

**LAPORAN HASIL PRAKTIKUM
PEMROGRAMAN WEB DAN MOBILE 1**



NAMA : AXEL BERKATI
NIM : 193010503007
KELAS : A
**MODUL : VI (SEARCH LIST REACT NATIVE
MEGGUNAKAN COMPONENT NATIVE
BASE)**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA
2021**

BAB 1

TUJUAN DAN LANDASAN TEORI

1.1. Tujuan

1.1.1. Mahasiswa Memahami Apa itu Native base dan Dapat Menghubungkannya ke React Native

1.1.2. Mahasiswa Dapat Membuat Search List React Native Menggunakan Component dari Native Base

1.2. Landasan Teori

1.2.1. React Native

React Native adalah framework open source besutan facebook yang dibuat setelah facebook sebelumnya membuat react.js, React.JS sendiri merupakan sebuah library dari facebook yang dapat digunakan untuk membangun antarmuka pengguna (UI). Jadi, react native adalah framework open source untuk membuat aplikasi multi-platform (android, ios dan windows platform "dalam tahap pengembangan") dengan bahasa javascript, sesuai dengan deskripsi di situs resminya "Learn once, write anywhere".(Wiguna et al., 2019)

React Native adalah salah satu framework untuk membuat aplikasi mobile dengan menggunakan kode JavaScript. Framework React Native memiliki seperangkat komponen bagi platform iOS dan Android untuk membangun aplikasi mobile dengan tampilan yang benar-benar seperti native. Dengan menggunakan Framework React Native, dapat merender User Interface untuk platform iOS dan Android. React Native ini adalah framework open source, yang bisa kompatibel dengan platform lain seperti Windows atau tvOS dalam waktu dekat.

Dengan Framework React Native, user tidak membangun "aplikasi mobile web", "aplikasi HTML5", atau "aplikasi hybrid". user membangun aplikasi mobile native yang tidak dapat dibedakan dari aplikasi yang dibuat menggunakan Objective-C atau Java. React Native

menggunakan blok bangunan User Interface dasar yang sama seperti aplikasi iOS dan Android biasa. Sehingga tinggal memasang blok-blok tersebut menggunakan JavaScript.

1.2.2. React JS

Sebelum membahas lebih jauh tentang apa itu React Native, terlebih dahulu mengetahui apa itu react JS.

- a. React JS adalah pustaka JavaScript yang dibuat oleh Open Source Facebook
- b. React JS adalah pustaka Antarmuka Pengguna (UI)
- c. React JS adalah alat untuk membangun komponen UI
- d. React Native mirip dengan React JS. Namun, React JS menggunakan komponen asli daripada komponen web sebagai blok bangunan penting.

Namun, jika tertarik untuk membuat aplikasi di React Native, mempelajari React JS tidak sepenuhnya wajib, user cukup memulai dengan React Native.(Warcholinski, Matt, 2020).

1.2.3. 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.

1.2.4. Kelebihan dan Kekurangan dari React Native

Berikut ini kelebihan dan kekurangan dari React Native yang perlu di pahami, antara lain:

Kelebihan:

a. 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.

b. 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.

c. 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.

d. 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 :

a. 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.

b. 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.

c. 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.

1.2.5. Cara Belajar React Native bagi Pemula

Ada beberapa hal yang perlu dipersiapkan pemula yang ingin mempelajari react native. Namun, tidak perlu khawatir sebab program ini khusus yang dirancang untuk memudahkan penggunaanya dengan langkah-langkah mudah. Berikut di antaranya yang wajib di perhatikan.

- a. Pertama harus menginstal Nodejs. Nodejs ini adalah runtime untuk bisa menjalankan semua program Javascript di luar browser yang di gunakan.
- b. Kedua, setelah menginstal Nodejs lalu instal NPM atau Yarn yang bisa digunakan untuk mengelola Project Nodejs. NPM akan terinstall secara otomatis pada komputer ketika menginstal Nodejs. Sementara untuk Yarn sedikit berbeda dan harus menginstalnya terlebih dahulu.
- c. Selanjutnya menginstal Java JDK untuk mengkompilasi sebuah aplikasi menjadi sebuah APK.
- d. Langkah berikutnya, membutuhkan Android SDK. Kemudian setelah diunduh dan diinstal, akan dimiliki direktori Android di dalam direktori home. Direktori tersebut akan berisi Android SDK yang dapat digunakan untuk pengembangan aplikasi Android.

1.2.6. List Views

React Native menyediakan seperangkat komponen untuk menyajikan daftar data. Umumnya, menggunakan *FlatList* atau *SectionList*.

The *FlatList* menampilkan komponen daftar bergulir perubahan, tapi sama terstruktur, data. *FlatList* berfungsi dengan baik untuk daftar data yang panjang, di mana jumlah item dapat berubah seiring waktu. Berbeda dengan yang lebih umum *ScrollView*, *FlatList* hanya merender elemen yang saat ini ditampilkan di layar, tidak semua elemen sekaligus.

Komponen *FlatList* membutuhkan dua alat peraga: data dan `renderItem`. Data adalah sumber informasi untuk daftar tersebut. `renderItem` mengambil satu item dari sumber dan mengembalikan komponen yang telah diformat untuk dirender.

Berikut contoh *FlatList* basic:

```
import React from 'react';
import { FlatList, StyleSheet, Text, View } from 'react-native';
const styles = StyleSheet.create({
  container: {
    flex: 1,
    paddingTop: 22
  },
  item: {
    padding: 10,
    fontSize: 18,
    height: 44,
  },
});
const FlatListBasics = () => {
  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>
);
}

export default FlatListBasics;

```

Jika ingin merender sekumpulan data yang dipecah menjadi beberapa bagian logis, mungkin dengan header bagian, mirip dengan yang ada *UITableView* di iOS, maka *SectionList* adalah cara yang tepat.

Berikut contoh *SectionList* basic:

```

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

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,
  },
});

const SectionListBasics = () => {
  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>
);
}

export default SectionListBasics;

```

1.2.7. Native Base

NativeBase adalah pustaka komponen UI sumber terbuka dan gratis untuk React Native guna membangun aplikasi seluler asli untuk platform iOS dan Android. Salah satu tujuan utama NativeBase 2.0 adalah membuatnya mudah untuk menentukan tema komponen dengan sedikit perubahan pada komponen itu sendiri.

Sintaks Umum Komponen NativeBase

```

import React, { Component } from 'react';
import { Container, Button, Text } from 'native-base';
export default class GeneralExample extends Component {
  render() {
    return (
      <Container>
        <Button>
          <Text>
            Button
          </Text>
        </Button>
      </Container>
    );
  }
}

```

1.2.8. Quick Setup Native Base

Untuk menjalankan komponen NativeBase ke aplikasi, yang perlu dilakukan hanyalah, membuat proyek React Native baru dan menginstal NativeBase menggunakan npm.

Persyaratan system:

- a. Node yang diinstal secara global ≥ 6.0
- b. Terpasang secara global npm ≥ 4.0
- c. React Native CLI yang diinstal secara global yang memungkinkan untuk membuat dan menginisialisasi proyek dengan mudah.

1.2.9. Setup With React Native

Membuat Aplikasi React Native Baru

```
react-native init AwesomeNativeBase  
cd AwesomeNativeBase
```

Instal NativeBase

```
npm install native-base --save
```

Instal Dependensi Peer Dependensi

Peer yang disertakan dari paket npm mana pun tidak terinstal secara otomatis. Aplikasi tidak akan bergantung padanya secara eksplisit.

```
react-native link
```

Setelah berhasil mengatur NativeBase dengan aplikasi React Native. Aplikasi React Native siap dijalankan di perangkat iOS dan Android.

BAB II

PEMBAHASAN

Pada modul kali ini, praktikan diinstruksikan untuk membuat Search List menggunakan React Native.

Pertama-tama, dibuat aplikasi baru dengan nama NativeBase.

```
D:\>cd praktikumreactnative  
D:\PraktikumReactNative>react-native init NativeBase
```

Gambar 2.1. Membuat Aplikasi Baru

Setelah aplikasi dibuat, kemudian install Native Base dengan mengetikkan perintah pada CMD sebagai berikut:

```
Np, install native-base --save
```

```
D:\>npm install native-base --save
```

Gambar 2.2. Menginstal Native Base

Setelah Native Base berhasil terinstal, kemudian hubungkan Native Base dengan aplikasi React Native yang telah dibuat sebelumnya dengan mengetikkan perintah pada CMD sebagai berikut:

```
React-native link
```

Pastikan sudah masuk kedalam folder aplikas yang dibuat, sebelum menghubungkannya.

```
D:\PraktikumReactNative>cd nativebase  
D:\PraktikumReactNative\NativeBase>react-native link
```

Gambar 2.3. Menghubungkan NativeBase dengan aplikasi React Native

Setelah berhasil menghubungkan NativeBase dengan aplikasi React Native. Aplikasi React Native siap dijalankan di perangkat iOS dan Android.

Setelah itu, masuk kedalam edit code. Berikut Code pada App.JS, untuk membuat Search List pada React Native.

```
1  import React, {Component} from 'react';
2  import {
3    Container,
4    Header,
5    Content,
6    Left,
7    Body,
8    Icon,
9    Text,
10   ListItem,
11   Thumbnail,
12   Input,
13   Item,
14 } from 'native-base';
```

Gambar 2.4. App.JS(1)

Pada potongan code diatas, pertama-tama di import component dari library React. Kemudian, di import juga Container, Header, Content, Left, Body, Icon, Text, ListItem, Thumbnail, Input, dan Item dari Library native-base, yang nantinya akan digunakan untuk membuat aplikasi search list.

```

18 let helperArray=require('./userList.json');
19 export default class app extends Component {
20   constructor(props){
21     super(props);
22     this.state = {
23       allUsers: helperArray,
24       usersFiltered: helperArray,
25     };
26   }

```

Gambar 2.5. App.JS(2)

Pada potongan code diatas, dideklarasikan variable scope helperArray secara global, dimana isi dari variable ini akan memanggil userList.json. kemudian terdapat export default yang digunakan untuk mengekspor kelas app. Didalam, export default ini, terdapat constructor, function seachUser, dan function Return.

Didalam constructor terdapat parameter Props, dimana props ini merujuk ke konstruktor kelas induk yaitu Component. Didalamnya juga terdapat super(props). Kemudian, terdapat juga this.state, dimana this.state mewakili nilai yang telah di render, yaiut apa yang saat ini ada dilayar. Didalam this.state ini, berisi variable helperArray yang didefinisikan kedalam allUsers, dan usersFiltered.

```

28 searchUser(textToSearch){
29   this.setState({
30     usersFiltered: this.state.allUsers.filter(i =>
31       i.name.toLowerCase().includes(textToSearch.toLowerCase()),
32     ),
33   });
34 }

```

Gambar 2.6. App.JS(3)

Pada potongan code diatas, dibuat method searchUser dengan inputan masukan berupa textToSearch. Didalam method ini, terdapat this.setState yang didalamnya

terdapat array `usersFiltered` yang dimana array ini akan menampilkan array sesuai dengan inputan text atau setelah difilter oleh method `searchUser`.

```

36   render(){
37     return (
38       <Container>
39         <Header searchBar rounded>
40           <Item>
41             <Icon name='search' />
42             <Input placeholder='Search user' onChangeText={text=>{
43               this.searchUser(text)}}/>
44           </Item>
45         </Header>
46         <Content>
47           {this.state.usersFiltered.map((item, index)=>(
48             <ListItem avatar>
49               <Left>
50                 <Thumbnail source={{uri: item.image}}/>
51               </Left>
52               <Body>
53                 <Text>{item.name}</Text>
54                 <Text note>{item.address}</Text>
55               </Body>
56             </ListItem>
57           ))}
58         </Content>
59       </Container>
60     );
61   }
62 }
63
64
65
66

```

Gambar 2.7. App.JS(4)

Pada potongan code diatas, terdapat method `render`, yang dimana `render` ini harus mengembalikan komponen Asli Bereaksi (elemen JSX) ke `render` (atau `null`, untuk tidak menghasilkan apa pun). Pada langkah ini, komponen UI diberikan. Didalam `render` ini dibuat tag `Container`, yang dimana didalam tag ini dibuat header berupa `searchBar` atau bar pencarian, dimana dalam tag ini, dibuat icon dengan tulisan `search`, dan juga berupa inputan dengan tulisan `search user`, dimana inputan ini akan diproses ke dalam method `searchUser` yang dibuat sebelumnya.

Kemudian, didalam tag content, akan menampilkan array userFiltered, yang telah difilter dari method searchUser sebelumnya. Yang akan ditampilkan, yaitu thumbnail berupa image di sisi kiri content. Pada bagian bodynya, terdapat text berupa name, dan address berupa note.

```

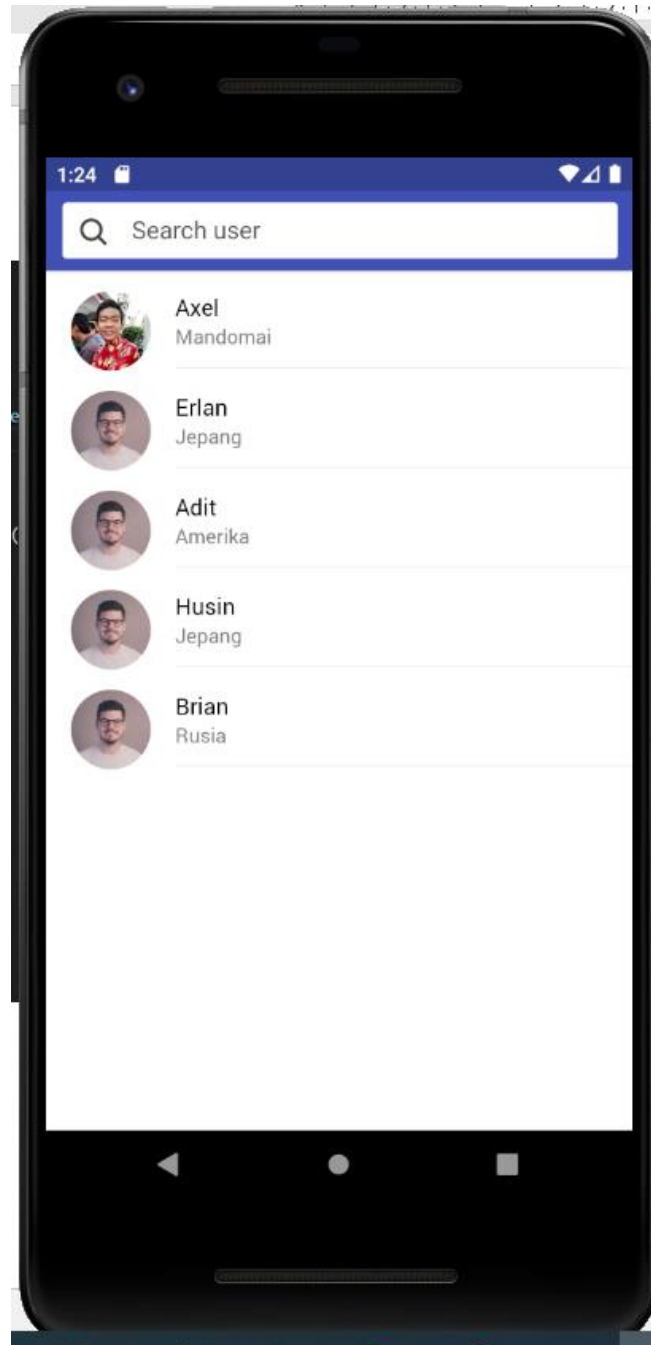
1
2  {
3    "name" : "Axel",
4    "address" : "Mandomai",
5    "image" : "https://scontent.fbdj4-1.fna.fbcdn.net/v/t1.6435-9/166525407_778989866058219_6164041
6  },
7
8  {
9    "name" : "Erlan",
10   "address" : "Jepang"
11  ,
12   "image" : "https://uifaces.co/our-content/donated/gPZwCbds.jpg"
13  },
14
15  {
16    "name" : "Adit",
17    "address" : "Amerika",
18    "image" : "https://uifaces.co/our-content/donated/gPZwCbds.jpg"
19  },
20  [
21    "name" : "Husin",
22    "address" : "Jepang",
23    "image" : "https://uifaces.co/our-content/donated/gPZwCbds.jpg"
24  ],
25  {
26    "name" : "Brian",
27    "address" : "Rusia",
28    "image" : "https://uifaces.co/our-content/donated/gPZwCbds.jpg"
29  }
30

```

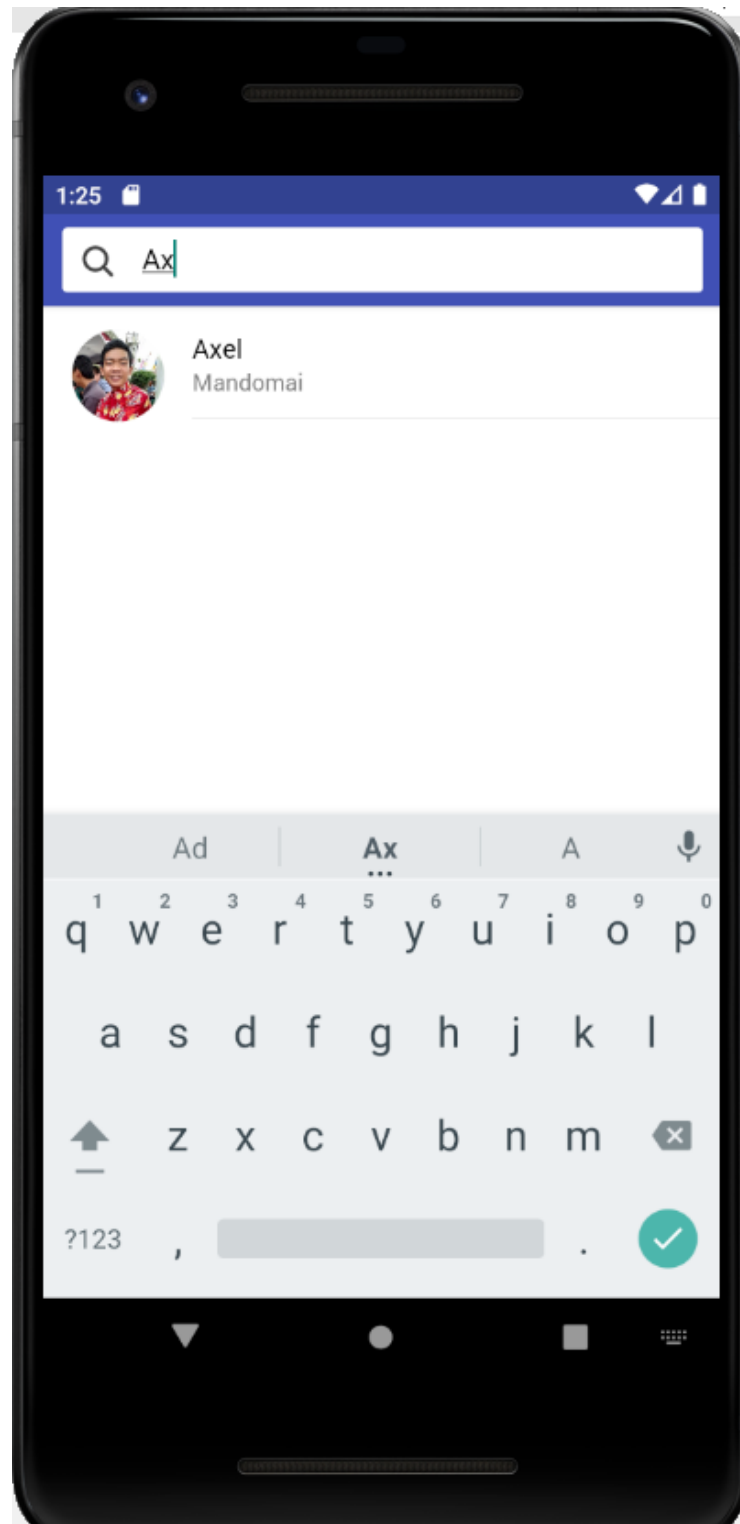
Gambar 2.8. userList.Json

File json diatas merupakan file yang digunakan untuk menyimpan data name, address, dan image yang nantinya akan digunakan untuk ditampilkan pada aplikasi.

Berikut hasil Output aplikasi pada emulator.



Gambar 2.9. Tampilan Aplikasi pada Emulator



Gambar 2.10. Tes Search User pada Aplikasi

BAB III

KESIMPULAN

Berdasarkan praktikum yang telah dilaksanakan, dapat disimpulkan bahwa NativeBase adalah pustaka komponen UI sumber terbuka dan gratis untuk React Native guna membangun aplikasi seluler asli untuk platform iOS dan Android. Untuk dapat menghubungkan Native Base dan React Native, pertama-tama install terlebih dahulu Native Base, kemudian hubungkan dengan mengetikkan react-native link, pada folder yang berisi aplikasi yang ingin dihubungkan dengan Native Base.

Search List digunakan untuk memfilter/menyaring data yang ingin diambil dari database dan kemudian hasil dari filter data tersebut ditampilkan, sesuai dengan hasil filter data yang dilakukan.

DAFTAR PUSTAKA

- Wiguna, P. D. A., Swastika, I. P. A., & Satwika, I. P. (2019). Rancang Bangun Aplikasi Point of Sales Distro Management System dengan Menggunakan Framework React Native. *Jurnal Nasional Teknologi Dan Sistem Informasi*, 4(3), 149–159. <https://doi.org/10.25077/teknosi.v4i3.2018.149-159>
- Anonymous. (2021). *Native Base IO*. Diakses pada tanggal 16 Mei 2021 di: <https://nativebase.io/>
- Anonymous. (2021). *Using List Views*. Diakses pada tanggal 14 Mei 2021 di: <https://reactnative.dev/docs/using-a-listview>
- Ismi, Trias. (2021). *React Native, Framework yang Jadi Topik Hangat di Kalangan Developer*. Diakses pada tanggal 9 Mei 2021 di: <https://glints.com/id/lowongan/react-native-adalah/#.YJfO8bUzZPY>
- Warcholinski, Matt. (2020). *What Is React Native? Is It Worth Using?*. Diakses pada tanggal 9 Mei 2021 di: <https://brainhub.eu/blog/what-is-react-native/>

LAMPIRAN

```
D:\>cd praktikumreactnative  
D:\PraktikumReactNative>react-native init NativeBase
```

Gambar 2.1. Membuat Aplikasi Baru

```
D:\>npm install native-base --save_
```

Gambar 2.2. Menginstal Native Base

```
D:\PraktikumReactNative>cd nativebase  
D:\PraktikumReactNative\NativeBase>react-native link
```

Gambar 2.3. Menghubungkan NativeBase dengan aplikasi React Native

```
1  import React, {Component} from 'react';  
2  import {  
3    Container,  
4    Header,  
5    Content,  
6    Left,  
7    Body,  
8    Icon,  
9    Text,  
10   ListItem,  
11   Thumbnail,  
12   Input,  
13   Item,  
14 } from 'native-base';
```

Gambar 2.4. App.JS(1)

```
18   let helperArray=require('./userList.json');
19   export default class app extends Component {
20     constructor(props){
21       super(props);
22       this.state = {
23         allUsers: helperArray,
24         usersFiltered: helperArray,
25       };
26     }
27   }
```

Gambar 2.5. App.JS(2)

```
28   searchUser(textToSearch){
29     this.setState({
30       usersFiltered: this.state.allUsers.filter(i =>
31         i.name.toLowerCase().includes(textToSearch.toLowerCase()),
32       ),
33     });
34   }
```

Gambar 2.6. App.JS(3)

```

36   render(){
37     return (
38       <Container>
39         <Header searchBar rounded>
40           <Item>
41             <Icon name='search' />
42             <Input placeholder='Search user' onChangeText={text=>{
43               this.searchUser(text)}} />
44           </Item>
45         </Header>
46         <Content>
47           {this.state.usersFiltered.map((item, index)=>(
48             <ListItem avatar>
49               <Left>
50                 <Thumbnail source={{uri: item.image}}/>
51               </Left>
52               <Body>
53                 <Text>{item.name}</Text>
54                 <Text note>{item.address}</Text>
55               </Body>
56             </ListItem>
57           ))}
58         </Content>
59       </Container>
60     );
61   }
62 }
63
64
65
66

```

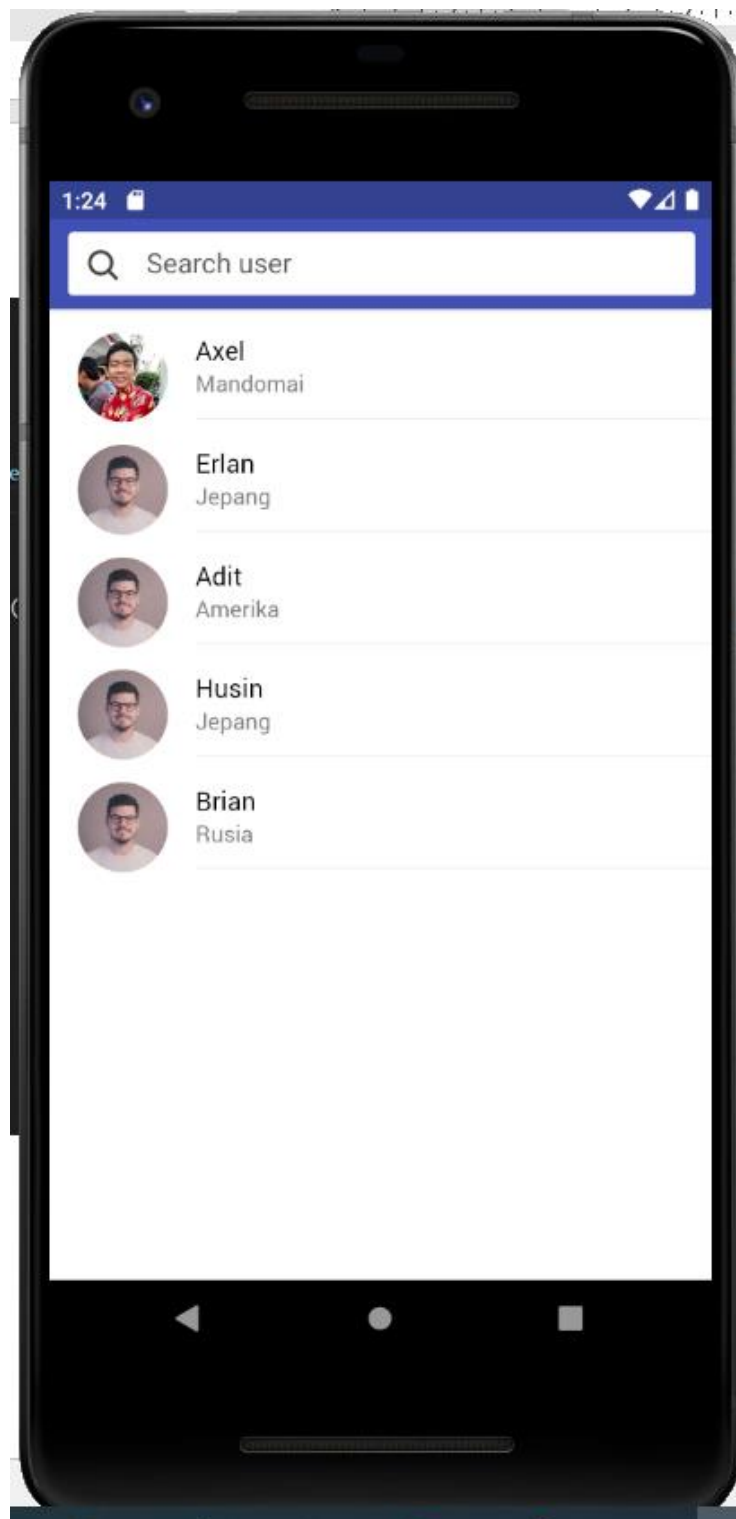
Gambar 2.7. App.JS(4)

```

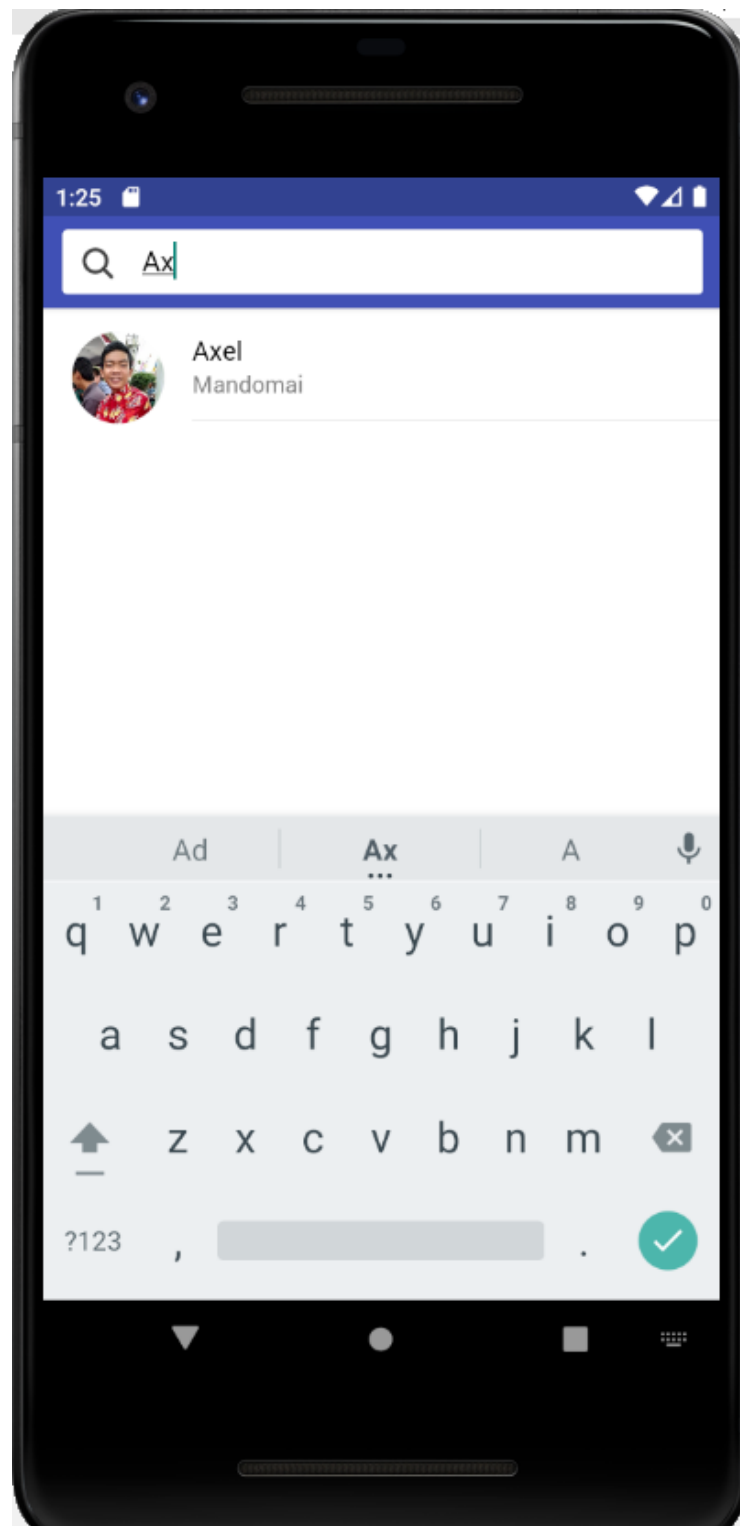
1
2  {
3    "name" : "Axel",
4    "address" : "Mandomai",
5    "image" : "https://scontent.fbdj4-1.fna.fbcdn.net/v/t1.6435-9/166525407_778989866058219_6164041
6  },
7
8  {
9    "name" : "Erlan",
10   "address" : "Jepang"
11   ,
12   "image" : "https://uifaces.co/our-content/donated/gPZwCbds.jpg"
13  },
14
15  {
16    "name" : "Adit",
17    "address" : "Amerika",
18    "image" : "https://uifaces.co/our-content/donated/gPZwCbds.jpg"
19  },
20  [
21    "name" : "Husin",
22    "address" : "Jepang",
23    "image" : "https://uifaces.co/our-content/donated/gPZwCbds.jpg"
24  ],
25  {
26    "name" : "Brian",
27    "address" : "Rusia",
28    "image" : "https://uifaces.co/our-content/donated/gPZwCbds.jpg"
29  }
30

```

Gambar 2.8. userList.Json



Gambar 2.9. Tampilan Aplikasi pada Emulator



Gambar 2.10. Tes Search User pada Aplikasi