***EKSPERIMEN PENGETAHUAN DAN WAWASAN   
Teknik Asynchronous JavaScript untuk Meningkatkan Kinerja HalamanChatGPT***

*Laporan ini disusun untuk memenuhi Tugas Mata Kuliah Pengembangan Web*



**Disusun Oleh:**

Arnanda Prasatya 221524034

Bhisma Chandra Yudha Setiawan 221524037

Jeihan Ilham Kusumawardhana 221524042

Keanu Rayhan Harits 221524043

Mochamad Fathur Rabbani 221524045

Muhammad Hanif 221524050

Reza Maulana Aziz 221524057

**Mata Kuliah :**

Nama Mata Kuliah : Pengembangan Web

Kode Mata Kuliah : 21TI3005

**Dosen Pengampu:**

Djoko Cahyo Utomo Lieharyani, S.Kom., M.MT

**PROGRAM STUDI D4 TEKNIK INFORMATIKA**

**JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA**

**POLITEKNIK NEGERI BANDUNG**

**2024**

# Metodology

Untuk eksperimen dengan teknik Asynchronous JavaScript guna meningkatkan kinerja halaman, kamu bisa mengikuti metodologi berikut:

**1. Tujuan dan Definisi Masalah**

* **Tujuan:** Mengevaluasi seberapa efektif teknik Asynchronous JavaScript dalam meningkatkan kinerja halaman web dibandingkan dengan teknik synchronous.
* **Definisi Masalah:** Menilai dampak teknik asynchronous terhadap waktu pemuatan halaman, pengalaman pengguna, dan efisiensi pemrosesan data di browser.

**2. Desain Eksperimen**

* **Variabel Independen:** Teknik JavaScript yang digunakan (synchronous vs asynchronous).
* **Variabel Dependen:** Waktu pemuatan halaman, waktu respons interaksi, dan pemanfaatan sumber daya (CPU dan memori).
* **Kontrol:** Gunakan halaman web yang serupa untuk memastikan perbandingan yang adil. Pastikan untuk mengontrol faktor eksternal seperti koneksi internet dan perangkat keras.

**3. Metodologi**

1. **Persiapan Lingkungan:**
   * Buat dua versi halaman web yang sama: satu menggunakan JavaScript synchronous dan satu lagi menggunakan teknik asynchronous (seperti async dan defer dalam tag <script>, atau menggunakan fetch dan Promise untuk operasi asinkron).
2. **Implementasi Teknik JavaScript:**
   * **Synchronous:** Implementasikan JavaScript secara langsung tanpa menggunakan teknik asynchronous.
   * **Asynchronous:** Terapkan teknik seperti async/await, Promises, dan fetch untuk memuat data secara asinkron.
3. **Pengukuran Kinerja:**
   * Gunakan alat seperti Chrome DevTools, Lighthouse, atau WebPageTest untuk mengukur waktu pemuatan halaman, waktu respons interaksi, dan pemanfaatan sumber daya untuk kedua versi halaman.
4. **Pengujian:**
   * Lakukan pengujian di berbagai kondisi (misalnya, berbagai kecepatan internet, perangkat keras yang berbeda) untuk mendapatkan data yang komprehensif.
5. **Pengumpulan Data:**
   * Catat hasil pengukuran kinerja dari kedua versi halaman web. Perhatikan perbedaan dalam waktu pemuatan, waktu respons, dan penggunaan sumber daya.
6. **Analisis Data:**
   * Bandingkan hasil pengukuran antara versi synchronous dan asynchronous. Gunakan statistik deskriptif untuk mengidentifikasi perbedaan signifikan.

# Environment

Untuk menjalankan eksperimen dengan teknik Asynchronous JavaScript, kamu memerlukan lingkungan yang mendukung pengembangan web dan pengujian performa. Berikut adalah beberapa komponen dan alat yang akan membantu:

**1. Perangkat Keras dan Perangkat Lunak**

* **Komputer atau Laptop:** Memadai untuk pengembangan dan pengujian. Spesifikasi perangkat keras yang cukup baik akan membantu dalam pengujian kinerja dan analisis data.
* **Sistem Operasi:** Sistem operasi seperti Windows, macOS, atau Linux yang mendukung alat pengembangan web.

**2. Perangkat Pengembangan Web**

* **Editor Kode:** IDE atau editor teks seperti Visual Studio Code, Sublime Text, atau Atom untuk menulis dan mengedit kode JavaScript.
* **Web Browser:** Browser modern seperti Google Chrome, Mozilla Firefox, atau Microsoft Edge untuk pengujian dan analisis. Chrome sering direkomendasikan karena alat pengembangannya yang kuat.

# Source Code

Async.js

A screen shot of a computer program

Description automatically generated

indexAsc.html

A screen shot of a computer

Description automatically generated

Sync.html

A screen shot of a computer program

Description automatically generated

Indexsc.html

A screen shot of a computer

Description automatically generated

Data.json

A black rectangular sign with orange text

Description automatically generated

# Eksperiment

 **Waktu Pemuatan Halaman:**

* **Synchronous:** Halaman cenderung memuat lebih lambat karena permintaan data (seperti AJAX) dilakukan secara synchronous. Ini berarti proses pengunduhan data memblokir rendering halaman hingga data sepenuhnya dimuat.
* **Asynchronous:** Halaman cenderung memuat lebih cepat karena permintaan data dilakukan secara asynchronous. Data dapat dimuat di latar belakang tanpa memblokir rendering halaman, memungkinkan pengguna melihat konten yang ada sementara data tambahan dimuat.

 **Waktu Respons Interaksi:**

* **Synchronous:** Interaksi pengguna (seperti klik atau input) mungkin terasa lebih lambat karena pemrosesan JavaScript yang memblokir eksekusi event lainnya selama data sedang dimuat.
* **Asynchronous:** Interaksi pengguna biasanya terasa lebih responsif karena JavaScript tidak memblokir event lain. Pengguna dapat berinteraksi dengan halaman sementara data tambahan dimuat secara latar belakang.

 **Penggunaan Sumber Daya:**

* **Synchronous:** Dapat menyebabkan penggunaan CPU dan memori yang lebih tinggi jika ada banyak pemrosesan yang dilakukan selama data dimuat secara synchronous.
* **Asynchronous:** Dapat mengurangi penggunaan CPU dan memori secara signifikan karena pemrosesan data dilakukan di latar belakang dan tidak memblokir thread utama.

 **Pengalaman Pengguna:**

* **Synchronous:** Pengalaman pengguna mungkin terganggu jika halaman tidak sepenuhnya dimuat sebelum pengguna mulai berinteraksi. Hal ini dapat menyebabkan frustrasi jika halaman terlihat tidak lengkap.
* **Asynchronous:** Pengalaman pengguna biasanya lebih baik karena halaman bisa tampil lebih cepat dan interaksi tidak terganggu oleh pemuatan data yang sedang berlangsung.

|  |  |
| --- | --- |
| Syncronous | Asyncronous |
|  |  |

# Result