pada Tabel 1 dan Tabel 2.

Tabel 1. Analisis Kebutuhan Perangkat Keras

No	Fungsi	Perangkat Keras	Jml	Biaya
1	Litesped Web server	Cloud Server	1	Rp 750000 / bulan
2	CDN	Object Storage	1	Rp 750000 / bulan
3	Client PC	PC atau Laptop	1	Rp0 / bulan

Tabel 2. Analisis Kebutuhan Perangkat Lunak

No	Fungsi	Perangkat Lunak	Versi	Biaya
1	Web server	Litespeed Starter	5.3.7	Rp0 / bulan
2	Sistem Operasi	Ubuntu	18.04	Rp0 / bulan
3	Control Panel	Plesk Onyx Web Admin	17.8.11	Rp0 / bulan
4	MySQL	Database Server	5.7.25	Rp0 / bulan
5	CMS	Wordpress	5.1.1	Rp0 / bulan

6	Plugin Cache Nginx	W3Total Cache	0.9.7.3	Rp0 / bulan
7	LS Cache	Plugin LS Cache	2.9.7.1	Rp0 / bulan

2. Solution Test Plan

Berikut adalah hasil audit performance metrics dari Lighthouse Engine pada tahapan yang sedang berjalan dari situs www.blast.co.id. Berikut adalah skor dari situs www.blast.co.id yang belum di optimasi dan belum menerapkan Litespeed cache.

Tabel 3. Performance metrics

Metrics	Waktu	Keterangan
First Contentful Paint	12,3s (lambat)	FCP menandai waktu ketika teks atau gambar mulai dilukis oleh browser.
First Meaningful Paint	14,5,0 s (lambat)	FMP mengukur kapan waktu konten utama dari halaman di tampilan.
Speed Index	19,3s (lambat)	SI mengukur seberapa cepat konten halaman terisi.
First CPU Idle	26s (lambat)	FCI mengukur seberapa cepat halaman interaktif secara minimal dan siap menerima input pengguna.
Time to Interactive	26,3s (lambat)	TTI mengukur seberapa cepat halaman interaktif secara penuh.
Max Potential First Input Delay	1030ms (lambat)	Potensi waktu maksimum yang dirasakan pengguna ketika memasukan input. Rekomendasi <50ms
Performance metrics Score	0 (lambat)	Skor Total berada pada kisaran (0-49) dikategorikan lambat.

2.2.3. Design

Pada fase ini akan dilakukan perancangan infrastruktur LS Cache untuk melakukan optimasi

pada situs www.blast.co.id. Pada fase ini hanya akan terdapat 1 tahapan, yakni *Low Level Design*.

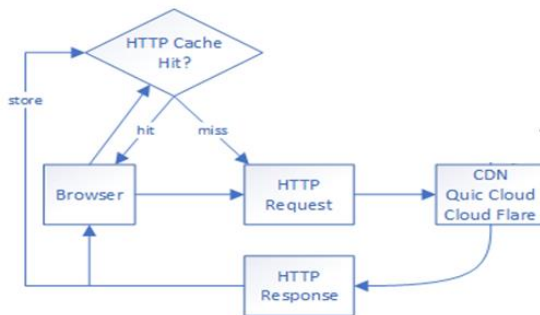


Gambar 3. *Low Level Design*

Low level Design memberikan Perancangan infrastruktur Litespeed *Cache* yang akan diterapkan pada situs www.blast.co.id dimana skemanya dapat dilihat pada Gambar 4.

1. **Browser/HTTP Cache**

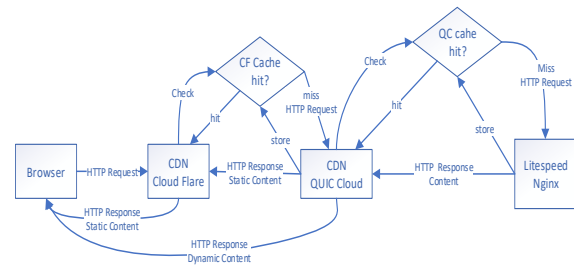
HTTP Cache atau *Browser cache* adalah strategi *cache* dari sisi klien dengan proses penyimpanan asset-aset statis pada device pengguna.[7]



Gambar 4. *Browser/HTTP Cache*

2. **CDN Cache**

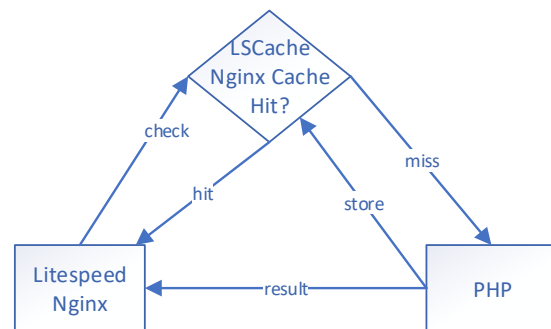
Content Delivery Network melakukan *cache* dengan menjadi Reverse Proxy didepan *web server*. Pada *CDN Cache* akan digunakan dua *CDN* yakni *Cloud flare* untuk *cache* konten statis dan *QUIC Cloud* untuk *cache* konten dinamis, dapat dilihat pada gambar 5



Gambar 5. *CDN Cache*

3. **Page Cache**

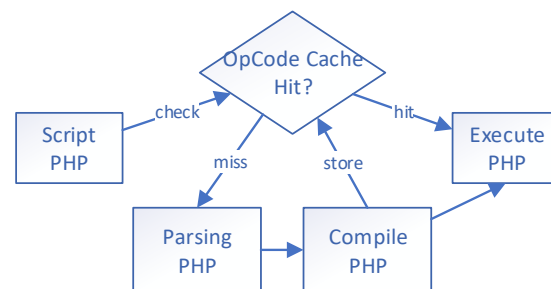
Page Cache merupakan strategi *cache* untuk mengurangi beban *application server* dan *database* karena *page cache* melewati proses eksekusi *PHP* (*parsing*, kompilasi, eksekusi) dan *query database* ke *MySQL* dengan cara meng *cache* halaman dinamis dalam bentuk halaman statis, dapat dilihat pada gambar 6.



Gambar 6. *Page Cache*

4. **OpCode Cache**

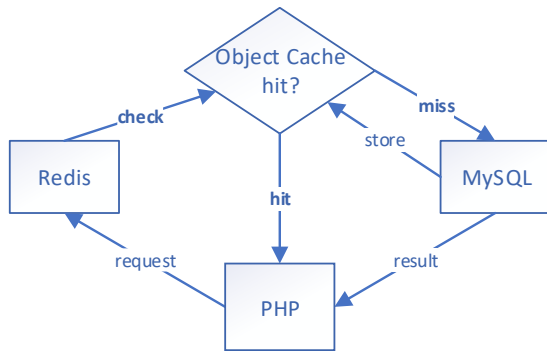
OpCode Cache mempersingkat proses waktu terjadinya eksekusi dengan melakukan dengan *cache* hasil kompilasi dari *script PHP* yang telah di *parsing* dan dikompilasi. Skema kerja *OpCode Cache* dapat dilihat pada Gambar 7.



Gambar 7. *OpCode Cache*

5. Object Cache

Object Cache atau dikenal juga *Database Cache* adalah strategi *cache* dari sisi server dengan mengurangi permintaan *query* ke *database* server dan menyimpannya menjadi objek dalam memory, dapat dilihat pada gambar 8.



Gambar 8. *Object Cache*

2.2.4. Implementation

Pada fase ini, dilakukan instalasi dan implementasi konfigurasi perangkat lunak dan Litespeed *cache*.

Terdapat tiga tahap pada fase ini yakni *Impelemtantion Log*, *Network Ready For Use (NRFU) Test*, dan *NRFU Test*.

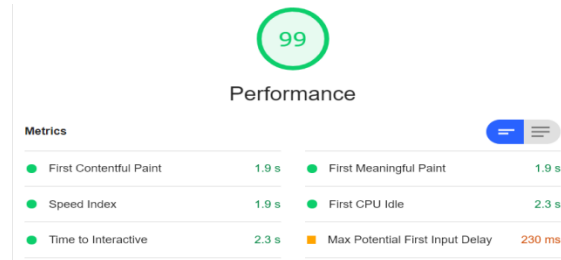
1. Implementation Log

Pada tahapan ini dilakukan langkah-langkah dan deskripsi tentang Instalasi, konfigurasi, dan proses optimasi *Litespeed Cache* pada situs www.blast.co.id.

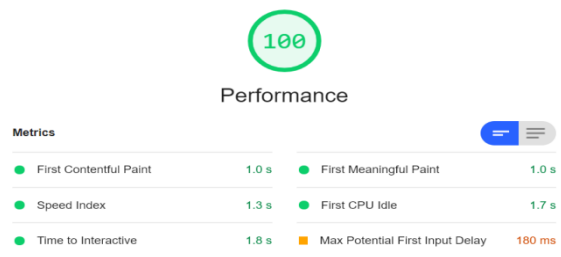
2. NRFU Testing

Selanjutnya tahapan implementasi dan optimasi *Litespeed web server* setelah diterapkannya *Litespeed Cache*. Testing meliputi :

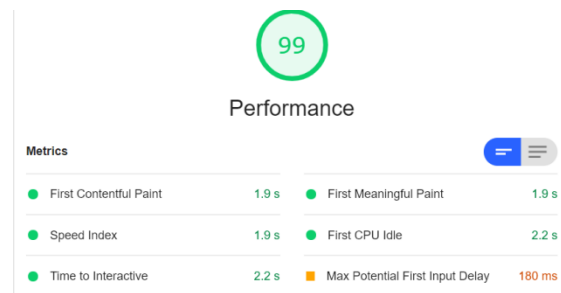
- Pengujian 1 : Mode mobile, Throtling Slow 4G, Clear Storages
 - Pengujian 2 : Mode mobile, Throtling Slow 4G
 - Pengujian 3 : Mode mobile, No Throtling, Clear Storages
 - Pengujian 4 : Mode mobile, No Throtling
 - Pengujian 5 : Mode desktop, Throtling Slow 4G, Clear Storages
 - Pengujian 6 : Mode desktop, Throtling Slow 4G
 - Pengujian 7 : Mode desktop, No Throtling, Clear Storages
 - Pengujian 8 : Mode desktop, No Throtling
- Hasil-hasil pengujian tersebut dapat dilihat pada gambar 9 sampai gambar 16.



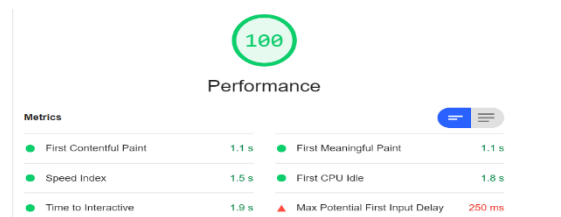
Gambar 9. Pengujian 1



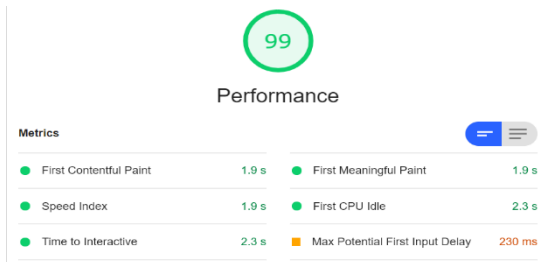
Gambar 10. Pengujian 2



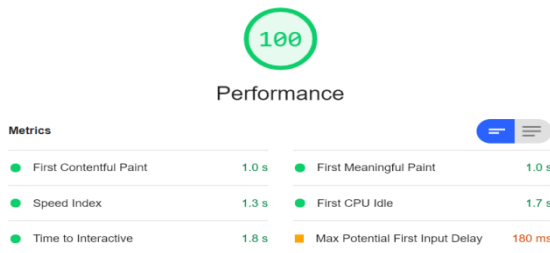
Gambar 11. Pengujian 3



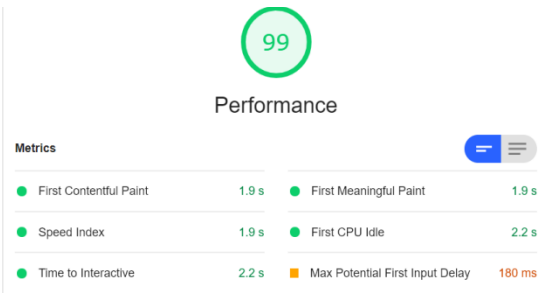
Gambar 12. Pengujian 4



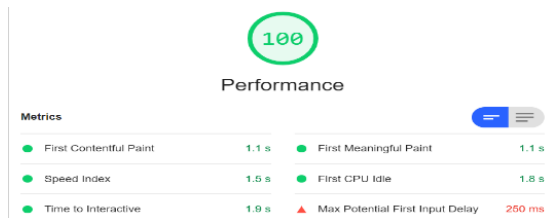
Gambar 13. Pengujian 5



Gambar 14. Pengujian 6



Gambar 15 Pengujian 7



Gambar 16 Pengujian 8

2.2.5. NRFU Testing Report

NRFU Testing Report adalah laporan data dari NRFU Test yang hasilnya dapat digunakan untuk memberikan interpretasi hasil pengujian.

1. Hasil Pengujian 1

Tabel 4. Performance metrics Report 1

Metrics	Waktu
First Contentful Paint	1,9s (cepat)
First Meaningful Paint	1,9s (cepat)
Speed Index	1,9s (cepat)
First CPU Idle	2,3s (cepat)
Time to Interactive	2,5s (cepat)
Estimated Input Latency	230ms (sedang)
Performance metrics Score	99 (cepat)

2. Hasil Pengujian 2

Tabel 5. Performance metrics Report 2

Metrics	Waktu
First Contentful Paint	1.0s (cepat)
First Meaningful Paint	1,0s (cepat)
Speed Index	1,0s (cepat)
First CPU Idle	1,0s (cepat)
Time to Interactive	1,3s (cepat)
Estimated Input Latency	180ms (sedang)
Performance metrics Score	100 (cepat)

3. Hasil Pengujian 3

Tabel 6. Performance metrics Report 3

Metrics	Waktu
First Contentful Paint	1,9s (cepat)
First Meaningful Paint	1,9s (cepat)
Speed Index	2,2s (cepat)
First CPU Idle	2,2s (cepat)
Time to Interactive	2,5s (cepat)
Estimated Input Latency	180 ms (sedang)
Performance metrics Score	99 (cepat)

4. Hasil Pengujian 4

Tabel 7. Performance metrics Report 4

Metrics	Waktu
First Contentful Paint	1.1s (cepat)
First Meaningful Paint	1,1s (cepat)
Speed Index	1,9s (cepat)
First CPU Idle	2,3s (cepat)
Time to Interactive	1,9s (cepat)
Estimated Input Latency	250 ms (sedang)
Performance metrics Score	100 (cepat)

5. Hasil Pengujian 5

Tabel 8. Performance metrics Report 5

Metrics	Waktu
First Contentful Paint	1,7s (cepat)
First Meaningful Paint	1,7s (cepat)
Speed Index	2,4s (cepat)
First CPU Idle	2,2s (cepat)
Time to Interactive	2,2s (cepat)
Estimated Input Latency	420ms (sedang)
Performance metrics Score	98 (cepat)

6. Hasil Pengujian 6

Tabel 9. Performance metrics Report 6

Metrics	Waktu
First Contentful Paint	1.1s (cepat)
First Meaningful Paint	1,1s (cepat)
Speed Index	1,6s (cepat)
First CPU Idle	2,8s (cepat)
Time to Interactive	1,9s (cepat)
Estimated Input Latency	420 ms (sedang)
Performance metrics Score	100 (cepat)

7. Hasil Pengujian 7.

Tabel 10. Performance metrics Report 7

Metrics	Waktu
First Contentful Paint	1.1s (cepat)
First Meaningful Paint	1,1s (cepat)
Speed Index	1,3s (cepat)
First CPU Idle	2,2s (cepat)
Time to Interactive	2,3s (cepat)
Estimated Input Latency	450 ms (sedang)
Performance metrics Score	100 (cepat)

8. Hasil Pengujian 8.

Tabel 11. Performance metrics Report 8

Metrics	Waktu
First Contentful Paint	1.0s (cepat)
First Meaningful Paint	1,0s (cepat)
Speed Index	1,5s (cepat)
First CPU Idle	1,6detik (cepat)
Time to Interactive	1,7s (cepat)
Estimated Input Latency	330 ms (sedang)
Performance metrics Score	100 (cepat)

2.3. Keseluruhan Hasil Pengujian .

Hasil pengujian secara keseluruhan dari setiap perbedaan *mode desktop* dan *mode mobile*, serta *throttling* dan *clear storages* dengan skor *performance metrics* berkisar diantara 99-100.

Berikut adalah skor rata-ratanya:

$$\text{Total Skor} = (99+100+99+100+98+100+100+100)$$

$$\text{Skor Rata-Rata} = 796 / 8 = 99,5$$

Percobaan diatas membuktikan telah banyak aspek-aspek dan parameter *Lighthouse Engine* yang telah lulus audit dengan skor rata-rata 99,5 dan berada pada kisaran **cepat (90-100)**. Skor *performance metrics* yang tinggi mengindikasikan situs dengan pemuatan yang relatif cepat dari persepsi *user*.

3. PENUTUP

Berdasarkan hasil penelitian yang telah dilakukan, diperoleh bahwa efisiensi dalam penggunaan *resources* dapat dicapai dengan menggunakan *web server* berbasis *event-driven* dan penerapan *cache*. Hal ini dapat dilihat dari skor *performance metrics* yang tinggi pada *web server*. Selain itu juga, optimasi *web server* dan *Litespeed Cache* dapat meningkatkan skor *performance metrics* dari *Google Pagespeed Insight API* dengan skor rata-rata 99,5. Serta skor *performance metrics* sebesar 99,5 pada situs www.blast.co.id akan berimplikasi pada meningkatnya *user experience*, *engagement*, retensi, konversi/penjualan dari pengunjung serta peringkat SEO.

DAFTAR PUSTAKA

- [1] Solikin, Imam, "Penerapan Metode PPDIOO dalam Pengembangan LAN dan WLAN," vol. 07, no. 01, hal. 65-73, 2017.
- [2] Wilkin, Seans, (2011, 5 April). Cisco's PPDIOO Network Cycle 2019.
- [3] Nguyen, VanNam, *Comparative Performance Evaluation of Web servers* , (6), 28-36.
- [4] Luthfi Muhammad, Data Mahendra, Widhi Yahya. (2018). *Perbandingan performa reverse proxy cache nginx dan varnish pada web server apache. Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(4), 1457
- [5] Mahad, Fairus Safwan., & Wan Kadir, Wahn Mohd. Nasir (2013). *Improving web server performance using two-tiered web cache. Journal of Theoretical and Applied Information Technology*, 52(3), 243-251.
- [6] Syahrul Mauluddin, Iskandar Ikbal, Agus Nursikuwagus. (2018). *Optimasi Aplikasi Penjadwalan Kuliah Menggunakan Algoritma Genetik* 2(3), 792-779

-
- [7] Manhas, Jatindes (2013). *A Study of Factors Affecting Websites Page Loading Speed for Efficient Web Performance*, (3), 32–35
- [8] Havala, Olavi. (2018). *Analyzing and Improving the Loading Performance of Large Scale Websites on Mobile Devices* , (6), 27–36
- [9] Pagespeed, Google (2018, 15 Mei). *Why Performance Matter* Dikutip 15 Mei2019: <https://developers.google.com/web/fundamentals/performance/>
- [10] *Web server*, Litespeed (2017, 16 Mei). *Litespeed Wiki* Dikutip 1 4 April 2019:https://www.litespeedtech.com/support/wiki/doku.php/litespeed_wiki