Journal of Innovative and Creativity
Journal Homepage: https://joecy.org/index.php/joecy
e-mail: journaljoecy@gmail.com

Faculty of Education
University of Pahlawan Tuanku Tambusai
ISSN 2775-771X (Online)
ISSN 2962-570X **(Printed)**
6 (1) 2026, Page: 12042-12052

Analisis Perbandingan *Client-Side Caching* dan *Server-side Caching* Terhadap File Statis untuk Kecepatan *Loading* Website Profil Desa

Shelamita Sefiani[1], I Made Suartana[2]
[1,2] Program Studi S1 Teknik Informatika, Universitas Negeri Surabaya
[1]shelamitasefiani.22018@mhs.unesa.ac.id
[2]madesuartana@unesa.ac.id

Published: Februari , 2026

ABSTRACT

Performa website merupakan faktor penting dalam penyampaian informasi secara efektif, khususnya pada website profil desa yang umumnya menyajikan konten statis. Waktu muat halaman yang lambat dapat menurunkan kenyamanan pengguna serta menghambat akses informasi. Penelitian ini bertujuan untuk menganalisis dan membandingkan efektivitas penerapan Client-side caching, Server-side caching, serta kombinasi keduanya (hybrid caching) dalam meningkatkan kecepatan loading website profil desa. Metode penelitian yang digunakan adalah eksperimen dengan menerapkan empat skenario caching pada website profil desa berbasis Wordpress. Evaluasi performa dilakukan menggunakan Google Lighthouse dan Chrome DevTools dengan parameter First Contentful Paint (FCP), Largest Contentful Paint (LCP), Total Blocking Time (TBT), Cumulative Layout Shift (CLS), Speed Index (SI), Cache Hit Ratio (CHR), trafik pemuatan data, serta perilaku pembaruan konten pada kondisi kunjungan pertama dan kunjungan ulang. Hasil penelitian menunjukkan bahwa hybrid caching memberikan peningkatan performa paling signifikan dengan rata-rata CHR sebesar 93,67% pada kunjungan ulang, penurunan transfer data hingga 99%, serta perbaikan nilai FCP dan LCP secara konsisten. Client-side caching efektif meningkatkan performa pada kunjungan berulang, sedangkan Server-side caching mampu menjaga stabilitas performa pada berbagai kondisi akses. Dengan demikian, penerapan kombinasi Client-side dan Server-side caching merupakan strategi paling optimal dalam meningkatkan kecepatan loading, stabilitas tata letak, dan efisiensi sumber daya server pada website profil desa.

Kata Kunci : Client-side caching, Server-side caching, hybrid caching, performa website, kecepatan loading

© The Author(s). 2021 Open Access This article is licensed under a Creative Commons Atibution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0

PENDAHULUAN

Perkembangan teknologi informasi mendorong pemerintah desa untuk memanfaatkan website sebagai media utama dalam penyampaian informasi kepada masyarakat. Website profil desa berperan sebagai sarana publikasi data administratif, transparansi kegiatan, serta layanan informasi publik yang dapat diakses secara luas dan berkelanjutan [1], [2]. Website jenis ini umumnya menyajikan konten statis sehingga dituntut memiliki performa yang cepat dan stabil agar informasi dapat diterima pengguna secara efektif [3].

Kecepatan *loading* website merupakan salah satu indikator utama kualitas performa yang berpengaruh langsung terhadap kenyamanan dan kepuasan pengguna [4], [5]. Website dengan waktu muat yang lambat berpotensi meningkatkan bounce rate serta menurunkan tingkat kepercayaan pengguna terhadap layanan digital yang disediakan [6]. Pada website profil desa, permasalahan performa sering muncul akibat keterbatasan sumber daya server serta pengelolaan konten yang belum dioptimalkan, khususnya pada website berbasis *Content Management System* (CMS) seperti *Wordpress* [7], [8].

Salah satu teknik optimasi yang umum digunakan untuk meningkatkan performa website adalah *caching*. *Caching* bekerja dengan menyimpan data atau halaman website dalam penyimpanan sementara sehingga permintaan berikutnya dapat dilayani tanpa perlu pemrosesan ulang di sisi server [9]. Penerapan *caching* yang tepat terbukti mampu menurunkan waktu respons server, mengurangi beban pemrosesan, serta menghemat penggunaan bandwidth [10], [11].

Caching dapat diterapkan melalui beberapa pendekatan, yaitu *Client-side caching*, *Server-side caching*, dan kombinasi keduanya yang dikenal sebagai *hybrid caching*. *Client-side caching* memanfaatkan penyimpanan pada browser pengguna untuk mempercepat akses ulang halaman website [12], sedangkan *Server-side caching* berfokus pada penyimpanan hasil pemrosesan halaman di sisi server untuk meningkatkan efisiensi layanan pada setiap permintaan [13]. Kombinasi kedua pendekatan tersebut dinilai mampu memberikan peningkatan performa yang lebih optimal, terutama pada website dengan pola akses berulang [14].

Meskipun berbagai penelitian telah membahas penerapan *caching* dalam optimasi performa website, sebagian besar masih mengkaji *Client-side* dan *Server-side caching* secara terpisah atau tidak membandingkannya secara langsung pada website dengan karakteristik konten statis seperti website profil desa [15], [16]. Selain itu, evaluasi performa sering kali belum mempertimbangkan perbedaan kondisi kunjungan pertama dan kunjungan ulang secara komprehensif [17].

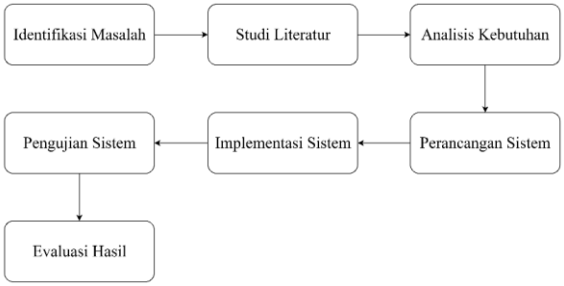
Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk menganalisis dan membandingkan efektivitas *Client-side caching, Server-side caching,* serta *hybrid caching* dalam meningkatkan kecepatan *loading* dan stabilitas performa website profil desa. Evaluasi dilakukan melalui pengujian eksperimental menggunakan metrik performa website seperti *First Contentful Paint* (FCP), *Largest Contentful Paint* (LCP), *Total Blocking Time* (TBT), *Cumulative Layout Shift* (CLS), *Speed Index* (SI), serta *Cache Hit Ratio* (CHR) [18]–[21]. Hasil penelitian ini diharapkan dapat menjadi acuan praktis dalam menentukan strategi *caching* yang paling optimal untuk website profil desa maupun website berbasis konten statis lainnya.

METODE PENELITIAN

Penelitian ini menggunakan metode eksperimen komparatif yang bertujuan untuk menganalisis dan membandingkan pengaruh penerapan *Client-side caching*, *Server-side caching*, serta kombinasi keduanya (*hybrid caching*) terhadap performa website profil desa.

A. Alur Penelitian

Tahapan penelitian disusun secara sistematis untuk memastikan proses penelitian berjalan terarah dan sesuai dengan tujuan yang telah ditetapkan. Tahap awal penelitian diawali dengan identifikasi permasalahan terkait performa website profil desa, khususnya pada aspek kecepatan *loading* halaman dan efisiensi pemanfaatan sumber daya server. Selanjutnya, dilakukan perancangan skenario pengujian *caching* yang mencakup kondisi tanpa *caching* dan kondisi dengan berbagai metode *caching*. Tahap berikutnya meliputi implementasi mekanisme *caching* pada website uji, pelaksanaan pengujian performa, pengumpulan data hasil pengujian, serta analisis data untuk menarik kesimpulan mengenai metode *caching* yang paling efektif, seperti yang diilustrasikan pada Gbr 1.



Gambar. 1 Diagram Alir Penelitian

B. Perancangan Sistem

Perancangan sistem dilakukan dengan menerapkan empat skenario pengujian, yaitu:

1. website tanpa penerapan *caching* sebagai kondisi awal (*baseline*),
2. website dengan penerapan *Client-side caching*,
3. website dengan penerapan *Server-side caching*, dan
4. website dengan penerapan *hybrid caching* yang mengombinasikan *Client-side* dan *Server-side caching*.

Seluruh skenario pengujian dijalankan pada lingkungan sistem yang sama, baik dari sisi konfigurasi server, jaringan, maupun perangkat pengujian, untuk meminimalkan pengaruh faktor eksternal. Dengan desain ini, perbandingan performa antar skenario dapat dilakukan secara objektif dan terukur.

C. Implementasi Sistem dan Caching

Website profil desa dikembangkan menggunakan CMS *Wordpress* dengan tema dan struktur halaman yang sama pada seluruh skenario pengujian. Konten yang digunakan bersifat statis agar karakteristik website sesuai dengan kondisi umum website profil desa.

Client-side caching diterapkan dengan memanfaatkan mekanisme penyimpanan *cache* pada browser pengguna melalui pengaturan *cache-control* dan plugin *W3 Total Cache* dengan hanya mengaktifkan *browser cache*, sehingga aset statis seperti file HTML, CSS, *JavaScript,* dan *image* dapat digunakan kembali pada kunjungan berikutnya.

Server-side caching diterapkan dengan menggunakan plugin *W3 Total Cache* juga, namun hanya mengaktifkan *page cache* dan *object cache* kemudian mengkonfigurasikan dengan CDN di *Cloudflare*, sehingga hasil permintaan halaman selanjutnya dapat dilayani tanpa menjalankan proses dinamis *Wordpress* secara berulang.

Pada skenario *hybrid caching*, kedua mekanisme tersebut diterapkan secara bersamaan untuk memperoleh efisiensi pemrosesan di sisi server serta peningkatan kecepatan akses di sisi pengguna.

D. Skenario dan Kondisi Pengujian

Pengujian performa dilakukan pada dua kondisi akses, yaitu kunjungan pertama (*first visit*) dan kunjungan ulang (*repeat visit*). Pada kondisi *first visit*, *cache browser* dikosongkan untuk mensimulasikan akses awal pengguna terhadap website. Pada kondisi *repeat visit*, *cache browser* dibiarkan aktif untuk mengevaluasi efektivitas mekanisme *caching* dalam mempercepat akses halaman.

Setiap skenario *caching* diuji secara berulang dengan jumlah pengujian yang sama untuk memastikan hasil yang diperoleh konsisten dan representatif. Seluruh pengujian dilakukan pada waktu yang relatif sama guna meminimalkan variasi kondisi jaringan.

E. Pengujian Sistem

Parameter pengujian dalam penelitian ini digunakan untuk mengukur performa website profil desa pada setiap skenario *caching* yang diterapkan. Pengujian difokuskan pada aspek kecepatan *loading*, stabilitas tampilan, responsivitas interaksi, serta efisiensi pemuatan data (*traffic loading*). Parameter yang digunakan meliputi:

1. *First Contentful Paint* (FCP)
2. *Largest Contentful Paint* (LCP)
3. *Total Blocking Time* (TBT)
4. *Cumulative Layout Shift* (CLS)
5. *Speed Index* (SI)
6. *Cache Hit Ratio* (CHR)
7. *Traffic Loading* (Transfer Data)

Seluruh pengujian dilaksanakan pada lingkungan yang terkontrol untuk memastikan hasil yang akurat, terukur, dan dapat direplikasi. Adapun rincian variabel lingkungan dan konfigurasi pengujian adalah sebagai berikut:

1. Perangkat menggunakan desktop dengan *browser Google Chrome* versi terbaru dengan mode *Incognito*.
2. Jaringan menggunakan simulasi *Fast 4G* melalui fitur *Network Throttling* pada *Chrome DevTools* dengan koneksi WiFi UNESA sebagai jaringan utama.
3. Lokasi pengujian dilakukan di Surabaya, Indonesia, dengan server utama (*origin server*) yang juga berlokasi domestik sehingga faktor latensi lokal dapat diamati secara presisi.
4. *Tools* Pengujian
   a. *Chrome DevTools* (*Network Inspect, Performance Tab).*
   b. *Lighthouse* untuk pengukuran metrik *Web Vital.*
   c. *Network* HAR *Export* untuk analisis *cache.*
   d. *Python Library* (JSON) untuk perhitungan *Cache Hit Ratio* (CHR) dari file HAR.

Target performa yang digunakan dalam pengujian mengacu pada standar umum pengujian performa web sederhana dan dianjurkan oleh komunitas developer internasional, termasuk *Chrome for Developers, web.dev,* dan penyedia CDN seperti *Cloudflare*. Penjelasan setiap target performa pengujian dapat dilihat pada TABEL 1.

Tabel I
Penjelasan Target Performa Pengujian

Metrik	Target	Penjelasan
FCP	<2.5 detik	Sesuai rekomendasi dari Chrome *for Developers*, nilai FCP < 2.5 detik dianggap cukup baik untuk website non-komersial dengan beban rendah [22].
LCP	<2.5 detik	Berdasarkan *web.dev*, *Google* menyarankan LCP ideal < 2.5 detik [20].
TBT	<300 ms	Berdasarkan *web.dev, Google* menyarankan TBT < 200ms, namun ≤ 300ms masih tergolong baik untuk website yang tidak terlalu interaktif [19].
CLS	<0.1	Menurut [21], skor CLS yang baik untuk pengalaman pengguna adalah kurang dari 0,1.
Speed Index	<6.0	Menurut *DebugBear, Speed Index* dengan nilai ≤ 6.0 masih masuk kategori *acceptable* untuk website ringan [21].
Cache Hit Ratio	>70%	*Cloudflare* menyatakan pada artikel resmi mereka bahwa *cache* hit ratio ≥ 60% sudah baik, dengan 70–90% sebagai nilai optimal [23].

F. Analisis Perbandingan Hasil Pengujian

Tahap analisis dilakukan dengan menghitung nilai rata-rata setiap metrik performa pada kondisi *First Visit* dan *Repeat Visit* untuk masing-masing skenario *caching*, yaitu tanpa *caching*, *Client-side caching*, *Server-side caching*, dan *hybrid caching*.

Nilai rata-rata yang diperoleh kemudian dibandingkan antar skenario untuk mengidentifikasi perbedaan performa yang dihasilkan oleh masing-masing metode *caching*. Selanjutnya dilakukan perhitungan persentase peningkatan performa, baik dari sisi kecepatan loading, stabilitas tampilan, maupun efisiensi trafik pemuatan data.

Hasil perbandingan tersebut digunakan untuk:

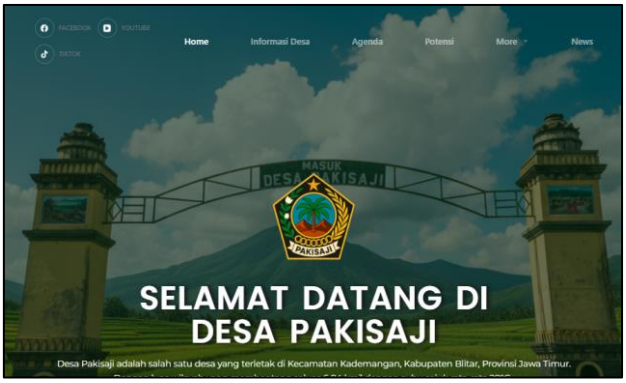
1. Menentukan kontribusi masing-masing teknik *caching* terhadap peningkatan performa website.
2. Mengidentifikasi metode *caching* yang memberikan peningkatan paling signifikan pada kondisi *first visit* dan *repeat visit*.
3. Menyimpulkan strategi *caching* yang paling optimal untuk diterapkan pada website profil desa berbasis konten statis.

HASIL PENELITIAN DAN PEMBAHASAN

A. Hasil Implementasi

Implementasi dilakukan melalui empat skenario utama, yaitu *cache* kosong, *Client-side Caching, Server-side Caching*, dan kombinasi keduanya (*Hybrid Caching)*. Dalam menyusun konten website, peneliti sengaja menggunakan banyak file statis seperti gambar, skrip, dan CSS pada halaman utama (*homepage*). Hal ini bertujuan untuk memusatkan proses pengujian pada satu halaman yang memiliki konten statis

yang padat, sehingga efek *caching* dapat diukur dengan lebih jelas. Tampilan halaman utama yang digunakan dalam pengujian dapat dilihat pada Gbr..2.



Gambar 2. Tampilan *Homepage Website* Desa

B. Hasil Pengujian Performa Website Menggunakan Lighthouse

Pengujian performa website dilakukan menggunakan *Google Lighthouse* pada empat skenario, yaitu tanpa *caching (no cache)*, *Client-side caching*, *Server-side caching*, dan *hybrid caching*. Setiap skenario diuji pada dua kondisi akses, yaitu *first visit* dan *repeat visit*, dengan lima kali iterasi pengujian dan hasil yang digunakan merupakan nilai rata-rata dari seluruh iterasi.

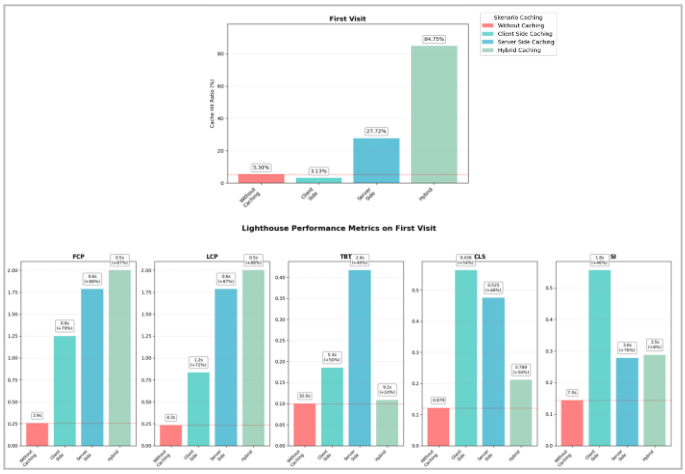
Tabel II
Hasil Rata-Rata Pengujian Dengan *Lighthouse* Pada *First Visit*

Skenario	FCP (s)	LCP (s)	TBT (s)	CLS	SI (s)	CHR
No cache	3.9	4.3	10	0.879	6.96	5.30%
Client-side	0.8	1.2	5.4	0.4362	1.8	3.13%
Server-side	0.56	0.56	2.4	0.525	3.6	26.52%
Hybrid	0.5	0.5	9.2	0.7888	3.48	84.75%

Hasil pengujian performa website untuk kondisi *first visit* dapat dilihat pada TABEL II. Berdasarkan tabel tersebut, terlihat bahwa penerapan *Client-side caching* pada *first visit* belum menunjukkan peningkatan performa yang signifikan. Hal ini disebabkan oleh proses inisialisasi *cache* (*cache warming*), di mana browser masih harus mengunduh sebagian besar resource untuk pertama kali. Kondisi ini sejalan dengan rendahnya nilai *Cache* Hit Ratio pada skenario ini.

Sebaliknya, *Server-side caching* mampu memberikan peningkatan performa yang lebih baik pada *first visit* karena halaman website dapat disajikan dari *cache* server tanpa melalui proses pemrosesan dinamis *Wordpress* secara penuh. Dampak ini terlihat pada penurunan waktu pemuatan halaman dan perbaikan stabilitas tampilan dibandingkan kondisi tanpa *caching*.

Performa terbaik pada *first visit* diperoleh pada skenario *hybrid caching*, di mana kombinasi *cache* di sisi klien dan server memungkinkan sebagian besar resource dilayani dari *cache*. Hal ini tercermin dari nilai CHR sebesar 84,75% serta peningkatan signifikan pada metrik Lighthouse, yang menunjukkan bahwa pendekatan *hybrid* mampu mengatasi keterbatasan masing-masing metode *caching* secara individual. Grafik perbandingan hasil pengujian *first visit* dapat dilihat pada Gambar. 3.



Gbr. 3 Grafik Hasil Pengujian dengan *Lighthouse* pada *First Visit*

C. Analisis Perbandingan Performa Lighthouse pada Repeat Visit

Pada kondisi *repeat visit*, seluruh skenario *caching* menunjukkan peningkatan performa dibandingkan *first visit*. Peningkatan ini terjadi karena *cache* telah terisi baik di sisi klien maupun server, sehingga sebagian besar *resource* tidak perlu dimuat ulang dari *origin server*.

Tabel III

Hasil Rata-Rata Pengujian Dengan *Lighthouse* Pada *Repeat Visit*

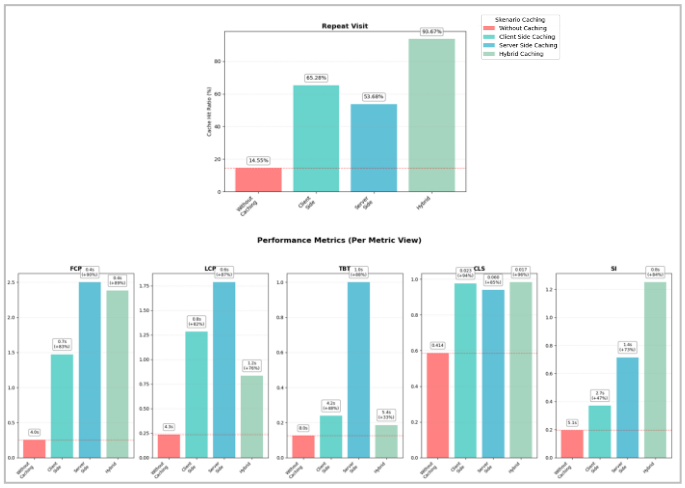
Skenario	FCP (s)	LCP (s)	TBT (s)	CLS	SI (s)	CHR
No cache	3.96	4.26	8	0.4138	5.14	14.55%
Client-side	0.68	0.78	4.2	0.0232	2.7	65.28%
Server-side	0.4	0.56	1.0	0.0602	1.4	53.68%
Hybrid	0.42	1.02	5.4	0.017	0.8	93.67%

Hasil pengujian untuk kondisi *repeat visit* disajikan pada TABEL III. Dari tabel tersebut dapat diamati bahwa pada skenario tanpa *caching*, nilai CHR meningkat menjadi 14,55% akibat *cache* bawaan browser. Namun, peningkatan ini masih belum cukup untuk menghasilkan performa optimal, yang terlihat dari nilai FCP dan LCP yang relatif tinggi dibandingkan skenario *caching* lainnya.

Client-side caching menunjukkan peningkatan performa yang sangat signifikan pada *repeat visit*, dengan CHR mencapai 65,28%. Tingginya rasio ini menunjukkan bahwa mayoritas resource berhasil dilayani dari *cache browser* menunjukkan efektivitas mekanisme *cache* di sisi klien setelah tahap inisialisasi selesai.

Server-side caching juga mengalami peningkatan performa pada *repeat visit*, dengan CHR sebesar 53,68%. Meskipun lebih rendah dibandingkan *Client-side caching*, pendekatan ini memberikan performa yang lebih konsisten karena tidak bergantung pada kondisi *cache* di sisi pengguna.

Skenario *hybrid caching* kembali menunjukkan performa paling optimal dengan CHR mencapai 93,67%. Nilai ini menunjukkan bahwa hampir seluruh permintaan dapat dilayani dari *cache*, sehingga waktu pemuatan halaman menjadi sangat cepat dan stabil. Temuan ini menegaskan bahwa kombinasi *cache* klien dan server merupakan strategi paling efektif untuk meningkatkan performa website pada kunjungan ulang. Grafik perbandingan hasil pengujian *repeat visit* dapat dilihat pada Gambar. 4.



Gambar. 4 Grafik Hasil Pengujian dengan *Lighthouse* pada *Repeat Visit*

D. Perbandingan Efektivitas Antara Metode Caching

Berdasarkan hasil pengujian performa dan CHR, dapat disimpulkan bahwa *hybrid caching* merupakan strategi paling efektif dalam meningkatkan kecepatan *loading* website profil desa. *Client-side caching* menunjukkan keunggulan pada kunjungan ulang, sementara *Server-side caching* memberikan performa yang lebih konsisten pada berbagai kondisi akses. Namun, kombinasi kedua metode tersebut mampu menghasilkan performa optimal secara menyeluruh, baik dari sisi kecepatan loading, stabilitas tampilan, maupun efisiensi pemanfaatan sumber daya server.

E. Analisis Perbandingan Hasil Pengujian Traffic Loading

Analisis *traffic loading* dilakukan untuk mengetahui pengaruh penerapan mekanisme *caching* terhadap jumlah data yang ditransfer selama proses pemuatan halaman website profil desa. Parameter *traffic loading* diperoleh dari hasil pengujian menggunakan *Chrome DevTools Network Inspect* dengan menganalisis total request, *cache* hit, dan *cache* miss pada setiap skenario *caching*, baik pada kondisi *first visit* maupun *repeat visit*. Nilai yang digunakan dalam analisis merupakan nilai rata-rata dari lima kali pengujian, sebagaimana tercantum pada tabel hasil pengujian.

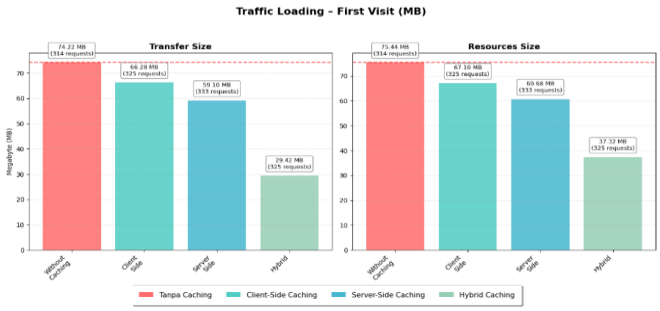
Tabel IV
Perbandingan Hasil Pengujian *Traffic Loading* Pada *First Visit*

Skenario	Request	Transferred	Resources
No cache	314.2	74.22 mb	75.44 mb
Client-side	324.8	66.28 mb	59.1 mb
Server-side	333	59.1 mb	60.69 mb
Hybrid	325.4	29.42 mb	37.32 mb

Berdasarkan hasil pengujian *traffic loading* yang disajikan pada TABEL IV untuk kondisi *first visit*, penerapan teknik *caching* menunjukkan pengaruh yang signifikan terhadap efisiensi transfer data dan pengelolaan sumber daya website. Pada skenario tanpa *caching*, total data yang ditransfer mencapai 74,22 MB dengan total sumber daya sebesar 75,44 MB. Penerapan *Client-side caching* menurunkan ukuran transfer menjadi 66,28 MB atau berkurang sekitar 11%, dengan total sumber daya turun menjadi 67,10 MB. Penurunan ini menunjukkan bahwa meskipun pada *first visit cache browser* belum terbentuk sepenuhnya, mekanisme *caching* mulai berkontribusi dalam mengurangi trafik data.

Efisiensi yang lebih besar ditunjukkan oleh *Server-side caching* dengan ukuran transfer sebesar 59,10 MB atau turun sekitar 20% dibandingkan tanpa *caching*, serta total sumber daya sebesar 60,68 MB. Skenario

hybrid caching memberikan hasil paling optimal dengan ukuran transfer hanya 29,42 MB atau berkurang sekitar 60%, dan total sumber daya turun menjadi 37,32 MB atau sekitar 51%. Hasil ini menunjukkan bahwa kombinasi *Client-side caching* dan *Server-side caching* mampu secara signifikan mengurangi konsumsi bandwidth dan mempercepat proses *loading* halaman, bahkan pada kunjungan pertama. Grafik perbandingan hasil pengujian *traffic loading* pada *first visit* dapat dilihat pada Gbr. 5.



Gambar. 5 Grafik Perbandingan Hasil Pengujian *Traffic Loading* pada *First Visit*

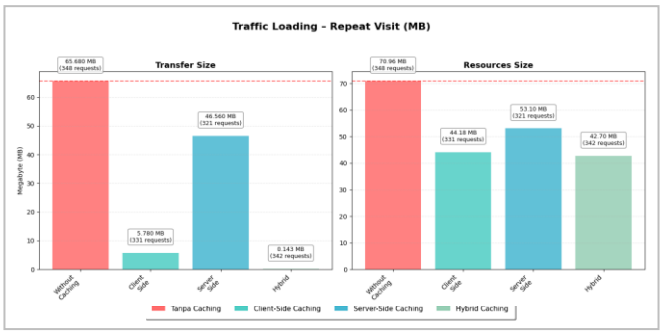
Tabel V
Perbandingan Hasil Pengujian *Traffic Loading* Pada *Repeat Visit*

Skenario	Request	Transferred	Resources
No cache	347.8	65.68 mb	70.96 mb
Client-side	330.8	5.78 mb	44.18 mb
Server-side	320.6	46.56 mb	53.1 mb
Hybrid	342.2	0.14346 mb	42.7 mb

Berdasarkan hasil pengujian *traffic loading* yang disajikan pada TABEL V untuk kondisi *repeat visit*, penerapan teknik *caching* menunjukkan pengaruh yang sangat signifikan terhadap efisiensi transfer data dan pengelolaan sumber daya website.

Pada skenario tanpa *caching*, total data yang ditransfer mencapai 65,68 MB dengan total sumber daya sebesar 70,96 MB. Penerapan *Client-side caching* menunjukkan penurunan yang sangat signifikan, dengan ukuran transfer hanya 5,78 MB atau berkurang sekitar 91% dibandingkan tanpa *caching*, serta total sumber daya turun menjadi 44,18 MB. Sedangkan pada skenario *Server-side caching*, ukuran transfer menurun menjadi 46,56 MB atau sekitar 29% lebih rendah dibandingkan tanpa *caching*, dengan total sumber daya sebesar 53,10 MB. Efisiensi paling tinggi dicapai oleh skenario *hybrid caching* dengan ukuran transfer hanya sebesar 0,143 MB atau hampir seluruh trafik data berhasil ditekan hingga mendekati nol, serta total sumber daya turun menjadi 42,70 MB. Hasil ini mengindikasikan bahwa hampir seluruh aset website telah tersedia baik di *cache browser* maupun *cache server*, sehingga proses *loading* halaman menjadi sangat ringan dan cepat. Grafik perbandingan hasil pengujian *traffic loading* pada *repeat visit* dapat dilihat pada Gbr. 6.

Gbr. 6 Grafik Perbandingan Hasil Pengujian *Traffic Loading* Pada *Repeat Visit*



G. *Analisis Perbandingan Hasil Pengujian Single User dan Multi User*

Analisis perbandingan pengujian single user dan multi user dilakukan untuk mengevaluasi ketahanan performa website profil desa terhadap perbedaan beban akses pengguna. Pengujian single user merepresentasikan kondisi ketika website diakses oleh satu pengguna dalam satu waktu, sedangkan pengujian multi user merepresentasikan kondisi ketika beberapa pengguna mengakses website secara bersamaan atau dalam interval waktu yang berdekatan, sehingga berpotensi meningkatkan beban server dan trafik jaringan.

Pada kondisi single user, performa website cenderung lebih stabil karena beban pemrosesan server dan trafik jaringan relatif rendah. Pada skenario tanpa *caching*, meskipun hanya diakses oleh satu pengguna, waktu pemuatan halaman tetap relatif tinggi karena seluruh *resource* harus diproses dan dikirim langsung dari *origin server*. Penerapan *Client-side caching* pada kondisi ini mulai menunjukkan peningkatan performa pada *repeat visit*, sementara *Server-side caching* dan *hybrid caching* mampu memberikan waktu respon yang lebih cepat dan konsisten sejak *first visit*.

Tabel VI
Perbandingan Hasil Pengujian *Single User* Dan *Multi User*

Parameter	***Single User***	***Multi User* 1**	***Multi User* 2**
Request	315.2	313.6	319
Transferred	0.18 mb	0.223 mb	0.2124 mb
Resources	35.04 mb	33.38 mb	32.06 mb
CHR	87.446%	84.58%	85.898%
FCP	0.3 s	0.34 s	0.3 s
LCP	0.9 s	0.58 s	0.52 s
TBT	36 ms	0 ms	0 ms
CLS	0.016	0.0204	0.032
SI	0.6 s	0.82 s	0.4 s

Berdasarkan hasil pengujian *single user* dan *multi user* yang disajikan pada TABEL VI untuk kondisi *multi user*, perbedaan performa antara skenario *caching* menjadi lebih jelas. Tanpa penerapan *caching*, peningkatan jumlah pengguna menyebabkan beban server meningkat secara signifikan, yang berdampak pada meningkatnya waktu *loading* halaman, khususnya pada metrik *First* Contentful Paint (FCP), *Largest Contentful Paint* (LCP), serta meningkatnya trafik *loading* akibat banyaknya permintaan resource yang harus dilayani secara bersamaan.

Penerapan *Client-side caching* pada kondisi multi user mampu mengurangi beban server pada kunjungan ulang karena sebagian besar resource statis dapat dilayani langsung dari *cache browser* masing-masing pengguna. Namun, pada kunjungan awal (*first visit*), *Client-side caching* belum memberikan dampak

signifikan karena *cache* belum terbentuk, sehingga server masih harus melayani sebagian besar permintaan dari pengguna secara langsung.

Sebaliknya, *Server-side caching* menunjukkan keunggulan yang lebih jelas pada kondisi multi user, karena halaman website yang telah disimpan dalam *cache* server dapat disajikan secara cepat kepada banyak pengguna tanpa perlu pemrosesan ulang. Hal ini menyebabkan waktu respon server lebih stabil serta mengurangi lonjakan trafik *loading* meskipun terjadi peningkatan jumlah pengguna yang mengakses website secara bersamaan.

Performa terbaik pada pengujian multi user diperoleh melalui penerapan *hybrid caching*. Kombinasi *cache* di sisi klien dan server memungkinkan distribusi beban yang lebih seimbang, di mana *Server-side caching* menangani permintaan awal dan akses bersamaan, sementara *Client-side caching* mengurangi permintaan ulang dari pengguna yang sama. Kondisi ini tercermin dari tingginya nilai *Cache* Hit Ratio (CHR), rendahnya trafik loading, serta konsistensi metrik Lighthouse baik pada kondisi single user maupun multi user.

Berdasarkan hasil analisis perbandingan tersebut, dapat disimpulkan bahwa *hybrid caching* merupakan strategi paling optimal untuk menghadapi variasi jumlah pengguna, baik pada kondisi single user maupun multi user. *Client-side caching* efektif pada kunjungan ulang pengguna individual, *Server-side caching* unggul dalam menangani beban akses simultan, dan kombinasi keduanya memberikan performa paling stabil, efisien, serta scalable untuk website profil desa berbasis konten statis.

KESIMPULAN

Berdasarkan hasil implementasi dan pengujian teknik *caching* pada website profil Desa Pakisaji, dapat disimpulkan bahwa penerapan *caching* mampu meningkatkan performa website secara signifikan dibandingkan dengan kondisi tanpa *caching*. Seluruh strategi yang diuji, yaitu *Client-side caching*, *Server-side caching*, dan *hybrid caching*, menunjukkan perbaikan yang konsisten pada metrik performa Lighthouse, meliputi *First* Contentful Paint (FCP), *Largest Contentful Paint* (LCP), *Speed Index* (SI), dan *Cumulative Layout Shift* (CLS), serta mampu menurunkan jumlah data yang ditransfer, khususnya pada kondisi *repeat visit*. Hal ini menunjukkan bahwa *caching* berperan penting dalam mempercepat waktu muat halaman dan meningkatkan stabilitas tampilan website.

Hasil penelitian juga menunjukkan bahwa strategi *hybrid caching* merupakan pendekatan yang paling efektif secara keseluruhan. Kombinasi antara *Client-side caching* dan *Server-side caching* yang didukung oleh layanan CDN Cloudflare mampu menghasilkan nilai *Cache* Hit Ratio (CHR) tertinggi dan performa terbaik pada berbagai skenario pengujian. Pada kondisi *repeat visit*, strategi ini mencapai CHR sebesar 93,67%, *Speed Index* sebesar 0,8 detik, serta penurunan trafik data hingga 99,8%. Temuan tersebut mengindikasikan bahwa sebagian besar permintaan dapat dilayani melalui lapisan *cache*, sehingga tidak hanya meningkatkan pengalaman pengguna, tetapi juga secara signifikan mengurangi beban pada origin server.

Selain itu, setiap strategi *caching* memiliki karakteristik dan keunggulan masing-masing. *Client-side caching* terbukti sangat efektif bagi pengguna yang melakukan kunjungan berulang, dengan peningkatan CHR yang signifikan serta perbaikan pada metrik FCP dan CLS. *Server-side caching* memberikan performa yang lebih konsisten untuk berbagai jenis kunjungan dan membantu mengurangi beban origin server, terutama bagi pengguna yang tersebar secara geografis melalui dukungan CDN. Sementara itu, *hybrid caching* mampu mengoptimalkan keunggulan kedua pendekatan tersebut dengan menghasilkan efisiensi bandwidth dan nilai CHR tertinggi, meskipun memerlukan konfigurasi sistem yang lebih kompleks.

Penerapan teknik *caching* juga berdampak langsung terhadap efisiensi pengelolaan file statis pada website. File statis seperti *image,* CSS, dan *JavaScript* dapat disajikan dari penyimpanan sementara pada browser atau server terdekat, sehingga dapat mengurangi ketergantungan terhadap origin server, penggunaan bandwidth menjadi lebih efisien, dan waktu akses halaman menjadi lebih cepat. Namun demikian, penerapan *caching* perlu disertai dengan mekanisme *cache* invalidation yang tepat, seperti *cache* purging, agar pembaruan konten dapat ditampilkan secara akurat kepada pengguna.

Lebih lanjut, hasil pengujian pada skenario single user dan multi user menunjukkan bahwa *caching* mampu meningkatkan skalabilitas dan ketahanan website terhadap peningkatan beban akses. Website tetap menunjukkan performa yang stabil dan responsif meskipun diakses oleh beberapa pengguna secara bersamaan, dengan nilai CHR yang tetap tinggi serta metrik utama seperti FCP dan LCP yang tidak

mengalami penurunan signifikan. Secara keseluruhan, penelitian ini menunjukkan adanya hubungan yang kuat antara nilai *Cache* Hit Ratio (CHR), peningkatan metrik performa Lighthouse, dan penurunan jumlah data yang ditransfer, dimana semakin tinggi nilai CHR maka semakin efisien dan cepat proses pemuatan halaman website.

REFERENSI

[1] P. Airlangga, H. Harianto, and A. Hammami, "Pembuatan dan Pelatihan Pengoperasian Website Desa Agrowisata Gondangmanis," *Jumat Informatika: Jurnal Pengabdian Masyarakat*, pp. 9–12, 2020.

[2] R. Indahsari and M. Son Muarie, "Sistem Informasi Profil Desa Berbasis Web (Studi Kasus: Desa Purwosari)," in *Prosiding Seminar Nasional Sains dan Teknologi Terapan*, pp. 33–41, 2020.

[3] G. Agrawal, D. Kumar, and M. Singh, "Assessing the Usability, Accessibility, and Mobile Readiness of E-Government Websites: A Case Study in India," *Universal Access in the Information Society*, pp. 737–748, 2022.

[4] S. Sounders, "High Performance Websites," *Communications of the ACM*, vol. 51, no. 12, pp. 36–41, 2008.

[5] Google, "Mobile Site Load Time Statistics," 2025. [Online]. Available:

[6] D. C. Haan, "Evaluasi Web Performance Pemerintah Daerah Kabupaten Magelang dalam Rangka Mendukung Penerapan E-Government untuk Pelayanan Publik," *Jurnal Ilmiah Multimedia dan Komunikasi*, pp. 23–34, 2020.

[7] M. Hembram, "Comparative Study of OpenSource Content Management Systems (CMS) in Digital Era," *Asian Journal of Electrical Sciences*, vol. 11, no. 1, pp. 12–16, 2022.

[8] O. Kuosmanen, *Evaluating Wordpress Optimization Plugins and Techniques*, LUT University, 2025.

[9] A. S. Widhiarta and F. W. Wibowo, "Optimasi Kinerja Web Menggunakan Application-Level *Cache* di Sisi Server dan Browser," *Respati*, pp. 63–74, 2018.

[10] I. Malavolta *et al.*, "Evaluating the Impact of *Caching* on the Energy Consumption and Performance of Progressive Web Apps," in *Proc. IEEE/ACM 7th Int. Conf. on Mobile Software Engineering and Systems*, pp. 109–119, 2020.

[11] M. Alam, T. Rahman, and S. Hasan, "Performance Enhancement of Web Applications Using Cloudflare CDN and Security Layer Integration," *Journal of Web Engineering*, vol. 22, no. 4, pp. 880–897, 2023.

[12] A. Goel, V. Ruamviboonsuk, R. Netravali, and H. V. Madhyastha, "Rethinking *Client-side Caching* for the Mobile Web," in *Proc. 22nd Int. Workshop on Mobile Computing Systems and Applications*, pp. 112–118, 2021.

[13] O. Södermark, *Client-side Caching: Reducing Server Load and Latency in a Network Traffic Analysis Tool*, Diva Portal, 2023.

[14] J. Chen *et al.*, "Darwin: Flexible Learning-Based CDN *Caching*," in *Proc. ACM SIGCOMM 2023*, pp. 981–999, 2023.

[15] M. Kinnunen, *Evaluating and Improving Web Performance Using Free-To-Use Tools*, University of Oulu, 2020.

[16] M. Fleming, *Digital Progression in Scotland's Colleges: Optimising CMS for Performance*, UCEM Repository, 2025.

[17] E. Ndubisi, *Performance Evaluation of HTTP/2 and HTTP/3 (QUIC) Using Lighthouse*, Carleton University, 2024.

[18] Chrome for Developers, "*First* Contentful Paint," 2025. [Online]. Available: https://developer.chrome.com/docs/lighthouse/performance/first-contentful-paint/

[19] Google, "*Largest Contentful Paint* (LCP)," 2023. [Online]. Available: https://web.dev/articles/lcp

[20] Google, "Total Blocking Time (TBT)," 2023. [Online]. Available: https://web.dev/articles/tbt

[21] DebugBear, "What Is *Speed Index*?" 2023. [Online]. Available: https://www.debugbear.com/docs/metrics/speed-index

[22] Google For Developer, 2025. *Understanding Core Web Vitals and Google Search Results.* [Online]. Available: https://developers.google.com/search/docs/appearance/core-web-vitals?hl=id

[23] Cloudflare, 2025. *What is a Cache Hit Ratio.* [Online] Available: https://www.cloudflare.com/learning/cdn/what-is-a-cache-hit-ratio/