

LAPORAN HASIL PRAKTIKUM
PEMROGRAMAN WEBSITE DAN MOBILE I



NAMA : MUHAMMAD ARIFIN
NIM : 193020503025
KELAS : A
MODUL : VI

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA

2021

BAB I

TUJUAN DAN LANDASAN TEORI

1. Tujuan Praktikum

- 1.1.** Mahasiswa mampu membuat search filter for flat list pada react native.

2. Landasan Teori

2.1. Flatlist

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 bekerja pada poros utama berikut

- **Data:** Ini adalah array data yang berisi item individual yang akan digunakan untuk membuat flatlist di react native.
- **RenderItem:** Ini adalah fungsi yang tujuannya adalah untuk mengambil item individual dari array dan mengubahnya menjadi bagian dari daftar data react-native.

Flatlist juga menyediakan beberapa fitur sebagai berikut :

- Pull to refresh support.
- Header support.
- Footer support.
- Separator support.
- Scroll loading support.
- ScrollToIndex support.
- Cross-Platform support.
- Configurable callbacks.
- Optional support for horizontal mode.

BAB II

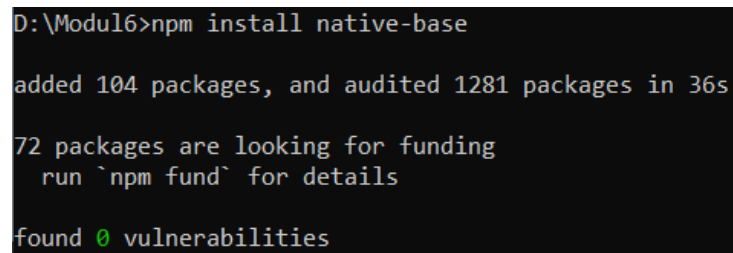
PEMBAHASAN

Tugas Praktikum

1. Implementasi search filter for flat list pada react native.

Pembahasan

Pertama-tama adalah membuat sebuah project baru, untuk membuat project baru diperlukan perintah react-native init “nama_project”. Setelah project selesai dibuat, project tersebut akan diinstal native-base, berikut adalah instalasi native-base pada project yang telah dibuat sebelumnya.



```
D:\Modul6>npm install native-base
added 104 packages, and audited 1281 packages in 36s
72 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
```

Gambar 2.1 Install native-base.

Selanjutnya memberikan perintah “react-native run-android” untuk menjalankan android pada AVD. Setelah android berhasil dijalankan, seluruh sintaks pada file app.js dihapus dan diganti dengan sintaks sebagai berikut.

Bagian import berfungsi agar dapat menggunakan berbagai komponen yang terdapat pada library react dan juga library native-base.

Selain bagian import, terdapat pula sintaks “require(‘./userList.json’);” yang berarti program memerlukan file bernama userList dengan ekstensi json untuk diproses oleh method helperArray.

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

Gambar 2.2 Import.

Selanjutnya, terdapat fungsi `searchUser`. Fungsi ini digunakan untuk melakukan pencarian data tertentu pada array yang disimpan dalam file `userList.json`. pencarian dilakukan berdasarkan variabel “name” dengan filter pencarian yang merubah semua nilai string menjadi bentuk huruf kecil dengan adanya `toLowerCase()`.

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

Gambar 2.3 fungsi `searchUser`.

Bagian terakhir pada file `app.js` adalah fungsi `render` yang digunakan untuk menampilkan semua data dalam array pada file `userList.json` dengan tampilan list. Pada bagian ini juga dibuat kotak pencarian yang digunakan untuk melakukan pencarian terhadap array tersebut.

```

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 />
            <Body>
              <Text>{item.name}</Text>
              <Text>{item.number}</Text>
            </Body>
            <Right />
          </ListItem>
        ))}
      </Content>
    </Container>
  );
}
}

```

Gambar 2.4 fungsi render.

Pada file userList.json, file ini berfungsi menyimpan data dalam bentuk array yang dipanggil pada file app.js. Agar bisa ditampilkan, dibuat beberapa variabel yang akan dipanggil pada fungsi render.

Berikut adalah data yang disimpan pada file userList.json.

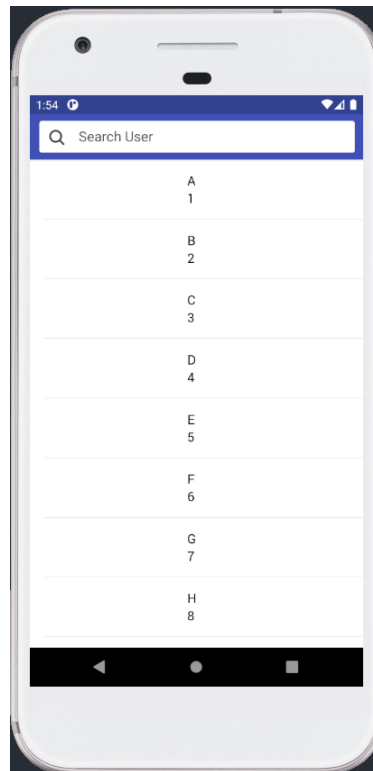
```
{
  {
    "name": "A",
    "number": "1"
  },
  {
    "name": "B",
    "number": "2"
  },
  {
    "name": "C",
    "number": "3"
  },
  {
    "name": "D",
    "number": "4"
  },
  {
    "name": "E",
    "number": "5"
  },
  {
    "name": "F",
    "number": "6"
  },
  {
    "name": "G",
    "number": "7"
  },
  {
    "name": "H",
    "number": "8"
  },
}
```

Gambar 2.5 file userList.json.

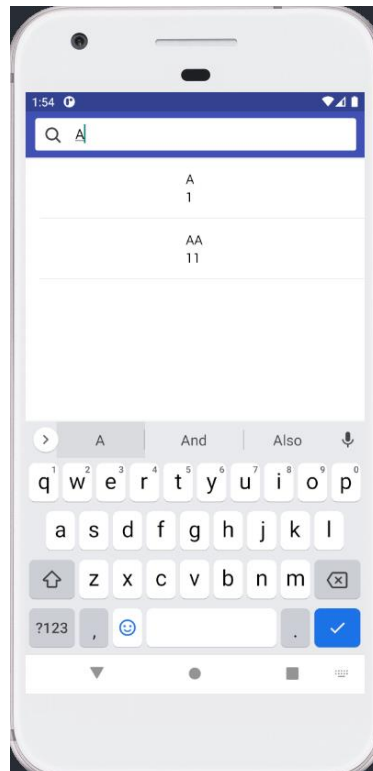
```
{
  "name": "AA",
  "number": "11"
},
{
  "name": "BB",
  "number": "22"
},
{
  "name": "CC",
  "number": "33"
},
{
  "name": "DD",
  "number": "44"
},
{
  "name": "EE",
  "number": "55"
},
{
  "name": "FF",
  "number": "66"
},
{
  "name": "GG",
  "number": "77"
},
{
  "name": "HH",
  "number": "88"
}
```

Gambar 2.6 file userList.json.

Berikut adalah output dari hasil pengkodean yang telah dilakukan sebelumnya.



Gambar 2.7 Output.



Gambar 2.8 Output.

BAB III

KESIMPULAN

Filter digunakan untuk melakukan penyaringan informasi atau data yang tersedia. Salah satu contoh implementasi dari filter adalah filter yang digunakan pada pencarian. Filter akan mencari data tertentu sesuai dengan apa yang ingin dicari oleh pengguna lalu mengabaikan data yang lain.

DAFTAR PUSTAKA

Pedamkar, Priya, 2020. React Native Flatlist.

<https://www.educba.com/react-native-flatlist/>. (Di akses pada 17 Mei 2021).

Reactnative, 2021. Flatlist – React Native.

<https://reactnative.dev/docs/flatlist>. (Di akses pada 17 Mei 2021).

Singh, Er, Harinder, 2020. #11 Search Filter For Lists In React Native.

<https://www.youtube.com/watch?v=jXgc6ctpEpo>. (Di akses pada 17 Mei 2021).

LAMPIRAN

```
D:\Modul6>npm install native-base

added 104 packages, and audited 1281 packages in 36s

72 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Gambar 2.1 Install native-base.

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

Gambar 2.2 Import.

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

Gambar 2.3 fungsi searchUser.

```

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 />
            <Body>
              <Text>{item.name}</Text>
              <Text>{item.number}</Text>
            </Body>
            <Right />
          </ListItem>
        ))}
      </Content>
    </Container>
  );
}

```

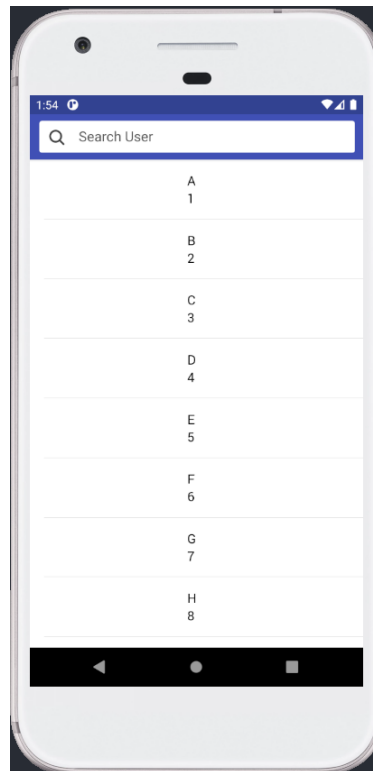
Gambar 2.4 fungsi render.

```
{
  {
    "name": "A",
    "number": "1"
  },
  {
    "name": "B",
    "number": "2"
  },
  {
    "name": "C",
    "number": "3"
  },
  {
    "name": "D",
    "number": "4"
  },
  {
    "name": "E",
    "number": "5"
  },
  {
    "name": "F",
    "number": "6"
  },
  {
    "name": "G",
    "number": "7"
  },
  {
    "name": "H",
    "number": "8"
  },
}
```

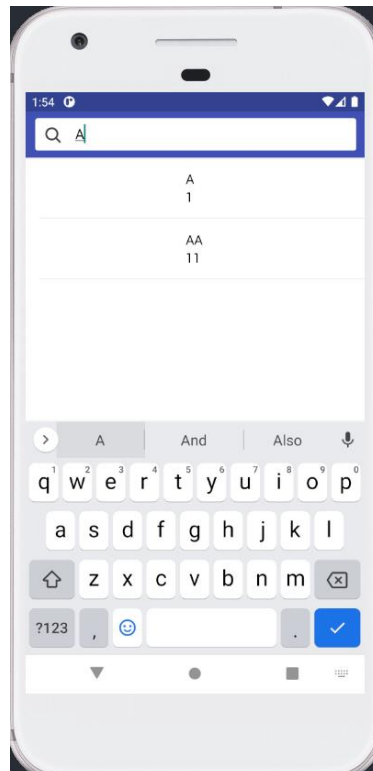
Gambar 2.5 file userList.json.

```
{
  {
    "name": "AA",
    "number": "11"
  },
  {
    "name": "BB",
    "number": "22"
  },
  {
    "name": "CC",
    "number": "33"
  },
  {
    "name": "DD",
    "number": "44"
  },
  {
    "name": "EE",
    "number": "55"
  },
  {
    "name": "FF",
    "number": "66"
  },
  {
    "name": "GG",
    "number": "77"
  },
  {
    "name": "HH",
    "number": "88"
  }
}
```

Gambar 2.6 file userList.json.



Gambar 2.7 Output.



Gambar 2.8 Output.