React and React Native Basics

Intro To React

4

PERHATIAN

Jangan lupa untuk isi form absensi. Bagi instruktur mohon mengisi form absensi yang telah diinfokan dan konfirmasi nomor urut peserta yang hadir.

Untuk student harap mengisi form absensi di sini sebelum kelas dimulai. Untuk kode peserta dapat ditanyakan kepada instruktur dan jangan lupa mencantumkan pertemuan ke 1

Pengenalan React

Pengenalan React

Hai React

Jika kamu seorang web developer, engineer, ataupun seseorang yang berkecimpung di dunia web dan aplikasi web, pasti kamu pernah mendengar React. Mungkin kamu pernah. mendengarnya dari sebuah newsletter, Hacker news atau seorang teman merekomendasikan kamu untuk segera belajar React. Dari manapun kamu mendengarnya, opini tentang React biasanya bisa dikelompokkan menjadi dua kategori: suka banget atau benci banget.\ Biasanya sesuatu yang disruptif memunculkan pro dan kontra seperti ini. Biasanya hanya sebagian kecil orang yang awalnya 'klik' dengan ide yang ditawarkan sebelum benar-benar booming. Nah, React adalah salah satu yang masuk kategori teknologi yang disruptif ini. React banyak melakukan distorsi, mengubah mindset dan mentransformasi cara berfikir kita dalam hal membangun user interface.

Pengenalan React

Hai React

Sederhananya, React adalah library JavaScript untuk membangun user interface. React dapat memiliki cara khusus yang dapat membantu kita membangu user interface dengan cara yang simpel, deklaratif dan intuitif. Kita akan lihat implementasinya nanti di bagian-bagian berikutnya.

React bukanlah satu-satunya library untuk mengembangkan user interface. Ada banyak sekali library atau framework yang juga memiliki tujuan yang sama. Sebut saja Angular, Backbone, Ember, Vue dan masih banyak yang lainnya.

Tim dibelakang React mendesain sebuah library yang membuat workflow seorang developer menjadi lebih sederhana sekaligus mampu memberikan performa dan skalabilitas tinggi untuk tim yang menggunakannya.

React berporos kepada komponen. Di React kita membangun komponen-komponen yang membentuk sebuah aplikasi web atau sebuah halaman web. Layaknya mainan lego yang mungkin secara satuan tidak berarti banyak namun ketika digabungkan dapat menjadi sebuah bangunan yang indah.

Pengenalan React

Hai React

Sebuah komponen di React memiliki alur hidup atau lifecycle yang mudah diprediksi dan ditulis dengan "regular old JavaScript" dan bantuan ekstensi JSX.

Yang membuat React berbeda dengan framework lainnya adalah React membantu untuk memahami konsep-konsep dasar JavaScript dan juga konsep functional programming yang diterapkan oleh React. Kita jadi dapat lebih memahami konsep-konsep seperti map, filter, reduce, binding, this scope dan lain sebagainya. Dengan kata lain React membantu kita mengerti fundamental JavaScript karena sebagian besar konsep diatas hanyalah JavaScript yang sama sehingga dapat diterapkan di bahasa JavaScript secara umum.

Pengenalan React

Hai React

Dengan belajar React, selain bisa membuat aplikasi web, kita juga dapat mengembangkan aplikasi mobile dengan React Native, aplikasi VR dengan React VR, aplikasi CLI atau terminal dengan React Blessed, aplikasi smart tv dengan React-TV dan lain sebagainya. Sehingga kita cukup belajar fundamental React kemudian kita dapat menggunakan skill React kita untuk mengembangkan aplikasi di platform lain.

Ditambah lagi konsep-konsep yang digunakan React merupakan nilai jual utama dari React, bukan librarynya. Terbukti banyak sekali library atau framework yang menggunakan salah satu konsep React atau bahkan semuanya. Seperti Preact, Inferno, dan lain sebagainya yang mengaku sebagai React-like UI library.

- ***"React is mainly a concept and a library just secondly" Peter Marton, CTO RisingStack***

Pengenalan React

Pengguna React

Biasanya, bagi sebuah library, tolak ukur paling tepat apakah sebuah library itu populer adalah siapa yang menggunakannya. React sendiri merupakan library yang sangat populer digunakan oleh banyak perusahaan teknologi ternama seperti Netflix, Uber, Heroku, PayPal, BBC, Microsoft, Asana, ESPN, Walmart, Codecademy, Atlassian, Airbnb, Khan Academy, Facebook tentunya, dan masih banyak lagi. Perusahaan diatas menggunakan React untuk case-case yang berbeda sesuai kebutuhan bisnis mereka untuk membuat user interface dari aplikasi mereka.

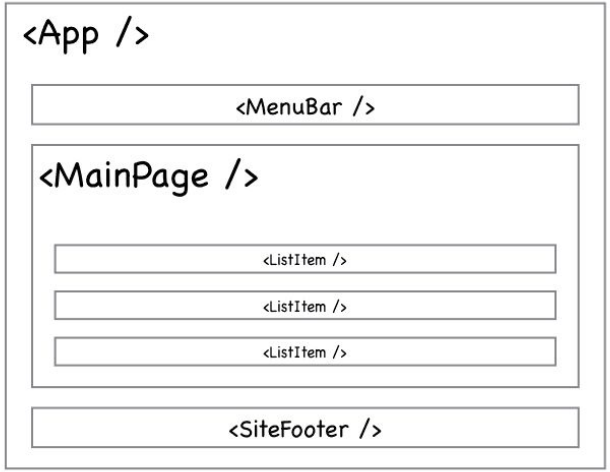
Pengenalan React

Konsep Utama React

Mindset utama dari React adalah **Component**. Sehingga, sebuah website yang menggunakan React adalah kumpulan komponen yang kita definisikan, lengkap dengan karakter, style, dan fitur-fitur masing-masing

Komponen utama dalam React biasanya adalah App.js. Dan dari contoh pada gambar di samping, halaman web ini memiliki komponen :
- MenuBar
- MainPage
- SiteFooter

Komponen ListItem adalah bagian dari komponen MainPage

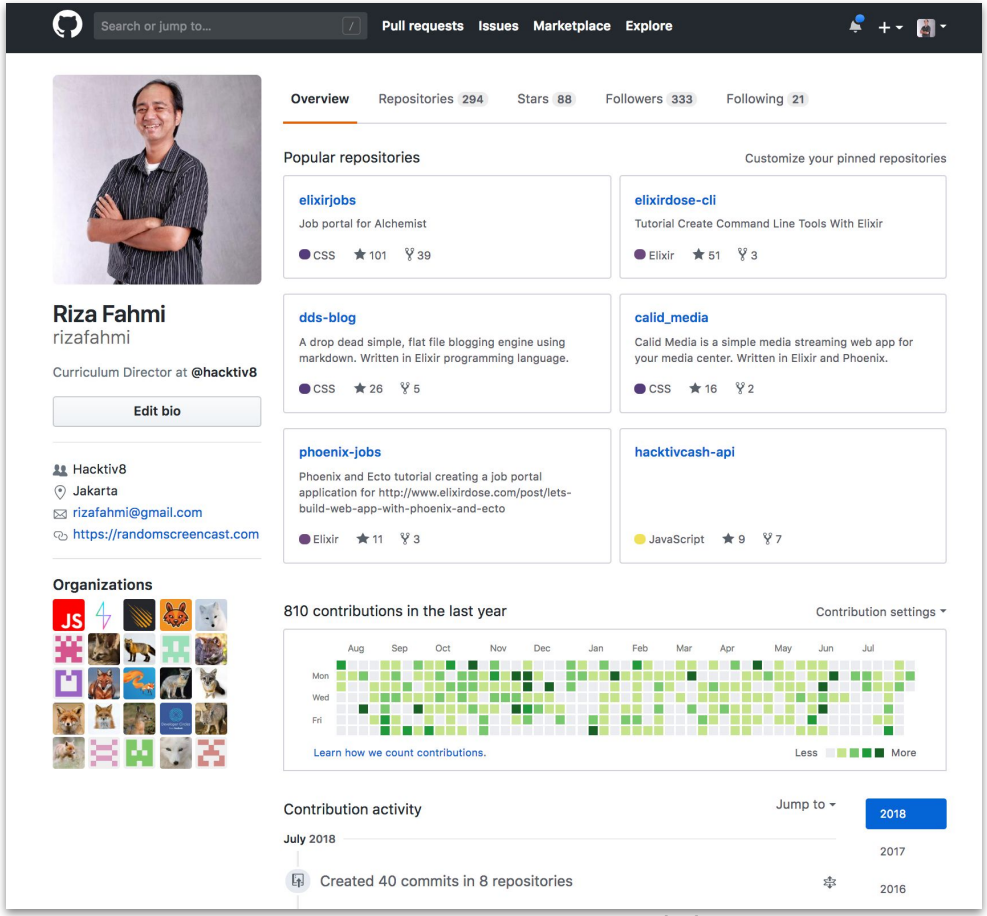


Pengenalan React

Konsep Utama React

Mari kita sedikti berlatih…

Pecah halaman github profile di samping menjadi komponen-komponen. Bayangkan kamu menjadi orang yang bertanggung jawab mengerjakan halaman profil tersebut. Coret-coretlah dan buatlah sketsa dengan menggunakan kertas dan pulpen. Ada berapa komponen yang akan kamu buat?



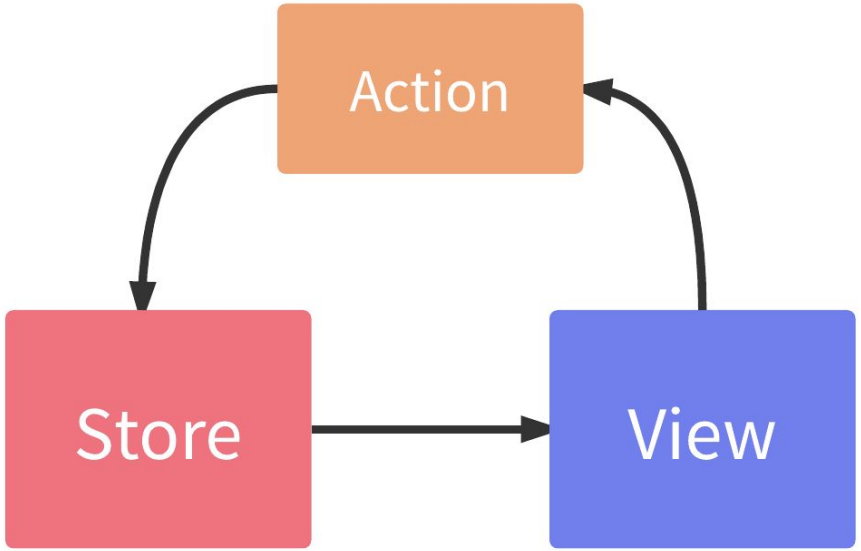
Aliran Data Satu Arah

Aliran Data Satu Arah

Penjelasan

Sedikit berbeda dengan beberapa framework seperti Angular yang menerapkan aliran data dua arah, di React data hanya mengalir satu arah. Hal ini menjadi signifikan karena dengan menerapkan hal ini, perubahan data menjadi lebih mudah dimengerti dan lebih mudah diprediksi.

Aliran data di React dapat diilustrasikan seperti berikut.

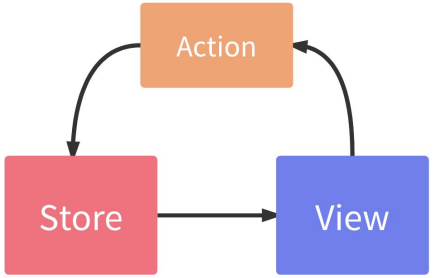


Intro to React - 4

Aliran Data Satu Arah

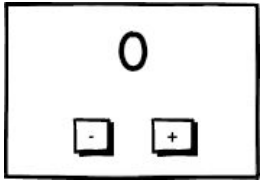
Penjelasan

View adalah kumpulan komponen yang sudah kita bahas di bagian sebelumnya. Di view juga ada berbagai action seperti onClick ketika button di klik, onChange ketika pengguna memasukkan data di form dan banyak lagi. Dan di React, semua data disimpan kedalam sebuah wadah yang disebut dengan store atau state. Uniknya, ketika terjadi perubahan data, katakanlah seorang pengguna meng-klik button atau action yang lainnya, view tidak bisa mengubah data atau apapun juga. View harus menghubungi action memberitahu bahwa ada sebuah action yang di trigger oleh pengguna. Dan tugas action-lah yang kemudian menghubungi store untuk kemudian mengubah data sesuai actionnya. Dan setelah itu baru kemudian store mengubah view hingga sesuai dengan data yang baru saja berubah. Jadi satu jalur, dari view ke action ke store dan kembali ke view. Dan tidak bisa sebaliknya. Untuk mengubah data, view harus selalu melalui action dan seterusnya.



Aliran Data Satu Arah

Contoh



Mari kita lihat contoh nyata. Misalkan kita punya sebuah aplikasi counter dengan button + untuk menambahkan angka dan - untuk mengurangi angka. Angka counter awal adalah 0.

Ketika pengguna memencet tombol +, view memberitahu action bahwa tombol + dipencet. View disini tidak memberitahu counter sekarang berapa. Dia hanya memberikan informasi bahwa tombol. + dipencet.

Action kemudian melanjutkan informasi tersebut ke store juga tidak membawa informasi tentang current counter number yang saat ini masih 0 yang tampil di view. Karena yang mengetahui informasi current counter number adalah store sendiri, bukan action dan juga bukan view.

Setelah informasi sampai ke store, tugas store adalah melakukan kalkulasi yang tadinya angkanya 0 ditambah 1 menjadi 1. Dan seteleh data diubah, store menginformasikan kepada view bahwa data berubah, silakan ubah view nya sehingga datanya dari 0 menjadi 1. Dan seterusnya.

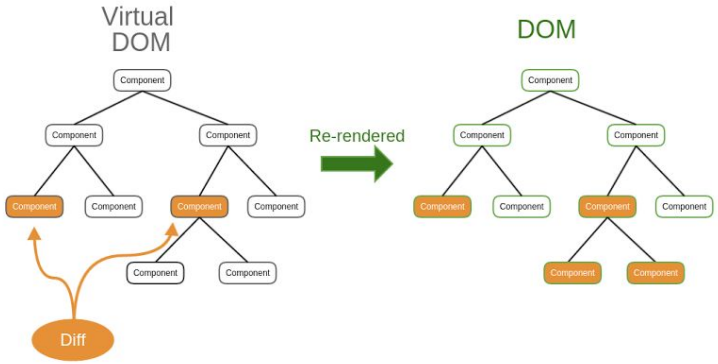
Virtual DOM

Intro to React - 4

Virtual DOM

Penjelasan

Masih berhubungan dengan alur data satu arah sebelumnya setiap kali terjadi perubahan data yang terjadi di store, React atau dalam hal ini view akan selalu melakukan proses render ulang. Ya, betul setiap kali ada perubahan data, React akan melakukan proses render ulang seluruh komponen dan hal ini yang membuat React keren! Dan React memang didesain seperti itu untuk memudahkan workflow kita sebagai developer. Hal ini juga menyebabkan React tidak membutuhkan data binding yang aneh-aneh (baca: magical).



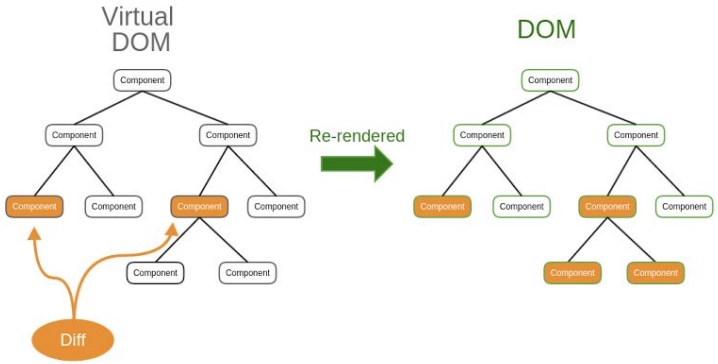
HACKTIV8

Intro to React - 4

Virtual DOM

Penjelasan

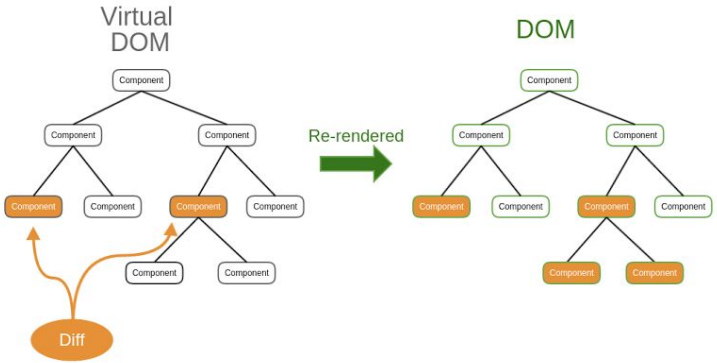
DOM ini strukturnya berbentuk *tree* atau pohon. Dan ketika sebuah elemen diakses, dimodifikasi ataupun membuat elemen baru, browser harus mencari ke seluruh cabang-cabang dari pohon DOM tersebut. Dan hal itu membutuhkan waktu yang lama dan komputasi yang cukup berat. Dan karena aplikasi web modern membutuhkan data yang terus menerus berubah (misalnya aplikasi social media seperti facebook atau twitter) dan setiap perubahan harus melakukan komputasi yang berat untuk mengubah DOM, makanya React memutuskan menggunakan Virtual DOM daripada harus mengubah DOM secara langsung.



Intro to React - 4

Virtual DOM

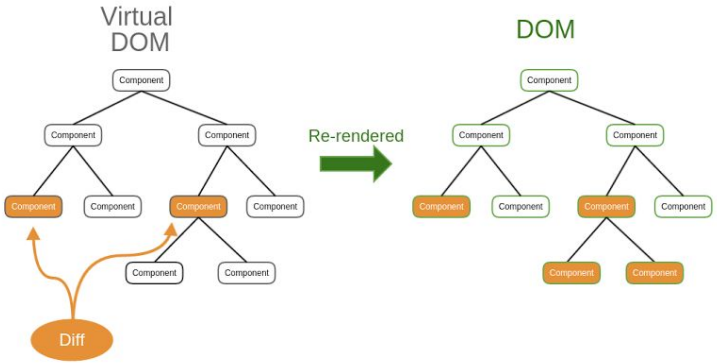
Penjelasan



Virtual DOM ini sebenarnya hanyalah struktur data yang menyimpan informasi lengkap tentang DOM. Seperti blueprint kalau kita mau membangun rumah atau ruangan. Jadi setiap kali terjadi perubahan data dan harus mengubah DOM, React melakukan perubahan tersebut di VirtualDOM terlebih dahulu. Baru kemudian React membandingkan VirtualDOM yang sudah berubah dengan DOM yang sebenarnya. Jika ada yang berbeda, VirtualDOM akan mengubah DOM yang berubah saja, jadi tidak semuanya diganti sehingga tidak seberat jika harus mengubah seluruh pohon DOM.\ Mengubah VirtualDOM jauh, jauh lebih ringan karena VirtualDOM sederhananya hanyalah struktur data atau kerangkanya saja jadi perubahan akan lebih cepat dan ringan.

Virtual DOM

Penjelasan



Sejak React mengadaptasi Virtual DOM dan berhasil mempermudah workflow developer sekaligus membuat aplikasi menjadi cepat secara performa, library/framework lain juga ikut mengadaptasi Virtual DOM. Sebut saja Vue, Angular hingga Ember saat ini juga menerapkan Virtual DOM.

JSX

JSX

Penjelasan

JSX dari awal merupakan bagian dari React. Banyak orang yang tadinya tertarik React ketika melihat JSX banyak yang enggan dan akhirnya mengurungkan niat untuk mencoba React. Makanya JSX ini wajib dibahas di bagian khusus.

JSX pada dasarnya adalah JavaScript dengan tambahan fungsi. Dengan JSX kita dapat menulis kode yang sangaaat mirip dengan HTML atau lebih tepatnya XML, dengan kemampuan untuk melakukan evaluasi kode JavaScript didalam JSX itu sendiri sehingga menjadi sebuah 'templating language' yang sangat powerful.

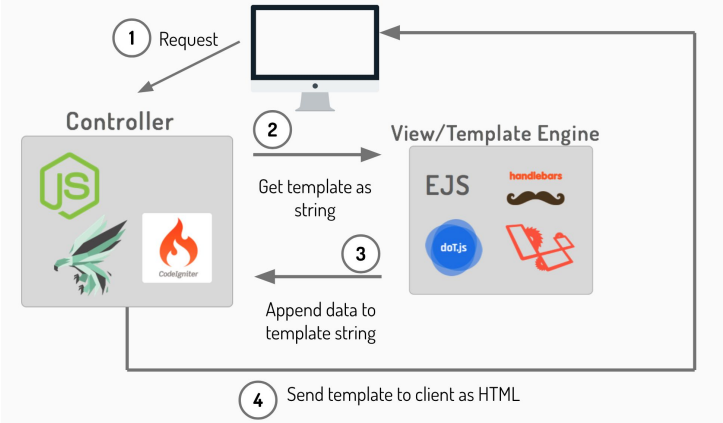
Dapat dimengerti kenapa banyak orang enggan menggunakan JSX. Banyak yang awalnya seperti itu dan best practice dalam dunia pemrograman web juga menyarankan memisahkan template dengan logic dan menggabungkan keduanya sudah seperti kembali ke zaman dimana sintaks html dan kode logic lainnya digabung menjadi satu.

Padahal sebenarnya solusi JSX ini cukup efisien dan masuk akal. Mari kita bahas dan bandingkan dengan solusi yang menggunakan templating language dengan JSX.

JSX

Penjelasan

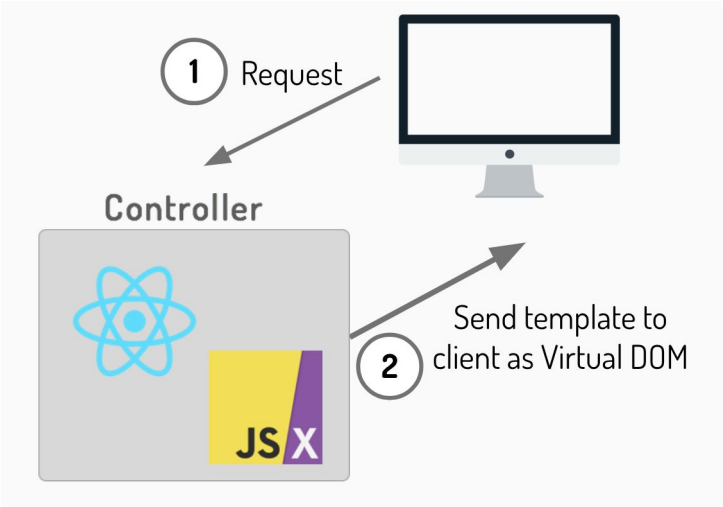
Ilustrasi di bawah menggambarkan bagaimana framework yang memanfaatkan templating language bekerja.



Ketika ada permintaan untuk memunculkan sebuah halaman melalui browser dan pengguna mengunjungi halaman tertentu atau dengan istilah yang lebih teknis, pengguna mengakses route tertentu dan route tersebut memanggil controller, yang kemudian akan merender halaman berdasarkan template yang berkaitan dengan controller dan route tersebut. Proses render tersebut sederhananya adalah melakukan concat terhadap string dengan menambahkan data-data yang disertakan di route atau controller tersebut. Proses append string template dengan data tersebut bukanlah operasi yang ringan apalagi ketika template sudah cukup kompleks dan besar ukurannya. Setelah proses append string dan rendering selesai baru kemudian string yang sudah ditambahkan data tersebut dikirimkan kembali ke pengguna.

JSX

Penjelasan



Sementara, ketika kita menggunakan JSX, proses menjadi lebih simpel. Framework tidak lagi perlu melakukan proses rendering dalam bentuk append string yang butuh tenaga komputasi cukup besar. Framework tinggal menterjemahkan hasil dari JSX yang ditranspilasi menjadi fungsi-fungsi (bukan string) menjadi tag-tag html dan langsung mengirimkan hasilnya kembali ke pengguna.

Dan beberapa framework seperti Vue dan Hyperapp pun akhirnya memutuskan menggunakan JSX sebagai pengganti templating language. Meskipun di Vue, penggunaan JSX sifatnya optional.

Dan, apabila teman-teman tetap bersikeras tidak ingin menggunakan JSX pun tidak ada masalah. Teman-teman tetap bisa menggunakan React tanpa JSX.

First React App

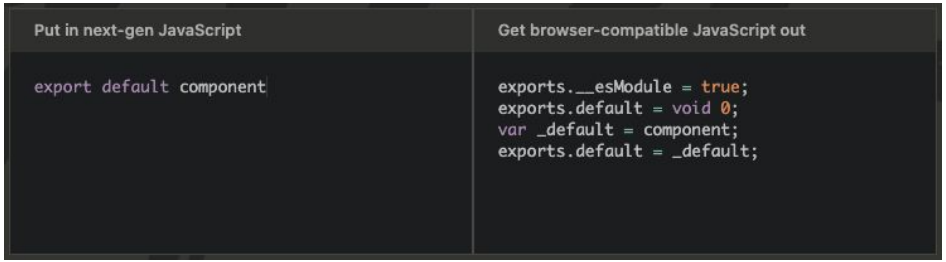
First React App

Instalasi menggunakan Babel dan Webpack

Javascript adalah salah satu bahasa pemrograman dengan perkembangan yang paling pesat. Berbagai macam fitur dan sintaks diperkenalkan di ES6. Namun perkembangan tadi ternyata membutuhkan waktu yang cukup lama untuk dapat diadaptasi baik oleh browser maupun NodeJS

Nah, Babel adalah penyelamat kita gaes. Hal ini disebabkan karena babel itu sendiri adalah sebuah transpiler, yang tugasnya men-terjemahkan sintaks-sintaks yang unsupported ke sintaks yang di-support oleh browser atau NodeJS

Silakan temen-temen pelajari dan pahami Babel langsung dari https://babeljs.io/

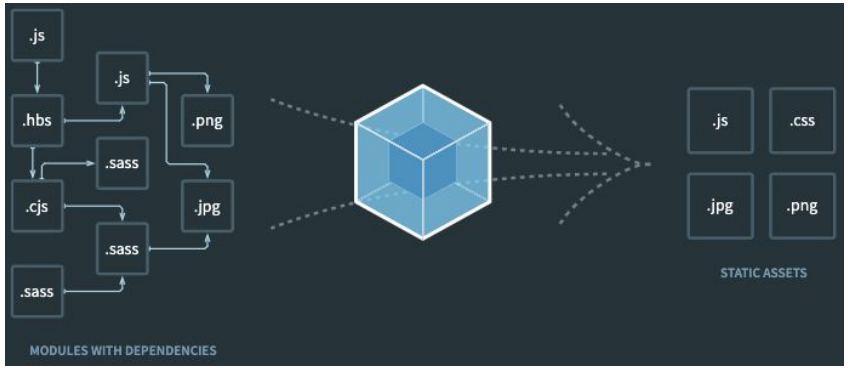


First React App

Instalasi menggunakan Babel dan Webpack

Semua berawal dari *module*. Sederhananya, *module* adalah sebuah berkas yang berisi script kode. Module memiliki sifat khusus, yakni dapat memuat atau dimuat oleh *module* lainnya. Berkat sifat inilah antar *module* dapat saling ekspor dan impor untuk bertukar fungsi.

Nah, *module* ini jumlahnya bisa banyak, kompleks, dan bahkan bisa jadi ada beberapa *module* dengan nama yang sama. Untuk itu, kita memerlukan sebuah *tool* yang dapat bertindak sebagai *module management*. Dalam hal ini, istilah yang dipakai oleh *module management* ini adalah *bundler*



First React App

Instalasi menggunakan Babel dan Webpack

Core concept dari **Webpack** adalah :

- Entry
- Output
- Loaders
- Plugins
- Mode
- Browser Compatibility

Untuk memahami lebih dalam soal **Webpack** langsung dari websitenya, silakan kunjungi : https://webpack.js.org/

First React App

Instalasi menggunakan Babel dan Webpack

Nah terus, apa sih gunanya kita mengenal Babel dan Webpack ini ? Jadi, di materi selanjutnya, kita akan berkenalan dengan React JS. Namun, sebelum kita masuk ke materi tersebut, kita akan melakukan instalasi React JS secara manual. Nah, langkah-langkah manual ini, mengikutsertakan Babel dan Webpack dalam proses instalasinya.

CATATAN : Semua perintah-perintah yang dicontohkan adalah untuk dilakukan pada OS Linux. Untuk WINDOWS, beberapa perintah harus dicari ekivalensinya, ATAU dilakukan dengan UI pada Windows Explorer

Berikut adalah langkah-langkahnya :

1. Buat folder aplikasi dan masuk ke folder tersebut

```
> mkdir reactApp
> cd reactApp
```

2. Generate file “package.json”

```
> npm init -y
```

3. Install React dan React Dom

```
> npm install react --save
> npm install react-dom --save
```

… lanjut ke slide selanjutnya ->

Intro to React - 4

First React App

Instalasi menggunakan Babel dan Webpack

… lanjutan :

4. Install webpack dan semua pendukungnya

```
> npm install webpack --save
> npm install webpack-dev-server --save
> npm install webpack-cli --save
```

5. Install babel dan semua pendukungnya

```
> npm install babel-core --save-dev
> npm install babel-loader --save-dev
> npm install babel-preset-env --save-dev
> npm install babel-preset-react --save-dev
> npm install html-webpack-plugin --save-dev
```

6. Buat folder src

```
> mkdir src
```

7. Buat file-file pendukungnya

```
> touch src/index.html
> touch src/index.js
> touch webpack.config.js
> touch .babelrc
```

Untuk WINDOWS, touch bisa digantikan dengan :

```
> type nul > index.html
```

… lanjut lagi ya gaes ->

First React App

Instalasi menggunakan Babel dan Webpack

```
const path = require("path");
const HtmlWebpackPlugin = require("html-webpack-plugin");

module.exports = {
  entry: path.join(__dirname, "src", "index.js"),
  output: { path: path.join(__dirname, "build"), filename: "index.bundle.js" },
  mode: process.env.NODE_ENV || "development",
  resolve: { modules: [path.resolve(__dirname, "src"), "node_modules"] },
  devServer: { static: path.join(__dirname, "src") },
  module: {
    rules: [
      {
        test: /\.(js|jsx)$/,
        exclude: /node_modules/,
        use: ["babel-loader"]
      },
    ],
  },
  plugins: [
    new HtmlWebpackPlugin({
      template: path.join(__dirname, "src", "index.html"),
    }),
  ],
};
```

… lanjutan :

8. **Buka file webpack.config.js dan isikan dengan kode di samping**

… ayo guys, lanjut lagi ->

Intro to React - 4

First React App

Instalasi menggunakan Babel dan Webpack

```
  "scripts": {
    "webpack": "webpack",
    "start": "webpack serve",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
```

… lanjutan :

9. Buka file package.json dan ubah bagian “script” menjadi seperti di atas

10. Buka file src/index.html dan isikan dengan kode seperti di samping

… ayo guys, lanjut lagi ->

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>React with Webpack and Babel</title>
  </head>
  <body>
    <noscript>
      You need to enable JavaScript to run this app.
    </noscript>
    <div id="root">
      <!-- This div is where our app will run -->
    </div>
  </body>
</html>
```

First React App

Instalasi menggunakan Babel dan Webpack

… lanjutan :

11. Buka file src/index.js dan isikan dengan kode seperti di bawah

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

const HelloWorld = () => {
  return (
    <h1>Hello World</h1>
  );
}

ReactDOM.render(<HelloWorld />,
  document.getElementById("root"));
```

12. Buka file .babelrc dan isikan dengan kode seperti di bawah

```
{
  "presets": [
    "@babel/env",
    "@babel/react"
  ]
}
```

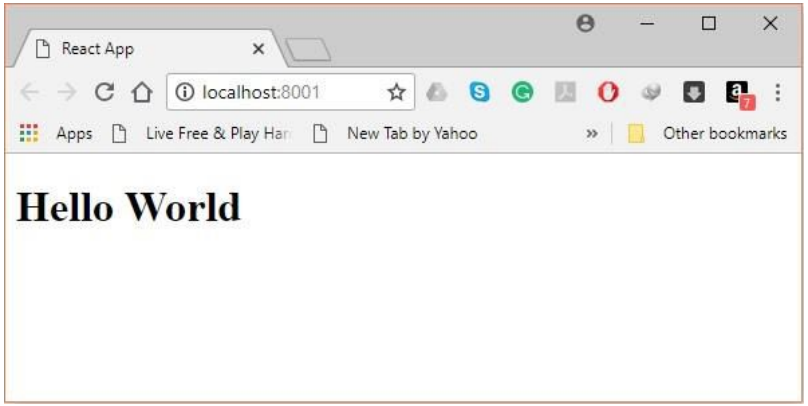
First React App

Instalasi menggunakan Babel dan Webpack

13. Jalankan server dengan perintah di bawah ini dan browser akan otomatis terbuka seperti di samping

```
> npm start
```

… ayo guys, lanjut lagi ->



First React App

Instalasi menggunakan Babel dan Webpack

Sebagai langkah terakhir, kita akan generate bundle nya. Bundle ini adalah hasil kompilasi akhir dengan bantuan Babel dan Webpack tadi, sehingga menjadi file HTML dan JS yang siap saji

```
> npm run build
```

Hasilnya, adalah sebuah folder yg bernama **build** dan berisi 2 buah file :

- index.html
- index.bundle.js

… dan silakan coba untuk membuka file index.html nya di browser kamu

SELAMAT !!! HELLO WORLD VERSI REACT JS TELAH BERHASIL KAMU BUAT

Untuk penjelasan lebih detail, silakan kunjungi : https://dev.to/deadwing7x/setup-a-react-app-with-webpack-and-babel-4o3k

First React App

Instalasi menggunakan create-react-app package

Sebelum kita lanjut ke materi selanjutnya, kita akan melakukan instalasi React terlebih dahulu, dengan cara yang lebih sederhana guys, yaitu memanfaatkan perintah built in dari package React itu sendiri. Dan sama seperti catatan sebelumnya, Semua perintah-perintah yang dicontohkan adalah untuk dilakukan pada OS Linux. Untuk WINDOWS, beberapa perintah harus dicari ekivalensinya, ATAU dilakukan dengan UI pada Windows Explorer

Berikut adalah langkah-langkahnya :

1. **Pastikan temen-temen sudah berada pada folder yang akan menampung folder aplikasi yang akan kita generate**

2. **Generate sebuah aplikasi dengan nama my-app**

```
> npx create-react-app my-app
```

3. **Masuklah ke dalam root folder aplikasi kita**

```
> cd my-app
```

… lanjut ke slide selanjutnya ->>

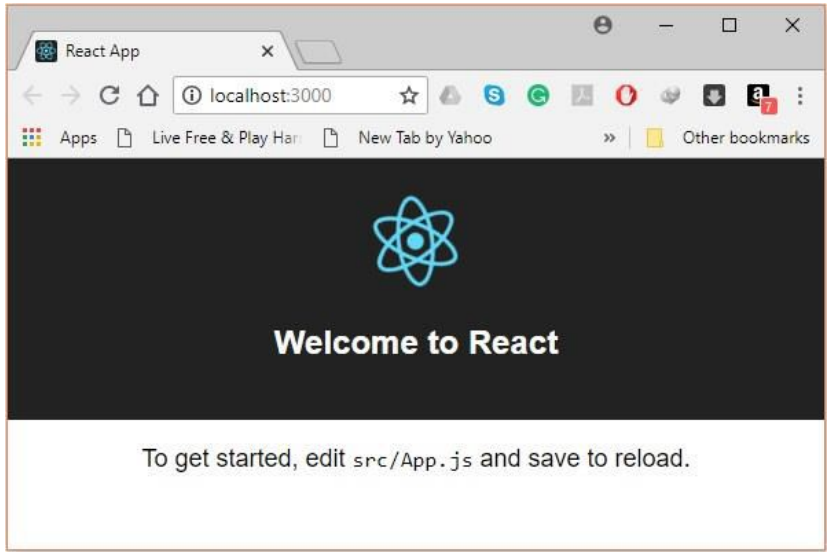
First React App

Instalasi menggunakan create-react-app package

… lanjut :

4. Jalankan aplikasi, dan dapatkan tampilan pada browser seperti di samping

```
> npm start
```



Thank You

PT Hacktivate Teknologi Indonesia
Gedung Aquarius Pondok Indah
Jalan Sultan Iskandar Muda No.7
Kebayoran Lama, Jakarta Selatan

www.hacktiv8.com