

Front-End Development



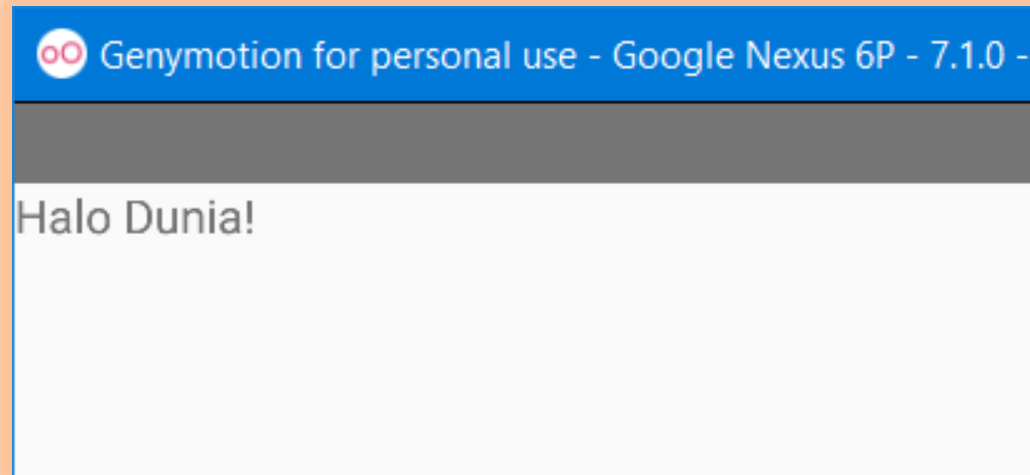
React Native

#2 Fundamental

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

class App extends Component {
  render() {
    return (
      <Text>
        Halo Dunia!
      </Text>
    );
  }
}

export default App;
```



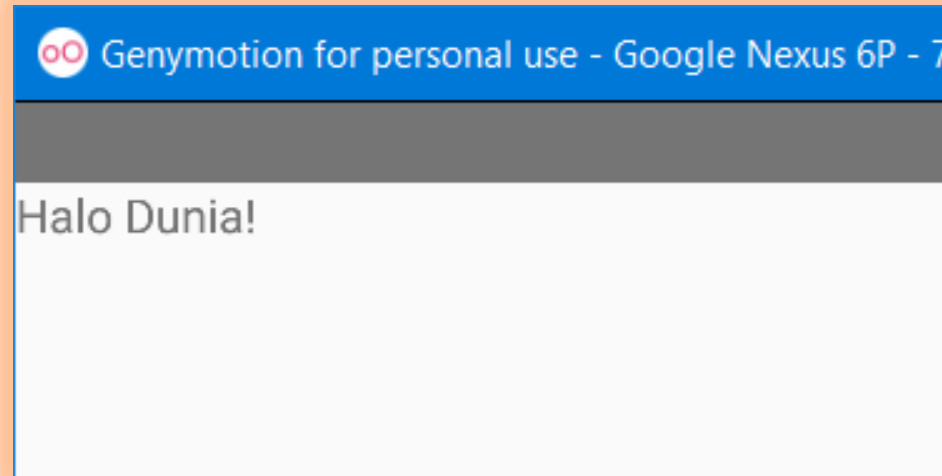
App.js

Hello World!
Insert var

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

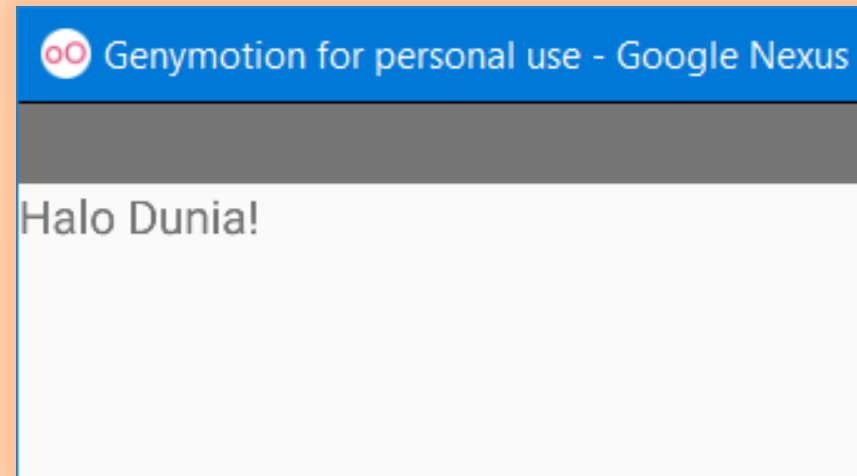
var kalimat = 'Halo Dunia!';
class App extends Component {
  render() {
    return (
      <View>
        <Text>
          { kalimat }
        </Text>
      </View>
    );
  }
}

export default App;
```



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

class App extends Component {
  constructor(){
    super();
    this.kata = 'Dunia';
  }
  render() {
    return (
      <View>
        <Text>
          { this.kata }
        </Text>
      </View>
    );
  }
}
export default App;
```

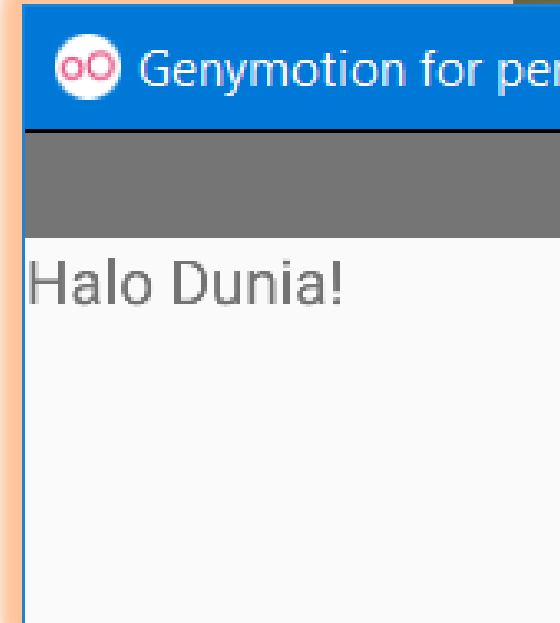


App.js

Hello World! *Insert func statement*

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

class App extends Component {
  kata(x){return x;}
  render() {
    return (
      <View>
        <Text>
          Halo {this.kata('Dunia')}
        </Text>
      </View>
    );
  }
}
export default App;
```

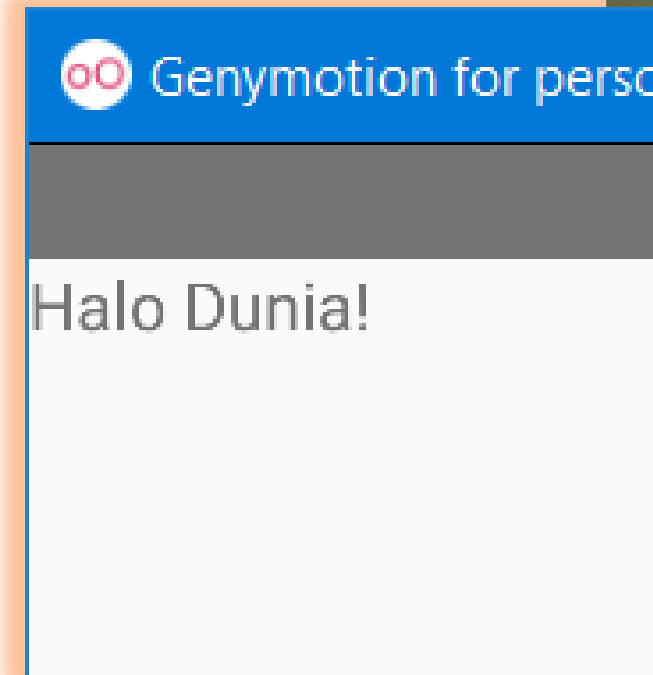


App.js

Hello World! *Insert func expression*

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

class App extends Component {
  render() {
    function kata(x){return x;}
    return (
      <View>
        <Text>
          Halo {kata('Dunia')}
        </Text>
      </View>
    );
  }
}
export default App;
```

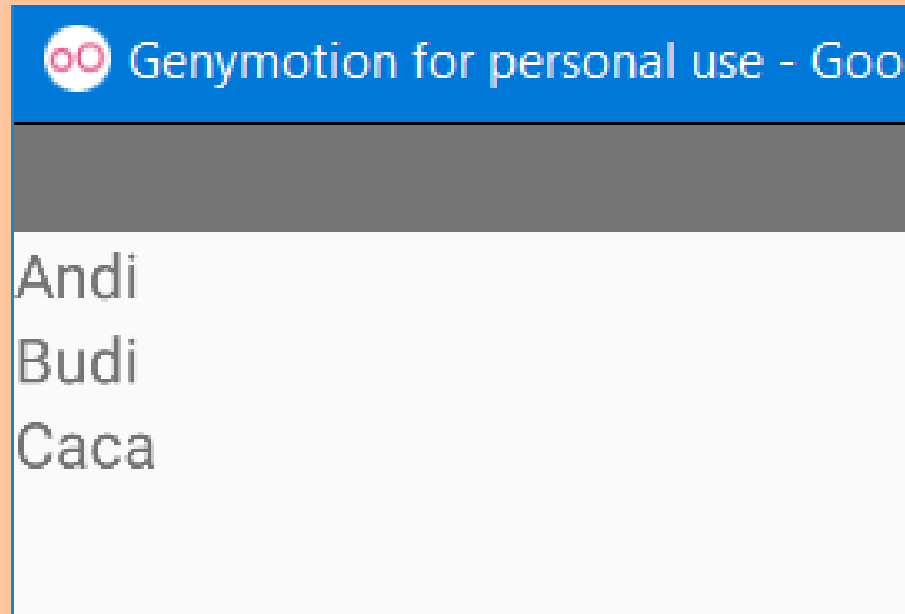


App.js

Rendering Multiple Elements

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

class App extends Component {
  render() {
    const siswa = ['Andi', 'Budi', 'Caca'];
    const listSiswa = siswa.map((siswa, index)=>
      <Text key={index}> {siswa} </Text>
    );
    return (
      <View>
        { listSiswa }
      </View>
    );
  }
}
export default App;
```



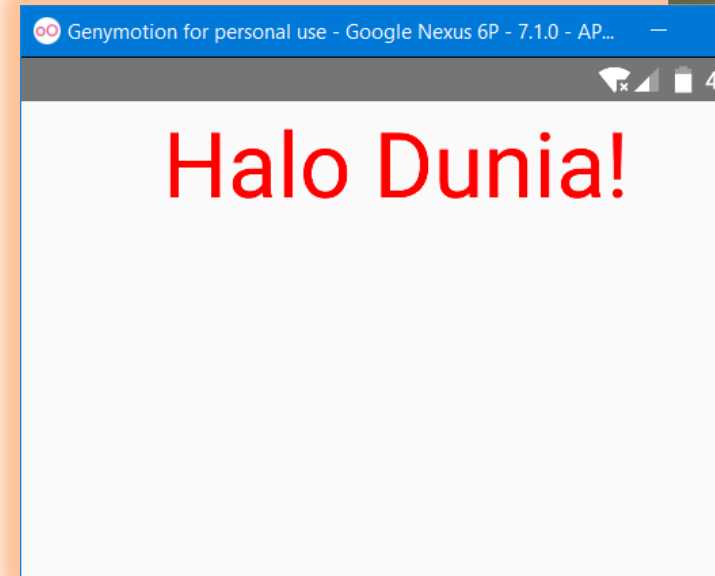
App.js

Styling

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

```
class App extends Component {  
  render() {  
    return (  
      <Text style={{  
        color: 'red',  
        fontSize: 50,  
        textAlign: 'center' }}>  
        Halo Dunia!  
      </Text>  
    );  
  }  
}
```

```
export default App;
```

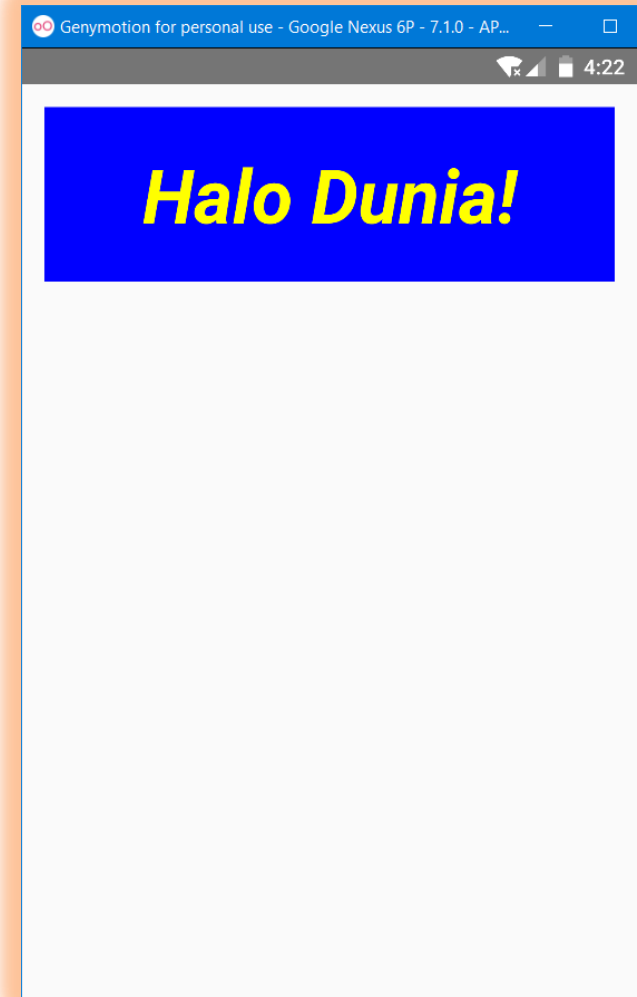


• • • • •

App.js

```
<View>
  <Text style={{
    color: 'yellow',
    backgroundColor: 'blue',
    fontSize: 50,
    fontStyle: 'italic',
    fontWeight: 'bold',
    textAlign: 'center',
    padding: 25,
    margin: 15
  }}>
    Halo Dunia!
  </Text>
</View>
```

• • • • •



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

```
class App extends Component {
  render() {
    return (
      <View>
        <Text style={gaya.judul}>
          Ini Gayaku!
        </Text>
      </View>
    );
  }
}
```

• • • • •

• • • • •

```
const gaya = StyleSheet.create({  
  judul: {  
    color: 'blue',  
    backgroundColor: 'pink',  
    fontSize: 50,  
    fontStyle: 'italic',  
    fontWeight: 'bold',  
    textAlign: 'center',  
    padding: 25,  
    margin: 15  
  },  
});  
  
export default App;
```



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

class App extends Component {
  render() {
    return (
      <View>
        <Text style={gaya.teksA}>
          Ini Teks A
        </Text>
        <Text style={gaya.teksB}>
          Ini Teks B
        </Text>
      </View>
    );
  }
}
```

• • • • •

• • • • •

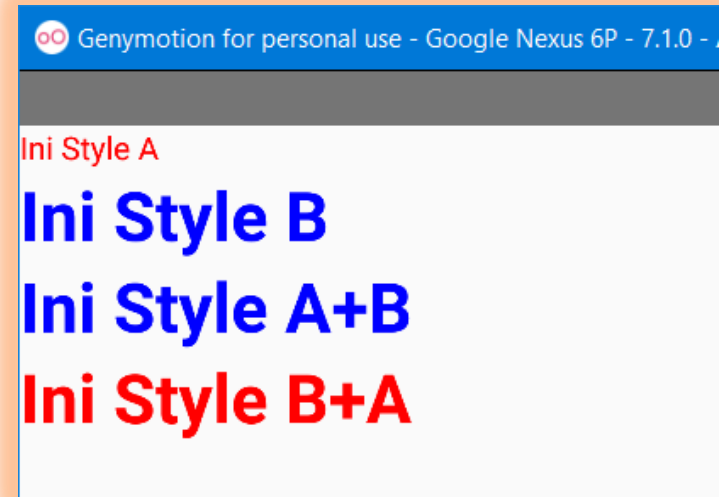
```
const gaya = StyleSheet.create({  
  teksA: {  
    color: 'red',  
  },  
  teksB: {  
    color: 'blue',  
    fontSize: 30,  
    fontWeight: 'bold',  
  },  
});  
  
export default App;
```



• • • • •

```
<View>
  <Text style={gaya.teksA}>
    Ini Style A
  </Text>
  <Text style={gaya.teksB}>
    Ini Style B
  </Text>
  <Text style={[gaya.teksA,gaya.teksB]}>
    Ini Style A+B
  </Text>
  <Text style={[gaya.teksB,gaya.teksA]}>
    Ini Style B+A
  </Text>
</View>
```

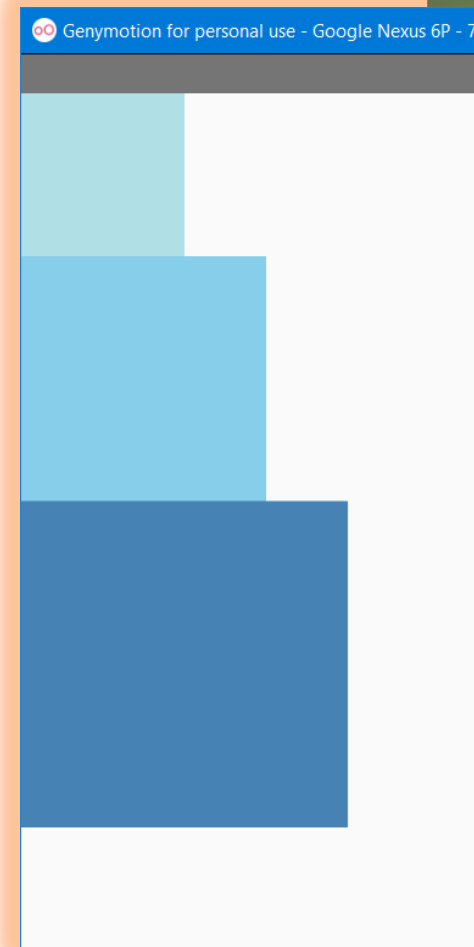
• • • • •



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

class App extends Component {
  render() {
    return (
      <View>
        <View style={{width: 100, height: 100,
          backgroundColor: 'powderblue'}} />
        <View style={{width: 150, height: 150,
          backgroundColor: 'skyblue'}} />
        <View style={{width: 200, height: 200,
          backgroundColor: 'steelblue'}} />
      </View>
    );
  }
}

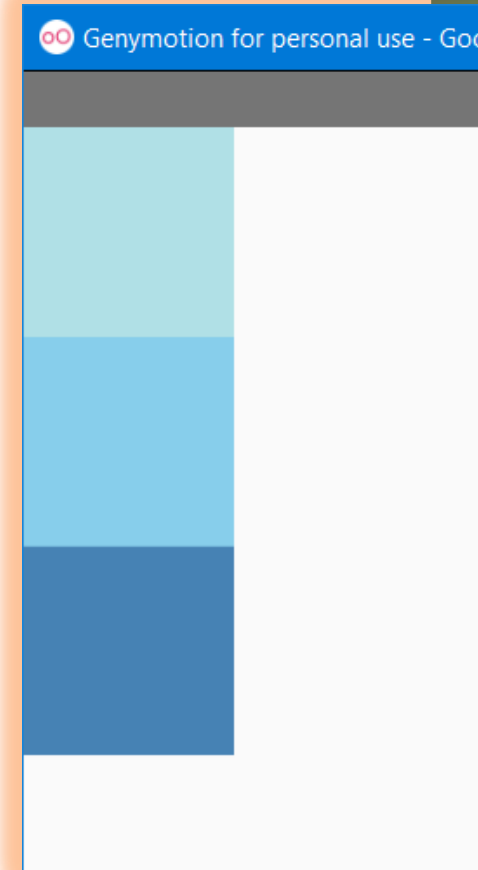
export default App;
```



• • • • •

```
<View style={{flexDirection: 'column'}}>  
  <View style={{width: 90, height: 90,  
    backgroundColor: 'powderblue'}} />  
  
  <View style={{width: 90, height: 90,  
    backgroundColor: 'skyblue'}} />  
  
  <View style={{width: 90, height: 90,  
    backgroundColor: 'steelblue'}} />  
</View>
```

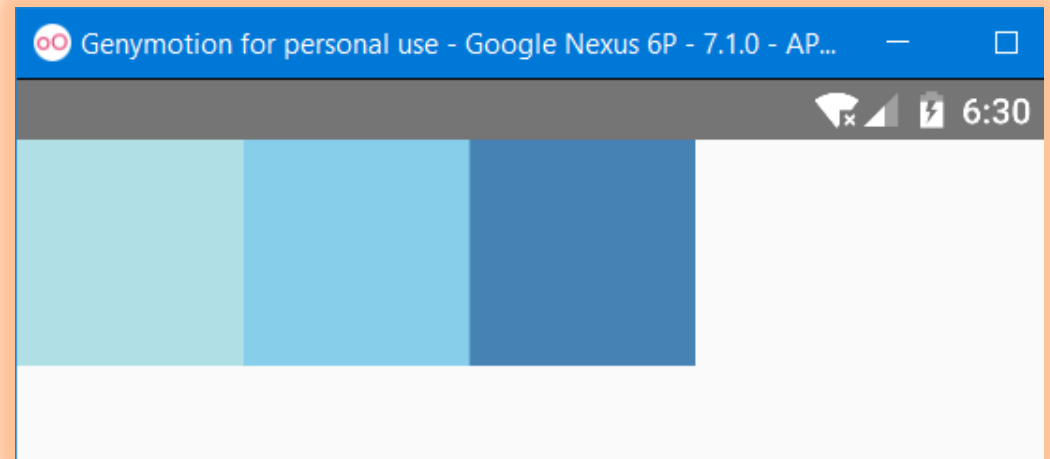
• • • • •



• • • • •

```
<View style={{flexDirection: 'row'}}>
  <View style={{width: 90, height: 90,
    backgroundColor: 'powderblue'}} />
  <View style={{width: 90, height: 90,
    backgroundColor: 'skyblue'}} />
  <View style={{width: 90, height: 90,
    backgroundColor: 'steelblue'}} />
</View>
```

• • • • •



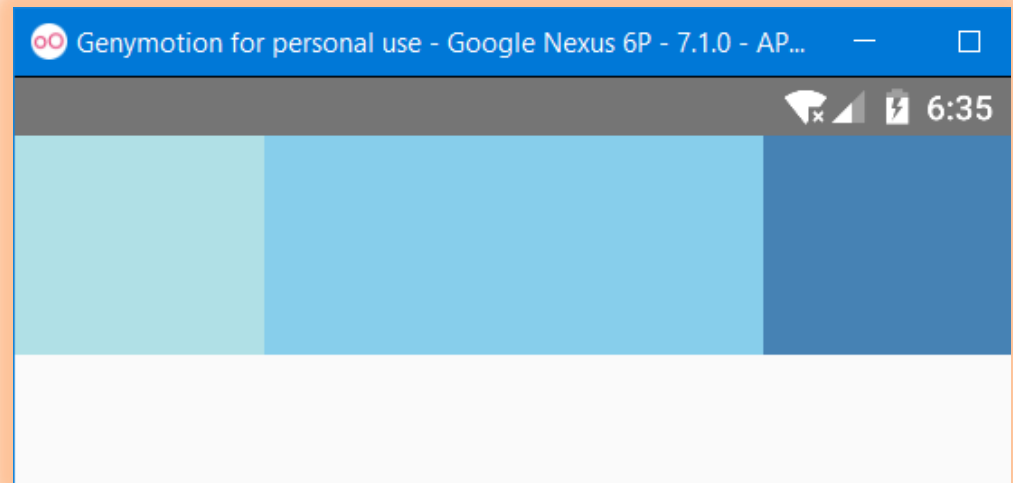
• • • • •

```
<View style={{flexDirection: 'row'}}>
  <View style={{flex:1, width: 90, height: 90,
    backgroundColor: 'powderblue'}} />

  <View style={{flex:2, width: 90, height: 90,
    backgroundColor: 'skyblue'}} />

  <View style={{flex:1, width: 90, height: 90,
    backgroundColor: 'steelblue'}} />
</View>
```

• • • • •



Justify Content #Center App.js

• • • • •

```
<View style={{flexDirection: 'row',  
justifyContent: 'center'}}>
```

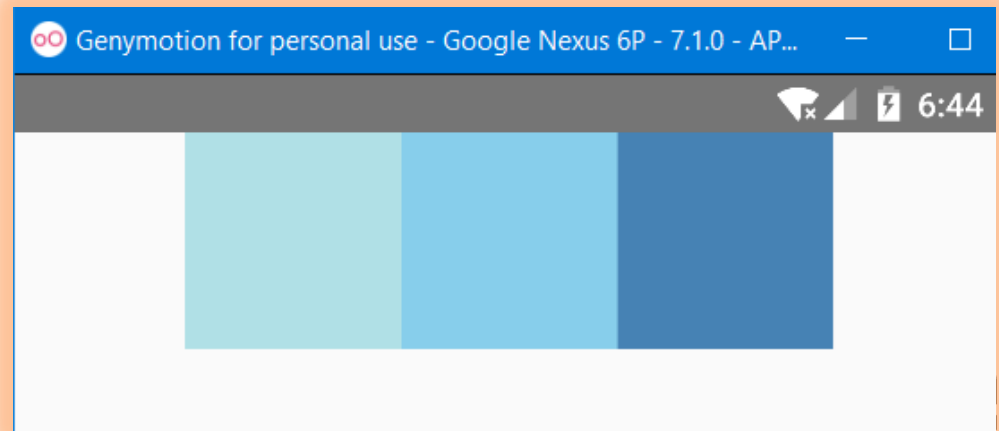
```
  <View style={{width: 90, height: 90,  
    backgroundColor: 'powderblue'}} />
```

```
  <View style={{width: 90, height: 90,  
    backgroundColor: 'skyblue'}} />
```

```
  <View style={{width: 90, height: 90,  
    backgroundColor: 'steelblue'}} />
```

```
</View>
```

• • • • •



Justify Content #Space App.js

• • • • •
`<View style={{flexDirection: 'row',
justifyContent: 'space-between'}}>`

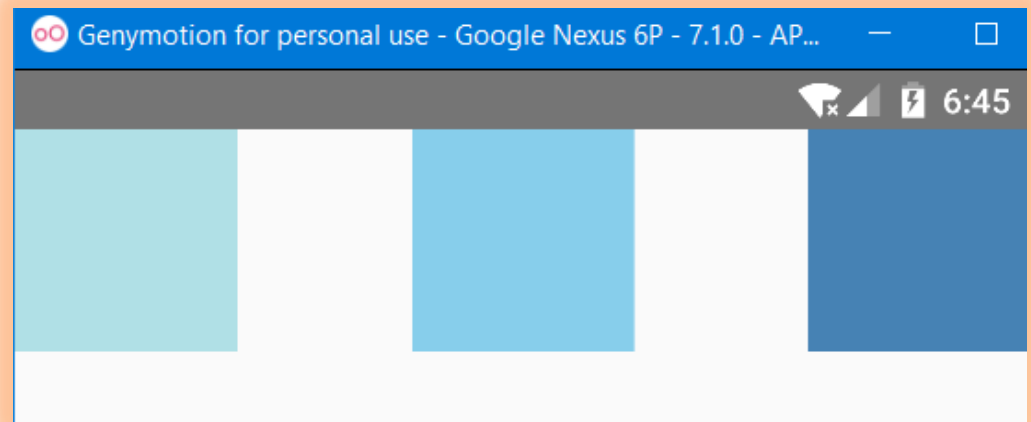
`<View style={{width: 90, height: 90,
backgroundColor: 'powderblue'}} />`

`<View style={{width: 90, height: 90,
backgroundColor: 'skyblue'}} />`

`<View style={{width: 90, height: 90,
backgroundColor: 'steelblue'}} />`

`</View>`

• • • • •



Justify Content

#Flex-End

App.js

• • • • •

```
<View style={{flexDirection: 'row',  
justifyContent: 'flex-end'}}>
```

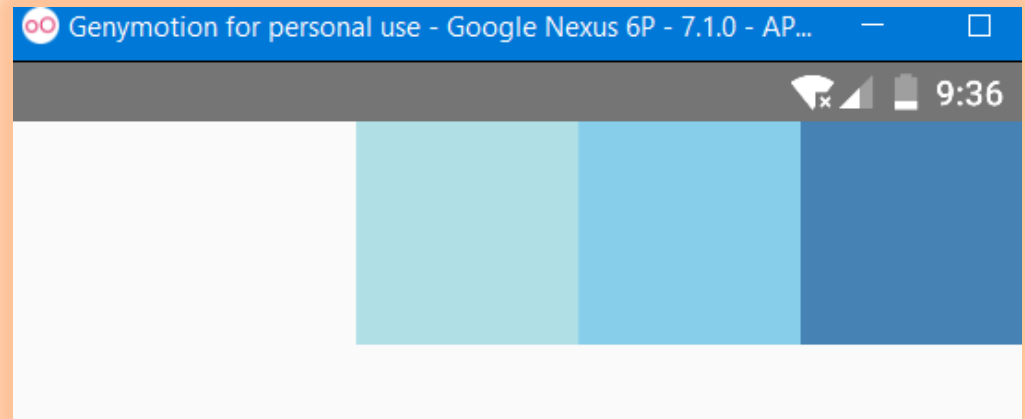
```
  <View style={{width: 90, height: 90,  
    backgroundColor: 'powderblue'}} />
```

```
  <View style={{width: 90, height: 90,  
    backgroundColor: 'skyblue'}} />
```

```
  <View style={{width: 90, height: 90,  
    backgroundColor: 'steelblue'}} />
```

```
</View>
```

• • • • •



Align Items

App.js

```
• • • • •  
<View style={{flexDirection: 'column',  
alignItems: 'center'}}>
```

```
<View style={{width: 90, height: 90,  
backgroundColor: 'powderblue'}} />
```

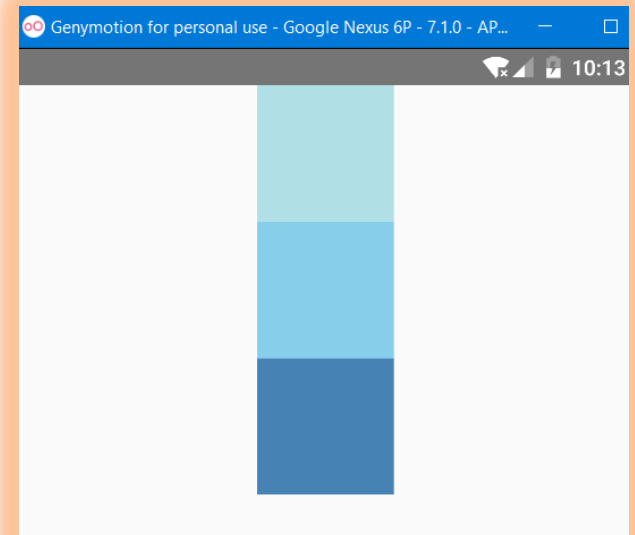
```
<View style={{width: 90, height: 90,  
backgroundColor: 'skyblue'}} />
```

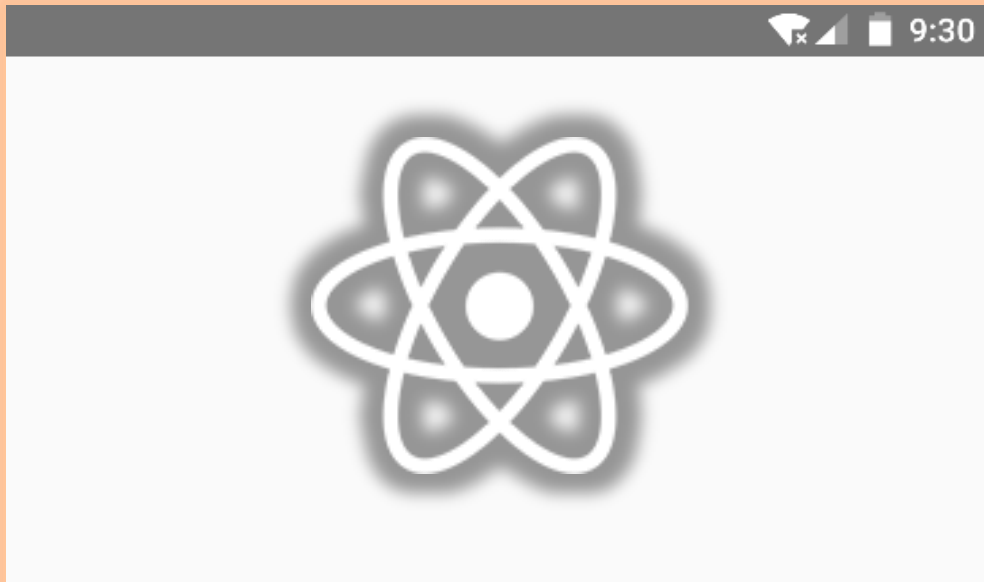
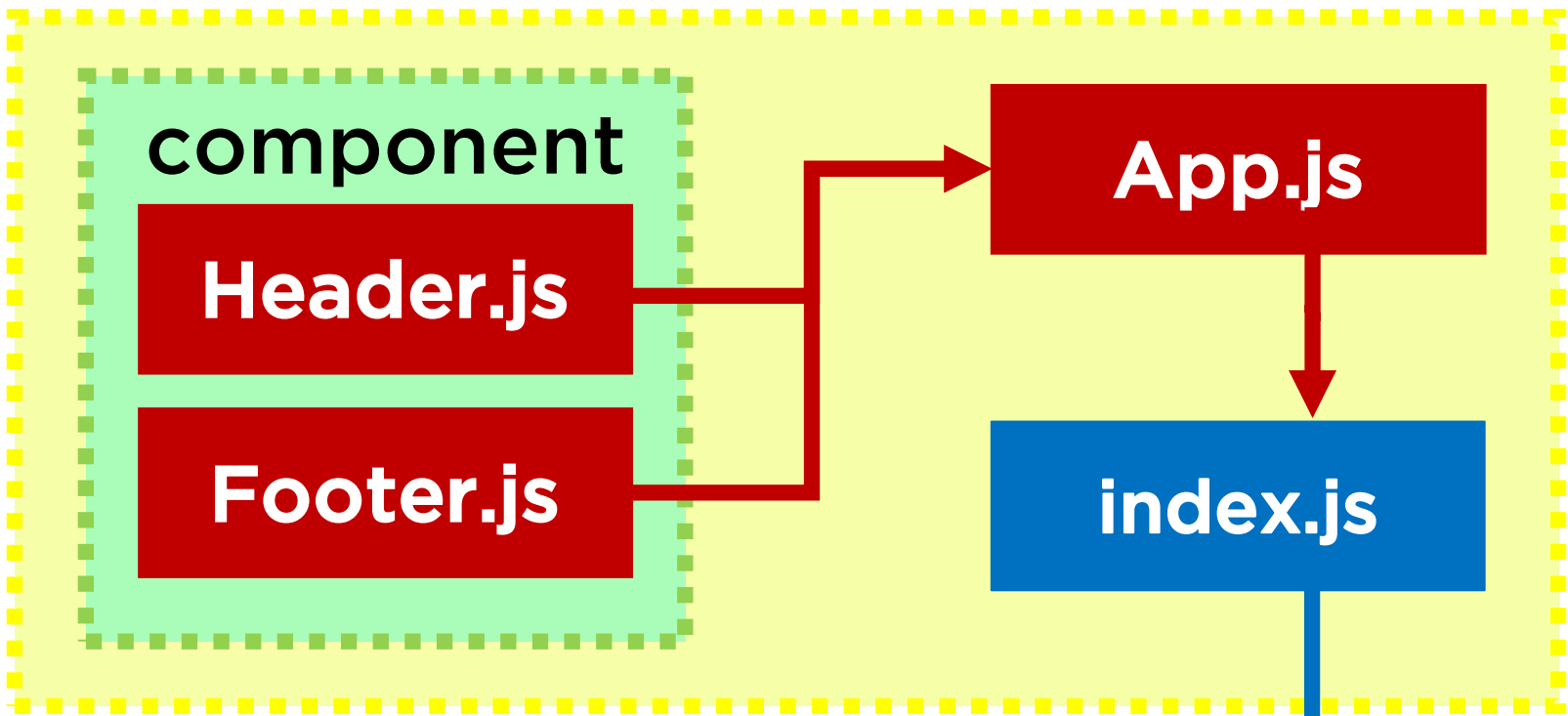
```
<View style={{width: 90, height: 90,  
backgroundColor: 'steelblue'}} />
```

```
</View>
```

```
• • • • •
```

```
// alignItems: 'flex-start'  
// alignItems: 'flex-end'
```





component/Header.js

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

```
class Header extends Component {  
  render() {  
    return (  
      <Text>  
        Ini dari Header.js  
      </Text>  
    );  
  }  
}
```

```
export default Header;
```


component/Footer.js

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

```
class Footer extends Component {  
  render() {  
    return (  
      <Text>  
        Ini dari Footer.js  
      </Text>  
    );  
  }  
}
```

```
export default Footer;
```

```
import React, { Component } from 'react';
import { View, Text } from 'react-native';
import Header from './component/Header';
import Footer from './component/Footer';
```

```
class App extends Component {
  render() {
    return (
      <View>
        <Header/>
        <Text>Ini dari App.js</Text>
        <Footer/>
      </View>
    );
  }
}
export default App;
```

Genymotion for pers

Ini dari Header.js
Ini dari App.js
Ini dari Footer.js



State & Props

- There are 2 types of data that control a component: *State* & *Props*.
- *Props* are set by the parent and they're fixed throughout the lifetime of a component. For data that is going to change, we have to use *State*.



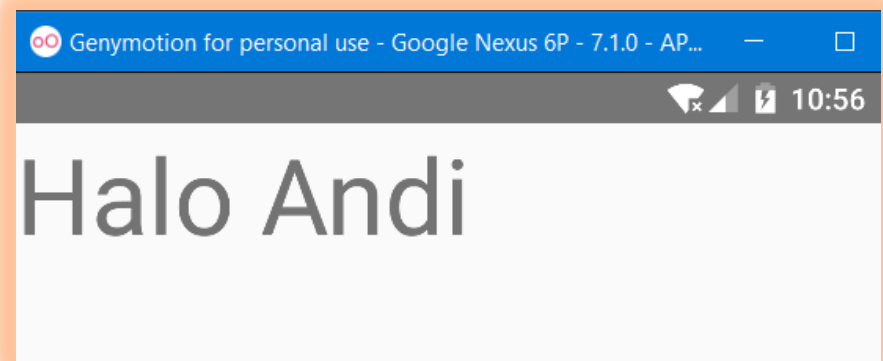
State

- In general, we should initialize state in the `constructor`, and then call `setState` when we want to change it.
- State is mutable, and defines at any given time, the current state of the React component that is being rendered.

State App.js

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

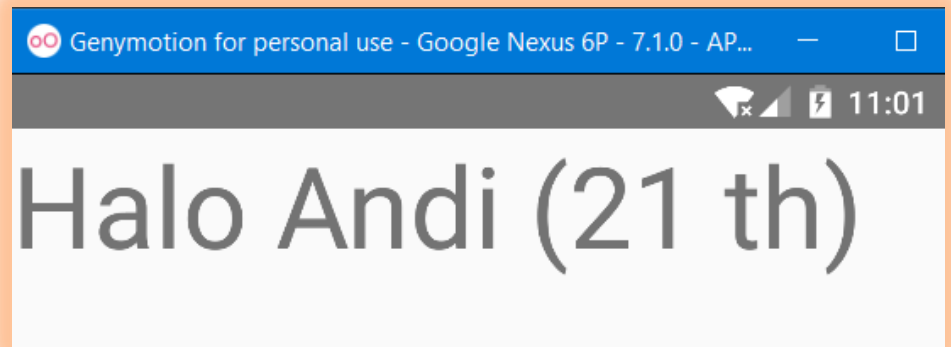
```
class App extends Component {  
  constructor(){  
    super();  
    this.state = {nama: 'Andi'}  
  }  
  render() {  
    return (  
      <Text style={{fontSize:50}}>  
        Halo { this.state.nama }  
      </Text>  
    );  
  }  
}  
  
export default App;
```



State App.js

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

```
class App extends Component {  
  constructor(){  
    super();  
    this.state = {nama:'Andi', usia:21}  
  }  
  render() {  
    return (  
      <Text style={{fontSize:50}}>  
        Halo { this.state.nama }  
        ({ this.state.usia } th)  
      </Text>  
    );  
  }  
}  
  
export default App;
```

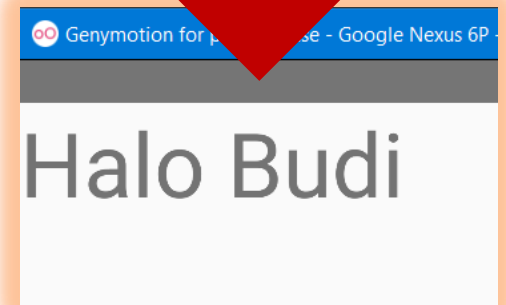
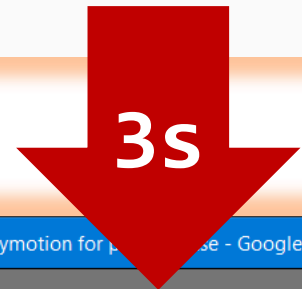
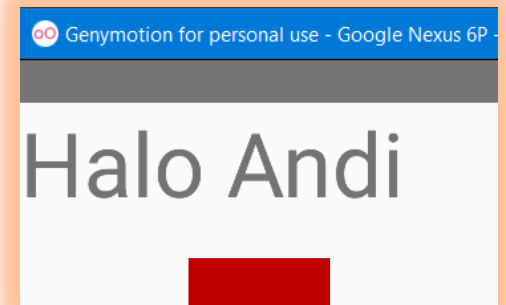


```
import React, { Component } from 'react';
import { Text } from 'react-native';
```

```
class App extends Component {
  constructor(){
    super();
    this.state = {nama: 'Andi'}
  }
  render() {
    setTimeout(() => {
      this.setState({nama: 'Budi'})
}, 3000)
    return (
      <Text style={{fontSize:50}}>
        Halo { this.state.nama }
      </Text>
    );
  }
}

export default App;
```

Updating State *App.js*





Props

- Most components can be customized with different parameters when they are created. These creation parameters are called *Props (Properties)*.
- It lets you make a component that is used in many different places in your app, with slightly different properties in each places.
- For short, props are static properties on a React component that are immutable (cannot be changed).

Props #1

App.js

```
import React, { Component } from 'react';
import { View, Text } from 'react-native';
import Header from './component/Header';
```

```
class App extends Component {
  render() {
    var teks = 'Ini Props!';
    return (
      <View>
        <Header konten={teks} />
      </View>
    );
  }
}
export default App;
```

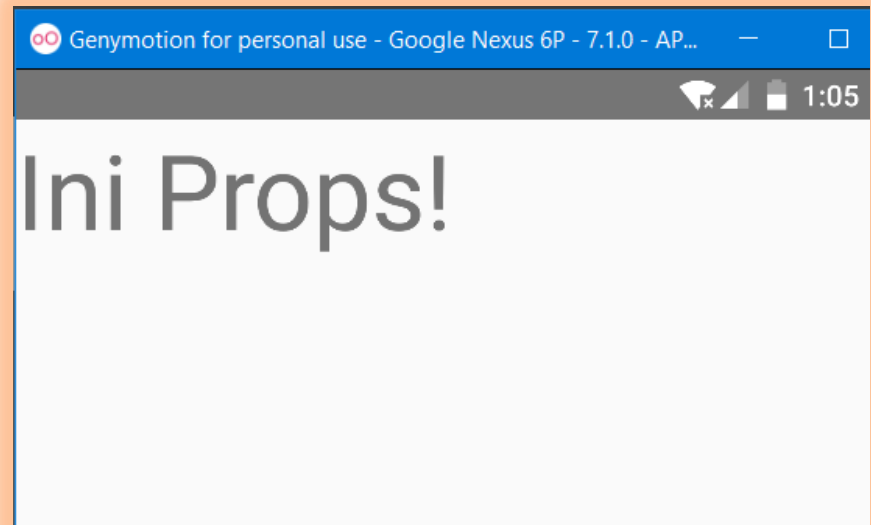
Props #1

component/Header.js

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

```
class Header extends Component {  
  render() {  
    return (  
      <Text style={{fontSize:50}}>  
        {this.props.konten}  
      </Text>  
    );  
  }  
}
```

```
export default Header;
```



Props #2

App.js

```
import React, { Component } from 'react';
import { View, Text } from 'react-native';
import Header from './component/Header';
```

```
class App extends Component {
  render() {
    var andi = {nama: 'Andi', usia: 21};
    return (
      <View>
        <Header x={andi.nama} y={andi.usia} />
      </View>
    );
  }
}
export default App;
```

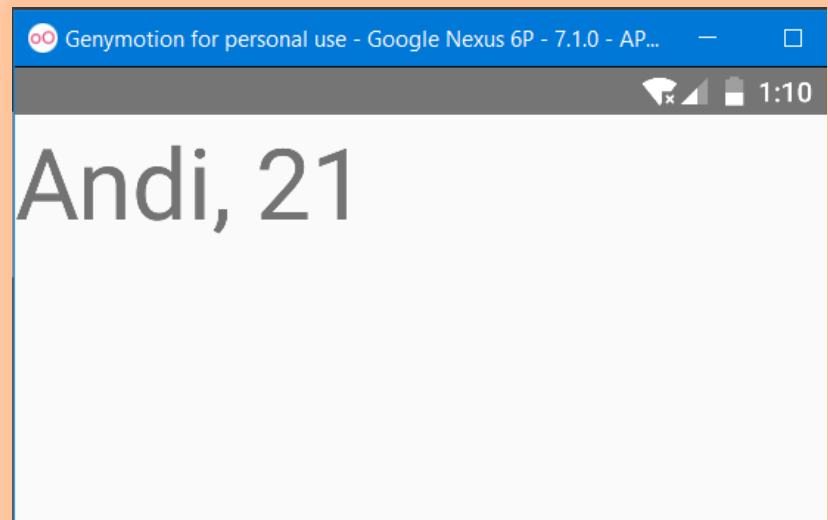
Props #2

component/Header.js

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

class Header extends Component {
  render() {
    return (
      <Text style={{fontSize:50}}>
        {this.props.x}, {this.props.y}
      </Text>
    );
  }
}

export default Header;
```



State → Props

App.js

```
import React, { Component } from 'react';
import { View, Text } from 'react-native';
import Header from './component/Header';
```

```
class App extends Component {
  constructor(){
    super();
    this.state={y: 'Ini Props dari State'}
  }
  render() {
    return (
      <View>
        <Header x={this.state.y} />
      </View>
    );
  }
}
export default App;
```

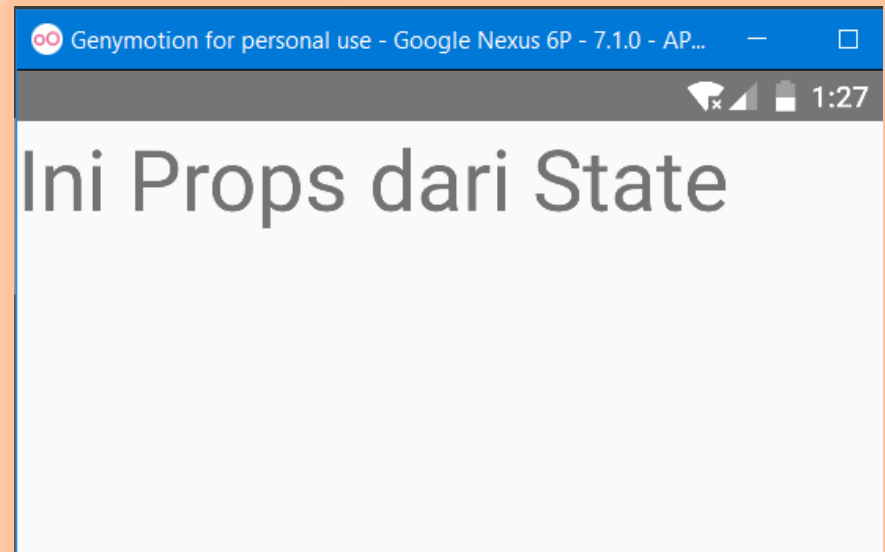
State → Props

component/Header.js

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

```
class Header extends Component {  
  render() {  
    return (  
      <Text style={{fontSize:35}}>  
        {this.props.x}  
      </Text>  
    );  
  }  
}
```

```
export default Header;
```





Life-cycle Methods

#1 Mounting

■ `componentWillMount()`

Invoked once on both client and server, immediately before the initial rendering occurs.

■ `componentDidMount()`

Invoked once only on client, immediately after the initial rendering occurs.



Life-cycle Methods

#2 Updating

■ `componentWillUpdate()`

Invoked immediately before rendering when new props or state are being received. This isn't called for the initial render.

■ `componentDidUpdate()`

Invoked immediately after the component's updates are flushed to the DOM. This isn't called for the initial render.



console.log

- To monitor React Native project through console, you can easily open a new terminal then type:

\$ react-native log-android

```
C:\WINDOWS\system32\cmd.exe - react-native log-android
04-15 10:46:15.737 2599 3727 I ReactNativeJS: Tombol diklik!
04-15 10:46:15.936 2599 3727 I ReactNativeJS: Tombol diklik!
04-15 10:46:16.103 2599 3727 I ReactNativeJS: Tombol diklik!
04-15 10:46:16.270 2599 3727 I ReactNativeJS: Tombol diklik!
04-15 10:46:16.402 2599 3727 I ReactNativeJS: Tombol diklik!
04-15 10:46:16.603 2599 3727 I ReactNativeJS: Tombol diklik!
04-15 10:46:16.770 2599 3727 I ReactNativeJS: Tombol diklik!
04-15 10:46:16.937 2599 3727 I ReactNativeJS: Tombol diklik!
04-15 10:46:17.103 2599 3727 I ReactNativeJS: Tombol diklik!
04-15 10:46:17.287 2599 3727 I ReactNativeJS: Tombol diklik!
04-15 10:46:17.436 2599 3727 I ReactNativeJS: Tombol diklik!
04-15 10:46:17.737 2599 3727 I ReactNativeJS: Tombol diklik!
```

Front-End Development



React Native

#2 Fundamental