Nama : Vika Alpiana  
Program : React and React Native for Frontend Developer  
Kode Kelas : RCTN-KS05-004

* **React**

Salah satu yang masuk kategori teknologi yang disprutif. React melakukan banyak disorti, Sederhananya react dalah library JavaScript untuk membangun user interface dengan cara yang simple, deklaratif dan intuitif. React berporos pada komponen. 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 lain adalah membatuu kita memahami konsep dasar dari JavaScript dan konsep functional programming yang diterapkan oleh react, seperti map, filter, reduce, binding, this scope dll. 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

* **Mindset utama dari React**
* **Component**

Sehingga sebuah website yang menggunakan React adalah kumpulan komponen yang didefinisikan, lengkap dengan karakter, style, dan fitur – fitur masing.

* **Aliran Data Satu Arah**

React menerapkan aliran data satu arah, hal ini lebih signifikan karena data menjadi lebih mudah dimengerti dan lebih mudah diprediksi.

* **View**

Adalah kumpulan komponen yang sudah kita bahas di bagian sebelumnya. Di view 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 sesuaiactionnya. 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 keview. Dan tidak bisa sebaliknya. Untuk mengubah data, view harus selalu melalui actiondan seterusnya.

Contoh : Misalkan kita punya sebuah aplikasi counter dengan button + untukmenambahkan angka dan - untuk mengurangi angka. Angka counter awal adalah 0.

* Virtual DOM

Setiap kali ada perubahan data, React akan melakukan render ulng seluruh komponen dan React mempermudah workflow kita. Hal ini menyebabkan React tidak membutuhkan data binding yang aneh. React hanya merender ulang virtual DOM. Virtual DOM ini hanyalah sebuah struktur data yang menyimpan informasi lengkap tentang DOM. 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.

* JSX

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

Dengan 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.

* **Installation menggunakan Babel dan Webpack**

Babel adalah sebuah transpiler, yang tugasnya menerjemahkan sintaks-sintaks yang unsupported ke sintaks yang disupport oleh browser atau NodeJS.

Semua berawal dari module. Sederhananya, module adalahsebuah berkas yang berisi script kode. Module memiliki sifatkhusus, yakni dapat memuat atau dimuat oleh modulelainnya. Berkat sifat inilah antar module dapat saling ekspordan impor untuk bertukar fungsi.Nah, module ini jumlahnya bisa banyak, kompleks, danbahkan bisa jadi ada beberapa module dengan nama yangsama. Untuk itu, kita memerlukan sebuah tool yang dapatbertindak sebagai module management. Dalam hal ini, istilahyang dipakai oleh module management ini adalah bundler.

Core concept dari Webpack adalah :

* Entry
* Output
* Loaders
* Plugins
* Mode
* Browser Compatibility