



## Meeting 6 - Challenge 2



Apakah tampilan aplikasi kamu sudah seperti gambar di samping?

Coba kamu masukkan jawaban nama negara lokasi Big Ben ya!



Apakah saat kamu tekan button submit dapat diketahui jawaban kamu benar atau salah?

Agar jawaban yang kamu input dapat dicocokkan dengan kunci jawaban, maka kamu harus **menyimpan** dulu **jawaban** kamu pada suatu **state** ya!

Yuk pergi ke halaman selanjutnya dan ikuti arahnya ya!



Ketik penambahan code berikut pada file **GuessTheCountryScreen.js** kamu ya!



## Codes

```
import React, { useState } from 'react';

const GuessTheCountryScreen = () => {
  const [answer, setAnswer] = useState('');
  return (
    ...
    <View ...>
      <TextInput
        style={{ borderWidth: 1, width: '50%' }}
        placeholder='Write your answer'
        keyboardType='default'
        onChangeText={(text) => setAnswer(text)}
      />
      ...
    </View>
    <Text>{answer}</Text>
    ...
  )
};
```

Yuk simak penjelasan masing-masing code yang sudah ditambahkan!

1

```
const [answer, setAnswer] = useState('');
```

Kamu harus **mendeklarasikan** dulu **state answer** di dalam functional component **GuessTheCountryScreen** ya!

Ingat kamu akan menyimpan data pada state **answer** berupa string lho!



2

```
onChangeText={ (text) => setAnswer(text) }
```

Teks yang dimasukkan oleh user pada `<TextInput/>` dapat diolah dengan props `onChangeText`

Apa yang ditulis tersimpan pada bagian ini, yang diberi nama "text". Tulisan disini tidak harus "text" ya, dapat kamu sesuaikan nantinya!

**State answer**  
kamu harus diubah nilainya sesuai dengan yang tersimpan pada **text** ya!

3

```
<Text>{answer}</Text>
```

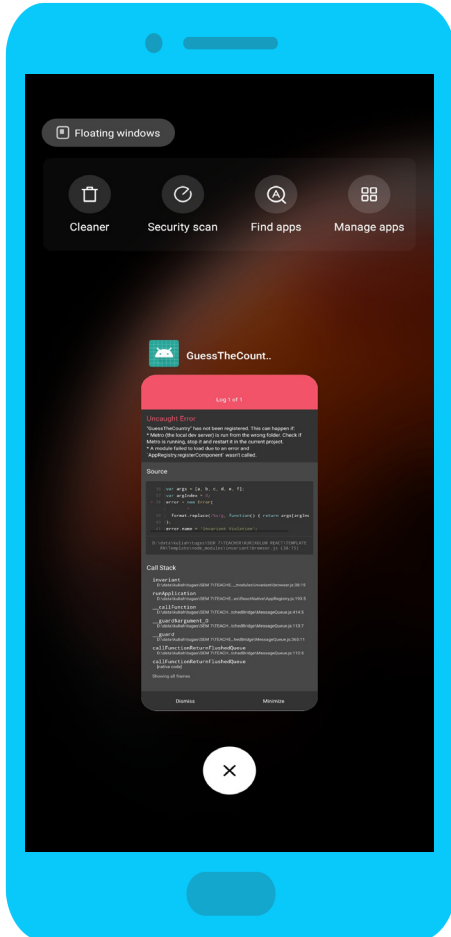
Panggil state answer kamu untuk memeriksa apakah yang di-input pada `<TextInput/>` sudah cocok dengan value pada state **answer**

Pergi ke halaman selanjutnya untuk melanjutkan progresmu ya!

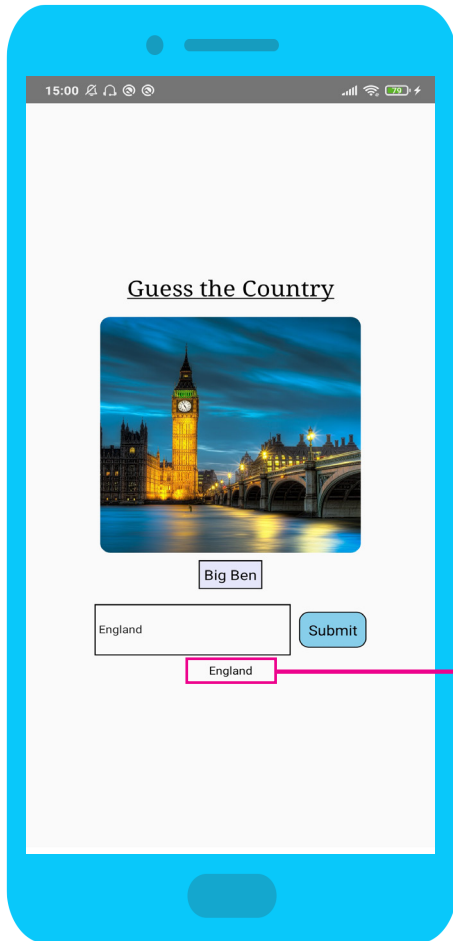


## CARA LAIN MENGATASI ERROR PADA APLIKASI

Solusi yang dapat kamu lakukan ketika aplikasi tidak berjalan sebagaimana mestinya walaupun sudah menekan huruf R di node terminal, yaitu melakukan **destroy app** dan **membuka ulang aplikasi** kamu melalui layar smartphone kamu ya!



Yuk pergi ke halaman berikutnya untuk melanjutkan progresmu ya!



Apakah tampilan aplikasi kamu sudah seperti gambar di samping?

Bukankah apa yang kamu ketik pada `<TextInput/>` juga akan muncul di sini?



Apakah game yang dibuat hanya memiliki gambar dan keterangan Big Ben saja? Tentu saja tidak ya!



Kamu sudah mengunduh file **Data.js** bukan sebelumnya? Data tersebutlah yang akan dimunculkan pada aplikasi secara random ya!



Coba buka file **Data.js** milikmu dan pahami strukturnya dengan membaca keterangan di bawah!

Terdapat 3 array, yaitu **countryList**, **objectNameList**, dan **objectImageList**

1. **countryList** = [] digunakan untuk menyimpan **daftar nama negara** yang berfungsi juga sebagai kunci jawaban
2. **objectNameList** = [] digunakan untuk menyimpan **daftar nama objek** yang ditampilkan untuk kamu tebak nantinya berada di negara mana
3. **objectImageList** = [] digunakan untuk menyimpan **daftar gambar objek** yang ditampilkan untuk kamu tebak nantinya berada di negara mana

Yuk sekarang ikuti langkah di bawah ini untuk menampilkan data yang berasal dari file **Data.js** kamu ya!

1. Lakukan import data yang diperlukan pada file **GuessTheCountryScreen.js** dengan code  

```
import { countryList, objectImageList, objectNameList }
from '../data/Data';
```

2. 

```
source={{uri: 'https://wallpaperaccess.com/full/180136.jpg' }}/>
```




**ganti code**

```
source={{ uri: objectImageList[0] }} />
```

3. 

```
<Text style={{ fontSize: 18 }}>Big Ben</Text>
```



**ganti code**

```
<Text style={{ fontSize: 18 }}>{objectNameList[0]}</Text>
```



Bukankah tampilan aplikasi kamu masih seperti gambar di samping?

Coba ganti index gambar dan keterangan yang dipanggil menjadi [2] ya!



Apakah yang terjadi?

Nah sekarang tugas kamu adalah membuat function untuk mengacak index yang digunakan untuk menampilkan objek!

Pergi ke halaman selanjutnya untuk melanjutkan progres ya!



Ketik penambahan code berikut pada file **GuessTheCountryScreen.js** ya!



## Codes

```
const GuessTheCountryScreen = () => {
  // letakkan di bawah code answer state
  const [index, setIndex] = useState();

  const randomIndex = () => {
    const pickRandomIndex = Math.floor(Math.random() *
    countryList.length);

    setIndex(pickRandomIndex);
  }
  // di sini code return
}
```

**randomIndex()** berfungsi untuk **mengacak angka** dari rentang **0 - 19** karena jumlah elemen pada array **countryList** adalah 20 ya!

Angka random yang didapat kemudian disimpan ke dalam **state index**.

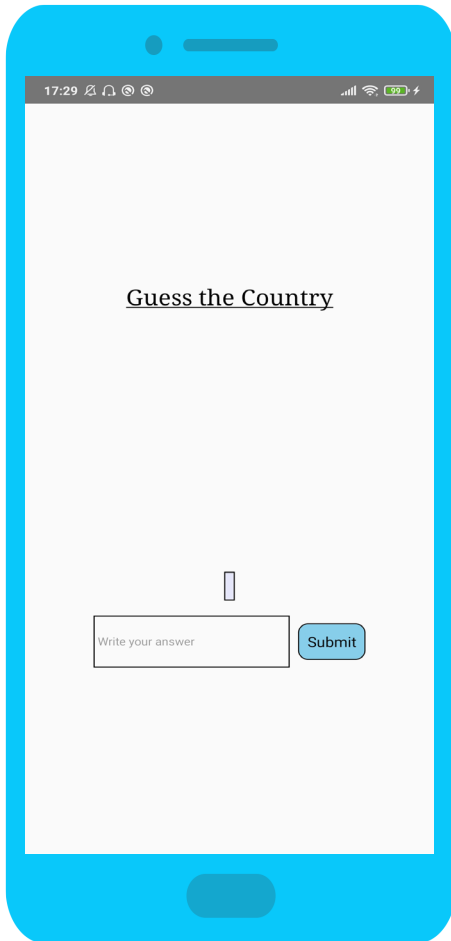


Tugas kamu sekarang adalah mengganti code berikut!

1. `source={{ uri: objectImageList[0] }} />`
2. `<Text style={{ fontSize: 18 }}>{objectNameList[0]}</Text>`

**Ganti angka 0** pada code nomor 1 dan 2 di atas menjadi **'index'**





Apakah tampilan aplikasi kamu seperti gambar di samping?

Hal tersebut terjadi karena kamu **belum memanggil function** `randomIndex()`, maka state index valuenya masih null.



Ingatkah kamu hook apa yang seharusnya digunakan untuk memanggil `randomIndex()` saat aplikasi pertama kali dijalankan?

Pergi ke halaman selanjutnya untuk melanjutkan progres ya!



Ketik penambahan code berikut pada file **GuessTheCountryScreen.js** ya!



### Codes

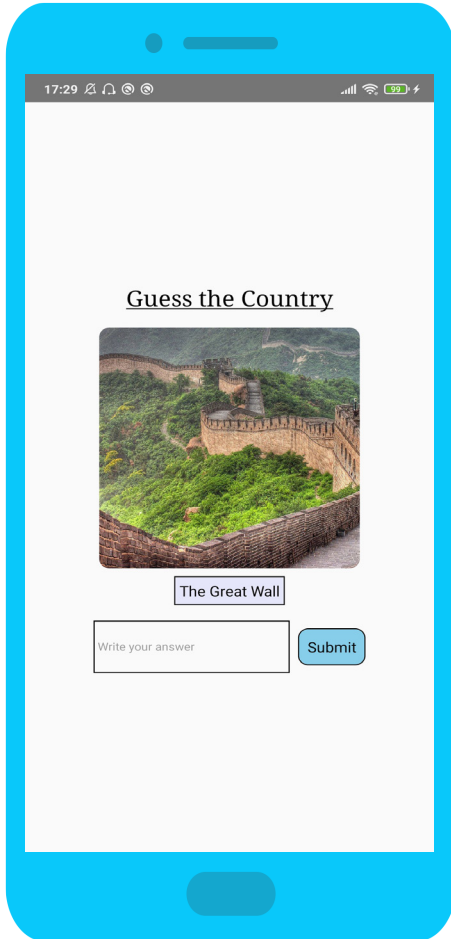
```
import React, { useState, useEffect } from 'react';

const GuessTheCountryScreen = () => {
  // letakkan di bawah code function randomIndex()
  useEffect() => {
    randomIndex();
  }, []);
  // di sini code return
}
```



Menurutmu mengapa perlu memanggil function `randomIndex()` di dalam `useEffect()`?

Pergi ke halaman selanjutnya untuk melanjutkan progres ya!



Apakah tampilan aplikasi kamu seperti gambar di samping?

Mungkin saja objek dan keterangan objek yang muncul berbeda ya, karena value state index diacak oleh aplikasi.

Nah sekarang baru kamu mulai masuk ke dalam logika untuk **memeriksa** apakah **jawaban** yang kamu input itu **benar** atau **salah** ya!

Pergi ke halaman selanjutnya untuk melanjutkan progres ya!



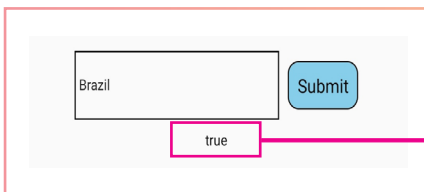
Ketik penambahan code berikut pada file **GuessTheCountryScreen.js** ya!



### Codes

```
const GuessTheCountryScreen = () => {
  // letakkan di bawah code index state
  const [result, setResult] = useState('');

  // letakkan function checkAnswer setelah function randomIndex()
  const checkAnswer = () => {
    if (answer.toLowerCase() === countryList[index]) {
      setResult('true');
    }
    else {
      setResult('false');
    }
    randomIndex();
  }
};
```



Coba kamu panggil **value** dari **state result** dan letakkan seperti gambar di samping ya!



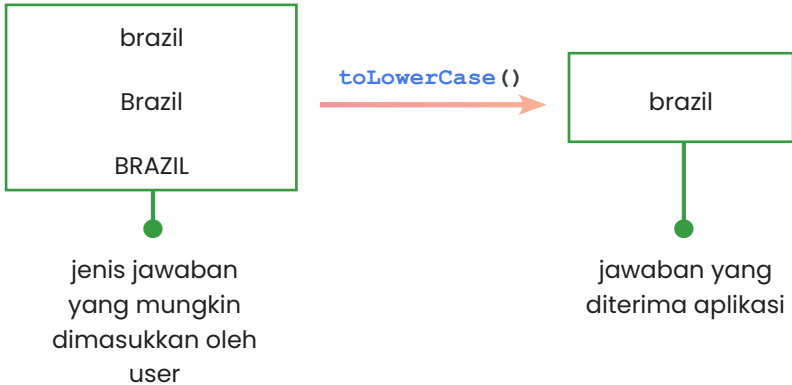
Dapatkan kamu mengerti logika dari function **checkAnswer()** ?



```
if (answer.toLowerCase() === countryList[index])
```

function `toLowerCase()` berfungsi untuk mengubah jawaban kamu yang kamu input menjadi huruf kecil

### EXAMPLE



Jangan lupa kamu perlu memanggil function `checkAnswer()` ketika submit button kamu klik ya! Tambahkan `onPress` prop pada `<TouchableOpacity>`



### Codes

```
onPress={ () => checkAnswer() }
```

Nah sekarang coba masukkan jawaban kamu mengenai objek yang ditampilkan pada aplikasmu dan jangan lupa submit jawabanmu ya!



**BEFORE SUBMIT**



**AFTER SUBMIT**



Apakah kamu mengalami kendala seperti gambar di atas?

`<TextInput/>` seharusnya menjadi bersih tanpa jawaban dan value state result harusnya selesai ditampilkan bukan?

Yuk kembali ke buku kamu untuk mempelajari materi baru dulu ya!