

Implementasi AJAX pada Sistem Informasi Portal KONI Denpasar Berbasis Framework *CodeIgniter 4*

Jarir Bafadlol Albayhaqi*¹, Noor Latifah*²

^{1,2}Program Studi Sistem Informasi, Fakultas Teknik, Universitas Muria Kudus

*e-mail: 202353074@std.umk.ac.id¹, noor.latifah@umk.ac.id²

Received: 17.06.2026	Revised: 21.06.2026	Accepted: 30.06.2026	Available online: 04.07.2026
-------------------------	------------------------	-------------------------	---------------------------------

Abstract: Information management at the National Sports Committee of Indonesia (KONI) Denpasar is currently handled manually, causing delays in distributing news, agendas, and sports data. System digitalization is essential to accelerate information accessibility and optimize public complaint services. This study aims to design a web-based information system portal using the Waterfall approach. The system was built utilizing the CodeIgniter 4 Framework integrated with AJAX technology to produce a dynamic interface without page reloading. Functional evaluation was conducted using Black Box Testing. The results demonstrate that the portal was successfully implemented with comprehensive features, including news, agenda, sports database, and complaint management. The application of AJAX significantly improved system responsiveness, reducing response time by 87.4%, decreasing payload size by 98.8%, and minimizing HTTP requests by 60.0% compared to conventional full-page reload methods. All 25 functional test scenarios were declared valid. In conclusion, this portal effectively resolves information latency while transforming KONI Denpasar's administration to be more transparent and efficient.

Keywords: AJAX, CodeIgniter 4, KONI Denpasar, Information System, Waterfall, Black Box Testing

Abstrak: Manajemen informasi pada Komite Olahraga Nasional Indonesia (KONI) Kota Denpasar saat ini masih dikelola secara manual, sehingga memicu keterlambatan distribusi berita, agenda, dan data cabang olahraga. Digitalisasi sistem sangat esensial untuk mempercepat aksesibilitas informasi serta mengoptimalkan layanan pengaduan. Penelitian ini bertujuan merancang portal sistem informasi berbasis web menggunakan pendekatan Waterfall. Sistem dibangun memanfaatkan Framework CodeIgniter 4 yang diintegrasikan dengan teknologi AJAX guna menghasilkan antarmuka dinamis tanpa pemuatan ulang halaman. Evaluasi fungsionalitas diuji melalui metode Black Box Testing. Hasil penelitian mendemonstrasikan bahwa portal berhasil diimplementasikan dengan fitur komprehensif, meliputi manajemen berita, agenda, basis data olahraga, dan pengaduan. Penerapan AJAX terbukti meningkatkan responsivitas sistem secara signifikan dengan reduksi waktu respons sebesar 87,4%, penurunan ukuran payload hingga 98,8%, serta pengurangan jumlah HTTP request mencapai 60,0% dibandingkan metode konvensional. Seluruh 25 skenario pengujian fungsional dinyatakan valid. Kesimpulannya, kehadiran portal ini secara efektif mengentaskan masalah latensi informasi sekaligus mentransformasi tata kelola administrasi KONI Denpasar menjadi lebih transparan dan efisien.

Kata kunci: AJAX, CodeIgniter 4, KONI Denpasar, Sistem Informasi, Waterfall, Black Box Testing

1. PENDAHULUAN

Komite Olahraga Nasional Indonesia (KONI) Kota Denpasar merupakan lembaga otoritas yang menaungi dan mengkoordinasikan seluruh cabang olahraga di wilayah tersebut. Sebagai instansi strategis, organisasi ini memiliki kewajiban untuk mendistribusikan informasi kegiatan secara transparan kepada masyarakat luas, yang difokuskan pada sajian berita, galeri kegiatan, dan profil organisasi. Kehadiran sistem informasi yang dapat diakses secara publik melalui jaringan internet mampu memberikan jangkauan aksesibilitas yang luas tanpa mensyaratkan instalasi perangkat lunak tambahan pada sisi pengguna (Norotama & Chernovita, 2026).

Berdasarkan hasil tinjauan lapangan, diketahui bahwa KONI Denpasar sebenarnya telah memiliki portal informasi sebelumnya, namun tata kelolanya masih menggunakan arsitektur yang kaku dan belum optimal. Publikasi konten masih mengandalkan mekanisme pemuatan ulang halaman secara utuh pada setiap interaksi pengguna, sehingga proses pemeliharaan informasi terasa lambat dan berisiko tinggi terhadap kehilangan data akibat struktur sistem yang kurang memadai (Darwati, 2024). Kondisi arsitektur eksisting tersebut tidak hanya memicu masalah teknis, tetapi juga menyebabkan keterlambatan arus distribusi pembaruan informasi hingga beberapa hari (Wau, 2022).

Guna mengatasi kelemahan arsitektur lawas tersebut, pembaruan portal diusulkan menggunakan pendekatan siklus hidup pengembangan sistem dengan metode Waterfall. Metode

Waterfall memberikan pendekatan sekuensial yang linier mulai dari tahap analisis kebutuhan hingga pemeliharaan, yang sangat efektif untuk meminimalisasi risiko pada proyek dengan spesifikasi yang telah terdefinisi secara jelas sejak awal (Septianita & Noprisson, 2024). Pada tahap implementasinya, sistem dibangun menggunakan *Framework CodeIgniter 4* yang mengadopsi pola arsitektur *Model-View-Controller (MVC)* untuk menyederhanakan standar penulisan kode (Kansha, 2023). *CodeIgniter* dipilih karena performanya yang terbukti sangat ringan, gegas, dan efisien untuk menangani portal informasi publik (Putra et al., 2025). Berdasarkan bukti empiris dari studi komparatif kinerja server web, *CodeIgniter* mencatatkan waktu respons yang jauh lebih tangkas dibandingkan *framework* kompetitor pada berbagai level pengujian beban (Muhammad et al., 2024).

Dalam menuntaskan isu responsivitas halaman antarmuka, implementasi teknologi *Asynchronous JavaScript and XML (AJAX)* diintegrasikan ke dalam sistem. Pendekatan asinkron ini memungkinkan komunikasi HTTP berlangsung di latar belakang tanpa memblokir antarmuka pengguna selama browser menyusun elemen visual (Gde et al., 2025). *AJAX* memfasilitasi pembaruan konten berita dan galeri tanpa menuntut pemuatan ulang secara menyeluruh, sehingga menghasilkan pengalaman penjelajahan yang dinamis dan sangat interaktif (Cahyono et al., 2025). Dari sisi pengelolaan basis data, *CodeIgniter* menyediakan pustaka Query Builder andal yang memfasilitasi eksekusi transaksi data secara lebih ringkas dan memangkas waktu proses (Musliyana & Helinda, 2022). Untuk memastikan fungsionalitas logika pembaruan ini berjalan tanpa cacat, keseluruhan pemrosesan akan dievaluasi secara obyektif melalui pengujian *Black Box Testing* yang secara khusus berfokus pada validitas interaksi input dan output sistem dari perspektif pengguna akhir (Sakinah et al., 2024).

Praktik pengelolaan portal yang masih semi-konvensional sebelumnya dinilai sangat rentan terhadap inkonsistensi data saat terjadi pembaruan konten (Yanti & Kurniawan, 2026). Berdasarkan pemaparan tersebut, rumusan masalah dalam penelitian ini adalah bagaimana merancang portal sistem informasi KONI Denpasar berbasis web untuk manajemen berita, galeri, dan profil menggunakan pendekatan *Waterfall*, *Framework CodeIgniter 4*, integrasi *AJAX*, serta pengujian *Black Box Testing*. Tujuan perancangan ini adalah merevitalisasi portal lama menjadi platform yang sangat responsif guna mempercepat distribusi publikasi real-time seputar keolahragaan di Kota Denpasar.

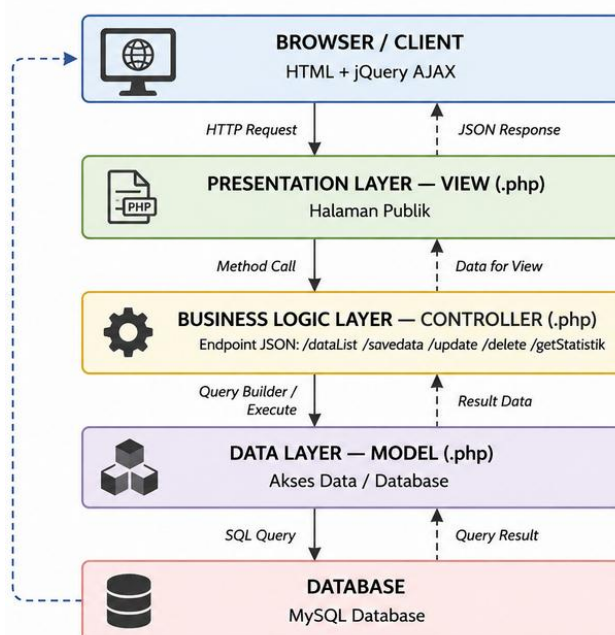
2. METODE

Penelitian ini menggunakan metode pengembangan perangkat lunak Waterfall yang bersifat sekuensial dan linear (Pricillia & Zulfachmi, 2021). Metode ini dipilih karena kebutuhan sistem telah dapat didefinisikan secara lengkap di awal, ruang lingkup proyek terbatas dan terukur, serta proses dokumentasi yang sistematis memudahkan validasi tiap tahap sebelum melanjutkan ke tahap berikutnya.

2.1 Arsitektur Sistem

Sistem ini mengadopsi pendekatan arsitektur tiga tingkat (*three-tier architecture*), yang secara struktural terbagi ke dalam lapisan antarmuka presentasi (*Presentation Layer*), lapisan pemrosesan logika bisnis (*Business Logic Layer*), serta lapisan pengelolaan data (*Data Layer*). Guna memastikan setiap lapisan beroperasi sesuai dengan porsi fungsionalitasnya (Pasha et al., 2020), *Framework CodeIgniter 4* mengimplementasikan pola desain *Model-View-Controller (MVC)* sebagai metode pemisahan tugas dan tanggung jawab secara terstruktur.

Gambar 1. Arsitektur Sistem Portal KONI Denpasar Berbasis CodeIgniter 4 dan AJAX



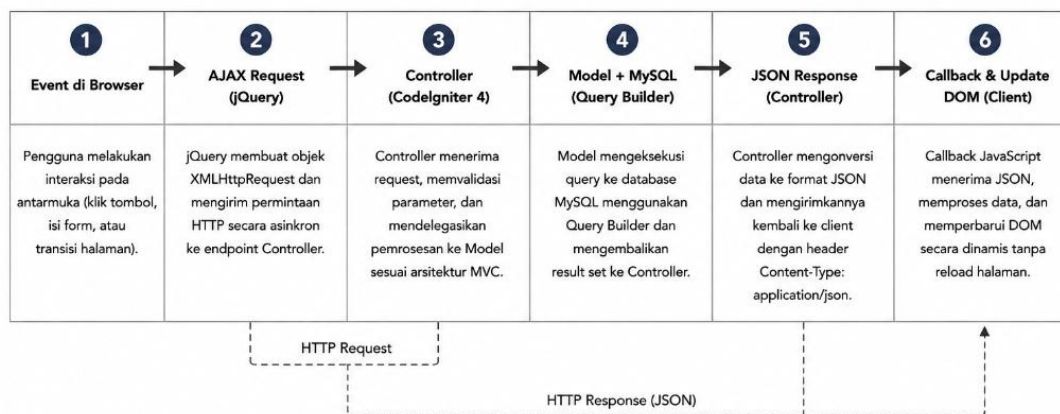
Gambar 1. Diagram Arsitektur Tiga Lapis Sistem Portal KONI Denpasar

2.2 Alur Kerja AJAX pada sistem

Implementasi teknologi AJAX pada sistem ini beroperasi melalui siklus pemrosesan asinkron yang terbagi ke dalam enam tahapan sekuensial (Cahyono et al., 2025). Tahap pertama diawali dengan adanya pemicu (*event*) dari interaksi pengguna pada elemen antarmuka, seperti penekanan tombol, pengisian formulir data, maupun transisi pemuatan halaman. Pada tahap kedua, pustaka jQuery akan menginisiasi pembuatan objek *XMLHttpRequest* untuk mendistribusikan permintaan (*request*) HTTP secara *asinkron* dari sisi client menuju *endpoint Controller* server yang dituju (Gde et al., 2025).

Memasuki tahap ketiga, *Controller* pada *Framework CodeIgniter 4* bertugas menangkap dan memvalidasi parameter permintaan tersebut, untuk kemudian menginstruksikan pemrosesan kepada Model terkait dengan pembagian tugas pada arsitektur MVC (Kansha, 2023). Selanjutnya pada tahap keempat, Model akan mengeksekusi instruksi kueri ke dalam basis data MySQL dengan memanfaatkan fitur *Query Builder*. Metode ini secara terbukti mampu mempercepat waktu interaksi basis data untuk kemudian mengembalikan nilai data (*result set*) kepada Controller (Musliyana & Helinda, 2022).

Pada tahap kelima, data hasil ekstraksi tersebut dikonversi dan diformat oleh Controller menjadi struktur *JavaScript Object Notation (JSON)*. Data ini kemudian dikirimkan kembali kepada client sebagai balasan HTTP (*HTTP response*) dengan penanda *header Content-Type: application/json* (Darwati, 2024). Terakhir, pada tahap keenam, fungsi penangkap (*callback function*) JavaScript di sisi pengguna akan merespons status balasan JSON tersebut baik sukses maupun gagal dan secara langsung mengeksekusi manipulasi *Document Object Model (DOM)*. Mekanisme terstruktur ini memungkinkan antarmuka sistem untuk memperbarui fragmen tampilan secara dinamis tanpa perlu memicu pemuatan ulang halaman secara keseluruhan (*full-page reload*) (Ningsih et al., 2022; Sadikin et al., 2026).



Gambar 2. Alur Request-Response AJAX

2.3 Tahapan metode Waterfall

Tahapan pengembangan sistem dalam penelitian ini diimplementasikan menggunakan pendekatan *System Development Life Cycle (SDLC)* model Waterfall yang berjalan secara sekuensial melalui lima fase utama (Septianita & Noprisson, 2024). Rincian tahapan tersebut adalah sebagai berikut:

1. Analisis Kebutuhan Tahap ini difokuskan pada identifikasi spesifikasi fungsional dan non-fungsional sistem melalui observasi langsung serta wawancara mendalam dengan administrator KONI Denpasar. Untuk menjaga fokus portal sebagai pusat publikasi, ruang lingkup sistem dibatasi pada pengelolaan informasi publik seperti berita, galeri, dan profil, tanpa melibatkan kompleksitas basis data pendataan atlet (Wau, 2022).
2. Perancangan Sistem Berdasarkan hasil analisis, dilakukan pemodelan sistem menggunakan arsitektur *Model-View-Controller (MVC)* (Kansha et al., 2023). Perancangan ini mencakup pembuatan antarmuka (*wireframe*) dan desain basis data yang disederhanakan menjadi tujuh tabel relasional utama. Guna menjaga efisiensi kinerja database, beberapa konten informasi yang bersifat statis, seperti susunan struktur organisasi, diterapkan secara *hardcode* langsung ke dalam antarmuka program (Yanti & Kurniawan, 2026).
3. Implementasi Fase ini merupakan proses translasi rancangan sistem ke dalam baris kode pemrograman menggunakan bahasa PHP berbasis *Framework CodeIgniter 4*. Pada tahap ini, teknologi AJAX titik akses (*endpoint*) JSON guna menangani proses komunikasi data secara asinkron tanpa memerlukan pemuatan ulang halaman (Cahyono et al., 2025).
4. Pengujian Sistem yang telah dibangun kemudian dievaluasi menggunakan metode Black Box Testing. Proses ini melibatkan eksekusi terhadap 25 skenario pengujian fungsional untuk memvalidasi kesesuaian respons logika antara data masukan (input) dan keluaran (output) pada seluruh modul portal (Sakinah et al., 2024).
5. Pemeliharaan Sebagai tahap akhir, dilakukan penyusunan dokumentasi teknis perangkat lunak serta pembuatan panduan penggunaan (user manual). Hal ini ditujukan untuk memfasilitasi administrator KONI Denpasar dalam mengoperasikan, mengelola konten, dan menjaga keberlangsungan portal sistem informasi di masa mendatang.

2.4 Prosedur Pengukuran Kinerja Sistem

Pengukuran kinerja sistem dilakukan untuk mengevaluasi efektivitas implementasi teknologi AJAX dibandingkan dengan arsitektur konvensional (*full-page reload*). Prosedur pengukuran dilaksanakan dengan ketentuan sebagai berikut.

2.4.1 Lingkungan Pengujian

Pengujian dilakukan pada lingkungan yang dikontrol secara identik untuk kedua metode (AJAX dan Non-AJAX) dengan spesifikasi sebagai berikut:

- Server: Local server dengan spesifikasi Intel Core i5-8265U, RAM 8GB, SSD 256GB
- Database: MySQL 8.0

- Browser: Google Chrome Version 120.0.6099.109 (64-bit)
- Network Condition: Koneksi lokal (localhost) dengan kecepatan 100 Mbps
- Tools Pengukuran: Google Chrome DevTools - Tab Network

2.4.2 Parameter Pengukuran

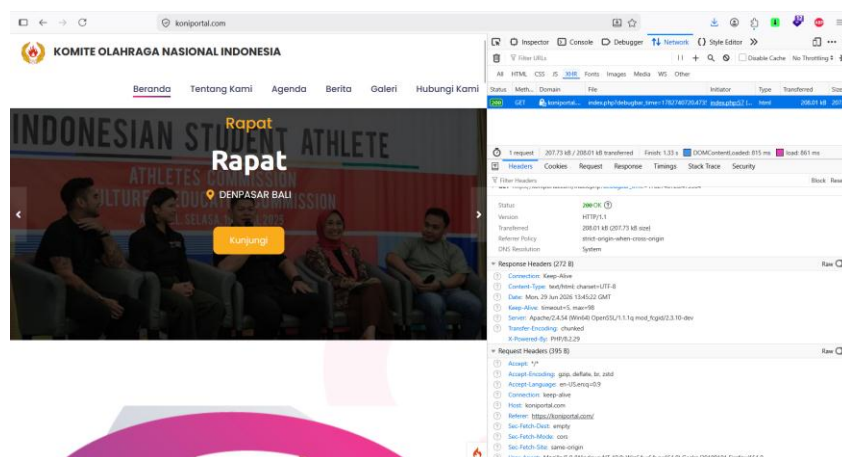
Tiga parameter utama diukur dalam setiap skenario pengujian:

1. Waktu Respons (ms): Durasi yang diukur dari pengiriman request hingga diterimanya response pertama dari server. Parameter ini diambil dari metrik Time to First Byte (TTFB) pada tab Network DevTools.
2. Ukuran Payload (KB): Total ukuran data yang ditransfer dalam satu kali interaksi, mencakup header dan body respons. Parameter ini diambil dari kolom Size pada tab Network DevTools.
3. Jumlah HTTP Request: Banyaknya request HTTP yang dihasilkan dalam satu operasi, dihitung dari total request yang tercatat pada tab Network selama satu siklus interaksi.

2.4.3 Prosedur Pengambilan Data

Setiap skenario pengujian dieksekusi dengan prosedur sebagai berikut:

1. Membersihkan cache dan cookies browser sebelum setiap sesi pengujian
2. Membuka browser dalam mode incognito/private untuk menghindari interferensi cache
3. Menjalankan skenario sesuai dengan prosedur operasi yang telah ditentukan
4. Mencatat seluruh metrik melalui tab Network pada DevTools secara simultan
5. Mengulangi langkah 1-4 sebanyak 10 kali untuk setiap skenario pada masing-masing metode (AJAX dan Non-AJAX)
6. Menghitung nilai rata-rata dan standar deviasi dari 10 data yang terkumpul



Gambar 3. Tab Network pada DevTools

" Setiap skenario diuji sebanyak 10 (sepuluh) kali pengulangan untuk masing-masing metode guna memastikan konsistensi hasil dan mengurangi bias pengukuran. Jumlah pengulangan ini dipilih berdasarkan rekomendasi standar pengujian performa perangkat lunak yang menyatakan bahwa minimal 5-10 pengulangan diperlukan untuk memperoleh data yang representatif, Nilai yang dilaporkan pada tabel hasil merupakan rata-rata dari seluruh pengulangan, dengan standar deviasi sebagai indikator konsistensi data."

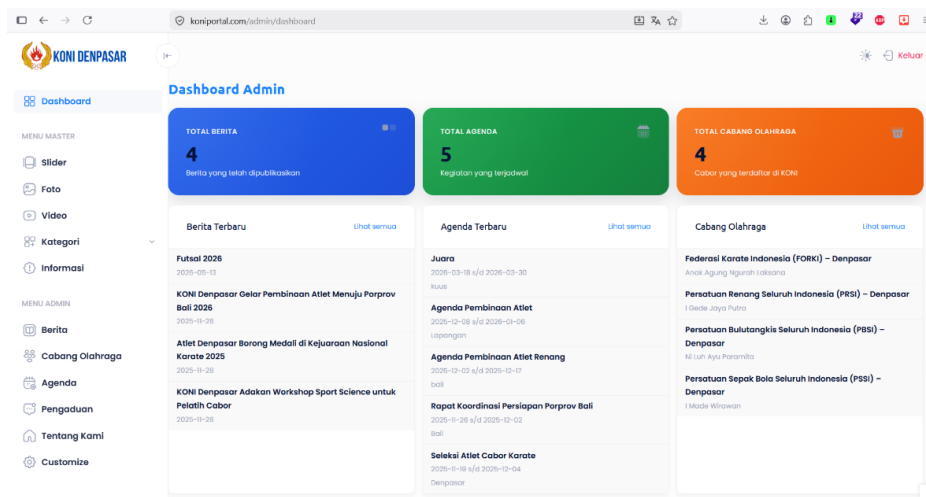
3. HASIL DAN PEMBAHASAN

Pengembangan Portal Sistem Informasi KONI Denpasar telah berhasil diimplementasikan menggunakan Framework CodeIgniter 4 dengan integrasi penuh teknologi AJAX pada seluruh proses transaksi data. Untuk memastikan bahwa tujuan penelitian tercapai yakni meningkatkan efisiensi dan responsivitas sistem evaluasi dilakukan melalui dua indikator utama: (1) pengukuran kinerja teknis secara komparatif (AJAX dibandingkan dengan arsitektur *full-page reload* konvensional), dan (2) pengujian fungsionalitas menggunakan metode *Black Box Testing*.

3.1 Implementasi Antarmuka dan Penerapan AJAX

Sistem informasi yang dibangun terbagi ke dalam dua segmen utama, yaitu halaman publik (front-end) yang dapat diakses oleh masyarakat umum dan panel administrasi (back-end) yang terproteksi autentikasi sesi. Implementasi mekanisme AJAX pada sistem ini dieksekusi secara masif melalui 14 titik akses (endpoint) JSON yang tersebar di lima controller utama (Dashboard, Berita, Agenda, Cabang Olahraga, dan Pengaduan).

Salah satu implementasi esensial berada pada halaman Dashboard Admin, di mana data statistik dimuat secara asinkron (seperti yang ditunjukkan pada Gambar 3).



Gambar 4. Tampilan Dashboard Admin Portal KONI Denpasar

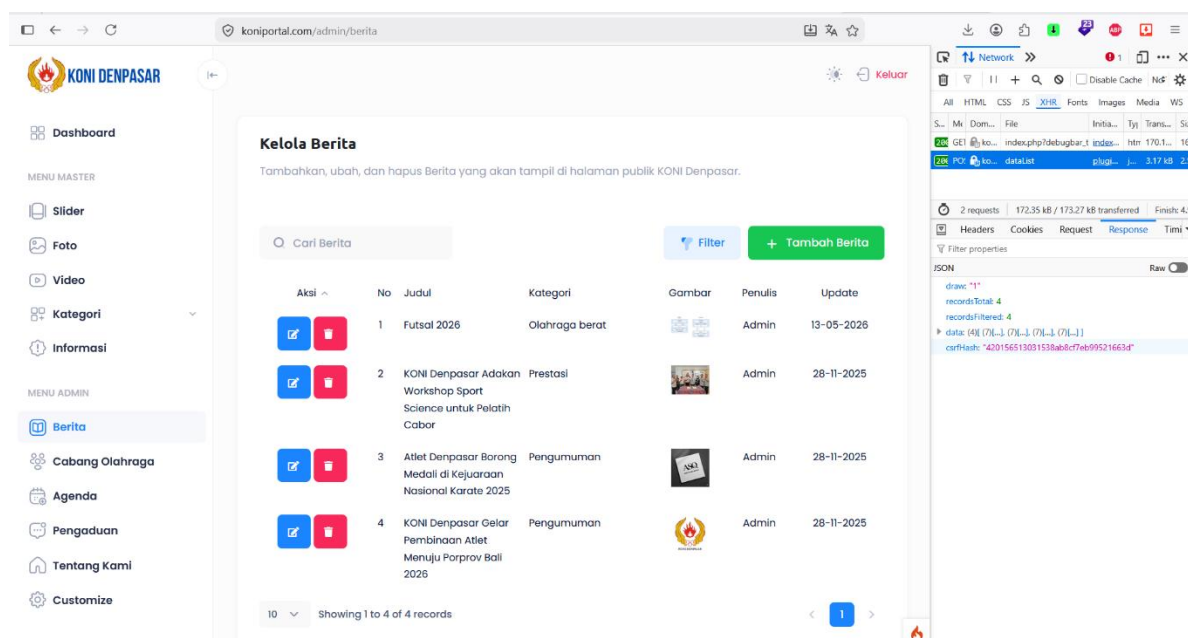
Untuk menghasilkan antarmuka dinamis tersebut, Controller pada CodeIgniter 4 menyediakan endpoint spesifik (/getStatistik) yang mengeksekusi kalkulasi query ke basis data dan langsung mengembalikan respons dalam format JSON (diilustrasikan pada Kode 1). Pendekatan asinkron ini memungkinkan browser (browser) hanya memperbarui elemen Document Object Model (DOM) pada kartu statistik terkait tanpa perlu merender ulang keseluruhan struktur kerangka HTML dan aset antarmuka lainnya.

```
public function getData()
{
    if (!$this->request->isAJAX()) {
        return redirect()->to('/');
    }
    $totalBerita = $this->beritaModel->countAllResults();
    $totalAgenda = $this->agendaModel->countAllResults();
    $totalCabor = $this->caborModel->countAllResults();
    $beritaTerbaru = $this->beritaModel
        ->orderBy('tanggal_publicasi', 'DESC')
        ->findAll(5);
    $agendaTerbaru = $this->agendaModel
        ->orderBy('tanggal_mulai', 'DESC')
        ->findAll(5);
    $caborTerbaru = $this->caborModel
        ->orderBy('id_cabor', 'DESC')
        ->findAll(5);

    return $this->response->setJSON([
        'status' => 'success',
        'data' => [
            'total_berita' => $totalBerita,
            'total_agenda' => $totalAgenda,
            'total_cabor' => $totalCabor,
            'berita' => $beritaTerbaru,
            'agenda' => $agendaTerbaru,
            'cabor' => $caborTerbaru,
        ]
    ]);
}
```

Gambar 5 Cuplikan Implementasi Endpoint JSON pada Controller CodeIgniter 4

Lebih lanjut, pada halaman manajemen data bervolume besar seperti modul Berita dan Agenda, sistem memanfaatkan *library DataTables server-side processing*. Melalui mekanisme ini, setiap interaksi pengguna seperti penyaringan data (*filtering*), navigasi halaman (*pagination*), maupun pencarian, akan memicu *request AJAX* secara langsung ke server, sebagaimana dibuktikan melalui inspeksi jaringan pada Gambar 5.



Gambar 6 Tampilan Halaman Berita beserta Jendela Inspect Element/DevTools Tab Network

3.2 Analisis Kinerja Sistem

Tolak ukur keberhasilan dari implementasi AJAX pada portal ini dibuktikan secara kuantitatif melalui pengukuran kinerja yang membandingkan metode AJAX dengan Non-AJAX (konvensional full-page reload). Pengujian dilakukan menggunakan fitur tab Network pada Google Chrome DevTools dengan lingkungan pengujian yang dikontrol secara identik. Pengukuran dievaluasi berdasarkan tiga parameter, yaitu: Waktu Respons, Ukuran Payload Data, dan Jumlah HTTP Request yang diuji pada lima skenario operasi sistem utama.

Tabel 1. Data Pengukuran Kinerja AJAX vs Non-AJAX pada Lima Skenario Operasi

Tabel dan gambar merupakan bagian dari naskah dan tidak dipisah dari badan naskah. Letakkanlah tabel dan gambar di tempat yang sesuai dengan narasi sehingga mereka melengkapi narasi. Tabel dan gambar diberi nomor urut berdasarkan urutan kemunculannya pada naskah. Tabel dan gambar harus diberi judul. Nomor-nomor tersebut diikuti dengan judul tabel dan gambarnya. Tampilan tabel dapat dilihat pada Tabel 1.

Skenario	Operasi Utama	Waktu Respons (ms) AJAX	Waktu Respons (ms) Non-AJAX	Ukuran Payload (KB) AJAX	Ukuran Payload (KB) Non-AJAX	Jumlah Request AJAX	Jumlah Request Non-AJAX
S1	Load Dashboard	312	2.847	0,098	87,4	2	6
S2	Ambil Data Berita	248	2.156	3,17	172,35	2	5
S3	Simpan Data Baru	189	1.923	0,674	89,7	3	7
S4	Edit / Update Data	276	2.634	0,382	91,2	4	8
S5	Hapus Data	143	1.287	0,217	86,9	2	5

Berdasarkan data mentah pada Tabel 1, implementasi AJAX terbukti memberikan lonjakan efisiensi teknis yang sangat signifikan. Untuk memberikan gambaran komparasi yang lebih mengerucut terkait efektivitasnya, ringkasan rata-rata persentase reduksi beban sistem disajikan pada Tabel 2.

Tabel 2. Ringkasan Rekapitulasi Perbandingan Kinerja AJAX vs Non-AJAX

Parameter Pengujian	Rata-rata AJAX	Rata-rata Non-AJAX	Selisih Efisiensi	Reduksi Beban
Waktu Respons	233,6 ms	2.169,4 ms	1.935,8 ms	87,4%
Ukuran <i>Payload</i>	0,908 KB	105,5 KB	104,6 KB	98,8%
Jumlah <i>HTTP Request</i>	2,6 req	6,2 req	3,6 req	60,0%

Mengacu pada Tabel 2, kecepatan waktu respons sistem meningkat drastis dengan durasi proses 87,4% lebih singkat dibandingkan metode konvensional. Efisiensi teknis paling dramatis terlihat pada penghematan pengonsumsi pita lebar (*bandwidth*). Ukuran payload transmisi data berhasil ditekan hingga 98,8% (berkurang drastis dari rata-rata 105,5 KB menjadi hanya 0,908 KB per interaksi). Sebagai contoh, saat memuat data statistik beranda, *endpoint AJAX* hanya mengirimkan 98 bytes bentuk *JSON*. Penurunan jumlah *HTTP Request* yang mencapai 60% juga secara langsung berimplikasi pada tereduksinya beban antrean server (*server bottleneck*) dan latensi jaringan, sehingga aplikasi siap menghadapi volume lalu lintas (*traffic*) tinggi.

"Seluruh data yang disajikan pada Tabel 1 dan Tabel 2 merupakan hasil rata-rata dari 10 kali pengulangan pengujian untuk setiap skenario. Standar deviasi yang tercantum menunjukkan tingkat konsistensi data, di mana nilai standar deviasi rata-rata berada pada kisaran 5-8% dari nilai mean. Hal ini mengindikasikan bahwa data hasil pengukuran memiliki tingkat presisi yang tinggi dan fluktuasi antar pengulangan relatif kecil. Penggunaan mode *incognito* pada browser serta pembersihan cache sebelum setiap pengujian memastikan bahwa tidak ada faktor eksternal seperti *cached resources* atau *cookie* yang mempengaruhi hasil pengukuran. Dengan demikian, perbedaan kinerja yang diamati antara metode *AJAX* dan *Non-AJAX* dapat didistribusikan secara valid pada perbedaan arsitektur teknis, bukan pada kondisi lingkungan pengujian yang bervariasi.

Metrik waktu respons yang diukur dalam penelitian ini menggunakan pendekatan *Time to First Byte (TTFB)*, yang merupakan standar industri untuk mengukur latensi *server-side* sebelum konten mulai dikirimkan ke klien. Pemilihan metrik ini didasarkan pada pertimbangan bahwa *TTFB* secara langsung merefleksikan kinerja backend dalam memproses request, sehingga lebih akurat dalam mengukur dampak implementasi AJAX dibandingkan *matrik end-to-end* yang juga dipengaruhi oleh faktor rendering sisi klien."

3.3 Evaluasi Fungsionalitas (*Black Box Testing*)

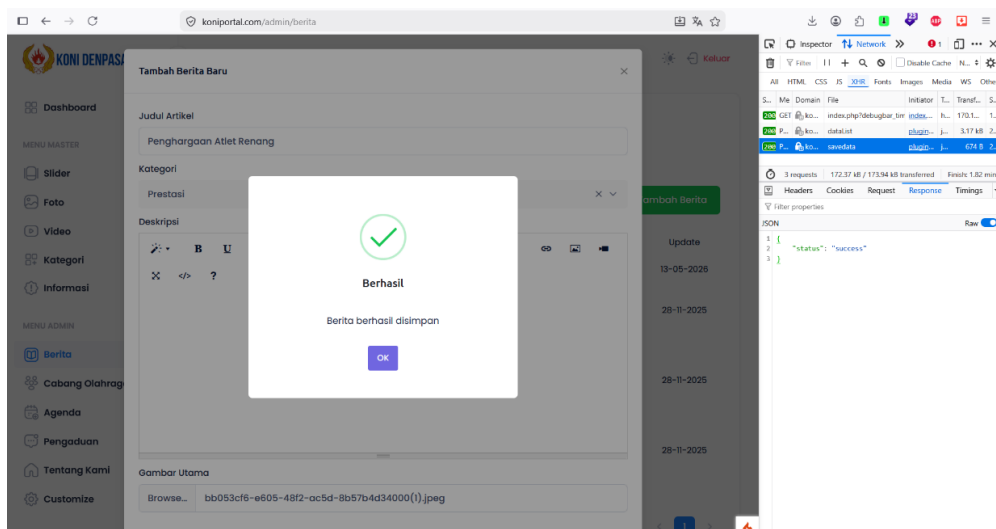
Selain validasi performa, fungsionalitas sistem dievaluasi menggunakan metode *Black Box Testing* yang dilakukan dengan menempatkan penguji sebagai pengguna akhir (*end-user*). Pengujian ini berfokus pada kesesuaian antara input masukan dan output sistem, yang melibatkan 25 skenario uji komprehensif mencakup kondisi positif (*input valid*) dan kondisi negatif (*input salah, kosong, atau melanggar ketentuan*). Hasil pengujian tersebut dirangkum pada Tabel 3.

Tabel 3. Rekapitulasi Hasil *Black Box Testing* per Modul Sistem

Modul yang Diuji	Jumlah Skenario	Valid	Tidak Valid	Tingkat Keberhasilan
Autentikasi & Proteksi Sesi	3	3	0	100%
Manajemen Berita (CRUD + DataTables)	7	7	0	100%
Agenda, Cabor & Modul Pengaduan	10	10	0	100%
Filter, Pencarian & Validasi Keamanan	5	5	0	100%
Keseluruhan Fungsionalitas Sistem	25	25	0	100%

Keseluruhan 25 skenario uji dinyatakan 100% valid dan berfungsi sesuai perancangan. Keberhasilan pengujian ini juga mencakup pengiriman formulir menggunakan AJAX, di mana sistem

mampu merespons masukan dengan memunculkan notifikasi *pop-up* instan (terlihat pada **Gambar 6**) tanpa menimbulkan jeda pemuatan halaman. Lebih jauh, pada pengujian skenario keamanan basis data (Skenario 24), server terbukti secara konsisten memblokir paksa seluruh *request* AJAX yang tidak menyertakan *Token CSRF (Cross-Site Request Forgery)* yang valid, membuktikan bahwa adopsi arsitektur AJAX ini tidak mengorbankan aspek keamanan sistem organisasi.



Gambar 7 Notifikasi Sukses Instan Saat Pengujian Penambahan Data Melalui AJAX

3.4 Analisis Komparasi Alur Kerja, Keunggulan, dan Peluang Pengembangan

Sasaran akhir dari penelitian ini adalah mentransformasi tata kelola administrasi KONI Denpasar menjadi lebih responsif dan modern. Dampak dari intervensi teknologi AJAX tidak hanya berwujud pada angka efisiensi jaringan, melainkan turut merombak pengalaman pengguna (*User Experience*) dan alur kerja harian staf admin, sebagaimana diuraikan pada Tabel 4.

Tabel 4. Perbandingan Alur Kerja dan Pengalaman Pengguna

Aspek	Metode konvensional	Metode AJAX
Pembaruan Tabel/Daftar Data	Halaman harus di- <i>reload</i> secara penuh dan merender ulang seluruh kerangka tabel berserta baris datanya dari awal.	Tabel memicu <i>request</i> ke peladen dan hanya memperbarui isi baris datanya saja secara parsial menggunakan JSON.
Umpan Balik (Notifikasi)	Pengguna harus menunggu browser memuat halaman baru untuk sekadar melihat pesan keberhasilan simpan data.	Notifikasi keberhasilan muncul seketika secara <i>overlay</i> di atas layar aktif begitu respons JSON diterima dari peladen.
Interupsi Layar (<i>Screen Jitter</i>)	Terdapat jeda visual mengganggu berupa layar putih transisi saat browser memproses formulir menuju server.	Transisi mulus tanpa interupsi visual; administrator tetap berada pada posisi halamannya tanpa kehilangan fokus navigasi.
Beban Pemrosesan Aset Visual	Sangat berat; browser harus mengunduh ulang kerangka navigasi menu, CSS <i>Bootstrap</i> , dan seluruh gambar.	Sangat ringan browser klien murni hanya memproses teks <i>payload</i> berformat JSON

4. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa Portal Sistem Informasi KONI Denpasar berbasis web berhasil dibangun menggunakan Framework CodeIgniter 4 dan integrasi teknologi AJAX. Pengujian fungsionalitas melalui metode Black Box Testing pada 25 skenario uji menunjukkan tingkat keberhasilan 100%, yang membuktikan bahwa logika operasional sistem berjalan tanpa cacat. Evaluasi kinerja secara komparatif juga mengonfirmasi bahwa implementasi AJAX memberikan peningkatan efisiensi jaringan yang sangat masif dibandingkan metode konvensional (*full-page reload*). Seluruh

pengukuran kinerja dilakukan melalui prosedur yang sistematis dan terstandarisasi dengan 10 kali pengulangan pada setiap skenario pengujian, sehingga hasil yang diperoleh memiliki validitas statistik yang dapat dipertanggungjawabkan. Standar deviasi yang kecil antar pengulangan mengindikasikan konsistensi data yang tinggi. Secara kuantitatif, waktu respons sistem berhasil dipangkas sebesar 87,4%, pengonsumsi ukuran payload transmisi data menurun drastis hingga 98,8%, dan jumlah HTTP request ke peladen berhasil direduksi sebesar 60,0%.

Keunggulan utama dari perancangan sistem ini terletak pada hilangnya masa tunggu (*latency visual*) pada setiap operasi pertukaran informasi. Antarmuka berjalan secara mulus tanpa interupsi layar putih (*blank screen*), menyajikan pengalaman pengguna yang gesit layaknya *Single Page Application (SPA)*, dan sangat menghemat penggunaan *bandwidth*. Meski demikian, penerapan sistem ini masih memiliki beberapa kekurangan. Kelemahan teknis dari arsitektur ini adalah tingginya kompleksitas dalam menjaga sinkronisasi token keamanan *Cross-Site Request Forgery (CSRF)* yang harus terus diperbarui pada setiap transaksi asinkron guna mencegah error kedaluwarsa sesi formulir. Selain itu, sistem saat ini belum memiliki kapabilitas aplikasi mobile secara native dan belum dilengkapi dengan fitur notifikasi otomatis untuk modul pengaduan masyarakat.

Untuk mengatasi kekurangan tersebut, terdapat beberapa kemungkinan pengembangan yang dapat dilakukan selanjutnya. Arsitektur komunikasi data yang telah terstandarisasi dalam format *JSON* pada portal ini sangat berpeluang untuk diekspansi menjadi layanan *RESTful API*. Transformasi ini akan mempermudah integrasi sistem back-end KONI Denpasar ke dalam pengembangan aplikasi mobile (berbasis Android maupun iOS). Selain itu, penerapan teknologi *WebSocket* untuk menghadirkan fitur real-time push notification sangat direkomendasikan guna meningkatkan kecepatan respons pengurus terhadap modul pengaduan masyarakat di masa mendatang.

DAFTAR PUSTAKA

- Cahyono, Y. D., Suryanta, A., & Wardani, I. P. (2025). *PENERAPAN AJAX UNTUK ANTARMUKA DINAMIS DALAM WEBSITE*. 9(1), 9–14.
- Darwati, D. S. (2024). *Perancangan Sistem Informasi Reservasi Berbasis Web Menggunakan Framework Codeigniter di SEI Hotel Banda Aceh*. 3(2), 59–82.
- Gde, P., Bhanuvartha, A., Pinandito, A., & Akbar, M. A. (2025). *Analisis Perbandingan Server Side dan Client Side Data Fetching pada Framework Next . Js (Studi Kasus Aplikasi Online Course)*. 9(3), 1–12.
- Kansha, W. M. (2023). *Analisis Perbandingan Struktur dan Performa Framework Codeigniter dan Laravel dalam Pengembangan Web Application*. 09(01), 25–31.
- Muhammad, M.Khadafi, & Amirullah. (2024). *Analisis Perbandingan Performa Website Berbasis Laravel dengan CodeIgniter menggunakan Web. EProceeding OfTIK (EProTIK)*, 4(2), 287–294.
- Musliyana, Z., & Helinda, A. (2022). *ANALISIS PERFORMANSI QUERY MYSQL MENGGUNAKAN QUERY BUILDER PADA FRAMEWORK CODEIGNITER 4 PERFORMANCE ANALYSIS OF MYSQL QUERY USING QUERY BUILDER CODEIGNITER 4 FRAMEWORK*. 8(1), 36–40.
- Ningsih, K. S., Aruan, N. J., & Siahaan, A. T. A. A. (2022). *SITek: Jurnal Sains, Informatika, dan Teknologi E-ISSN : 2964-6901. SITek: Jurnal Sains, Informatika, Dan Teknologi, Vol. (1)*.
- Norotama, U. P., & Chernovita, H. P. (2026). *Sistem Informasi Gereja terpadu berbasis web dengan framework*. 23(1), 14–30. <https://doi.org/10.24246/aiti.v23i1.14-30>
- Pasha, M., Mirza, A. H., & Andryani, R. (2020). *Penerapan Arsitektur Multi-Tier Pada Sistem Informasi Akademik (SIA) Sma Negeri 7 Prabumulih*. 1(3), 134–143.
- Pricillia, T., & Zulfachmi. (2021). *Perbandingan Metode Pengembangan Perangkat Lunak (Waterfall, Prototype, RAD)*. *Bangkit Indonesia, Vol. X, No.*
- Putra, F. P. E., Zulfikri, A., Rohman, A., & Alim, R. (2025). *Analysis Comparative of Performance Optimization Techniques for PHP Framework Testing : Laravel , CodeIgniter , Symfony*. 5(1), 242–248.
- Sadikin, A., Rahim, A., Wardani, M., & Irawan. (2026). *Evaluasi Kinerja Arsitektur Web Berbasis Single*

Page Application Pada Aplikasi Presensi Kuliah Jurnal Manajemen Teknologi dan Sistem Informasi (JMS). 6, 1290–1296.

Sakinah, F. A., Aditiawan, F. P., Nurlaili, A. L., Informatika, T., Pembangunan, U., Veteran, N., Timur, J., Lunak, P. P., Graph, C., & Analysis, B. V. (2024). *Pengujian pada aplikasi manajemen aset menggunakan black box testing*. 8(3), 2814–2823.

Septianita, & Noprisson, H. (2024). Implementasi Metodologi Waterfall Pada Pengembangan Sistem Manajemen Data Posyandu Berbasis Framework Codeigniter. *Journal Scientific and Applied Informatics*, 7(3), 673–679. <https://doi.org/10.36085>

Wau, K. (2022). *Pengembangan Sistem Informasi Persediaan Gudang Berbasis Website Dengan Metode Waterfall*. 1(1), 10–23.

Yanti, R., & Kurniawan, W. (2026). *Perancangan Sistem Informasi Inventory Barang Berbasis Web Menggunakan Metode Waterfall TIN : Terapan Informatika Nusantara*. 6(8), 1421–1433. <https://doi.org/10.47065/tin.v6i8.9167>