

**LAPORAN HASIL PRAKTIKUM  
PEMROGRAMAN WEB DAN MOBILE**



**Nama** : Ahmad Daffa Fahrezi  
**NIM** : 193010503008  
**Kelas** : C  
**Modul** : VI ( React Native Search )

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

**2021**

## BAB I

### TUJUAN DAN LANDASAN TEORI

#### 1.1. Tujuan

Tujuan dari menyelesaikan modul ini, diharapkan dapat:

1.1.1. Mahasiswa dapat memahami tentang React Native

1.1.2. Mahasiswa dapat memahami cara install React Native.

#### 1.2. Landasan Teori

1.2.1. Pengertian React Native.

(DewaWeb, 2021) Pada dasarnya *react native* adalah sebuah *framework* dari Javascript yang banyak dikembangkan oleh perusahaan besar dalam bidang teknologi seperti Facebook. Jadi, kamu tidak perlu membuat aplikasi *hybrid*.

*React native* juga mampu mengkompilasi aplikasi ke dalam sebuah *native code* pada Android maupun iOS. Ulasan kali ini akan membahas mengenai apa saja keunggulan yang ada dalam *react native* dan bagaimana cara menerapkannya dalam membuat sebuah aplikasi.

1.2.2. Keunggulan React Native.

(Sekawanmedia, 2020) *React native* digunakan oleh beberapa perusahaan besar di dunia. Sebut saja mulai dari perusahaan kelas Tesla, Walmart, hingga Instagram. Selain itu, Facebook juga menggunakannya begitu pula dengan Airbnb, Baidu, dan Gyroscope.

*React native* memberikan kemudahan bagi para pemula, khususnya bagi kamu yang belum begitu mahir soal *coding*. Kamu bisa melihat tutorial yang ada dan tutorial ini akan dibawakan dengan gaya yang mudah dipahami baik itu oleh *developer* dan beberapa praktisi lain di bidangnya.

Dalam proses pembuatan, aplikasi *ini* memiliki keunggulan, yakni lebih cepat dan lebih efisien di dalam pengeluaran. Di sini akan ada banyak *template* dengan cara *copy paste* dari kode tersebut dan kamu langsung bisa melihat hasilnya.

Keunggulan lainnya adalah sangat cocok digunakan untuk *startup*. Hal ini karena dibangun dengan Javascript sehingga sangat mudah ketika diprogram. Selain itu, dalam proses pembuatannya juga bisa *cross platform*.

### 1.2.3. Cara kerja React Native.

(Glints, 2021)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.

## **BAB II**

### **PEMBAHASAN**

Pada Modul 6 ini praktikkan diminta untuk menggunakan fitur search pada react native.

Pada langkah pertama adalah memulai react native dengan cara membuka cmd lalu mengetikan “npx react-native run-android”.

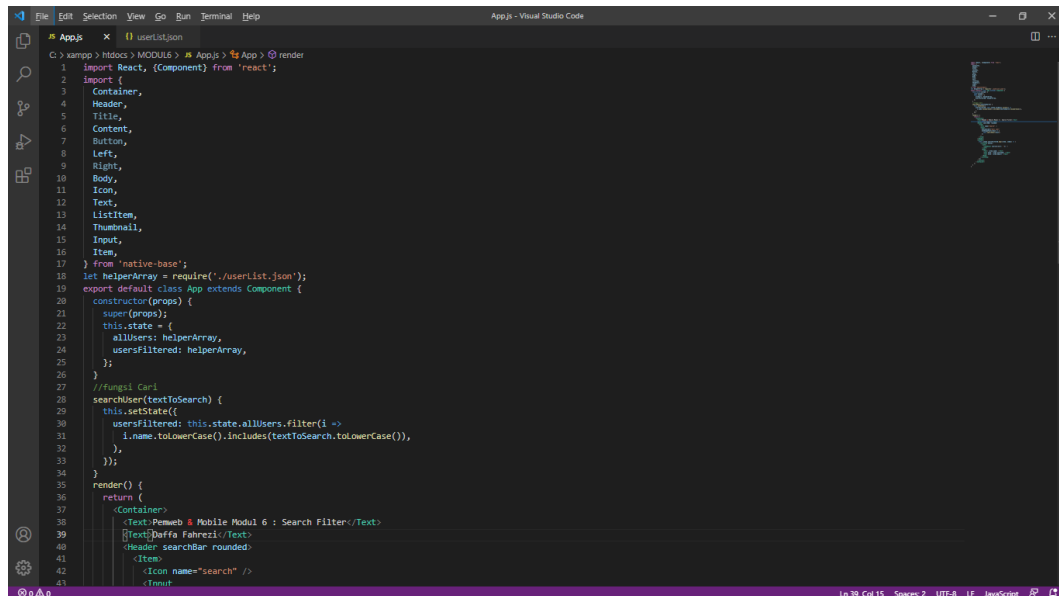
Berikut adalah gambarnya



**Gambar 2.1. tampilan awal**

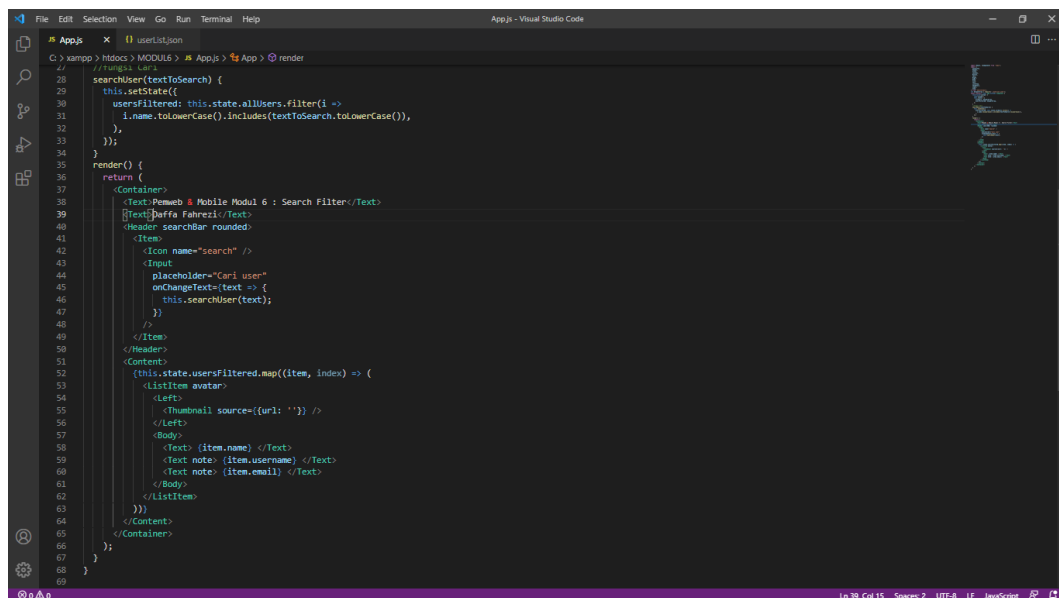
Selanjutnya, langkah kedua adalah membuat tampilan search dan mengimport beberapa library. Setelah itu menambahkan variable elperArray yang berisi array data-data yang akan ditampilkan.

Berikut adalah gambar Code App.js dan Userlist.json



```
1 import React, {Component} from 'react';
2 import {
3   Container,
4   Header,
5   Title,
6   Content,
7   Button,
8   Left,
9   Right,
10  Body,
11  Icon,
12  Text,
13  ListItem,
14  Thumbnail,
15  Input,
16  Item,
17 } from 'native-base';
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   //fungsi Cari
28   searchUser(textToSearch) {
29     this.setState({
30       usersFiltered: this.state.allUsers.filter(i =>
31         i.name.toLowerCase().includes(textToSearch.toLowerCase()),
32       ),
33     });
34   }
35   render() {
36     return (
37       <Container>
38         <Text>Pemweb & Mobile Modul 6 : Search Filter</Text>
39         <Text>Daffa Fahrezi</Text>
40         <Header searchBar rounded>
41           <Item>
42             <Icon name="search" />
43             <Input
```

Gambar 2.2. App Code\_1

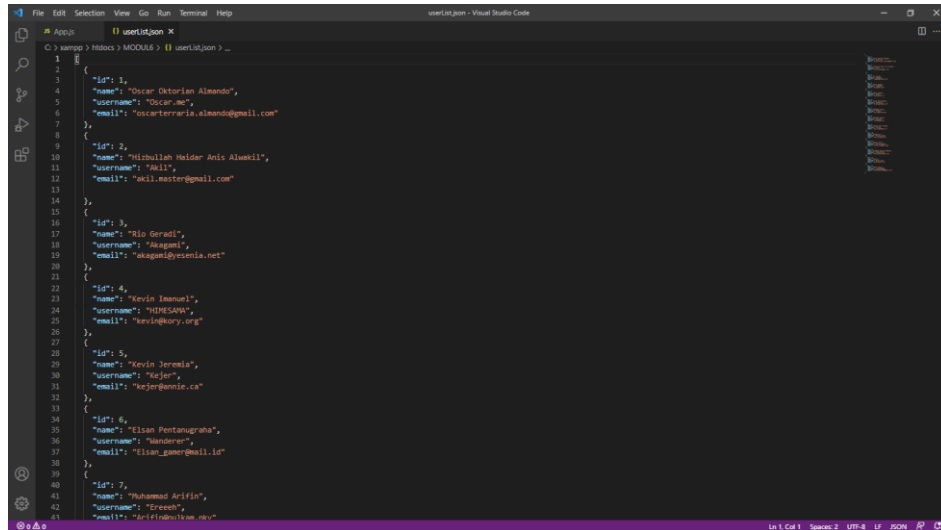


```
28 searchUser(textToSearch) {
29   this.setState({
30     usersFiltered: this.state.allUsers.filter(i =>
31       i.name.toLowerCase().includes(textToSearch.toLowerCase()),
32     ),
33   });
34 }
35 render() {
36   return (
37     <Container>
38       <Text>Pemweb & Mobile Modul 6 : Search Filter</Text>
39       <Text>Daffa Fahrezi</Text>
40       <Header searchBar rounded>
41         <Item>
42           <Icon name="search" />
43           <Input
44             placeholder="Cari user"
45             onChangeText={text => {
46               this.searchUser(text);
47             }}
48           />
49         </Item>
50       </Header>
51       <Content>
52         {this.state.usersFiltered.map((item, index) => (
53           <ListItem avatar>
54             <Left>
55               <Thumbnail source={{url: ''}} />
56             </Left>
57             <Body>
58               <Text>{item.name}</Text>
59               <Text note={item.username}</Text>
60               <Text note={item.email}</Text>
61             </Body>
62           </ListItem>
63         ))}
64       </Content>
65     </Container>
66   );
67 }
68 }
```

Gambar 2.3. App Code\_2

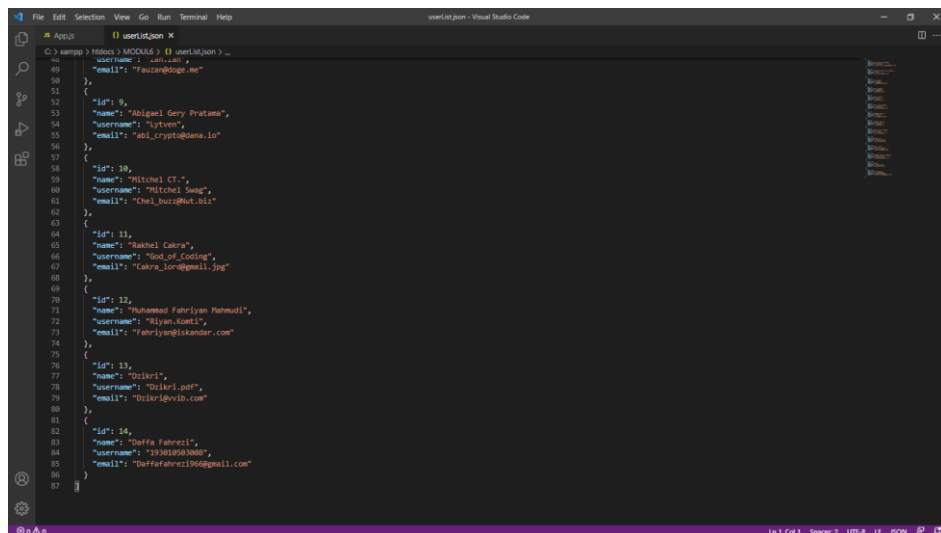
Pada source code diatas terdapat fungsi search yang berfungsi sebagai filter data-data yang terdapat dalam data user list

Berikut gambar data usernya :



```
1 {
2   "id": 1,
3   "name": "Oscar Oktorlan Alambdo",
4   "username": "Oscar.me",
5   "email": "oscarterraria.alambdo@gmail.com"
6 },
7 {
8   "id": 2,
9   "name": "Hizbullah Halder Anis Aluskill",
10  "username": "Akili",
11  "email": "akil.sartor@gmail.com"
12 },
13 {
14   "id": 3,
15   "name": "Rlo Geradi",
16   "username": "Wagani",
17   "email": "wagani@vsnia.net"
18 },
19 {
20   "id": 4,
21   "name": "Kevin Immanuel",
22   "username": "HIDEJAWA",
23   "email": "kevin@ury.org"
24 },
25 {
26   "id": 5,
27   "name": "Kevin Jeremia",
28   "username": "Kajfer",
29   "email": "kejerjeremia.ca"
30 },
31 {
32   "id": 6,
33   "name": "Elsan Pentanugraha",
34   "username": "Wandorer",
35   "email": "elsan_gamer@gmail.id"
36 },
37 {
38   "id": 7,
39   "name": "Muhammad Arifin",
40   "username": "Ereesh",
41   "email": "arifin@vsnia.com"
42 }
```

Gambar 2.4. Userlist\_1



```
43 {
44   "id": 8,
45   "name": "Fauzan@doge.me",
46   "username": "Fauzan",
47   "email": "Fauzan@doge.me"
48 },
49 {
50   "id": 9,
51   "name": "Miguel Gery Pratana",
52   "username": "lytven",
53   "email": "mbl_crypto@dana.io"
54 },
55 {
56   "id": 10,
57   "name": "Mitchel Cg",
58   "username": "Mitchel Soag",
59   "email": "Chel_buzz@ut.hiz"
60 },
61 {
62   "id": 11,
63   "name": "Rahel Cakra",
64   "username": "Host_of_Coding",
65   "email": "Cakra_lor@gmail.jpg"
66 },
67 {
68   "id": 12,
69   "name": "Muhammad Fahriyan Rahmudi",
70   "username": "Rlyan_Koriti",
71   "email": "Fahriyankitander.com"
72 },
73 {
74   "id": 13,
75   "name": "Dikriti",
76   "username": "Dikriti.pdf",
77   "email": "Dikriti@vsn.com"
78 },
79 {
80   "id": 14,
81   "name": "Daffa Fahrezil",
82   "username": "DaffaDaffa",
83   "email": "Daffafahrezil90@gmail.com"
84 },
85 }
```

Gambar 2.5. Userlist\_2

Setelah terbuat userlist data seperti diatas maka fitur dari search pada react native dapat digunakan.

Berikut contoh gambarnya :

Cari user

Oscar Oktorian Almando

Oscar.me

oscarterraria.almando@gmail.com

Hizbullah Haidar Anis Alwakil

Akil

akil.master@gmail.com

Rio Geradi

Akagami

akagami@yesenia.net

Kevin Imanuel

HIMESAMA

kevin@ikory.org

Kevin Jeremia

Kejer

kejer@annie.ca

Elsan Pentanugraha

Wanderer

Elsan\_gamer@mail.id

Muhammad Arifin

Ereesh

Arifing@pulkam.pky

Gambar 2.6. Hasil

### **BAB III**

#### **KESIMPULAN**

Dari partikum modul 6 ini dapat diambil kesimpulan yakni didalam rezt native terdapat fitur Search, sebuah program yang mana program tersebut memerlukan sebuah fitur yang namanya pencarian untuk memudahkan pengguna dalam mencari user dengan menggunakan fitur search tersebut.



## DAFTAR PUSTAKA

DewaWeb. (2021). *Apa itu react native*. 2021. <https://www.dewaweb.com/blog/apa-itu-react-native/>

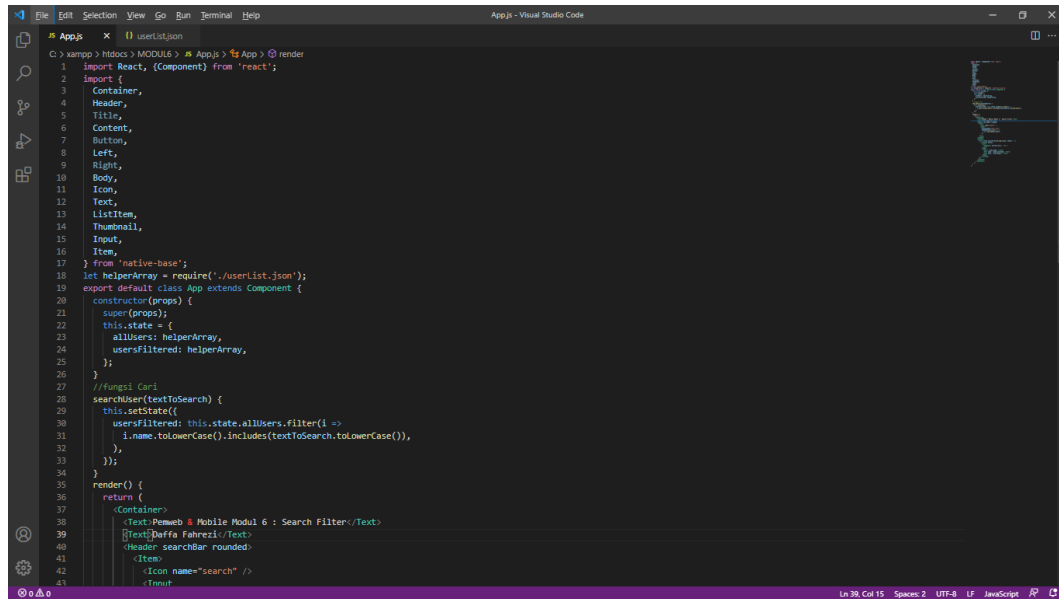
Glints. (2021). *react native adalah*. 2021. <https://glints.com/id/lowongan/react-native-adalah/#.YJj1E6GyS00>

Sekawanmedia. (2020). *keunggulan react*. 2020. <https://www.sekawanmedia.co.id/apa-itu-react-native/>

## LAMPIRAN

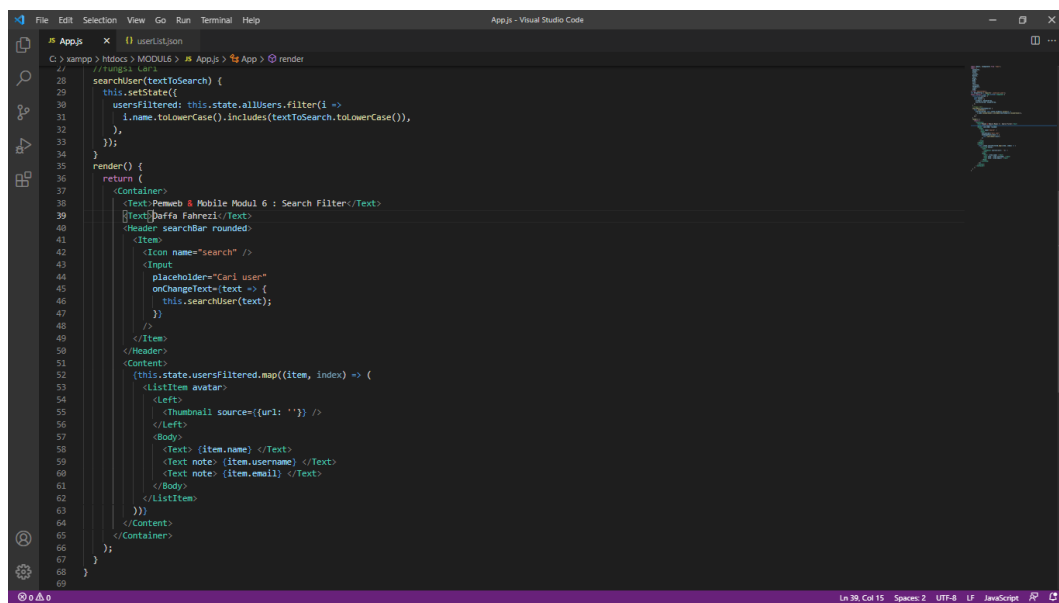


**Gambar 2.1. tampilan awal**



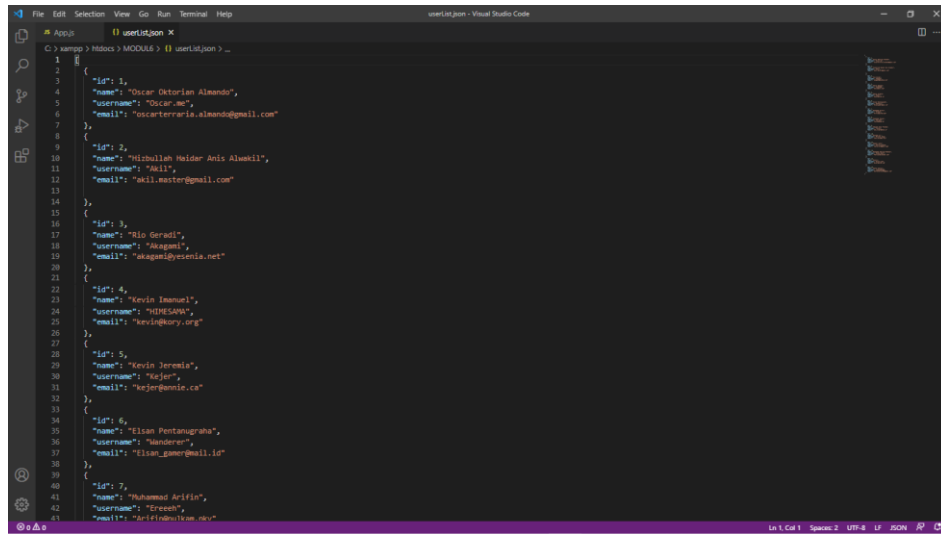
```
1 import React, {Component} from 'react';
2 import {
3   Container,
4   Header,
5   Title,
6   Content,
7   Button,
8   Left,
9   Right,
10  Body,
11  Icon,
12  Text,
13  ListItem,
14  Thumbnail,
15  Input,
16  Item,
17 } from 'native-base';
18 let helperArray = require('./userList.json');
19 export default class App extends Component {
20   constructor(props) {
21     super(props);
22     this.state = {
23       allUsers: helperArray,
24       usersFiltered: helperArray,
25     };
26   }
27   //fungsi Cari
28   searchUser(textToSearch) {
29     this.setState({
30       usersFiltered: this.state.allUsers.filter(i =>
31         i.name.toLowerCase().includes(textToSearch.toLowerCase()),
32       ),
33     });
34   }
35   render() {
36     return (
37       <Container>
38         <Text>Pemweb & Mobile Modul 6 : Search Filter</Text>
39         <Text>Daffa Fahreri</Text>
40         <Header searchBar rounded>
41           <Item>
42             <Icon name="search" />
43             <Input
```

Gambar 2.2. App Code\_1



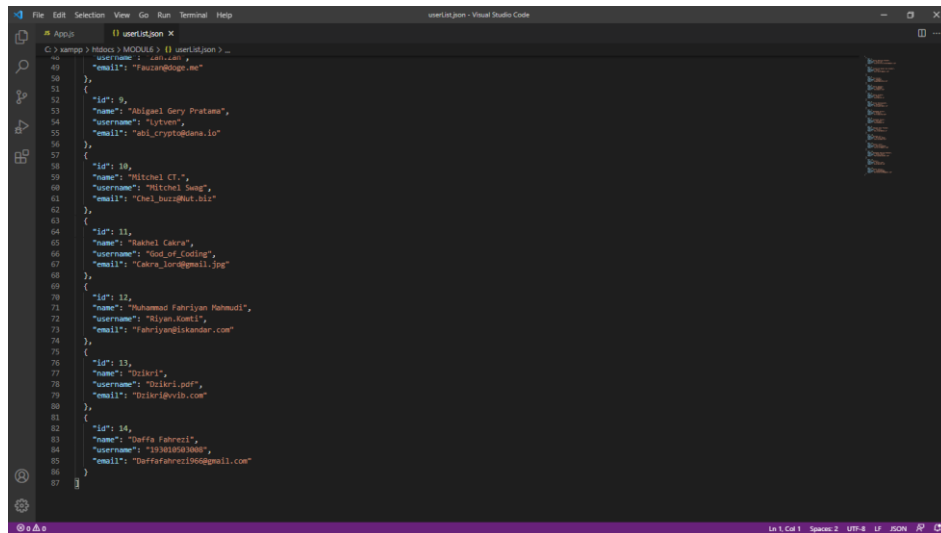
```
44     placeholder="Cari user"
45     onChange={text => {
46       this.searchUser(text);
47     }}
48   />
49 </Item>
50 </Header>
51 <Content>
52   <Listitem avatar>
53     <Left>
54       <Thumbnail source={url: ''} />
55     </Left>
56     <Body>
57       <Text> {item.name} </Text>
58       <Text note={item.username} </Text>
59       <Text note={item.email} </Text>
60     </Body>
61   </Listitem>
62 </Content>
63 </Container>
64 </Container>
65 </Container>
66 </Container>
67 </Container>
68 </Container>
69 </Container>
```

Gambar 2.3. App Code\_2



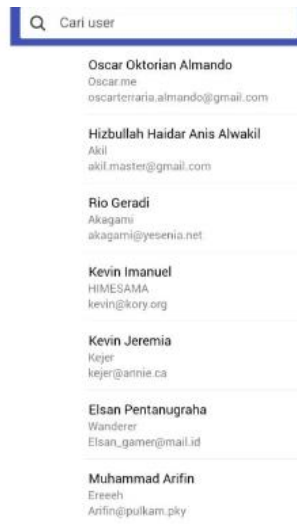
```
1 {
2   {
3     "id": 1,
4     "name": "Oscar Oktavian Almando",
5     "username": "Oscar.me",
6     "email": "oscarterraria.almand@gmail.com"
7   },
8   {
9     "id": 2,
10    "name": "Muhammad Heider Anis Alwakil",
11    "username": "Akil",
12    "email": "akil.warter@gmail.com"
13  },
14  {
15    "id": 3,
16    "name": "Rio Geradi",
17    "username": "Magami",
18    "email": "magami@ycomia.net"
19  },
20  {
21    "id": 4,
22    "name": "Kevin Emanuel",
23    "username": "KINEMAN",
24    "email": "kavin@ury.org"
25  },
26  {
27    "id": 5,
28    "name": "Kevin Jeremia",
29    "username": "Kajer",
30    "email": "kejerjeremia.ca"
31  },
32  {
33    "id": 6,
34    "name": "Elsan Pentanugraha",
35    "username": "Mandorer",
36    "email": "elsan_pemer@gmail.io"
37  },
38  {
39    "id": 7,
40    "name": "Muhammad Arifin",
41    "username": "Ereesh",
42    "email": "arifin@rice.riv"
43  }
44 }
```

Gambar 2.4. Userlist\_1



```
44 {
45   "username": "satikan",
46   "email": "Fauzan@dogz.me"
47 },
48 {
49   "id": 9,
50   "name": "Miguel Gery Pratama",
51   "username": "Tipton",
52   "email": "ab1.crypt@dana.io"
53 },
54 {
55   "id": 10,
56   "name": "Mitchel Cf.",
57   "username": "Mitchel Swag",
58   "email": "Chel_buzz@ut.hiz"
59 },
60 {
61   "id": 11,
62   "name": "Mahdi Cakra",
63   "username": "God of Coding",
64   "email": "Cakra_lor@gmail.jpg"
65 },
66 {
67   "id": 12,
68   "name": "Muhammad Fahriyan Fahmudi",
69   "username": "Rilyan.Kusli",
70   "email": "Fahriyan@ikander.com"
71 },
72 {
73   "id": 13,
74   "name": "Dikari",
75   "username": "Dikari.pdf",
76   "email": "Dikari@vish.com"
77 },
78 {
79   "id": 14,
80   "name": "Doffa fahrezil",
81   "username": "Doffa fahrezil",
82   "email": "Duffafahrezil90@gmail.com"
83 }
84 }
```

Gambar 2.5. Userlist\_2



Gambar 2.6. Hasil