

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



**Nama : Brian Agustian Kristianto**  
**NIM : 193030503066**

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

# **BAB I**

## **TUJUAN DAN LANDASAN TEORI**

### **1.1. TUJUAN**

**1.1.1.** Mahasiswa mampu membuat program dengan menggabungkan fitur PHP dan JavaScript.

**1.1.2.** Mahasiswa mampu membuat program web yang dinamis.

### **1.2. LANDASAN TEORI**

React Native adalah kerangka kerja JavaScript untuk menulis aplikasi seluler rendering asli dan asli untuk iOS dan Android. Ini didasarkan pada React, pustaka JavaScript Facebook untuk membangun antarmuka pengguna, tetapi alih-alih menargetkan browser, ini menargetkan platform seluler. Dengan kata lain: pengembang web sekarang dapat membuat aplikasi seluler yang terlihat dan terasa benar-benar "asli", semuanya dari kenyamanan pustaka JavaScript yang sudah kita kenal dan sukai. Selain itu, karena sebagian besar kode yang Anda tulis dapat dibagikan antar platform, React Native memudahkan pengembangan secara bersamaan untuk Android dan iOS.

Mirip dengan React for the Web, aplikasi React Native ditulis menggunakan campuran JavaScript dan markup XML-esque, yang dikenal sebagai JSX. Kemudian, di bawah tenda, "jembatan" React Native memanggil API rendering asli di Objective-C (untuk iOS) atau Java (untuk Android). Dengan demikian, aplikasi Anda akan dirender menggunakan komponen UI seluler yang sebenarnya, bukan tampilan web, dan akan terlihat serta terasa seperti aplikasi seluler lainnya. React Native juga mengekspos antarmuka JavaScript untuk API platform, sehingga aplikasi React Native Anda dapat mengakses fitur platform seperti kamera ponsel, atau lokasi pengguna.

React Native saat ini mendukung iOS dan Android, dan juga berpotensi untuk berkembang ke platform masa depan. Dalam buku ini, kami akan membahas iOS dan Android. Sebagian besar kode yang kami tulis akan bersifat lintas platform. Dan ya: Anda benar-benar dapat menggunakan React Native untuk membangun aplikasi seluler yang siap produksi! Beberapa anekdota: Facebook, Palantir, dan TaskRabbit sudah menggunakannya dalam produksi untuk aplikasi yang menghadap pengguna.



**Gambar 1.1** react native

Fakta bahwa React Native benar-benar merender menggunakan API rendering standar platform host-nya memungkinkannya menonjol dari sebagian besar metode pengembangan aplikasi lintas platform yang ada, seperti Cordova atau Ionic. Metode penulisan aplikasi seluler yang ada menggunakan kombinasi JavaScript, HTML, dan CSS biasanya dirender menggunakan tampilan web. Meskipun pendekatan ini bisa berhasil, ia juga memiliki kekurangan, terutama seputar kinerja. Selain itu, mereka biasanya tidak memiliki akses ke kumpulan elemen UI asli platform host. Saat framework ini mencoba meniru elemen UI asli, hasilnya biasanya "terasa" sedikit; merekayasa balik semua detail halus dari hal-hal seperti animasi membutuhkan banyak upaya, dan dapat dengan cepat menjadi usang.

Untuk pengembang yang terbiasa bekerja di Web dengan React, ini berarti Anda dapat membuat aplikasi seluler dengan kinerja dan tampilan serta nuansa aplikasi asli, sambil menggunakan alat yang sudah dikenal.

React Native juga mewakili peningkatan atas pengembangan seluler normal di dua area lainnya: pengalaman pengembang dan potensi pengembangan lintas platform.

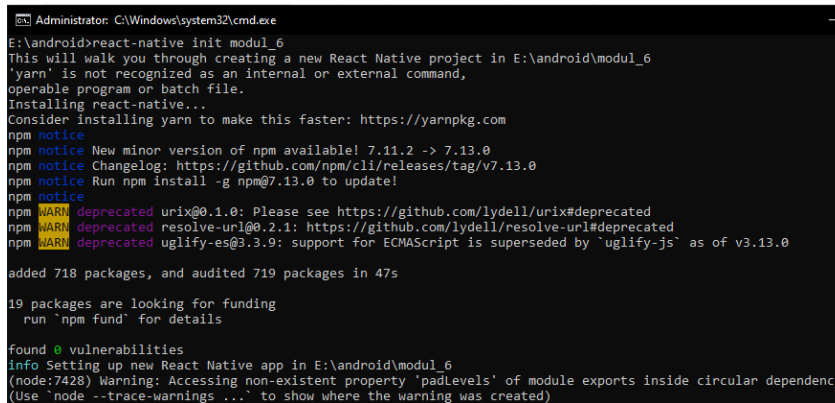
## BAB II

### PEMBAHASAN

Berdasarkan tugas yang diberikan dalam group WhatsApp berikut ini adalah pembahasannya.

#### 1. Membuat Project Baru Untuk Modul 6.

Untuk membuat project dengan cara membuka cmd dan ketik sesuai dengan gambar berikut.

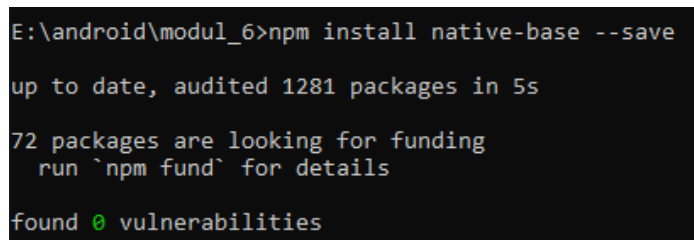


```
Administrator: C:\Windows\system32\cmd.exe
E:\android>react-native init modul_6
This will walk you through creating a new React Native project in E:\android\modul_6
'yarn' is not recognized as an internal or external command,
operable program or batch file.
Installing react-native...
Consider installing yarn to make this faster: https://yarnpkg.com
npm notice New minor version of npm available! 7.11.2 -> 7.13.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v7.13.0
npm notice Run npm install -g npm@7.13.0 to update!
npm notice deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm notice deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm notice deprecated uglify-es@3.3.9: support for ECMAScript is superseded by `uglify-js` as of v3.13.0
added 718 packages, and audited 719 packages in 47s
19 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
info Setting up new React Native app in E:\android\modul_6
(node:7428) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
```

**Gambar 2.1** project baru dengan nama “modul\_6”

#### 2. Menginstall native base dan menghubungkan pada project modul\_6

Untuk menginstall native base pada project modul\_6, pertama cmd harus berada dalam folder modul\_6. Dan lakukan seperti pada gambar dibawah ini.



```
E:\android\modul_6>npm install native-base --save
up to date, audited 1281 packages in 5s
72 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
```

**Gambar 2.2** native base

Setelah menginstall native base, selanjutnya menghubungkan react native dan native base. Untuk menghubungkannya dapat dilakukan dengan cara pada gambar dibawah ini.

```

E:\android\modul_6>react-native link
info Linking assets to ios project
warn Group 'Resources' does not exist in your Xcode project. We have created it automatically for you.
info Linking assets to android project
success Assets have been successfully linked to your project
(node:3000) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(Use 'node --trace-warnings ...' to show where the warning was created)

```

**Gambar 2.3** berhasil terhubung

### 3. Membuat Tampilan Header pada Android

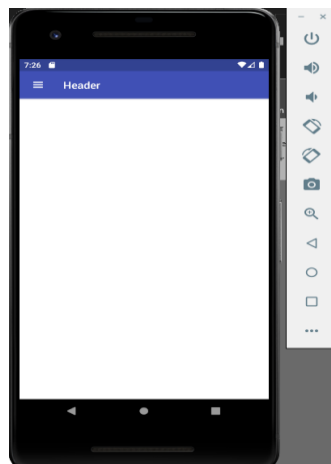
Dalam membuat tampilan header ini menggunakan perintah di bawah ini dan hasilnya akan seperti pada gambar di bawah **gambar 2.4**.

```

JS App.js > HeaderExample
1  import React, { Component } from 'react';
2  import { Container, Header, Title, Button, Left, Right, Body, Icon } from
3  'native-base';
4  export default class HeaderExample extends Component {
5    render() {
6      return (
7        <Container>
8          <Header>
9            <Left>
10             <Button transparent>
11               <Icon name='menu' />
12             </Button>
13            </Left>
14            <Body>
15              <Title>Header</Title>
16            </Body>
17            <Right />
18          </Header>
19        </Container>
20      );
21    }
22  }

```

**Gambar 2.4** app.js



**Gambar 2.5** tampilan android

### 4. Membuat sebuah Search Filter List dalam App.js

Untuk Search Filter ini, menggunakan perintah di bawah ini.

**Tabel 2.1** app.js

```
import React, {Component} from 'react';
import { Container, Header, Content, Left, Body, Icon, Text, ListItem,
Thumbnail, Input, Item, } from 'native-base';
let helperArray=require('./userList.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 avatar>
              <Left>
                <Thumbnail source={ { uri: item.image } } />
              </Left>
              <Body>
                <Text>{ item.name }</Text>
                <Text note>{ item.address }</Text>
              </Body>
            </ListItem>
          ))}
        </Content>
      </Container>
    );
  }
}
```

Dalam perintah di atas terdapat perintah import Container, Header, Content, Left, Body, Icon, Text, ListItem, Thumbnail, Input, dan Item. Ini berguna untuk aplikasi search filter nantinya.

Variable HelperArray berfungsi untuk memanggil userList.json. Pemanggilan userList.json ini untuk mengambil data di dalamnya yang berupa nama, alamat, dan foto profil.

Di bawah variable helperarray tadi terdapat constructor props. Props ini adalah kelas suatu component. Di bawahnya terdapat super(props) dan this.state.

Selanjutnya ada searchUser(texttoSearch), ini berfungsi apabila kita memasukkan teks yang ingin dicari maka app.js tersebut akan memfilter nama sesuai dengan teks yang diinputkan tadi.

Kemudian terdapat render, dalam header ini ada searchBar rounded. Fungsi keseluruhan dari searchBar rounded adalah pada saat memasukan teks ke dalam place holder, onChangeText akan memanggil setText yang selanjutnya akan memeriksakannya kondisi itu.

Dalam bagian tag content ini memiliki usersFiltered, usersFiltered ini memiliki fungsi untuk memfilter pada tag header tadi. usersFiltered juga akan menampilkan thumbnail user yang telah difilter tadi. Di dalam bagian tag body, akan menampilkan name dan address (yang berupa note).

## 5. Menambahkan userList.json

Dikarenakan pada salah satu code terdapat perintah memerlukan userList.json, berikut adalah codenya.

**Tabel 2.2** userList.json

```
[
  {
    "name" : "Brian A",
    "address" : "Palangka Raya, Indonesia",
```



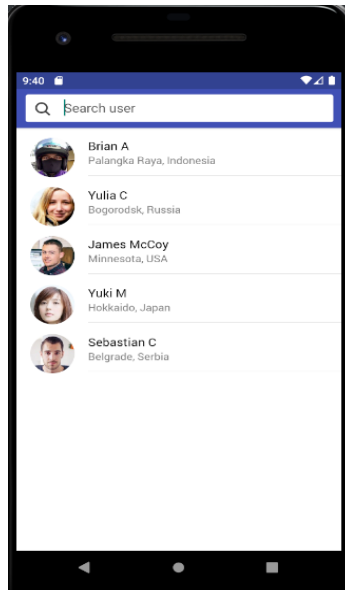
```

"image" : "https://instagram.fbdf4-1.fna.fbcdn.net/v/t51.2885-19/s150x150/96584567_2616337145348134_7041195683147153408_n.jpg?tp=1&_nc_ht=instagram.fbdf4-1.fna.fbcdn.net&_nc_cat=110&_nc_ohc=u0zPBdRKZv0AX9-rasY&edm=ABfd0MgBAAAA&ccb=7-4&oh=b1d12818e41e0be8a99268c3ef99a82c&oe=60CA7D2A&_nc_sid=7bff83"
  },
  {
    "name" : "Yulia C",
    "address" : "Bogorodsk, Russia",
    "image" : "https://randomuser.me/api/portraits/women/31.jpg"
  },
  {
    "name" : "James McCoy",
    "address" : "Minnesota, USA",
    "image" : "https://randomuser.me/api/portraits/men/94.jpg"
  },
  {
    "name" : "Yuki M",
    "address" : "Hokkaido, Japan",
    "image" : "https://randomuser.me/api/portraits/women/60.jpg"
  },
  {
    "name" : "Sebastian C",
    "address" : "Belgrade, Serbia",
    "image" : "https://randomuser.me/api/portraits/men/46.jpg"
  }
]

```

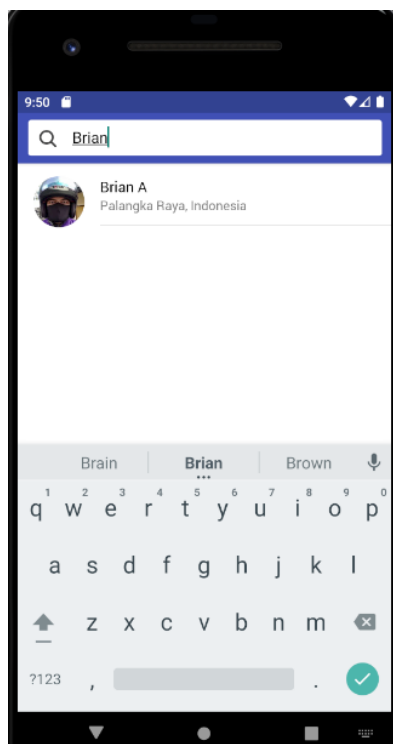
Perintah di atas merupakan data-data dari user yang akan ditampilkan dalam app.js. Dalam userList ini terdiri nama, address, dan image yang berupa link.

Apabila semua yang di atas telah dilakukan, maka hasil dari modul\_6 ini dapat dilihat pada gambar di bawah ini.

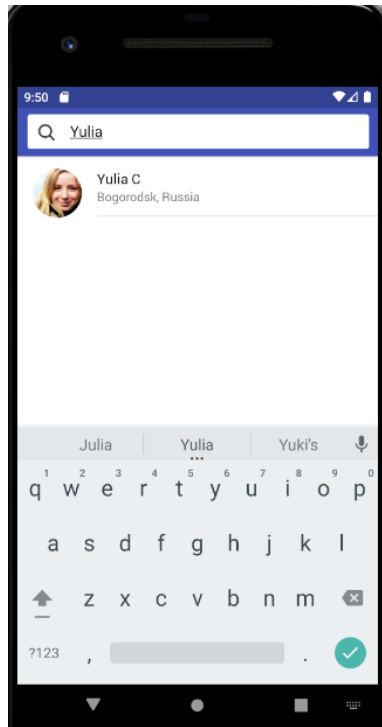


**Gambar 2.6** tampilan pada android

Untuk menjalani tes apakah program tersebut dapat menyaring/memfilter dengan cara memasukan teks pada place holder Search user, sebagai contoh Brian dan Yulia.



**Gambar 2.7** hasil pencarian “Brian”



**Gambar 2.8** hasil pencarian “Yulia”

### **BAB III**

### **KESIMPULAN**

React Native adalah kerangka kerja yang membangun hierarki komponen UI untuk membangun kode JavaScript. Ini memiliki seperangkat komponen untuk platform iOS dan Android untuk membangun aplikasi seluler dengan tampilan dan nuansa asli.

React Native sangat membantu ketika membuat aplikasi seluler yang nyata dan menarik hanya dengan bantuan JavaScript, yang dapat didukung untuk platform Android dan iOS. Cukup kode sekali, dan aplikasi REACT Native tersedia untuk platform iOS dan Android yang membantu menghemat waktu pengembangan. Menemukan popularitas besar dan juga didukung oleh Facebook, REACT Native, memiliki dukungan komunitas yang sangat besar hari ini.

## DAFTAR PUSTAKA

Thinkwik. (2018). *React Native: What is it? and, Why is it used?* Jan 31.

<https://medium.com/@thinkwik/react-native-what-is-it-and-why-is-it-used-b132c3581df>

Native, R. (n.d.). *Setting Up The Development Environment*.

<https://reactnative.dev/docs/environment-setup>

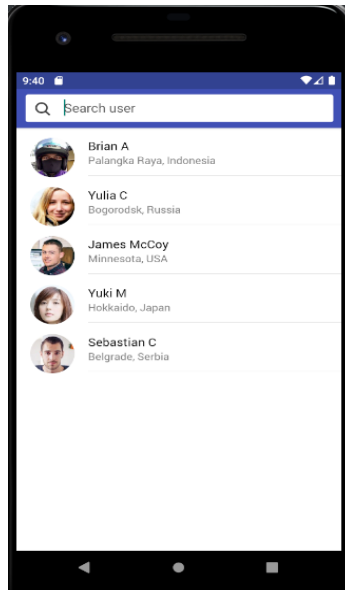
## LAMPIRAN

```
import React, {Component} from 'react';
import { Container, Header, Content, Left, Body, Icon, Text, ListItem,
Thumbnail, Input, Item, } from 'native-base';
let helperArray=require('./userList.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 avatar>
              <Left>
                <Thumbnail source={ { uri: item.image } } />
              </Left>
              <Body>
                <Text>{item.name}</Text>
                <Text note>{item.address}</Text>
              </Body>
            </ListItem>
          ))}
        </Content>
      </Container>
    );
  }
}
```

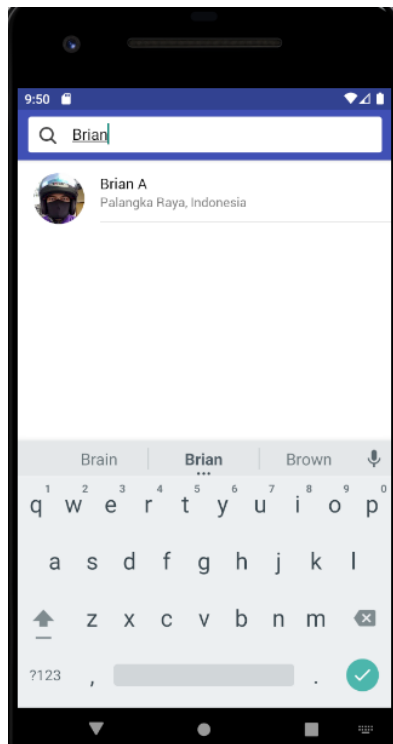
**Tabel 2.2** userList.json

```
[
  {
    "name" : "Brian A",
    "address" : "Palangka Raya, Indonesia",
```

```
    "image" : "https://instagram.fbdj4-1.fna.fbcdn.net/v/t51.2885-19/s150x150/96584567_2616337145348134_7041195683147153408_n.jpg?tp=1&_nc_ht=instagram.fbdj4-1.fna.fbcdn.net&_nc_cat=110&_nc_ohc=u0zPBdRKZv0AX9-rasY&edm=ABfd0MgBAAAA&ccb=7-4&oh=b1d12818e41e0be8a99268c3ef99a82c&oe=60CA7D2A&_nc_sid=7bff83"
  },
  {
    "name" : "Yulia C",
    "address" : "Bogorodsk, Russia",
    "image" : "https://randomuser.me/api/portraits/women/31.jpg"
  },
  {
    "name" : "James McCoy",
    "address" : "Minnesota, USA",
    "image" : "https://randomuser.me/api/portraits/men/94.jpg"
  },
  {
    "name" : "Yuki M",
    "address" : "Hokkaido, Japan",
    "image" : "https://randomuser.me/api/portraits/women/60.jpg"
  },
  {
    "name" : "Sebastian C",
    "address" : "Belgrade, Serbia",
    "image" : "https://randomuser.me/api/portraits/men/46.jpg"
  }
]
```

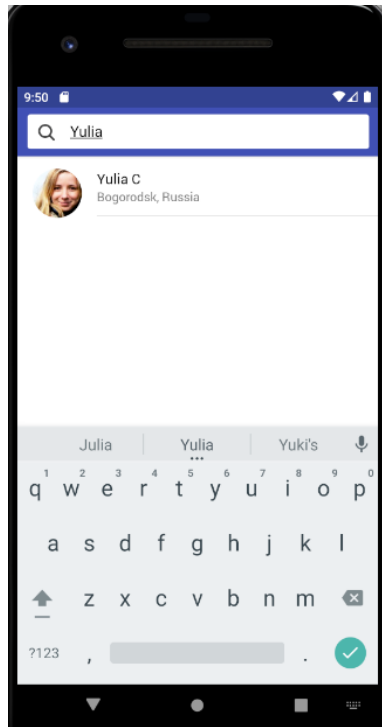


**Gambar 2.6** tampilan pada android



**Gambar 2.7** hasil pencarian “Brian”





**Gambar 2.8** hasil pencarian “Yulia”