

MAKALAH
LIBRARY JAVASCRIPT
PEMROGRAMAN WEB



DISUSUN OLEH :

Anang Sukana	(2255201085)
Adrian Nabil M	(2255201311)
Andika Triwardana	(2255201144)
Dita Fitrianingrum	(2255201311)
Diyo Putra .S	(2255201036)
Fairuz Zahran .Z	(2255201071)
M. Zulfa Hafidz	(2255201022)
M. Farhan Romy .I	(2255201292)
Rian Suliyanto	(2255201068)

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH TANGERANG
2024

KATA PENGANTAR

Puji syukur kami panjatkan atas kehadiran Tuhan Yang Maha Esa yang telah melimpahkan nikmat, taufik, serta hidayah-Nya, sehingga kami dapat menyelesaikan Makalah Pemrograman web tepat pada waktu. Terima kasih juga kami ucapkan kepada guru pembimbing yang selalu memberikan dukungan dan bimbingannya.

Makalah ini kami buat dengan tujuan untuk memenuhi nilai tugas Pemrograman Web. Tak hanya itu, kami juga berharap makalah ini bisa bermanfaat untuk penulis pada khususnya dan pembaca pada umumnya. Walaupun demikian, kami menyadari dalam penyusunan makalah ini masih banyak kekurangan. Maka dari itu, kami sangat mengharapkan kritik dan saran untuk kesempurnaan makalah ini.

Akhir kata, kami berharap semoga makalah Pemrograman Web ini bisa memberikan informasi dan ilmu yang bermanfaat bagi kita semua. Kami juga mengucapkan terimakasih kepada para pembaca yang telah membaca makalah ini hingga akhir.

Tangerang, 25 Juli 2024

Penulis

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB I	
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penulisan.....	2
BAB II	
PEMBAHASAN.....	3
2.1 Pengertian JavaScript.....	3
2.1.1 Beberapa Karakteristik JavaScript.....	3
2.2 Library JavaScript.....	4
2.2.1 Library JavaScript yang populer meliputi:.....	4
2.3 React JS.....	5
2.3.1 Konsep dan Fitur Utama React JS.....	5
2.3.2 Kelebihan dan Kekurangan React JS.....	6
2.3.3 Modul - modul yang disediakan React JS.....	7
2.3.4 Prerequisite React JS.....	8
2.3.5 Contoh Penerapan React JS.....	9
2.4 jQuery.....	10
2.4.1 Konsep dan Fitur Utama jQuery.....	10
2.4.2 Kelebihan dan Kekurangan jQuery.....	11
2.4.3 Modul - modul yang disediakan jQuery.....	11
2.4.4 Prerequisite jQuery.....	12
2.4.5 Contoh Penerapan jQuery.....	12
2.5 Chart.JS.....	14
2.5.1 Konsep dan Fitur Utama Chart.JS.....	14
2.5.2 Kelebihan dan Kekurangan Chart.JS.....	14
2.5.3 Modul - modul yang disediakan Chart.JS.....	16
2.5.4 Prerequisite Chart.JS.....	16
2.5.5 Contoh Penerapan Chart.JS.....	17
2.6 Perbedaan utama antara framework dan library.....	19
BAB III	
PENUTUP.....	20
3.1 Kesimpulan.....	20
3.2 Saran.....	20
DAFTAR PUSTAKA.....	21

BAB I

PENDAHULUAN

1.1 Latar Belakang

JavaScript diciptakan selama 10 hari saat bulan Mei 1995 oleh Brendan Erich, yang bekerja di Netscape (sekarang Mozilla). Sebelum bernama JavaScript, ia terlebih dinamai Mocha dan selanjutnya pada September 1995 namanya diubah kembali menjadi Livescript. Hingga pada Desember 1995 dinamai menjadi JavaScript sejak lisensinya dimiliki Sun.

Selanjutnya pada tahun 1996 - 1997 ECMA membuat standar spesifikasi untuk JavaScript. Standarisasi ini dilakukan agar semua vendor browser pada saat itu dapat menjalankan dan mengimplementasikan JavaScript, Versi rilis pertama dari standarisasi yang dilakukan ECMA ini dinamai ECMA 262 Ed.1. Framework JavaScript sendiri berkembang saat dimulainya teknologi JQuery. JQuery merupakan sebuah library yang memuat sekumpulan fungsi yang berguna untuk mempermudah programmer dalam membuat website.

Hingga sampai saat ini sudah banyak framework yang dikembangkan seperti React, Ember, Backbone, Vue, Marko dll, untuk dapat digunakan oleh programmer dalam pengembangan website. Dengan maksud ini diharapkan para programmer website (khususnya Front End Developer) dapat melakukan migrasi yang semula membangun website tanpa menggunakan JavaScript Framework (menggunakan vanilla JavaScript maupun jQuery library) menjadi menggunakan JavaScript framework.

Dengan semakin kompleksnya aplikasi web saat ini, pemilihan *library* atau *framework* yang tepat menjadi sangat penting. Pengembang tidak hanya perlu mempertimbangkan kemudahan penggunaan, tetapi juga faktor seperti performa, keamanan, dan dukungan komunitas. Hal ini mendorong penelitian lebih lanjut mengenai perbandingan berbagai *library* dan *framework* JavaScript untuk menentukan mana yang paling sesuai dengan kebutuhan spesifik proyek pengembangan web.

1.2 Rumusan Masalah

1. Pengertian JavaScript
2. Library JavaScript meliputi apa saja?
3. Apa saja kelebihan dan kekurangan dari berbagai *library* JavaScript yang populer
4. Apa faktor-faktor yang harus dipertimbangkan dalam pemilihan library JavaScript untuk pengembangan aplikasi web?

1.3 Tujuan Penulisan

Tujuan dari penulisan ini adalah untuk mengeksplorasi dan membandingkan berbagai *library* JavaScript, serta menganalisis dampaknya terhadap efisiensi dan efektivitas pengembangan aplikasi web. Dengan memahami karakteristik masing-masing *library*, diharapkan pengembang dapat membuat keputusan yang lebih baik dalam pemilihan alat yang akan digunakan dalam proyek mereka.

BAB II

PEMBAHASAN

2.1 Pengertian JavaScript

JavaScript adalah bahasa yang berbentuk kumpulan skrip yang ada fungsinya berjalan pada suatu dokumen HTML, sepanjang sejarah internet bahasa ini adalah bahasa skrip pertama untuk web. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap bahasa HTML dengan mengizinkan pengeksekusian perintah-perintah di sisi user, yang artinya di sisi browser bukan di sisi server web.

JavaScript bergantung kepada browser(navigator) yang memanggil halaman web yang berisi skrip skrip dari JavaScript dan tentu saja terselip di dalam dokumen HTML. JavaScript juga tidak memerlukan kompilator atau penerjemah khusus untuk menjalankannya (pada kenyataannya kompilator JavaScript sendiri sudah termasuk di dalam browser tersebut). Lain halnya dengan bahasa “Java” (dengan mana JavaScriptselalu dibanding bandingkan) yang memerlukan kompilator khusus untuk menerjemahkannya di sisi user/klien.

2.1.1 Beberapa Karakteristik JavaScript

1. **Berbasis Objek** : JavaScript mendukung pemrograman berbasis objek, yang berarti Anda dapat membuat objek yang memiliki properti dan metode
2. **Client-Side** : Sebagian besar kode JavaScript dijalankan di sisi klien (browser) meskipun JavaScript juga dapat dijalankannya di sisi server menggunakan platform seperti Node.js.
3. **Dinamis** : JavaScript adalah bahasa yang dinamis, yang berarti variabelnya bisa berubah tipe data selama runtime.
4. **Sintaksis Sederhana** : JavaScript memiliki sintaksis yang relatif mudah dipelajari, mirip dengan bahasa pemrograman lain seperti C atau Java.

2.2 Library JavaScript

JavaScript library adalah kumpulan kode JavaScript yang telah ditulis sebelumnya, yang dirancang untuk memudahkan pengembangan aplikasi berbasis JavaScript, terutama dalam konteks teknologi web seperti AJAX. Library ini dapat disertakan dalam situs web dengan menyematkannya langsung di HTML melalui tag skrip.

2.2.1 Library JavaScript yang populer meliputi:

1. React, JSL dari Facebook paling populer yang berisi *code* yang kompleks dan interaktif
2. Vue.js, JSL yang ringan dan mudah diintegrasikan dengan JSL atau Framework JavaScript lainnya
3. Ember.js, JSL yang memiliki banyak *add-ons* dan mudah di-*update* meski *code*-nya sudah lama
4. Bideo.js, JSL untuk membuat latar belakang video pada *website*
5. Anime.js, JSL khusus untuk membangun animasi modern
6. Chart.js, JSL untuk mendesain grafik dari data
7. Meteor.js, JSL *back-end* yang bisa digunakan untuk manajemen *logic* dari *server*, meski juga bisa digunakan untuk *full-stack*

2.3 React JS

React JS adalah sebuah library milik bahasa pemrograman JavaScript. Biasanya, orang-orang juga menyebut React JS sebagai framework. React Native dan React JS sama-sama berasal dari JavaScript. Namun, React JS digunakan untuk pengembangan aplikasi berbasis web yang menarik dan interaktif.

Sama seperti React Native, React JS juga dikembangkan oleh Facebook. Tepatnya oleh seorang software engineer bernama Jordan Walke. Ia mengembangkan library ini pada tahun 2011. Akhirnya, dua tahun kemudian, React JS dipublikasikan untuk umum dan bersifat open source. Semenjak saat itu, React JS menjadi populer di kalangan developer. React js adalah sebuah library dari JavaScript yang biasa digunakan untuk merancang UI suatu aplikasi dan merupakan salah satu library JavaScript yang paling populer.

2.3.1 Konsep dan Fitur Utama React JS

1. **Berbasis Komponen** : React JS berpusat pada konsep komponen UI yang dapat digunakan kembali. Komponen dalam React adalah kelas atau fungsi JavaScript yang secara opsional menerima properti input (props) dan mengembalikan elemen React yang menjelaskan bagaimana bagian UI seharusnya muncul.
2. **Virtual DOM** : React JS menggunakan virtual DOM untuk mengoptimalkan pembaruan pada DOM yang sebenarnya. Saat status atau properti berubah, React membuat representasi virtual DOM baru dan membandingkannya dengan yang sebelumnya secara efisien. Kemudian, React hanya memperbarui bagian dari DOM yang perlu diubah, meminimalkan manipulasi DOM dan meningkatkan kinerja.
3. **JSX (JavaScript XML)** : JSX adalah ekstensi sintaksis untuk JavaScript yang memungkinkan pengembang untuk menulis kode mirip HTML langsung di dalam JavaScript. Ini membuat struktur dan logika komponen React lebih mudah dibaca dan intuitif.
4. **Pengikatan Data Satu Arah** : Data mengalir ke bawah hierarki komponen di React. Hal ini memastikan bahwa perubahan pada komponen anak tidak memengaruhi komponen induknya, sehingga aliran data tetap jelas dan dapat diprediksi.

5. **Deklaratif** : React JS menggunakan pendekatan deklaratif untuk menjelaskan bagaimana tampilan UI seharusnya. Pengembang menentukan seperti apa tampilan UI berdasarkan status terkini dan React secara efisien memperbarui dan merender komponen yang tepat saat status berubah
6. **Metode Siklus Hidup** : Komponen React memiliki metode siklus hidup yang dapat ditimpa oleh pengembang untuk menjalankan kode pada titik tertentu dalam masa pakai komponen. Misalnya, `componentDidMount()` dipanggil setelah komponen dirender untuk pertama kalinya.
7. **Hooks** : Diperkenalkan di React 16.8, hooks adalah fungsi yang memungkinkan pengembang untuk menggunakan status dan fitur React lainnya dalam komponen fungsional. Hooks seperti `useState` dan `useEffect` memungkinkan komponen fungsional untuk mengelola status komponen lokal dan menjalankan efek samping.

2.3.2 Kelebihan dan Kekurangan React JS

A. Kelebihan

1. **Mudah untuk dipelajari** : Ada banyak sekali tutorial React js di internet, sehingga memudahkan untuk pemula yang ingin mempelajari React js.
2. **Performa yang tinggi** : Dengan adanya Virtual DOM, React js bisa memperbarui data dengan seefisien mungkin. Selain itu, Virtual DOM juga memungkinkan pembuatan aplikasi menjadi lebih cepat dan responsive
3. **Komponen yang dapat digunakan kembali** : Pada React js, anda dapat menggunakan lagi komponen yang sudah anda buat. Sehingga dapat meningkatkan efisiensi dalam pengembangan kode
4. **JSX** : Adanya JSX pada React js memungkinkan Anda untuk memodifikasi DOM menggunakan kode html yang lebih sederhana. Akan tetapi, JSX bukanlah HTML. Bahasa sederhananya adalah seperti: JSX terlihat seperti HTML, tetapi memiliki fungsi seperti JavaScript
5. **Dapat digunakan untuk mengembangkan aplikasi mobile** : React js memiliki framework yang dapat digunakan untuk membuat aplikasi mobile disebut React Native. Dengan React Native, memungkinkan developer untuk membangun

sebuah aplikasi dengan berbasis system operasi Android dan IOS dengan menggunakan kode yang sama dengan aplikasi web.

B. Kekurangan

1. **Dokumentasi kurang baik** : Dokumentasi pada React js bisa dibilang kurang baik, karena informasi yang diberikan pada dokumentasi tidak lengkap. Tidak seperti framework lain yang dokumentasi nya lengkap dan terstruktur
2. **Kompatibilitas** : Browser versi lama tidak mendukung React js, hanya bisa menggunakan browser versi terbaru. Versi terbaru React js hanya mendukung Internet Explorer versi 9 keatas.
3. **Level kesulitan** : Meskipun React js mudah untuk dipelajari, namun penggunaan React js akan menyulitkan developer baru ketika mempelajari konsep dasarnya. Konsep dasar yang dimaksud disini seperti state, props, dan lifecycle.
4. **View Layer** : React js adalah library dengan bentuk view layer. Untuk membangun sebuah aplikasi yang besar kita harus menyusun sendiri tiap-tiap kebutuhan aplikasi yang lain, seperti data layer, router, struktur aplikasi dan event system (kecuali even DOM).

2.3.3 Modul - modul yang disediakan React JS

1. **Komponen** : React memungkinkan pengembang untuk membuat komponen yang dapat digunakan kembali. Komponen adalah blok bangunan dasar dari aplikasi React, yang dapat berupa komponen kelas atau komponen fungsi.
2. **Props** : Props (properties) adalah cara untuk mengirim data dari satu komponen ke komponen lainnya. Props bersifat read-only dan digunakan untuk mengonfigurasi komponen.
3. **State** : State adalah objek yang digunakan untuk menyimpan data yang dapat berubah seiring waktu. State memungkinkan komponen untuk merender ulang ketika data berubah. Penggunaan state sering dikombinasikan dengan hooks seperti `useState`.
4. **ContextAPI** : Context API memungkinkan pengembang untuk berbagi data antar komponen tanpa harus mengoper props secara manual di setiap level. Ini berguna untuk manajemen state global.

5. **Router** : React Router adalah modul yang memungkinkan pengembang untuk mengelola navigasi dalam aplikasi React. Ini memungkinkan pembuatan aplikasi satu halaman (SPA) dengan berbagai rute.

2.3.4 Prerequisite React JS

Prerequisite untuk menggunakan React.js mencakup beberapa pengetahuan dan keterampilan dasar yang perlu dimiliki oleh pengembang. Berikut adalah beberapa hal yang perlu dipahami sebelum mulai menggunakan React :

1. **HTML dan CSS** : Pengembang harus memiliki pemahaman yang baik tentang HTML dan CSS, karena React digunakan untuk membangun antarmuka pengguna yang berbasis pada elemen-elemen HTML. Memahami cara kerja elemen HTML dan bagaimana CSS digunakan untuk styling sangat penting.
2. **JavaScript dasar** : Pengetahuan dasar tentang JavaScript sangat penting, termasuk:
 - Sintaks dasar JavaScript
 - Tipe data, variabel, dan operator
 - Struktur kontrol (if, switch, loop)
 - Fungsi dan objek
3. **Pemrograman Berbasis Objek** : Memahami konsep dasar pemrograman berbasis objek (OOP) di JavaScript, seperti objek, kelas, dan inheritance, akan membantu dalam memahami cara kerja komponen di React.

2.3.5 Contoh Penerapan React JS

```
import React from 'react';
import ReactDOM from 'react-dom';

function App() {
  return (
    <div>
      <h1>Halo, ini aplikasi React!</h1>
    </div>
  );
}

ReactDOM.render(<App />, document.getElementById('root'));
```

2.4 jQuery

Bahasa pemrograman jQuery ditemukan oleh seorang pria bernama John Resig pada 2006. Pada tahun itulah jQuery ditetapkan sebagai library JavaScript yang bersifat lintas platform. Maksudnya apa? Saat itu, kerap terjadi inkonsistensi antara implementasi JavaScript di perangkat Firefox, Internet Explorer, dan Google Chrome (yang dirilis pada 2008). Dulu, ketika teknologi belum berkembang terlalu pesat seperti sekarang, kondisi tersebut mungkin masih dimaklumi. Apalagi menulis coding bukanlah pekerjaan yang mudah. Belum lagi jika ada begitu banyak string code yang harus ditambahkan.

jQuery adalah library JavaScript yang cukup andal, ringkas, dan mempunyai fitur yang cukup lengkap. Library ini membuat pemrosesan di HTML seperti perubahan dan manipulasi dokumen, *event handling*, animasi, dan AJAX JavaScript menjadi lebih sederhana. Hal ini didukung dengan API yang mudah digunakan dan dapat bekerja di berbagai macam browser. Tidak mengherankan jika hingga sekarang jQuery adalah salah satu library JavaScript yang paling populer atau paling banyak digunakan oleh seluruh web developer di seluruh dunia. Fungsi jQuery adalah sebagai library JavaScript yang akan membantu Anda mengatur interaksi antara JavaScript dengan HTML yang berjalan di sisi klien.

2.4.1 Konsep dan Fitur Utama jQuery

1. **Manipulasi DOM** : Pilih dan manipulasi elemen pada halaman web dengan mudah menggunakan pemilih seperti CSS.
2. **Penanganan peristiwa** : Sederhanakan pemasangan penanganan peristiwa ke elemen dan pengelolaan penyebaran peristiwa.
3. **Ajax** : Memperlancar pembuatan permintaan HTTP asinkron dengan fungsi Ajax seperti \$.ajax().
4. **Animasi** : Buat animasi dan efek halus dengan fungsi sederhana seperti .fadeIn(), .slideUp(), dll.
5. **Utilitas** : Mencakup berbagai fungsi utilitas untuk tugas-tugas umum, seperti \$.each() untuk mengulang objek dan array.
6. **Plugin** : Ekosistem plugin yang luas memperluas kemampuan jQuery untuk tugas-tugas seperti slider, validasi formulir, dan banyak lagi.

2.4.2 Kelebihan dan Kekurangan jQuery

A. Kelebihan

1. Memudahkan untuk mengakses elemen-elemen HTML
2. Penanganan event HTML
3. Manipulasi elemen HTML
4. Manipulasi CSS
5. Memudahkan penerapan efek-efek animasi dan JavaScript.

B. Kekurangan

1. jQuery memiliki ukuran yang relatif besar untuk sebuah library. Ini dapat memperlambat waktu muat halaman web, terutama jika hanya digunakan untuk beberapa fitur sederhana.
2. Meskipun jQuery berusaha untuk kompatibel dengan berbagai browser, beberapa fitur mungkin tidak berfungsi dengan baik di browser modern yang mendukung standar terbaru.
3. Beberapa pengembang mengandalkan jQuery terlalu banyak, bahkan untuk tugas-tugas yang bisa diatasi dengan JavaScript murni.
4. Dalam beberapa kasus, menggunakan JavaScript murni lebih cepat daripada menggunakan jQuery.

2.4.3 Modul - modul yang disediakan jQuery

1. **Manipulasi DOM** : Pilih dan manipulasi elemen pada halaman web dengan mudah menggunakan pemilih seperti CSS.
2. **Penanganan peristiwa** : Sederhanakan pemasangan penanganan peristiwa ke elemen dan pengelolaan penyebaran peristiwa.
3. **Ajax** : Memperlancar pembuatan permintaan HTTP asinkron dengan fungsi Ajax seperti \$.ajax().
4. **Animasi** : Buat animasi dan efek halus dengan fungsi sederhana seperti .fadeIn(), .slideUp(), dll.
5. **Utilitas** : Mencakup berbagai fungsi utilitas untuk tugas-tugas umum, seperti \$.each() untuk mengulang objek dan array.
6. **Plugin** : Ekosistem plugin yang luas memperluas kemampuan jQuery untuk tugas-tugas seperti slider, validasi formulir, dan banyak lagi.

2.4.4 Prerequisite jQuery

1. Pengetahuan Dasar:

Anda sebaiknya memiliki pemahaman dasar tentang:

- a. HTML: Struktur halaman web.
- b. CSS: Styling elemen HTML.
- c. JavaScript: Dasar-dasar bahasa pemrograman JavaScript.

2. Browser Compatibility:

- a. jQuery mendukung berbagai browser, tetapi pastikan Anda menggunakan versi jQuery yang kompatibel dengan browser yang ingin Anda dukung.
- b. Beberapa versi jQuery mungkin tidak berfungsi dengan baik di browser modern yang mendukung standar terbaru.

3. Penggunaan jQuery Itself:

- a. Tidak ada prasyarat khusus untuk menggunakan jQuery itu sendiri. Yang Anda butuhkan hanyalah implementasi JavaScript di browser.

4. Penggunaan Plugin atau Add-Ons:

- a. Jika Anda menggunakan plugin atau add-ons khusus yang memanfaatkan jQuery, periksalah apakah ada prasyarat khusus yang dinyatakan oleh plugin tersebut.

2.4.5 Contoh Penerapan jQuery

<!DOCTYPE html>

```
<html>
<head>
  <title>Contoh jQuery</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <script>
    $(document).ready(function() {
      $("#tombol").click(function() {
        $("#pesan").fadeIn();
      });
    });
  </script>
</head>
<body>
  <button id="tombol">Tampilkan Pesan</button>
  <div id="pesan" style="display:none;">Halo, ini pesan jQuery!</div>
</body>
</html>
```


2.5 Chart.JS

Chart.js adalah sebuah library JavaScript yang digunakan untuk membuat grafik dan chart interaktif di halaman web. Ini adalah alat yang kuat dan fleksibel untuk visualisasi data, dan banyak digunakan oleh pengembang web untuk menyajikan data dalam bentuk grafik yang lebih mudah dimengerti dan atraktif bagi pengguna. Chart.js menyediakan berbagai jenis grafik, termasuk bar chart, line chart, pie chart, doughnut chart, radar chart, dan sebagainya.

2.5.1 Konsep dan Fitur Utama Chart.JS

Chart.js dirancang untuk memudahkan pengembang dalam membuat grafik interaktif dan responsif. Dengan menggunakan API yang sederhana, Chart.js memungkinkan pengguna untuk menampilkan data dalam berbagai bentuk visual, seperti grafik garis, batang, lingkaran, dan lainnya. Library ini sangat berguna dalam konteks analisis data, pelaporan, dan dashboard aplikasi.

1. **Responsif** : Grafik yang dibuat dengan Chart.js dapat disesuaikan secara otomatis untuk berbagai ukuran layar, menjadikannya ideal untuk aplikasi web yang responsif.
2. **Animasi** : Chart.js menyediakan opsi untuk menambahkan animasi pada grafik, memberikan pengalaman visual yang lebih menarik bagi pengguna.
3. **Kustomisasi** : Pengguna dapat dengan mudah menyesuaikan tampilan grafik dengan berbagai opsi konfigurasi, termasuk warna, label, dan skala.
4. **Plugin** : Chart.js mendukung penggunaan plugin untuk menambahkan fungsionalitas tambahan, seperti anotasi dan label data, yang dapat meningkatkan interaktivitas grafik.
5. **Beragam Tipe Grafik** : Chart.js mendukung berbagai jenis grafik,

2.5.2 Kelebihan dan Kekurangan Chart.JS

A. Kelebihan

1. Menyediakan grafik responsif dan animasi dengan tampilan & nuansa yang bersih
2. Pustaka inti ringan: 199kb diperkecil dan 67kb di-Gzipped menurut Bundlephobia

3. Sumber terbuka gratis, dengan lisensi MIT yang permisif
4. Sederhana, mudah digunakan dan memulai
5. Dokumentasi & contoh yang bagus untuk pustaka sumber terbuka
6. Populer, dengan komunitas besar dan banyak kontributor
7. Dapat diperluas, melalui arsitektur plugin yang memungkinkan jenis bagan, fitur, atau interaksi tambahan untuk dibuat.

B. Kekurangan

1. Hanya 8 jenis dan fitur grafik dasar yang tersedia di perpustakaan inti
2. Jenis dan fitur grafik tambahan memerlukan plugin yang mungkin tidak memiliki kecepatan pengembangan atau komunitas yang sama dengan pustaka inti utama
3. Tidak mendukung jenis grafik lanjutan seperti peta panas
4. Menambahkan hal-hal seperti interaktivitas memerlukan plugin
5. Tidak ada dukungan untuk grafik 3D
6. Menurut penelusuran internet, Chart.js lambat dan mengalami kendala kinerja dengan kumpulan data besar. Misalnya, beberapa masalah Github & posting Stackoverflow melaporkan:
 - “Chart.js sangat lambat” dan tidak dapat digunakan dengan angka besar tetapi kumpulan data kecil
 - “Chart.js sangat lambat” dengan banyak titik data
 - “Grafik garis js dengan data besar berubah ukuran dengan sangat lambat”
 - “Kecepatan zoom pada data besar lambat dengan Chart.js”
7. Pembaruan waktu nyata atau dinamis merupakan tantangan, dengan masalah kinerja yang dilaporkan di sini juga
8. Terkadang masalah rendering karena ketidakstabilan API atau keluhan dengan API dilaporkan dalam ulasan G2
9. Open source berarti kurangnya dukungan teknis perusahaan, meskipun seperti yang disebutkan sebelumnya komunitasnya kuat

2.5.3 Modul - modul yang disediakan Chart.JS

Chart.js menyediakan berbagai modul dan fitur yang memungkinkan pengembang untuk membuat grafik dan visualisasi data dengan mudah. Berikut adalah beberapa modul utama yang disediakan oleh Chart.js:

- 1. Tipe Grafik :** Chart.js mendukung berbagai jenis grafik, termasuk:
 - Grafik Garis: Untuk menampilkan data yang berubah seiring waktu.
 - Grafik Batang: Untuk membandingkan nilai di antara kategori.
 - Grafik Lingkaran dan Doughnut: Untuk menunjukkan proporsi dari keseluruhan.
 - Grafik Radar: Untuk membandingkan beberapa variabel.
 - Grafik Polar Area: Mirip dengan grafik lingkaran, tetapi dengan area yang berbeda-beda.
 - Grafik Scatter: Untuk menampilkan nilai dalam dua dimensi.
- 2. Responsif :** Grafik yang dibuat dengan Chart.js dapat disesuaikan secara otomatis untuk berbagai ukuran layar, menjadikannya ideal untuk aplikasi web yang responsif.
- 3. Animasi :** Chart.js menyediakan opsi untuk menambahkan animasi pada grafik, memberikan pengalaman visual yang lebih menarik bagi pengguna.
- 4. Kustomisasi :** Pengguna dapat menyesuaikan tampilan grafik dengan berbagai opsi konfigurasi, termasuk warna, label, dan skala. Ini memungkinkan penyesuaian yang mendalam sesuai dengan kebutuhan visualisasi data.
- 5. Plugin :** Chart.js mendukung penggunaan plugin untuk menambahkan fungsionalitas tambahan, seperti anotasi dan label data, yang dapat meningkatkan interaktivitas grafik.

2.5.4 Prerequisite Chart.JS

Untuk menggunakan Chart.js, ada beberapa prerequisite yang perlu dipahami oleh pengembang. Berikut adalah ringkasan dari persyaratan tersebut:

1. Pengetahuan dasar JavaScript :

Pengembang harus memiliki pemahaman yang baik tentang JavaScript, terutama konsep dasar seperti variabel, fungsi, dan objek. Ini penting karena Chart.js

adalah *library* JavaScript yang memerlukan pengetahuan dasar tentang bahasa ini untuk integrasi dan penggunaan yang efektif.

2. HTML dan CSS :

Memahami HTML dan CSS sangat penting, karena Chart.js digunakan untuk menggambar grafik di dalam elemen `<canvas>` HTML. Pengetahuan tentang cara menyusun halaman web dan menerapkan gaya akan membantu dalam mengintegrasikan grafik dengan baik ke dalam antarmuka pengguna.

3. Penggunaan NPM (Node Package Manager)

Jika menggunakan Chart.js dalam proyek yang lebih besar atau dengan framework seperti React atau Angular, pemahaman tentang NPM untuk mengelola paket dan dependensi sangat diperlukan.

4. Dasar-dasar Command Line

Kemampuan untuk menggunakan command line atau terminal untuk menjalankan perintah dasar, seperti menginstal paket dan menjalankan aplikasi, juga diperlukan.

5. Pemahaman tentang Data dan Struktur Data

Memahami cara menyusun dan mengelola data yang akan digunakan dalam grafik sangat penting. Ini termasuk pemahaman tentang format data seperti JSON, yang sering digunakan untuk mengirim data ke Chart.js.

2.5.5 Contoh Penerapan Chart.JS

```
<!DOCTYPE html>
<html>
<head>
  <title>Contoh Chart.js</title>
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>
<body>
  <canvas id="myChart" width="400" height="200"></canvas>
  <script>
    const ctx = document.getElementById('myChart').getContext('2d');
    const myChart = new Chart(ctx, {
```

```

type: 'bar',
data: {
  labels: ['Merah', 'Biru', 'Kuning', 'Hijau', 'Ungu', 'Oranye'],
  datasets: [{
    label: '# of Votes',
    data: [12, 19, 3, 5, 2, 3],
    backgroundColor: [
      'rgba(255, 99, 132, 0.2)',
      'rgba(54, 162, 235, 0.2)',
      'rgba(255, 206, 86, 0.2)',
      'rgba(75, 192, 192, 0.2)',
      'rgba(153, 102, 255, 0.2)',
      'rgba(255, 159, 64, 0.2)'
    ],
    borderColor: [
      'rgba(255, 99, 132, 1)',
      'rgba(54, 162, 235, 1)',
      'rgba(255, 206, 86, 1)',
      'rgba(75, 192, 192, 1)',
      'rgba(153, 102, 255, 1)',
      'rgba(255, 159, 64, 1)'
    ],
    borderWidth: 1
  }]
},
options: {
  scales: {
    y: {
      beginAtZero: true
    }
  }
}

```

```
});  
</script>  
</body>  
</html>
```

2.6 Perbedaan utama antara framework dan library

Perbedaan utama antara framework dan library terletak pada inversi kontrol. Saat menggunakan library, kontrol tetap ada pada developer sehingga dapat menentukan kapan aplikasi kapan harus menggunakan fungsi library. Sebaliknya, framework memiliki kontrol yang akan memberi tahu developer di mana kode harus disediakan dan menggunakannya sesuai kebutuhan.

Lebih lanjut, berikut perbedaan framework dan library dirangkum dari Geeks for Geeks:

1. Framework terdiri dari banyak API, program pendukung, library, dan compiler. Library merupakan kumpulan helper module, class, object, function, dan pre-written code.
2. Framework sulit diganti. Sedangkan Library mudah diganti dengan library lain.
3. Tidak mungkin memasukkan framework ke dalam proyek yang sudah ada. Library dapat dengan mudah diintegrasikan ke dalam proyek yang sudah ada untuk menambahkan beberapa fungsi tertentu.
4. Mengembangkan framework membutuhkan banyak kode sehingga waktu muat (load time) lebih lama.
5. Library memerlukan lebih sedikit kode yang tidak memerlukan load time terlalu lama

BAB III

PENUTUP

3.1 Kesimpulan

Makalah ini membahas perkembangan dan relevansi JavaScript sebagai bahasa pemrograman yang telah ada lebih dari 20 tahun. Dengan pesatnya perkembangan teknologi, khususnya dalam ekosistem JavaScript, penting bagi pengembang untuk terus mengikuti tools, library, dan framework terbaru untuk meningkatkan efisiensi dan keterampilan mereka.

JavaScript memiliki berbagai library dan framework yang dapat membantu dalam pengembangan aplikasi web. Beberapa library populer termasuk React.js, jQuery, dan Chart.js. Setiap library dan framework memiliki kelebihan dan kekurangan.

Selain itu, makalah ini juga menjelaskan perbedaan utama antara framework dan library, di mana framework memberikan kontrol lebih besar atas struktur aplikasi, sedangkan library memungkinkan pengembang untuk menggunakan fungsionalitas spesifik tanpa mengubah struktur aplikasi yang ada.

Secara keseluruhan, pengembang harus mempertimbangkan kelebihan dan kekurangan masing-masing library dan framework dalam konteks kebutuhan proyek mereka. Dengan memahami karakteristik masing-masing, pengembang dapat memilih alat yang paling sesuai untuk mencapai hasil yang optimal dalam pengembangan aplikasi web.

3.2 Saran

Dalam memilih antara React.js, jQuery, atau Chart.js, pengembang harus mempertimbangkan kebutuhan spesifik proyek. Misalnya, jika proyek memerlukan pembuatan UI interaktif dan dinamis, React.js adalah pilihan yang tepat karena kemampuannya untuk menangani perubahan UI secara efisien dengan Virtual DOM. Sebaliknya, jika fokus proyek adalah pada visualisasi data, Chart.js menyediakan berbagai jenis grafik yang dapat mempermudah presentasi data.

DAFTAR PUSTAKA

NOSUKE (2022, 03 31). “Apa perbedaan Framework dan Library? Yuk Cari Infonya!”.

(23 Juli 2024) Diakses dari

<https://appkey.id/pembuatan-website/web-programming/framework-2/>

PT BIO GIZNOT Nusantara. “Mengenal JQuery , fungsi dan contoh penggunaanya”.

(25 Juli 2024) Diakses dari

<https://www.biznetgio.com/news/apa-itu-jquery>

Perbedaan.co.id (2024, 04 17). “perbedaan framework dan library”.

(25 Juli 2024) Diakses dari

<https://www.perbedaan.co.id/perbedaan-framework-dan-library/>

Ratna Patria (2020 05 23). “Apa itu jQuery? Ketahui Kelebihan dan Kekurangannya”.

(25 Juli 2024) Diakses dari

<https://www.domainsia.com/berita/apa-itu-jquery/>

Bima Bayu Saputra (2024). “Pengembangan Website Dokumentasi Chart.js dan laravel untuk menampilkan data”.

(25 Juli 2024) Diakses dari

<https://etd.umy.ac.id/id/eprint/44524/>