



E-Book

Fullstack

EmpoweringITPeople

Pendahuluan

Semakin bertambahnya jaman semakin berkembang kemajuan teknologi yang memungkinkan kehidupan manusia untuk menjadi lebih mudah. Dengan semakin mempermudah kehidupan manusia dan membuat semakin simple pekerjaan manusia, terciptalah teknologi baru yang membutuhkan skill lebih dari teknologi sebelumnya. Bahasa pemrograman pun semakin banyak terbuat dan tercipta untuk memenuhi kebutuhan teknologi masa kini, seperti teknologi *Web*, *Mobile*, *IoT* dan lain lain.

Bahasa Javascript merupakan Bahasa yang sudah ada namun sekarang menjadi naik kembali trennya di karenakan dapat di aplikasikan tidak hanya 1 platform yang Bernama Web, melainkan bisa di *convert* atau dirubah menjadi *platform Mobile*, *IoT*, *Cloud* dan lain lain. Sehingga demand atau permintaan akan *developer Javascript* semakin naik dan semakin dibutuhkan.

Dengan adanya perkembangan *Javascript* yang sangat luas dan banyak, maka banyak para *developer* pemula mengalami kesulitan ingin memulai dari mana. Untuk mempermudah pembelajaran, *Fullstack Js* ini dibagi menjadi 3 bagian utama:

1. *Javascript Basic*
 - Mempelajari basic IT skills seperti Algoritma dan *Pseudocode*, *Variable* dan *Tipe Data*, *Conditional*, *Looping*, *Array*, *Function*, *Object*, *ES6*, *OOP* dan *MVC* serta basic lainnya
2. *Back End Javascript*
 - Mempelajari teknologi *ter up to date* dalam *Back End* dan menggunakan framework
3. *Front End*
 - Mempelajari interface dan tampilan UI untuk web serta menggunakan framework.

Javascript Basic

Setiap Langkah pasti di mulai dari Langkah kecil dahulu, demikian juga untuk mempelajar dalam dunia pemrograman dengan Bahasa Javascript. Dalam basic ini akan di perkenalkan mengenai dasar-dasar dari programming hingga ke konsep Intermediate.

Algoritma dan Pseudocode

Algoritma adalah sebuah proses menemukan solusi dari sebuah masalah dan menjabarkan nya menjadi Langkah demi Langkah

Cara menemukan Algoritma

1. Temukan permasalahan
2. Buat Langkah penyelesaian
3. Buat kesimpulan

Contoh:

- Buat algoritma untuk membuat mie instans
 1. Buka bungkus mie isntan
 2. Tuangkan bumbu ke dalam mangkok
 3. Rebus air
 4. Setelah mendidih, letakkan mie
 5. Tiriskan jika sudah matang
 6. Letakan di mangkok lalu aduk
 7. Selamat makan

Pseudocode adalah sebuah proses mengubah algoritma menjadi Bahasa yang lebih terstruktur.

Contoh :

Rumus dari luas persegi Panjang adalah Panjang x lebar. Tentukan algoritma dan pseudocode untuk menyelesaikan masalah tsb.

ALGORITMA:

1. Buat *variable* Panjang
2. Buat *variable* lebar
3. Buat *variable* luas dan isi dengan rumus Panjang x
4. Proses luas
5. Tampilkan

PSEUDOCODE

```

STORE Panjang WITH ANY NUMBER
STORE lebar WITH ANY NUMBER

SET luas WITH Panjang MULTIPLY lebar

DISPLAY luas

```

Variable dan Tipe Data

Variable adalah sebuah penampung untuk menyimpan sebuah nilai atau value.

Tipe Data adalah tipe dari sebuah nilai atau value dalam sebuah variable. Sama seperti sebuah ember yang dapat diisi oleh beberapa jenis air, demikian juga tipe data seperti itu analoginya

Dalam Javascript terdapat 6 tipe data Primitif

1. Number
2. String
3. Boolean
4. Object
5. Undefined
6. Null

Operasi dalam Js:

- A. Operasi Math
 - Operasi Tambah, Kurang, Kali, Bagi, dan Modulus
- B. Operasi Logic
 - Operasi yang berhubungan dengan logika math.
Seperti *EQUAL(==)*, *NOT(!=)*

Conditional

Conditional merupakan sebuah konsep dimana program akan mengambil dari beberapa kondisi dan menjalankannya.

Contoh:

```

Var x = 10;

If (x === 10) {

    Console.log("Angka 10");

}

```

Maka hasilnya adalah "Angka 10".

Karena sesuai kondisi *true* maka akan menjalankan kode di bawahnya.

Looping

Looping merupakan sebuah konsep dimana program akan melakukan Tindakan berulang hingga titik tertentu atau Ketika kondisi false.

Contoh:

```
For(var I =0; i< 5; i++) {  
    Console.log(i);  
}
```

Maka akan menghasilkan :

```
0  
1  
2  
3  
4
```

Proses tersebut dilakukan Ketika I lebih kecil dari 5 dan hasilnya true. Maka akan menjalankan kode di bawah nya.

Array

Array adalah kumpulan value dalam 1 variable yang memiliki index secara berurutan. Sehingga untuk mengakses salah satu dari value tersebut menggunakan index.

Contoh:

```
Var numbers = [1,2,3,4,5];  
  
Var fruits = ["apple","pear","grapes"]  
  
fruits[0] berisi "apple"
```

dan seterusnya.

Cara untuk menghitung jumlah panjang dari sebuah array dapat menggunakan

Function

Function adalah sekumpulan kode yang di jalankan untuk melakukan suatu task tertentu. Efektivitas dari sebuah function di lihat dari task yang di kerjakan, dan hanya 1 task yang di kerjakan.

Contoh:

```
function checkGanjilGenap(num) {
    If (num %2 === 0) {
        Console.log("ganjil")
    } else {
        Console.log("genap")
    }
}
```

Dalam javascript terdapat 3 jenis pembuatan function.

1. Function Declaration
2. Function Expression
3. Arrow Function

Object

Object adalah kumpulan nilai mirip seperti Array tetapi memiliki key dan method. Key digunakan sebagai pengganti index, sehingga lebih detail dan lebih mudah untuk dikenali. Method merupakan function yang berada dalam object.

Contoh:

```
var user = {
    name:"Admin",
    status:"admin"
};

console.log(user.name) // Admin
```

```
console.log(user.status) // admin
```

ECMAScript6

ES6 merupakan versi terbaru JS yang diciptakan pada tahun 2015. Ada beberapa fitur perubahan yang dibuat dalam ES6 ini, seperti:

1. Perbedaan antara var, let, dan const
2. Template Literal
3. Function Arrow
4. High Order Function
5. Destructuring Array dan

Object Oriented Programming

OOP adalah paradigma pemrograman dengan menggunakan object.

Istilah dalam OOP:

- Class: Blueprint dari object
- Key/Attribute dan Method: Berfungsi sama seperti key dan method dalam object
- Instance: Merupakan hasil dari object yang sudah jadi dari class

Class adalah blueprint dari object.

Contoh: Denah rumah, tetapi belum jadi rumah secara fisik.



Class merupakan bagian dari ES6.

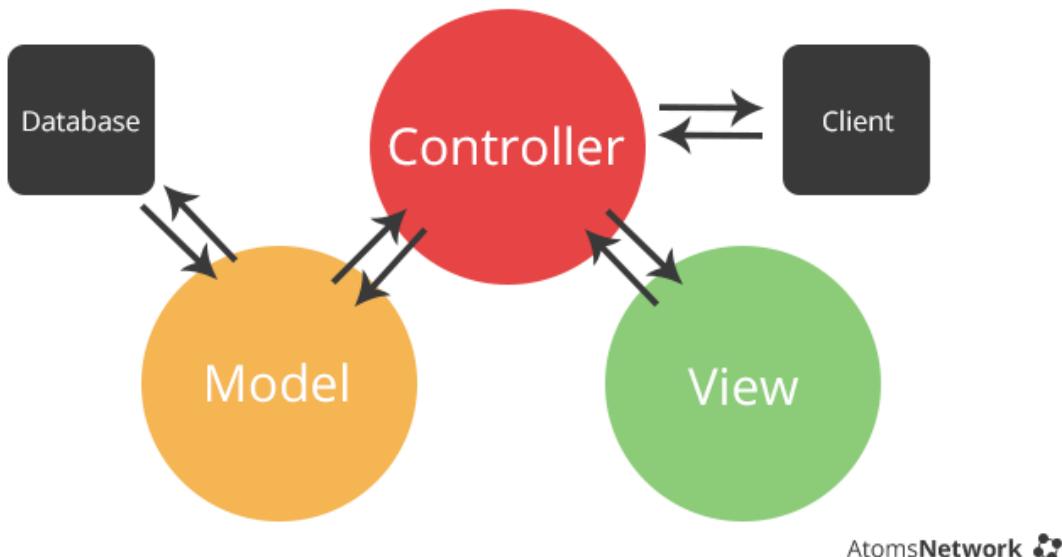
Dalam class terdapat attribute/key dan juga method sama seperti object.

Dalam OOP juga terdapat teiru utama yang sering di sebut dengan 4 Pilar Utama :

1. *Inheritance*
 - Merupakan teknik untuk membuat class memiliki turunannya.
2. *Polymorphism*
 - Merupakan teknik yang mirip dengan Inheritance tetapi memiliki banyak rupa
3. *Encapsulation*
 - Class memiliki akses *level*: *Public*, *Protected* dan *Private*.
4. *Abstraction*
 - Menyembunyikan informasi yang tidak perlu.

Ada juga beberapa istilah dalam OOP yang harus di ketahui

Model View Controller



Design pattern merupakan metode untuk menyelesaikan permasalahan yang biasanya berulang/memiliki pola dalam pengembangan software.

Design Pattern bukan merupakan metode yang langsung dapat diubah menjadi kode program melainkan sebuah pola dasar atau template dari cara untuk menyelesaikan suatu masalah dalam situasi yang berbeda-beda.

Design Pattern membantu mempercepat pengembangan software karena pola yang dijelaskan merupakan paradigma yang telah teruji kegunaannya.

Jenis Design Pattern

1. *Creational design pattern* — berhubungan dengan instansiasi suatu class.
2. *Structural design pattern* — berhubungan dengan komposisi class dan object.
3. *Behavioral design pattern* — berhubungan dengan komunikasi antarobject.

BACK END

API = Application Programming Interface;

software yang memungkinkan para *developer* untuk mengintegrasikan dua aplikasi yang berbeda secara bersamaan untuk saling terhubung satu sama lain.

REST API merupakan salah satu dari desain arsitektur yang terdapat di dalam *API* itu sendiri. Cara kerja *RESTful API* yaitu: *REST client* akan melakukan akses pada data/resource *REST* server di mana masing-masing resource tersebut akan dibedakan oleh URL.

Fitur yang bisa digunakan pada *Web API* dalam mengembangkan aplikasi:

1. Mendukung fungsi *CRUD* yang bekerja melalui protokol *HTTP* dengan *method GET, POST, PUT dan DELETE*.
2. Memiliki *response Accept Header* dan *HTTP status code*.
3. Mendukung banyak format teks, Seperti *JSON*, *XML* atau format lain yang diinginkan. Kebanyakan format *JSON* lah yang dipakai.
4. Mendukung fitur *MVC* seperti *routing*, *controllers*, *action results*, *filter*, *model*, *IOC container*, dan lainnya.
5. Dapat berjalan di *Apache* atau *server web* lainnya yang didukung sesuai bahasa pemrograman yang digunakan.

Express.js

Adalah *framework Node.JS* yang dirancang secara fleksibel dan sederhana untuk membantu tahap pengembangan aplikasi back end.

Express.JS juga dapat digunakan untuk kebutuhan aplikasi Representational State Transfer (*RESTful*) yang dimanfaatkan untuk pengembangan desain aplikasi network. Aplikasi *RESTful* memakai *request HTTP* untuk melakukan operasi *CRUD* (*Create, Read, Update, dan Delete*).

Keuntungan yang lain dari pemanfaatan *RESTful* ini adalah bisa membangun berbagai macam teknologi client – side seperti pada *website*, *mobile*, atau pun *desktop*.

Link: <https://expressjs.com/>

Sequelize ORM

Object Relational Mapping (ORM) = teknik untuk mengubah data dari database ke dalam object yang sesuai dengan bahasa yang dipakai, sehingga mudah untuk digunakan.

Sequelize adalah salah satu ORM yang sering digunakan untuk membantu *developer*.

Sequelize adalah *promised-based node.js* yang bisa melakukan *transactions*, seperti *select, insert, update, dan delete tanpa query manual*, sehingga memudahkan.

Link : <https://sequelize.org>

FRONT END

Pengertian Front End

Front End dan *Back End* adalah dua hal yang berkaitan dengan bagaimana sebuah *website* maupun aplikasi dapat bekerja dan diakses oleh pengguna.

Berkaitan dengan proses *web development*, *front end* adalah apa yang pengguna lihat pada tampilan sebuah *website*. Sedangkan *back end* adalah sistem di balik layar yang mengolah *database* dan juga server.

Bagian *front end* sering pula disebut sebagai "*client-side*" dan *back-end* disebut dengan "*server-side*".

Orang yang mengelola *front end* dan *back end* biasa disebut dengan panggilan Full Stack Developer, sedangkan apabila hanya salah satunya saja, dapat disebut dengan Front End Developer dan Back End Developer.

Dari segi peranan, Front End Developer berperan mengembangkan tampilan situs dengan menggunakan bahasa pemrograman seperti CSS (*Cascading Style Sheets*), HTML (*Hypertext Markup Language*), dan Javascript. Sedangkan untuk Back End Developer bertugas memastikan bahwa sebuah situs dapat berfungsi dan diakses melalui *monitoring* "di balik layar".

Tugas Front End:

- Mengoptimalkan *user experience* supaya program dapat diakses oleh pengguna dengan cepat (selanjutnya tugas ini dibebankan pada back end developer).
- Menggunakan *HTML*, *JavaScript*, dan *CSS* sebagai "senjata" andalan dalam membuat tampilan program aplikasi atau *website* yang *user friendly*.
- Mengembangkan dan *maintaining user interface* agar grafis program selalu berkualitas tinggi dan konsistensi brand.
- Ikut merancang fitur program berbasis mobile dan device lainnya.
- Menerapkan desain responsif untuk program di semua device.
- Membuat tools yang meningkatkan interaksi situs terlepas dari browser.
- Mengelola alur kerja software.
- Mengikuti pedoman praktik terbaik SEO.
- Memperbaiki bug.
- Berkolaborasi dengan *back end developer* dan *web designer* untuk melakukan *usability testing* (pengujian kegunaan program) dan meningkatkan kegunaannya.

Pengertian HTML dan Sejarahnya

Hypertext Markup Language atau **HTML** adalah bahasa markup standar yang digunakan untuk membuat halaman website dan aplikasi web.

Sejarah HTML dimulai oleh [**Tim Berners-Lee**](#), seorang ahli fisika di lembaga penelitian CERN yang berlokasi di Swiss. Versi pertamanya dirilis pada tahun 1991, dengan 18 tag. Sejak saat itu, setiap kali ada versi barunya, pasti akan selalu ada tag dan attribute (tag modifier) yang juga baru.

Ketika bekerja dengan bahasa markup ini , Anda menggunakan struktur kode sederhana (tag dan attribute) untuk *mark up* halaman website. Misalnya, Anda membuat sebuah paragraf dengan menempatkan *enclosed text* di antara tag pembuka **<p>** dan tag penutup **</p>**.

```
<p>This is how you add a paragraph in HTML.</p>
<p>You can have more than one!</p>
```

Berkat popularitasnya yang terus meningkat, bahasa markup ini kini dianggap sebagai standar web resmi. Spesifikasi HTML dikelola dan dikembangkan oleh World Wide Web Consortium (W3C). Berdasarkan [**HTML Element Reference**](#) milik Mozilla Developer Network, saat ini ada 140 tag HTML, meskipun sebagiannya sudah tidak lagi didukung oleh beberapa versi terbaru browser.

Kemudian, pada tahun 2014, **HTML5** mulai diperkenalkan. Dengan diperkenalkannya HTML5, terdapat semantic baru seperti **<article>**, **<header>**, dan **<footer>** yang memperjelas bagian-bagian dalam konten. Nah, setelah Anda tahu apa itu HTML, selanjutnya kami akan membahas cara kerjanya.

Bagaimana Cara Kerja HTML ?

File HTML diakhiri dengan ekstensi **.html atau .htm**. Ekstensi file ini bisa dilihat dengan menggunakan web browser apa pun (seperti Google Chrome, Safari, atau Mozilla Firefox). Browser tersebut membaca file HTML dan merender kontennya sehingga user internet bisa melihat dan membacanya.

Biasanya, rata-rata situs web menyertakan sejumlah halaman HTML yang berbeda-beda. Contohnya, halaman beranda, 'Tentang Kami', halaman kontak memiliki dokumennya sendiri-sendiri.

Masing-masing halaman tersebut terdiri atas serangkaian tags (bisa disebut juga elements), yang tersusun untuk membentuk sebuah halaman website. Tag tersebut membuat hierarki yang menyusun konten hingga menjadi bagian, paragraf, heading, dan block konten lainnya.

Sebagian besar elemen bahasa markup ini memiliki tag pembuka dan penutup yang menggunakan syntax `<tag></tag>`.

Berikut contoh kode dari susunan atau struktur HTML:

```
<div>
  <h1>The Main Heading</h1>
  <h2>A catchy subheading</h2>
  <p>Paragraph one</p>
  
  <p>Paragraph two with a <a
    href="https://example.com">hyperlink</a></p>
</div>
```

- Elemen teratas dan terbawah adalah division sederhana (`<div></div>`) yang bisa Anda gunakan untuk mark up bagian konten yang lebih besar.
- Susunan di atas terdiri atas heading (`<h1></h1>`), subheading (`<h2></h2>`), dua paragraf (`<p></p>`), dan satu gambar (``).
- Paragraf kedua meliputi sebuah link (`<a>`) dengan attribute href yang terdiri atas URL tujuan.
- Tag gambar memiliki dua attribute, **src** untuk path gambar dan **alt** untuk deskripsi gambar.

Pengertian Tag HTML dan Fungsinya

Tag HTML memiliki dua tipe utama: ***block-level*** dan ***inline tags***.

- Elemen *block-level* memakai semua ruang yang tersedia dan selalu membuat line baru di dalam dokumen. Contoh dari tag *block* adalah heading dan paragraf.
- Elemen *inline* hanya memakai space sesuai dengan kebutuhannya dan tidak membuat line baru di halaman. Biasanya elemen ini akan memformat isi konten dari elemen *block-level*. Contoh dari tag inline adalah link dan *emphasized strings*.

Tag Block-Level

Tiga tag block-level yang harus dimiliki oleh setiap dokumen HTML adalah **<html>**, **<head>**, dan **<body>**.

- Tag **<html></html>** adalah elemen level tertinggi yang menyertakan setiap halaman HTML.
- Tag **<head></head>** menyimpan informasi meta, seperti judul dan *charset* halaman.
- Tag **<body></body>** melampirkan semua konten yang muncul pada suatu halaman

```
<html>
  <head>
    <!-- META INFORMATION -->
  </head>
  <body>
    <!-- PAGE CONTENT -->

  </body>
</html>
```

- Heading memiliki 6 level. Level tersebut bervariasi, mulai dari `<h1></h1>`sampai `<h6></h6>`, di mana h1 merupakan level heading tertinggi dan h6 adalah level terendah. Paragraf dibuka dan ditutup dengan tag `<p></p>`, sedangkan blockquote menggunakan tag `<blockquote></blockquote>`.
- Division merupakan bagian konten yang lebih besar dan biasanya terdiri atas beberapa paragraf, gambar, kadang-kadang blockquote, dan elemen lebih kecil lainnya. Kita bisa membuat *mark up* dengan menggunakan tag `<div></div>`. Di dalam elemen div juga terdapat tag div lainnya.
- Anda juga bisa menggunakan tag `` untuk list yang berurutan dan `` untuk list yang tidak berurutan. Masing-masing list item harus dibuka dan ditutup dengan tag ``. Sebagai contoh, di bawah ini adalah tampilan dasar daftar tidak berurutan dalam HTML:

```
<ul>
    <li>List item 1</li>
    <li>List item 2</li>
    <li>List item 3</li>
</ul>
```

Tag Inline

Sebagian besar tag inline digunakan untuk memformat teks. Sebagai contoh, tag `` akan render elemen ke format **bold**, sedangkan tag `` akan ditampilkan dalam format *italic*.

Hyperlink adalah elemen inline yang wajibkan adanya tag `<a>` dan attribute `href` untuk mengindikasi tujuan link:

```
<a href="https://example.com/">Click me!</a>
```

Gambar (image) juga merupakan elemen inline. Anda dapat menambahkan satu gambar dengan menggunakan **** tanpa harus membubuhkan tag penutup. Hanya saja, Anda disarankan menggunakan attribute **src** untuk menentukan path gambar, misalnya:

```

```

Pengertian CSS

CSS adalah singkatan dari *cascading style sheets*, yaitu bahasa yang digunakan untuk menentukan tampilan dan format halaman website. Dengan CSS, Anda bisa mengatur jenis font, warna tulisan, dan latar belakang halaman.

CSS digunakan bersama dengan bahasa markup, seperti [HTML](#) dan [XML](#) untuk membangun sebuah website yang menarik dan memiliki fungsi yang berjalan baik.

CSS juga berguna untuk mengatasi keterbatasan HTML dalam mengatur format halaman website. Kenapa demikian?

Apabila hanya menggunakan HTML ketika membangun website dengan beberapa halaman, Anda harus menulis tag untuk sebuah elemen HTML di semua halaman tersebut.

Jenis Jenis CSS

Jenis jenis CSS dibagi menjadi tiga berdasarkan penempatan kodennya, yaitu:

1. Inline CSS

Inline CSS adalah kode CSS yang dituliskan di dalam file HTML. Jenis CSS ini hanya mempengaruhi satu baris kode HTML.

Perhatikan baris kode di bawah ini sebagai contohnya:

```
<h1 style="font-size:30px;color:blue;">Cek beritama utama  
ini!</h1>
```

Dengan kode tersebut, Anda akan mendapatkan hasil ini:

Cek beritama utama ini!

Inline CSS tidak bisa diaplikasikan ke semua halaman website sekaligus. Namun, jenis CSS ini tepat digunakan ketika Anda ingin membuat elemen HTML dengan format khusus di sebuah halaman.

2. Internal CSS

Internal CSS dituliskan di bagian header file HTML. Fungsinya untuk menentukan tampilan sebuah halaman.

Contohnya, jika Anda ingin agar halaman memiliki latar belakang biru dan teks berukuran 20px yang berwarna putih, kodennya seperti di bawah ini:

```
<head>  
  
<style>  
  
Body { background-color:blue; }  
  
P { font-size:20px; color:white; }  
  
</style>  
  
</head>
```

```
<p>Ini adalah contoh kalimat.</p>
```

Ini adalah contoh kalimat.

Internal CSS sangat membantu ketika Anda ingin membuat halaman website yang tampilannya berbeda dari halaman lain.

3. External CSS

Sesuai namanya, *external* CSS adalah kode CSS yang diletakkan di luar dokumen HTML sebagai *file .css*. Jenis CSS ini berfungsi untuk mengatur tampilan semua halaman website yang Anda tentukan. Jadi, *external* CSS berguna ketika Anda ingin mengatur tampilan beberapa halaman sekaligus.

Agar halaman *website* bisa menggunakan *external* CSS, Anda perlu menambahkan kode di bagian header kode HTML-nya. Contohnya seperti berikut ini:

```
<head>  
<link rel="stylesheet" type="text/css" href=fileCSSAnda.css">  
</head>
```

Dengan kode tersebut, halaman website Anda akan menggunakan fileCSSAnda.css untuk mengatur tampilannya.

REACT.js

React JS adalah *library JavaScript* yang biasa digunakan saat membangun UI suatu *website atau aplikasi web*.

Jadi, *React JS* bisa dianggap seperti perpustakaan yang berisi berbagai kode *JavaScript* yang sudah tertulis (*pre-written*). Anda tinggal mengambil kode yang ingin Anda gunakan. Sehingga, ini membuat proses coding menjadi lebih efisien.

Dengan *JavaScript*, Anda dapat membuat tampilan website yang lebih menarik. Nah, inilah mengapa *library* ini berguna untuk membangun UI, karena pengaplikasian kode *JavaScript* di *website* Anda bisa menjadi lebih mudah.

Library ini juga bersifat open source. Artinya, isinya akan terus berkembang karena semua orang bebas memodifikasi kode di dalamnya.

Berkat kegunaannya, React JS adalah salah satu library yang sangat populer. Saat ini ada [lebih dari satu juta](#) website yang sudah menggunakannya. Bahkan website-website terkenal seperti bbc.com, paypal.com, dan yahoo.com juga menggunakannya.

Kelebihan React JS

Di atas, kita sudah membahas sebagian fungsi dari library ini, yaitu untuk mempermudah penggunaan JavaScript. Selain itu, ada dua fitur tambahan yang menjadi keunggulannya yaitu JSX dan Virtual DOM.

Kedua fitur ini yang seringkali disebut jika React JS sedang dibandingkan dengan library JavaScript yang lain – seperti Inferno JS atau Preact. Oleh karena itu, mari kita lihat kedua fitur tersebut secara lebih detail!

JSX

JSX adalah extension syntax JavaScript yang memungkinkan Anda untuk memodifikasi *Document Object Model (DOM)* dengan kode bergaya HTML.

Untuk mengetahui fungsi JSX dengan lebih jelas, Anda perlu tahu tentang DOM terlebih dahulu.

DOM adalah *application programming interface* (API) yang berfungsi untuk mengatur struktur halaman web. Nah, untuk menambah konten dinamis ke dalam halaman web, developer mesti memodifikasi DOM.

Dengan kata lain, JSX akan mempermudah Anda untuk menambah konten dinamis. Karena extension ini dapat membantu Anda untuk memasukkan syntax bergaya HTML ke dalam DOM.

Akan tetapi, JSX bukanlah HTML. Mungkin bahasa sederhananya seperti ini: JSX terlihat seperti HTML, tapi memiliki fungsi seperti JavaScript.

Selain itu, Anda juga bisa menggunakan JSX di berbagai browser. Karena JSX cocok dengan semua platform browser modern – seperti Chrome atau Firefox.

Virtual DOM

Ketika *developer* mengupdate DOM dengan menggunakan JSX, React JS akan membuat Virtual DOM, yaitu salinan dari DOM asli yang ingin diupdate.

Nah, Virtual DOM berguna untuk melihat bagian dari DOM asli yang berubah. Contohnya, ketika pengguna website mengklik tombol like atau comment, tentu yang perlu berubah hanya bagian like dan comment saja, bukan?

Ketika menemukan bagian yang perlu diubah, React JS akan mengubah bagian itu saja. Jadi, pengguna tidak perlu reload satu halaman untuk melihat perubahannya.

Hal ini dapat berpengaruh terhadap performa website. Karena setiap perubahan hanya dilakukan pada bagian yang diperlukan saja.

Tanpa Virtual DOM, website Anda akan menggunakan HTML untuk update DOM. Sehingga, seluruh DOMnya mesti direload untuk menampilkan perubahan di satu bagian – seperti mengklik tombol like atau menambahkan komentar.