

## BAB II

### LANDASAN TEORI

#### 2.1 React Native

React Native merupakan sebuah Javascript *framework* untuk membangun sebuah aplikasi berbasis *mobile* untuk Android dan iOS dengan menggunakan React. React sendiri adalah sebuah *framework* yang dirilis oleh Facebook untuk membangun sebuah tampilan yang kaya dan interaktif. React Native akan melakukan kompilasi terhadap komponen React menjadi komponen Native iOS dan Android sehingga React Native akan melakukan *render* aplikasi dengan menggunakan komponen UI *mobile*, bukan *web view*. Selain itu, kode yang ditulis dengan menggunakan React Native dapat dijalankan kedua *platform* iOS dan Android. Biasanya, aplikasi berbasis iOS dan Android merupakan dua aplikasi yang terpisah. Aplikasi iOS dibuat menggunakan bahasa pemrograman Swift dan aplikasi Android dibuat menggunakan Kotlin atau Java. Dengan React Native, pembuatan aplikasi hanya perlu dibuat dengan satu Bahasa pemrograman saja dan dapat dibuat menjadi aplikasi iOS dan Android (Lim, 2020).

Sebuah halaman aplikasi React Native terdiri dari gabungan dari beberapa komponen. Ada dua cara pembuatan komponen React Native yaitu dengan menggunakan *class* dan *function* (Paul & Nalwaya, 2019).

### 2.1.1 Komponen *Class*

Komponen dapat dibuat dengan menggunakan *class* dari Javascript. *Class* kemudian di-*extends* dengan *Component* yang di-*import* dari *library* React. Terdapat beberapa *method* yang dapat digunakan salah satunya adalah *render*. *Render* akan mengembalikan potongan kode yang akan ditampilkan ke layar. Contoh penggunaan *class* dapat dilihat pada Gambar 2.1.

```
// Import Libraries yang dibutuhkan
import React, { Component } from 'react';
import { StyleSheet, Text, View } from 'react-native';

// Type untuk Props dan State
type Props = {};
type State = {};

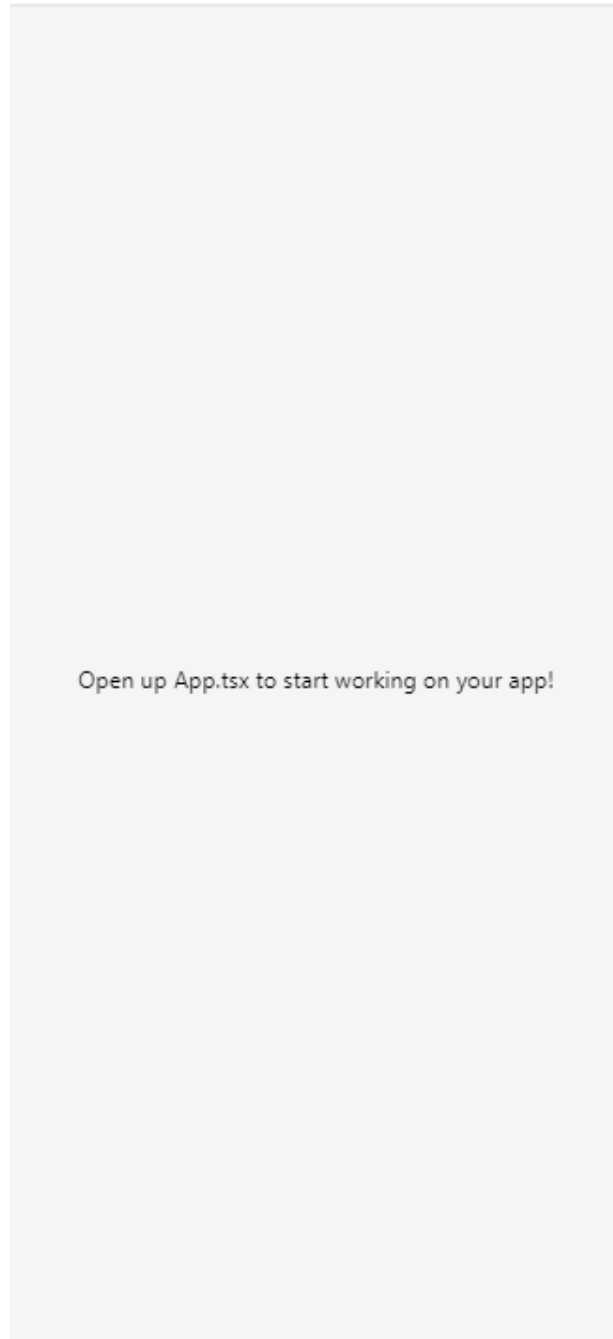
// Komponen Halaman
export default class App extends Component<Props, State> {
  render() {
    return (
      <View style={styles.container}>
        <Text>Open up App.tsx to start working on your app!</Text>
      </View>
    );
  }
}

// Object untuk style
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: 'whitesmoke',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```

Gambar 2.1 Potongan Kode Komponen *Class*

*Class* tersebut memiliki *method* yang bernama *render* yang dimana akan mengembalikan elemen yang akan dimunculkan di layar (Boduch & Derks,

2020). Hasil dari potongan kode pada Gambar 2.1 akan menghasilkan tampilan seperti pada Gambar 2.2.



Gambar 2.2 Hasil dari Potongan Kode Komponen *Class*

Selain *method render*, ada beberapa *method* lain seperti `componentDidMount` dan `componentWillUnmount` yang merupakan *method* dari

siklus hidup komponen *class* React Native. `componentDidMount` akan dipanggil ketika komponen akan ditampilkan. `componentDidUpdate` akan dipanggil ketika `componentWillUnmount` akan dipanggil ketika komponen akan dihilangkan (Boduch & Derks, 2020).

### **2.1.2 Komponen *Function***

Seperti pada Gambar 2.3, dapat dilihat potongan kode dari komponen *function*. Perbedaan yang jelas terlihat adalah komponen tersebut dibuat menggunakan *function*. Perbedaan lain adalah *function* tersebut tidak memiliki *method render* karena komponen *function* tersebut langsung membalikan elemen yang akan dimunculkan di layar. Selain itu, komponen *function* tidak memiliki *method* siklus hidup seperti komponen *class* sehingga memerlukan React Hooks apabila ingin menjalankan sesuatu dari salah satu siklus hidupnya (sebagai contoh *method* `useEffect` digunakan untuk menggantikan *method* `componentDidMount`).

```
// Import Libraries yang dibutuhkan
import React, { Component } from 'react';
import { StyleSheet, Text, View } from 'react-native';

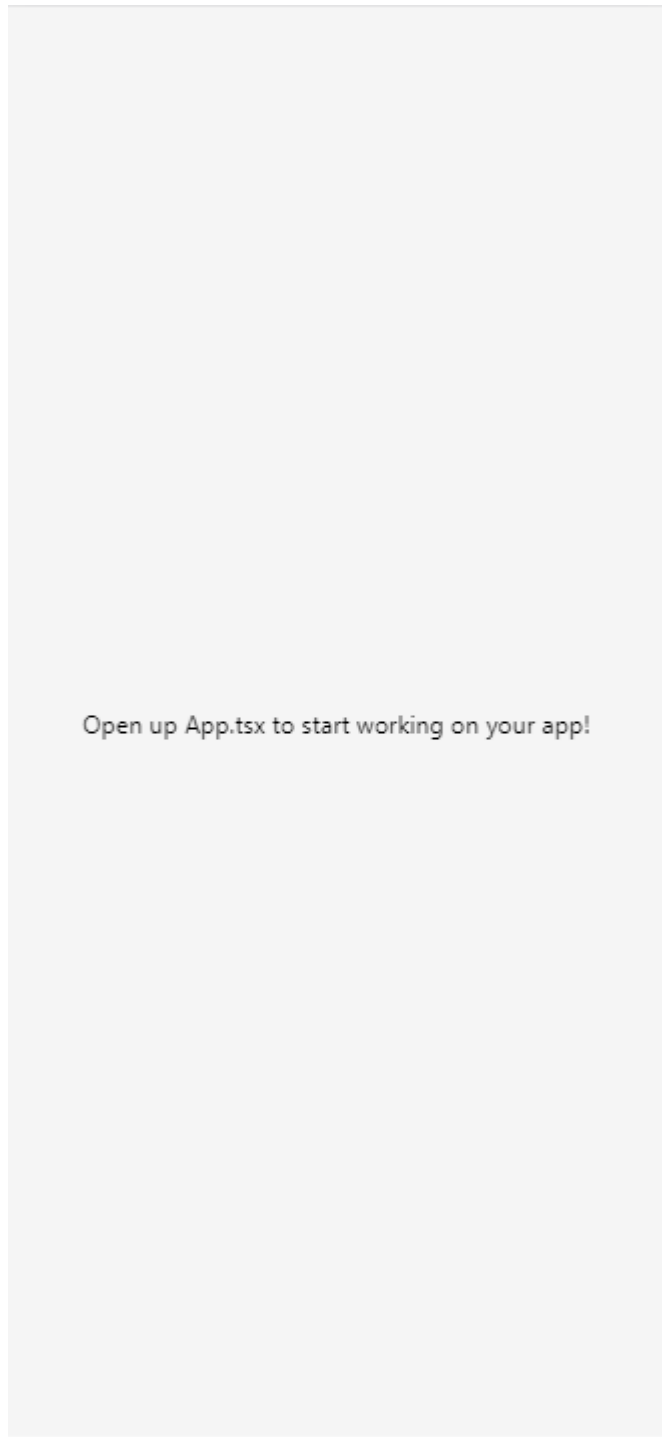
// Type untuk Props dan State
type Props = {};
type State = {};

// Komponen Halaman
export default function App(props: Props) {
  return (
    <View style={styles.container}>
      <Text>Open up App.tsx to start working on your app!</Text>
    </View>
  );
}

// Object untuk style
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: 'whitesmoke',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```

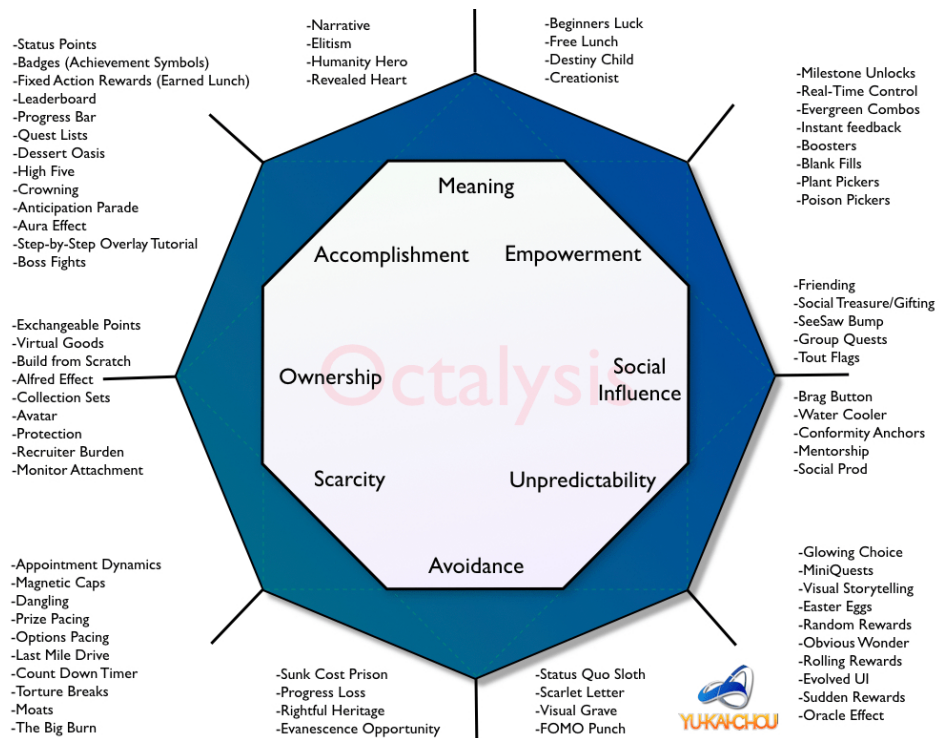
Gambar 2.3 Potongan Kode Komponen *Function*

Hasil dari potongan kode komponen *function* pada Gambar 2.3 akan menghasilkan hasil yang sama seperti pada kode komponen *class* sebelumnya yang dapat dilihat pada Gambar 2.4.



Gambar 2.4 Hasil dari Potongan Kode Komponen *Function*

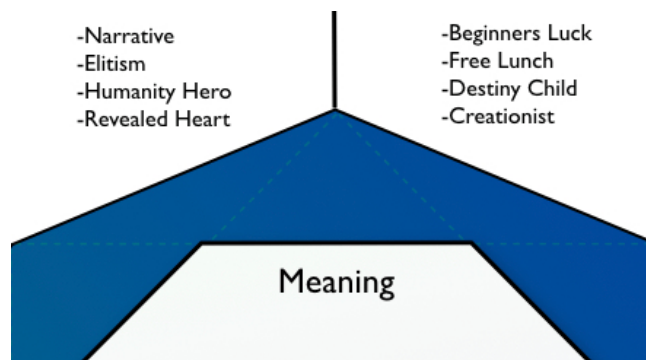
## 2.2 Metode Gamifikasi Octalysis



Gambar 2.5 Delapan Inti dari Metode Gamifikasi Octalysis (Chou, 2015)

Metode gamifikasi yang diciptakan oleh Yu-kai Chou ini memiliki delapan inti yang difokuskan ke perilaku manusia, yaitu *meaning*, *accomplishment*, *empowerment*, *ownership*, *social influence*, *scarcity*, *unpredictability*, *avoidance* (Chou, 2015).

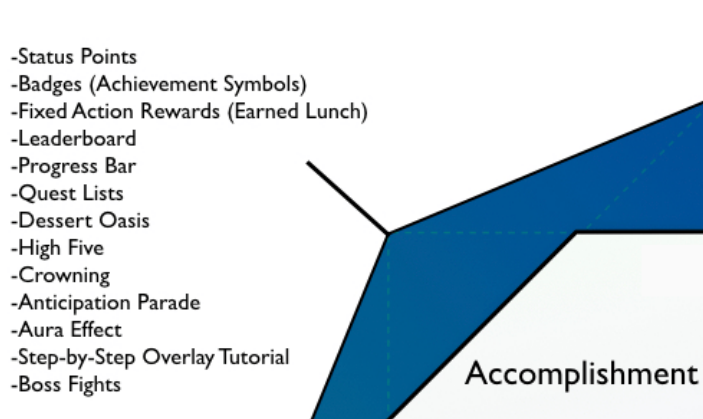
### a. *Meaning*



Gambar 2.6 Poin-poin pada *Meaning*

*Meaning* berarti pemain percaya bahwa dia sedang melakukan sesuatu yang lebih besar dari dirinya atau merasa terpilih untuk melakukan sesuatu. Selain itu, efek *beginner's luck* juga dapat membuat *meaning* bagi pemainnya. Sebagai contoh, pemain dapat *item* permainan yang langka diawal permainan.

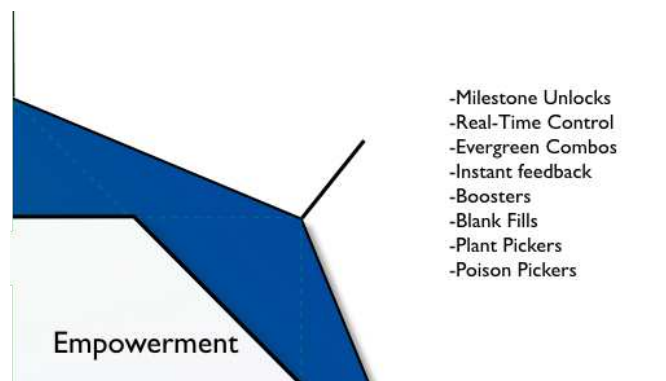
b. *Accomplishment*



Gambar 2.7 Poin-poin pada *Accomplishment*

*Accomplishment* adalah sebuah dorongan untuk membuat sebuah kemajuan, mengembangkan kemampuan bermain, dan mengatasi tantangan-tantangan. Kata tantangan disini menjadi sangat penting karena sebuah *badge* atau *trophy* tanpa tantangan tidak akan berarti.

c. *Empowerment*

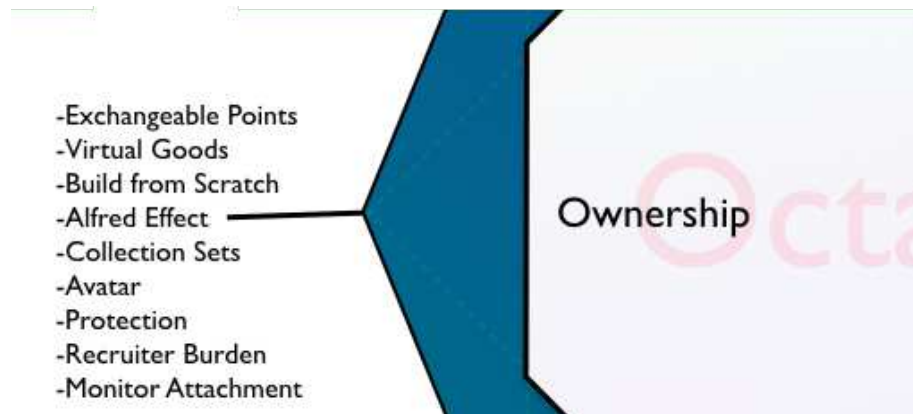


Gambar 2.8 Poin-poin pada *Empowerment*



*Empowerment* adalah saat dimana pengguna berpikir kreatif untuk menyelesaikan tantangan dengan cara memikirkan beberapa kombinasi yang dapat dilakukan.

d. *Ownership*



Gambar 2.9 Poin-poin pada *Ownership*

*Ownership* adalah saat pengguna merasa memiliki sesuatu di dalam permainan seperti pemain memiliki perasaan ingin mendapatkan mata uang atau *item* dalam permainan. Tujuan dari poin ini adalah pengguna menghabiskan waktu yang lama untuk mengatur profil atau *avatar*.

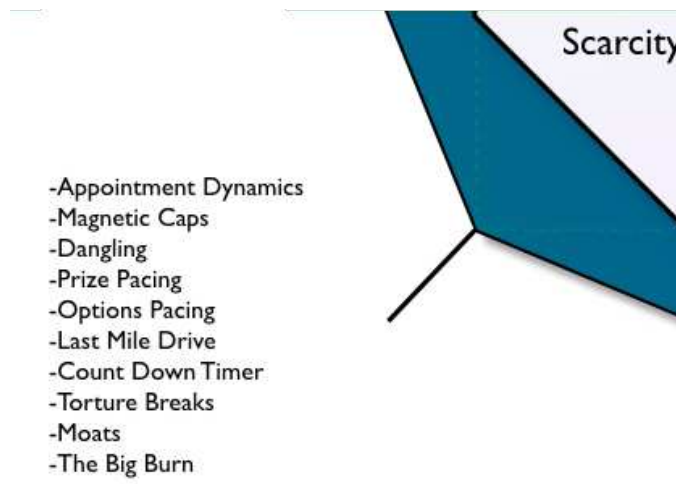
e. *Social Influence*



Gambar 2.10 Poin-poin pada *Social influence*

*Social influence* adalah mengimplementasikan elemen sosial seperti pertemanan dan kompetisi kedalam elemen permainan. Pengguna akan berkompetisi apabila *level* dari teman pengguna lebih jauh darinya.

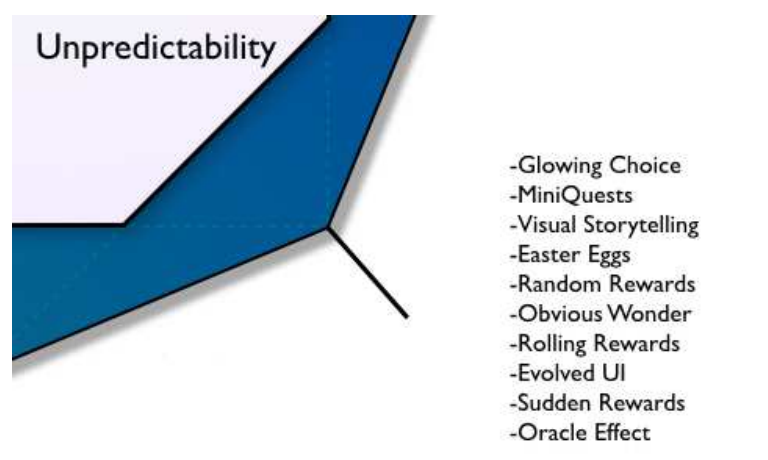
f. *Scarcity*



Gambar 2.11 Poin-poin pada *Scarcity*

*Scarcity* berarti pemain menginginkan sesuatu akan tetapi tidak dapat memilikinya. Banyak permainan yang sudah menetapkan *appointment dynamics* sebagai contoh kotak hadiah yang didapat hanya dapat dibuka setelah 10 jam.

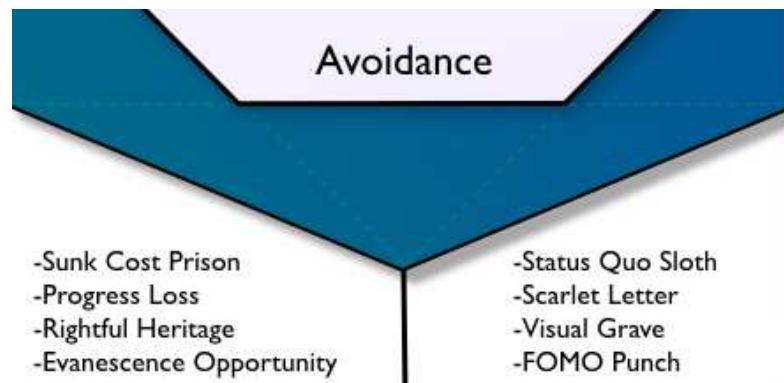
g. *Unpredictability*



Gambar 2.12 Poin-poin pada *Unpredictability*

*Unpredictability* adalah pengguna tidak dapat mengantisipasi atau mengetahui apa yang akan terjadi selanjutnya. *Unpredictability* menjadi salah satu penyebab dari kecanduan judi.

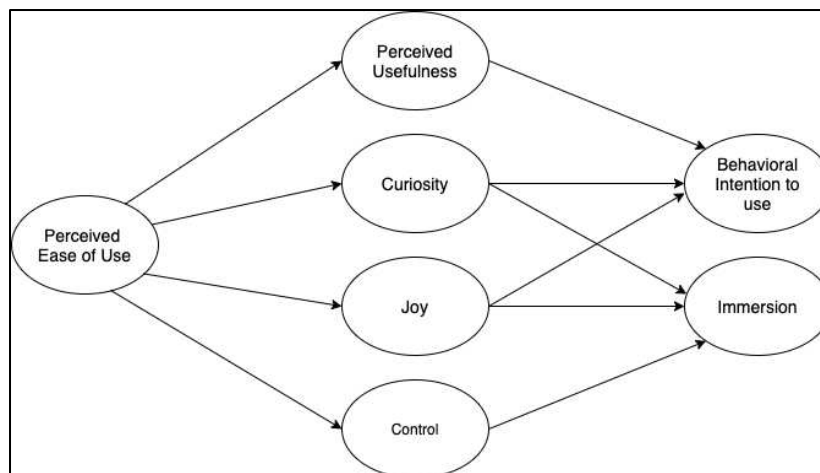
h. *Avoidance*



Gambar 2.13 Poin-poin pada *Avoidance*

*Avoidance* adalah saat dimana pengguna berusaha menghindari konsekuensi negatif yang terjadi saat pengguna tidak bertindak cepat. Apabila pengguna tidak bertindak cepat maka pengguna dapat kehilangan kesempatan selamanya.

**2.3 Hedonic Motivation System Adoption Model (HMSAM)**



Gambar 2.14 HMSAM Model (Lowry, Gaskin, Twyman, Hammer, & L., 2013)

HMSAM adalah sebuah model yang digunakan untuk mengukur motivasi intrinsik dari suatu sistem yang mengimplementasi motivasi hedonis. Model dari HMSAM diilustrasikan pada Gambar 2.14.

1. *Perceived ease of use* (PEOU), mengukur kemudahan dalam penggunaan dari suatu sistem.
2. *Perceived usefulness* (PU), mengukur kinerja penggunaan dari suatu sistem.
3. *Curiosity* (C), mengukur rasa ingin tahu dari pengguna.
4. *Joy* (J), mengukur rasa kesenangan yang didapat saat berinteraksi dengan sistem.
5. *Control* (Co), mengukur persepsi pengguna terhadap interaksi dengan sistem.
6. *Behavioral intention to use* (BIU), mengukur keinginan pengguna untuk menggunakan aplikasi.
7. *Focused immersion* (I), mengukur seberapa fokus pengguna dalam menggunakan sistem saat pengguna menggunakan aplikasi.

HMSAM memiliki dua aspek yaitu *Behavioral intention to use* dan *Focused immersion*. Aspek *Behavioral intention to use* ditentukan dari aspek *Perceived ease of use*, *perceived usefulness*, *curiosity*, dan *joy* sedangkan *imeersion* ditentukan dari aspek *curiosity*, *joy*, dan *control*. Adapun skala pengukuran dari HMSAM yang ditunjukkan oleh Tabel 2.1.

Tabel 2.1 Skala Pengukuran (Lowry, Gaskin, Twyman, Hammer, & L., 2013)

Construct	Items
Perceived Ease of Use	<p><b>PEOU1.</b> My interaction with the game was clear and understandable.</p> <p><b>PEOU2.</b> Interacting with the game did not require a lot of my mental effort.</p> <p><b>PEOU3.</b> I found the game to be trouble free.</p> <p><b>PEOU4.</b> I found it easy to get the game to do what I want it to do.</p> <p><b>PEOU5.</b> Learning to operate the game was easy for me.</p> <p><b>PEOU6.</b> It was simple to do what I wanted with the game.</p> <p><b>PEOU7.</b> It was be easy for me to become skillful at using the game.</p> <p><b>PEOU8.</b> I found the game easy to use.</p>
Perceived Usefulness	<p><b>PU1.</b> The game decreased my stress.</p> <p><b>PU2.</b> The game helped me better pass time.</p> <p><b>PU3.</b> The game provided a useful escape.</p> <p><b>PU4.</b> The game helped me think more clearly.</p> <p><b>PU5.</b> The game helped me feel rejuvenated.</p>
Curiosity	<p><b>CUR1.</b> This experience excited my curiosity.</p> <p><b>CUR2.</b> This experience made me curious.</p> <p><b>CUR3.</b> This experience aroused my imagination.</p>
Control	<p><b>CTL1.</b> I had a lot of control. Modified original scale from (Liu, 2003) to be more general so that it did not focus solely on Web sites. Also, added items on control from.</p> <p><b>CTL2.</b> I could choose freely what I wanted to see or do.</p> <p><b>*CTL3.</b> I had little control over what I could do.</p> <p><b>CTL4.</b> I was in control.</p> <p><b>*CTL5.</b> I had no control over my interaction.</p> <p><b>CTL6.</b> I was allowed to control my interaction.</p>
Joy	<p><b>JOY1.</b> I found playing the game to be enjoyable.</p> <p><b>JOY2.</b> I had fun using the game.</p> <p><b>*JOY3.</b> Using the game was boring.</p> <p><b>*JOY4.</b> The game really annoyed me</p> <p><b>JOY5.</b> The game experience was pleasurable.</p> <p><b>*JOY6.</b> The game left me unsatisfied.</p>
Behavioral Intention to Use	<p><b>BIU1.</b> I would plan on using it in the future.</p> <p><b>BIU2.</b> I would intend to continue using it in the future.</p> <p><b>BIU3.</b> I expect my use of it to continue in the future.</p>
Focused Immersion	<p><b>FI1.</b> I was able to block out most other distractions.</p> <p><b>FI2.</b> I was absorbed in what I was doing.</p> <p><b>FI3.</b> I was immersed in the game.</p> <p><b>*FI4.</b> I was distracted by other attentions very easily.</p> <p><b>FI5.</b> My attention was not diverted very easily.</p>

Tanda \* menandakan bahwa perhitungan dengan tanda tersebut diskalakan terbalik seperti dalam Skala Likert “Sangat Tidak Setuju” memiliki nilai yang sama dengan “Sangat Setuju”.

#### 2.4 Skala Likert

Skala Likert adalah skala yang digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang mengenai fenomena sosial (Sugiyono, 2012). Pengukuran dapat dilakukan dengan cara memberikan sebuah pertanyaan kepada responden dengan jawaban dari 5 pilihan jawaban yang memiliki nilai jawaban yang berbeda. Skala Likert pada umumnya menggunakan 5 angka penelitian.

1. Sangat Setuju (SS) memiliki nilai 5.
2. Setuju (S) memiliki nilai 4.
3. Ragu-ragu (RR) memiliki nilai 3.
4. Tidak Setuju (TS) memiliki nilai 2.
5. Sangat Tidak Setuju (STS) memiliki nilai 1.

Rumus 2.1 dapat digunakan untuk menentukan persentase skor yang didapat dari pengambilan data kuisisioner (Sugiyono, 2012). Rumus 2.2 digunakan untuk kode pertanyaan HMSAM yang memiliki tanda \*.

$$\text{Persentase} = \frac{(SS*5)+(S*4)+(RR*3)+(TS*2)+(STS*1)}{5*\text{Jumlah Responden}} \quad (2.1)$$

$$\text{Persentase} = \frac{(STS*5)+(TS*4)+(RR*3)+(S*2)+(SS*1)}{5*\text{Jumlah Responden}} \quad (2.2)$$

Keterangan:

PS = Persentase Skor

SS = Sangat Setuju

S = Setuju

RR = Ragu-ragu

TS = Tidak Setuju

STS = Sangat Tidak Setuju