

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB DAN MOBILE**



NAMA : ABIGael GERRY PRATAMA
NIM : 193020503021
KELAS : A
MODUL : VI (REACT NATIVE)

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

BAB I

LANDASAN TEORI

1.1. Landasan Teori

React Native adalah kerangka aplikasi seluler sumber terbuka yang dibuat oleh Facebook, Inc. Ini digunakan untuk mengembangkan aplikasi untuk Android, Android TV, iOS, macOS, tvOS, Web, Windows, dan UWP dengan memungkinkan pengembang menggunakan kerangka kerja React bersama dengan platform asli kemampuan. Dalam membuat sebuah aplikasi, Anda perlu belajar bahasa pemrograman yang spesifik atau khusus (disebut juga dengan ‘Native’) untuk platform tersebut. Hal ini dikarenakan setiap sistem operasi memiliki bahasa pemrograman yang berbeda-beda.

Untuk membuat aplikasi di Android, Anda butuh belajar bahasa pemrograman Java. Sedangkan untuk membuat aplikasi di iOS, Anda perlu belajar menggunakan bahasa pemrograman Objective C atau Swift. Namun dengan menggunakan React Native sebagai framework, Anda tidak perlu lagi melakukan hal spesifik tersebut. Alih-alih menggunakan Java atau Objective C maupun Swift, Anda hanya butuh belajar bahasa pemrograman Javascript untuk bisa membuat aplikasi dengan menggunakan platform ini. Javascript sendiri biasa digunakan oleh para developer dalam membuat website. Jadi, dengan menggunakan platform ini, Anda hanya perlu membuat satu aplikasi. Anda tidak perlu lagi membuat banyak versi aplikasi yang compatible dengan sistem operasi yang ada.

Flatlist React Native seperti tampilan daftar yang digunakan untuk menyimpan item dalam daftar dan menyediakan fitur penting seperti menggulir secara horizontal dan vertikal. Flatlist di react native dirancang untuk menangani kumpulan data besar yang mungkin tidak muat di layar perangkat. Daftar data ini melakukan rendering hanya item yang sedang ditampilkan di layar dan tidak semua item.

Flatlist menyediakan beberapa fitur, yaitu:

1. Pull to refresh support.
2. Header support.
3. Footer support.
4. Separator support.
5. Scroll loading support.
6. ScrollToIndex support.
7. Cross-Platform support.
8. Configurable callbacks.
9. Optional support for horizontal mode.

BAB II

PEMBAHASAN

Project ini diberikan dengan nama modul6, dengan membuat project dari cmd dengan koder react-native init modul6. Setelah itu menghubungkan hp dengan pc menggunakan kabel konektor usb dengan mode debug. Setelah itu mengecek di CMD apakah sudah terdeteksi dengan cara ketik dengan kode adb devices. Lalu menjalankan react native di cmd dengan kode react-native run-android.

Lalu membuat kode program pada app.js dan nama.json untuk membuat list-nya. Berikut kode program untuk menjalankan flatlist react native.

```
import React, {Component} from 'react';
import{
  Container,
  Header,
  Content,
  Left,
  Right,
  Body,
  Icon,
  Text,
  ListItem,
  Item,
  Input,
} from 'native-base';
let helperArray = require('./nama.json');
export default class App extends Component {
  constructor(props){
    super(props);
    this.state = {
      allUsers: helperArray,
      usersFiltered:helperArray,
    };
  }

  searchUser(textToSearch) {
    this.setState({
      usersFiltered: this.state.allUsers.filter(i =>
        i.name.toLowerCase().includes(textToSearch.toLowerCase()),
      ),
    });
  }

  render() {
    return (
      <Container>
        <Header searchBar rounded>
          <Item>
            <Icon name="search"/>
            <Input
              placeholder="Search User"
              onChangeText={text => {
                this.searchUser(text);
              }}
            />
          </Item>
        </Header>
      </Container>
    );
  }
}
```

```

        </Item>
      </Header>
    <Content>
      {this.state.usersFiltered.map((item,index)=>(
        <ListItem>
          <Left></Left>
          <Body>
            <Text>{item.name}</Text>
            <Text>{item.address}</Text>
          </Body>
          <Right />
        </ListItem>
      ))}
    </Content>
  </Container>
);
}
}

```

Gambar 2.1 Code app.js

Pada fungsi searchUser digunakan untuk melakukan pencarian data tertentu pada array yang disimpan dalam file nama.json. Pencarian dilakukan berdasarkan variabel “name” dengan memfilter pencarian yang merubah semua nilai string menjadi bentuk huruf kecil dengan fungsi toLowerCase(). Dan berikut data yang dimasukan pada nama.json.

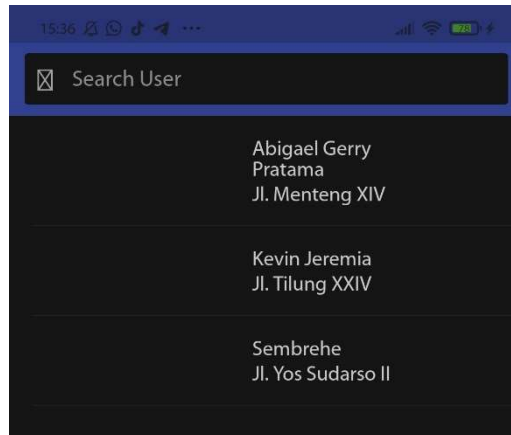
```

[
  {
    "name": "Abigael Gerry Pratama",
    "address": "Jl. Menteng XIV"
  },
  {
    "name": "Kevin Jeremia",
    "address": "Jl. Tilung XXIV"
  },
  {
    "name": "Sembrehe",
    "address": "Jl. Yos Sudarso II"
  }
]

```

Gambar 2.2 Code nama.json

Berikut adalah tampilan program saan dijalankan.



Gambar 2.3 Output Program

BAB III

KESIMPULAN

Flatlist React Native seperti tampilan daftar yang digunakan untuk menyimpan item dalam daftar dan menyediakan fitur penting seperti menggulir secara horizontal dan vertikal. Flatlist di react native dirancang untuk menangani kumpulan data besar.

BAB IV
DAFTAR PUSTAKA

Singh, Er, Harinder. 2020. #11 Search Filter For Lists In React Native.
<https://www.youtube.com/watch?v=jXgc6ctpEpo>.

Reactnative. 2021. Flatlist – React Native.
<https://reactnative.dev/docs/flatlist>.

How to use the FlatList Component — React Native Basics. 2019. Spencer Carli.
<https://medium.com/react-native-development/how-to-use-the-flatlist-component-react-native-basics-92c482816fe6>

LAMPIRAN

```
import React, {Component} from 'react';
import{
  Container,
  Header,
  Content,
  Left,
  Right,
  Body,
  Icon,
  Text,
  ListItem,
  Item,
  Input,
} from 'native-base';
let helperArray = require('./nama.json');
export default class App extends Component {
  constructor(props){
    super(props);
    this.state = {
      allUsers: helperArray,
      usersFiltered:helperArray,
    };
  }

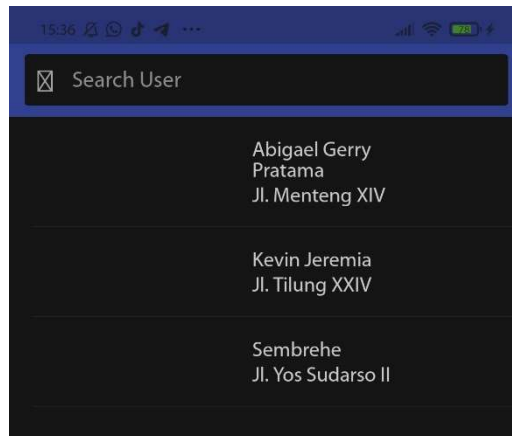
  searchUser(textToSearch) {
    this.setState({
      usersFiltered: this.state.allUsers.filter(i =>
        i.name.toLowerCase().includes(textToSearch.toLowerCase()),
      ),
    });
  }

  render() {
    return (
      <Container>
        <Header searchBar rounded>
          <Item>
            <Icon name="search"/>
            <Input
              placeholder="Search User"
              onChangeText={text => {
                this.searchUser(text);
              }}
            />
          </Item>
        </Header>
        <Content>
          {this.state.usersFiltered.map((item,index)=>(
            <ListItem>
              <Left></Left>
              <Body>
                <Text>{item.name}</Text>
                <Text>{item.address}</Text>
              </Body>
              <Right />
            </ListItem>
          ))}
        </Content>
      </Container>
    );
  }
}
```

Gambar 2.1 Code app.js

```
[
  {
    "name": "Abigael Gerry Pratama",
    "address": "Jl. Menteng XIV"
  },
  {
    "name": "Kevin Jeremia",
    "address": "Jl. Tilung XXIV"
  },
  {
    "name": "Sembrehe",
    "address": "Jl. Yos Sudarso II"
  }
]
```

Gambar 2.2 Code nama.json



Gambar 2.3 Output Program