**BAB II**

**REACT JS STATEMENTS AND HANDLER EVENT**

1. **Operation**
2. **Selection**

Pernyataan if/else akan mengeksekusi blok kode dengan kondisi yang ditentukan benar. Jika kondisinya salah, blok kode lain yang dapat dieksekusi. Pernyataan if/else adalah bagian dari Pernyataan "Bersyarat" JavaScript, yang digunakan untuk melakukan tindakan yang berbeda berdasarkan kondisi yang berbeda. Dalam JavaScript kami memiliki pernyataan bersyarat berikut:

* Statement *if* untuk menentukan blok kode yang akan dieksekusi, jika kondisi yang ditentukan benar
* Statement *else* untuk menentukan blok kode yang akan dieksekusi, jika kondisi yang sama salah
* Statement *else if* untuk menentukan kondisi baru yang akan diuji, jika kondisi pertama salah
* Statement *switch* untuk memilih salah satu dari banyak blok kode yang akan dieksekusi

Berikut adalah contoh penggunaan statement selection yaitu IF dan SWITCH pada React Function Component:

|  |
| --- |
| Penggunaan short *IF* pada JSX |

Berikut adalah contoh keluaran dari script diatas:

Graphical user interface, text

Description automatically generated

***Gambar 1.1.*** *Output penerapan selection if dan switch*

1. **Repetition**

Loop berguna, jika ingin menjalankan kode yang sama berulang kali, setiap kali dengan nilai yang berbeda. Biasanya loop digunakan ketika bekerja dengan bentuk value array. Namun bentuk pengulangan pada javascript sama halnya dengan bahasa pemograman umum lainnya yang menggunakan *for(..), while(..)* atau *do…while(…)*. Penerapan terhadap bentuk-bentuk statement pengulangan sama saja, sebagai contoh:

|  |
| --- |
| var text = "";  for (let i = 0; i < 5; i++) {   text += "The number is " + i + "<br>"; } |
| while (i < 10) {   text += "The number is " + i;   i++; } |

Namun pada javascript jika memiliki bentuk value dengan menggunakan sebuah array maka yang perlu digunakan ialah dengan *map()*.

* *map()* membuat array baru dari hasil memanggil fungsi untuk setiap elemen array.
* *map()* memanggil fungsi satu kali untuk setiap elemen dalam array.
* *map()* tidak menjalankan fungsi untuk elemen kosong.
* *map()* tidak mengubah array asli.

Contoh penerapan map() pada sebuah data array:

|  |
| --- |
|  |

Berikut adalah output dari kodingan diatas:

Table

Description automatically generated

***Gambar 1.2.*** *Penerapan repetition map untuk array*

1. **Handler Event**

*Handling event* dengan elemen React sangat mirip dengan penanganan event pada elemen DOM. Namun pada React memiliki beberapa perbedaan penerapan handler event pada DOM JS dan HTML. Handler event pada React diberi nama menggunakan format *camelCase*, bukan huruf kecil semua atau huruf besar. Karena nama aksi event akan diteruskan sebagai bentuk JSX bukan sebagai string.

Misalnya, pada HTML:

|  |
| --- |
| <button onclick="hitmeBabe()">  Click me please  </button> |

Namun pada React:

|  |
| --- |
| <button onClick="hitmeBabe">  Click me please  </button> |

Berikut ini adalah contoh penerapan event handler pada sebuah componen function untuk membuat sebuah toogle:

Text

Description automatically generated Graphical user interface, application

Description automatically generated

Berikut ini adalah contoh handler untuk membuat sebuah aplikasi sederhana sticky notes, dimana script dibawah ini akan menambahkan atau menghapus item berdasarkan aksi klik pada sebuah icon :

|  |
| --- |
|  |

Chart, treemap chart

Description automatically generated

***Gambar 2.*** *Contoh handler event*

1. **Layout UI**

Komunitas React menyediakan banyak koleksi framework komponen UI tingkat lanjut. Salah satunya ialah Bootstrap, React Bootstrap adalah salah satu framework React UI yang populer. Berikut adalah contoh instalasi framework CSS pada react.

* Installasi

Menginstall React Bootstrap dapat menggunakan node npm

|  |
| --- |
| npm install react-bootstrap bootstrap |

Setelah berhasil menginstall library react bootstrap, untuk memanggil react bootstrap panggilah library tersebut di dalam file index.js

Graphical user interface, text, email

Description automatically generated

* Cara Penggunaan

Pada tugas sebelumnya, mengenai pembuatan form biodata, kita dapat memanfaatkan react bootstrap untuk mengatur layout ui menjadi lebih indah dan atraktif.

Graphical user interface

Description automatically generated with low confidenceGraphical user interface

Description automatically generated

***Gambar 3.*** *Contoh penggunaan layout UI Bootstrap*

|  |
| --- |
|  |

Source: <https://github.com/FebryFairuz/IBIK-20222023-GENAP-PWL/tree/master/PROJECT/my-app/src>/components/pertemuan-2/compositions/

1. **Composition**

React memiliki model komposisi yang kuat, komposisi pada react merekomendasikan penggunaan node-node sebagai komposisi alih-alih sebagai pewarisan untuk menggunakan kembali kode di antara komponen. Pada bagian ini, akan mempertimbangkan beberapa masalah di mana pengembang yang baru mengenal React sering menggunakan pewarisan, dan menunjukkan bagaimana kita dapat menyelesaikannya dengan komposisi. Beberapa komponen tidak mengenal node sebelumnya. Seperti untuk komponen untuk membuat sebuah Sidebar atau Dialog yang mewakili “kotak” parentnya. Untuk hal menangani hal tersebut react merekomendasikan agar komponen menggunakan *prop* khusus *children* untuk meneruskan elemen *children* secara langsung sebagai sebuah output JSX, contoh:

Graphical user interface, text

Description automatically generated

Selanjutnya membuat node turunan dari RFC Layout, yaitu component Widget1:

Graphical user interface, text, application

Description automatically generated

Apa pun yang ada di dalam tag JSX *<Layout>* diteruskan ke komponen Layout sebagai prop *children*. Karena Layout merender *{props.children}* di dalam *<div>*, elemen yang diteruskan akan muncul menjadi bagian JSX.

Graphical user interface, application

Description automatically generated

***Gambar 4.*** *Contoh penggunaan Composition pada React Component Function*

1. **Latihan Praktikum**
2. Pada project sebelumnya buatlah sebuah package baru dengan nama *latihan-5\_1*. Ubahlah script pada contoh kasus Gambar 4, dengan menggunakan RCC untuk *Widget1*.
3. Pada tugas praktikum sebelumnya tenang form biodata anda, ubahlah tugas tersebut sehingga mampu menampung data dalam bentuk multiple. Berikut adalah contoh ilustrasinya:

Graphical user interface, text, application, email

Description automatically generated

Pada gambar diatas jika mengklik tombol Add New Row maka akan membentuk baris baru yang berisikan form sesuai masing-masing field. Tombol remove yang berwarna merah diperuntukan untuk menghapus list item object yang dipilih. Tombol ini hanya tampil jika jumlah data lebih dari satu.

Graphical user interface, application

Description automatically generated

Jika isian sudah terpenuhi, ketika mensubmit form maka akan menampilkan sebuah pop up window yang berisikan NPM, Fullname, Birthdate dan Age.

Pengumpulan tugas Latihan praktikum dikumpulkan kedalam GITHUB masing-masing mahasiswa berdasarkan repository yang telah dibuat PWL-TI-20-PA-NPM. File source code disimpan sesuai nama project-praktikum dan masukan kedalam repositori tersebut. Buatkanlah file dokumen dalam bentuk file pdf yang berisi Screen Capture dari hasil program yang telah dikerjakan. Simpan dalam file PDF tersebut kedalam project tersebut.

Tambahkan Collaborator management access pada repository anda ke *@FebryFairuz*