

## KATA PENGANTAR

Puji syukur ke hadirat Allah SWT, Tuhan semesta alam. Modul pembelajaran ini dibuat sebagai pedoman dan penunjang kegiatan training pada kelas Fullstack Android Developer di Nurul Fikri Computer. Modul ini diharapkan dapat membantu peserta dalam mempersiapkan dan mengikuti kegiatan training dengan lebih baik, terarah, dan terencana. Pada setiap bagian telah disusun materi belajar berupa tutorial atau instruksi sederhana sesuai dengan materi-materi yang akan di pelajari pada setiap pertemuannya.

Setiap bahasan materi akan diperdalam dengan pengerjaan tugas-tugas yang sesuai dengan tujuan untuk memperdalam pemahaman peserta mengenai materi yang dibahas. Tentu saja tingkat keaktifan dan kreatifitas setiap peserta diharapkan dapat ditingkatkan sehingga dapat mencapai hasil maksimal dalam belajar.

Penyusun meyakini bahwa pembuatan modul React Native ini masih jauh dari sempurna. Oleh sebab itu penyusun mengharapkan kritik dan saran yang membangun guna penyempurnaan modul pembelajaran ini di masa yang akan datang. Akhir kata, penyusun mengucapkan terima kasih kepada semua pihak yangtelah membantu, baik secara langsung maupun tidak langsung.

**Jakarta, 30 Oktober 2019**

**Penyusun**

## DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR.....	v
BAB-1 PENGENALAN Dasar .....	1
1.1.    Apa Itu React Native? .....	1
1.2.    Sejarah React Native.....	2
1.3.    Bagaimana Cara Kerja React Native? .....	2
1.4.    Keuntungan Menggunakan React Native .....	3
BAB-2 INSTALASI REACT NATIVE .....	4
2.1.    Install JDK dan Android Studio .....	4
2.2.    Install chocolatey .....	5
2.3.    Install Node JS dan Python2 .....	6
2.4.    Install React Native .....	7
2.5.    Setting Android Studio.....	7
2.5.1.  Setting SDK .....	7
2.5.2.  Setting AVD Manager .....	9
2.6.    Create Project React Native Sederhana .....	11
BAB-3 PROP DAN STATE PADA REACT NATIVE .....	16
3.1.    State .....	16
3.2.    Prop.....	19
BAB-4 HEIGHT DAN WIDTH PADA REACT NATIVE .....	21

6.1.	Fixed Dimensions .....	21
6.2.	Flex Dimensions .....	22
BAB-5 STYLE PADA REACT NATIVE .....		24
BAB-6 LAYOUT MENGGUNAKAN FLEXBOX .....		26
6.1.	Flex Direction .....	26
6.2.	Justify Content .....	28
6.3.	Align Items .....	30
BAB-7 HANDLING TEXT INPUT .....		33
7.1.	Pengertian Text Input .....	33
7.2.	Properti pada Text Input .....	33
7.3.	Contoh Penggunaan Text Input .....	34
7.3.1.	Konfigurasi pada App.js .....	34
7.3.2.	Halaman Input.js .....	34
7.3.3.	Tampilan pada Android .....	37
BAB-8 HANDLING TOUCHES .....		38
8.1.	Menampilkan Basic Button .....	38
8.2.	Jenis Komponen Touches .....	41
8.2.1.	TouchableHighlight .....	41
8.2.2.	TouchableNativeFeedback .....	45
8.2.3.	TouchableOpacity .....	47
8.2.4.	TouchableWithoutFeedback .....	51
BAB-9 MENGGUNAKAN SCROLL VIEW .....		55
9.1.	Pengertian Scroll View .....	55

9.2.	Property Scroll View .....	55
9.3.	Contoh Penggunaan Scroll View.....	55
BAB-10 LIST VIEW .....		58
10.1.	Flat List.....	58
10.2.	Section List.....	60
BAB-11 NETWORKING.....		63
11.1.	Fetch .....	63
11.1.1.	Request Menggunakan Fetch .....	63
11.1.2.	Handling Response.....	64
11.2.	XMLHttpRequest .....	65
11.3.	Web Socket.....	66

## DAFTAR GAMBAR

Gambar- 1. Logo React Native .....	1
Gambar- 2. Contoh Aplikasi React Native.....	2
Gambar- 3. Cara Kerja React Native .....	3
Gambar- 4. Cek JDK.....	4
Gambar- 5. Download Android Studio.....	4
Gambar- 6. Install Android Studio .....	5
Gambar- 7. Install Chocolatey.....	6
Gambar- 8. Install NodeJS.....	6
Gambar- 9. Install Python2 .....	7
Gambar- 10. Install React Native .....	7
Gambar- 11. Setting SDK Platform.....	8
Gambar- 12. Setting SDK Tools .....	8
Gambar- 13. Setting Android Home .....	9
Gambar- 14. Setting AVD Manager .....	9
Gambar- 15. Create Device AVD Manager .....	10
Gambar- 16. Select Hardware Virtual.....	10
Gambar- 17. Verifikasi Konfigurasi AVD .....	11
Gambar- 18. Create MyReactNative.....	11
Gambar- 19. Isi Folder MyReactNative yang Terbentuk.....	12
Gambar- 20. Perintah Run Android .....	12
Gambar- 21. Finish Run Android.....	13
Gambar- 22. Tampilan Log Running Project .....	13
Gambar- 23. Tampilan Aplikasi Pada Emulator .....	14
Gambar- 24. Tampilan Setelah Perubahan.....	15
Gambar- 25. Contoh Penggunaan State .....	17
Gambar- 26. Contoh updatState.....	18
Gambar- 27. Contoh Penggunaan Prop .....	20

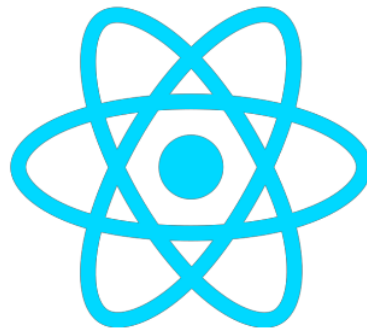
Gambar- 28. Fixed Dimensions .....	22
Gambar- 29. Flex Dimensions .....	23
Gambar- 30. Style React Native .....	25
Gambar- 31. flexDirection Column .....	27
Gambar- 32. flexDirection Row.....	27
Gambar- 33. flexDirection Row-Reverse .....	28
Gambar- 34. flexDirection Column-Reverse .....	28
Gambar- 35. justifyContent flex-start.....	29
Gambar- 36. justifyContent flex-end .....	29
Gambar- 37. justifyContent center .....	29
Gambar- 38. justifyContent space-around .....	30
Gambar- 39. justifyContent space-between.....	30
Gambar- 40. justifyContent space-evenly .....	30
Gambar- 41. alignItems stretch .....	31
Gambar- 42. alignItems flex-start .....	31
Gambar- 43. alignItems flex-end .....	31
Gambar- 44. alignItems Center .....	32
Gambar- 45. alignItems Baseline .....	32
Gambar- 46. Tampilan Text Input.....	37
Gambar- 47. Basic Button .....	41
Gambar- 48. touchableHighlight.....	44
Gambar- 49. TouchableOpacity .....	50
Gambar- 50. touchableWithoutFeedback .....	54
Gambar- 51. Scroll View.....	57
Gambar- 52. Flat List .....	60
Gambar- 53. Section List .....	62

## BAB-1 PENGENALAN DASAR

Tujuan pembelajaran:

- Siswa dapat mengenal lebih dalam mengenai React Native
- Siswa dapat mengerti bagaimana cara kerja React Native
- Siswa mengetahui contoh contoh aplikasi React Native

### 1.1. Apa Itu React Native?



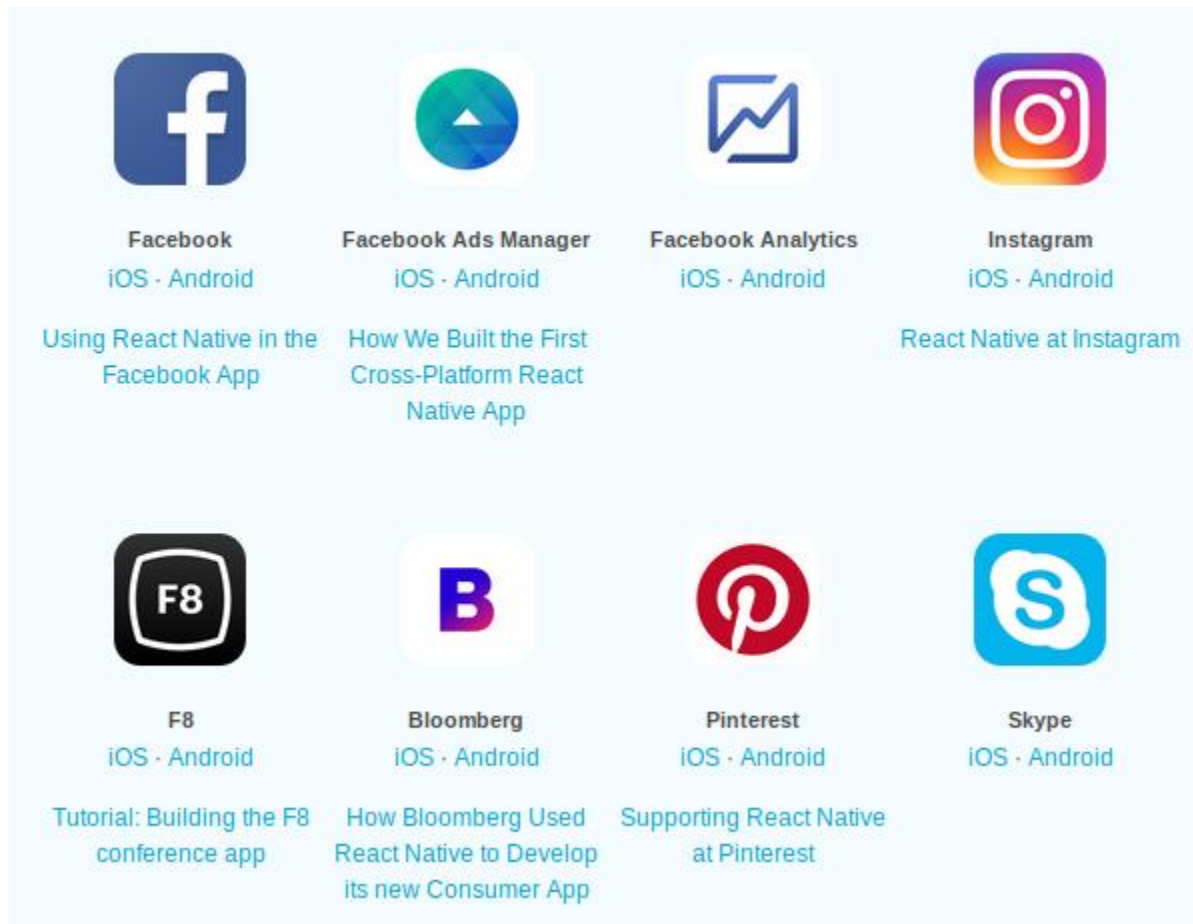
# React Native

*Gambar- 1. Logo React Native*

React Native merupakan sebuah framework javascript yang dikembangkan oleh facebook untuk membuat aplikasi mobile native untuk Android dan IOS. React Native ini hanya menggunakan JavaScript untuk membangun aplikasinya.

Aplikasi react native bukan merupakan aplikasi web. Aplikasi ini hanya berjalan pada perangkat seluler, sehingga tidak dapat dimuat di browser. Aplikasi react native juga bukan termasuk aplikasi hybrid yang biasanya dibangun menggunakan Ionic, Phone Gap, dll. Aplikasi react native akan mengkompilasi aplikasi kedalam native kode Android dan IOS, sedangkan Ionic biasanya akan membungkus aplikasi di dalam sebuah browser.

Berikut ini ada beberapa contoh aplikasi react native, diantaranya:



Gambar- 2. Contoh Aplikasi React Native

## 1.2. Sejarah React Native

Facebook mengembangkan React Native pada 2013 untuk proyek internal mereka Hackathon. Kemudian, dirilis secara publik pada Januari 2015 sebagai React.js, dan pada Maret 2015, Facebook mengumumkan bahwa React Native terbuka dan tersedia di GitHub.

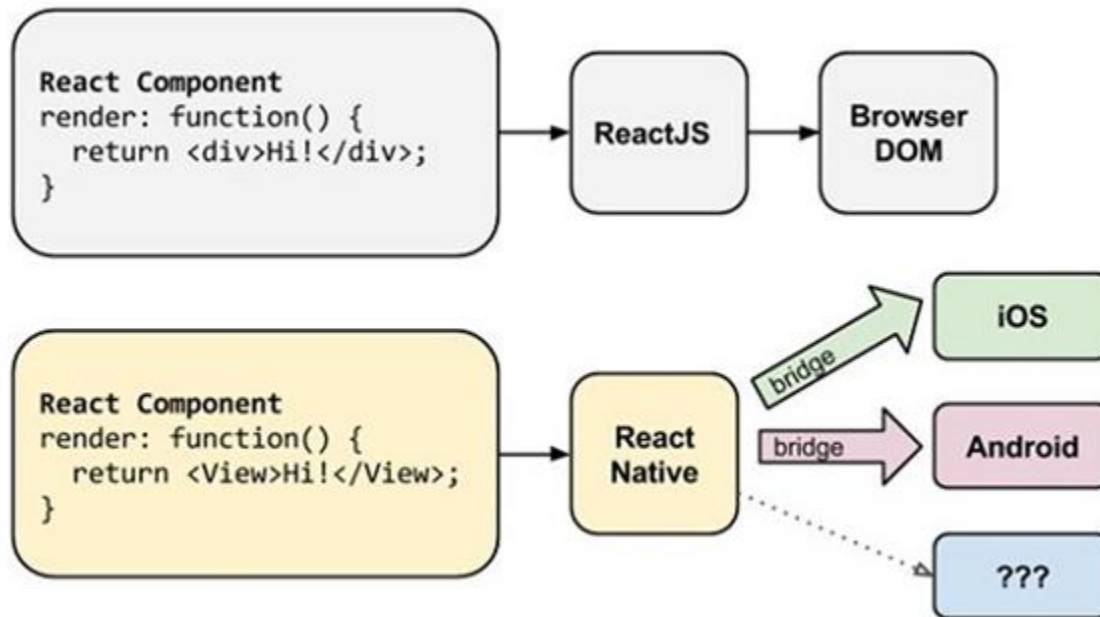
React Native pada awalnya dikembangkan untuk aplikasi iOS. Namun, baru-baru ini juga mendukung sistem operasi Android.

## 1.3. Bagaimana Cara Kerja React Native?

JavaScript menanamkan file berupa komponen react dalam bentuk bundle javascript di dalam aplikasi, dan menjalankannya secara local aplikasi yang kita buat. Namun kita juga bisa meletakkannya didalam server, sehingga kita bisa mengambil kapan saja melalui koneksi. Hal ini



lah yang membuat proses update lebih cepat. Untuk dapat lebih memahami cara kerja react native coba perhatikan gambar dibawah ini.



Gambar- 3. Cara Kerja React Native

#### 1.4. Keuntungan Menggunakan React Native

Berikut ini ada beberapa keuntungan dari penggunaan react native dalam membangun aplikasi seluler. Beberapa keuntungan tersebut diantaranya:

- Penggunaan Cross-Platform
- Performa Class
- javascript
- komunitas yang besar
- hot reloading

## BAB-2 INSTALASI REACT NATIVE

Tujuan pembelajaran

- Peserta mengetahui cara install React Native
- Peserta dapat mengimplementasikan cara instalasi

### 2.1. Install JDK dan Android Studio

Sebelum menginstall react native kita perlu menginstall beberapa tools pendukung seperti JDK 8 dan android studio. Pastikan computer kamu sudah terinstall JDK 8 dengan cara seperti berikut ini:

```
C:\Users\PC>java -version
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)

C:\Users\PC>
```

Gambar- 4. Cek JDK

Pastikan computer kamu juga sudah terinstall Android Studio. Jika belum maka kamu bisa download aplikasi android studio di [developer.android.com](https://developer.android.com).

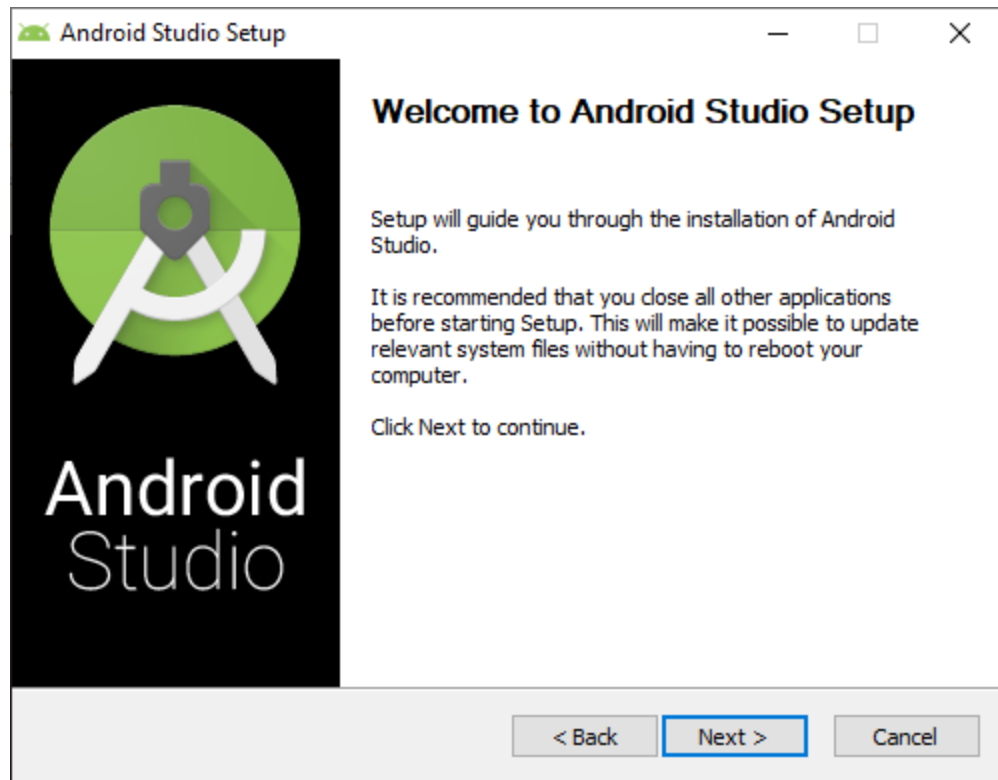
## android studio

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

**DOWNLOAD ANDROID STUDIO**

3.5.1 for Windows 64-bit (718 MB)

Gambar- 5. Download Android Studio



Gambar- 6. Install Android Studio

## 2.2. Install chocolatey

Facebook menyarankan agar kita menginstall chocolatey untuk mempermudah dalam proses penginstallan paket paket yang lain. Cara menginstall dapat dilihat pada <https://chocolatey.org/docs/installation>.

```

C:\Windows\system32>
C:\Windows\system32>@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoPr
m.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))" && SET "PATH=%PA
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\PC\AppData\Local\Temp\chocolatey\chocInstall\chocolatey.zip to C:\Us
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.
chocolatey.nupkg file not installed in lib.
Attempting to locate it from bootstrapper.
PATH environment variable does not have C:\ProgramData\chocolatey\bin in it. Adding...
WARNING: Not setting tab completion: Profile file does not exist at
'C:\Users\PC\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

C:\Windows\system32>

```

Gambar- 7. Install Chocolatey

### 2.3. Install Node JS dan Python2

Ketikan perintah **choco install nodejs** pada cmd > tekan **Y** > enter dan tunggu hingga prosesnya selesai.

```

C:\Windows\system32>choco install nodejs
Chocolatey v0.10.15
Installing the following packages:
nodejs
By installing you accept licenses for the packages.
Progress: Downloading nodejs.install 13.0.1... 18%

```

Gambar- 8. Install NodeJS

Install python2 jika proses instalasi node js telah selesai. Adapun caranya sama dengan diatas yaitu tinggal ketik **choco install python2 > tekan Y > enter** maka proses instalasi akan berjalan seperti dibawah ini.

```
C:\Windows\system32>choco install python2
Chocolatey v0.10.15
Installing the following packages:
python2
By installing you accept licenses for the packages.
Progress: Downloading python2 2.7.17... 65%
```

Gambar- 9. Install Python2

## 2.4. Install React Native

Untuk dapat install react native perlu kita tuliskan sebagai berikut :

```
npm install -g react-native-cli
```

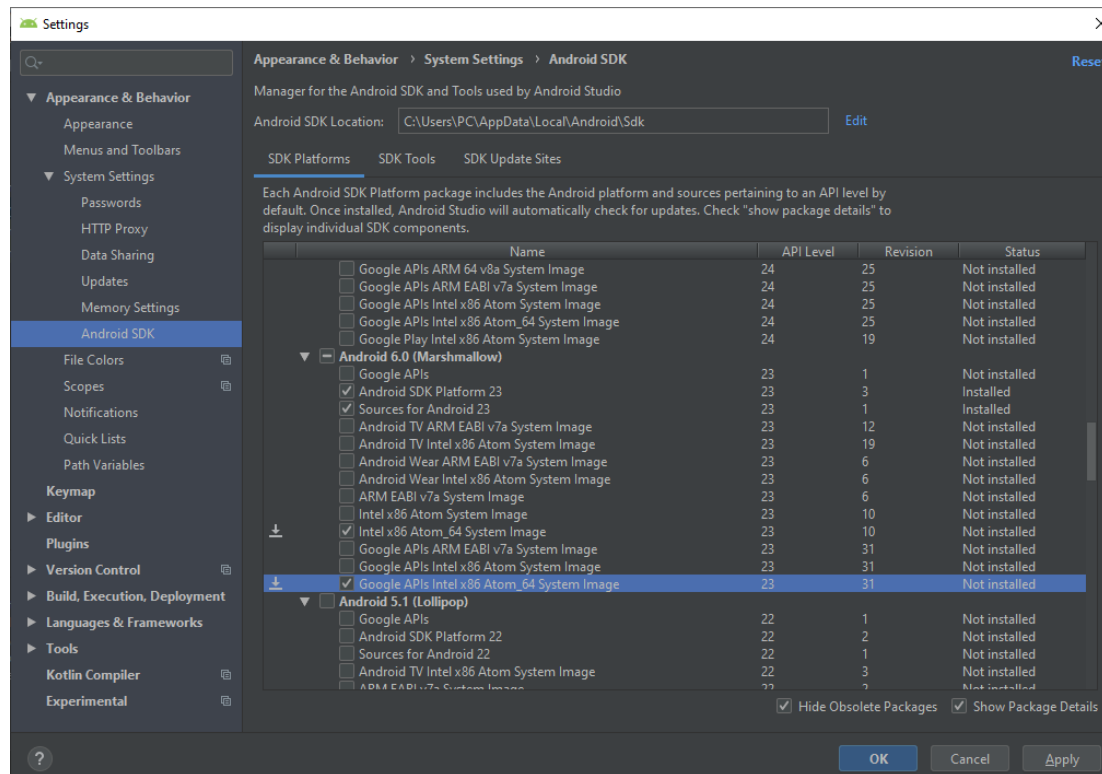
```
C:\Windows\system32>npm install -g react-native-cli
C:\Users\PC\AppData\Roaming\npm\react-native -> C:\Users\PC\AppData\Roaming\npm\node_modules\react-native-cli\index.js
+ react-native-cli@2.0.1
added 50 packages from 24 contributors in 12.002s
C:\Windows\system32>
```

Gambar- 10. Install React Native

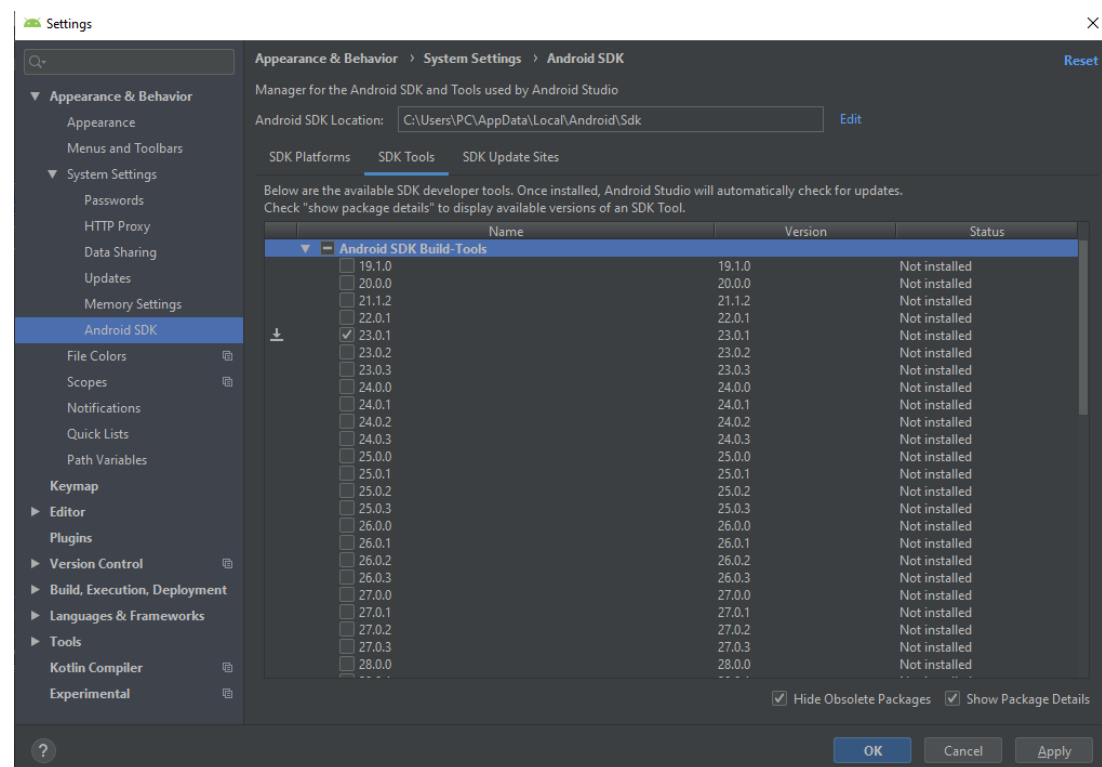
## 2.5. Setting Android Studio

### 2.5.1. Setting SDK

Untuk dapat menjalankan android studio kita perlu mendownload SDK pendukung seperti berikut ini. Buka Android Studionya lalu pergi ke Android SDK Managernya untuk mengunduh paket yang di rekomendasikan oleh Facebook. Unduh paket pada SDK Platform > Android 6.0 (Marshmallow) > pilih apply. Unduh juga SDK Tool dengan cara pilih SDK Tool > Versi 23.0.1 > Apply



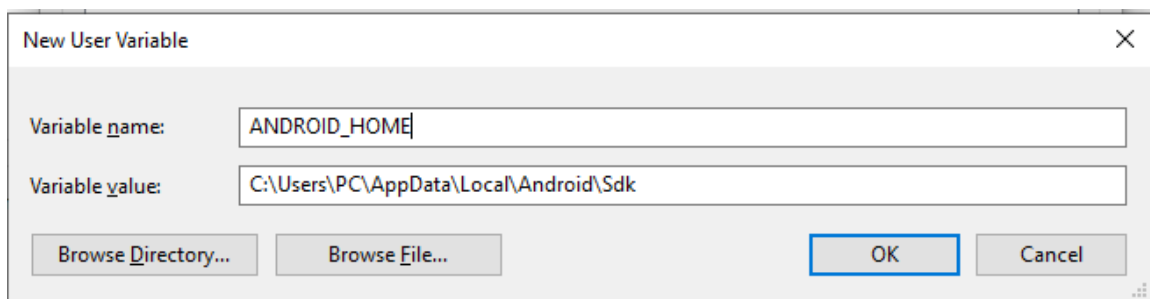
Gambar- 11. Setting SDK Platform



Gambar- 12. Setting SDK Tools

Kita juga perlu mendaftarkan Android Home agar bisa dikenali oleh windows dengan cara seperti dibawah ini:

- Buka control panel pada PC > System and Security > System > Advanced system settings.
- Tambahkan Environment Variablesnya lalu klik New dan ketikkan ANDROID\_HOME pada kolom variable name.
- Pada kolom variable valuenya tujuan ke direktori SDK kalian yang bisa kalian lihat pada SDK Manager tadi.

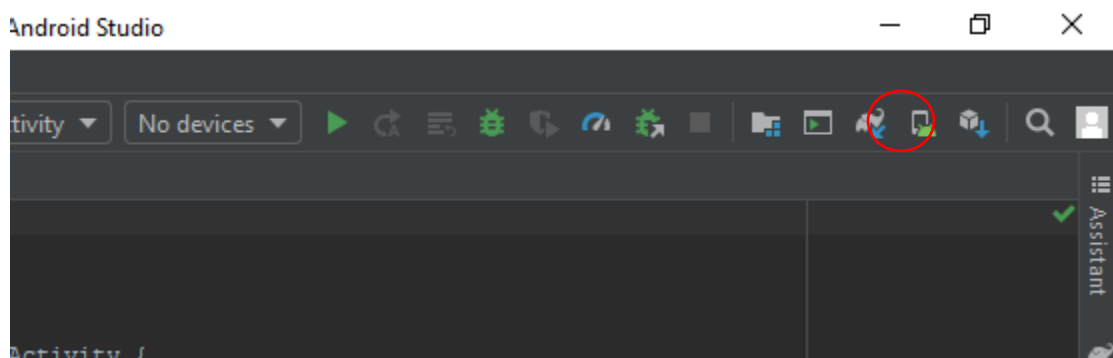


Gambar- 13. Setting Android Home

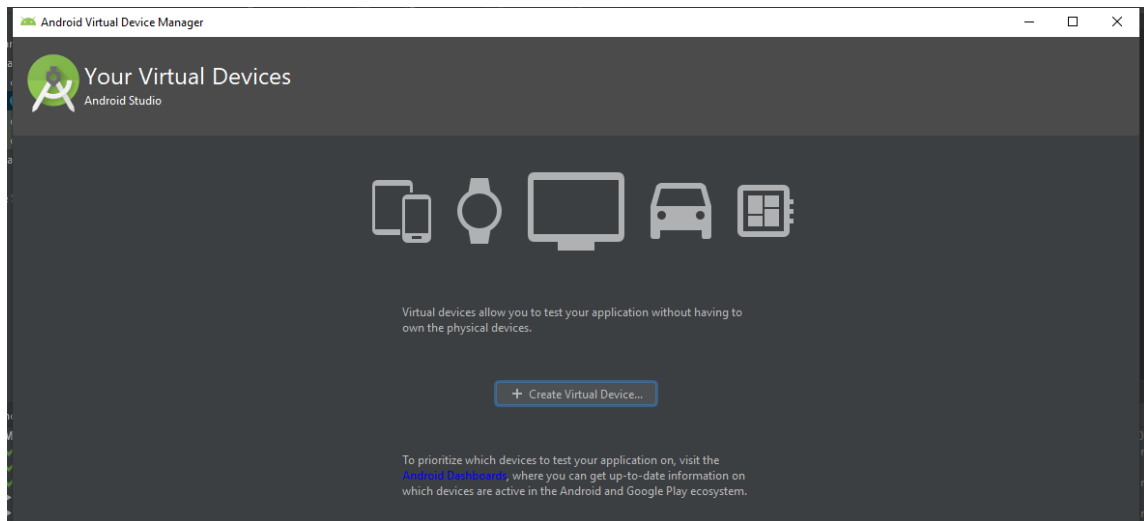
### 2.5.2. Setting AVD Manager

Untuk dapat menjalankan debugging aplikasi maka kita membutuhkan emulator. Untuk itu perlu kita setting seperti langkah dibawah ini.

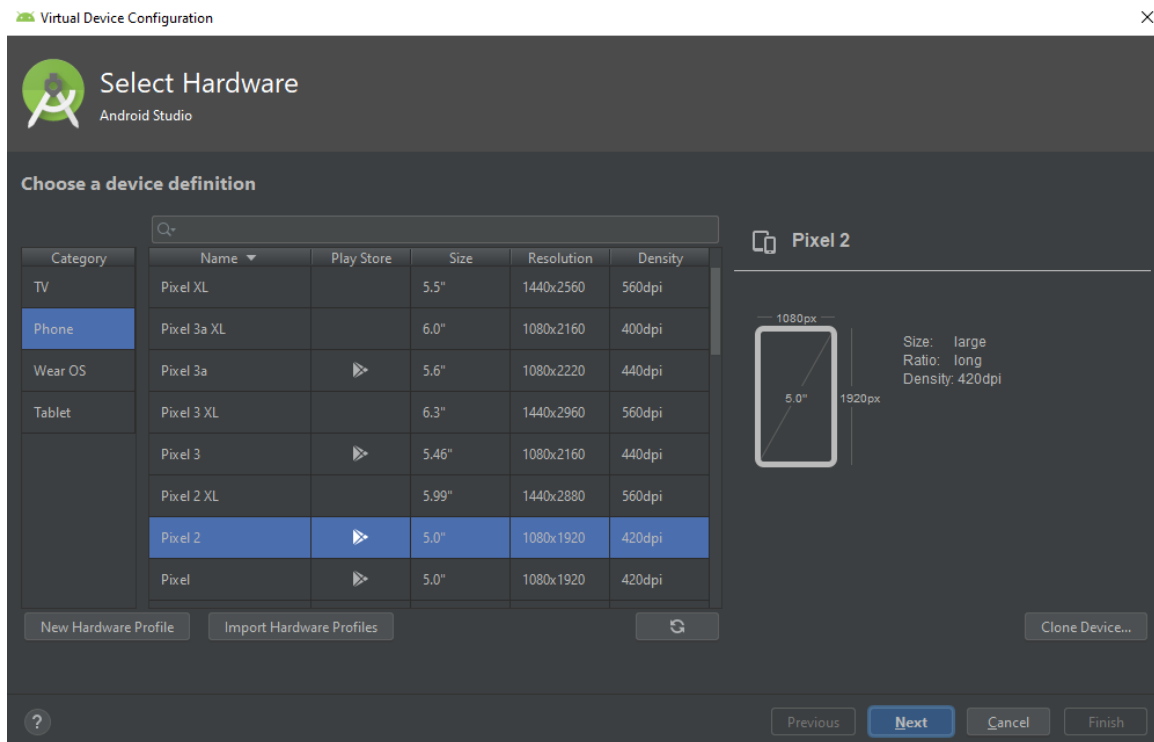
Klik pada icon AVD Manager



Gambar- 14. Setting AVD Manager

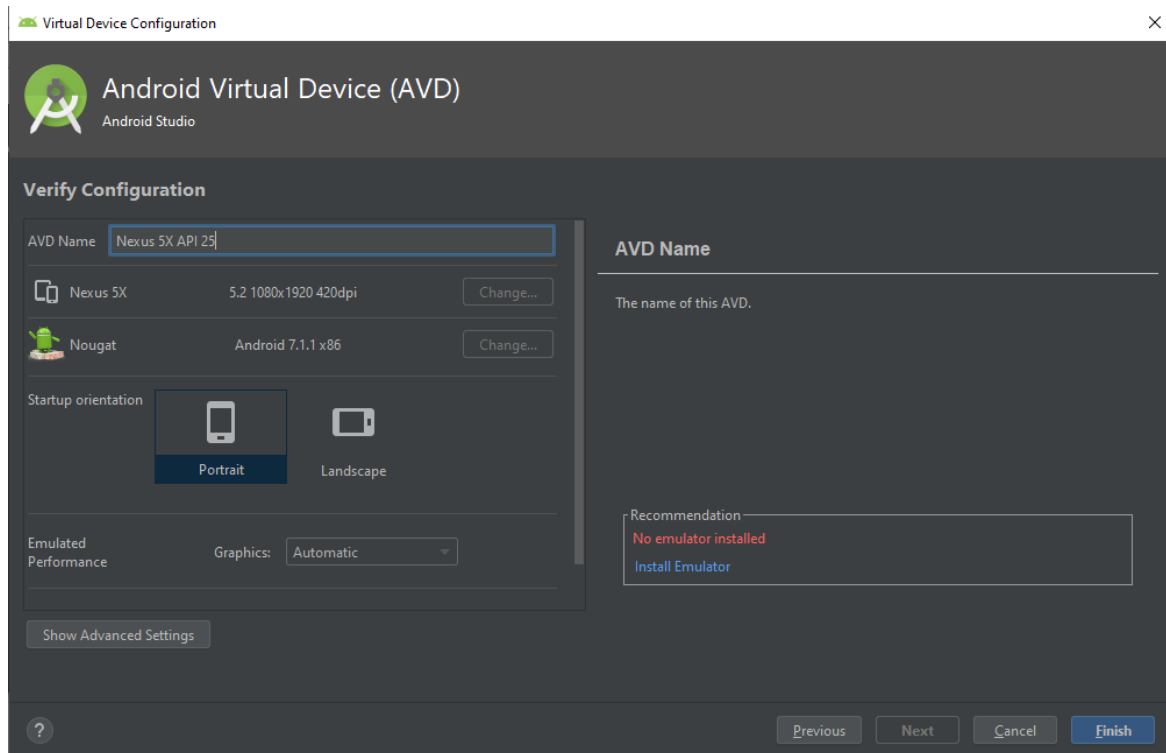


Gambar- 15. Create Device AVD Manager



Gambar- 16. Select Hardware Virtual





Gambar- 17. Verifikasi Konfigurasi AVD

## 2.6. Create Project React Native Sederhana

Setelah semua tools pendukung telah terinstall saatnya kita mulai membuat aplikasi react native pertama yang berisi hello world.

- Pastikan emulator telah siap
- Buka cmd lalu buat direktori baru yang akan kita gunakan untuk menyimpan project react native pertama kita. Pindahkan pada lokasi penyimpana seperti Documents lalu ketik perintah react-native init nama project.

```
C:\Users\PC\Documents>react-native init MyReactNative
This will walk you through creating a new React Native project in C:\Users\PC\Documents\MyReactNative
'yarn' is not recognized as an internal or external command,
operable program or batch file.
Installing react-native...
Consider installing yarn to make this faster: https://yarnpkg.com
[.....] - fetchMetadata: sill resolveWithNewModule sprintf-js@1.0.3 checking installable status
```

Gambar- 18. Create MyReactNative

- Jika proses sudah selesai maka akan terbentuk sebuah folder dengan nama project, dalam contoh ini folder MyReactNative. Berikut ini isi folder yang terbentuk.

is PC > Documents > MyReactNative

Name	Date modified	Type	Size
__tests__	31/10/2019 13:51	File folder	
android	31/10/2019 13:51	File folder	
ios	31/10/2019 13:51	File folder	
node_modules	31/10/2019 13:52	File folder	
.buckconfig	31/10/2019 13:51	BUCKCONFIG File	1 KB
.eslintrc	31/10/2019 13:51	JavaScript File	1 KB
.flowconfig	31/10/2019 13:51	FLOWCONFIG File	3 KB
.gitattributes	31/10/2019 13:51	GITATTRIBUTES File	1 KB
.gitignore	31/10/2019 13:51	GITIGNORE File	1 KB
.prettierrc	31/10/2019 13:51	JavaScript File	1 KB
.watchmanconfig	31/10/2019 13:51	WATCHMANCON...	1 KB
App	31/10/2019 13:51	JavaScript File	3 KB
app.json	31/10/2019 13:51	JSON File	1 KB
babel.config	31/10/2019 13:51	JavaScript File	1 KB
index	31/10/2019 13:51	JavaScript File	1 KB
metro.config	31/10/2019 13:51	JavaScript File	1 KB
package.json	31/10/2019 13:52	JSON File	1 KB
package-lock.json	31/10/2019 13:52	JSON File	331 KB

Gambar- 19. Isi Folder MyReactNative yang Terbentuk

- Setelah terbentuk folder kita perlu mendownload nodeJS. Maka kita masuk ke dalam folder yang terbentuk lalu ketik perintah react-native run-android. Pada proses ini juga akan mendownload gredle dan sdk.

```
C:\Users\PC\Documents\MyReactNative>react-native run-android
info Running jetifier to migrate libraries to AndroidX. You can disable it using "--no-jetifier" flag.
Jetifier found 864 file(s) to forward-jetify. Using 4 workers...
info Starting JS server...
* daemon not running; starting now at tcp:5037
* daemon started successfully
info Installing the app...
Downloading https://services.gradle.org/distributions/gradle-5.5-all.zip
.....
```

Gambar- 20. Perintah Run Android

- Setelah proses selesai maka akan sekaligus build aplikasi pada emulator.

```

03:42:07 V/ddms: execute 'am get-config' on '32b2bc937d14' : EOF hit. Read: -1
03:42:07 V/ddms: execute: returning
Installing APK 'app-debug.apk' on 'Redmi 4X - 7.1.2' for app:debug
03:42:07 D/app-debug.apk: Uploading app-debug.apk onto device '32b2bc937d14'
03:42:07 D/Device: Uploading file onto device '32b2bc937d14'
03:42:07 D/ddms: Reading file permission of C:\Users\PC\Documents\MyReactNative\android\app\build\outp
ebug.apk as: rwx-----
03:42:09 V/ddms: execute: running pm install -r -t "/data/local/tmp/app-debug.apk"
03:42:22 V/ddms: execute 'pm install -r -t "/data/local/tmp/app-debug.apk"' on '32b2bc937d14' : EOF h
03:42:22 V/ddms: execute: returning
03:42:22 V/ddms: execute: running rm "/data/local/tmp/app-debug.apk"
03:42:22 V/ddms: execute 'rm "/data/local/tmp/app-debug.apk"' on '32b2bc937d14' : EOF hit. Read: -1
03:42:22 V/ddms: execute: returning
Installed on 1 device.

Deprecated Gradle features were used in this build, making it incompatible with Gradle 6.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/5.5/userguide/command_line_interface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 53s
27 actionable tasks: 2 executed, 25 up-to-date
info Connecting to the development server...
info Starting the app on "32b2bc937d14"...
Starting: Intent { cmp=com.myreactnative/.MainActivity }

```

Gambar- 21. Finish Run Android

```

node "C:\Users\PC\Documents\MyReactNative\node_modules\react-native\scripts\...\cli.js" start

Running Metro Bundler on port 8081.

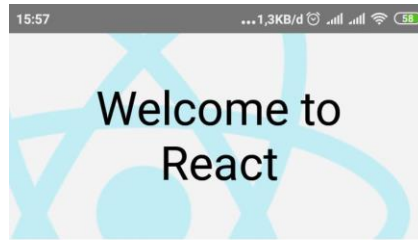
Keep Metro running while developing on any JS projects. Feel free to
close this tab and run your own Metro instance if you prefer.

https://github.com/facebook/react-native

Looking for JS files in
  C:\Users\PC\Documents\MyReactNative
Loading dependency graph, done.
BUNDLE [android, dev] ./index.js 100.0% (457/457), done.
LOG Running "MyReactNative" with {"rootTag":1}

```

Gambar- 22. Tampilan Log Running Project



### Step One

Edit **App.js** to change this screen and then come back to see your edits.

### See Your Changes

Double tap **R** on your keyboard to reload your app's code.

### Debug

Press **Cmd or Ctrl + M** or **Shake** your device to open the React Native debug menu.

### Learn More

*Gambar- 23. Tampilan Aplikasi Pada Emulator*

- Edit app.js untuk merubah tampilan screen pada aplikasi react native. Seperti contoh di bawah ini.

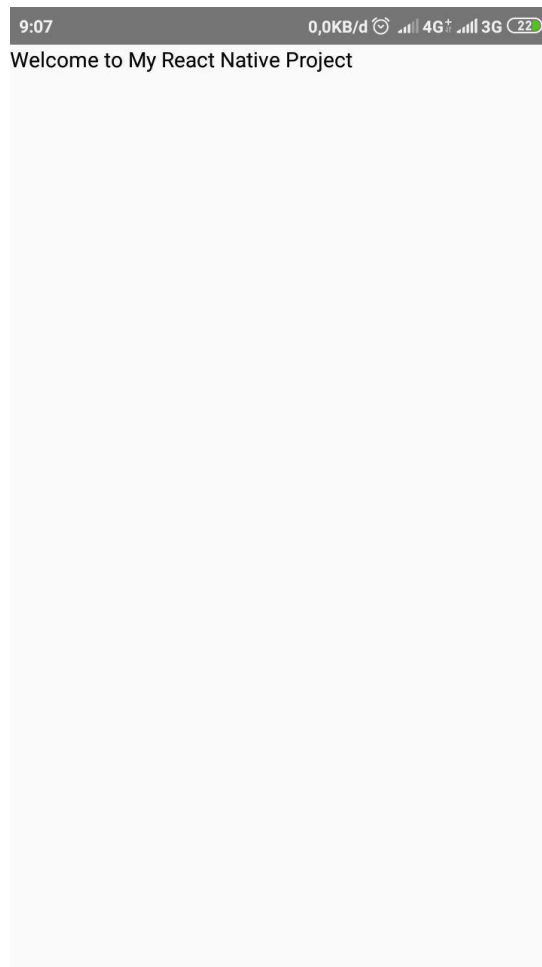
App.js

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

export default class App extends React.Component {
  render() {
    return (
      <View>
        <Text>Welcome to My React Native Project</Text>
      </View>
    );
  }
}
```

```
}
```

Hasilnya



*Gambar- 24. Tampilan Setelah Perubahan*

## BAB-3 PROP DAN STATE PADA REACT NATIVE

Tujuan pembelajaran:

- Peserta mampu memahami konsep prop dan state pada react native
- Peserta dapat mengimplementasikan prop dan state pada react native

### 3.1. State

Pada pemrograman lain kita mengenal variable, pada react native state dapat diartikan sebagai variable default yang dimiliki oleh computer. State ini bersifat mutable(dapat berubah) sehingga kita menggunakan state pada data yang berubah.

Secara umum kita perlu menginisialisasikan state pada konstraktor, kemudian memanggil setState ketika kita ingin melakukan suatu perubahan.

Untuk lebih memahami konsep tersebut coba kita ubah file app.js seperti berikut ini.

#### App.js

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

export default class App extends React.Component {
  state = {
    myState: 'Lorem ipsum dolor sit amet, consectetur adipisicing elit, used do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.'
  }
  render() {
    return (
      <View>
        <Text>{this.state.myState}</Text>
      </View>
    );
  }
}
```

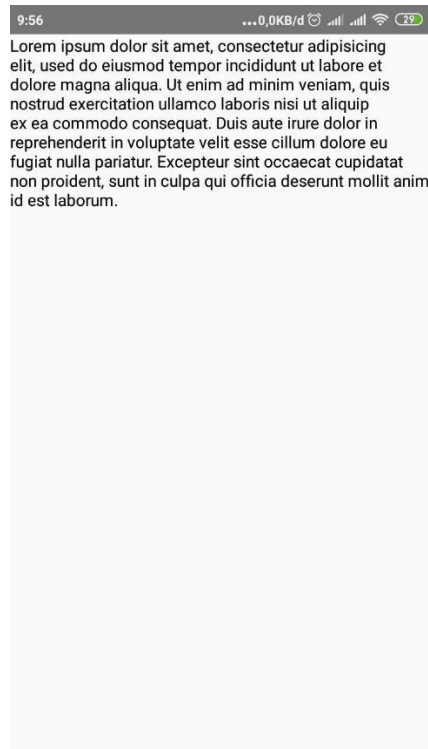
```
</View>

);

}

}
```

### Hasilnya



Gambar- 25. Contoh Penggunaan State

State dikatakan dapat berubah sehingga kita dapat melakukan perubahan dengan membuat fungsi `deleteState` dan memanggil nya dengan cara memanggil event `onPress={this.deleteText}`.

Berikut ini ada sebuah contoh **update state**.

### App.js

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

export default class App extends React.Component {
```

```

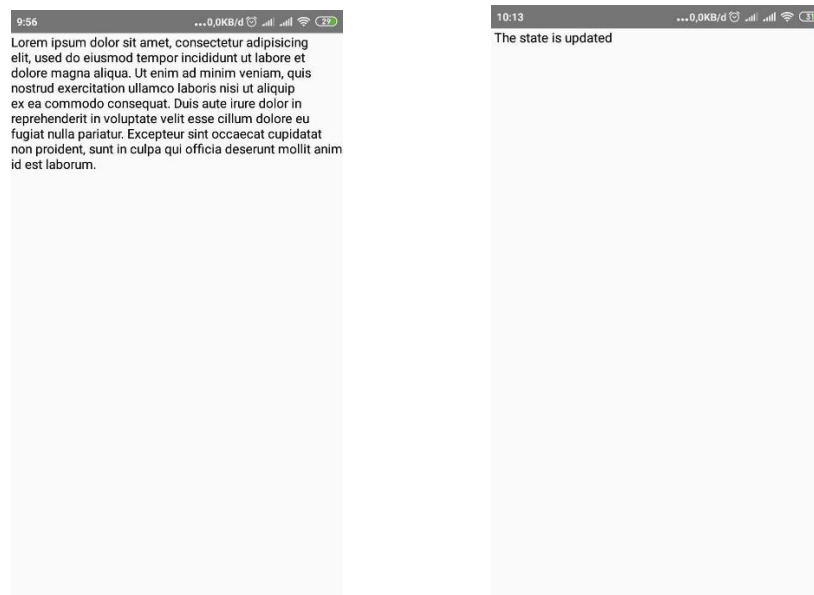
state = {
  myState: 'Lorem ipsum dolor sit amet, consectetur adipisicing elit, used do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint
occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.'
}

updateState = () => this.setState({ myState: 'The state is updated' })

render() {
  return (
    <View>
      <Text onPress = {this.updateState}> {this.state.myState} </Text>
    </View>
  );
}
}

```

Hasilnya



Gambar- 26. Contoh updatState



### 3.2. Prop

Prop merupakan kependekan dari property. Props sendiri bisa dibilang adalah sebuah parameter default yang dimiliki oleh suatu component dimana ketika anda memanggil sebuah component. Prop berguna untuk mempermudah developer mengelola parameter pada suatu component. Untuk lebih mendalami coba perhatikan dibawah ini.

#### App.js

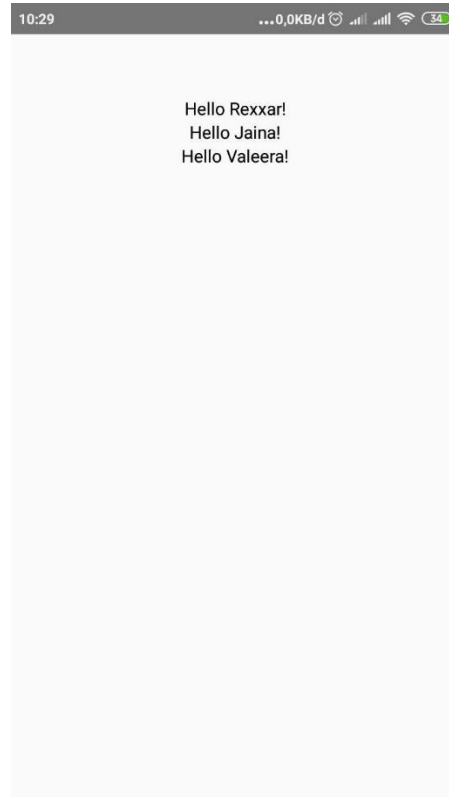
```
import React, { Component } from 'react';
import { Text, View } from 'react-native';

class Greeting extends Component {
  render() {
    return (
      <View style={{alignItems: 'center'}}>
        <Text>Hello {this.props.name}!</Text>
      </View>
    );
  }
}

export default class LotsOfGreetings extends Component {
  render() {
    return (
      <View style={{alignItems: 'center', top: 50}}>
        <Greeting name='Rexxar' />
        <Greeting name='Jaina' />
        <Greeting name='Valeera' />
      </View>
    );
  }
}
```

```
}  
}
```

### Hasilnya



*Gambar- 27. Contoh Penggunaan Prop*

## BAB-4 HEIGHT DAN WIDTH PADA REACT NATIVE

### Tujuan Pembelajaran

- Peserta mampu memahami konsep height dan width
- Peserta mampu mengimplementasikan height dan width pada react native

Height dan width merupakan sebuah komponen pendukung dalam membuat tampilan visual sesuai dengan keinginan. Pada react native height dan width dikelompokkan dalam dua komponen yakni fixed dimensions dan flex dimensions.

### 6.1. Fixed Dimensions

Cara paling sederhana untuk mengatur dimensi komponen yakni dengan menambahkan width dan height. Semua dimensi pada react native bersifat unitless, dan mewakili kepadatan pixel.

Untuk contoh fixed dimension, silahkan edit view style yang terdapat pada file index.android.js . Disini saya masih melanjutkan baris kode pada tutorial yang pertama. Sekarang kita masukkan perintah berikut didalam <View style={styles.container}>.

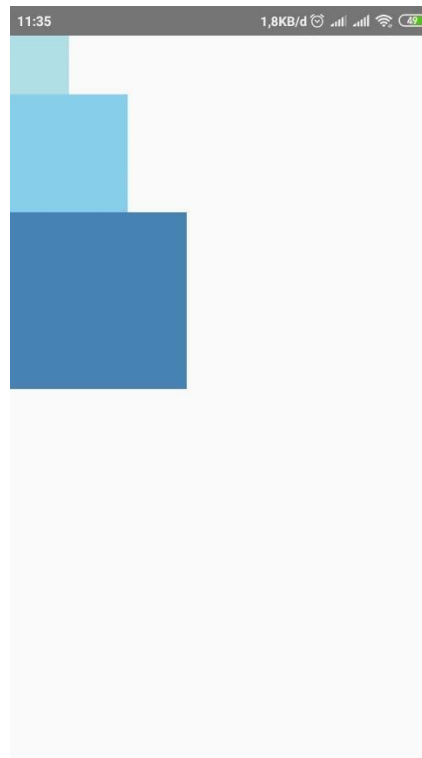
#### Contoh penggunaan

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

export default class FixedDimensionsBasics extends Component {
  render() {
    return (
      <View>
        <View style={{width: 50, height: 50, backgroundColor: 'powderblue'}} />
        <View style={{width: 100, height: 100, backgroundColor: 'skyblue'}} />
        <View style={{width: 150, height: 150, backgroundColor: 'steelblue'}} />
      </View>
    );
  }
}
```

```
}
```

**Hasilnya**



*Gambar- 28. Fixed Dimensions*

## 6.2. Flex Dimensions

Flex dimension digunakan untuk memperluas dan mengecilkan komponen secara dinamis biasanya kita akan menggunakan flex: 1, yang memberi tahu komponen untuk mengisi semua ruang yang tersedia, dibagi rata di antara komponen lainnya dengan ukuran seluruh ruang yang tersedia. Semakin besar flex yang diberikan, semakin tinggi rasio ruang komponen akan dibandingkan dengan flex: 1. Untuk lebih jelas lagi mari simak contoh berikut ini.

**App.js**

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

```
export default class FlexDimensionsBasics extends Component {  
  render() {  
    return (  
      // Try removing the `flex: 1` on the parent View.  
      // The parent will not have dimensions, so the children can't expand.  
      // What if you add `height: 300` instead of `flex: 1`?  
      <View style={{flex: 1}}>  
        <View style={{flex: 1, backgroundColor: 'powderblue'}} />  
        <View style={{flex: 2, backgroundColor: 'skyblue'}} />  
        <View style={{flex: 3, backgroundColor: 'steelblue'}} />  
      </View>  
    );  
  }  
}
```

**Hasilnya**



*Gambar- 29. Flex Dimensions*

## BAB-5 STYLE PADA REACT NATIVE

### Tujuan Pembelajaran:

- Peserta mengetahui cara styling pada react native
- Peserta mampu mengimplementasikan style pada react native

Pada react native styling menggunakan Bahasa Javascript. Semua komponen menerima property bernama style. Nama style dan valuenya bekerja sama seperti CSS pada web. Akan tetapi styling pada react native ini ditulis dengan cara camel case seperti backgroundColor tidak seperti CSS yang ditulis background-color.

Style prop dapat digunakan untuk menyederhanakan object javascript yang lama. Itulah yang paling sederhana dan biasanya kita gunakan sebagai contoh code. Kita juga dapat memberikan styling pada array, array terakhirlah yang akan menjadi prioritas. Jadi kita bisa menggunakannya untuk inherit style.

Saat komponen bertambah kompleks, seringkali lebih mudah menggunakan StyleSheet. Dibuat untuk mendefinisikan beberapa style di satu tempat. Berikut ini contoh style secara sederhana:

### App.js

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

const styles = StyleSheet.create({
  bigBlue: {
    color: 'blue',
    fontWeight: 'bold',
    fontSize: 30,
  },
  red: {
    color: 'red',
```

```
    },  
  });  
  
export default class LotsOfStyles extends Component {  
  render() {  
    return (  
      <View>  
        <Text style={styles.red}>just red</Text>  
        <Text style={styles.bigBlue}>just bigBlue</Text>  
        <Text style={[styles.bigBlue, styles.red]}>bigBlue, then red</Text>  
        <Text style={[styles.red, styles.bigBlue]}>red, then bigBlue</Text>  
      </View>  
    );  
  }  
}
```

### Hasilnya



Gambar- 30. Style React Native

## BAB-6 LAYOUT MENGGUNAKAN FLEXBOX

Tujuan Pembelajaran:

- Peserta mengetahui macam macam property pada flexbox
- Peserta memahami cara layouting
- Peserta mampu mengimplementasikan layouting menggunakan flexbox

Untuk membuat tampilan sesuai yang di inginkan flexbox menyediakan property utama seperti pada table berikut ini.

Property	Value	Deskripsi
flexDirection	'row', 'column', 'row-reverse', 'column-reverse'	Menambahkan flexDirection pada style komponen untuk menentukan poros utama tata letaknya
justifyContent	'flex-start', 'flex-end', 'center', 'space-between', 'space-around', 'space-evenly'	Menambahkan justifyContent ke Style komponen digunakan untuk menentukan distribusi anak-anak di sepanjang sumbu utama. Haruskah anak dibagikan di awal, pusat, akhir, atau spasi secara merata.
alignItems	'stretch', 'flex-start', 'flex-end', 'center', 'baseline'	Menambahkan alignItems ke Style komponen digunakan untuk menentukan keselarasan anak-anak di sepanjang sumbu sekunder (jika sumbu utama adalah baris, maka kolom sekunder adalah kolom, dan sebaliknya).

### 6.1. Flex Direction

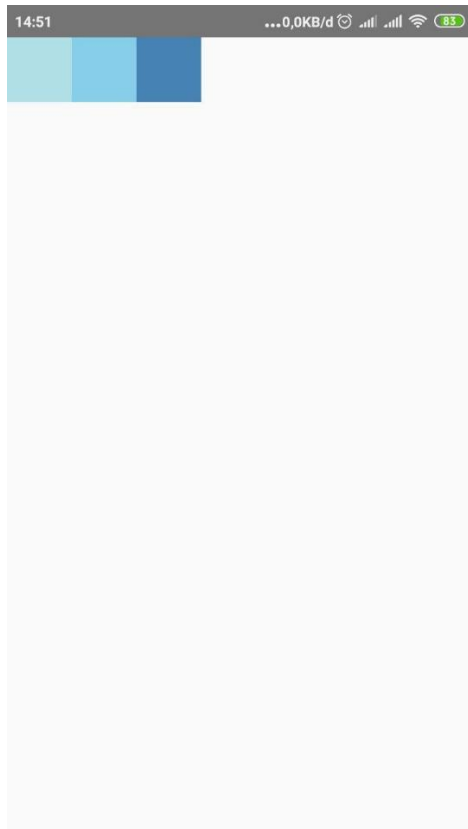
App.js

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

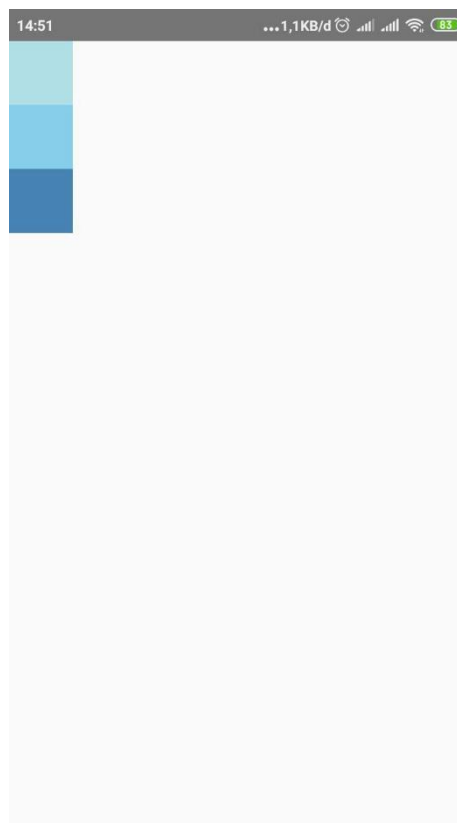


```
export default class FlexDirectionBasics extends Component {  
  render() {  
    return (  
      // Try setting `flexDirection` to `column`.  
      <View style={{flex: 1, flexDirection: 'row-reverse'}}>  
        <View style={{width: 50, height: 50, backgroundColor: 'powderblue'}} />  
        <View style={{width: 50, height: 50, backgroundColor: 'skyblue'}} />  
        <View style={{width: 50, height: 50, backgroundColor: 'steelblue'}} />  
      </View>  
    );  
  }  
};
```

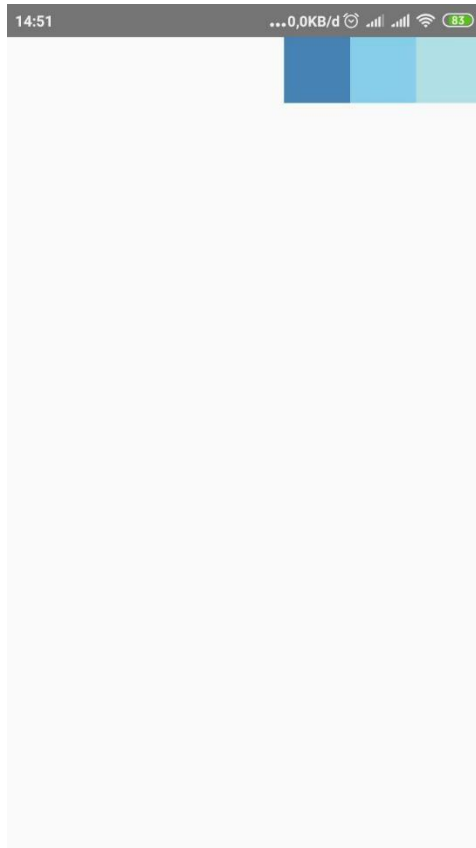
## Hasil



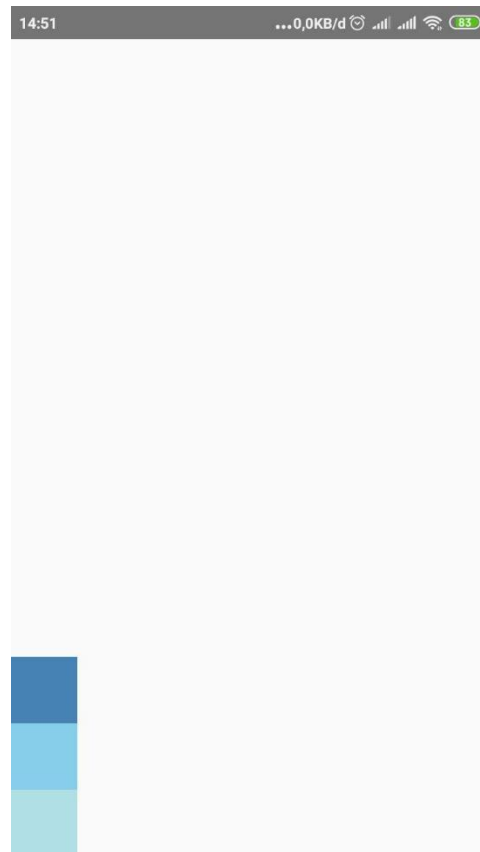
Gambar- 32. flexDirection Row



Gambar- 31. flexDirection Column



Gambar- 33. *flexDirection Row-Reverse*



Gambar- 34. *flexDirection Column-Reverse*

## 6.2. Justify Content

### App.js

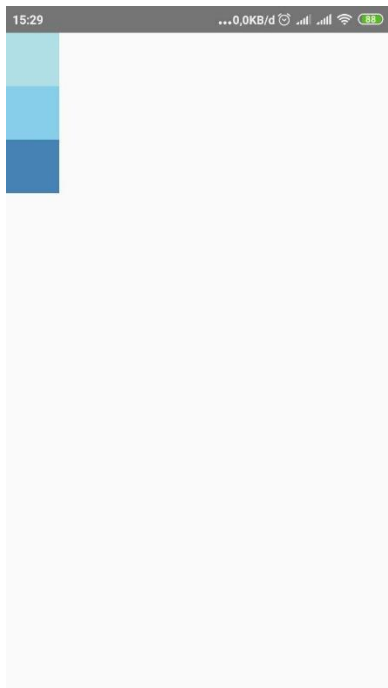
```
import React, { Component } from 'react';
import { View } from 'react-native';

export default class JustifyContentBasics extends Component {
  render() {
    return (
      // Try setting `justifyContent` to `center`.
      // Try setting `flexDirection` to `row`.
      <View style={{
        flex: 1,
```

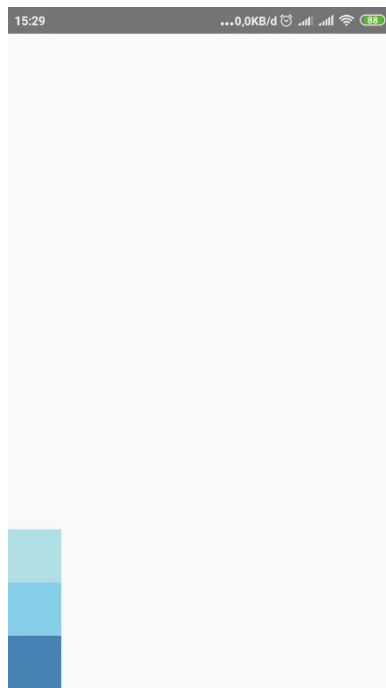
```
flexDirection: 'column',
justifyContent: 'flex-start',
}}>

<View style={{width: 50, height: 50, backgroundColor: 'powderblue'}} />
<View style={{width: 50, height: 50, backgroundColor: 'skyblue'}} />
<View style={{width: 50, height: 50, backgroundColor: 'steelblue'}} />
</View>
);
}
};
```

## Hasil



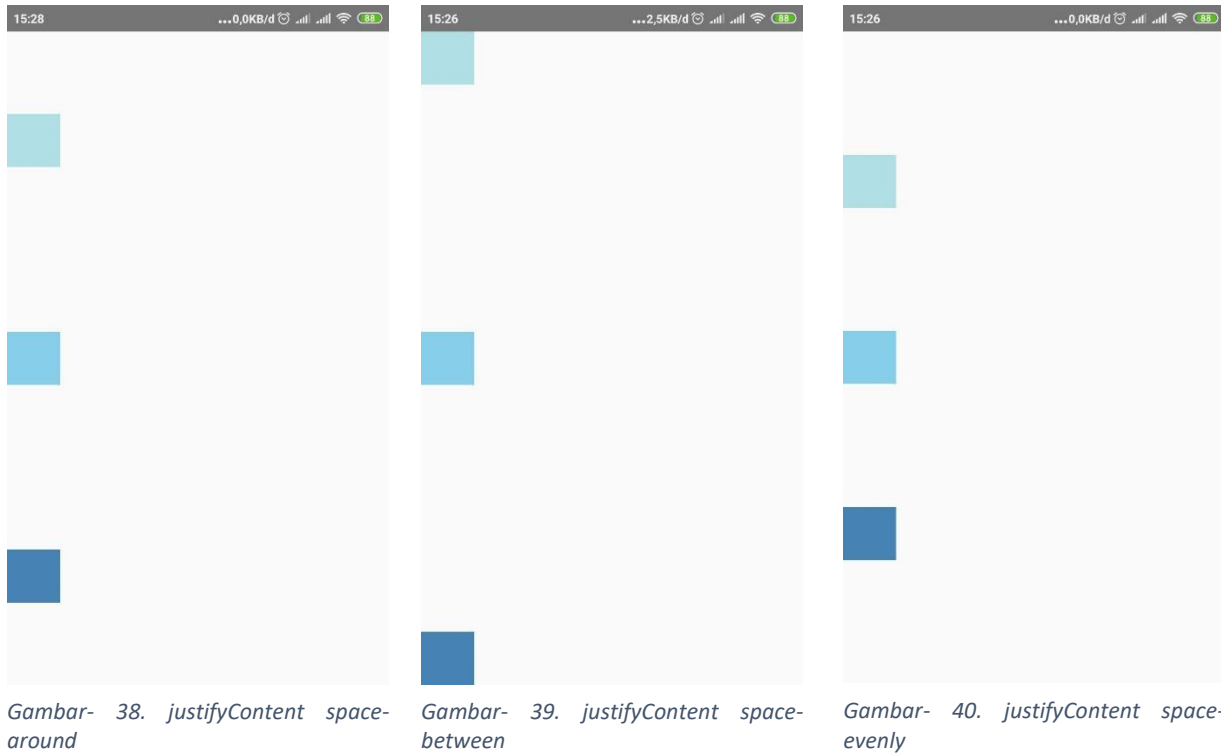
Gambar- 35. *justifyContent flex-start*



Gambar- 36. *justifyContent flex-end*



Gambar- 37. *justifyContent center*



### 6.3. Align Items

Contoh program app.js

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

export default class AlignItemsBasics extends Component {
  render() {
    return (
      // Try setting `alignItems` to 'flex-start'
      // Try setting `justifyContent` to 'flex-end`.
      // Try setting `flexDirection` to 'row`.
      <View style={{
        flex: 1,
        flexDirection: 'column',
```

```
      justifyContent: 'center',  
      alignItems: 'stretch',  
    }}>  
    <View style={{width: 50, height: 50, backgroundColor: 'powderblue'}} />  
    <View style={{width: 50, height: 50, backgroundColor: 'skyblue'}} />  
    <View style={{width: 50, height: 50, backgroundColor: 'steelblue'}} />  
  </View>  
);  
}  
};
```

## Hasil



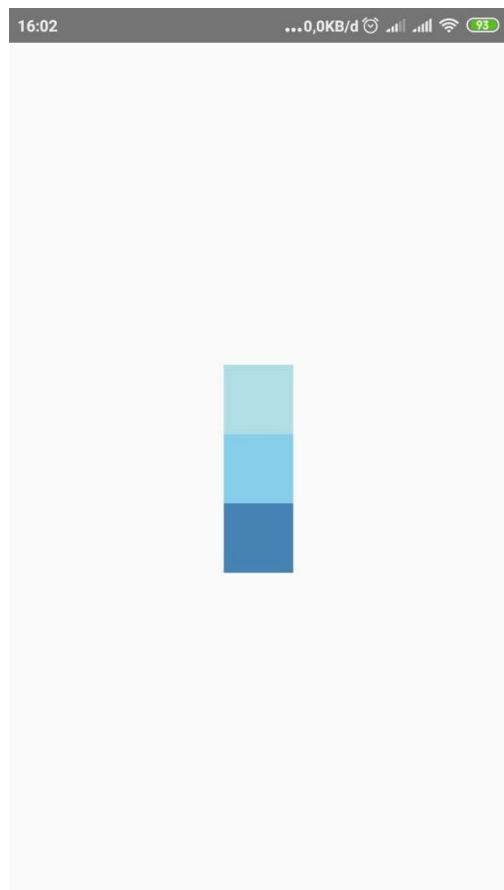
Gambar- 41. alignItems stretch



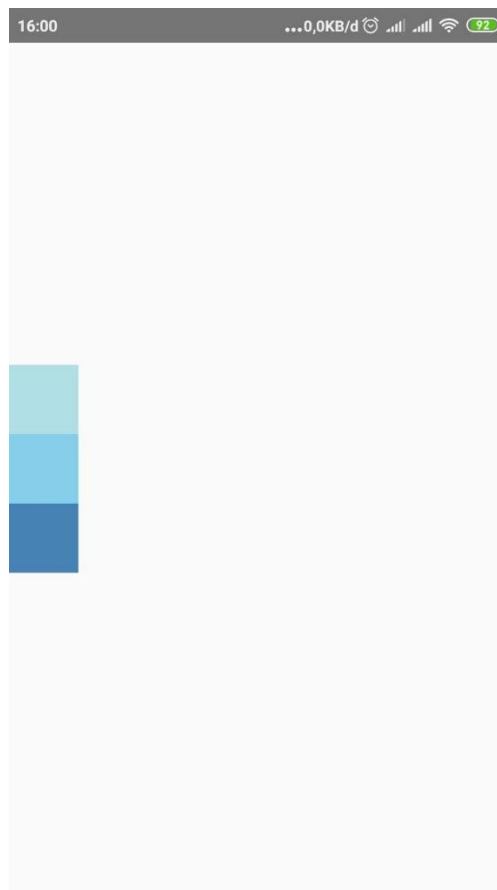
Gambar- 42. alignItems flex-start



Gambar- 43. alignItems flex-end



*Gambar- 44. alignItems Center*



*Gambar- 45. alignItems Baseline*

## BAB-7 HANDLING TEXT INPUT

Tujuan Pembelajaran:

- Peserta mengetahui apa itu text input pada React Native
- Peserta mengetahui cara penggunaan text input pada React Native
- Peserta mampu mengimplementasikan penggunaan text input pada React Native

### 7.1. Pengertian Text Input

Text input merupakan komponen dasar dalam input teks. Text input ini memiliki prop `onChangeText` yang mengambil sebuah fungsi untuk dipanggil setiap kali teks berubah, dan sebuah prop `onSubmitEditing` yang mengambil sebuah fungsi untuk dipanggil saat teks dikirimkan.

### 7.2. Properti pada Text Input

<code>allowFontScaling</code>	<code>autoCapitalize</code>	<code>autoCorrect</code>	<code>autoFocus</code>
<code>blurOnSubmit</code>	<code>caretHidden</code>	<code>clearButtonMode</code>	<code>clearTextOnFocus</code>
<code>contextMenuHidden</code>	<code>dataDetectorTypes</code>	<code>defaultValue</code>	<code>disableFullscreenUI</code>
<code>editable</code>	<code>enablesReturnKeyAutomatically</code>	<code>inlineImageLeft</code>	<code>inlineImagePadding</code>
<code>keyboardAppearance</code>	<code>keyboardType</code>	<code>maxLength</code>	<code>multiline</code>
<code>numberOfLines</code>	<code>onBlur</code>	<code>onChange</code>	<code>onChangeText</code>
<code>onContentSizeChange</code>	<code>onEndEditing</code>	<code>onFocus</code>	<code>onKeyPress</code>
<code>onLayout</code>	<code>onScroll</code>	<code>onSelectionChange</code>	<code>onSubmitEditing</code>
<code>placeholder</code>	<code>placeholderTextColor</code>	<code>returnKeyLabel</code>	<code>returnKeyType</code>
<code>scrollEnabled</code>	<code>secureTextEntry</code>	<code>selection</code>	<code>selectionColor</code>
<code>selectionColor</code>	<code>selectionState</code>	<code>selectTextOnFocus</code>	<code>spellCheck</code>
<code>textContentType</code>	<code>style</code>	<code>textBreakStrategy</code>	<code>underlineColorAndroid</code>

## 7.3. Contoh Penggunaan Text Input

### 7.3.1. Konfigurasi pada App.js

Pada app.js merupakan home yang akan memanggil fungsi yang ada pada halaman input.

#### App.js

```
import React from 'react';
import Inputs from './inputs.js'

const App = () => {
  return (
    <Inputs />
  )
}
export default App
```

### 7.3.2. Halaman Input.js

Kita akan menuliskan inialisai state pada file ini. Setelah mendefinisikan inialisasi state, kita akan membuat fungsi handleEmail dan akan membuat fungsi handlePassword. Kedua fungsi ini akan berfungsi untuk updateState.

Fungsi login akan menampilkan email dan password yang telah di input menggunakan alert. Disini juga ditambahkan beberapa property seperti disable auto capitalization, remove bottom border pada android dan set placeholder.

#### Input.js

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

class Inputs extends Component {
  state = {
    email: "",
```



```
password: ""
}

handleEmail = (text) => {
  this.setState({ email: text })
}

handlePassword = (text) => {
  this.setState({ password: text })
}

login = (email, pass) => {
  alert('email: ' + email + ' password: ' + pass)
}

render() {
  return (
    <View style = {styles.container}>
      <TextInput style = {styles.input}
        underlineColorAndroid = "transparent"
        placeholder = "Email"
        placeholderTextColor = "#9a73ef"
        autoCapitalize = "none"
        onChangeText = {this.handleEmail}/>

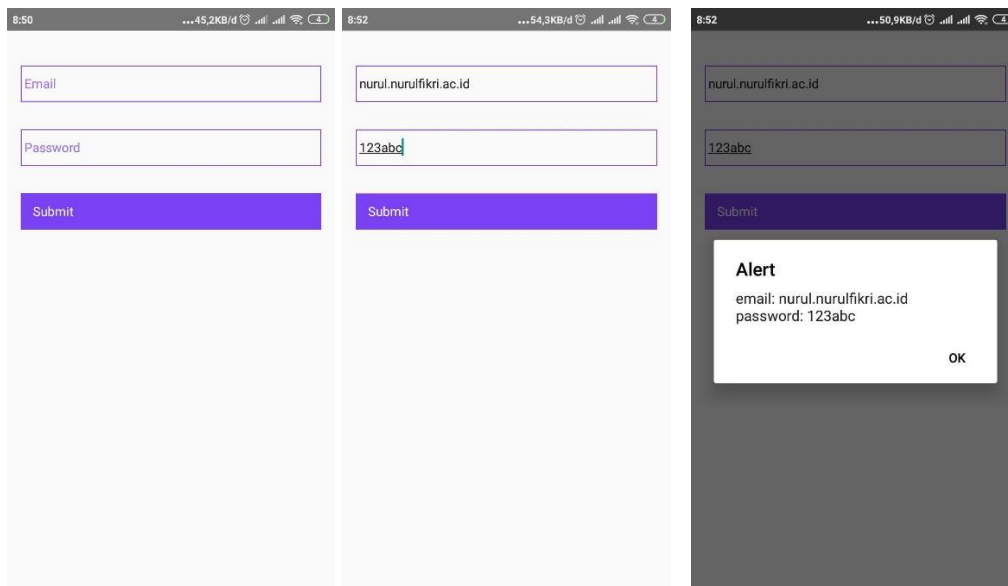
      <TextInput style = {styles.input}
        underlineColorAndroid = "transparent"
        placeholder = "Password"
        placeholderTextColor = "#9a73ef"
        autoCapitalize = "none"
        onChangeText = {this.handlePassword}/>
    </View>
  )
}
```

```
<TouchableOpacity
  style = {styles.submitButton}
  onPress = {
    () => this.login(this.state.email, this.state.password)
  }>
  <Text style = {styles.submitButtonText}> Submit </Text>
</TouchableOpacity>
</View>
)
}
}
export default Inputs

const styles = StyleSheet.create({
  container: {
    paddingTop: 23
  },
  input: {
    margin: 15,
    height: 40,
    borderColor: '#7a42f4',
    borderWidth: 1
  },
  submitButton: {
    backgroundColor: '#7a42f4',
    padding: 10,
    margin: 15,
    height: 40,
```

```
},  
submitButtonText:{  
  color: 'white'  
}  
})
```

### 7.3.3. Tampilan pada Android



Gambar- 46.Tampilan Text Input

## BAB-8 HANDLING TOUCHES

### Tujuan Pembelajaran

- Peserta mengetahui macam macam sentuhan(touches) pada React Naative
- Peserta mengetahui property(prop) yang menangani jenis sentuhan
- Peserta mampu menggunakan touches pada aplikasi react native

Pengguna berinteraksi dengan aplikasi mobile melalui sentuhan. Mereka dapat mengkombinasikan antar Gerakan seperti, mengetuk tombol, mengulir daftar, atau memperbesar peta.

React native menyediakan semua komponen yang dapat menangani semua semua Jenis sentuhan. serta sistem respons gerakan komprehensif untuk memungkinkan pengenalan gerakan lebih lanjut, tetapi satu komponen yang kemungkinan besar akan Anda minati adalah Tombol dasar.

### 8.1. Menampilkan Basic Button

Button merupakan komponen dasar dari react native yang harus ditampilkan secara baik pada platform apapun.

Berikut ini contoh program untuk basic button.

#### App.js

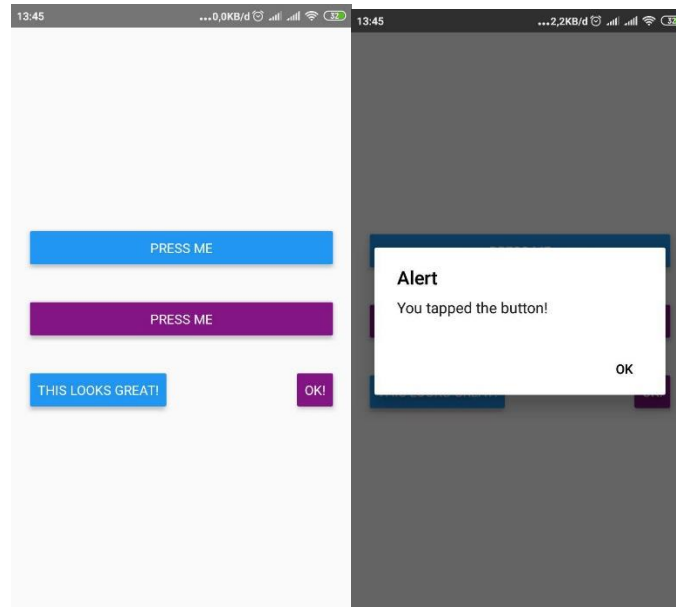
```
import React, { Component } from 'react';
import { Button, StyleSheet, View } from 'react-native';

export default class ButtonBasics extends Component {
  _onPressButton() {
    alert('You tapped the button!')
  }
}
```

```
render() {  
  return (  
    <View style={styles.container}>  
      <View style={styles.buttonContainer}>  
        <Button  
          onPress={this._onPressButton}  
          title="Press Me"  
        />  
      </View>  
      <View style={styles.buttonContainer}>  
        <Button  
          onPress={this._onPressButton}  
          title="Press Me"  
          color="#841584"  
        />  
      </View>  
      <View style={styles.alternativeLayoutButtonContainer}>  
        <Button  
          onPress={this._onPressButton}  
          title="This looks great!"  
        />  
        <Button  
          onPress={this._onPressButton}  
          title="OK!"  
          color="#841584"  
        />  
      </View>  
    </View>  
  )  
}
```

```
    );  
  }  
}  
  
const styles = StyleSheet.create({  
  container: {  
    flex: 1,  
    justifyContent: 'center',  
  },  
  buttonContainer: {  
    margin: 20  
  },  
  alternativeLayoutButtonContainer: {  
    margin: 20,  
    flexDirection: 'row',  
    justifyContent: 'space-between'  
  }  
});
```

### Hasilnya



Gambar- 47. Basic Button

## 8.2. Jenis Komponen Touches

Jika tombol dasar tidak cocok untuk aplikasi Anda, Anda dapat membuat tombol sendiri menggunakan komponen "Touchable" yang disediakan oleh React Native. Komponen "Touchable" menyediakan kemampuan untuk menangkap gerakan mengetuk, dan dapat menampilkan umpan balik saat gerakan dikenali. Namun, komponen-komponen ini tidak memberikan gaya bawaan apa pun, jadi Anda harus melakukan sedikit pekerjaan untuk membuatnya terlihat bagus di aplikasi Anda.

Ada empat komponen yang dapat disentuh yang disediakan oleh React Native. Pemilihan komponen ini tergantung pada jenis umpan balik yang ingin Anda berikan:

### 8.2.1. TouchableHighlight

Wrapper untuk membuat tampilan merespon sentuhan dengan benar. Saat ditekan ke bawah, opacity dari tampilan yang dibungkus berkurang, yang memungkinkan warna yang mendasari terlihat, gelap atau mewarnai tampilan.

Underlay berasal dari membungkus anak dalam Tampilan baru, yang dapat memengaruhi tata letak, dan terkadang menyebabkan artefak visual yang tidak diinginkan jika tidak digunakan dengan benar, misalnya jika warna latar belakang tampilan yang dibungkus tidak secara eksplisit diatur ke warna buram.

TouchableHighlight harus memiliki satu anak (bukan nol atau lebih dari satu). Jika Anda ingin memiliki beberapa komponen anak, bungkus dalam Tampilan.

Contoh

App.js

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

export default class App extends Component {
  constructor(props) {
    super(props)
    this.state = { count: 0 }
  }

  onPress = () => {
    this.setState({
      count: this.state.count+1
    })
  }

  render() {
    return (
      <View style={styles.container}>
        <TouchableHighlight
```

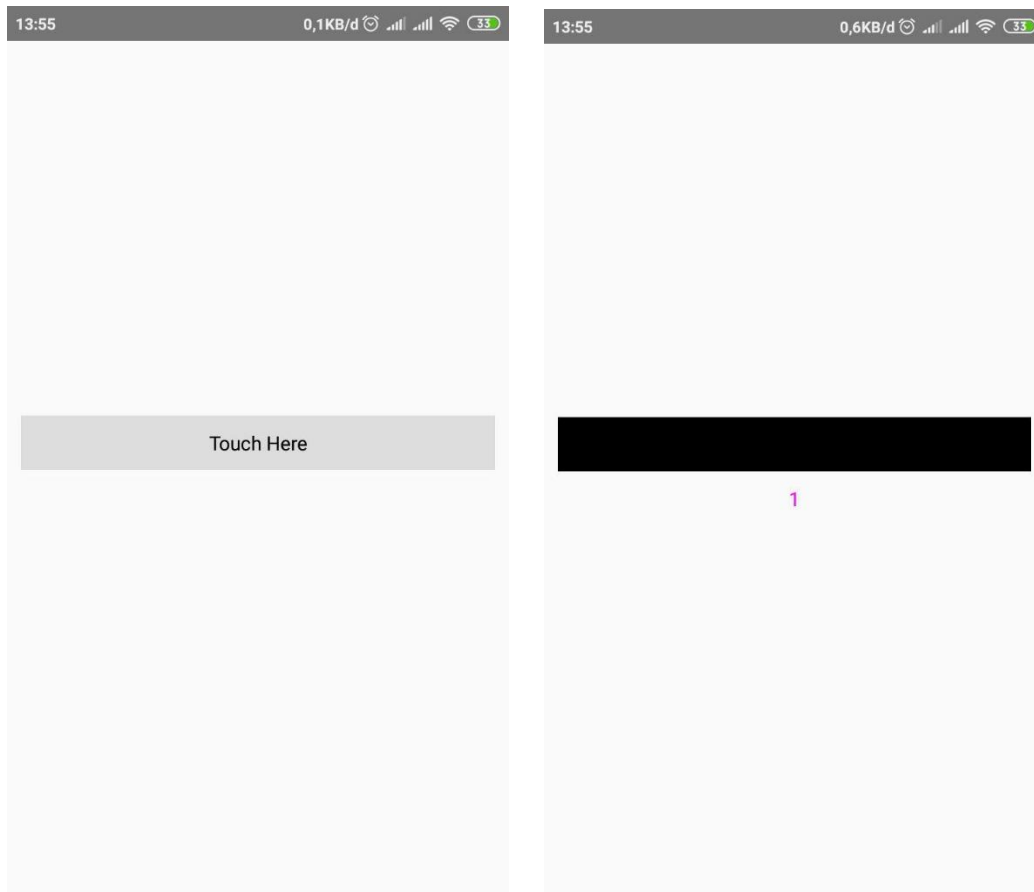


```
        style={styles.button}
        onPress={this.onPress}
      >
        <Text> Touch Here </Text>
      </TouchableHighlight>
    <View style={[styles.countContainer]}>
      <Text style={[styles.countText]}>
        { this.state.count !== 0 ? this.state.count: null}
      </Text>
    </View>
  </View>
)
}
}
```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    paddingHorizontal: 10
  },
  button: {
    alignItems: 'center',
    backgroundColor: '#DDDDDD',
    padding: 10
  },
  countContainer: {
    alignItems: 'center',
```

```
padding: 10
},
countText: {
  color: '#FF00FF'
}
})
```

## Hasil



Gambar- 48. touchableHighlight

**Props**

Props	Type	Require	Platform	Description
<b>activeOpacity</b>	Number	No		Menentukan opacity tampilan yang dibungkus ketika disentuh.
<b>OnHideUnderlay</b>	Fungsi	No		Panggilan langsung setelah underlay disembunyikan.
<b>onShowUnderlay</b>	Fungsi	No		Panggilan langsung setelah underlay ditampilkan.
<b>Style</b>	View.style	No		
<b>underlayColor</b>	Warna	No		Tampilkan warna underlay ketika sentuhan aktif.
<b>hasTVPreferredFocus</b>	Bool	No	IOS	Ini memfokuskan TV yang disukai, hanya berfungsi untuk iOS.
<b>TvParallaxProperties</b>	Objek	No	IOS	objek dengan properti yang mengontrol efek parallax Apple TV.

**8.2.2. TouchableNativeFeedback**

Wrapper untuk membuat tampilan merespons sentuhan dengan benar (hanya Android). Pada Android, komponen ini menggunakan native state yang dapat digambar untuk menampilkan umpan balik sentuhan.

Saat ini hanya mendukung memiliki instance Lihat tunggal sebagai simpul anak, karena diterapkan dengan mengganti Lihat itu dengan instance lain dari simpul RCTView dengan beberapa properti tambahan yang ditetapkan.

Latar belakang yang dapat digambar dari umpan balik asli yang dapat disentuh dapat disesuaikan dengan properti latar belakang.

Contoh :

```
import React, { Component } from 'react'
import {
```

```
StyleSheet,
TouchableNativeFeedback,
Text,
View,
} from 'react-native'

export default class App extends Component {

  render() {
    return (
      <View style={styles.container}>
        <TouchableNativeFeedback
          onPress={this._onPressButton}
          background={TouchableNativeFeedback.SelectableBackground()}>
          <View style={{width: 150, height: 100, backgroundColor: 'red'}}>
            <Text style={{margin: 30}}>Button</Text>
          </View>
        </TouchableNativeFeedback>

        </View>
      )
    }
  }

  const styles = StyleSheet.create({
    container: {
      flex: 1,
      justifyContent: 'center',
```

```

    alignItems: 'center',
    paddingHorizontal: 10
  }
})

```

### Props

Props	Type	Require	Platform	Description
<b>background</b>	backgroundPropType	No		menentukan latar belakang yang dapat digambar yang akan ditampilkan sebagai umpan balik.
<b>useForeground</b>	bool	No		menambahkan efek riak ke latar depan tampilan, bukan latar belakang.

### 8.2.3. TouchableOpacity

TouchableOpacity digunakan untuk mengurangi opacity tombol. Ini memungkinkan latar belakang terlihat saat pengguna menekan. Keburaman tombol akan dikontrol dengan membungkus anak-anak dalam Animasi.

### Props

Props	Type	Require	Platform	Description
<b>activeOpacity</b>	number	No		menentukan opacity dari tampilan yang dibungkus ketika disentuh.
<b>tvParallaxProperties</b>	objek	No	iOS	objek dengan properti yang digunakan untuk mengontrol efek paralaks Apple TV.

<b>hasTVPreferredFocus</b>	Bool	No	iOS	memfokuskan TV yang disukai, hanya bekerja di Apple TV.
----------------------------	------	----	-----	---

### Method

Method	Deskripsi
<b>setOpacityTo()</b>	menggerakkan layar sentuh ke opacity baru.

### Contoh

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

export default class App extends Component {
  constructor(props) {
    super(props)
    this.state = { count: 0 }
  }

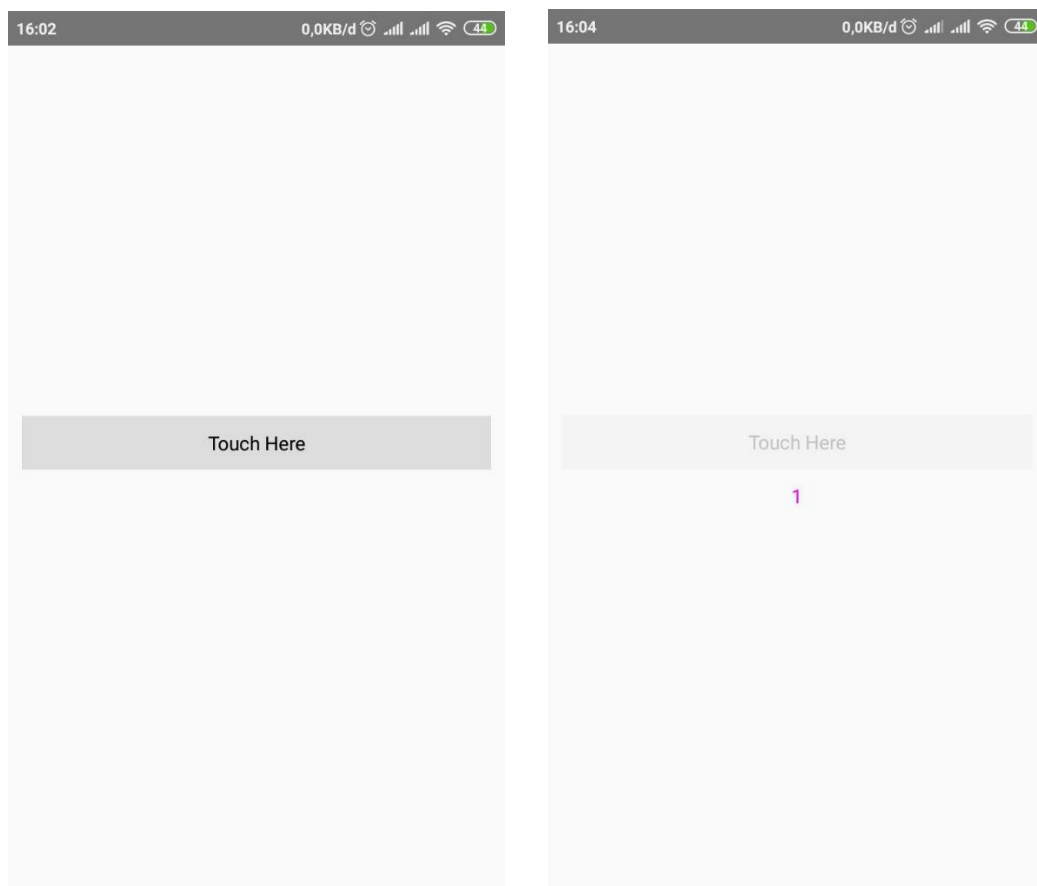
  onPress = () => {
    this.setState({
      count: this.state.count+1
    })
  }
}
```

```
render() {
  return (
    <View style={styles.container}>
      <TouchableOpacity
        style={styles.button}
        onPress={this.onPress}
      >
        <Text> Touch Here </Text>
      </TouchableOpacity>
      <View style={[styles.countContainer]}>
        <Text style={[styles.countText]}>
          { this.state.count !== 0 ? this.state.count: null}
        </Text>
      </View>
    </View>
  )
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    paddingHorizontal: 10
  },
  button: {
    alignItems: 'center',
    backgroundColor: '#DDDDDD',
```

```
padding: 10
},
countContainer: {
  alignItems: 'center',
  padding: 10
},
countText: {
  color: '#FF00FF'
}
})
```

Hasil



Gambar- 49. TouchableOpacity



### 8.2.4. TouchableWithoutFeedback

`TouchableWithoutFeedback` digunakan ketika ingin menangani fungsionalitas tap tetapi tidak ingin menampilkan umpan balik.

`TouchableWithoutFeedback` hanya mendukung satu anak. Jika Anda ingin memiliki beberapa komponen anak, bungkus dalam `Tampilan`. Yang penting, `TouchableWithoutFeedback` bekerja dengan mengkloning anaknya dan menerapkan alat peraga responden untuknya. Oleh karena itu diperlukan bahwa setiap komponen perantara melewati alat peraga tersebut ke komponen Penduduk Asli yang mendasarinya.

#### Props

Props	Type	Require	Description
<b>hitSlop</b>	objek	No	menentukan opacity dari tampilan yang dibungkus ketika disentuh.
<b>onAccessibilityTap</b>	fungsi	No	Jika diakses diatur ke true, sistem memanggil fungsi ini ketika pengguna melakukan gerakan tap aksesibilitas.
<b>accessibilityHint</b>	string	No	membantu pengguna untuk memahami apa yang akan terjadi ketika mereka melakukan tindakan pada elemen aksesibilitas.
<b>AccessibilityLabel</b>	node	No	Ini menimpa teks, yang dibaca oleh pembaca layar ketika pengguna berinteraksi dengan elemen.
<b>DelayLongPress</b>	Number	No	menunda <code>onLongPress</code> dalam panggilan mili-detik <code>onPressIn</code> .

Beberapa waktu pengguna menekan tampilan dan menahannya untuk set waktu. menekan yang panjang ini ditangani oleh fungsi menggunakan prop `onLongPress` dari komponen "Touchable" di atas.

Contoh penggunaan

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

export default class App extends Component {

  _onPressButton() {
    alert('You tapped the button!')
  }

  _onLongPressButton() {
    alert('You long-pressed the button!')
  }

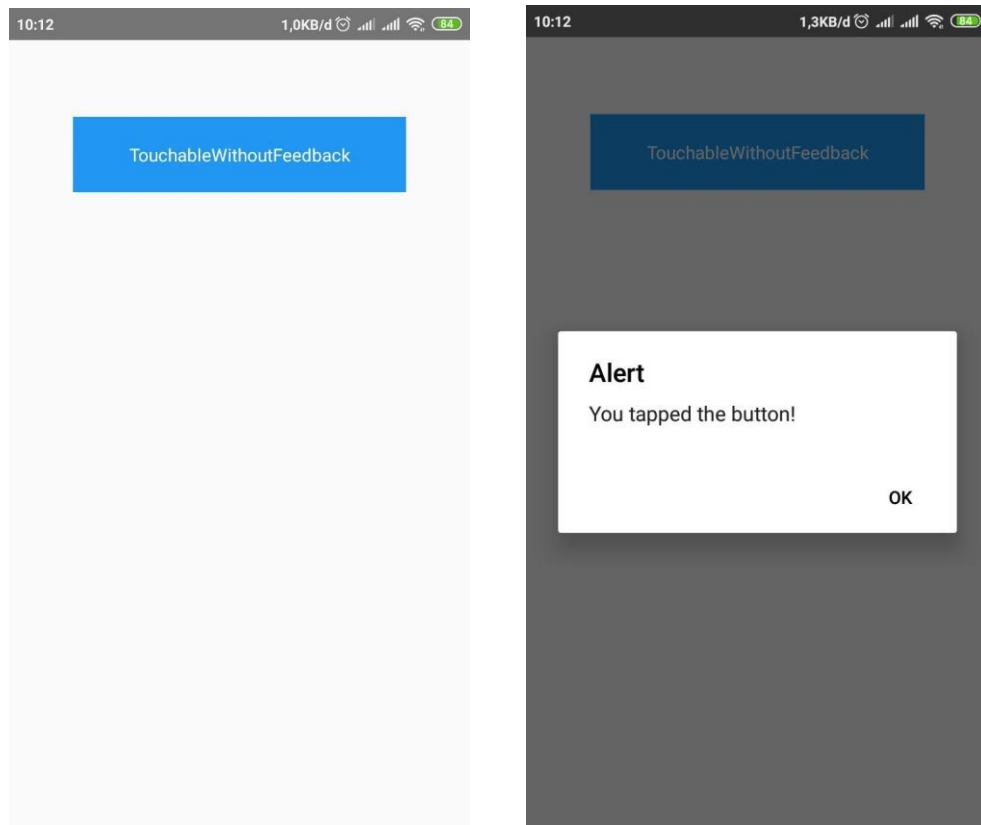
  render() {
    return (
      <View style={styles.container}>
        <TouchableWithoutFeedback
          onPress={this._onPressButton}

```

```
    >
    <View style={styles.button}>
      <Text style={styles.buttonText}>TouchableWithoutFeedback</Text>
    </View>
  </TouchableWithoutFeedback>
</View>
)
}
}

const styles = StyleSheet.create({
  container: {
    paddingTop: 60,
    alignItems: 'center'
  },
  button: {
    marginBottom: 30,
    width: 260,
    alignItems: 'center',
    backgroundColor: '#2196F3'
  },
  buttonText: {
    textAlign: 'center',
    padding: 20,
    color: 'white'
  }
})
```

## Hasil



Gambar- 50. touchableWithoutFeedback

## BAB-9 MENGGUNAKAN SCROLL VIEW

Tujuan Pembelajaran:

- Peserta mengenal scrollview pada react native
- Peserta mengetahui property yang mendukung element scrollview
- Peserta mampu mengimplementasikan element scrollview

### 9.1. Pengertian Scroll View

Scroll view merupakan container scrollable khusus. Yang mana digunakan untuk scroll beberapa komponen dalam satu tampilan. Scroll view kita dapat scroll komponen secara vertical dan secara horizontal.

### 9.2. Property Scroll View

alwaysBounceVertical	onScroll	horizontal	
contentContainerStyle	scrollEnabled	BouncesZoom	zoomScale
onScrollBeginDrag	onContentSizeChange	MaximumZoomScale	minimumZoomScale
OnScrollEndDrag	centerContent	contentInset	refreshControl
pagingEnabled	scrollsToTop	SnapToAlignment	showsHorizontalScrollIndicator
snapToStart	SnapToEnd	indicatorStyle	showsHorizontalScrollIndicator

### 9.3. Contoh Penggunaan Scroll View

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

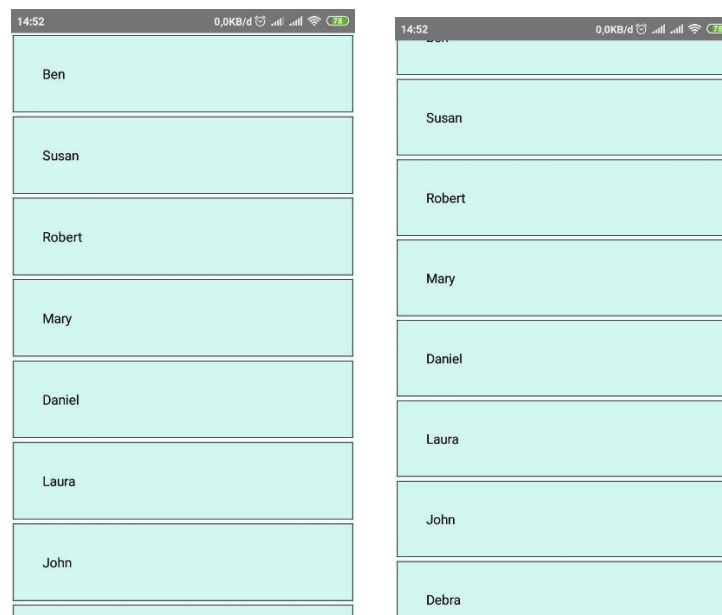
export default class ScrolledViewExample extends Component {
  state = {
    names: [
```

```
{'name': 'Ben', 'id': 1},
{'name': 'Susan', 'id': 2},
{'name': 'Robert', 'id': 3},
{'name': 'Mary', 'id': 4},
{'name': 'Daniel', 'id': 5},
{'name': 'Laura', 'id': 6},
{'name': 'John', 'id': 7},
{'name': 'Debra', 'id': 8},
{'name': 'Aron', 'id': 9},
{'name': 'Ann', 'id': 10},
{'name': 'Steve', 'id': 11},
{'name': 'Olivia', 'id': 12}
]
}

render() {
  return (
    <View>
      <ScrollView>
        {
          this.state.names.map((item, index) => (
            <View key = {item.id} style = {styles.item}>
              <Text>{item.name}</Text>
            </View>
          ))
        }
      </ScrollView>
    </View>
  )
}
```

```
}  
}  
const styles = StyleSheet.create ({  
  item: {  
    flexDirection: 'row',  
    justifyContent: 'space-between',  
    alignItems: 'center',  
    padding: 30,  
    margin: 2,  
    borderColor: '#2a4944',  
    borderWidth: 1,  
    backgroundColor: '#d2f7f1'  
  }  
})
```

## Hasil



Gambar- 51. Scroll View

## BAB-10 LIST VIEW

Tujuan Pembelajaran:

- Peserta mampu mengenal element list view pada react native
- Peserta mengetahui jenis jenis list view
- Peserta mampu menggunakan list view dalam aplikasi react native

React native menyediakan serangkaian komponen untuk menyajikan daftar data. React native memberikan dua pilihan penyajian list dengan FlatList atau SectionList.

### 10.1. Flat List

FlatList menampilkan data berupa daftar yang perubahannya dapat dilakukan dengan cara scrolling. FlatList mampu menampilkan daftar Panjang secara baik serta mampu beradaptasi ketika sewaktu waktu terjadi perubahan data. Berbeda dengan ScrollView yang lebih umum, FlatList hanya merender elemen yang saat ini ditampilkan di layar, tidak semua elemen sekaligus.

Komponen FlatList membutuhkan dua property(prop): data dan renderItem. data adalah sumber informasi untuk daftar. renderItem mengambil satu item dari sumber dan mengembalikan komponen yang diformat ke render.

Contoh ini membuat FlatList dasar dari data hardcode. Setiap item dalam prop data ditampilkan sebagai komponen Teks. Komponen FlatListBasics kemudian merender FlatList dan semua komponen Teks.

#### Contoh

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

export default class FlatListBasics extends Component {
  render() {
    return (
```

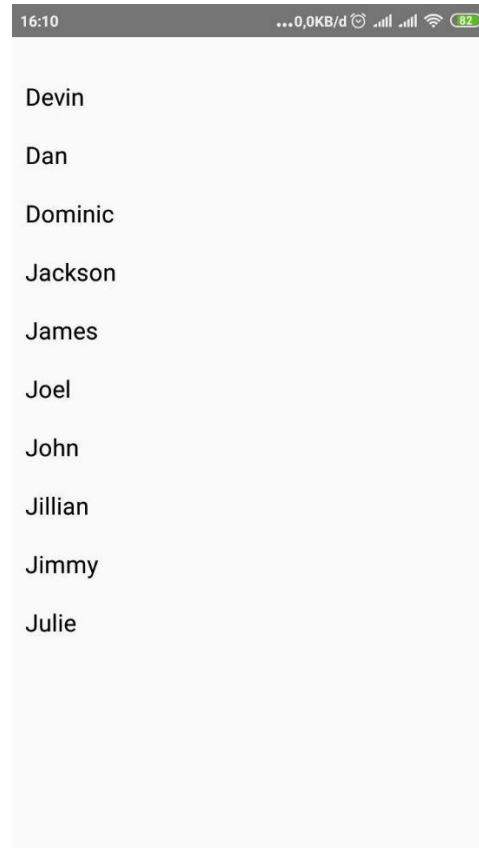


```
<View style={styles.container}>
  <FlatList
    data={[
      {key: 'Devin'},
      {key: 'Dan'},
      {key: 'Dominic'},
      {key: 'Jackson'},
      {key: 'James'},
      {key: 'Joel'},
      {key: 'John'},
      {key: 'Jillian'},
      {key: 'Jimmy'},
      {key: 'Julie'},
    ]}
    renderItem={({item}) => <Text style={styles.item}>{item.key}</Text>
  />
</View>
);
}
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    paddingTop: 22
  },
  item: {
    padding: 10,
```

```
fontSize: 18,  
height: 44,  
},  
})
```

Hasil



Gambar- 52. Flat List

## 10.2. Section List

Komponen React Native `SectionList` adalah komponen tampilan daftar yang menetapkan daftar data menjadi bagian logis yang rusak. Data yang rusak dapat diimplementasikan menggunakan prop header bagian `renderSectionHeader`.

Untuk mengimplementasikan komponen `SectionList`, kita perlu mengimpor `SectionList` dari pustaka `'react-native'`.

## Contoh

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

export default class SectionListBasics extends Component {
  render() {
    return (
      <View style={styles.container}>
        <SectionList
          sections={[
            {title: 'D', data: ['Devin', 'Dan', 'Dominic']},
            {title: 'J', data: ['Jackson', 'James', 'Jillian', 'Jimmy', 'Joel', 'John', 'Julie']},
          ]}
          renderItem={({item}) => <Text style={styles.item}>{item}</Text>}
          renderSectionHeader={({section}) => <Text
            style={styles.sectionHeader}>{section.title}</Text>}
          keyExtractor={(item, index) => index}
        />
      </View>
    );
  }
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    paddingTop: 22
  },
});
```

```
sectionHeader: {
  paddingTop: 2,
  paddingLeft: 10,
  paddingRight: 10,
  paddingBottom: 2,
  fontSize: 14,
  fontWeight: 'bold',
  backgroundColor: 'rgba(247,247,247,1.0)',
},
item: {
  padding: 10,
  fontSize: 18,
  height: 44,
},
})
```

## Hasil



Gambar- 53. Section List

## BAB-11 NETWORKING

Tujuan Pembelajaran:

- Peserta mengetahui prinsip prinsip networking dalam react native
- Peserta mampu mengenal prinsip post dan get data pada react native
- Peserta mampu mengimplementasikan pada aplikasi

Banyak aplikasi seluler perlu memuat sumber daya dari URL jarak jauh. Anda mungkin ingin membuat permintaan POST ke API REST, atau Anda mungkin perlu mengambil sepotong konten statis dari server lain.

### 11.1. Fetch

React Native menyediakan Fetch API untuk kebutuhan jaringan Anda. Fetch akan tampak familier jika Anda telah menggunakan XMLHttpRequest atau API jaringan lain sebelumnya. Anda dapat merujuk pada panduan MDN tentang Menggunakan Mengambil untuk informasi tambahan.

#### 11.1.1. Request Menggunakan Fetch

Kita dapat mengambil konten dari url web service dengan menuliskan perintah sebagai berikut:

```
fetch('https://mywebsite.com/mydata.json');
```

Kita juga dapat membuat perintah post request menggunakan fetch seperti berikut ini:

```
fetch('https://mywebsite.com/endpoint/', {  
  method: 'POST',  
  headers: {  
    Accept: 'application/json',  
    'Content-Type': 'application/json',  
  },  
  body: JSON.stringify({
```

```
firstParam: 'yourValue',
secondParam: 'yourOtherValue',
  },
});
```

### 11.1.2. Handling Response

Setelah kita tau cara penulisan request menggunakan fetch maka selanjutnya bagaimana handling responsenya?

Networking merupakan turunan operasi asynchronous. Method fetch akan mengembalikan nilai promise yang mana akan membuatnya mudah dalam penulisan kode yang bekerja secara asynchronous.

```
function getMoviesFromApiAsync() {
  return fetch('https://facebook.github.io/react-native/movies.json')
    .then((response) => response.json())
    .then((responseJson) => {
      return responseJson.movies;
    })
    .catch((error) => {
      console.error(error);
    });
}
```

Kita juga dapat menuliskan menggunakan sintak ES2017 sesuai saran dari react native seperti dibawah ini:

```
async function getMoviesFromApi() {
  try {
    let response = await fetch(
      'https://facebook.github.io/react-native/movies.json',
    );
```

```
let responseJson = await response.json();
return responseJson.movies;
} catch (error) {
  console.error(error);
}
}
```

### 11.2. XMLHttpRequest

API XMLHttpRequest dibuat ke dalam react native. artinya anda harus menggunakan library sebagai pihak ketiga seperti frisbee atau axios yang bergantung kepadanya. atau Anda dapat menggunakan API XMLHttpRequest secara langsung jika Anda mau.

```
var request = new XMLHttpRequest();
request.onreadystatechange = (e) => {
  if (request.readyState !== 4) {
    return;
  }

  if (request.status === 200) {
    console.log('success', request.responseText);
  } else {
    console.warn('error');
  }
};

request.open('GET', 'https://mywebsite.com/endpoint/');
request.send();
```

### 11.3. Web Socket

React Native juga mendukung WebSockets, protokol yang menyediakan saluran komunikasi dupleks penuh melalui koneksi TCP tunggal.

```
var ws = new WebSocket('ws://host.com/path');
```

```
ws.onopen = () => {  
  // connection opened  
  ws.send('something'); // send a message  
};
```

```
ws.onmessage = (e) => {  
  // a message was received  
  console.log(e.data);  
};
```

```
ws.onerror = (e) => {  
  // an error occurred  
  console.log(e.message);  
};
```

```
ws.onclose = (e) => {  
  // connection closed  
  console.log(e.code, e.reason);  
};
```



