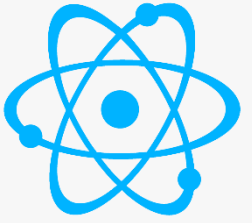


# Workshop React Native

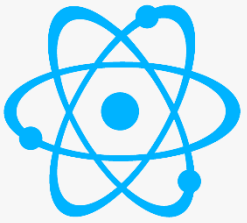
“Membangun Aplikasi Mobile Android / iOS  
dengan Framework React Native”



# Kebutuhan Sistem

## Laptop / Komputer

- Processor Core-i3/i5/i7
- Memori RAM 8GB
- Windows: Android Studio, SDK, AVD (Emulator)
- Mac OSX: XCode, iOS Emulator
- Node.js terbaru (NPM atau Yarn)
- Visual Studio Code sebagai text editor
- Koneksi Internet yang stabil



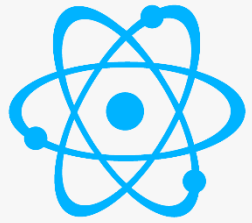
# Agenda

- Instalasi *React App Creator*
- Mengunduh dan Mengatur Ekstensi *Visual Studio Code*
- Membuat Aplikasi dan Menjalankannya di *Emulator* atau device
- Mengidentifikasi dan Mendeteksi Kesalahan di *React Native*
- Membedakan *State* dan *Props*
- Menggunakan *React Navigation*
- Studi Kasus (Menampilkan data dari Web API)

<https://github.com/openfootball/world-cup.json>


# Visual Studio Code

<https://code.visualstudio.com/Download>




Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.



↓ Windows  
Windows 7, 8, 10

User Installer	64 bit	32 bit
System Installer	64 bit	32 bit
.zip	64 bit	32 bit




↓ .deb  
Debian, Ubuntu

↓ .rpm  
Red Hat, Fedora, SUSE

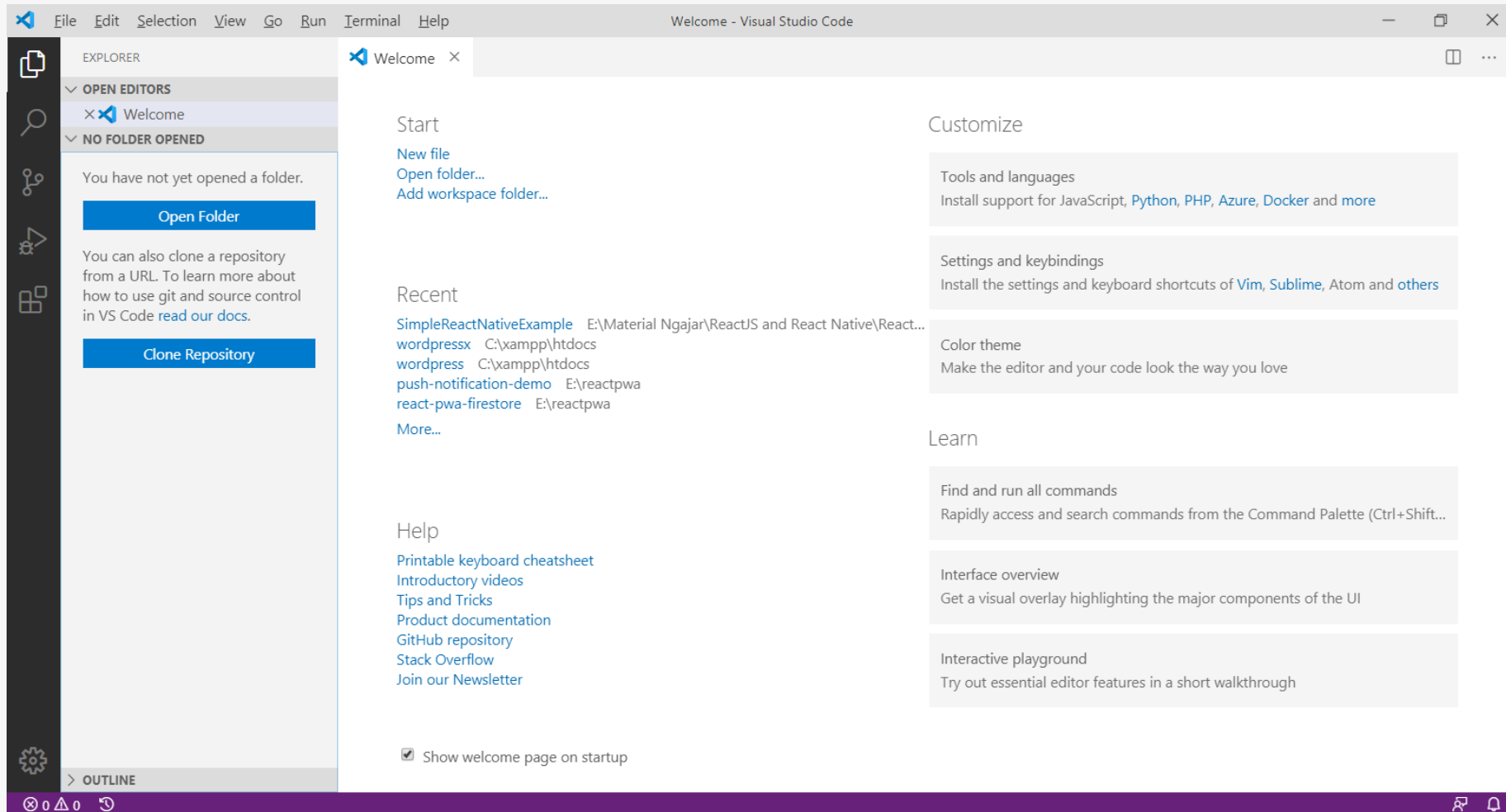
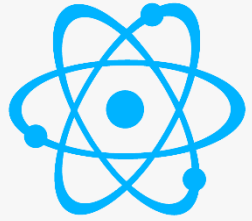
.deb	64 bit
.rpm	64 bit
.tar.gz	64 bit

Snap Store

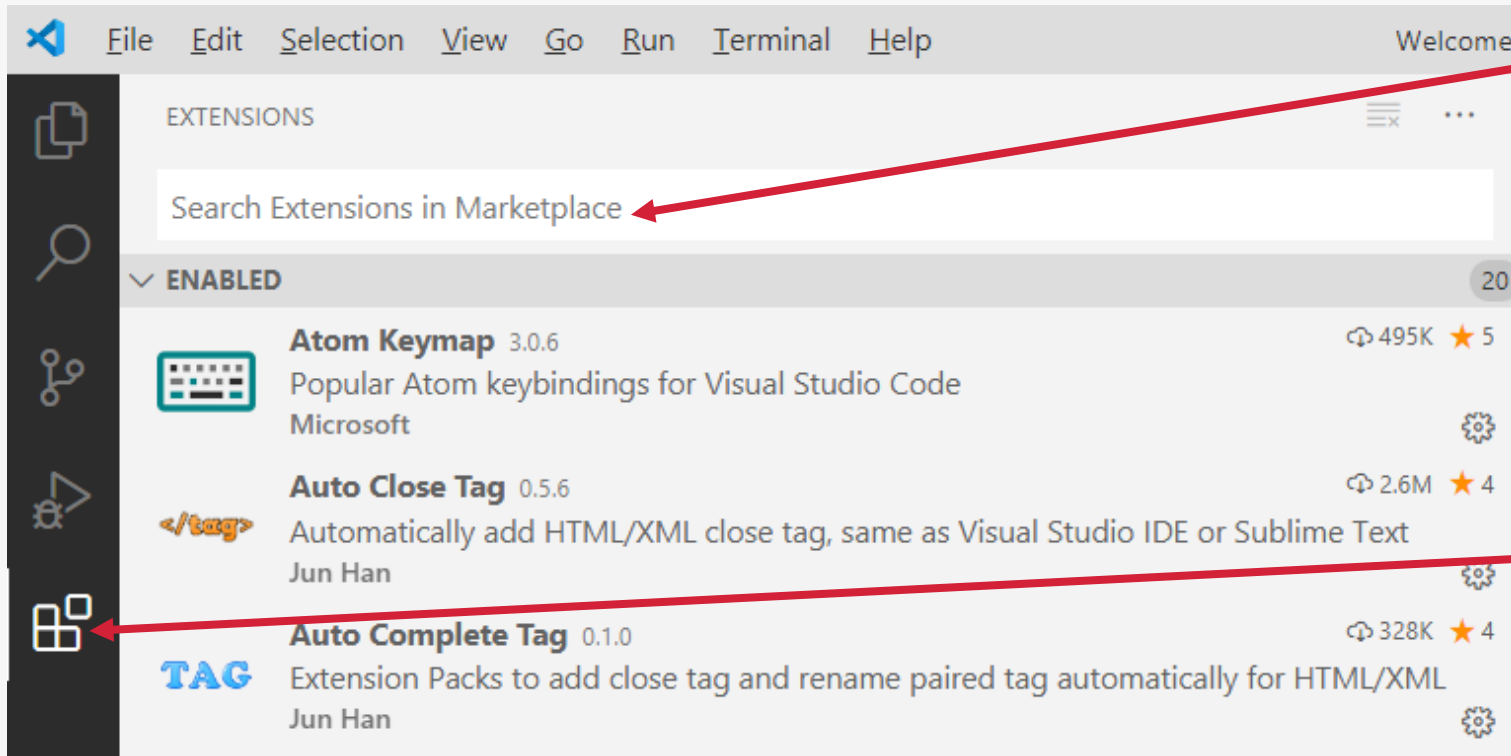
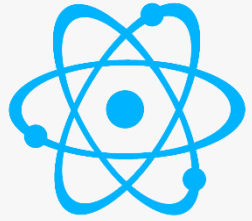


↓ Mac  
macOS 10.10+

# Visual Studio Code (2)



# Mengatur Ekstensi Visual Studio Code

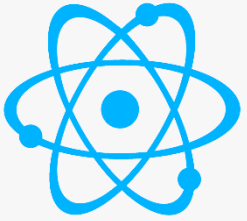


2

Cari ekstensi dengan mengetikkan nama ekstensi pada bagian ini

1

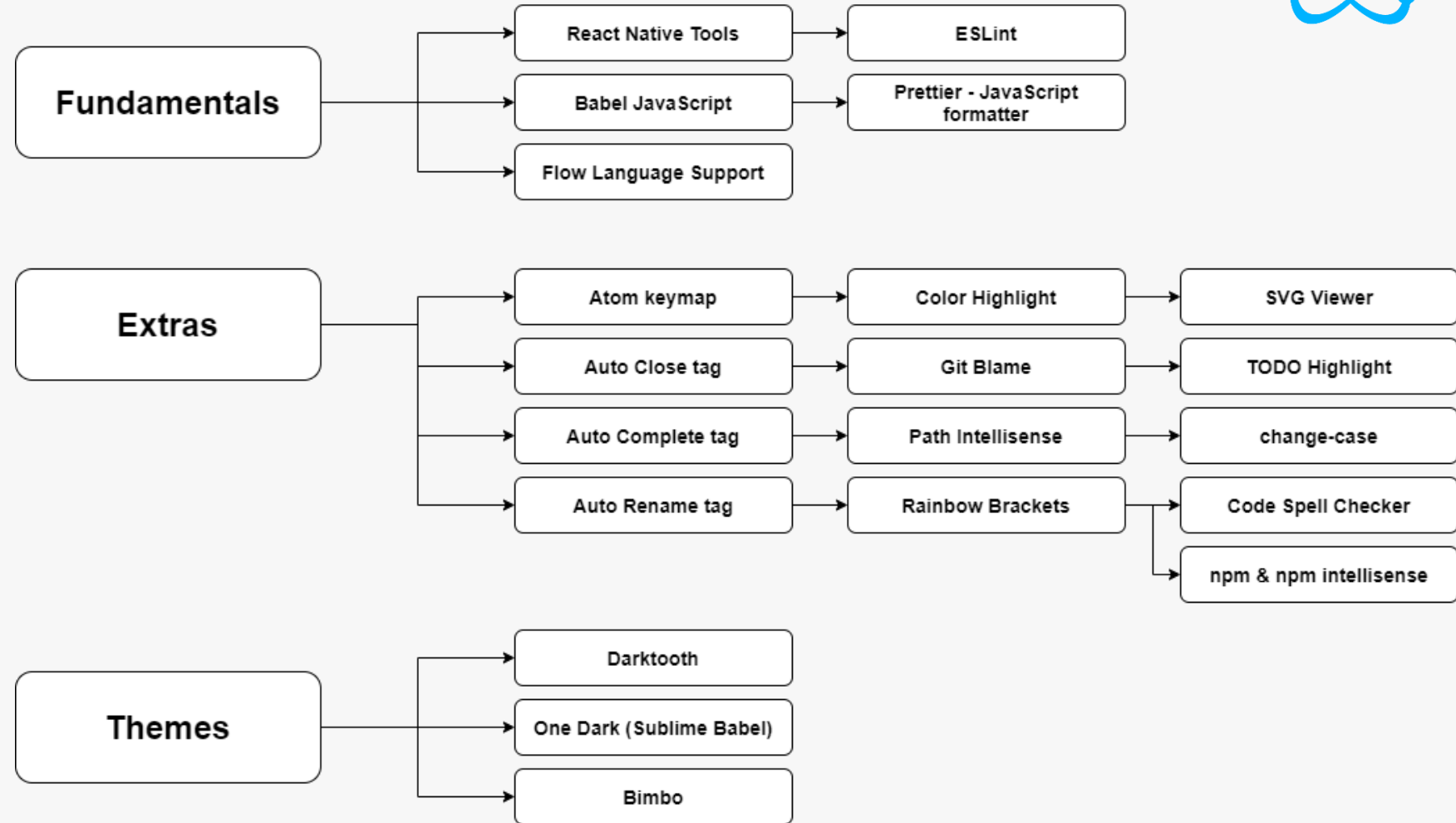
Klik pada menu Extension (Ctrl + Shift + X) untuk membuka dan mengaktifkan ekstensi

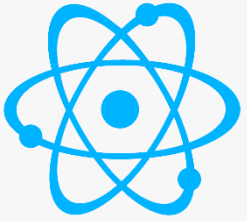


# Ekstensi yang diaktifkan

Cari dan unduh semua ekstensi seperti yang tertera pada bagan disamping ini:

**Fundamentals** harus diunduh, **Extras** dan **Themes** bersifat tidak harus diunduh hanya sebagai tambahan.





# Membuat Aplikasi Pertama

Sebelum membuat aplikasi pertama, pastikan sudah mengunduh dan menginstal **Node.js** dan bisa menjalankan **npm / yarn** melalui 'command line' atau 'terminal'


<https://nodejs.org/en/download>


### Downloads


Latest LTS Version: **12.16.1** (includes npm 6.13.4)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS  
Recommended For Most Users

  
Windows Installer  
node-v12.16.1-x86.msi

  
macOS Installer  
node-v12.16.1.pkg

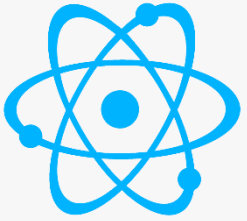
  
Source Code  
node-v12.16.1.tar.gz

Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit	
macOS Binary (.tar.gz)	64-bit	
Linux Binaries (x64)	64-bit	
Linux Binaries (ARM)	ARMv7	ARMv8
Source Code	node-v12.16.1.tar.gz	

Continuous Learning Keep Up to Date

[www.inixindosurabaya.id](http://www.inixindosurabaya.id)





# Cek versi Node, NPM, Yarn

```
Node.js command prompt
Your environment has been set up for using Node.js 13.5.0 (x64) and npm.
C:\Users\Yusuf Rizal>node -v
v13.5.0
C:\Users\Yusuf Rizal>npm -v
6.14.2
C:\Users\Yusuf Rizal>yarn -v
1.22.4
C:\Users\Yusuf Rizal>_
```

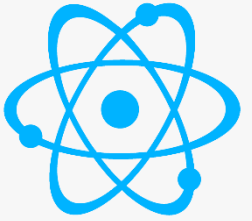
Versi Node.js

Versi NPM

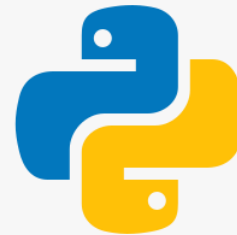
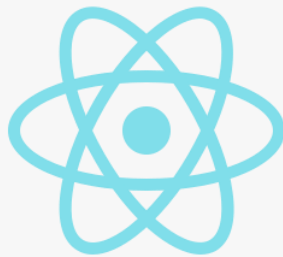
Versi Yarn

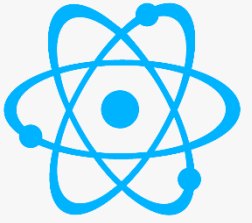
Diatas merupakan versi NPM atau Yarn, pada workshop ini kita akan menggunakan Yarn secara umum.

# Instalasi Kebutuhan Sistem



Kita memerlukan Node, React Native CLI, Python2, JDK dan Android Studio (SDK, AVD)

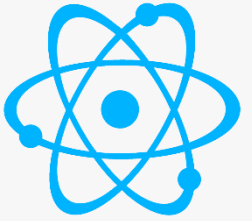




## Instalasi Kebutuhan Sistem (2)

Kami merekomendasikan untuk menginstal kebutuhan sistem melalui Chocolatey, sebuah package manager yang populer digunakan di Windows.



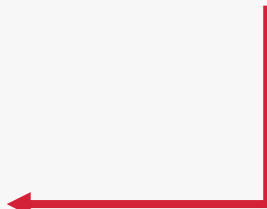


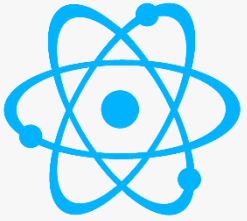
# Instalasi Chocolatey

- 1) Pertama pastikan Windows Power Shell dalam posisi 'Run as Administrator'
- 2) Copy dan Paste perintah dibawah kedalam Power Shell

```
Set-ExecutionPolicy Bypass -Scope Process -Force;  
[System.Net.ServicePointManager]::SecurityProtocol =  
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-  
Object  
System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
```

Buka <https://chocolatey.org/install>





# Instalasi Chocolatey (2)

A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The window has a blue background and a white border. The text inside shows the PowerShell prompt and a command to set execution policy and download the Chocolatey installer script. A red box highlights the command line. Two red arrows point from the box to the text labels below: one from the title bar area to "Powershell posisi Administrator" and one from the command line to "Copy dan Paste perintah di Powershell".

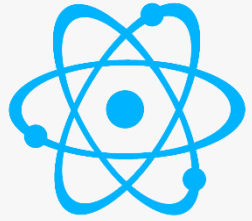
```
Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
```

Powershell posisi Administrator

Copy dan Paste perintah di Powershell

# Instalasi Chocolatey (3)



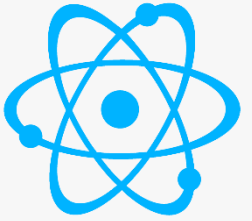
```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
Getting latest version of the Chocolatey package for download.
Getting Chocolatey from https://chocolatey.org/api/v2/package/chocolatey/0.10.15.
Extracting C:\Users\YUSUF~1\AppData\Local\Temp\chocolatey\chocInstall\chocolatey.zip to C:\Users\YUSUF~1\AppData\Local\Temp\chocolatey\chocInstall...
Installing chocolatey on this machine
Creating ChocolateyInstall as an environment variable (targeting 'Machine')
  Setting ChocolateyInstall to 'C:\ProgramData\chocolatey'
WARNING: It's very likely you will need to close and reopen your shell
  before you can use choco.
Restricting write permissions to Administrators
We are setting up the Chocolatey package repository.
The packages themselves go to 'C:\ProgramData\chocolatey\lib'
  (i.e. C:\ProgramData\chocolatey\lib\yourPackageName).
A shim file for the command line goes to 'C:\ProgramData\chocolatey\bin'
  and points to an executable in 'C:\ProgramData\chocolatey\lib\yourPackageName'.

Creating Chocolatey folders if they do not already exist.

WARNING: You can safely ignore errors related to missing log files when
  upgrading from a version of Chocolatey less than 0.9.9.
  'Batch file could not be found' is also safe to ignore.
  'The system cannot find the file specified' - also safe.
WARNING: Not setting tab completion: Profile file does not exist at 'C:\Users\Yusuf
Rizal\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1'.
Chocolatey (choco.exe) is now ready.
You can call choco from anywhere, command line or powershell by typing choco.
Run choco /? for a list of functions.
You may need to shut down and restart powershell and/or consoles
  first prior to using choco.
Ensuring chocolatey commands are on the path
Ensuring chocolatey.nupkg is in the lib folder
PS C:\Windows\system32>
```

Proses instalasi hingga siap digunakan

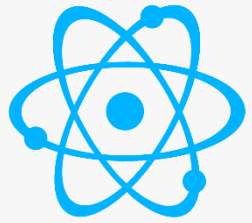


# Instalasi Kebutuhan Sistem (3)

Buka CMD 'Run as Administrator' dan jalankan perintah dibawah:

```
Administrator: Command Prompt
Microsoft windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.
C:\windows\system32>choco install -y nodejs.install python2 jdk8
```

# Instalasi Android Studio, SDK, AVD (Emulator)



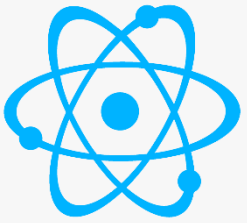
## android studio

Android Studio provides the fastest tools for building apps on every type of Android device.

**DOWNLOAD ANDROID STUDIO**

3.6.2 for Windows 64-bit (748 MB)

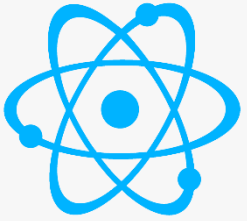




# Instalasi React App Creator

React Native memiliki antarmuka perintah tertulis yang built-in, dapat digunakan untuk menghasilkan proyek baru. Kita dapat mengaksesnya tanpa menginstal apapun menggunakan **npx**.

```
# npx react-native init firstreactnative
```



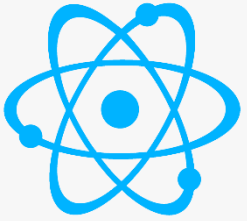
# Instalasi React App Creator (2)

1

```
Administrator: Node.js command prompt - npx react-native init firstreactnative
E:\reactnativeworkshop>npx react-native init firstreactnative
This will walk you through creating a new React Native project in E:\reactnativeworkshop\firstreactnative
Using yarn v1.22.4
Installing react-native...
yarn add v1.22.4
info No lockfile found.
[1/4] Resolving packages...
warning react-native > fbjs-scripts > core-js@2.6.11: core-js@<3 is no longer maintained and not recommended for usage due to the number of issues. Please, upgrade your dependencies to the actual version of core-js@3.
warning react-native > fbjs > core-js@2.6.11: core-js@<3 is no longer maintained and not recommended for usage due to the number of issues. Please, upgrade your dependencies to the actual version of core-js@3.
warning react-native > create-react-class > fbjs > core-js@1.2.7: core-js@<3 is no longer maintained and not recommended for usage due to the number of issues. Please, upgrade your dependencies to the actual version of core-js@3.
warning react-native > metro-babel-register > core-js@2.6.11: core-js@<3 is no longer maintained and not recommended for usage due to the number of issues. Please, upgrade your dependencies to the actual version of core-js@3.
warning react-native > @react-native-community/cli > metro > jest-haste-map > micromatch > snapdragon > source-map-resolve > resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
warning react-native > @react-native-community/cli > metro > jest-haste-map > micromatch > snapdragon > source-map-resolve > urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
[2/4] Fetching packages...
[#####-----] 149/617
```

2

```
Administrator: Node.js command prompt - npx react-native init firstreactnative
- jsx-ast-utils@2.2.3
- lolex@5.1.2
- metro-react-native-babel-preset@0.59.0
- node-notifier@6.0.0
- nwsapi@2.2.0
- object.entries@1.1.1
- object.fromentries@2.0.2
- object.values@1.1.1
- p-each-series@2.1.0
- parse5@5.1.0
- picomatch@2.2.2
- pkg-dir@4.2.0
- prettier-linter-helpers@1.0.0
- prettier@2.0.4
- regexp.prototype.flags@1.3.0
- request-promise-native@1.0.8
- request@2.88.2
- resolve-cwd@3.0.0
- sax@3.1.1
- string.prototype.matchall@4.0.2
- strip-bom@4.0.0
- strip-final-newline@2.0.0
- supports-hyperlinks@2.1.0
- terminal-link@2.1.1
- test-exclude@6.0.0
- to-regex-range@5.0.1
- type-detect@4.0.8
- typedarray-to-buffer@3.1.5
- v8-to-istanbul@4.1.3
- w3c-xmlserializer@1.1.2
- write-file-atomic@3.0.3
- xmlchars@2.2.0
- xregexp@4.3.0
Done in 57.66s.
```



# Struktur Folder Proyek

## ▼ FIRSTREACTNATIVE

> \_\_tests\_\_

> android

> ios

> node\_modules

≡ .buckconfig

⊗ .eslintrc.js

≡ .flowconfig

◆ .gitattributes

◆ .gitignore

JS .prettierrc.js

{ } .watchmanconfig

JS App.js

{ } app.json

JS babel.config.js

JS Halo.js

JS index.js

JS metro.config.js

{ } package-lock.json

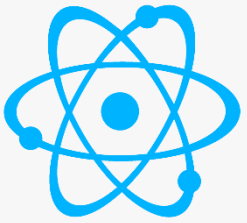
{ } package.json

**node\_modules:** Berisi seluruh modul yang dibutuhkan

**App.js:** berisi pemrograman javascript yang mengatur React native

**index.js:** sebagai entry point aplikasi, class mana yang dibuka pertama kali

**package.json:** berisi informasi dependensi apa saja yang ada dalam aplikasi



# Menjalankan aplikasi di Emulator Android

Pada Terminal, tuliskan perintah dibawah untuk memulai server:

**# npx react-native start**

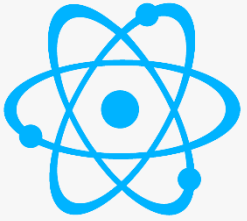
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

```
Microsoft Windows [Version 6.2.9200]  
(c) 2012 Microsoft Corporation. All rights reserved.
```

```
E:\reactnativeworkshop\firstreactnative>npx react-native start
```

## Menjalankan aplikasi di Emulator Android (2)

Tunggu hingga terminal memberikan keluaran seperti pada gambar disamping



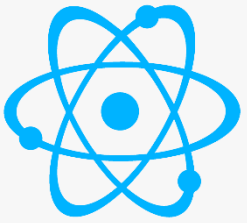
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
E:\reactnativeworkshop\firstreactnative>npx react-native start
```

```
#####
###      ####      ####      ###
##        ##      ##        ##
##          ####      ##
##          ####      ##
##          ##      ##
##          ##      ##
##          ##      ##
##          ##      ##
#####
### #####
#####      ##      #####
###      ##      ##      ##      ###
###      ##      ##      ##      ##
##          ####      ##
##          ####      ##
##          ####      ##
##          ##      ##
###      ##      ##      ##      ###
###      ##      ##      ##      ###
#####      ##      #####
##          #####
##          ##      ##
##          ##      ##
##          ##      ##
##          ##      ##
##          ##      ##
##          ##      ##
###      ##      ##      ##
#####
#####

Welcome to React Native!
Learn once, write anywhere
```

```
To reload the app press "r"
To open developer menu press "d"
```



# Menjalankan aplikasi di Emulator Android (3)

Buka Terminal baru dan tuliskan perintah dibawah untuk mulai menjalankan aplikasi di Emulator:

**# npx react-native run-android**

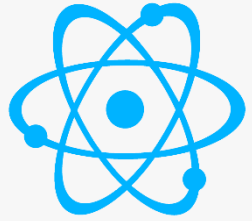
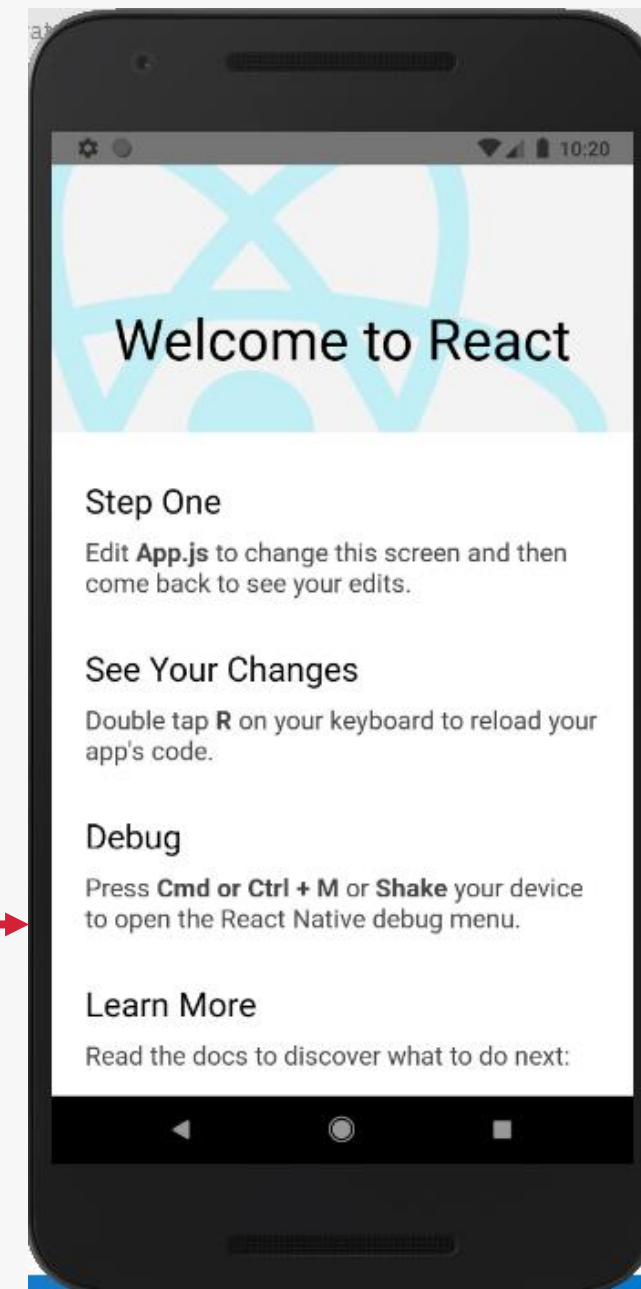
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

E:\reactnativeworkshop\firstreactnative>npx react-native run-android
```

# Menjalankan aplikasi di Emulator Android (4)

```
BUILD SUCCESSFUL in 2m 14s
27 actionable tasks: 27 executed
info Connecting to the development server...
info Starting the app...
Starting: Intent { cmp=com.firstreactnative/.MainActivity }
```

Proses berhasil maka aplikasi dijalankan di emulator



# Mengubah tampilan pertama

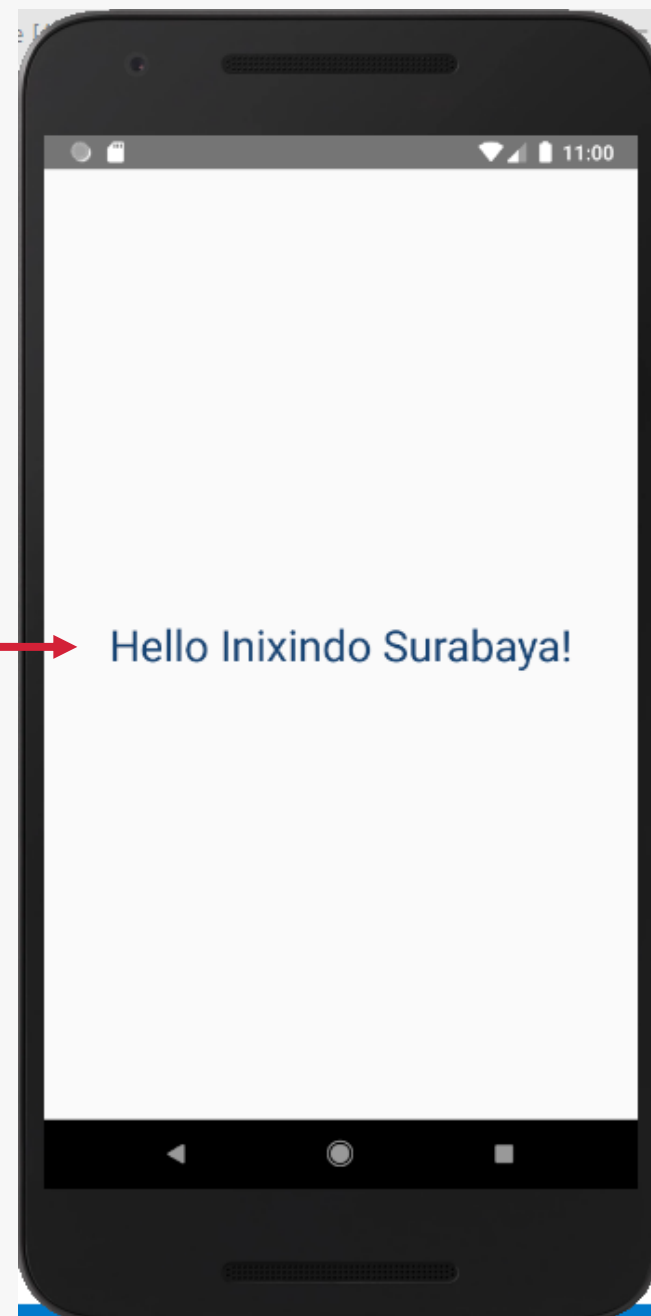
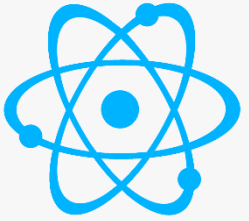
Buka file App.js dan ubah sesuai kode disamping

```
const App: () => React$Node = () => {  
  return (  
    <>  
      <View style={styles.container}>  
        <Text style={styles.simpletext}>Hello Inixindo Surabaya!</Text>  
      </View>  
    </>  
  );  
};  
  
const styles = StyleSheet.create({  
  container: {  
    flex: 1,  
    textAlign: 'center',  
    alignItems: 'center',  
    justifyContent: 'center',  
  },  
  simpletext: {  
    textAlign: 'center',  
    alignItems: 'center',  
    justifyContent: 'center',  
    fontSize: 30,  
  },  
});
```

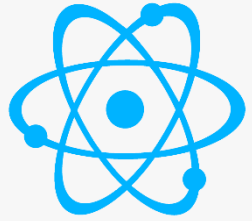


# Jalankan aplikasi

Muncul tulisan 'Hello Inixindo Surabaya!'  
Dengan posisi teks berada ditengah layar

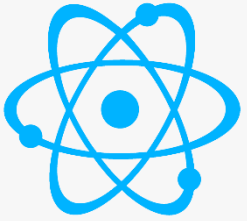


# Mengidentifikasi dan mendeteksi kesalahan



Tekan tombol Ctrl + M untuk membuka menu Debug





# Membedakan State dan Props

## State

State merupakan data yang tersimpan dalam sebuah component. State bersifat private dan hanya relevan terhadap component itu sendiri, tidak bisa di akses dari component lain. Berbeda dengan props yang valuenya dilempar dari component lain, state justru dapat menyimpan dan mengubah datanya sendiri dari dalam.

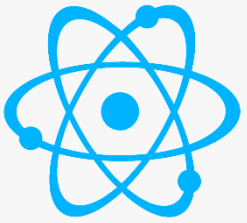
- Dimiliki oleh komponennya sendiri
- Bisa diubah dari dalam classnya sendiri (mutable)
- Memiliki `setState()` untuk mengubah nilainya

## Props

Prop singkatan dari Property. Ini mirip dengan atribut pada tag HTML, dalam pembuatannya value dari props bisa dilempar dan diakses dari atau ke component lain.

- Bersifat immutable (tidak bisa diubah)
- Bisa diakses dan dilempar dari dan ke component lain

# Membedakan State dan Props



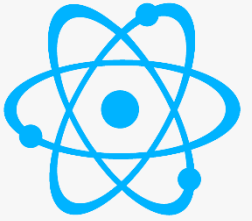
Buka lagi file App.js untuk memahami **State**

Ubah sesuai petunjuk kode berikut ini:

```
export default class App extends Component {
  constructor(props) {
    super(props);
    // buat state nilai awal disini
    this.state = {
      value: 'Hello Inixindo Surabaya',
    };

    // fungsi untuk mengubah isi text input
    onChangeTextInput = (text) => {
      // buat setState untuk mengubah state nilai awal sebelumnya
      this.setState({value: text});
    };
  }
}
```

```
render() {
  let {container, textStyle, textInput} = styles;
  let {value} = this.state;
  return (
    <View style={container}>
      {/* <Text style={textStyle}>Hello Inixindo Surabaya</Text> */}
      <Text style={textStyle}>{this.state.value}</Text>
      <TextInput
        style={textInput}
        onChangeText={this.onChangeTextInput}
        // value={'value text input'}
        // ubah value diatas sesuai dengan state saat ini
        value={value}
      />
    </View>
  );
}
```



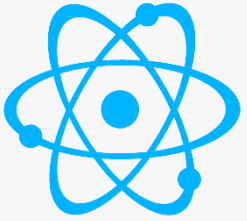
# Membedakan State dan Props

Buka lagi file App.js untuk memahami Props  
Buat sebuah file baru ChildComponent.js

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

export default class ChildComponent extends Component {
  constructor(props) {
    super(props);
  }

  render() {
    let {data1} = this.props;
    return (
      <View>
        <Text style={{fontSize: 20}}>{data1}</Text>
      </View>
    );
  }
}
```



# Membedakan State dan Props

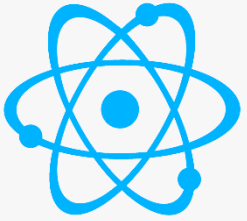
Buka kembali file App.js dan tambahkan baris kode seperti pada gambar berikut:

```
import {ChildComponent} from './ChildComponent';

render() {
  let {container, textStyle, textInput} = styles;
  let {value} = this.state;
  return (
    <View style={container}>
      <ChildComponent data1={value} data2={'100'} data3={'33'} />

      /* <Text style={textStyle}>Hello Inixindo Surabaya</Text> */
      /* <Text style={textStyle}>{this.state.value}</Text> */

      <TextInput
        style={textInput}
        onChangeText={this.onChangeTextInput}
        // value={'value text input'}
        // ubah value diatas sesuai dengan state saat ini
        value={value}
      />
    </View>
  );
}
```



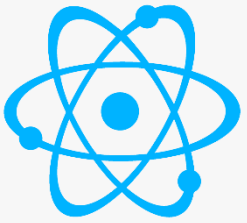
# Hasil State dan Props



**State** => nilai awal yang ditentukan dan bisa diubah menggunakan `setState()` dari dalam component itu sendiri (mutable)

**Props** => nilai asli dari component dan tidak bisa diubah (immutable) tapi dapat dilempar ke component yang lain

Text Input untuk menuliskan teks



# Menggunakan React Navigation

Untuk menggunakan React Navigation, harus mengunduhnya terlebih dahulu, tuliskan perintah dibawah ini:

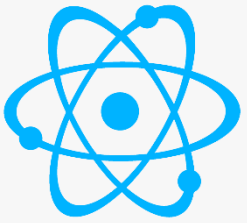
**# yarn add react-navigation**

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

```
Microsoft Windows [Version 6.2.9200]  
(c) 2012 Microsoft Corporation. All rights reserved.
```

```
E:\reactnativeworkshop\firstreactnative>yarn add react-navigation
```

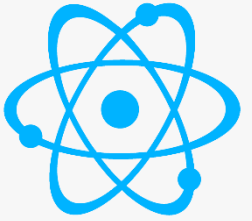




# Lihat Package.json

```
{
  "name": "firstreactnative",
  "version": "0.0.1",
  "private": true,
  "scripts": {
    "android": "react-native run-android",
    "ios": "react-native run-ios",
    "start": "react-native start",
    "test": "jest",
    "lint": "eslint ."
  },
  "dependencies": {
    "react": "16.11.0",
    "react-native": "0.62.1",
    "react-navigation": "^4.3.7"
  },
}
```

React Navigation berhasil ditambahkan



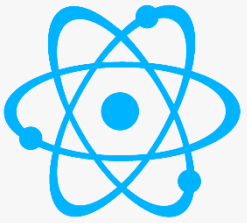
# Membaca data dari Web API

Pada bagian ini kita akan membuat sebuah aplikasi React Native yang digunakan untuk membaca data dari Web API berbentuk json.

Buat sebuah folder baru dengan nama **components**

Didalam folder tersebut buat 2 file baru dengan nama masing-masing **HomeScreen.js** dan **DetailScreen.js**

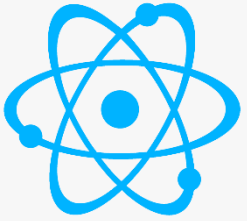
# Buka file HomeScreen.js



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

class HomeScreen extends Component {
  static navigationOptions = {
    title: 'Home',
  };
  render() {
    return (
      <View style={{ flex: 1, alignItems: 'center', justifyContent: 'center' }}>
        <Text>Home Screen</Text>
        <Button
          title="Go to Details"
          onPress={() => this.props.navigation.navigate('Details')}
        />
      </View>
    );
  }
}

export default HomeScreen;
```

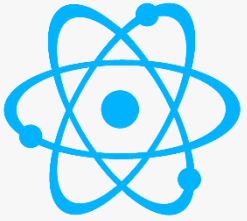


# Buka file DetailScreen.js

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

class DetailsScreen extends Component {
  static navigationOptions = {
    title: "Details"
  };
  render() {
    return (
      <View style={{ flex: 1, alignItems: "center", justifyContent: "center" }}>
        <Text>Details Screen</Text>
        <Button
          title="Buka Details... lagi"
          onPress={() => this.props.navigation.push("Details")}
        />
        <Button
          title="Kembali ke Home"
          onPress={() => this.props.navigation.navigate("Home")}
        />
        <Button
          title="Kembali"
          onPress={() => this.props.navigation.goBack()}
        />
      </View>
    );
  }
}

export default DetailsScreen;
```



# Buka file App.js

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

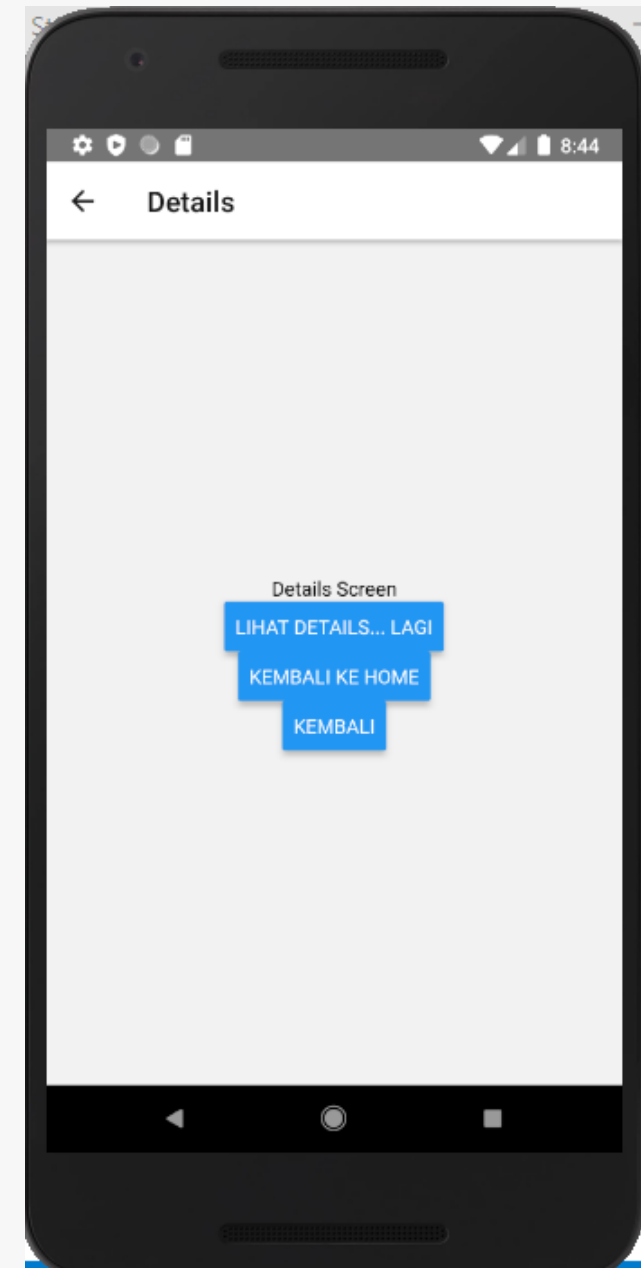
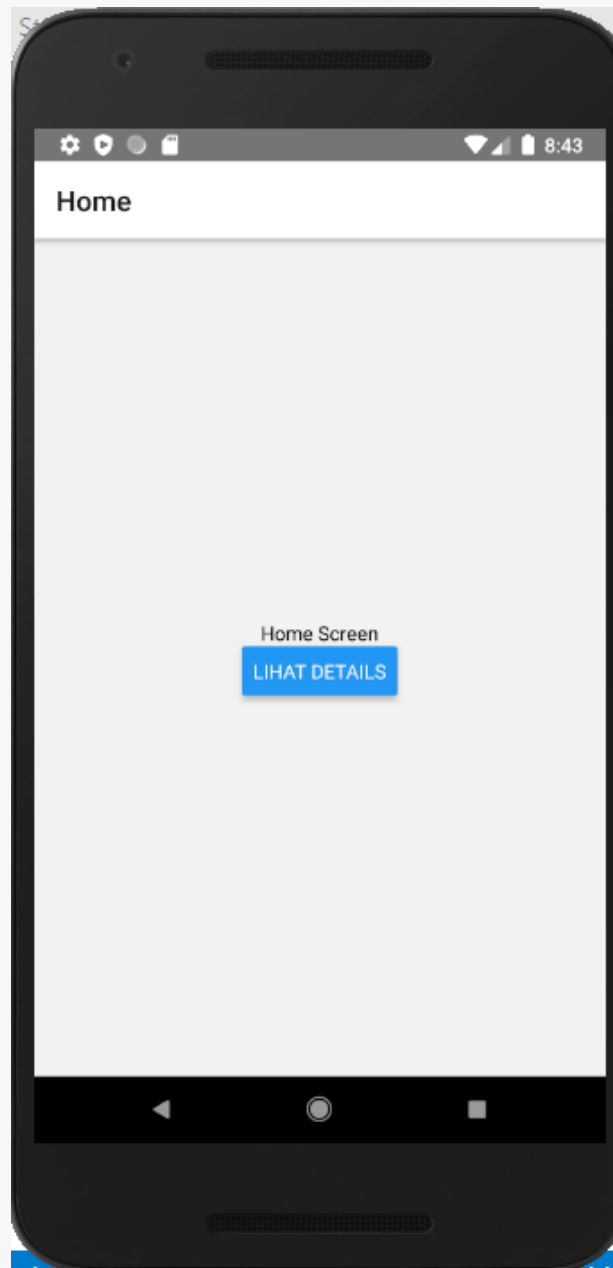
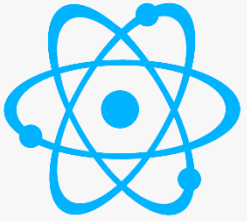
import { createAppContainer, createSwitchNavigator } from "react-navigation";
import { createStackNavigator } from "react-navigation-stack";

import HomeScreen from "../components/HomeScreen";
import DetailsScreen from "../components/DetailsScreen";

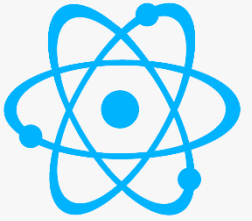
const StackNavigator = createStackNavigator(
  {
    Home: {screen: HomeScreen},
    Details: {screen: DetailsScreen}
  },
  {
    initialRouteName: "Home",
    navigationOptions: {
      headerStyle: {
        backgroundColor: "#f4511e"
      },
      headerTintColor: "#fff",
      headerTitleStyle: {
        fontWeight: "bold"
      }
    }
  }
);

export default createAppContainer(StackNavigator);
```

# Jalankan aplikasi di Emulator

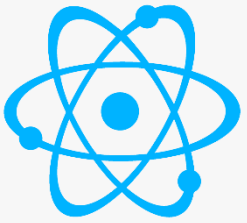


# Menampilkan data Web API World Cup 2018



Pada components/HomeScreen.js tambahkan constructor sebelum fungsi rendering data

```
constructor(props){  
  super(props);  
  this.state = { isLoading: true}  
}
```



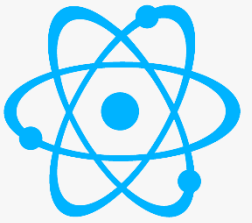
# Fungsi menampilkan data API json

```
componentDidMount(){
  return fetch('https://raw.githubusercontent.com/openfootball/world-cup.json/master/2018/worldcup.json')
    .then((response) => response.json())
    .then((responseJson) => {
      // console.log(responseJson);
      this.setState({
        isLoading: false,
        dataSource: responseJson.rounds,
      }, function(){

      });

    })
    .catch((error) =>{
      console.error(error);
    });
}
```





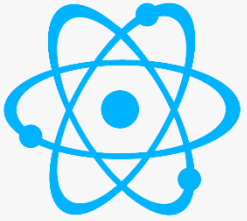
# Instalasi React Native Elements

Kita bisa menggunakan komponen List dan ListItem yang ada pada react-native-elements, cara instalasinya sebagai berikut:

```
# yarn add react-native-elements
```

Buka kembali HomeScreen.js dan tambahkan importnya  
`import { List, ListItem } from 'react-native-elements';`

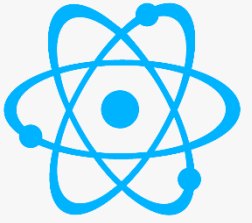
# Render HomeScreen.js



```
render() {
  if(this.state.isLoading){
    return(
      <View style={styles.activity}>
        <ActivityIndicator/>
      </View>
    )
  }

  return(
    <ScrollView style={styles.container}>
      <List>
        {
          this.state.dataSource.map((item, i) => (
            <ListItem
              key={i}
              title={item.name}
              leftIcon={{name: 'soccer-ball-o', type: 'font-awesome'}}
              onPress={() => {
                this.props.navigation.navigate('Details', {
                  matches: `${JSON.stringify(item.matches)}`,
                });
              }}
            </ListItem>
          ))
        }
      </List>
    </ScrollView>
  );
}
```

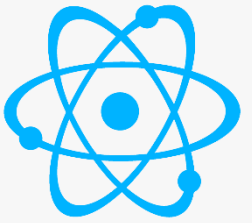
```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    paddingBottom: 22
  },
  item: {
    padding: 10,
    fontSize: 18,
    height: 44,
  },
  activity: {
    flex: 1,
    padding: 20,
  }
})
```



# Menampilkan Details Pertandingan

Buka kembali file DetailsScreen.js dan tambahkan importnya:

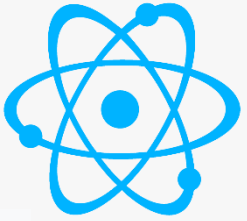
```
import React, { Component } from 'react';  
import { StyleSheet, ScrollView, View } from 'react-native';  
import { List, ListItem, Text, Card } from 'react-native-elements';
```



# Render DetailScreen.js

```
render() {  
  const { navigation } = this.props;  
  const matches = JSON.parse(navigation.getParam('matches', 'No matches found'));  
  console.log(matches);  
}
```

```
const styles = StyleSheet.create({  
  container: {  
    flex: 1,  
    padding: 20  
  },  
  subContainer: {  
    flex: 1,  
    paddingBottom: 20,  
    borderBottomWidth: 2,  
    borderBottomColor: '#CCCCCC',  
  }  
})
```

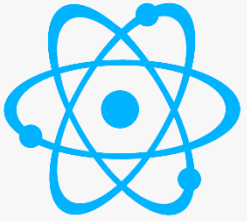


# Render DetailScreen.js (2)

```
return (  
  <ScrollView>  
    <Card style={styles.container}>  
      {  
        matches.map((item, key) => (  
          <View key={key} style={styles.subContainer}>  
            if(item.group) {  
              <View>  
                <Text h3>{item.group}</Text>  
              </View>  
            }  
            <View>  
              <Text h3>{item.team1.name} vs {item.team2.name}</Text>  
            </View>  
            <View>  
              <Text h5>{item.date}</Text>  
            </View>  
            <View>  
              <Text h4>{item.score1} - {item.score2}</Text>  
            </View>  
            if(item.goals1.length > 0) {  
              item.goals1.map((item2, key2) => (  
                <View key={key2}>  
                  <Text h4>{item2.name} ({item2.minute})</Text>  
                </View>  
              ))  
            }  
          </View>  
        ))  
      }  
    </Card>  
  </ScrollView>  
)
```

```
    if(item.goals2.length > 0) {  
      item.goals2.map((item3, key3) => (  
        <View key={key3}>  
          <Text h5>{item3.name} ({item3.minute})</Text>  
        </View>  
      ))  
    }  
  </View>  
))  
}</Card>  
</ScrollView>  
);  
}
```

# Hasil



Home	Details
<ul style="list-style-type: none"><li>Matchday 1</li><li>Matchday 2</li><li>Matchday 3</li><li>Matchday 4</li><li>Matchday 5</li><li>Matchday 6</li><li>Matchday 7</li><li>Matchday 8</li><li>Matchday 9</li><li>Matchday 10</li><li>Matchday 11</li><li>Matchday 12</li><li>Matchday 13</li></ul>	<div><div>Group A</div><div>Egypt vs Uruguay</div><div>2018-06-15</div><div>0 - 1</div><div>Giménez (89)</div></div> <div><div>Group B</div><div>Portugal vs Spain</div><div>2018-06-15</div><div>3 - 3</div><div>Ronaldo (4)</div><div>Ronaldo (44)</div><div>Ronaldo (88)</div><div>Costa (24)</div><div>Costa (55)</div><div>Nacho (58)</div></div> <div><div>Group B</div><div>Morocco vs Iran</div><div>2018-06-15</div><div>0 - 1</div><div>Bouhaddouz (90)</div></div>

