



Fundamental React Native

Silver- Chapter 2 - Topic 4

**Selamat datang di Chapter 2 Topic 4 pada
course React Native dari Binar Academy!**





Haalloo Guys! 🙌

Di Chapter 2 Topic 3 kemarin kita sudah belajar tentang komponen React dan cara kerjanya ya. Nah sekarang, di dalam Topic 4 ini, kita bakal belajar tentang **Fundamental React Native**. Nanti di akhir sesi, akan ada latihan yang harus kamu kerjain, jadi pastikan kamu paham materi ini ya.

Ayo kita mulai!





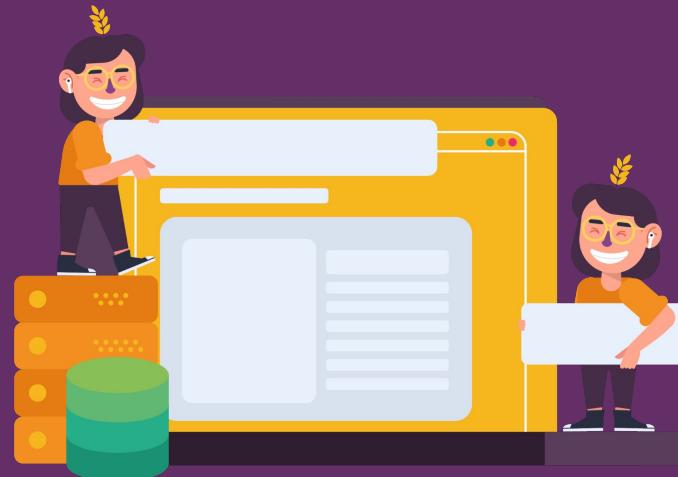
Detailnya, kita bakal bahas hal-hal berikut ini:

- Penulisan Struktur Tag pada React Native
- Pembuatan layout basic pada React Native
- Struktur folder pada React Native





Pada sesi ini, kamu akan bikin aplikasi mobile pertama react native. Hal pertama yang perlu kamu lakukan adalah bikin file projectnya, yaitu dengan melakukan **Setup Project**.





Yang pertama banget, Setup Project!

Sebelum bikin aplikasi mobile, ikuti langkah setup project-nya dulu yuk!

1. Pastikan semua environment node.js, npm, java, android studio, dan configuration path sudah terinstall di laptop.
2. Bikin project baru via terminal. Misalnya, kamu mau nama project-nya “dasar-react-native”, cukup klik command di terminal kayak gini:

```
$ npx react-native init dasar-react-native
```

1. Setelah download selesai, bakal muncul folder baru namanya dasar-react-native. Buka foldernya pake visual studio code.





Nah, kalau proses setup sudah selesai, kamu bisa buka folder project react native. Di situ bakal muncul beberapa file dan folder dalam folder project.

Ngomongin folder project, meski sekilas kayaknya sama, tapi **ada beberapa perbedaan dalam struktur folder** antara React Native dengan React.JS yang udah kamu pelajari sebelumnya.

Inget kan?





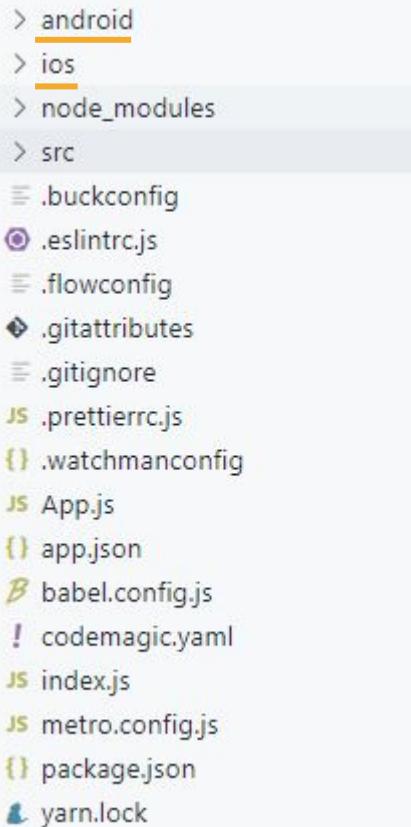
Folder-folder ini kayak kembar identik~

Folder yang serupa tapi nggak sama ini kalau kita analogikan kayak anak kembar identik. Sekilas mereka kelihatan mirip banget, nggak ada bedanya. Sama-sama berambut lurus, berkulit gelap, dan bermata besar. Meskipun terlihat sama, tapi sebenarnya ada ciri yang beda. Si A punya gigi gingsul, sementara Si B nggak. Si B pake kacamata, Si A nggak.

Nah, sama kayak struktur folder React Native dan React.JS, meski sebagian besar sama tapi mereka punya beberapa perbedaan lho!

Apa bedanya? Kita cek yuk!





Pertama kita cari tahu dulu yuk di **Folder Structure** ada apa aja sih?

android

Folder ini berisi **script atau code Java/Kotlin (Native Android)** dan berfungsi untuk build aplikasi react native ke dalam bentuk aplikasi android.

ios

Folder ini **script atau code Objective-C/Swift (Native iOS)** dan juga berfungsi untuk build aplikasi react native ke dalam bentuk aplikasi iOS/iPad.



```
> android
> ios
> node_modules
> src
≡ .buckconfig
● .eslintrc.js
≡ .flowconfig
◆ .gitattributes
≡ .gitignore
JS .prettierrc.js
{ .watchmanconfig
JS App.js
{ app.json
B babel.config.js
! codemagic.yaml
JS index.js
JS metro.config.js
{ package.json
L yarn.lock
```

node_modules

Folder yang berisi **paket-paket module**. Semua library yang kamu install dengan npm akan tersimpan di sini.

src

Folder ini berisi **kode-kode react native** yang akan kamu tulis. Di dalam folder ini kamu bisa nambahin folder-folder baru sesuai kebutuhan. Misalnya nambahin folder 'assets' di dalam 'src' untuk nyimpan gambar-gambar yang digunakan pada aplikasi mobile-mu.



.gitignore

file yang berisi **list dari file atau folder** yang akan diabaikan oleh git.



App.js

File ini berisi **kode component App atau komponen utama** aplikasi.



```
> android
> ios
> node_modules
> src
≡ .buckconfig
❶ .eslintrc.js
≡ .flowconfig
❖ .gitattributes
≡ .gitignore
JS .prettierrc.js
{} .watchmanconfig
JS App.js
{} app.json
B babel.config.js
! codemagic.yaml
JS index.js
JS metro.config.js
{} package.json
❸ yarn.lock
```



metro.config.js

File ini berisi **kode konfigurasi metro bundler** ketika mengompilasi file JavaScript menjadi file-file Native.



package.json

File JSON yang berisi **keterangan project dan daftar package- package eksternal** yang dibutuhkan.

Hayoo ada yang sadar bedanya nggak nih? Boleh kok intip-intip topik sebelumnya untuk refresh ingatan kalau kamu lupa ;)



Lanjutttt..

Setelah kenalan sama struktur folder, sekarang coba kita jalankan aplikasi react native-nya dulu yuk.

Langsung ketik perintah **npm run android** di terminal ya.

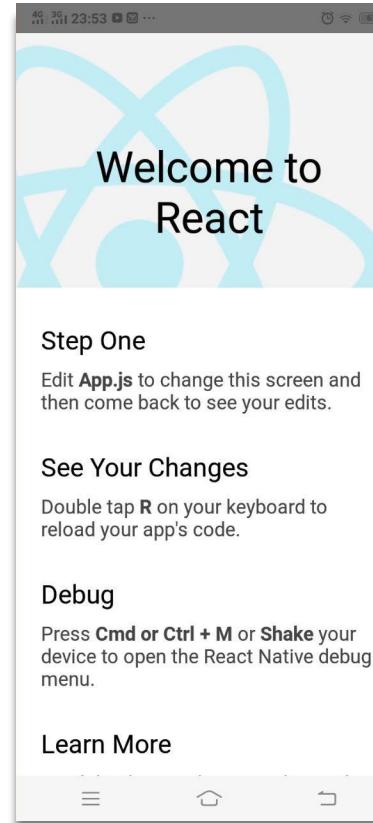




Setelah proses compile selesai, bakal muncul ikon aplikasi baru di layar androidmu. Coba kamu buka aplikasinya deh, nanti bakal muncul tampilan ‘Welcome to React’ kayak gambar di samping.

Selamat! Ini berarti kamu udah berhasil menjalankan aplikasi react native pertama di androidmu. Gimanaaa, makin seru kan?

Nah lanjut kita bahas tentang tag struktur penulisan di React Native yuk!





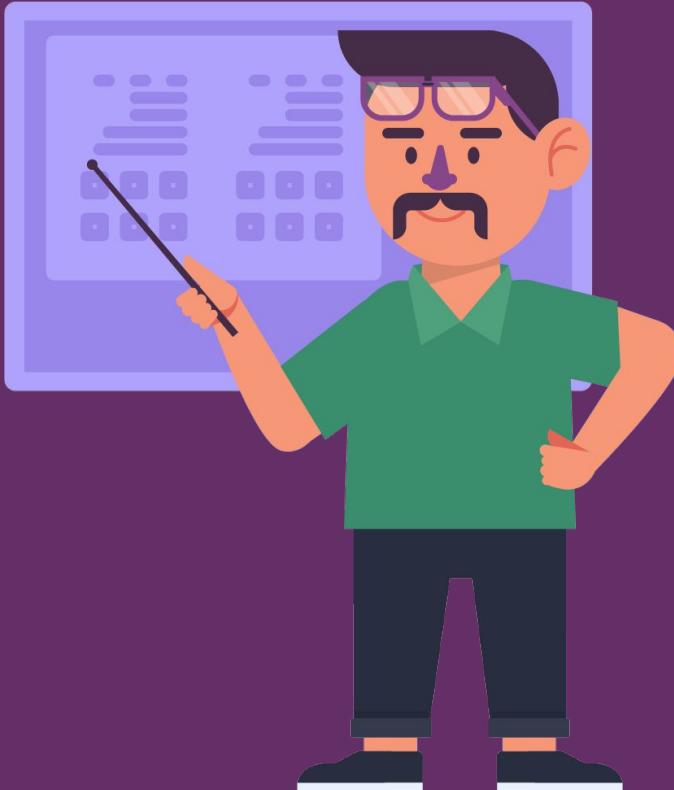
Setelah belajar HTML, pasti kamu udah familiar sama cara penulisan dan cara penggunaannya kodenya, kan?





**Cara menulis kode React Native bisa
dibilang mirip banget kayak cara penulisan
kode di React.JS/HTML, cuma memang ada
beberapa perbedaan Tag dasar yang harus
kamu tahu..**

Yuk cekidot!

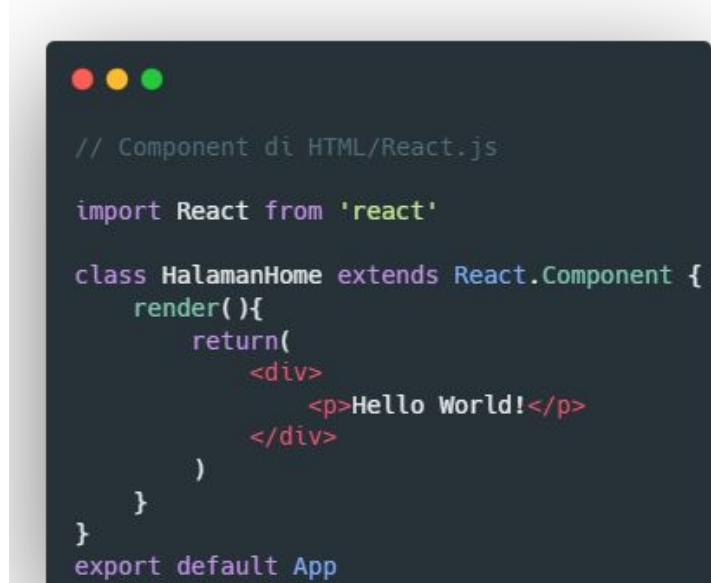




Begini nih kalau tag di HTML:

Tag HTML	Fungsi
<div>	Membuat blok element
 / <p>	Menampilkan ext
<input type='text'>	Menampilkan Input Text
<a> / <button>	Menampilkan Button
	Menampilkan Gambar

Di HTML atau React.JS, kamu pasti bakalan pakai beberapa tag yang ada pada tabel di atas pas mau bikin component.



```
// Component di HTML/React.js

import React from 'react'

class HalamanHome extends React.Component {
  render(){
    return(
      <div>
        <p>Hello World!</p>
      </div>
    )
  }
}

export default App
```



Perhatiin bedanya dengan tag yang dipakai di React Native:

Tag di React Native	Fungsi
<View>	Membuat blok element
<Text>	Menampilkan Text
<InputText>	Menampilkan Input Text
<Button>	Menampilkan Button
<Image>	Menampilkan Gambar



Tag Pada React Native

Beda dengan React Native, semua tag HTML bisa kita langsung pakai **tanpa harus diimport**. Sementara itu, semua tag pada React Native, **harus di-import dulu dari 'react-native'**, baru bisa dipakai.

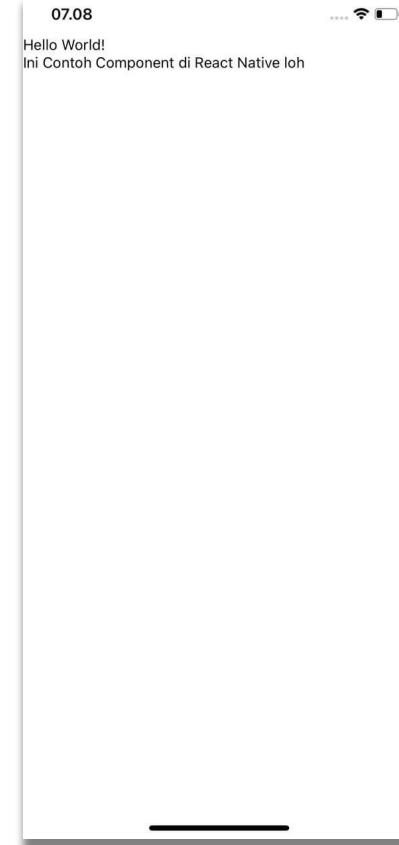
```
//Buat Component di React Native

import React from 'react';
// di React Native kamu Harus Import terlebih dahulu
// semua tag yang ingin kamu pake
import { View, Text } from 'react-native';
```



Jadinya seperti ini bentuk hasil coding di React Native:

```
● ● ●  
//Buat Component di React Native  
  
import React from 'react';  
// di React Native kamu Harus Import terlebih dahulu  
// semua tag yang ingin kamu pake  
import { View, Text } from 'react-native';  
  
class HalamanHome extends React.Component {  
    render(){  
        return(  
            <View>  
                <Text>Hello World!</Text>  
                <Text>Ini Contoh Component di React Native loh</Text>  
            </View>  
        )  
    }  
}  
export default HalamanHome
```





Tahu struktur folder? Checked! 🤓
Paham struktur tag? Checked! 🤓
Sekarang saatnya kamu belajar empat
Gaya Penulisan di dunia pemrograman.





Gaya Penulisan

Kalau pelajaran Bahasa Indonesia punya materi gaya bahasa, di dunia programming pun ada materi gaya penulisan. Yak! Ada empat jenis gaya penulisan yang umum dipakai di dunia pemrograman, yaitu:

1. Snake Case

Snake case adalah penulisan yang cirinya **menggunakan garis bawah atau underscore (_)**. Contohnya kayak gambar di samping ya!

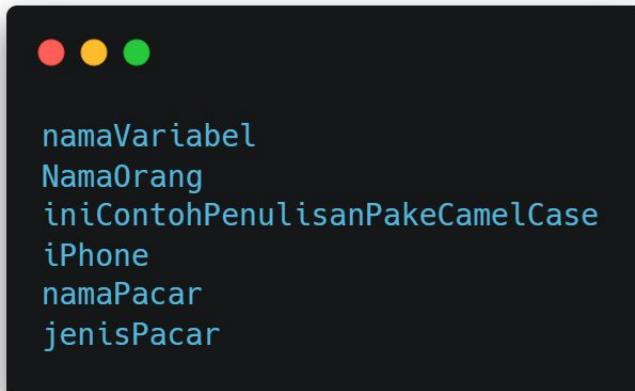


```
nama_variable  
ini_cantoh_penulisan_snake_case  
Nama_class  
nama_pacar
```



2. Camel Case

Ciri camel case adalah **semua suku kata menyatu**, nggak ada spasi, tapi ada huruf kapital untuk memisahkan kata. Contohnya kayak gambar di bawah ini ya!



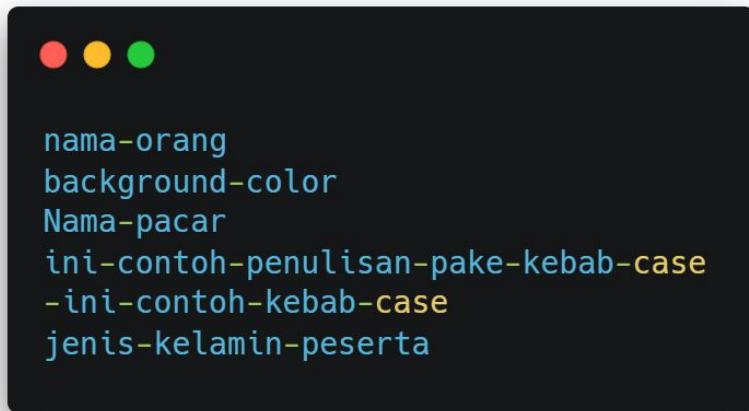
```
● ● ●  
namaVariabel  
NamaOrang  
iniContohPenulisanPakeCamelCase  
iPhone  
namaPacar  
jenisPacar
```





3. Kebab Case

Ciri kebab case adalah **pakai tanda minus (-)** untuk memisahkan antar suku kata, sehingga bentuknya terlihat seperti kebab. Contohnya kayak gambar di bawah ini ya!



```
nama-orang
background-color
Nama-pacar
ini-contoh-penulisan-pake-kebab-case
-ini-contoh-kebab-case
jenis-kelamin-peserta
```



kebab-case

4. All Case

Ciri all case ini adalah **semua pakai huruf kapital**. Biasanya sering digunakan untuk menuliskan nama sebuah konstanta. Contohnya kayak gambar di samping ya!





Selain gaya penulisan, ada juga yang disebut **React Native Styling**. Antara CSS dan React Native ternyata **styling-nya berbeda**. Mau tahu bedanya?

Cuss klik slide selanjutnya!





Biar gak bingung waktu ngoding, simak dulu bedanya yuk~

Ini nih beberapa perbedaan aturan dalam penulisan di CSS dan React Native, guys. Perhatikan ya:

1. Pada CSS, penulisan syntax pakai kebab case, sedangkan di React Native pakai camel case.
2. Pada CSS, penulisan value properti yang bersifat string ditulis tanpa tanda kutip. Di React Native value property yang bersifat string ditulis di dalam tanda kutip satu ('') atau kutip dua ("")



Diagram illustrating the difference in CSS syntax between React Native and standard CSS:

- Property with Kebab-CASE:** `text-align: center;` (highlighted in orange)
- Value Property format String without quotes ("):** `center` (highlighted in blue)
- Terminates with a brace (}):** `}` (highlighted in orange)

```
// CSS Propertinya pake KEBAB-CASE
// dan Value properti yang bentuknya string, dituliskan
// tanpa tanda kutip
{
  text-align: center;
}
```

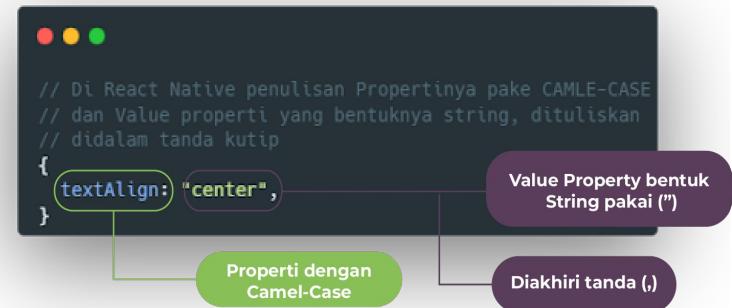


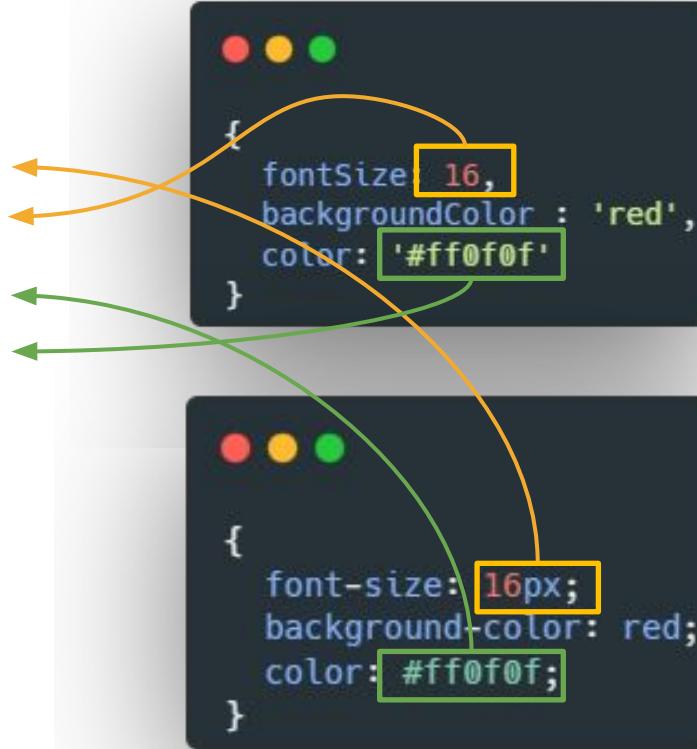
Diagram illustrating the difference in CSS syntax between React Native and standard CSS:

- Property with Camel-Case:** `textAlign:` (highlighted in green)
- Value Property format String with quotes ('' or ""):** `"center"` (highlighted in blue)
- Terminates with a brace (}):** `}` (highlighted in purple)

```
// Di React Native penulisan Propertinya pake CAMLE-CASE
// dan Value properti yang bentuknya string, dituliskan
// didalam tanda kutip
{
  textAlign: "center",
}
```



3. Pada CSS, properti value yang bersifat ukuran bisa dibuat dengan **satuan 'px'**, sedangkan di react native dibuat **jadi number**.
4. Pada CSS, properti value warna hexadecimal bisa **ditulis langsung** tanpa tanda kutip, sedangkan di React Native **harus di dalam tanda kutip**.



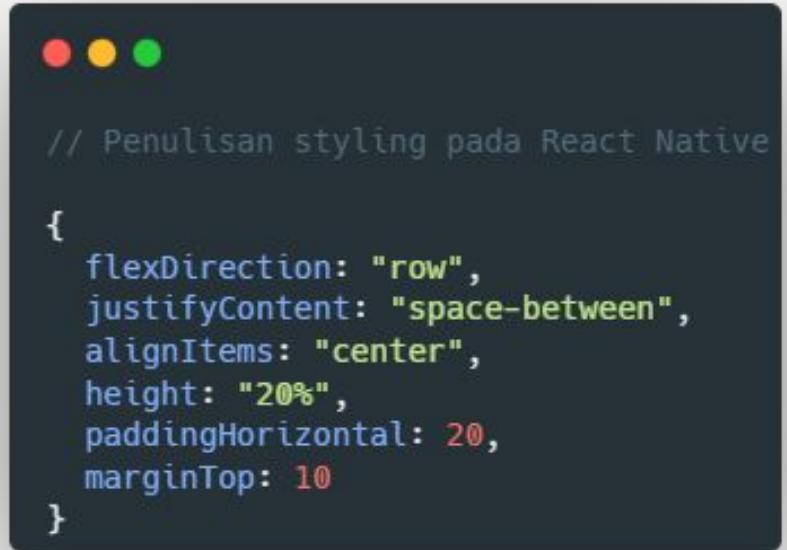
```
{  
  fontSize: 16,  
  backgroundColor: 'red',  
  color: '#ff0f0f'  
}  
  
{  
  font-size: 16px;  
  background-color: red;  
  color: #ff0f0f;  
}
```



Biar lebih jelas bedanya, coba perhatikan gambar di bawah ini ya!



```
// Penulisan styling pada CSS
{
  flex-direction: row;
  justify-content: space-between;
  align-items: center;
  height: 20%;
  padding-horizontal: 20px;
  margin-top: 10px;
}
```



```
// Penulisan styling pada React Native
{
  flexDirection: "row",
  justifyContent: "space-between",
  alignItems: "center",
  height: "20%",
  paddingHorizontal: 20,
  marginTop: 10
}
```



Selanjutnya, tiga **Metode Styling** dalam React Native bakal jadi materi terakhir kita di topik 4.

Masih semangat kan nihh? ;)





Sebelum membahas metode penulisan styling, kira-kira bagaimana ya caranya bikin tampilan sederhana seperti gambar di bawah?



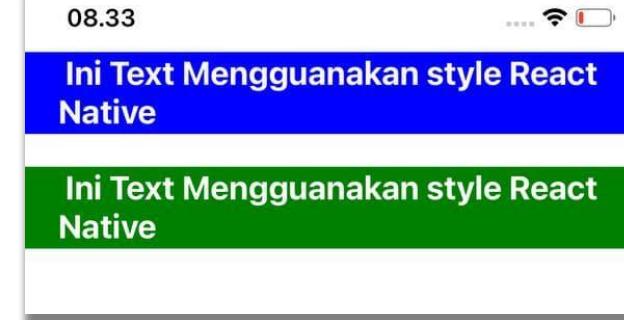
Jangan bingung kalau belum tahu jawabannya, geng.
Langsung cek caranya di slide berikut.





Ada tiga metode penulisan styling pada React Native yang bisa kamu pilih, untuk bikin tampilan kayak gambar di samping, yaitu:

1. **Inline**, membuat style di dalam component yang dijadikan sebagai props.
2. **Embed/Internal**, membuat style dengan React Native Stylesheet di dalam file yang sama.
3. **External**, membuat style dengan React Native StyleSheet di dalam file yang berbeda.





Penulisan Inline

Memasukan object styling
ke dalam props style.
Setiap Tag React Native
selalu mempunyai props style.

```
import React from 'react';
import { View, Text } from 'react-native';

function App() {
  return (
    //Code style langsung ditulisan dalam component contohnya seperti berikut ini
    <View
      style={{
        height: 50,
        width: '100%',
        backgroundColor: 'blue',
        justifyContent: 'center',
        alignItems: 'center',
      }}
    >
      <Text style={{ fontSize: 20, color: 'white', fontWeight: 'bold' }}>
        Ini Text Menggunakan style React Native
      </Text>
    </View>
  );
}

export default App;
```



Penulisan Embed

Import method `StyleSheet` terlebih dulu.

Panggil method `StyleSheet.create()`,
lalu masukin kode styling-mu di
dalam method tersebut.

Simpan return method tersebut
ke dalam sebuah variabel.

Kemudian panggil nama property
yang sudah kamu tulis pada Method
`StyleSheet.Create()`
di dalam props style.

```
● ● ●

import React from 'react';
import { View, Text, StyleSheet } from 'react-native';

const KodeStyling = StyleSheet.create({
  elemenContainer: {
    height: 50,
    width: '100%',
    backgroundColor: 'blue',
    justifyContent: 'center',
    alignItems: 'center',
  },
  textStyling: { fontSize: 20, color: 'white', fontWeight: 'bold' },
});

function App() {
  return (
    //Masukan object yang telah dibuat pada Variable KodeStyling
    <View style={KodeStyling.elemenContainer}>
      <Text style={KodeStyling.textStyling}>
        Ini Text Menggunakan style React Native
      </Text>
    </View>
  );
}

export default App;
```



Penulisan External

Buat kode object style react native
di file terpisah, kamu boleh kasih
nama filenya `style.js`

```
●●●  
1 // file style.js  
2  
3 import { StyleSheet } from 'react-native';  
4  
5 const Style = StyleSheet.create({  
6   elemenContainer: {  
7     height: 50,  
8     width: '100%',  
9     backgroundColor: 'blue',  
10    justifyContent: 'center',  
11    alignItems: 'center'  
12  },  
13  textStyling: {  
14    fontSize: 20,  
15    color: 'white',  
16    fontWeight: 'bold'  
17  }  
18 })
```

Kemudian import file `style.js` tersebut

Panggil variabel import yang berasal dari
hasil import file `style.js`

```
●●●  
1 import React from 'react';  
2 import { View, Text } from 'react-native';  
3  
4 // Import file styling yang telah kamu buat terpisah  
5 import KodeStyling from './style'  
6  
7  
8  
9 function App(){  
10  // Masukan object yang telah dibuat dan di import dari file styleing  
11  <View style={KodeStyling.elemenContainer}>  
12    <Text style={KodeStyling.textStyling}>  
13      Int Text Menggunakan style React Native  
14    </Text>  
15  </View>  
16 }
```

Yeayy! Materi Chapter 2 React Native sudah selesai. Sekarang saatnya latihan!

Coba kamu buat layout sesuai dengan gambar di samping, lalu jalankan aplikasinya di Android kamu. Good luck!

Link gambar logo binar bisa diambil di sini ya:
https://res.cloudinary.com/diyjm9ayd/image/upload/v164324917/19025216_1439860629412992_3671167199250762358_o_hzryz8.png



Saatnya kita Quiz!





1. Dibawah ini manakah pernyataan yang tidak benar?

- A. Pada react native harus menggunakan bawaan Core Components dari react native seperti Text, Image, View, dll
- B. Tag **** pada react native berfungsi membuat text tulisan tebal di mobile
- C. Tag **<p>** pada react berfungsi untuk membuat paragraph pada React.JS



1. Dibawah ini manakah pernyataan yang tidak benar?

- A. Pada react native harus menggunakan bawaan Core Components dari react native seperti Text, Image, View, dll
- B. Tag **** pada react native berfungsi membuat text tulisan tebal di mobile
- C. Tag **<p>** pada react berfungsi untuk membuat paragraph pada React.JS

Tidak ada tag **** dan tag html lainnya pada React Native



2. Apa yang terjadi pada apabila syntax kode di samping dijalankan?

- A. Ukuran text akan menjadi 20 pixel
- B. Error: Invalid styling properti value for 'fontSize'
- C. Ukuran text tidak akan berubah

```
● ● ●  
1 import React from 'react';
2 import { View, Text, StyleSheet } from 'react-native';
3
4 const StyleComponent = StyleSheet.create({
5   textStyle: {
6     fontSize: 20px
7   }
8 })
9
10 function App(){
11   <View>
12     <Text style={StyleComponent.textStyle}>
13       Ini Text Mengguanakan style React Native
14     <Text>
15   </View>
16 }
```



2. Apa yang terjadi pada apabila syntax kode di samping dijalankan?

- A. Ukuran text akan menjadi 20 pixel
- B. Error: Invalid styling properti value for 'fontSize'
- C. Ukuran text tidak akan berubah

Tidak ada satuan ukuran di React Native, yang ada hanya angka saja



```
1 import React from 'react';
2 import { View, Text, StyleSheet } from 'react-native';
3
4 const StyleComponent = StyleSheet.create({
5   textStyle: {
6     fontSize: 20px
7   }
8 })
9
10 function App(){
11   <View >
12     <Text style={StyleComponent.textStyle}>
13       Ini Text Mengguanakan style React Native
14     <Text>
15   </View>
16 }
```



3. Apa yang terjadi pada apabila syntax kode di samping dijalankan?

- A. Text akan berubah menjadi warna merah, dan ukuran text menjadi 20
- B. Text akan berubah menjadi warna biru, dan ukuran text tidak berubah menjadi 20
- C. Text akan berubah menjadi warna biru, dan ukuran text menjadi 20

```
● ● ●  
1 import React from 'react';
2 import { View, Text, StyleSheet } from 'react-native';
3
4 const StyleComponent = StyleSheet.create({
5   textStyle1: {
6     fontSize: 20,
7     color: 'red'
8   },
9   textStyle2: {
10     color: 'blue'
11   }
12 })
13
14 function App(){
15   <View>
16     <Text style={[StyleComponent.textStyle1, StyleComponent.textStyle2]}>
17       Ini Text Menggunakan style React Native
18     <Text>
19     </View>
20 }
```



3. Apa yang terjadi pada apabila syntax kode di samping dijalankan?

- A. Text akan berubah menjadi warna merah, dan ukuran text menjadi 20
- B. Text akan berubah menjadi warna biru, dan ukuran text tidak berubah menjadi 20
- C. Text akan berubah menjadi warna biru, dan ukuran text menjadi 20

Apabila terdapat dua style, kemudian digabung menjadi satu, maka properti yang sama pada style yang ditulis di paling akhir sisi kananlah yang akan ter-apply

```
● ● ●  
1 import React from 'react';
2 import { View, Text, StyleSheet } from 'react-native';
3
4 const StyleComponent = StyleSheet.create({
5   textStyle1: {
6     fontSize: 20,
7     color: 'red'
8   },
9   textStyle2: {
10     color: 'blue'
11   }
12 })
13
14 function App(){
15   <View>
16     <Text style={[StyleComponent.textStyle1, StyleComponent.textStyle2]}>
17       Ini Text Menggunakan style React Native
18     </Text>
19   </View>
20 }
```

Terima Kasih!



Next Topic

loading...