



# React and React Native ○ Basics

---

React Native : Styling and Layout

15

## PERHATIAN

Jangan lupa untuk isi form absensi. Bagi instruktur mohon mengisi form absensi yang telah diinfokan dan konfirmasi nomor urut peserta yang hadir.

Untuk student harap mengisi form absensi [di sini](#) sebelum kelas dimulai. Untuk kode peserta dapat ditanyakan kepada instruktur dan jangan lupa mencantumkan pertemuan ke 1





# React Native<sup>+</sup> Fundamentals

## React Native : Styling and Layout - 15

# REACT NATIVE FUNDAMENTALS

### CORE COMPONENTS

Diambil dari dokumentasi resminya, berikut daftar CORE COMPONENTS dari React Native beserta penjelasannya

REACT NATIVE UI COMPONENT	ANDROID VIEW	IOS VIEW	WEB ANALOG	DESCRIPTION
<code>&lt;View&gt;</code>	<code>&lt;ViewGroup&gt;</code>	<code>&lt;UIView&gt;</code>	A non-scrolling <code>&lt;div&gt;</code>	A container that supports layout with flexbox, style, some touch handling, and accessibility controls
<code>&lt;Text&gt;</code>	<code>&lt;TextView&gt;</code>	<code>&lt;UITextView&gt;</code>	<code>&lt;p&gt;</code>	Displays, styles, and nests strings of text and even handles touch events
<code>&lt;Image&gt;</code>	<code>&lt;ImageView&gt;</code>	<code>&lt;UIImageView&gt;</code>	<code>&lt;img&gt;</code>	Displays different types of images
<code>&lt;ScrollView&gt;</code>	<code>&lt;ScrollView&gt;</code>	<code>&lt;UIScrollView&gt;</code>	<code>&lt;div&gt;</code>	A generic scrolling container that can contain multiple components and views
<code>&lt;TextInput&gt;</code>	<code>&lt;EditText&gt;</code>	<code>&lt;UITextField&gt;</code>	<code>&lt;input type="text"&gt;</code>	Allows the user to enter text



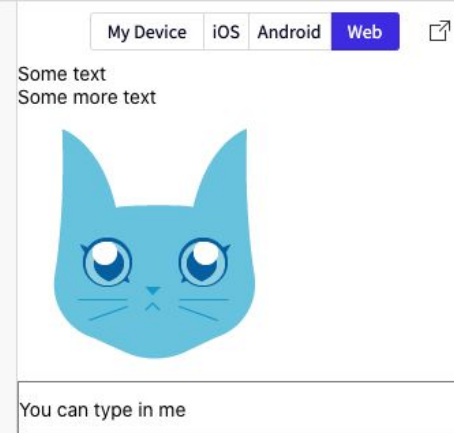
## React Native : Styling and Layout - 15

# REACT NATIVE FUNDAMENTALS

### CORE COMPONENTS

.. dan berikut penampakan kode dan hasilnya ketika kita masukkan semua core components nya di 1 component

```
1 import React from 'react';
2 import { View, Text, Image, ScrollView, TextInput } from 'react-native';
3
4 const App = () => {
5   return (
6     <ScrollView>
7       <Text>Some text</Text>
8       <View>
9         <Text>Some more text</Text>
10        <Image
11          source={{
12            uri: 'https://reactnative.dev/docs/assets/p_cat2.png',
13          }}
14          style={{ width: 200, height: 200 }}
15        </Image>
16      </View>
17      <TextInput
18        style={{
19          height: 40,
20          borderColor: 'gray',
21          borderWidth: 1
22        }}
23        defaultValue="You can type in me"
24      </TextInput>
25    </ScrollView>
26  );
27 }
28
29 export default App;
```



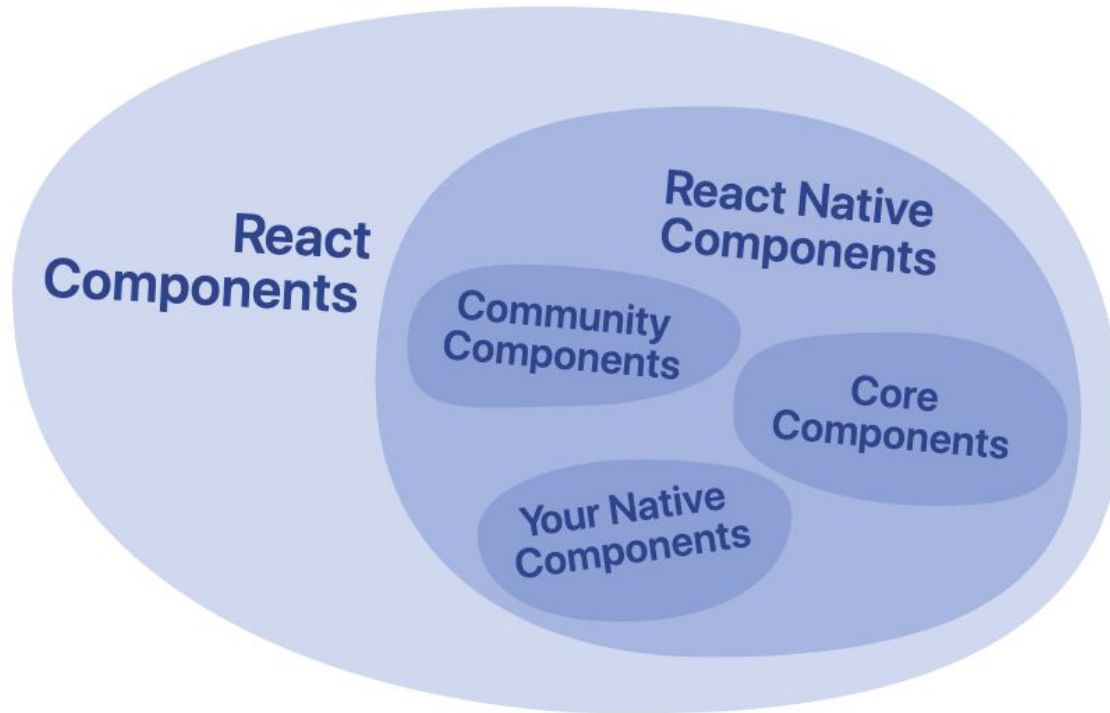
HACKTIV8

## *React Native : Styling and Layout - 15*

# REACT NATIVE FUNDAMENTALS

### COMPONENTS RELATION

Dan karena kita sudah belajar React, maka kita dapat memakai nya sebagai dasar untuk melanjutkan ke React Native



## React Native : Styling and Layout - 15

# REACT NATIVE FUNDAMENTALS

### FLEXBOX IN REACT NATIVE

Familiar dengan Flexbox di CSS3 ? React native pun menyediakannya. Bedanya, default direction nya adalah **column**

Update function return dan tambahkan stylesheet kita menjadi seperti ini :

```
<View style={styles.container}>  
  <View style={styles.box1} />  
  <View style={styles.box2} />  
  <View style={styles.box3} />  
  <StatusBar style="auto" />  
</View>
```

```
container: {  
  flex: 1  
},  
box1: {  
  flex: 1,  
  backgroundColor: 'red'  
},  
box2: {  
  flex: 1,  
  backgroundColor: 'yellow'  
},  
box3: {  
  flex: 1,  
  backgroundColor: 'green'  
}
```

Ini adalah behaviour default dari React Native. Flexbox direction secara default adalah column. Kemudian, karena container menggunakan nilai flex sebanyak 1, maka seluruh tinggi layar akan digunakan.

Box-box ( view ) yang ada di dalam container menggunakan flex sebanyak 1, berarti, tinggi layar akan dibagi rata

Next, kita akan coba menggunakan ukuran flex yang berbeda untuk setiap box nya



HACKTIV8

## React Native : Styling and Layout - 15

# REACT NATIVE FUNDAMENTALS

### FLEXBOX IN REACT NATIVE

Update stylesheet kita menjadi seperti ini :

```
container: {  
  flex: 1  
},  
box1: {  
  flex: 1,  
  backgroundColor: 'red'  
},  
box2: {  
  flex: 2,  
  backgroundColor: 'yellow'  
},  
box3: {  
  flex: 3,  
  backgroundColor: 'green'  
}
```

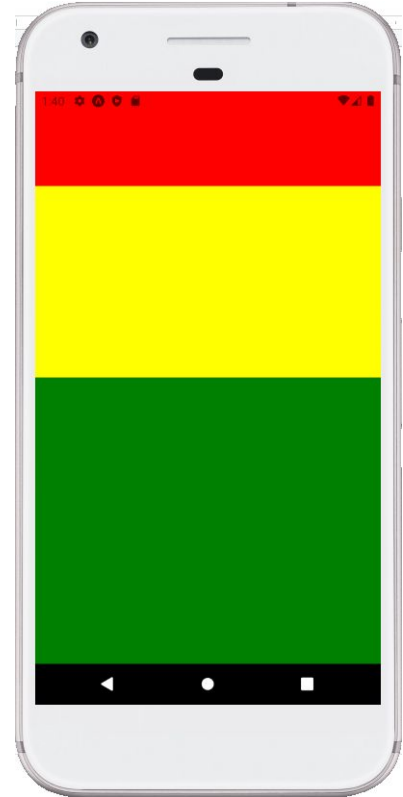
Cara menghitungnya seperti ini :

Jumlahkan total area flex yang diminta :  $1 + 2 + 3 = 6$

Kemudian, masing-masing box akan mendapatkan area sebanyak yang diminta. Contoh : box1 meminta 1 area flex, berarti, box 1 akan mendapatkan  $\frac{1}{6}$  dari tinggi layar

Begitupun untuk box2 dan box3

Next, bagaimana kalau kita coba ganti flex direction nya ?



HACKTIV8

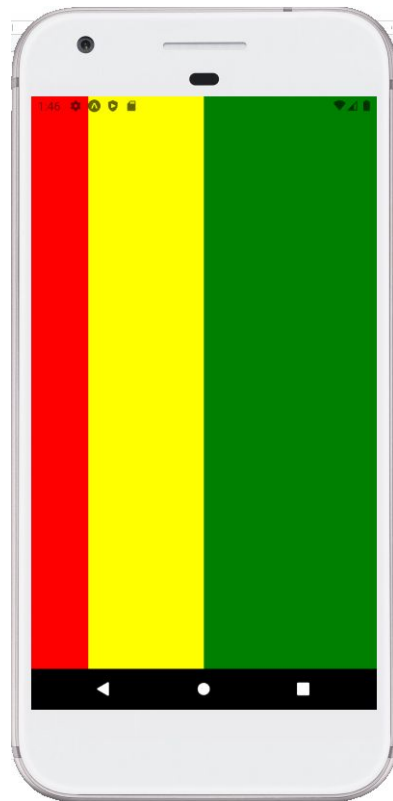


Update stylesheet kita menjadi seperti ini :

```
container: {  
  flex: 1,  
  flexDirection: 'row'  
},  
box1: {  
  flex: 1,  
  backgroundColor: 'red'  
},  
box2: {  
  flex: 2,  
  backgroundColor: 'yellow'  
},  
box3: {  
  flex: 3,  
  backgroundColor: 'green'  
}
```

Kita ubah flexDirection container kita menjadi **row**

Dan inilah yang terjadi =>



## React Native : Styling and Layout - 15

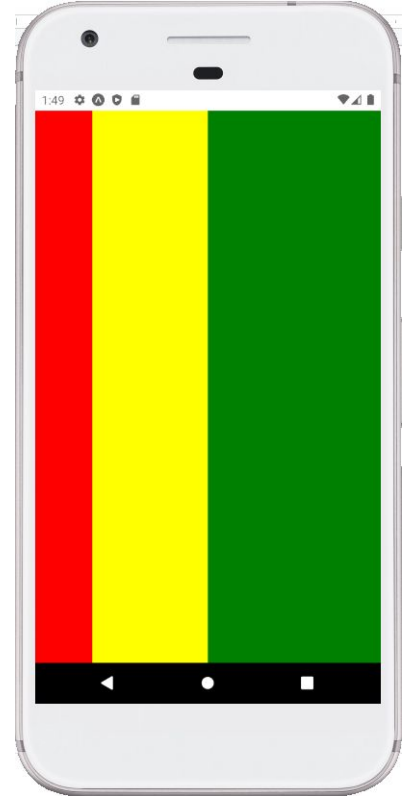
# REACT NATIVE FUNDAMENTALS

### FLEXBOX IN REACT NATIVE

Update stylesheet kita menjadi seperti ini :

```
container: {  
  flex: 1,  
  flexDirection: 'row',  
  paddingTop: 25  
},  
box1: {  
  flex: 1,  
  backgroundColor: 'red'  
},  
box2: {  
  flex: 2,  
  backgroundColor: 'yellow'  
},  
box3: {  
  flex: 3,  
  backgroundColor: 'green'  
}
```

Berikan sedikit `paddingTop` pada container kita, maka kita akan mendapatkan area box di bawah status bar =>



HACKTIV8

## React Native : Styling and Layout - 15

# REACT NATIVE FUNDAMENTALS

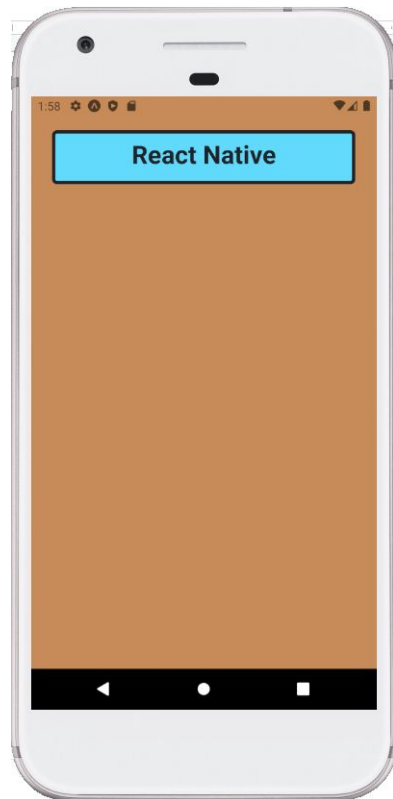
### STYLING BASIC

Adakah yang bisa kita tweak dengan basic styling yang disediakan oleh React Native ? Tentu saja ada. Check out <https://reactnative.dev/docs/stylesheet> untuk informasi lebih lanjut

Update function return dan tambahkan stylesheet kita menjadi seperti ini :

```
<View style={styles.container}>
  <Text style={styles.title}>React Native</Text>
  <StatusBar style="auto" />
</View>
```

```
container: {
  flex: 1,
  backgroundColor: "#C68B59",
  padding: 25
},
title: {
  marginTop: 16,
  paddingVertical: 8,
  borderWidth: 4,
  borderColor: "#20232a",
  borderRadius: 6,
  backgroundColor: "#61dafb",
  color: "#20232a",
  textAlign: "center",
  fontSize: 30,
  fontWeight: "bold"
}
```



HACKTIV8

## React Native : Styling and Layout - 15

# REACT NATIVE FUNDAMENTALS

### HANDLING TEXT INPUT

Cara kita handle text input sama persis seperti apa yang kita lakukan dengan React sebelumnya. Handling ini melibatkan state dan event handler

#### 1. Import useState dan implementasi di component App

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

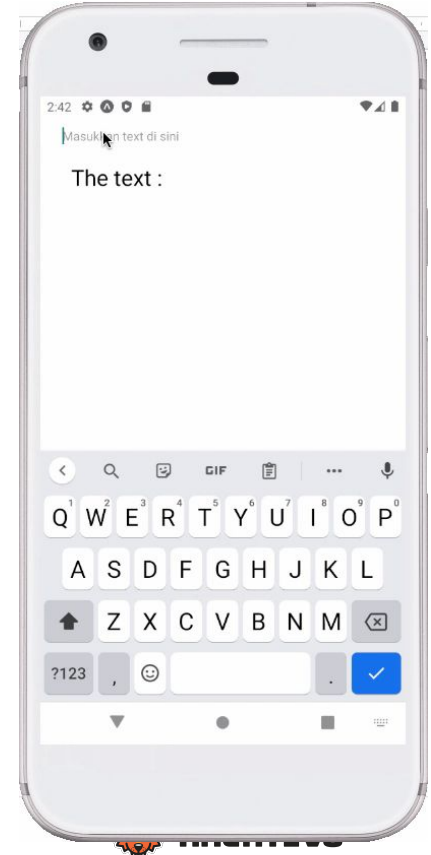
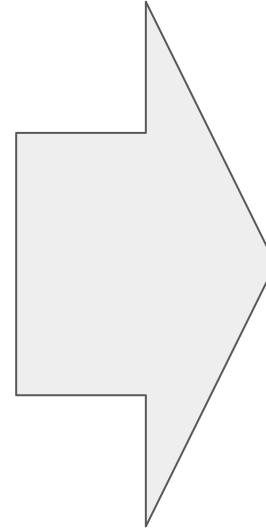
export default function App() {
  const [input, setInput] = useState('')
```

#### 2. Update function return dan stylesheet kita menjadi seperti ini :

```
<View style={styles.container}>
  <TextInput
    style={styles.textInput}
    placeholder="Masukkan text di sini"
    onChangeText={text => setInput(text)}
    defaultValue={input}
  />
  <Text style={{padding: 10, fontSize: 25}}>
    The text : {input}
  </Text>
  <StatusBar style="auto" />
</View>
```

```
container: {
  flex: 1,
  padding: 25
},
textInput: {
  height: 40
}
```

Untuk melihat simulasi lebih lengkap, silakan buka Lampiran 1



## React Native : Styling and Layout - 15

# REACT NATIVE FUNDAMENTALS

### SCROLLVIEW

Bagaimana untuk handle tampilan yang bisa jadi melebihi tinggi layar kita ? Misal : menampilkan banyak text box ?  
Gunakanlah ScrollView

#### 1. Import ScrollView

```
import { ScrollView } from 'react-native';
```

#### 2. Buatlah simulasi data di dalam App

```
let posts = []

for (let i = 1; i <= 10; i++) {
  posts.push({
    title: `Title ${i}`,
    description: `Deskripsi untuk title ${i}`
  })
}
```

#### 3. Update function return kita

```
<View style={styles.container}>
  <ScrollView style={styles.scroll}>
    {
      posts.map((post, idx) => (
        <View style={styles.box} key={idx}>
          <Text style={styles.title}>{post.title}</Text>
          <Text>{post.description}</Text>
        </View>
      ))
    }
  </ScrollView>
  <StatusBar style="auto" />
</View>
```

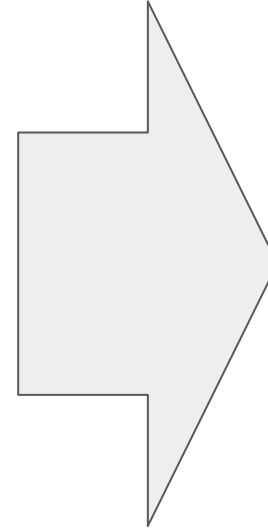


## React Native : Styling and Layout - 15

# REACT NATIVE FUNDAMENTALS

### 4. Update stylesheet kita

```
container: {
  flex: 1,
  marginTop: 25
},
scroll: {
  backgroundColor: '#C2B8A3',
  padding: 10
},
box: {
  backgroundColor: '#FEF7DC',
  marginBottom: 10,
  padding: 10
},
title: {
  fontSize: 20,
  fontWeight: 'bold'
}
```



Untuk melihat simulasi lebih lengkap,  
silakan buka Lampiran 2

### SCROLLVIEW



## React Native : Styling and Layout - 15

# REACT NATIVE FUNDAMENTALS

### LISTVIEW - FLATLIST

Dan kita pun bisa menampilkan list yang terlihat flat, terutama untuk menampilkan daftar dengan hanya nama atau title nya saja, dengan menggunakan FlatList

#### 1. Import FlatList

```
import { FlatList } from 'react-native';
```

#### 2. Buatlah simulasi data di dalam App

```
let posts = []

for (let i = 1; i <= 20; i++) {
  posts.push({
    title: `Title ${i}`
  })
}
```

#### 3. Update function return kita

```
<View style={styles.container}>
  <FlatList
    data={posts}
    renderItem={({item}) => <Text style={styles.item}>{item.title}</Text>}
  />
  <StatusBar style="auto" />
</View>
```

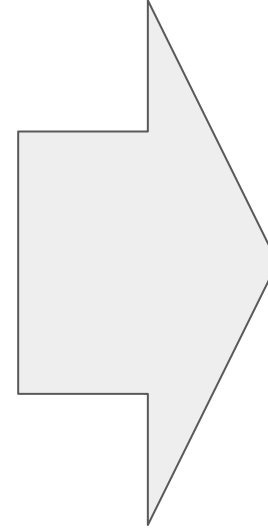


## React Native : Styling and Layout - 15

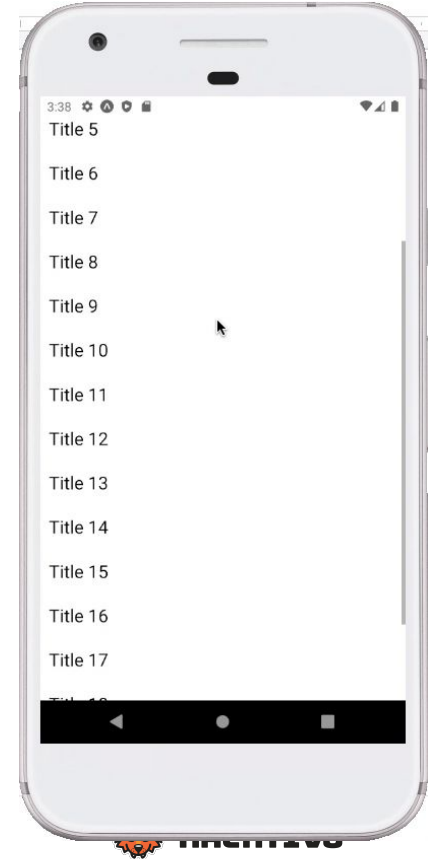
# REACT NATIVE FUNDAMENTALS

### 4. Update stylesheet kita

```
container: {  
  flex: 1,  
  marginTop: 25  
},  
item: {  
  padding: 10,  
  fontSize: 20,  
  height: 50  
}
```



### LISTVIEW - FLATLIST



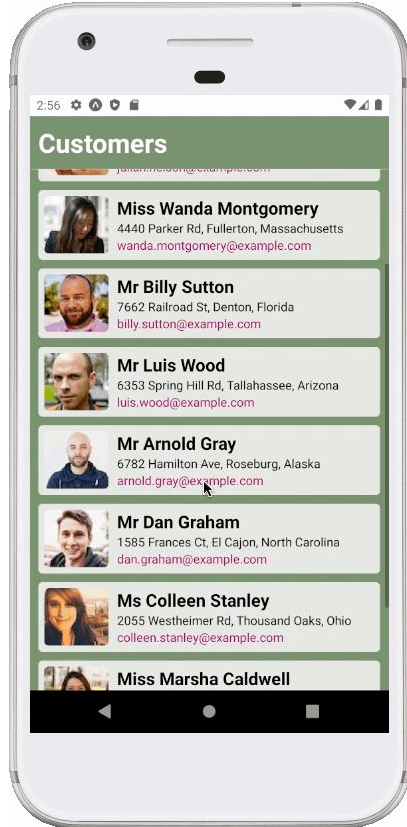
Untuk melihat simulasi lebih lengkap, silakan buka Lampiran 3



## React Native : Styling and Layout - 15

# REACT NATIVE FUNDAMENTALS

### LISTVIEW - FLATLIST



Pelajari API randomuser dari <https://randomuser.me>, kemudian manfaatkan untuk mendapatkan list user dengan atribut :

- nama lengkap
- alamat
- email
- foto

Dengan menggunakan fundamental yang telah kita pelajari, buatlah tampilan seperti di samping ini. Petunjuk :

- di sini list users di map menjadi list customers
- di dalam View, bisa ada View lagi, mainkan flexbox nya...

Design, warna, bentuk, hanyalah contoh, tidak mengikat. Silakan berkreasi

**HAPPY CODING !!!**

**Untuk melihat simulasi lebih lengkap, silakan buka Lampiran 4**



# React Native<sup>+</sup> Debugging Tools

Untuk React Native kita mempunyai beberapa opsi untuk *debugging*. Untuk mengakses *debugging tools* kita bisa menekan tombol command + d untuk simulator iOS dan command + m untuk emulator Android. Kita akan bahas opsinya satu per satu.

1. Reload. Digunakan untuk me-reload ulang aplikasi kita.
2. Debug JS Remotely akan membuka chrome browser dan kita dapat melakukan debug seperti `console.log` atau `debugger` untuk setting breakpoint.
3. Enable Live Reload secara otomatis me-reload ulang aplikasi kita ketika ada perubahan file.
4. Enable Hot Reloading jika opsi sebelumnya me-reload seluruh aplikasi kita, hot reload akan me-reload sebagian aplikasi kita dengan tetap mempertahankan posisi atau *current state* dan hanya me-reload bagian yang berubah saja.
5. Toggle Inspector digunakan untuk mencari komponen-komponen seperti inspect element jika di browser.
6. Show Perf Monitor dapat digunakan untuk memonitor performance dari aplikasi kita.

Selain itu kita juga bisa menggunakan `console.warn` untuk menampilkan warning di hp atau emulator.



# Thank You

---

**PT Hacktivate Teknologi Indonesia**

Gedung Aquarius Pondok Indah  
Jalan Sultan Iskandar Muda No.7  
Kebayoran Lama, Jakarta Selatan

*[www.hacktiv8.com](http://www.hacktiv8.com)*

