

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



**NAMA : KEVIN IMMANUEL RUI COSTA**

**NIM : 193020503030**

**KELAS : A**

**MODUL : VI (Flatlist Search React Native)**

**JURUSAN/PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PALANGKARAYA**

**2021**

## BAB I

### LANDASAN TEORI

#### 1.1 Landasan Teori

##### 1.1.1 React Native

React Native adalah *framework mobile app development* yang memungkinkan pengembangan aplikasi secara multi-*platform* yaitu Android dan iOS. Sederhananya, React Native adalah *framework* yang memungkinkan *developer* membuat aplikasi untuk berbagai *platform* dengan menggunakan basis kode yang sama.

Penggabungan antara *native mobile app* dan React juga bisa dilakukan dengan mudah. Jadi, pengembang bisa membuat aplikasi Android dan iOS dengan lebih cepat.

React Native sendiri pertama kali dirilis pada tahun 2015 oleh Facebook dan menjadi bagian dari proyek *open source* mereka.

Kemudian, hanya dalam beberapa tahun saja, React Native telah menjadi andalan bagi banyak *developer* untuk mengembangkan aplikasi seluler.

Beberapa contoh aplikasi seluler yang menggunakan *framework* React Native adalah Instagram, Facebook, Pinterest, hingga Skype.

Ada beberapa alasan lain mengapa *React Native* begitu populer, antara lain:

- *Developer* hanya perlu membuat satu kode untuk mengembangkan aplikasi berbasis Android atau iOS. Jadi, *framework* ini benar-benar menghemat waktu.

- React Native adalah *framework* yang dibangun berdasarkan React, yaitu *library* dari JavaScript yang sudah populer.
- *Framework* ini mendorong *frontend developer* yang sebelumnya hanya dapat bekerja dengan teknologi berbasis web saja. Kini, mereka bisa dengan mudah untuk mengembangkan *mobile platform*.

Selain beberapa alasan di atas, *React Native* juga memiliki keunggulan lain yaitu tidak akan *me-render webviews* dalam kodenya. Jadi program akan dijalankan sesuai dengan tampilan dan komponen dari *native* yang awalnya digunakan.

### 1.1.2 Flatlist

React Native Flatlist merupakan komponen-komponen sederhana untuk menampilkan list daftar data namun banyak digunakan, contoh flatlist yang akan membantu anda untuk memahami bagaimana menggunakannya.

FlatList menampilkan data terstruktur yang serupa dalam daftar yang dapat di-scroll. Ini berfungsi dengan baik untuk daftar data yang besar di mana jumlah item daftar dapat berubah seiring waktu. FlatList hanya menampilkan elemen render yang saat ini ditampilkan di layar, tidak semua elemen daftar sekaligus.

Komponen FlatList membutuhkan dua props yang diperlukan: data dan renderItem. Data adalah sumber elemen untuk daftar, dan renderItem mengambil satu item dari sumber dan mengembalikan komponen yang diformat untuk dirender.

## **BAB II**

### **PEMBAHASAN**

Pada praktikum ini, diminta untuk membuat sebuah program aplikasi mobile, dimana dalam program tersebut terdapat program search menggunakan flatlist react-native. Pertama, buka cmd terlebih dahulu lalu buat projectnya dengan menggunakan perintah pada cmd.

`react-native init ModulVI`

Tunggu perintah tersebut berjalan hingga selesai. Setelah selesai, maka akan ada folder project baru dengan nama ModulVI. Lalu pada project ModulVI tersebut akan diinstall native-base, caranya dengan menuliskan perintah pada cmd, perintahnya adalah,

`npm install native-base`

Tunggu hingga selesai. Setelah selesai, buka folder ModulVI tersebut lalu buka file app.js di visual studio code. Hapus semua code yang terdapat dalam file tersebut. Lalu diganti dengan kode yang dibagi menjadi beberapa bagian sebagai berikut,

#### **Bagian 1**

```
import React, {Component} from 'react';
import{
  Container,
  Header,
  Content,
  Left,
  Right,
  Body,
  Icon,
  Text,
  ListItem,
  Item,
```

```
Input,  
} from 'native-base';
```

Merupakan bagian import, yang digunakan untuk dapat menggunakan komponen-komponen yang terdapat pada library react dan juga library native-base.

## Bagian 2

```
let helperArray = require('./nama.json');  
export default class App extends Component {  
  constructor(props){  
    super(props);  
    this.state = {  
      allUsers: helperArray,  
      usersFiltered:helperArray,  
    };  
  }  
}
```

Bagian kedua, terdapat deklarasi variabel dengan nama helperArray dimana variabel tersebut berisi require, require digunakan untuk dapat menghubungkan file app.js tersebut dengan file nama.json. nama.json merupakan file yang berisi daftar nama-nama yang akan digunakan pada program ini. Berikut adalah isi dari file nama.json,

## Nama.json

```
[  
  {  
    "name":"Kevin",  
    "address":"Jalan K"  
  },  
  {  
    "name":"Immanuel",  
    "address":"Jalan I"  
  },  
  {  
    "name":"Rui",  
    "address":"Jalan R"  
  },  
]
```

```
{
  "name": "Costa",
  "address": "Jalan C"
},

{
  "name": "Kirc",
  "address": "Jalan A"
},

{
  "name": "Irc",
  "address": "Jalan B"
},

{
  "name": "Kev",
  "address": "Jalan C"
},

{
  "name": "Rci",
  "address": "Jalan D"
},

{
  "name": "Kevirc",
  "address": "Jalan E"
},

{
  "name": "rcikev",
  "address": "Jalan F"
},

{
  "name": "test",
  "address": "Jalan G"
},

{
  "name": "set",
  "address": "Jalan H"
},
```

```
{
  "name": "sot",
  "address": "Jalan I"
}
]
```

Setelah itu, terdapat kelas dengan nama App, yang berisi, konstruktor, dimana konstruktor tersebut berisi inisiasi atribut dengan nilai sama dengan variabel helperArray. Atribut tersebut akan mengambil data-data dari variabel helperArray.

#### Bagian 4

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

Bagian 4 merupakan fungsi yang terdapat dalam kelas App, merupakan fungsi searchUser, yang mana fungsi tersebut akan digunakan untuk mencari nama berdasarkan kata yang diinputkan pada kolom pencarian. Fungsi searchUser ini menggunakan filter, dimana akan menyeleksi sesuai dari kata yang diinputkan. Lalu terdapat toLowerCase() dimana digunakan untuk merubah nilai string ke nilai string yang terdiri dari huruf kecil semua.

#### Bagian 5

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

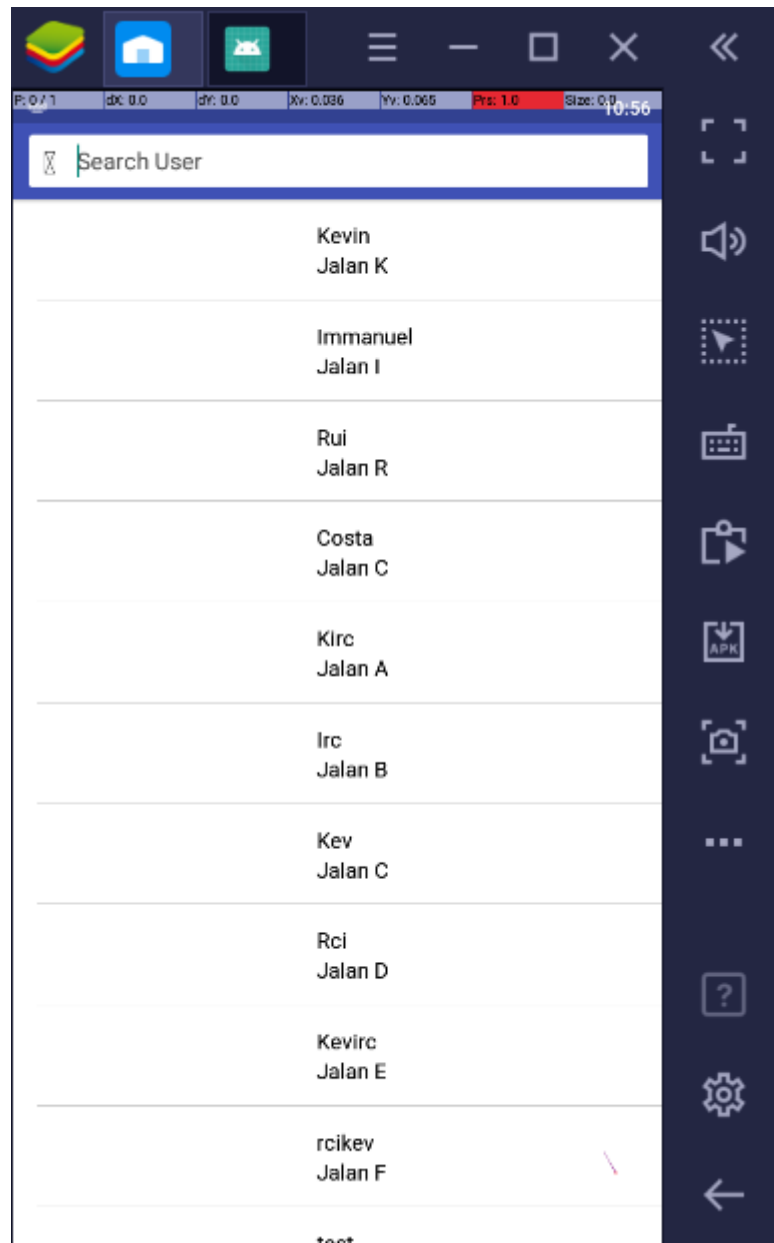
```

        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>
);
}
}

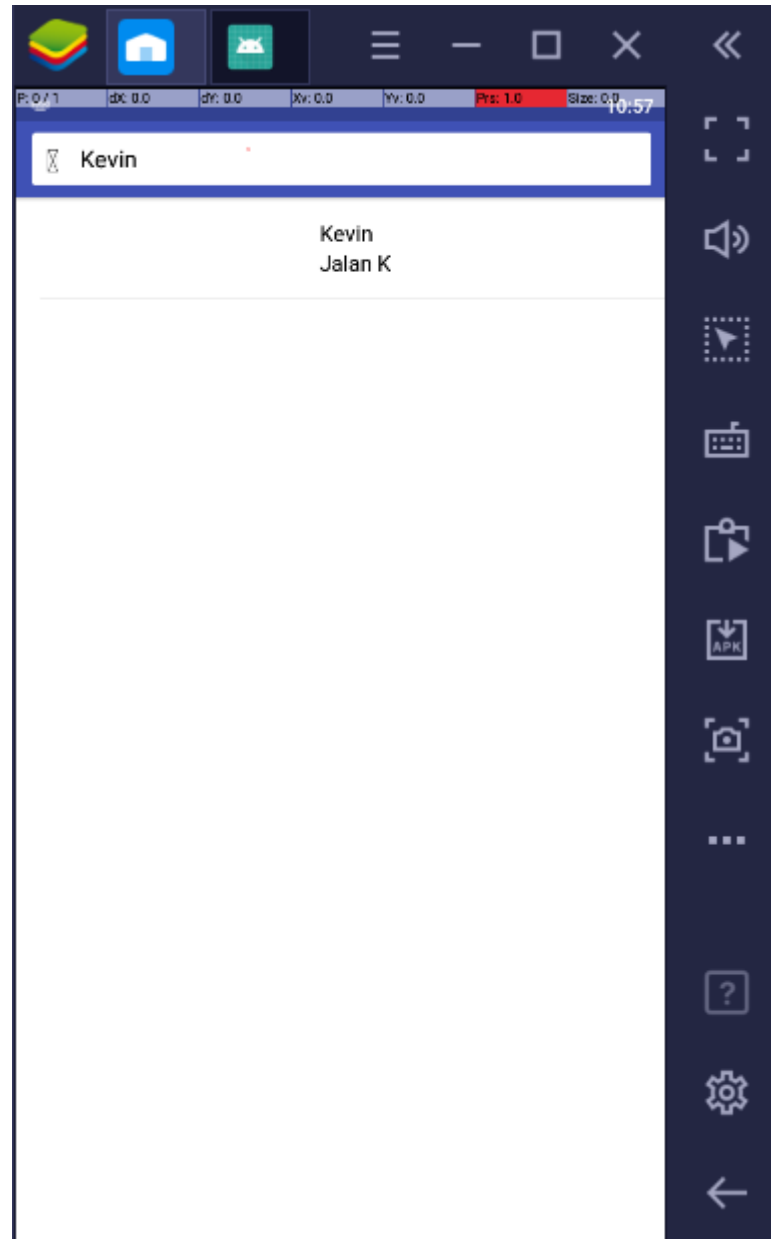
```

Bagian kelima, merupakan bagian fungsi render, dimana fungsi ini berisi perintah-perintah untuk menampilkan data-data dalam bentuk list. Dimana data-data yang ditampilkan adalah data yang terdapat pada variabel helperArray. Berikut adalah tampilan dari data tersebut serta implementasi dari pencarian kata pada aplikasi tersebut.

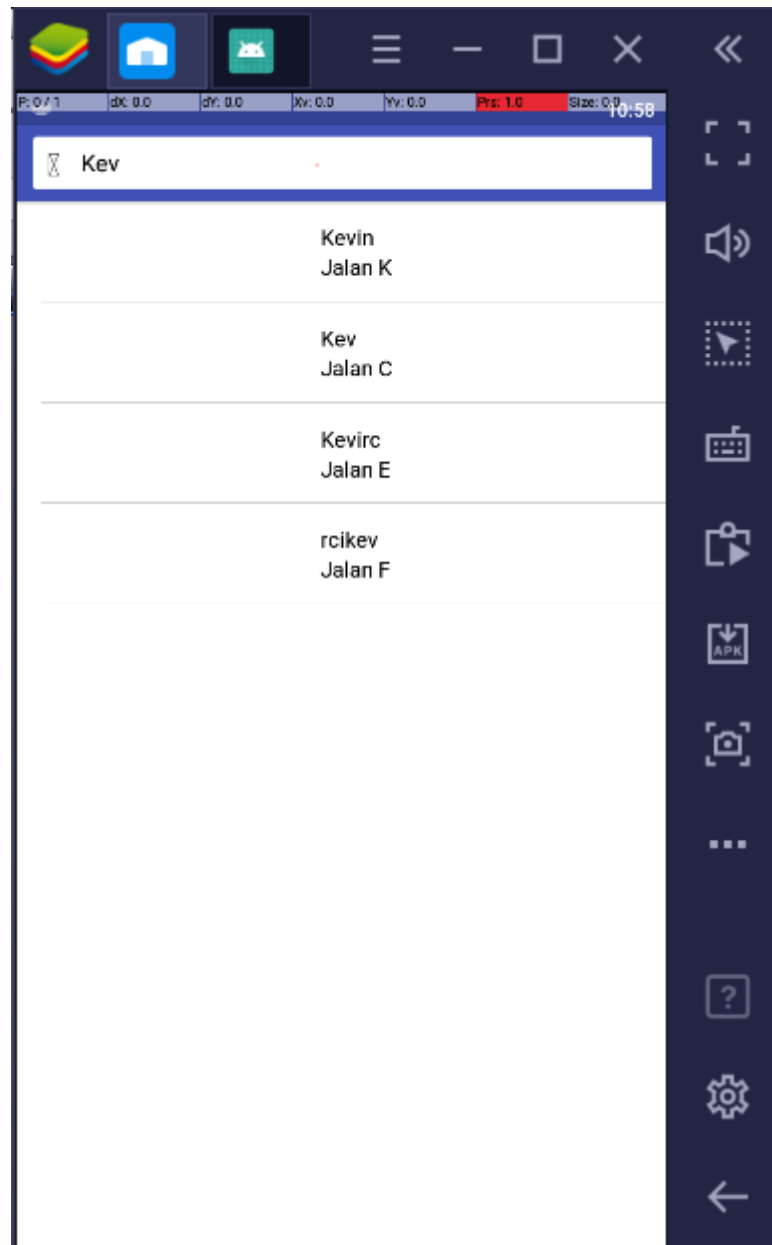




**Gambar 2.1 Tampilan Aplikasi ModulVI**



**Gambar 2.2 Tampilan Aplikasi ModulVI**



**Gambar 2.3 Tampilan Aplikasi ModulVI**

### **BAB III**

### **KESIMPULAN**

Kesimpulannya, search flatlist pada react native, sangat membantu dalam membuat sebuah program yang dimana program tersebut memerlukan sebuah fitur pencarian di dalamnya sehingga pengguna menjadi lebih nyaman dalam menggunakan program yang di dalamnya terdapat fitur search.

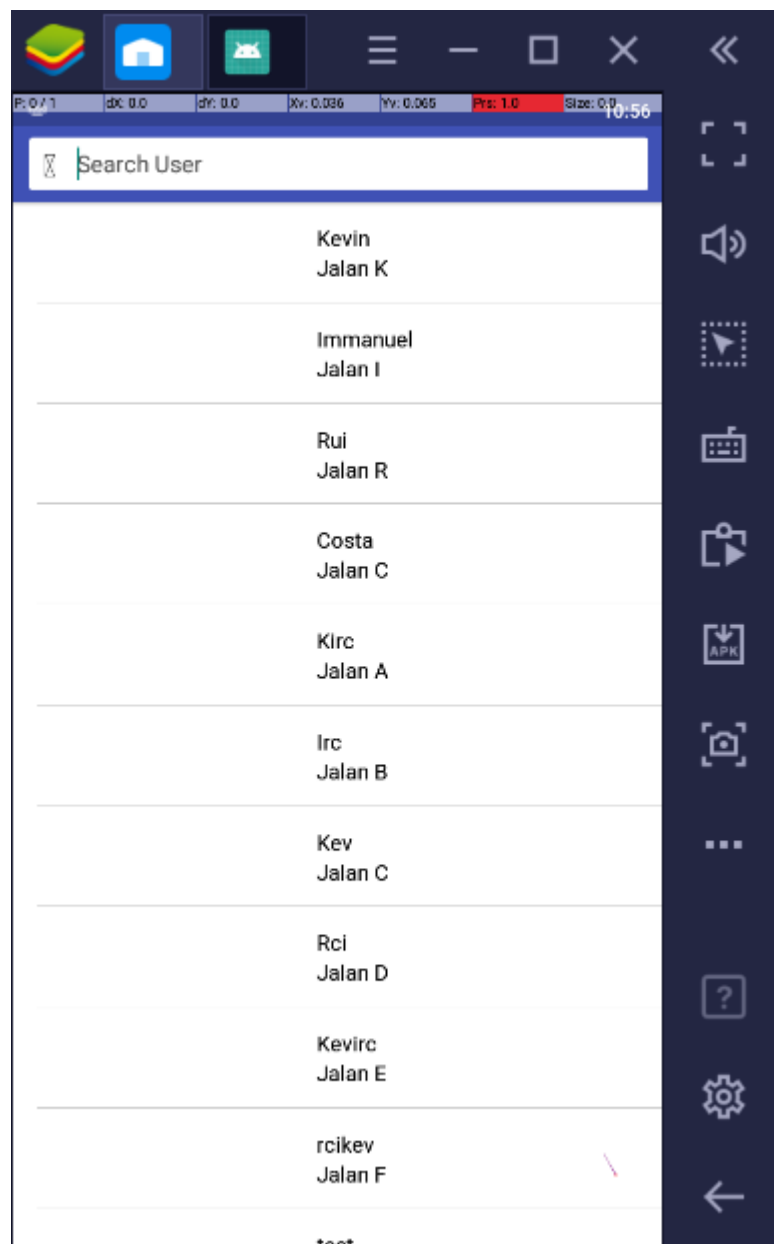
## DAFTAR PUSTAKA

Ismi, T. (2021). *React Native: Ketahui Pengertian serta Kelebihan dan Kekurangannya*.  
<https://glints.com/id/lowongan/react-native-adalah/#.YHOqeT8RXDc>

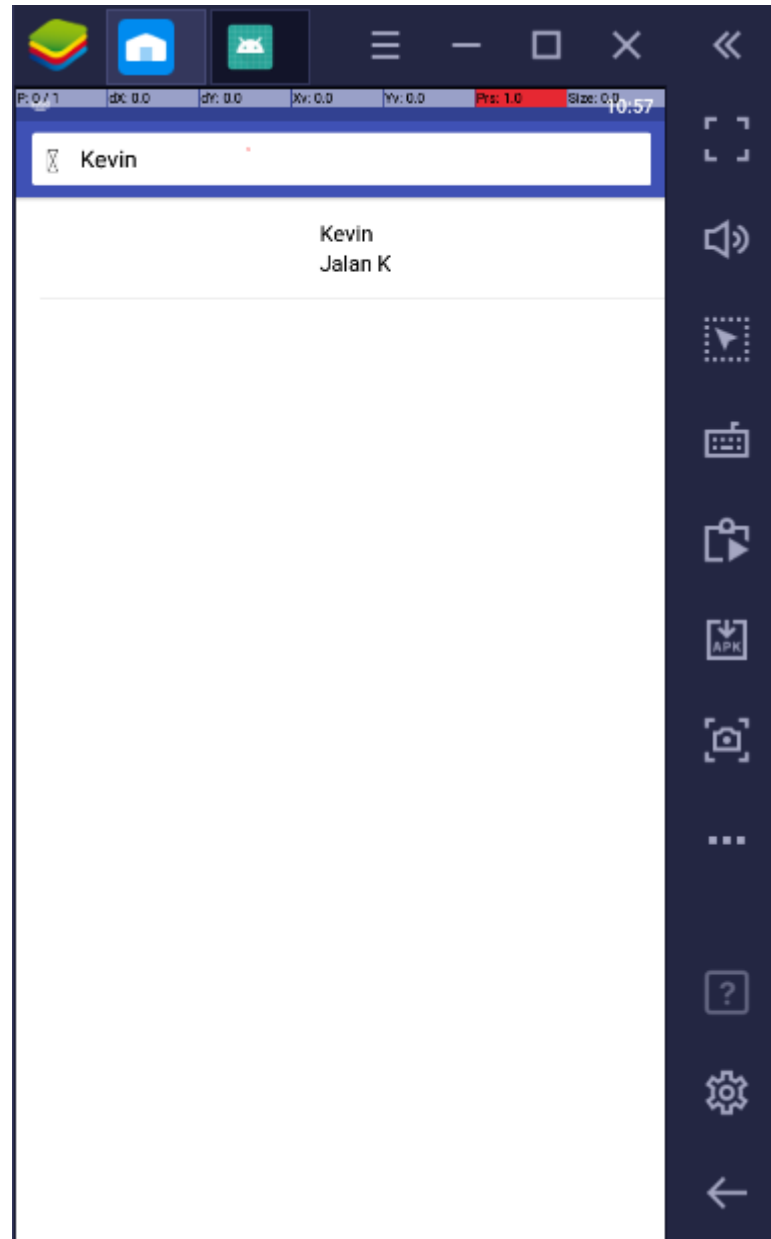
Javatpoint. (n.d.). *React Native FlatList*.  
<https://www.javatpoint.com/react-native-flatlist>

React-native. (2020). *Basic React Native Part 25 - React Native Flatlist*.  
<https://cloudborneo.com/basic-react-native-part-25-react-native-flatlist/>

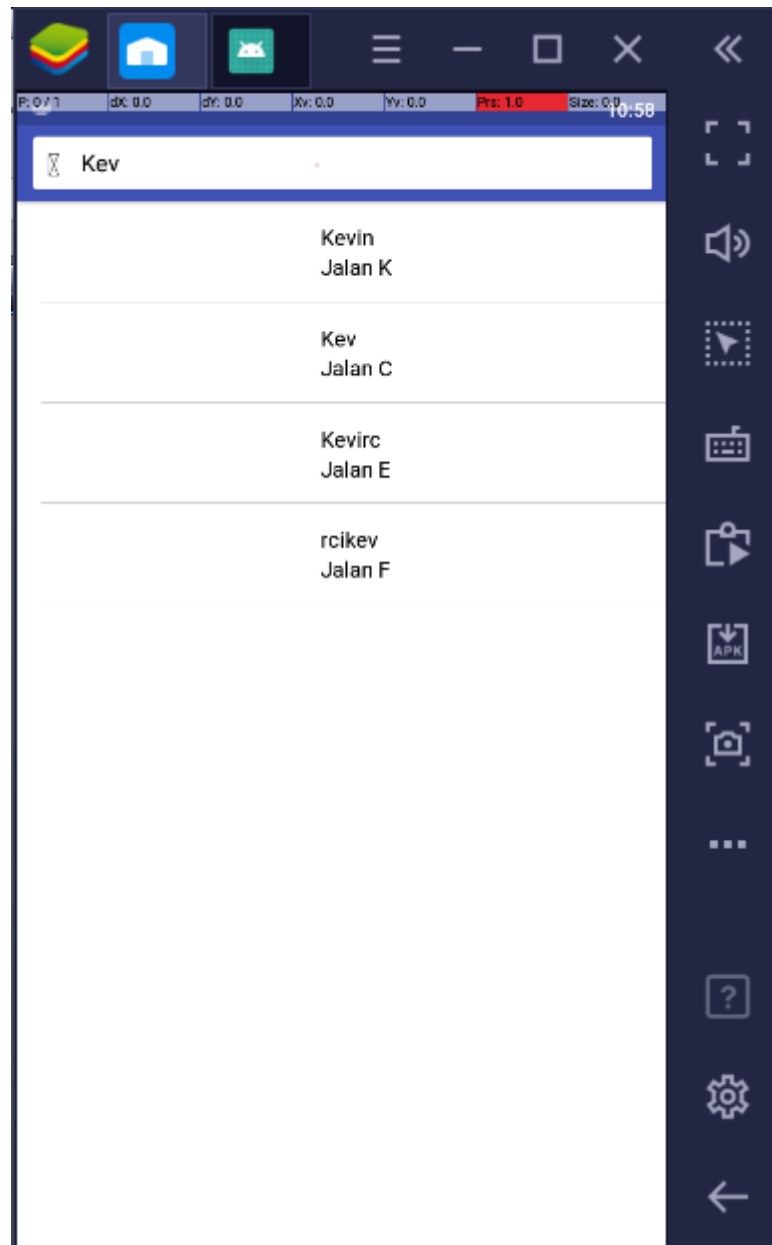
## LAMPIRAN



**Gambar 2.1 Tampilan Aplikasi ModulVI**



**Gambar 2.2 Tampilan Aplikasi ModulVI**



**Gambar 2.3 Tampilan Aplikasi ModulVI**