

MINGGU 3

Kombinasi Ionic & React.js (part 2)



DESKRIPSI TEMA

- Memecah App menjadi Components
- Menampilkan Alert
- Menambahkan Segment Button
- Menghubungkan Components

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Mahasiswa mampu menerapkan Ionic dengan React.js untuk membangun logika aplikasi.

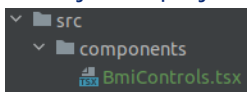
PERALATAN YANG DIGUNAKAN

Web Storm or Visual Studio Code

LANGKAH-LANGKAH PRAKTIKUM

Splitting App into Components

1. Melanjutkan project sebelumnya, buat dua file component baru dengan nama **BmiControls.tsx**



2. Tambahkan import React pada file BmiControls.tsx

```
import React from 'react';
```

3. Tambahkan kode berikut pada file BmiControls.tsx. Kode tag **<IonRow>**, **<IonCol>**, dst dipindahkan dari App.tsx (*cut-paste*).

```
const BmiControls: React.FC = () => {  
  return (  
    <IonRow>  
      <IonCol className="ion-text-left">  
        <IonButton onClick={calculateBMI}>  
          <IonIcon slot="start" icon={calculatorOutline}></IonIcon>  
          Calculate  
        </IonButton>  
      </IonCol>  
      <IonCol className="ion-text-right">  
        <IonButton onClick={resetInputs}>  
          <IonIcon slot="start" icon={refreshOutline}></IonIcon>  
          Reset  
        </IonButton>  
      </IonCol>  
    </IonRow>  
  );  
};  
  
export default BmiControls;
```

4. Pastikan terdapat import dari @ionic dan ionicons.

```
import {IonButton, IonCol, IonIcon, IonRow} from "@ionic/react";
import {calculatorOutline, refreshOutline} from "ionicons/icons";
```

5. Tambahkan import BmiControls pada App.tsx

```
import BmiControls from '../components/BmiControls';
```

6. Tambahkan tag `<BmiControls />` ditempat sebelumnya Anda memindahkan tombol (step no. 3).

```
    </IonCol>
  </IonRow>
  <BmiControls />
  {calculatedBMI && <IonRow>
    <IonCol>
```

7. Tambahkan attribute `onCalculate` dan `onReset` pada `<BmiControls />` agar function calculate BMI dan resetInputs dapat dijalankan.

```
<BmiControls onCalculate={calculateBMI} onReset={resetInputs} />
```

8. Tambahkan `props` untuk menghubungkan `onCalculate` dan `onReset` yang ada di App.tsx dengan BmiControls.tsx.

```
const BmiControls: React.FC<{onCalculate: () => void; onReset: () => void}> = props => {
  return (
    <IonRow>
      <IonCol className="ion-text-left">
        <IonButton onClick={props.onCalculate}>
          <IonIcon slot="start" icon={calculatorOutline}></IonIcon>
          Calculate
        </IonButton>
      </IonCol>
      <IonCol className="ion-text-right">
        <IonButton onClick={props.onReset}>
          <IonIcon slot="start" icon={refreshOutline}></IonIcon>
          Reset
        </IonButton>
      </IonCol>
    </IonRow>
  );
};
```

9. Coba aplikasi di browser, apakah tetap berjalan seperti sebelumnya.
10. Lakukan Tugas no. 1 sebelum melanjutkan tahapan berikutnya.

Menampilkan Alert

1. Pada file App.tsx, tambahkan sebuah const useState.

```
const [ error, setError ] = useState<string>();
```

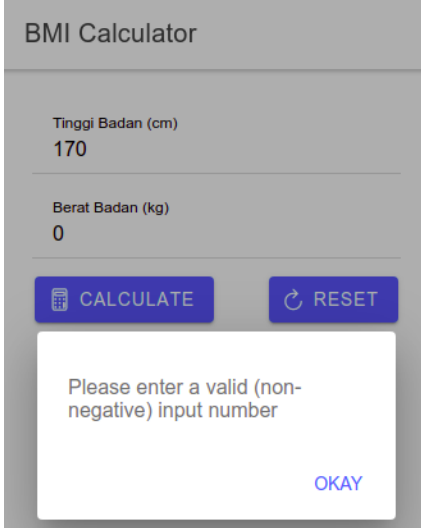
2. Modifikasi filter pada function calculateBMI menjadi seperti berikut.

```
if(!enteredWeight || !enteredHeight || +enteredHeight <= 0 || +enteredWeight <= 0) {
  setError( value: 'Please enter a valid (non-negative) input number');
  return;
}
```

3. Tambahkan komponen `<IonAlert>`

```
return (
  // <= ~ <React.Fragment>
  <=
    <IonAlert
      isOpen={!error}
      message={error}
      buttons={[
        {text: 'Okay', handler: clearError}
      ]} />
    <IonApp>
      <IonHeader ...>
      <IonContent className="ion-padding" ...>
    </IonApp>
  </>
)
```

4. Jalankan aplikasi di browser dan coba input dengan nilai yang salah.



BMI Calculator

Tinggi Badan (cm)
170

Berat Badan (kg)
0

CALCULATE RESET

Please enter a valid (non-negative) input number

OKAY

Menambahkan Segment Buttons

1. Buat file `./components/InputControl.tsx`
2. Tambahkan kode berikut pada file `InputControl.tsx`

```
const InputControl: React.FC = () => {
  return(
    <IonSegment value="cmkg">
      <IonSegmentButton value="cmkg">
        <IonLabel>cm/kg</IonLabel>
      </IonSegmentButton>
      <IonSegmentButton value="ftlbs">
        <IonLabel>ft/lbs</IonLabel>
      </IonSegmentButton>
    </IonSegment>
  );
};

export default InputControl;
```

3. Tambahkan `<InputControl />` di bagian atas input field pada App.tsx

```
<IonContent className="ion-padding">
  <IonGrid>
    <IonRow>
      <IonCol>
        <InputControl />
      </IonCol>
    </IonRow>
  </IonGrid>
</IonContent>
```



4. Lihat hasilnya di browser.

BMI Calculator

CM/KG FT/LBS

Tinggi Badan (cm)
170

Berat Badan (kg)
60

 CALCULATE  RESET

5. Modifikasi InputControl untuk menggunakan `props`

```
const InputControl: React.FC<{selectedValue: 'cmkg' | 'ftlbs'}> = props => {
  return(
    <IonSegment value={props.selectedValue}>
      <IonSegmentButton value="cmkg">
        <IonLabel>cm/kg</IonLabel>
      </IonSegmentButton>
      <IonSegmentButton value="ftlbs">

```

6. Tambahkan attribute `selectedValue` pada komponen InputControl

```
<InputControl selectedValue="cmkg" />
```

Connecting Components

1. Pada tahap ini, SegmentButton sudah bias ditampilkan, namun perhitungan BMI hanya menggunakan ukuran cm/kg. Lanjutkan langkah-langkah berikut ini agar perhitungan BMI dapat dilakukan sesuai dengan pilihan pada Segement Button.
2. Tambahkan `const useState calcUnits` pada App.tsx

```
const [ calcUnits, setCalcUnits ] = useState<'cmkg' | 'ftlbs'>( initialState: 'cmkg');
```

3. Modifikasi InputControl untuk menggunakan calcUnits

```
<InputControl selectedValue={calcUnits} />
```

4. Tambahkan props `onSelectValue` pada InputControl

```
const InputControl: React.FC<{
  selectedValue: 'cmkg' | 'ftlbs';
  onSelectValue: (value: 'cmkg' | 'ftlbs') => void
}> = props => {

  const inputChangeHandler = (event: CustomEvent) => {
    props.onSelectValue(event.detail.value);
  };

  return(
    <IonSegment value={props.selectedValue} onIonChange={inputChangeHandler}>
```

5. Tambahkan function selectCalcUnitHandler pada App.tsx

```
const selectCalcUnitHandler = (selectedValue: 'cmkg' | 'ftlbs') => {
  setCalcUnits(selectedValue);
};

return (
```

6. Tambahkan attribute **onSelectValue** pada komponen InputControl

```
<InputControl selectedValue={calcUnits} onSelectValue={selectCalcUnitHandler} />
```

7. Modifikasi **<IonLabel>** untuk menampilkan satuan sesuai Segment Button yang terpilih

```
<IonLabel position="floating">Tinggi Badan ({calcUnits === 'cmkg' ? 'cm' : 'feet'})</IonLabel>
```

8. Coba jalankan aplikasi di browser, apakah label input berubah sesuai dengan pilihan segment button.

BMI Calculator

CM/KG

FT/LBS

Tinggi Badan (feet)

Berat Badan (lbs)

CALCULATE

RESET

9. Lanjutkan dengan mengerjakan tugas No. 2.

BMI Calculator

CM/KG

FT/LBS

Tinggi Badan (feet)

5.85

Berat Badan (lbs)

158.7

CALCULATE

RESET

22.68

Normal

TUGAS

1. Buat file `./components/BmiResult.tsx`, jadikan hasil perhitungan BMI menjadi sebuah komponen React baru.
2. Modifikasi kode pada `App.tsx` sehingga perhitungan BMI disesuaikan dengan pilihan ukuran pada Segment Button.
1 cm = 0.0328 feet
1 kg = 2.2 lbs

REFERENSI

<https://reactjs.org/docs/components-and-props.html>

<https://ionicframework.com/docs/api/alert>

<https://ionicframework.com/docs/api/segment-button/>