

LAPORAN PRATIKUM
PEMROGRAMAN WEB DAN MOBILE I



NAMA : Rio Geradi Muke
NIM : 193030503051
MODUL : VI
KELAS : A

JURUSAN/PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA
2021

BAB I

TUJUAN DAN LANDASAN TEORI

1. Landasan Teori

React Native adalah salah satu topik hangat di bidang *mobile development* selama beberapa tahun terakhir. Salah satu alasan mengapa React Native sangat populer karena menawarkan cara untuk mengembangkan *mobile app* atau aplikasi selular dengan mudah. Membuat sebuah aplikasi selular bukanlah hal mudah karena membutuhkan *coding* khusus yang pengembangannya memerlukan waktu lama. Namun, React Native berhasil membawa perubahan. Bahkan, *framework* itu dapat membantu pengembang untuk membuat aplikasi Android dan iOS dengan lebih cepat dan mudah.

Pengertian React native

React Native adalah framework mobile app development yang memungkinkan pengembangan aplikasi secara multi-platform yaitu Android dan iOS. Sederhananya, React Native adalah framework yang memungkinkan developer membuat aplikasi untuk berbagai platform dengan menggunakan basis kode yang sama. Lewat website resmi React Native, mereka mengklaim bahwa framework berbasis JavaScript tersebut adalah cara yang terbaik untuk membangun user interface mobile app. Penggabungan antara native mobile app dan React juga bisa dilakukan dengan mudah. Jadi, pengembang bisa membuat aplikasi Android dan iOS dengan lebih cepat. React Native sendiri pertama kali dirilis pada tahun 2015 oleh Facebook dan menjadi bagian dari proyek open source mereka. Kemudian, hanya dalam beberapa tahun saja, React Native telah menjadi andalan bagi banyak developer untuk mengembangkan aplikasi seluler. Beberapa contoh aplikasi seluler yang menggunakan *framework* React Native adalah Instagram, Facebook, Pinterest, hingga Skype. Ada beberapa alasan lain mengapa *React Native* begitu populer, antara lain:

- *Developer* hanya perlu membuat satu kode untuk mengembangkan aplikasi berbasis Android atau iOS. Jadi, *framework* ini benar-benar menghemat waktu.
- React Native adalah *framework* yang dibangun berdasarkan React, yaitu *library* dari JavaScript yang sudah populer.
- *Framework* ini mendorong *frontend developer* yang sebelumnya hanya dapat bekerja dengan teknologi berbasis web saja. Kini, mereka bisa dengan mudah untuk mengembangkan *mobile platform*.

Selain beberapa alasan di atas, *React Native* juga memiliki keunggulan lain yaitu tidak akan *me-render webviews* dalam kodenya.

Cara Kerja React Native

React Native adalah framework yang ditulis dengan campuran bahasa JavaScript dan JXL, sebuah kode markup khusus yang menyerupai XML. Framework ini memiliki kemampuan untuk berkomunikasi dengan kedua ranah sekaligus, yaitu threads yang berbasis JavaScript dan threads dari native app. Jadi, bagaimana caranya untuk berkomunikasi? Rupanya React Native menggunakan apa yang disebut dengan “bridge” atau jembatan. JavaScript dan threads native memang ditulis dengan dua bahasa pemrograman yang berbeda. Namun, fitur bridging dari React Native tetap memungkinkan untuk komunikasi dua arah. Itu artinya saat pengembang sudah memiliki aplikasi Android atau iOS, mereka masih tetap bisa menggunakan komponennya saat menggunakan React Native.

Kelebihan

1. Hemat biaya

Kelebihan pertama dari React Native adalah biaya yang lebih hemat jika dibandingkan dengan proses pengembangan yang lainnya. Itulah mengapa perusahaan startup lebih menyukai untuk menggunakan React Native dalam mengembangkan aplikasi seluler mereka. Saat melakukan pengembangan dengan React Native tentu

perusahaan tidak perlu menyewa dua tim pengembang Android dan iOS secara terpisah. Jadi, anggaran untuk membayar dua tim tersebut bisa dialokasikan untuk keperluan yang lain.

2. Aplikasi bisa berjalan lebih cepat

Sebelumnya banyak yang beranggapan bahwa JavaScript tidak akan bisa membuat aplikasi berjalan lebih cepat seperti saat menggunakan kode dari native. Namun, dalam percobaan yang dilakukan oleh Netguru ditemukan bahwa kinerja dari aplikasi yang menggunakan React Native dan kode native memiliki hasil yang sama. Jadi, selain mudah dalam tahapan membuat kode, rupanya React Native juga memiliki pengaruh dalam kecepatan berjalannya aplikasi.

3. Lebih hemat waktu

Kelebihan selanjutnya dari React Native adalah bisa menghemat waktu lebih banyak untuk proses pengembangannya. Kode dari React Native bisa digunakan kembali tanpa banyak mengubah komponennya. Jadi, proses pengembangan aplikasi seluler pun bisa menjadi lebih cepat.

4. Memiliki komunitas pengembang yang besar

React Native adalah framework JavaScript open source yang memungkinkan para pengembang untuk saling berbagi pengetahuannya. Oleh karena itu, saat ada kesulitan saat menggunakan React Native, developer tidak perlu khawatir karena bisa bertanya di komunitas pengembang. Selain itu, para pengembang pemula yang ikut komunitas juga mendapatkan keuntungan yaitu bisa meningkatkan kemampuan coding.

Kekurangan

1. Kompatibilitas dan debugging

React Native memang menjadi salah satu framework yang terbilang masih muda. Jadi, saat ini mereka masih dalam fase beta. Hal tersebut bisa membuat pengembang menemukan berbagai masalah dengan kompatibilitas dan debugging. Jika pengembang tersebut masih belum

terlalu mahir dalam menggunakan React Native, tentu akan membuatnya kesulitan untuk memecahkan masalah tersebut.

2. Kinerja dan kualitasnya lebih rendah

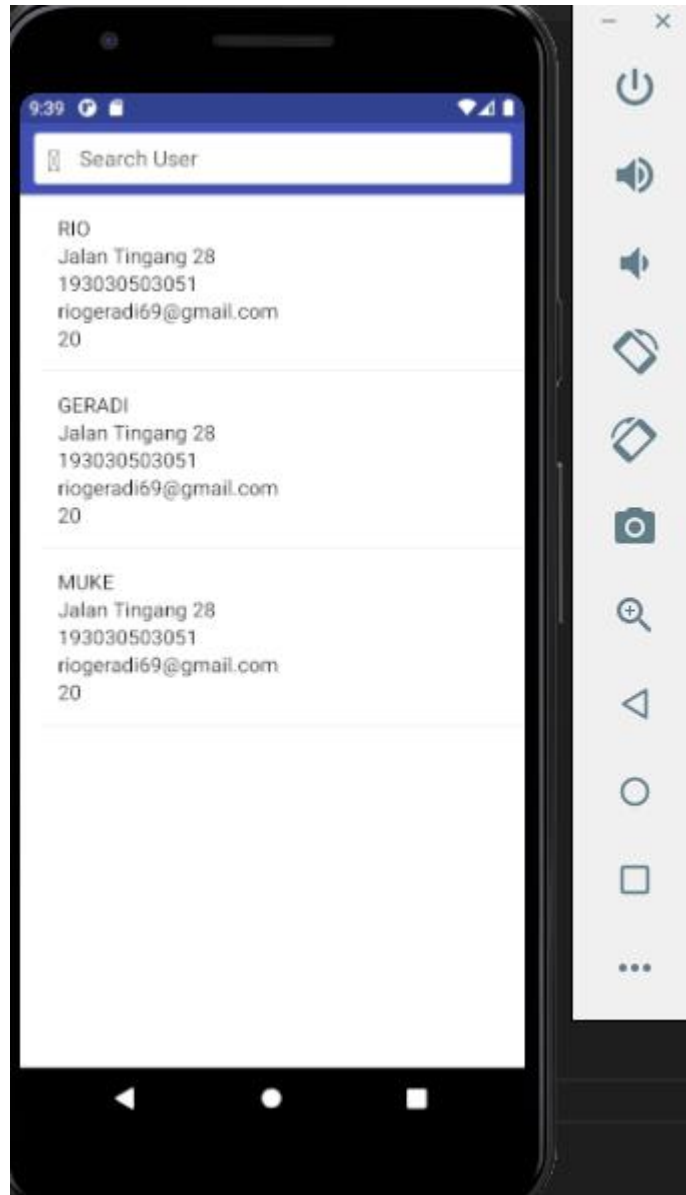
Kekurangan dari React Native yang selanjutnya adalah hasil kinerja dan kualitas dari aplikasi yang cukup rendah. React Native merupakan pengembangan lintas platform sehingga tidak semua aspek akan membuahkan hasil yang sempurna. Kinerja dan kualitas menjadi salah satu hal yang harus dikorbankan. Hal itu berbeda dengan saat menggunakan pengembangan kode native yang tentunya bisa memberikan hasil yang lebih maksimal. Itulah mengapa React Native tidak dijadikan pilihan untuk membuat aplikasi yang menjalankan animasi yang berat atau pun game.

3. Tetap membutuhkan native developer

React Native memang memiliki fitur bridging untuk menjembatani JavaScript dan kode native. Akan tetapi, jika perusahaan tidak memiliki native developer tentu akan cukup kesulitan. Native developer masih dibutuhkan untuk memasukkan kode native ke dalam basis kode React Native. Jadi, jika pengembang tidak memiliki keterampilan menggunakan kode native tentu akan kebingungan. Itulah mengapa masih banyak perusahaan yang akhirnya tetap membutuhkan seorang native developer seperti Android developer atau iOS developer. Proses pengembangan pun bisa menjadi lebih rumit karena ada tiga jenis pengembang yang harus mengelola tiga basis kode yang berbeda. Bukannya semakin mudah, hal itu malah bisa memakan waktu yang lama dalam proses pengembangan aplikasi seluler.

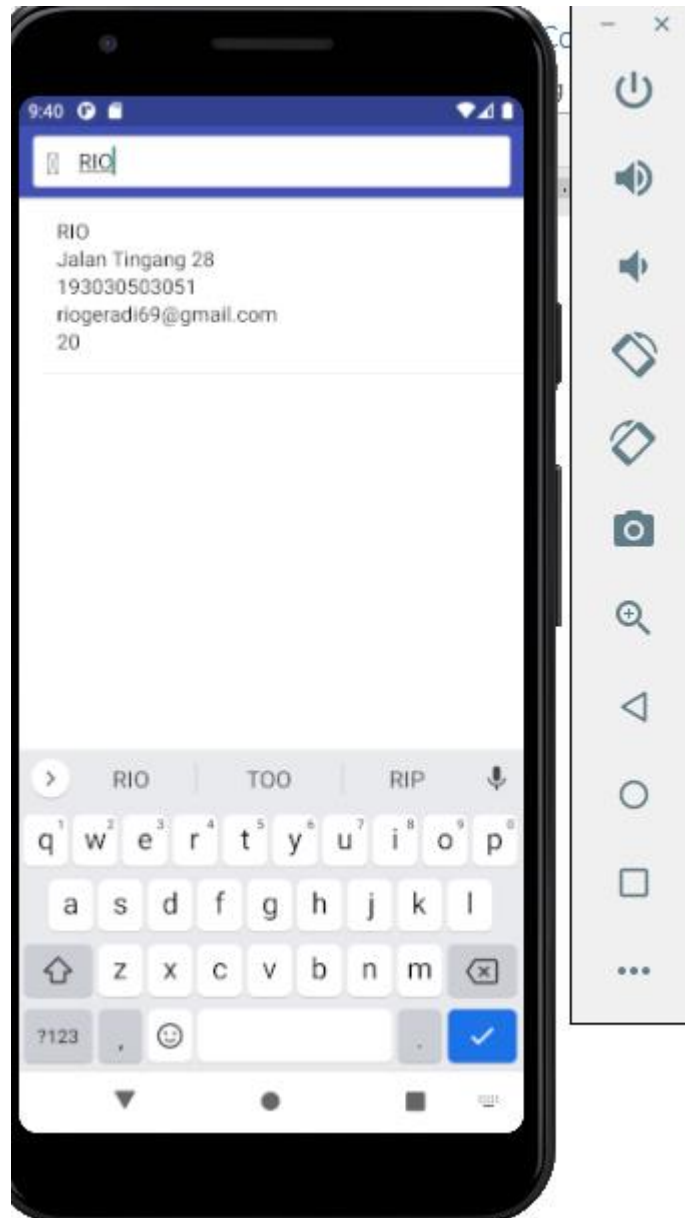
BAB II

PEMBAHASAN



Gambar 2.1 Tampilan Emulator Android Bagian 1

Gambar 2.1 adalah hasil dari CMD React-native run-android lalu setelah di bagian VCS untuk mengubah kata-kata tersebut di dalam file nama.json setelah itu ubah lah coding tersebut sesuai dengan data diri kita dan contohnya nama, jl, nim, email, dan umur. Dan menggunakan sebuah perintah search User dari nama nama tersebut.

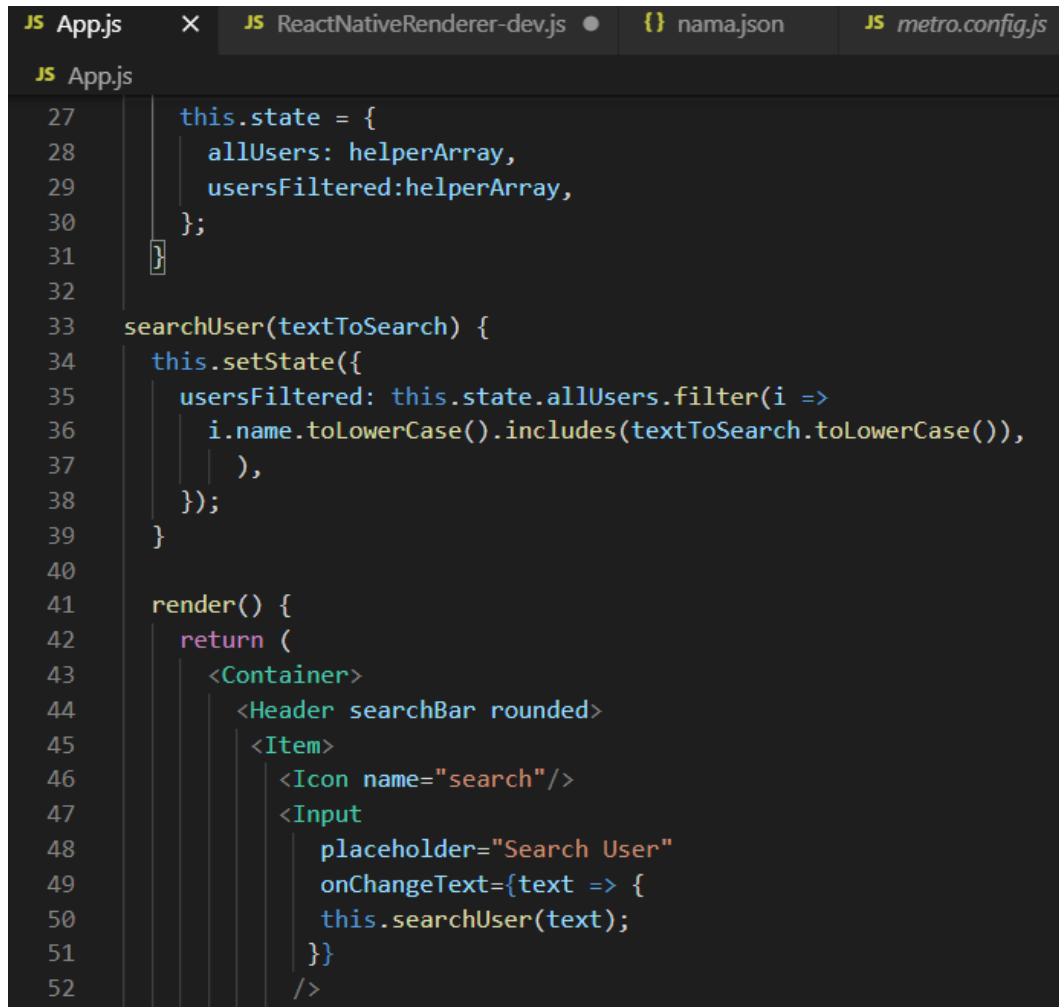


Gambar 2.2 Tampilan Emulator Android Search Bagian 2

Gambar 2.2 adalah cara untuk mencari sebuah nama-nama dari sebuah perintah search karena di tampilan tersebut menggunakan kata rio maka hasil pencariannya yaitu RIO dan data-data lengkapnya. Dan akan muncul seperti gambar 2.2

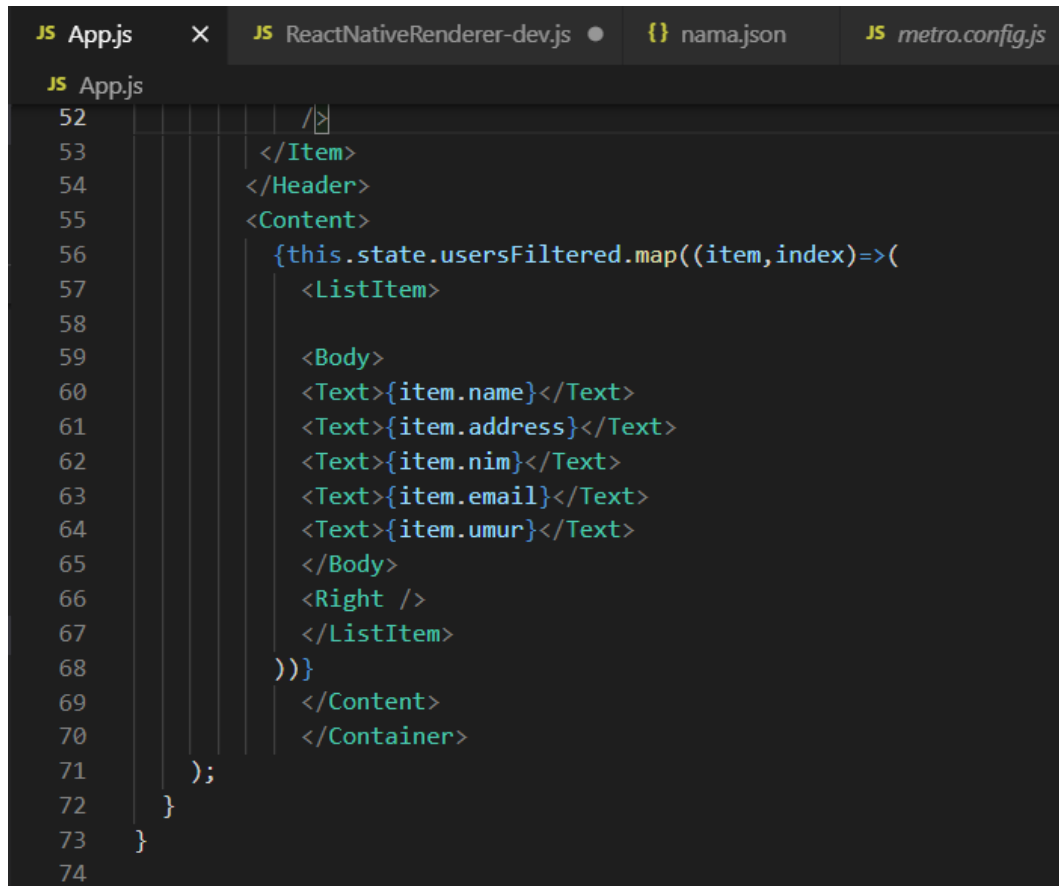
```
JS App.js X JS ReactNativeRenderer-dev.js {} nama.json
JS App.js
1  /**
2   * Sample React Native App
3   * https://github.com/facebook/react-native
4   *
5   * @format
6   * @flow strict-local
7   */
8
9  import React, {Component} from 'react';
10 import{
11   Container,
12   Header,
13   Content,
14   Left,
15   Right,
16   Body,
17   Icon,
18   Text,
19   ListItem,
20   Item,
21   Input,
22 } from 'native-base';
23 let helperArray = require('./nama.json');
24 export default class App extends Component {
25   constructor(props){
26     super(props);
```

Gambar 2.3 APP JS Bagian 1



```
JS App.js  X  JS ReactNativeRenderer-dev.js  {} nama.json  JS metro.config.js
JS App.js
27     this.state = {
28         allUsers: helperArray,
29         usersFiltered: helperArray,
30     };
31 }
32
33 searchUser(textToSearch) {
34     this.setState({
35         usersFiltered: this.state.allUsers.filter(i =>
36             i.name.toLowerCase().includes(textToSearch.toLowerCase()),
37         ),
38     });
39 }
40
41 render() {
42     return (
43         <Container>
44             <Header searchBar rounded>
45                 <Item>
46                     <Icon name="search"/>
47                     <Input
48                         placeholder="Search User"
49                         onChangeText={text => {
50                             this.searchUser(text);
51                         }}
52                     />
```

Gambar 2.4 APP JS Bagian 2



```
JS App.js x JS ReactNativeRenderer-dev.js {} nama.json JS metro.config.js
JS App.js
52 //
53 </Item>
54 </Header>
55 <Content>
56   {this.state.usersFiltered.map((item,index)=>{
57     <ListItem>
58
59       <Body>
60         <Text>{item.name}</Text>
61         <Text>{item.address}</Text>
62         <Text>{item.nim}</Text>
63         <Text>{item.email}</Text>
64         <Text>{item.umur}</Text>
65       </Body>
66       <Right />
67     </ListItem>
68   )}
69 </Content>
70 </Container>
71 );
72 }
73 }
74
```

Gambar 2.5 APP JS Bagian 3

Gambar 2.3, 2.4 dan 2.5 terdapat deklarasi variabel dengan nama helperArray dimana variabel tersebut berisi require, require digunakan untuk dapat menghubungkan file app.js tersebut dengan file nama.json. nama.json merupakan file yang berisi daftar nama-nama yang akan digunakan pada program ini. Berikut adalah isi dari file nama.json dan untuk baris 53 sampai 70 adalah sebuah perintah text untuk manggil sebuah variabel ke dalam nama.jason setelah itu sesuai dengan perintahnya adalah name, address, nim, email dan umur dan setelah itu tinggal run saja kedalam sebuah emulator android apa bila telah berhasil;.

```

{} nama.json > {} 2
1  [
2    {
3      "name": "RIO ",
4      "address": "Jalan Tingang 28",
5      "nim": "193030503051",
6      "email": "riogeradi69@gmail.com",
7      "umur": "20"
8    },
9
10   {
11     "name": "GERADI",
12     "address": "Jalan Tingang 28",
13     "nim": "193030503051",
14     "email": "riogeradi69@gmail.com",
15     "umur": "20"
16   },
17
18   [
19     {
20       "name": "MUKE",
21       "address": "Jalan Tingang 28",
22       "nim": "193030503051",
23       "email": "riogeradi69@gmail.com",
24       "umur": "20"
25     }
26   ]

```

Gambar 2.6 Nama.json

Gambar 2.6 adalah sebuah masukan variabel Setelah itu, terdapat kelas dengan nama App, yang berisi, konstruktor, dimana konstruktor tersebut berisi inisiasi atribut dengan nilai sama dengan variabel helperArray di dalam file app.js . Atribut tersebut akan mengambil data-data dari variabel helperArray. Dan setelah itu di isi sesuai dengan name, address, nim, email dan umur. Apa bila sudah selesai mengisi maka tampilannya seperti gambar 2.6 dan jangan sampai salah dalam mengisi sebuah variabel tersebut agar dapat di baca di dalam sebuah emulator android. Lebih penting lagi koma agar tidak ketinggalan.

```
C:\WINDOWS\system32\cmd.exe

C:\Users\ASUS\Documents\RIOGERADIMUKE>react-native run-android
info Running jetifier to migrate libraries to AndroidX. You can disable it using "--no-jetifier" flag.
(node:4812) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
Jetifier found 945 file(s) to forward-jetify. Using 8 workers...
info JS server already running.
info Installing the app...

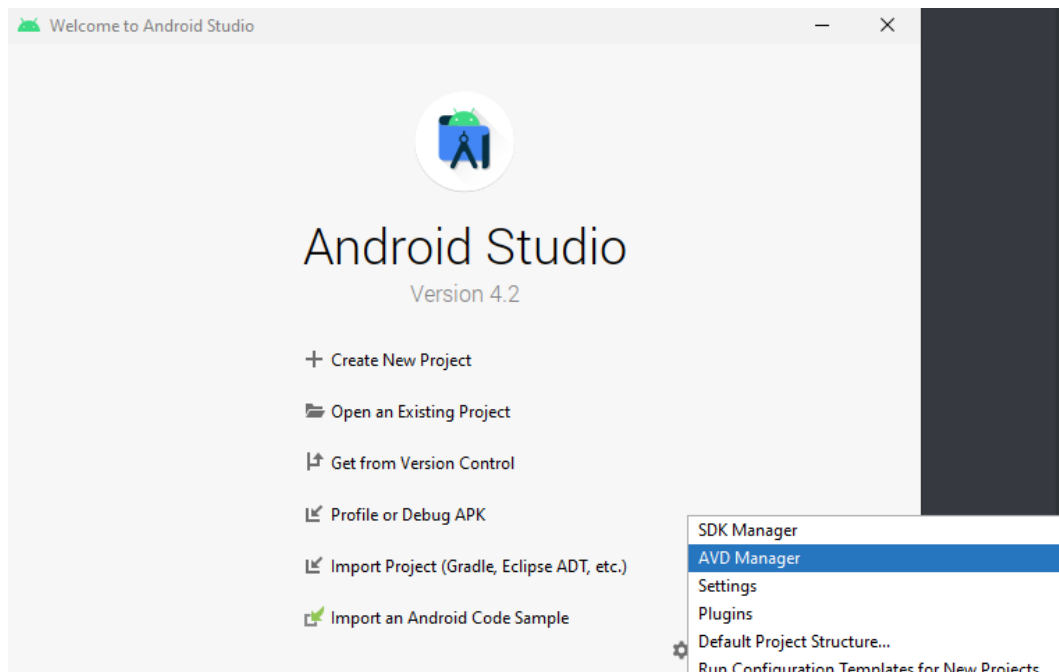
> Task :app:installDebug
Installing APK 'app-debug.apk' on 'Pixel_3a_API_30_x86(AVD) - 11' for app:debug
Installed on 1 device.

BUILD SUCCESSFUL in 1m 16s
29 actionable tasks: 2 executed, 27 up-to-date
info Connecting to the development server...
8081
info Starting the app on "emulator-5554"...
Starting: Intent { cmp=com.riogeradimuke/.MainActivity }

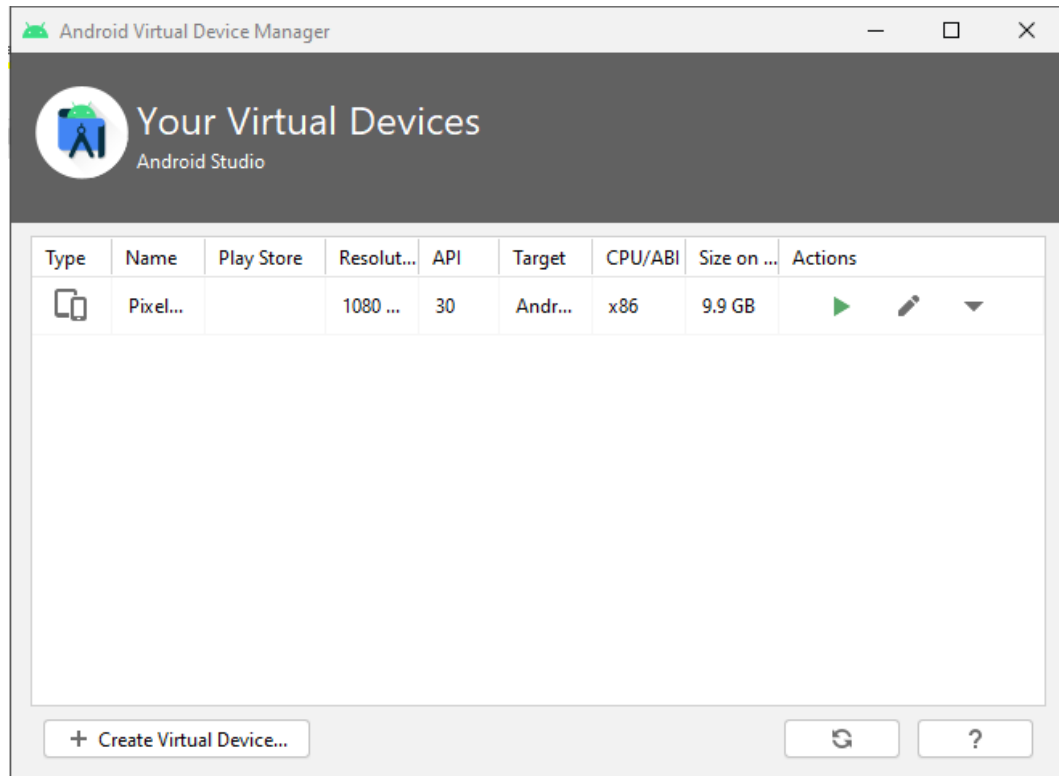
C:\Users\ASUS\Documents\RIOGERADIMUKE>
```

Gambar 2.7 CMD Project Saya.

Gambar 2.7 adalah sebuah perintah untuk menjalankan sebuah react-native run-android adalah menjalankan sebuah emulator android di dalam CMD dan menunggu beberapa saat setelah itu apa bila telah sukses maka emulator android telah berjalan dengan sesuai pratikum dan terdapat data-data name, address, nim, email dan umur. Yang tertampil di dalam tersebut.



Gambar 2.8 Android Studio Bagian 1



Gambar 2.9 Android Studio Bagian 2

Gambar 2.8 dan 2.9 adalah untuk membuka sebuah android melalui android studio yang telah di jalankan ke dalam CMD. Dan apa bila telah sukses maka yang harus di lakukan adalah menjalankan action tersebut di dalam android studio dan maka tampilannya akan muncul emulator android tapi harus menunggu beberapa saat.

BAB III

KESIMPULAN

Dari partikum modul 6 ini kesimpulannya adalah Search yaitu di dalam react native, untuk sebuah program yang dimana program tersebut memerlukan sebuah fitur yang namanya pencarian lalu pengguna menjadi lebih mudah dalam mencari user dan menggunakan program yang di dalamnya terdapat fitur search

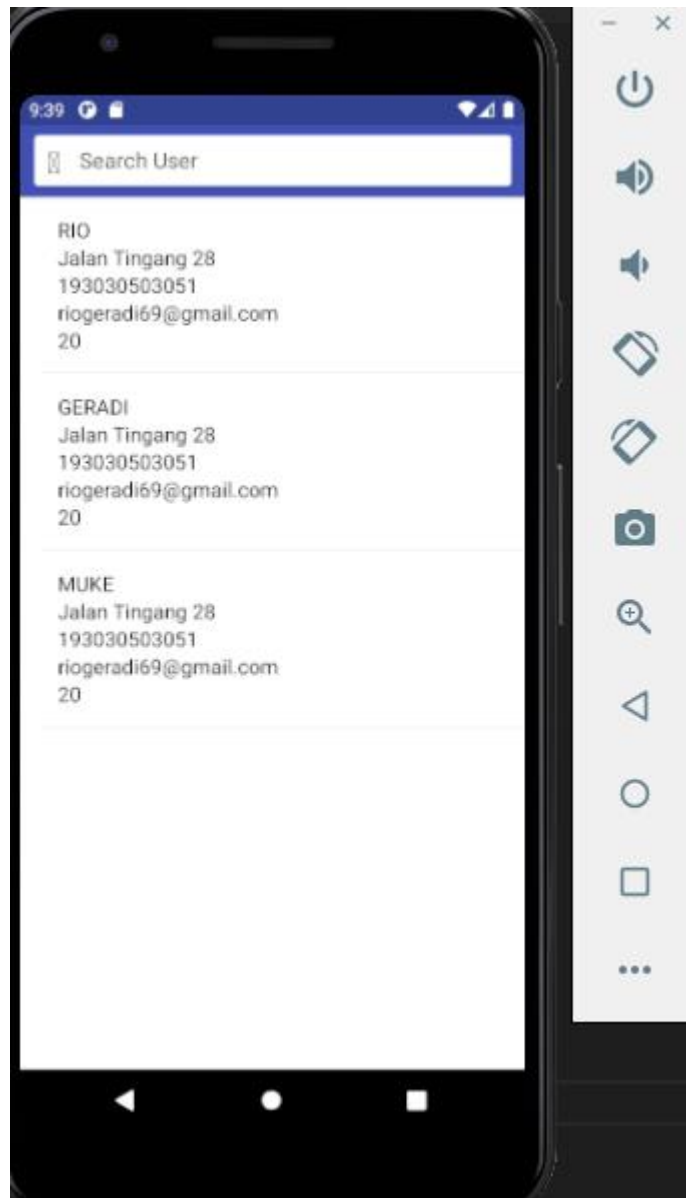
DAFTAR PUSTAKA

Facebook. (2021). *React Native: Learn once, write anywhere*.
<https://reactnative.dev/>

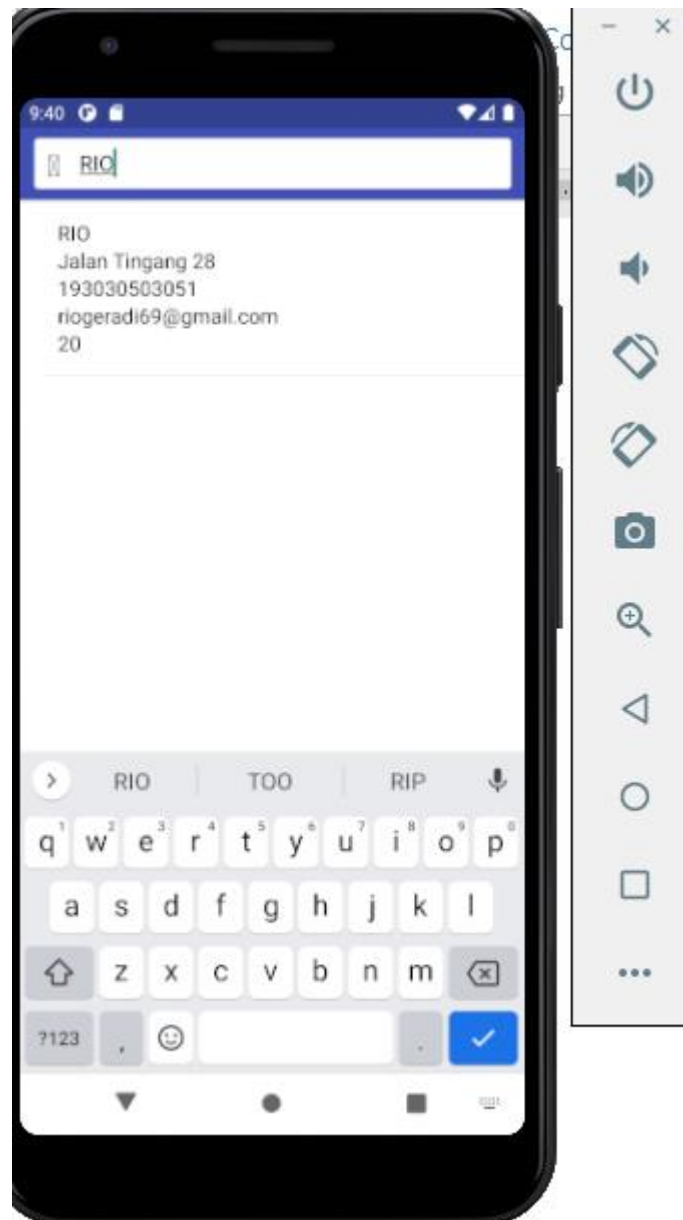
Praktikum, K. (2021). *MODUL PRAKTIKUM PEMROGRAMAN WEB I Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya*.

Trias, I. (2021). *React Native: Ketahui Pengertian serta Kelebihan dan Kekurangannya*.
<https://glints.com/id/lowongan/react-native-adalah/#.YHOqeT8RXDc>

LAMPIRAN



Gambar 2.1 Tampilan Emulator Android Bagian 1

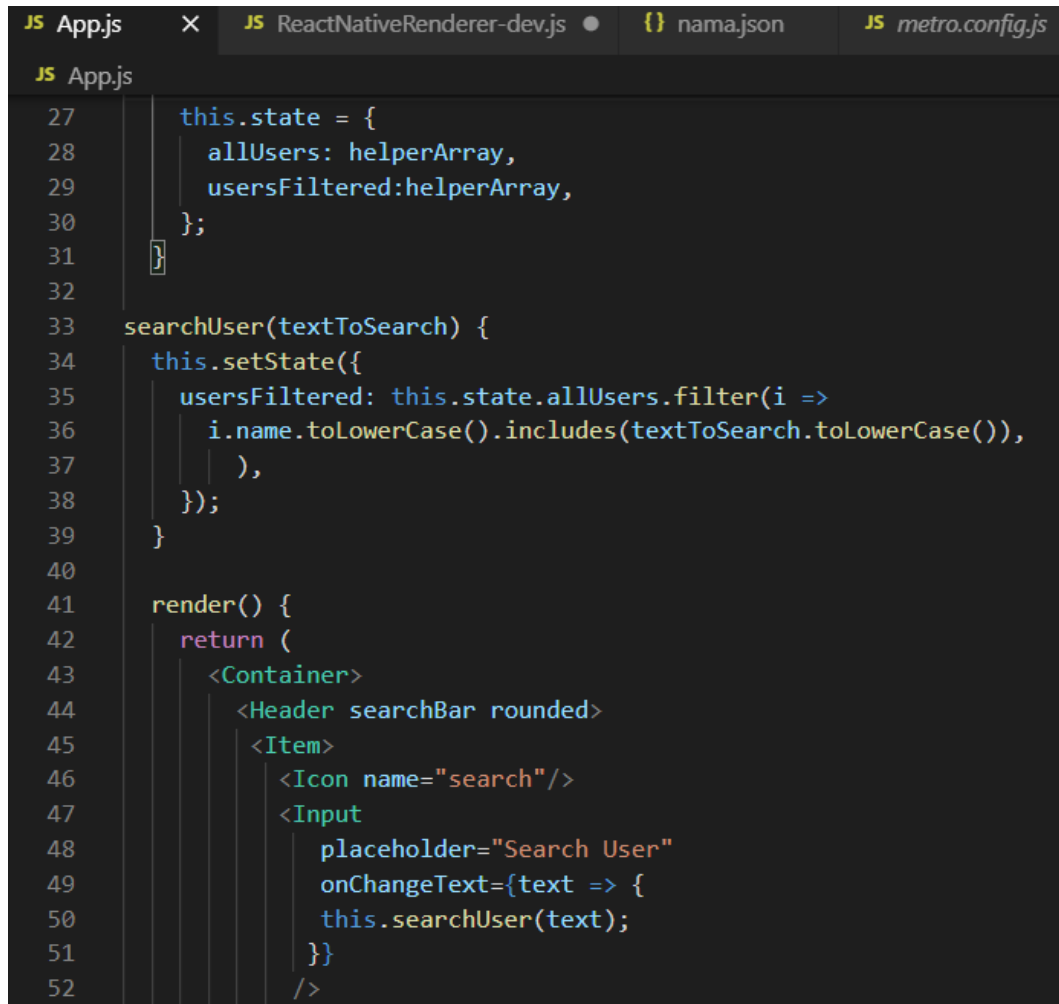


Gambar 2.2 Tampilan Emulator Android Search Bagian 2



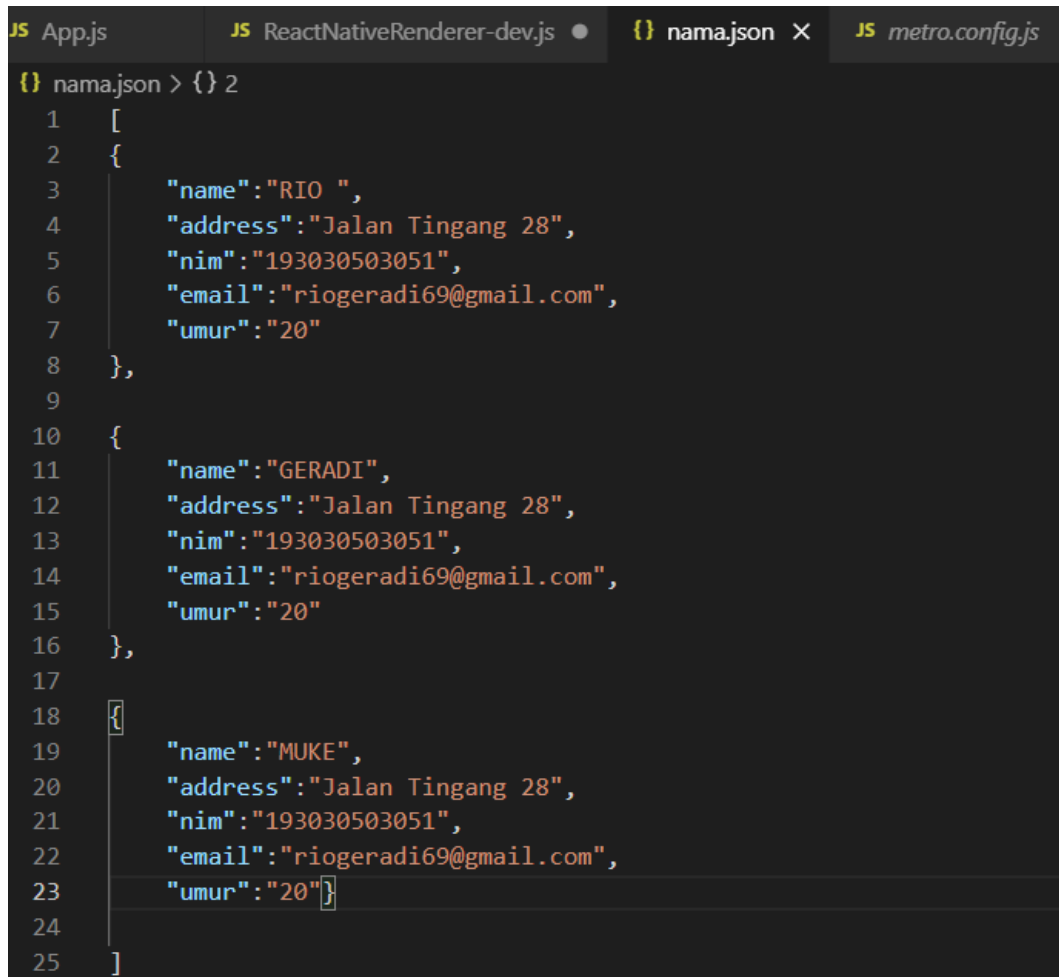
```
JS App.js X JS ReactNativeRenderer-dev.js {} nama.json
JS App.js
1  /**
2   * Sample React Native App
3   * https://github.com/facebook/react-native
4   *
5   * @format
6   * @flow strict-local
7   */
8
9  import React, {Component} from 'react';
10 import{
11   Container,
12   Header,
13   Content,
14   Left,
15   Right,
16   Body,
17   Icon,
18   Text,
19   ListItem,
20   Item,
21   Input,
22 } from 'native-base';
23 let helperArray = require('./nama.json');
24 export default class App extends Component {
25   constructor(props){
26     super(props);
```

Gambar 2.3 APP JS Bagian 1



```
JS App.js x JS ReactNativeRenderer-dev.js {} nama.json JS metro.config.js
JS App.js
27     this.state = {
28       allUsers: helperArray,
29       usersFiltered: helperArray,
30     };
31   }
32
33   searchUser(textToSearch) {
34     this.setState({
35       usersFiltered: this.state.allUsers.filter(i =>
36         i.name.toLowerCase().includes(textToSearch.toLowerCase()),
37       ),
38     });
39   }
40
41   render() {
42     return (
43       <Container>
44         <Header searchBar rounded>
45           <Item>
46             <Icon name="search"/>
47             <Input
48               placeholder="Search User"
49               onChangeText={text => {
50                 this.searchUser(text);
51               }}
52             />
```

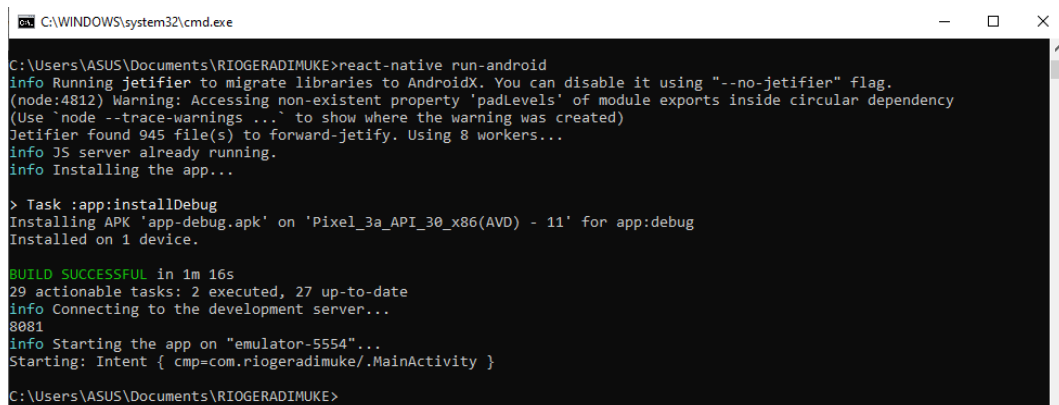
Gambar 2.4 APP JS Bagian 2



The image shows a code editor with four tabs: 'App.js', 'ReactNativeRenderer-dev.js', 'nama.json', and 'metro.config.js'. The 'nama.json' tab is active, displaying a JSON array with three objects. Each object contains the fields 'name', 'address', 'nim', 'email', and 'umur'. The first object has 'name': 'RIO ', the second has 'name': 'GERADI', and the third has 'name': 'MUKE'. All other fields are identical across the objects.

```
{} nama.json > {} 2
1  [
2    {
3      "name": "RIO ",
4      "address": "Jalan Tingang 28",
5      "nim": "193030503051",
6      "email": "riogeradi69@gmail.com",
7      "umur": "20"
8    },
9
10   {
11     "name": "GERADI",
12     "address": "Jalan Tingang 28",
13     "nim": "193030503051",
14     "email": "riogeradi69@gmail.com",
15     "umur": "20"
16   },
17
18   [
19     {
20       "name": "MUKE",
21       "address": "Jalan Tingang 28",
22       "nim": "193030503051",
23       "email": "riogeradi69@gmail.com",
24       "umur": "20"
25     }
26   ]
27 ]
```

Gambar 2.6 Nama.json



The image shows a terminal window with the command 'react-native run-android' executed. The output includes information about running Jetifier, installing the app, and starting the app on an emulator. The build is successful, and the app is running on the 'emulator-5554'.

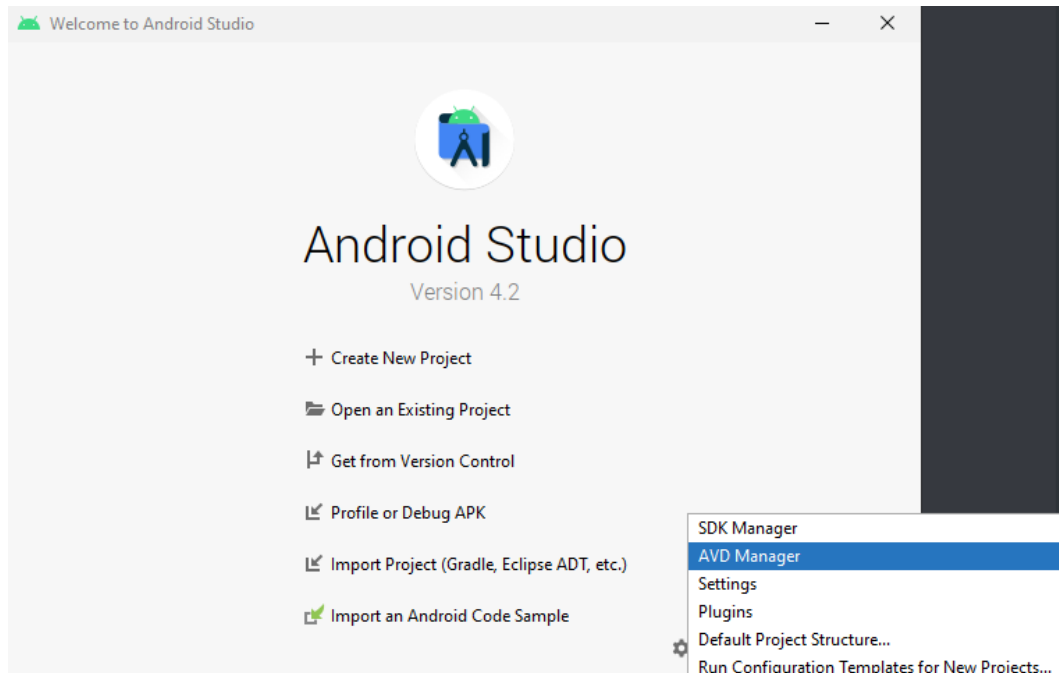
```
C:\WINDOWS\system32\cmd.exe
C:\Users\ASUS\Documents\RIOGERADIMUKE>react-native run-android
info Running jetifier to migrate libraries to AndroidX. You can disable it using "--no-jetifier" flag.
(node:4812) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(Use 'node --trace-warnings ...' to show where the warning was created)
Jetifier found 945 file(s) to forward-jetify. Using 8 workers...
info JS server already running.
info Installing the app...

> Task :app:installDebug
Installing APK 'app-debug.apk' on 'Pixel_3a_API_30_x86(AVD) - 11' for app:debug
Installed on 1 device.

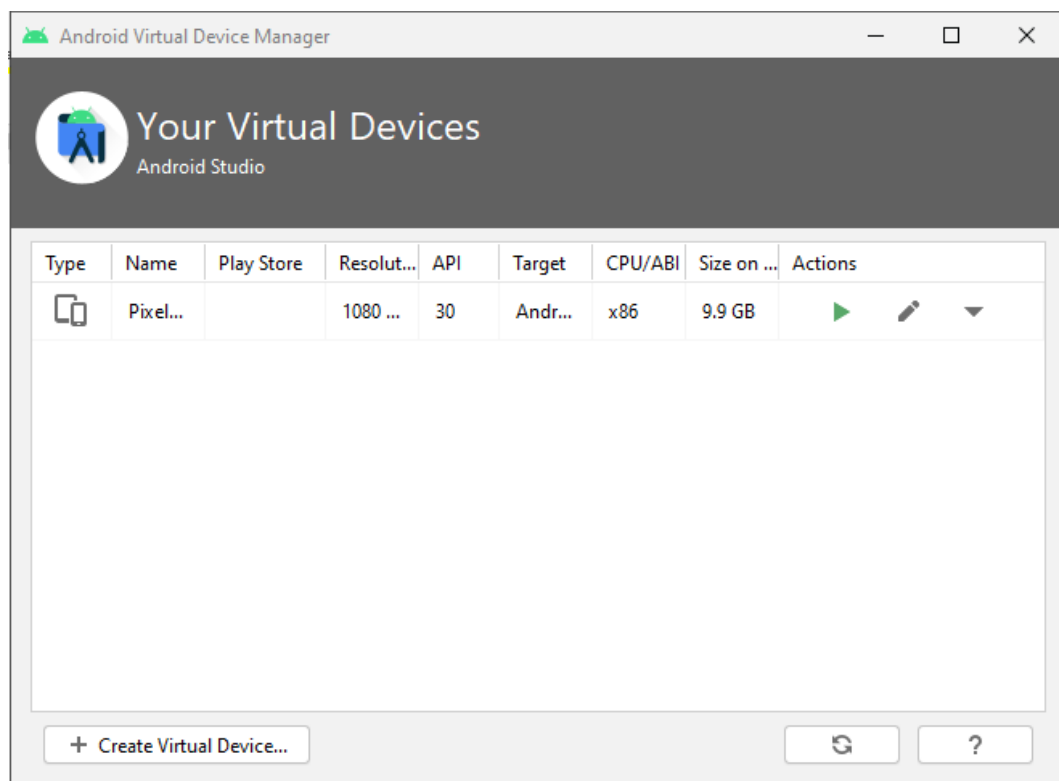
BUILD SUCCESSFUL in 1m 16s
29 actionable tasks: 2 executed, 27 up-to-date
info Connecting to the development server...
8081
info Starting the app on "emulator-5554"...
Starting: Intent { cmp=com.riogeradimuke/.MainActivity }

C:\Users\ASUS\Documents\RIOGERADIMUKE>
```

Gambar 2.7 CMD Project Saya.



Gambar 2.8 Android Studio Bagian 1



Gambar 2.9 Android Studio Bagian 2