



Front End React Development sesi 18



Introduction⁺ React Native

INTRO

Setelah materi-materi sebelumnya kita sudah belajar tentang React untuk membuat aplikasi web, sekarang kita akan belajar bagaimana menggunakan skill React kita untuk membangun aplikasi Mobile baik itu Android maupun iOS. Selain web dan mobile, React pun juga dapat kita gunakan untuk membangun aplikasi di platform seperti Virtual Reality, aplikasi terminal, aplikasi smart tv dan lain sebagainya.

React Native sendiri dapat membantu kita untuk membangun User Interface native dengan memanfaatkan komponen React. Dengan React Native, kita dapat membuat aplikasi mobile menggunakan JavaScript. React Native akan secara dinamis men-generate elemen-elemen native menggunakan JavaScript. Dengan demikian kita dapat memiliki aplikasi native dengan kecepatan dan tenaga dari sebuah aplikasi native, dengan kemudahan proses development selayaknya mengembangkan aplikasi web.

Itu artinya teman-teman yang sudah menguasai React dapat dengan mudah masuk ke dunia Mobile dengan berbekal kemampuan React.



Sesi 18 | Introduction React Native

INTRO

Secara umum React Native hampir sama dengan React untuk web. Mari kita lihat contoh kode berikut.

```
<div>
  <h1>Awesome Apps</h1>
  <form>
    <input type="text" />
    <button type="submit">Save</button>
  </form>
</div>
```



```
<View>
  <Text>Awesome Apps</Text>

  <TextInput />
  <TouchableHighlight>
    <Text>Save</Text>
  </TouchableHighlight>

</View>
```



Perbedaan yang tidak signifikan, bukan?!

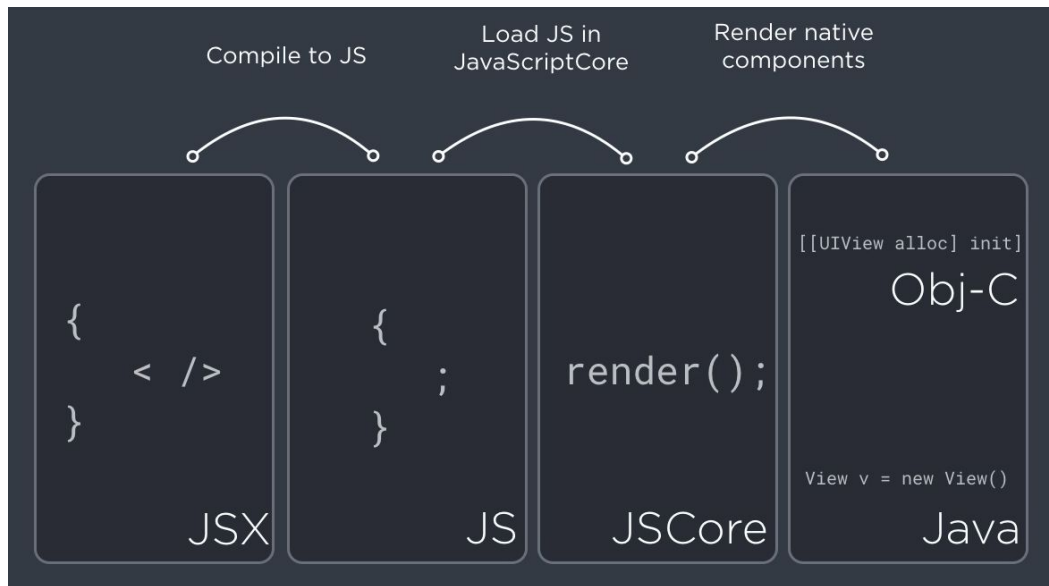


HACKTIV8

Sesi 18 | Introduction React Native

HOW REACT NATIVE WORKS

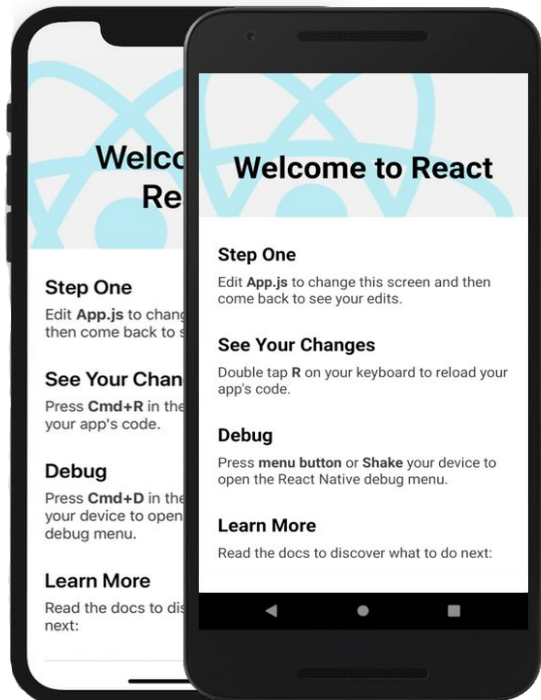
React Native melakukan transpilasi kode JSX yang kita tulis menjadi JavaScript murni sama halnya seperti yang terjadi di pengembangan aplikasi web. Kemudian dari kode JavaScript hasil transpilasi tersebut, JavaScript diterjemahkan menjadi elemen-elemen native dengan bantuan JavaScriptCore sehingga menjadi kode native baik itu Objective-C di iOS ataupun Java di Android.



Sesi 18 | Introduction React Native

HOW REACT NATIVE WORKS

Jadi, in the end, kita bisa membuat aplikasi iOS dan Android menggunakan 1 code base



android



HACKTIV8

DEVELOPMENT ENVIRONMENT SETUP

Untuk membuat React Native app, kita bisa gunakan 2 tool yaitu React Native CLI dan Expo.

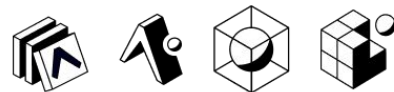
Jika kita membutuhkan native code pada React Native app yang kita buat, pilihan terbaik adalah React Native CLI. Tentu saja kita akan membutuhkan Android Studio atau Xcode. Jadi React Native CLI ini lebih banyak dipakai oleh developer yang sudah berpengalaman membuat mobile apps dengan native code seperti Java/Kotlin dan Swift.

Sementara Expo adalah sebuah SDK untuk mempermudah proses React Native app development. Kita tidak perlu install Android Studio maupun Xcode untuk membuat sebuah project. Kelemahan Expo adalah tidak bisa digabungkan dengan native code (Java/Kotlin/Swift).

Untuk keperluan belajar kita, kita akan menggunakan Expo. Silakan install Expo pada laptop kamu dengan perintah :

```
> npm install -g expo-cli
```

Expo



Sesi 18 | Introduction React Native

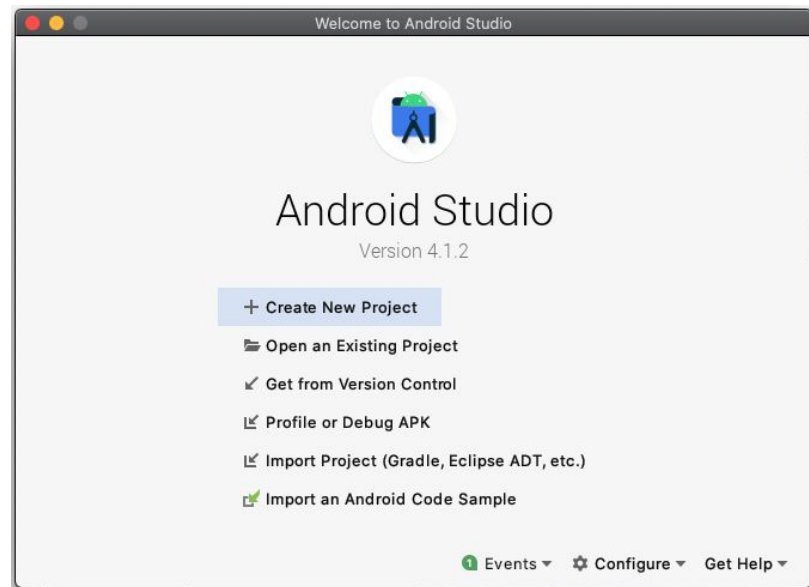
DEVELOPMENT ENVIRONMENT SETUP

Ada 3 pilihan **yang mudah** untuk kita dapat menjalankan / melihat hasil koding React Native kita : Android emulator, Android based phone, dan iOS based phone. Silakan pilih salah satu jenis instalasi berdasarkan media paling nyaman. Mari kita bahas cara instalasi masing-masing pilihan

ANDROID EMULATOR

Untuk menggunakan Android Emulator, kita harus download dan install Android Studio terlebih dahulu

Setelah itu, kita bisa melakukan instalasi AVD (Android Virtual Device) dari Android Studio. AVD ini akan bertindak sebagai emulator kita

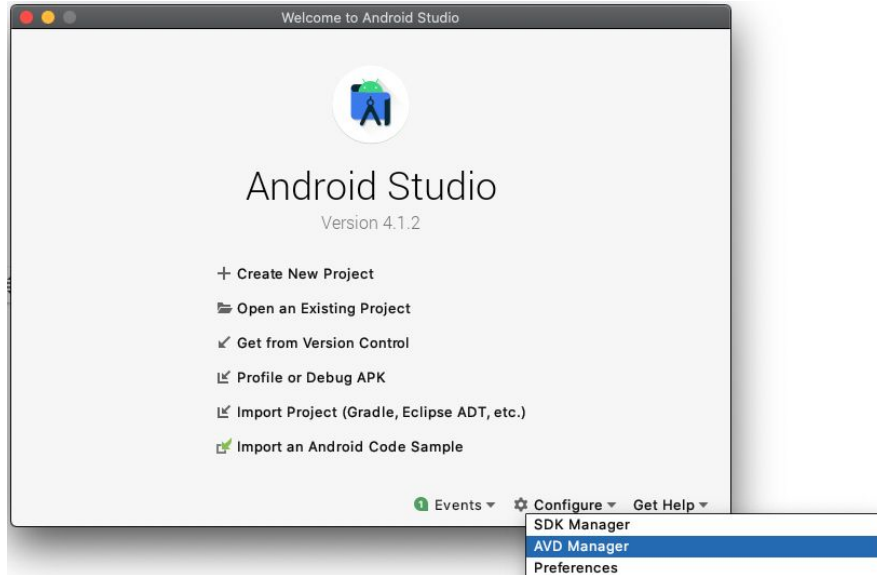


Sesi 18 | Introduction React Native

DEVELOPMENT ENVIRONMENT SETUP

1. Masuk AVD Manager

Dari Android Studio, buka menu Configure, kemudian pilih AVD Manager



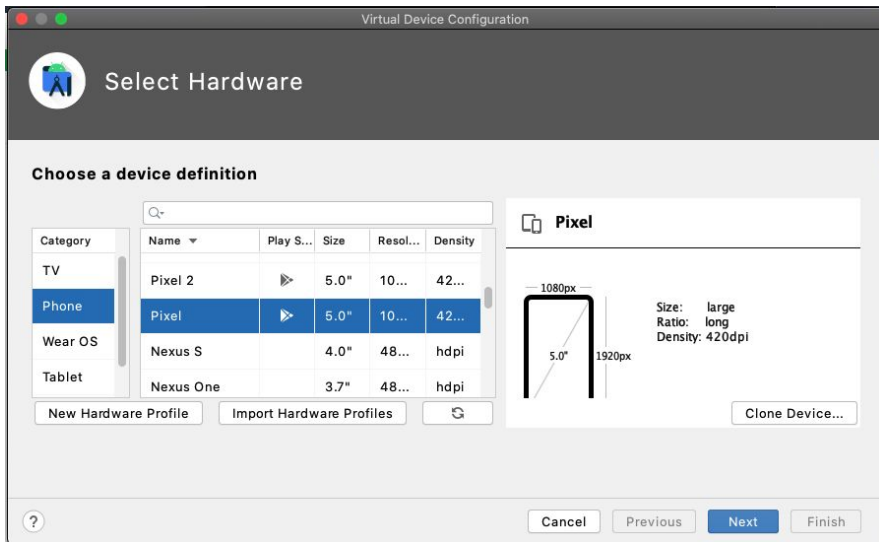
2. Pilih Create Virtual Device



DEVELOPMENT ENVIRONMENT SETUP

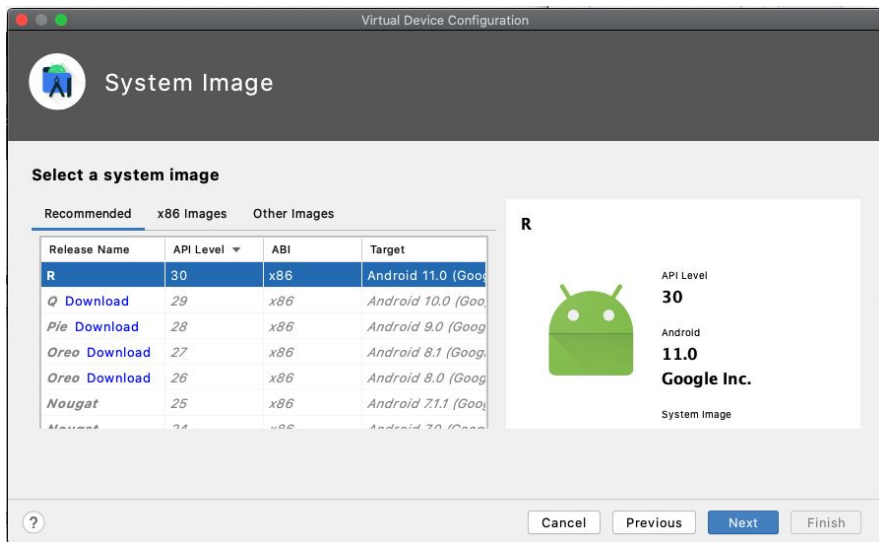
3. Pilih Device definition

Category : Phone, Device : Pixel (silakan pilih yang lain jika ingin), kemudian click Next



4. Pilih System Image

Disarankan untuk memilih Release dan API terbaru, kemudian click Next



Sesi 18 | Introduction React Native

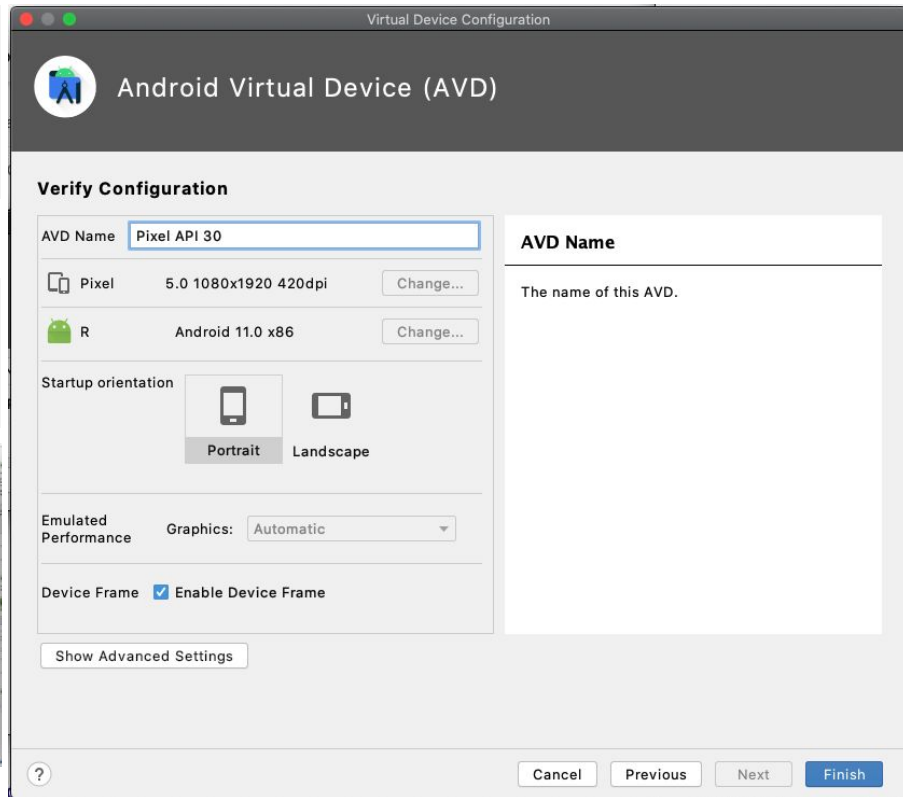
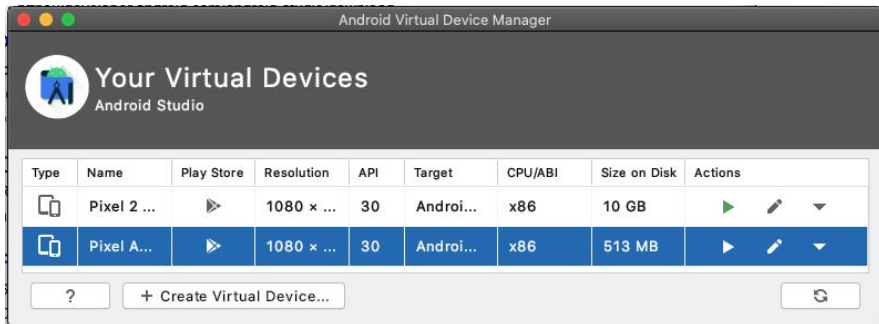
DEVELOPMENT ENVIRONMENT SETUP

5. Verify Configuration

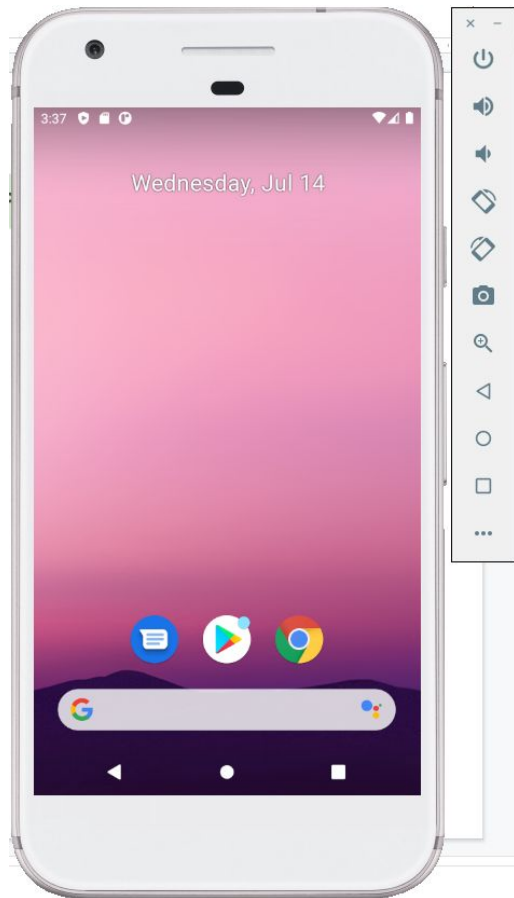
Kita bisa kasih nama untuk device kita. Kemudian click Finish

6. Emulator Ready

Click tombol play berwarna hijau untuk menjalankannya



Sesi 18 | Introduction React Native



DEVELOPMENT ENVIRONMENT SETUP

6. Emulator Ready !!

Selamat, emulator kita telah ter-install dan kita dapat segera menjalankan aplikasi React Native pertama kita.

Silakan explore menu-menu yang ada, dan mungkin bisa mempelajari dan eksplorasi dari menu-menu yang disediakan

Selanjutnya, mari kita setup aplikasi AwesomeProject kita dan kita run di emulator baru kita

DEVELOPMENT ENVIRONMENT SETUP



Kita sudah melakukan instalasi Expo pada laptop kita sebelumnya, mari kita initiate aplikasi React Native pertama kita

7. Initiate new project

```
> expo init AwesomeProject
```

8. Pilih template

Untuk kepentingan pembelajaran, kita akan pilih blank. Gunakan tombol panah untuk memilih, kemudian tekan Enter

```
~ expo init AwesomeProject
? Choose a template: > - Use arrow-keys. Return to submit.
  ----- Managed workflow -----
>  blank      a minimal app as clean as an empty canvas
    blank (TypeScript) same as blank but with TypeScript configuration
    tabs (TypeScript)  several example screens and tabs using react-navigation and TypeScript
  ----- Bare workflow -----
    minimal      bare and minimal, just the essentials to get you started
```



```
~ expo init AwesomeProject
✓ Choose a template: > blank      a minimal app as clean as an empty canvas
✓ Downloaded and extracted project files.
📦 Using npm to install packages.
✓ Installed JavaScript dependencies.

✓ Your project is ready!

To run your project, navigate to the directory and run one of the following npm commands.

- cd AwesomeProject
- npm start # you can open iOS, Android, or web from here, or run them directly with the commands below.
- npm run android
- npm run ios
- npm run web
```

Sesi 18 | Introduction React Native

DEVELOPMENT ENVIRONMENT SETUP

9. Run the app on terminal

```
> cd AwesomeProject  
> npm start
```

Di samping kanan adalah tampilan yang akan muncul pada terminal.

Maju ke slide selanjutnya untuk melihat tampilan website yang dimunculkan

```
~ npm start  
  
> start  
> expo start  
  
Starting project at /Users/figifitransyah/Documents/PTP/KampusMerdeka/18/  
Developer tools running on http://localhost:19002  
Opening developer tools in the browser...  
Starting Metro Bundler  
  
  
  
> Waiting on exp://192.168.1.108:19000  
> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)  
  
> Press a | open Android  
> Press i | open iOS simulator  
> Press w | open web  
  
> Press r | reload app  
> Press m | toggle menu  
> Press d | show developer tools  
> shift+d | toggle auto opening developer tools on startup (enabled)  
  
> Press ? | show all commands
```

Sesi 18 | Introduction React Native

DEVELOPMENT ENVIRONMENT SETUP

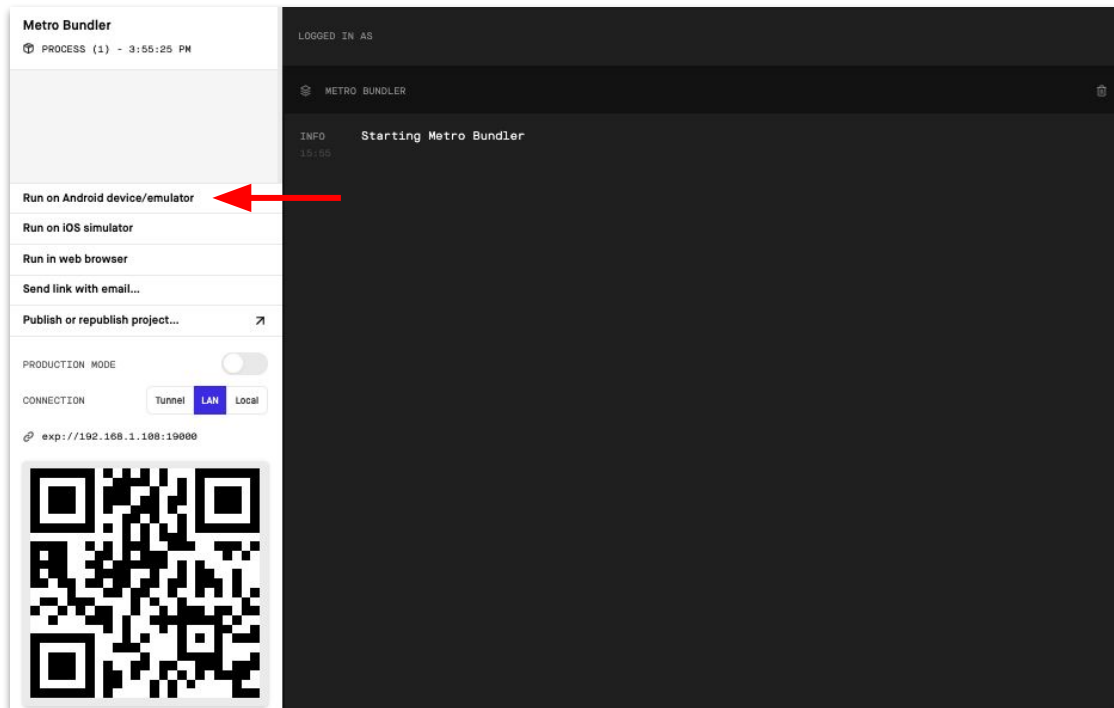
10. Web version

11. Open simulator

... dengan salah satu dari 2 cara ini :

ketik **a** di terminal ATAU klik menu **Run on Android device / emulator** dari Web

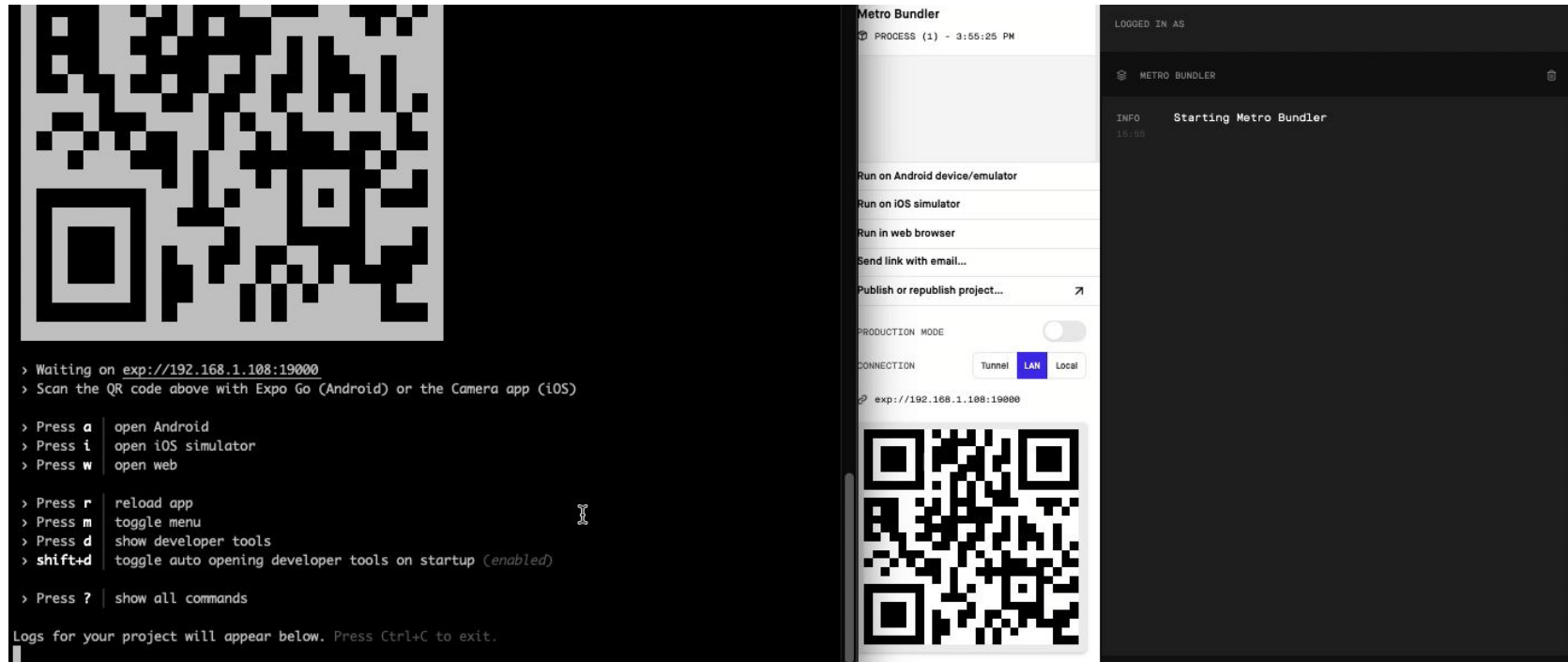
```
> Press a | open Android
> Press i | open iOS simulator
> Press w | open web
```



Sesi 18 | Introduction React Native

DEVELOPMENT ENVIRONMENT SETUP

12. Let it run... Akan ada serangkaian proses yang terjadi di terminal dan juga di web



The image displays the development environment setup for React Native, showing a terminal window, a Metro Bundler web interface, and a mobile app preview.

Terminal Window:

```
> Waiting on exp://192.168.1.108:19000
> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

> Press a | open Android
> Press i | open iOS simulator
> Press w | open web

> Press r | reload app
> Press m | toggle menu
> Press d | show developer tools
> Press shift+d | toggle auto opening developer tools on startup (enabled)

> Press ? | show all commands

Logs for your project will appear below. Press Ctrl+C to exit.
```

Metro Bundler Web Interface:

Metro Bundler
PROCESS (1) - 3:55:25 PM

Run on Android device/emulator
Run on iOS simulator
Run in web browser
Send link with email...
Publish or republish project...

PRODUCTION MODE ☐

CONNECTION Tunnel LAN Local

exp://192.168.1.108:19000

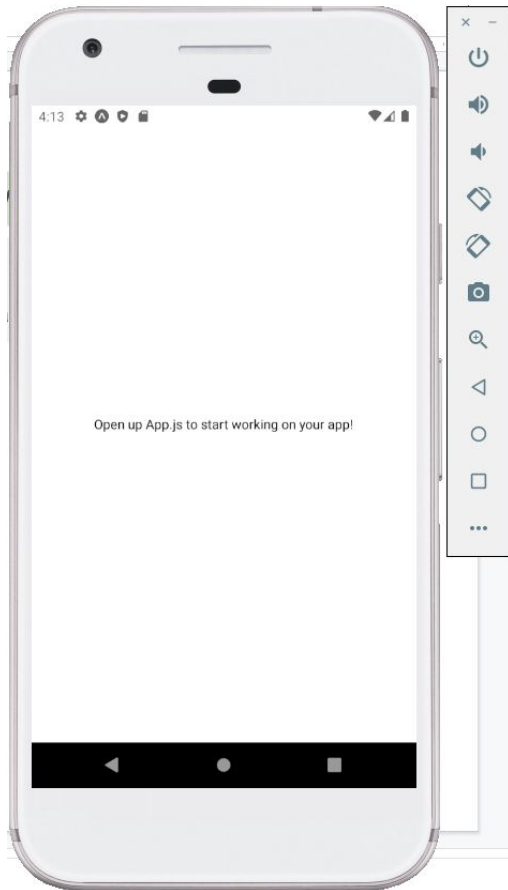
Mobile App Preview:

LOGGED IN AS

METRO BUNDLER

INFO Starting Metro Bundler
15:55

Sesi 18 | Introduction React Native



DEVELOPMENT ENVIRONMENT SETUP

13. Dan tentunya..

Yeay... Aplikasi kita muncul di Emulator yang telah kita generate. Perhatikan gambar di bawah yang menunjukkan status di web dan terminal

Untuk aktifitas code update kita setelah ini, jangan dulu dimatikan ya Emulator nya. Karena apa yang kita lakukan pada kode, akan Hot Reload secara realtime pada Emulator ini

Selanjutnya, kita akan bahas opsi menjalankan aplikasi dengan menggunakan gadget Android dan iOS

Metro Bundler

PROCESS (3) - 4:07:10 PM

sdk_gphone_x86

DEVICE (1) - 4:07:19 PM

```
Logs for your project will appear below. Press Ctrl+C to exit.
> Opening on Android...
> Opening exp://192.168.1.108:19000 on Pixel_API_30
> Press ? | show all commands
Finished building JavaScript bundle in 3322ms.
Running application on sdk_gphone_x86.
```



HACKTIV8

Sesi 18 | Introduction React Native

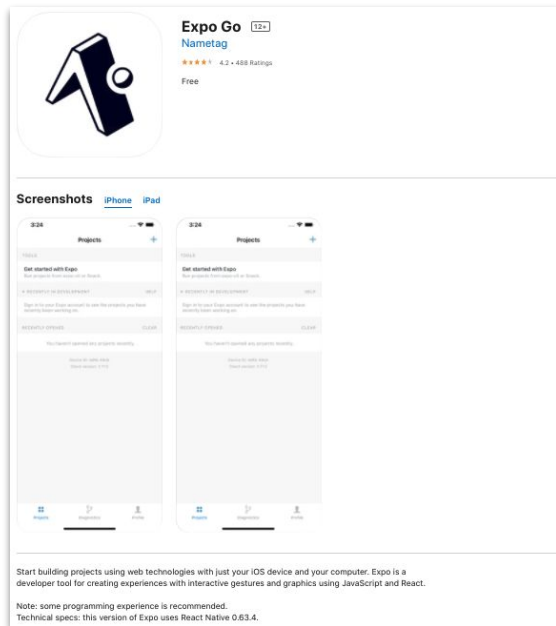
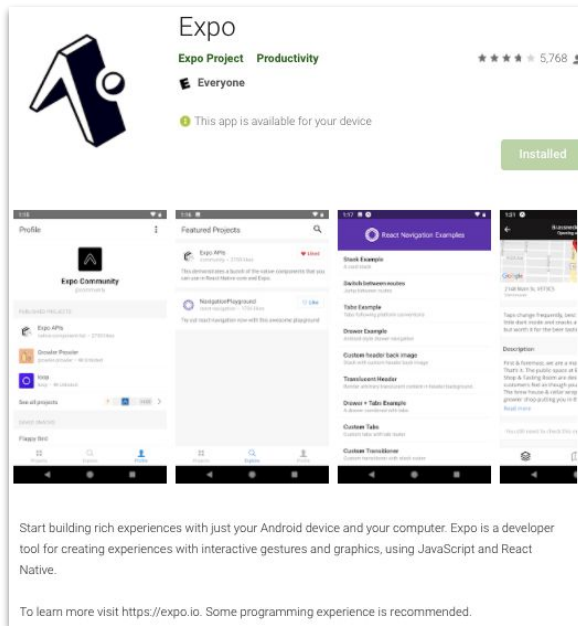
DEVELOPMENT ENVIRONMENT SETUP

ANDROID or iOS GADGET

Silakan install aplikasi Expo GO dari masing-masing store

CATATAN !

Untuk menggunakan metode ini, laptop dan gadget kita harus berada dalam 1 WiFi yang sama



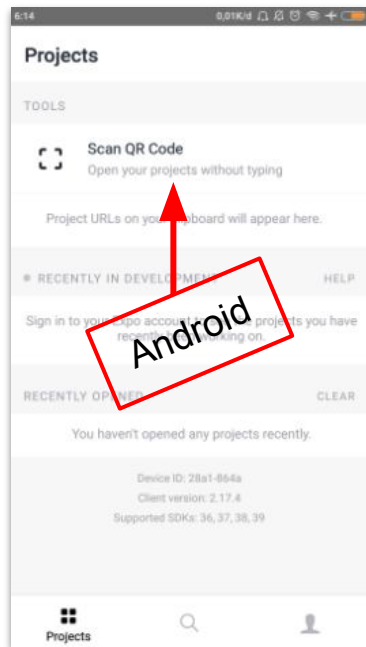
Sesi 18 | Introduction React Native

DEVELOPMENT ENVIRONMENT SETUP

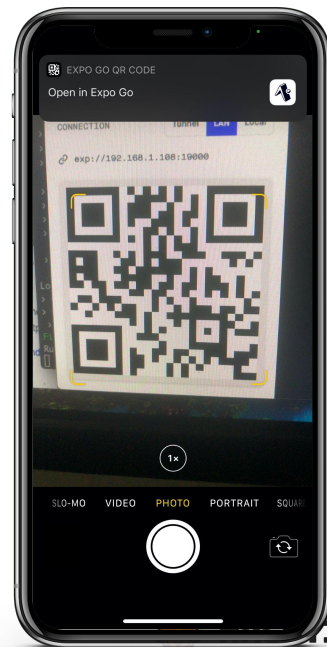
ANDROID or iOS GADGET

Setelah melakukan instalasi, kita bisa membuka aplikasi kita dengan scan QR yang muncul pada web atau terminal

Untuk Android, bisa scan QR dari Expo Go



Untuk iOS, silakan gunakan camera langsung untuk scan QR code nya, dan klik notifikasi yang muncul



DEVELOPMENT ENVIRONMENT SETUP

And finally...



WARNING !! Bagaimana kalau kita **TIDAK** terhubung dengan WiFi dan ingin running app kita ? Cara ke 3 yaitu dengan **Android Debug Brige (adb)**. Dan dengan cara ini, kita cukup menyambungkan gadget Android kita dengan kabel data ke laptop. Jangan lupa aktifkan Developer Mode, dan USB Debugging pada gadget kita.

Penjelasan adb : <https://developer.android.com/studio/command-line/adb>

Instalasi adb : <https://developer.android.com/studio/releases/platform-tools>

Jika sudah terinstall, sambungkan gadget dengan laptop, jalankan perintah :

```
> adb start-server
```

Kemudian kita bisa lihat daftar devices yang terhubung dengan perintah :

```
> adb devices
```

... jika ada device yang terdaftar maka lakukan **step 11**, dan kita sudah siap untuk beraksi

```
~ adb start-server
* daemon not running; starting now at tcp:5037
* daemon started successfully
```

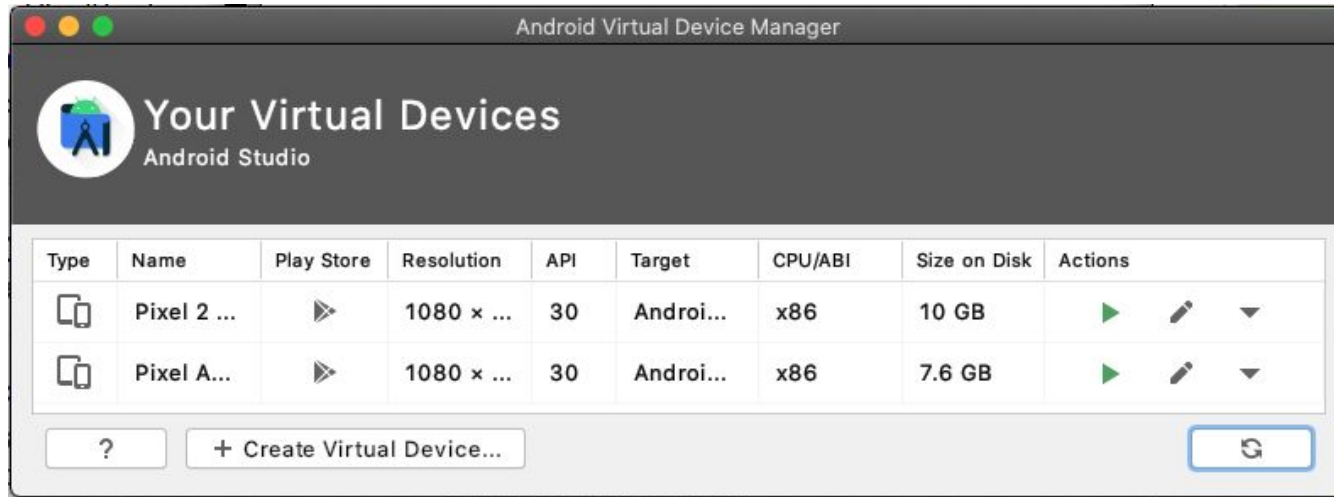
```
~ adb devices
List of devices attached
OZWWOFSOGMDQ85IV    device
```



Sesi 18 | Introduction React Native

DEVELOPMENT ENVIRONMENT SETUP

WARNING !! Kalau kita perhatikan, penggunaan Emulator cukup memakan space dan memory. Oleh karena itu, sangat dianjurkan agar kita memakai gadget kita untuk menikmati hasil koding kita



Sesi 18 | Introduction React Native

BERMAIN DENGAN KODE REACT NATIVE

Sekarang mari kita perhatikan kode yang ter-generate dari hasil perintah expo init tadi (./App.js)

```
1  import { StatusBar } from 'expo-status-bar';
2  import React from 'react';
3  import { StyleSheet, Text, View } from 'react-native';
4
5  export default function App() {
6    return (
7      <View style={styles.container}>
8        <Text>Open up App.js to start working on your app!</Text>
9        <StatusBar style="auto" />
10      </View>
11    );
12  }
13
14  const styles = StyleSheet.create({
15    container: {
16      flex: 1,
17      backgroundColor: '#fff',
18      alignItems: 'center',
19      justifyContent: 'center',
20    },
21  });
```

Mari kita perhatikan beberapa potongan kode dari gambar

View bertindak sama seperti sebuah elemen `<div>` pada aplikasi berbasis web

Text bertindak sama dengan sebuah elemen `<p>` pada aplikasi berbasis web

StyleSheet menjadi tempat kita menyimpan style dari elemen yang kita pakai

Seperti yang terlihat, tidak berbeda jauh dengan yang biasa kita pakai pada aplikasi web bukan ?

Sesi 18 | Introduction React Native

BERMAIN DENGAN KODE REACT NATIVE

Mari kita ubah ./App.js kita sedikit menjadi seperti di bawah ini, dan saksikan perubahan yang terjadi pada gadget atau emulator kita

```
1  import { StatusBar } from 'expo-status-bar';
2  import React from 'react';
3  import { StyleSheet, Text, View } from 'react-native';
4
5  export default function App() {
6    return (
7      <View style={styles.container}>
8        <Text style={styles.txt}>
9          Mari kita belajar React Native
10        </Text>
11        <StatusBar style="auto" />
12      </View>
13    );
14  }
15
16  const styles = StyleSheet.create({
17    container: {
18      flex: 1,
19      backgroundColor: '#fff',
20      alignItems: 'center',
21      justifyContent: 'center',
22    },
23    txt: {
24      fontSize: 30
25    }
26  });
```

Yes, sama seperti pada aplikasi React untuk web, hot reload akan langsung membuat perubahan yang kita lakukan dan save ditampilkan pada emulator atau gadget



Sesi 18 | Introduction React Native

BERMAIN DENGAN KODE REACT NATIVE

Masih bermain dengan ./App.js, kita akan explore beberapa components yang disediakan oleh React Native. Kita akan explorasi dokumentasi yang terdapat pada <https://reactnative.dev/docs/components-and-apis>

IMAGE

Carilah sebuah gambar, mungkin logo React Native, untuk kita masukkan ke dalam aplikasi kita. Ada 2 hal minimal yang harus ada ketika kita akan memasukkan image ke dalam app : **source** dan **style**

1. Import

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

2. Style

Tambahkan style berikut ini ke dalam deklarasi styles

```
29 |   img: {  
30 |     height: 100,  
31 |     width: 100  
32 |   }
```



Sesi 18 | Introduction React Native

BERMAIN DENGAN KODE REACT NATIVE

IMAGE

3. Tambahkan Image tag lengkap dengan props nya

... ke dalam function return dari komponen App

```
<Image
  style={styles.img}
  source={{
    uri: 'https://www.shareicon.net/data/256x256/2016/07/08/117367_logo_512x512.png'
  }} />
```

4. Save

... dan hasilnya akan auto reload ke emulator / gadget kita



Sesi 18 | Introduction React Native

BERMAIN DENGAN KODE REACT NATIVE

BUTTON

Yep, sebuah elemen / komponen yang akan dan sering kita pakai untuk trigger berbagai aksi

1. Import

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

2. Style

... sayangnya, secara default, component Button tidak bisa kita tambahkan style. Kita hanya bisa menambahkan color ketika pemanggilan komponen nya.

Mari kita ke slide selanjutnya



Sesi 18 | Introduction React Native

BERMAIN DENGAN KODE REACT NATIVE

BUTTON

3. Tambahkan Button tag lengkap dengan props nya

... ke dalam function return dari komponen App

```
<Button  
  title="Click Me"  
  color="blue"  
/>
```

4. Save

... dan hasilnya akan auto reload ke emulator / gadget kita



BERMAIN DENGAN KODE REACT NATIVE

Masih banyak lagi komponen dan api lain yang bisa kita explore. Silakan mengacu kepada dokumen resmi React Native untuk mengetahui apa saja yang bisa kita implementasi. Kita akan bermain dengan beberapa di antaranya, plus aturan main layouting di React Native.

Core Components and APIs

React Native provides a number of built-in [Core Components](#) ready for you to use in your app. You can find them all in the left sidebar (or menu above, if you are on a narrow screen). If you're not sure where to get started, take a look at the following categories:

- [Basic Components](#)
- [User Interface](#)
- [List Views](#)
- [Android-specific](#)
- [iOS-specific](#)
- [Others](#)

You're not limited to the components and APIs bundled with React Native. React Native has a community of thousands of developers. If you're looking for a library that does something specific, please refer to [this guide about finding libraries](#).

Basic Components

User Interface

List Views

Android Components and APIs

iOS Components and APIs

Others

