

BAB II TINJAUAN PUSTAKA

2.1. Jejak Digital (*Digital Footprint*)

Menurut Website *Internet Society* (2020), jejak digital (*digital footprint*) adalah semua data yang ditinggalkan saat seseorang menggunakan internet, seperti komentar di media sosial, panggilan telepon atau video, serta pengiriman *email*, yang menjadi bagian dari riwayat *online* pengguna dan berpotensi dilihat oleh pengguna lain atau dilacak di dalam *database*.

Menurut Önder dan Hubmann-Haidvogel (2014), jejak digital dapat digunakan untuk melakukan penelitian dengan berbagai bentuk, seperti foto yang diunggah, ulasan yang diposting, atau tautan yang ditekan. Data ini dikumpulkan oleh sebagian besar situs *web* sebagai sumber data untuk penelitian dan pemasaran. Misalnya, rekomendasi produk yang dibuat untuk seseorang oleh situs *web* didasarkan pada data-data yang terkumpul dari aktivitas *online* orang tersebut.

Menurut Sjöberg *et al.* (2017), jejak digital dapat menunjukkan identitas seseorang, sejarah perjalannya di dunia maya, karakter dari orang tersebut, bahkan dapat menunjukkan lokasi keberadaan orang tersebut secara *real time*. Jejak digital dalam jumlah besar tersebut dapat dianalisis untuk mendapatkan wawasan tentang perilaku pengguna, preferensi dan kebutuhan pribadi, serta memprediksi tindakan pengguna di masa depan.

Menurut Website Tech Terms (2019), jejak digital terdiri atas jejak digital pasif dan aktif. Jejak digital pasif adalah jejak data yang pengguna tinggalkan secara tidak sengaja atau tanpa sepengetahuan pengguna saat berselancar di internet. Misalnya, ketika pengguna mengunjungi situs web, server web dapat mencatat alamat *Internet Protocol (IP)* pengguna, mengidentifikasi penyedia layanan internet pengguna dan perkiraan lokasi pengguna dan riwayat pencarian pengguna. Sedangkan jejak digital aktif adalah data-data yang pengguna kirimkan secara sengaja di internet, contohnya yaitu aktivitas mengirim email. Aktivitas mengirimkan *email* berkontribusi pada jejak digital aktif pengguna, karena pengguna mengharapkan data dilihat atau disimpan oleh pengguna lain.

Adapun pada penelitian ini, jejak digital yang akan disimpan dan dikelola berupa foto, titik lokasi pengunjung tempat wisata yang valid berada di area tempat wisata, serta takarir (*caption*), penilaian (*rating*) dan ulasan (*review*) dari pengunjung terhadap suatu tempat wisata.

2.2. Tempat wisata

Tempat wisata diartikan sebagai lokasi yang menjadi tujuan wisata karena mempunyai objek wisata dan hal-hal yang menarik untuk sebuah kunjungan wisata (Raharjanto, 1993). Dengan kata lain, tempat wisata ini dapat disebut sebagai destinasi pariwisata.

Menurut Undang-Undang Nomor 10 Tahun 2009 tentang Kepariwisata, destinasi pariwisata adalah kawasan geografis yang berada dalam satu atau lebih wilayah administratif yang di dalamnya terdapat daya tarik wisata, fasilitas umum, fasilitas pariwisata, aksesibilitas, serta masyarakat yang saling terkait dan melengkapi terwujudnya kepariwisataan. Pariwisata adalah berbagai macam kegiatan wisata dan didukung berbagai fasilitas serta layanan yang disediakan oleh masyarakat, pengusaha, dan pemerintah. Wisata adalah kegiatan perjalanan yang dilakukan oleh seseorang atau sekelompok orang dengan mengunjungi tempat tertentu untuk tujuan rekreasi, pengembangan pribadi, atau mempelajari keunikan daya tarik wisata yang dikunjungi dalam jangka waktu sementara. Daya tarik wisata adalah segala sesuatu yang memiliki keunikan, keindahan, dan nilai yang berupa keanekaragaman kekayaan alam, budaya, dan hasil buatan manusia yang menjadi sasaran atau tujuan kunjungan wisatawan.

Menurut Ali (2015), wisata berdasarkan jenis-jenisnya dapat dibagi ke dalam dua kategori, yaitu:

1. Wisata Alam, yang terdiri dari:
 - a) Wisata pantai (*Marine Tourism*), merupakan kegiatan wisata yang ditunjang oleh sarana dan prasarana untuk berenang, memancing, menyelam, dan olahraga air lainnya.
 - b) Wisata Etnik (*Ethnic Tourism*), merupakan perjalanan untuk mengamati perwujudan kebudayaan dan gaya hidup masyarakat yang dianggap menarik.

- c) Wisata Cagar Alam (*Eco Tourism*), merupakan wisata yang banyak dikaitkan dengan kegemaran akan keindahan alam, kesegaran hawa di pegunungan, keajaiban hidup binatang (margasatwa) yang langka, serta tumbuh-tumbuhan yang jarang terdapat di tempat-tempat lain.
 - d) Wisata Pertanian (Agrowisata), merupakan jenis wisata yang mengorganisasikan perjalanan ke proyek-proyek pertanian, perkebunan, dan ladang pembibitan.
2. Wisata Sosial Budaya, yang terdiri dari peninggalan sejarah kepurbakalaan, monumen nasional, museum dan tempat-tempat lainnya, yang memiliki nilai sosial dan budaya bagi pengunjung.

Selain kedua jenis wisata tersebut, ada juga jenis-jenis wisata lainnya yaitu: wisata kuliner, wisata belanja, wisata olahraga, wisata konvensi/politik, wisata religi dan wisata edukasi.

Di kota Pontianak, yang merupakan ibukota dari provinsi Kalimantan Barat, terdapat banyak tempat wisata yang cukup terkenal di kalangan para wisatawan (pengunjung tempat wisata), yakni: Tugu Khatulistiwa, Taman Digulis, Taman Catur Untan, Taman Alun Kapuas, Keraton Kadriyah, Museum Kalimantan Barat, Masjid Raya Mujahidin, Rumah Betang Radakng, Taman Akcaya, Kampung Wisata Kuantan, Aloe vera Center, Makam Kesultanan Batulayang, Mega Mall, Gaia Bumi Raya City, Transmart, Universitas Tanjungpura, Perpustakaan Kalimantan Barat dan lain-lain. **Gambar 2.1** berikut ini menunjukkan salah satu tempat wisata di kota Pontianak, yakni Tugu Khatulistiwa.



Gambar 2.1. Tempat Wisata Tugu Khatulistiwa di Pontianak (Bragi, 2020)

Umumnya terdapat dua aktor utama yang terlibat dalam suatu sistem tempat wisata, yakni pengunjung tempat wisata dan pengelola tempat wisata. Pengunjung tempat wisata adalah orang-orang yang mengunjungi tempat wisata untuk berbagai tujuan, seperti untuk *refreshing*, belajar, pertemuan dan sebagainya karena adanya objek wisata di tempat tersebut. Pengelola tempat wisata adalah orang-orang yang memiliki dan mengelola suatu tempat wisata dengan tujuan untuk mendapatkan keuntungan dari pengunjung. Interaksi yang terjadi di antara kedua aktor tersebut dapat dilakukan melalui sebuah aplikasi *mobile* berbasis Android.

2.3. Android

Android adalah sebuah *platform* untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android bersifat *open source* bagi para pengembang. Awalnya, Android didirikan oleh *Android Inc.* untuk mewujudkan *mobile device* yang lebih peka terhadap lokasi dan preferensi pengguna. Namun, *Android Inc.* pada akhirnya diakuisisi oleh *Google Inc.* dan Android dikembangkan menjadi sebuah *platform*. Saat ini versi terbaru android adalah Android 11, sedangkan Android 12 masih dalam proses pengembangan (Website Android Developers, 2021).

Menurut Safaat (2012), kelebihan Android adalah sebagai berikut:

1. Lengkap (*complete*)

Android merupakan sistem operasi yang aman dan banyak menyediakan *tools* dalam membangun *software* sehingga memfasilitasi pengembangan aplikasi-aplikasi berbasis Android.

2. Terbuka (*open source*)

Platform Android bersifat *open-source*. Pengembang dapat dengan bebas untuk mengembangkan aplikasi berbasis Android. Android sendiri menggunakan Linux Kernel 2.6.

3. Gratis (*free*)

Android adalah *platform* aplikasi yang bebas, tidak ada lisensi atau biaya royalti, serta tidak ada biaya keanggotaan yang diperlukan maupun biaya untuk pengujian.

2.4. *Location-Based Services (LBS)*

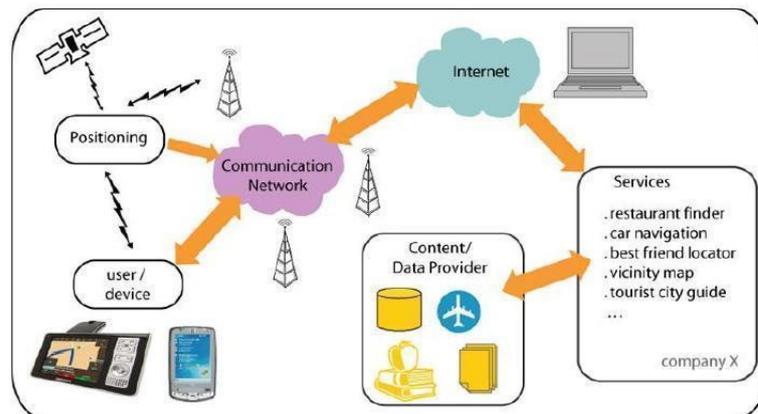
Location-Based Services (LBS) adalah layanan yang memberikan informasi berdasarkan lokasi suatu *device* dengan teknologi *Global Positioning System (GPS)*. Menurut Budiman (2016), terdapat dua unsur utama pada *LBS* yaitu:

1. *Location Manager*

Location Manager menyediakan *tools/source* untuk *LBS*, yang berupa *Maps Application Programming Interface (API)* untuk menyediakan fasilitas menampilkan dan memanipulasi peta beserta fitur-fitur lainnya seperti tampilan satelit, street (jalan), dan gabungan keduanya.

2. *Location Provider*

Location Provider menyediakan teknologi pencarian lokasi yang digunakan oleh *device*. *Location Application Programming Interface (API)* berhubungan dengan data *Global Positioning System (GPS)* dan data lokasi *real time*. Dengan *Location Manager*, pengguna dapat menentukan lokasi *real time*, track gerakan/perpindahan, serta kedekatan dengan lokasi tertentu dengan mendeteksi perpindahan.



Gambar 2.2. Komponen *Location-Based Services (LBS)*.

Menurut (Udka *et al.*, 2015), komponen-komponen pendukung dari *Location-Based Services (LBS)* seperti yang ditunjukkan pada **Gambar 2.2** diatas, terdiri atas:

1. *Mobile Devices*, yakni alat untuk meminta informasi. Hasil dari informasi yang diminta dapat berupa tulisan, suara, gambar dan sebagainya. *Mobile*

devices yang dapat digunakan bisa berupa *personal digital assistant (PDA)*, *smartphone*, dan *laptop*.

2. *Communication Network*, yakni komponen yang berfungsi sebagai jalur penghubung yang dapat mengirimkan data-data yang dikirim oleh pengguna dari *mobile device* ke *service provider*. Hasil permintaan tersebut dikirimkan kembali oleh *service provider* kepada pengguna. *Communication network* dapat berupa jaringan seluler (*GSM* dan *CDMA*), *Wireless Local Area Network (WLAN)*, atau *Wireless Wide Area Network (WWAN)*.
3. *Position Component*, yakni komponen yang menentukan posisi pengguna. Posisi ini dapat didapatkan melalui *Communication Network* ataupun *Global Positioning System (GPS)*.
4. *Service and Application Provider*, yakni penyedia layanan yang menawarkan berbagai macam layanan kepada pengguna dan bertanggung jawab untuk memproses informasi yang diminta oleh pengguna. Contohnya ketika pengguna meminta layanan agar diketahui posisinya secara *real time* menuju suatu tempat, maka *service and application provider* akan memproses permintaan tersebut, mulai dari menghitung dan menentukan posisi pengguna, sampai menemukan rute jalan ke tempat tujuan.
5. *Data and Content Provider*, yakni penyedia layanan yang menyimpan semua data yang dibutuhkan yang bisa diakses pengguna.

Pada penelitian ini, layanan yang akan digunakan dalam *LBS* adalah layanan yang dapat menampilkan dan memanipulasi peta, contohnya menampilkan titik lokasi tempat wisata dalam bentuk *marker* dan nama dari tempat wisata tersebut pada peta, sehingga aplikasi jejak digital yang akan dihasilkan dapat lebih mudah dipahami oleh pengguna.

2.5. *Global Positioning System (GPS)*

Menurut Susilo dkk. (2014), *GPS* adalah singkatan dari *Global Positioning System* yang merupakan suatu sistem untuk menentukan posisi dan navigasi secara global dengan menggunakan satelit dan metode triangulasi (*triangulation*). Sistem

ini pertama kali dikembangkan oleh departemen pertahanan Amerika yang awalnya diperuntukkan untuk kepentingan militer.

Menurut Tambunan dan Putra (2019), *GPS* memiliki tiga segmen yaitu: segmen satelit (*space segment*), segmen pengendali (*control segment*), dan segmen penerima/pengguna (*user segment*). Satelit *GPS* yang mengorbit bumi seluruhnya berjumlah 24 buah yang terdiri atas 21 buah yang aktif bekerja dan 3 buah sisanya adalah cadangan. Ilustrasi Satelit *GPS* yang mengorbit bumi ditunjukkan pada **Gambar 2.3** berikut ini.



Gambar 2.3. Ilustrasi Satelit *GPS* yang mengorbit Bumi (Erickson, 2019).

Menurut Susilo dkk. (2014), satelit berfungsi untuk menerima dan menyimpan data yang ditransmisikan oleh stasiun-stasiun pengendali, serta menyimpan dan menjaga informasi waktu berketelitian tinggi berdasarkan jam atom di satelit, serta memancarkan sinyal dan informasi secara berkelanjutan ke penerima (*receiver*). Segmen pengendali berfungsi untuk mengendalikan satelit dari bumi untuk melihat keadaan satelit, penentuan serta prediksi orbit, sinkronisasi waktu antar satelit, dan mengirimkan data ke satelit. Segmen penerima berfungsi untuk menerima data dari satelit dan memprosesnya untuk menentukan posisi, arah, jarak dan waktu yang diperlukan oleh pengguna *GPS* pada perangkat.

Adapun cara kerja *GPS* menurut Sumantri dkk. (2019) adalah sebagai berikut ini:

1. Satelit berputar mengelilingi bumi selama 12 jam di dalam orbit yang akurat dan mengirimkan sinyal ke *receiver*. Kemudian, *receiver* menerima sinyal itu dan menentukan posisi *user* melalui perbandingan antara waktu

sinyal tersebut dikirim dan diterima dengan menggunakan metode *triangulation*.

2. Sebuah *receiver* harus mengunci sinyal minimal dari tiga satelit untuk menghitung titik lokasi 2D (*latitude* dan *longitude*) dan pergerakan (*track*) *user*. Jika *receiver* dapat menerima sinyal dari empat atau lebih satelit, maka *receiver* dapat menghitung titik lokasi 3D (*latitude*, *longitude* dan *altitude*) *user*. Setelah posisi *user* sudah ditentukan, selanjutnya *receiver* akan menghitung informasi lainnya, seperti kecepatan, arah yang dituju, jalur tujuan perjalanan, jarak ke tujuan, waktu matahari terbit dan terbenam, dan lain-lain.
3. Satelit sangat akurat dalam mengirim informasi waktu karena satelit menggunakan jam atom yang digerakkan oleh partikel atom yang diisolasi. Akurasi informasi waktu ini akan menentukan akurasi perhitungan lokasi *user*. Selain itu, semakin banyak sinyal satelit yang diterima *receiver* maka semakin akurat pula data yang diterima *receiver* karena satelit-satelit mengirim *pseudo-random code* dan data waktu secara bersamaan. Ketinggian satelit juga berpengaruh pada proses kerja *GPS*, karena semakin tinggi posisi satelit maka semakin sedikit gangguan pada satelit di atmosfer sehingga satelit akan selalu mengorbit pada posisi yang tepat. Adapun manfaat dari *GPS* menurut Sumantri dkk. (2019) adalah sebagai berikut ini:

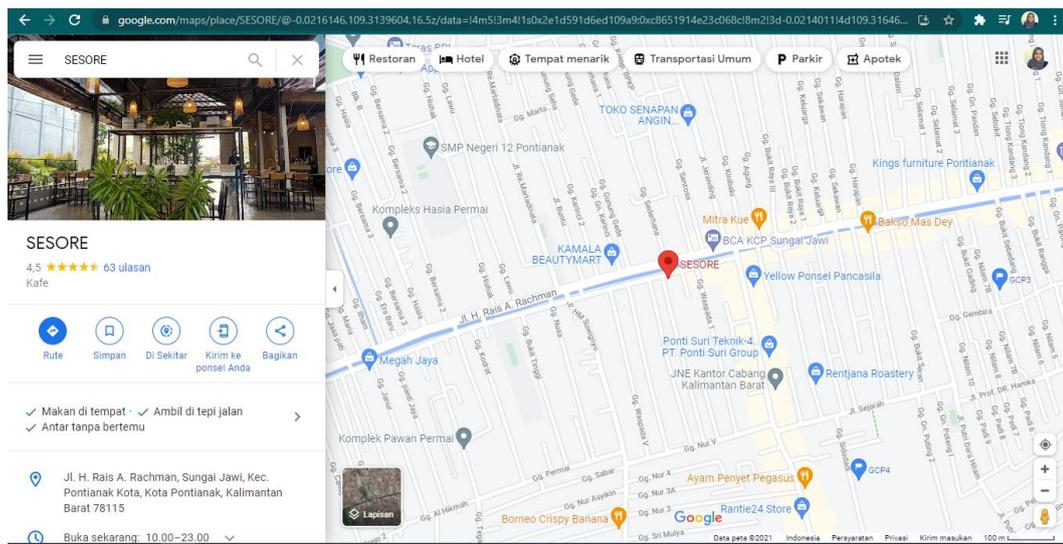
1. *GPS* untuk militer yang dimanfaatkan untuk mendukung sistem pertahanan militer, seperti memantau pergerakan musuh saat terjadi peperangan atau penuntun arah jatuhnya bom sehingga tepat pada target.
2. *GPS* untuk navigasi yang dimanfaatkan untuk navigasi kendaraan, dimana *GPS* mendeteksi posisi perangkat pengendara atau *GPS tracker* pada kendaraan secara *realtime*, kemudian diterjemahkan ke dalam bentuk peta digital. Sehingga pengendara dapat mengetahui arah, jalur dan tujuan jalan melalui web/aplikasi peta digital tersebut.
3. *GPS* untuk sistem informasi geografis yang digunakan untuk keperluan sistem informasi geografis, seperti untuk pembuatan peta, mengukur jarak perbatasan, atau sebagai referensi pengukuran suatu wilayah.

4. *GPS* untuk sistem pelacakan kendaraan yang digunakan untuk melacak posisi kendaraan.
5. *GPS* untuk pemantau gempa, yang dimanfaatkan untuk memantau pergerakan tanah di bumi untuk memantau kemungkinan terjadinya gempa.

Pada penelitian ini, *GPS* akan digunakan untuk menentukan lokasi pengguna aplikasi saat akan membagikan jejak digital di suatu tempat wisata pada aplikasi jejak digital yang akan dibangun. *GPS* tersebut terdapat pada *smartphone* Android pengguna

2.6. Google Maps

Google Maps adalah sebuah layanan peta global virtual gratis dan *online* yang disediakan oleh Google. *Google Maps* dapat diakses di alamat <http://maps.google.com>. *Google Maps* menyediakan peta yang dapat digeser serta pencitraan satelit untuk seluruh dunia. *Google Maps* juga menawarkan pencarian suatu tempat dan rute perjalanan. **Gambar 2.4** berikut ini adalah tampilan dari *Google Maps*.



Gambar 2.4. Tampilan *Google Maps*.

Google Maps dapat digunakan untuk pengembangan aplikasi dengan memanfaatkan *Google Maps API*, yakni *Maps API* dari Google untuk menyediakan

fasilitas untuk menampilkan, memanipulasi peta beserta fitur-fitur lainnya seperti tampilan satelit, street (jalan), maupun gabungannya, menambah konten melalui berbagai jenis pelayanan yang dimiliki, serta mengizinkan kepada pengembang untuk membangun aplikasi *enterprise* di dalam *website*-nya.

Adapun fitur yang terdapat pada *Google Maps* antara lain adalah menjelajah peta, mencari lokasi tertentu seperti hotel, tempat hiburan, dan lokasi bisnis, serta menghitung rute dalam berkendara. *Google Maps* mempunyai fasilitas *zoom* yang berfungsi untuk mengubah fokus. Terdapat 18 tingkatan *zoom* yang dapat dipilih dengan cara menekan tombol *plus/minus*, atau dengan menggeser bilah geser yang terdapat di antara tanda *plus/minus*.

Salah satu keunggulan dari *Google Maps* adalah dapat menyediakan dua jenis tampilan yaitu *maps* dan *satellite*. *Maps* merupakan pencitraan yang diwujudkan dalam bentuk ilustrasi peta. *Satellite* merupakan pencitraan yang dihasilkan dalam bentuk citra satelit sehingga pengguna dapat melihat keadaan permukaan bumi secara nyata dari lokasi yang diminta.

Pada penelitian ini, *Google Maps API* yang akan diintegrasikan ke dalam aplikasi jejak digital diakses dengan *framework* *React Native* dan *Expo*. Data-data yang dimanipulasi pada peta berupa data *JavaScript Object Notation (JSON)*, yang disimpan dalam *Firebase* yang diakses melalui mekanisme pertukaran data di *Node JS*.

2.7. React Native

Menurut buku Eisenman (2016), *React Native* adalah *framework* bahasa pemrograman *JavaScript* yang digunakan untuk membuat aplikasi *native* yang mampu berjalan di *platform* *Android* dan *iOS*. *Framework* ini berbasis *React JS* yang merupakan *framework JavaScript* buatan *Facebook* untuk membuat *web*. *React Native* digunakan untuk membuat tampilan aplikasi yang berfokus pada aplikasi *mobile*. Aplikasi *React Native* ditulis menggunakan campuran *JavaScript* dan *markup XML*, yang dikenal sebagai *JSX*. *React Native* ini menjembatani pemanggilan *Native Rendering API* di *Objective-C* (untuk *iOS*) atau di *Java* (untuk *Android*).

Framework React Native lebih tepatnya digunakan untuk pembangunan aplikasi dengan model *Hybrid*, yakni model pengembangan aplikasi yang menggabungkan pendekatan aplikasi *native* dan aplikasi *web* untuk dijadikan aplikasi *mobile* yang kemudian dikonversi ke lebih dari satu *platform*. Kelebihan dari pengembangan aplikasi *hybrid* adalah proses pengembangannya cepat dikarenakan sebuah kode program dapat dijalankan pada lebih dari satu sistem operasi *mobile*.

Penelitian perbandingan React Native dengan *framework* aplikasi *mobile* lainnya dilakukan oleh Wijonarko dkk. pada tahun 2018 yang berjudul “Perbandingan Phonegap dan React Native sebagai *Framework* Pengembangan Aplikasi *Mobile*”. Phonegap (yang saat ini dikenal sebagai Apache Cordova) merupakan *framework* aplikasi *mobile* berbasis HTML5, CSS, dan JavaScript. Pada penelitian tersebut, *framework* Phonegap dan *framework* React Native digunakan untuk membangun aplikasi *mobile* Sistem Informasi Jadwal dan Nilai Politeknik Kota Malang. Hasil penelitiannya adalah kedua *framework* tersebut dapat digunakan untuk membangun aplikasi tersebut, namun React Native menghasilkan tampilan yang lebih alami dan asli layaknya aplikasi yang dibangun dengan bahasa *native* Java untuk Android, sedangkan Phonegap sangat bergantung pada *framework* CSS yang digunakan peneliti, yakni Framework7 CSS, untuk tampilan aplikasinya.

Sebelum aplikasi *mobile* dibangun dengan *React Native*, diperlukan adanya pengaturan *environment* pembangunan aplikasi terlebih dahulu. Pada penelitian ini, kode-kode React Native dijalankan dengan menggunakan *framework* Expo. Expo adalah sebuah *framework* dan *platform* untuk pembangunan aplikasi *React* universal, yakni aplikasi *web*, Android, dan iOS. Expo berupa sekumpulan alat-alat *Software Development Kit (SDK)*, *library*, dan layanan-layanan yang digunakan dalam pengembangan aplikasi berbasis *React Native*.

Penelitian mengenai ada atau tidaknya akseptabilitas pengembang terhadap sistem React Native dengan Expo berdasarkan kriteria *usability* telah dilakukan oleh Yunandar dan Priyono pada tahun 2018 dengan judul “Pengujian *Usability System Framework* React Native dengan Expo untuk Pengembang Aplikasi Android Menggunakan *Use Questionnaire*”. Instrumen penelitian tersebut

menggunakan kuesioner yang terdiri dari 13 pertanyaan yang dikelompokkan menjadi lima variabel *usability*, yakni *Learnability*, *Efficiency*, *Memorability*, *Errors* dan *Satisfaction*. Responden dari penelitian tersebut berjumlah 25 responden dengan latar belakang para pengembang aplikasi *mobile* yang sudah mengetahui aplikasi Android. Berdasarkan penelitian tersebut, hasil pengujian rekap nilai *usability* menunjukkan keseluruhan variabel pengujian mendapatkan nilai *usability* oleh pengembang, rata-rata di atas nilai 3, sehingga dapat dikatakan bahwa sistem pada React Native dengan Expo memiliki nilai aspek *Usability* dan sangat layak digunakan oleh semua kalangan pengembang aplikasi *mobile*, dikarenakan mudah dipelajari serta dimengerti oleh pengembang.

Adapun komponen-komponen penting yang dibutuhkan pada aplikasi jejak digital kunjungan tempat tempat wisata yang akan dibangun adalah sebagai berikut ini:

2.7.1. Kamera

Kamera adalah komponen yang digunakan untuk merekam pratinjau pada kamera depan ataupun kamera belakang di *smartphone*. Komponen ini didapatkan dari *library expo-camera*. Komponen kamera membutuhkan konfigurasi *permission.camera* yang diletakkan di *file* JSON agar aplikasi yang akan dibuat terdaftar untuk dapat menggunakan kamera pada perangkat Android. **Kode 2.1** menunjukkan cara mengakses kamera pada Expo. Pada kode tersebut, terdapat suatu *method* `Camera.getCameraPermissionAsync()` untuk meminta pengguna aplikasi memberikan izin akses ke kamera Android.

Kode Program 2.1. Mengakses Kamera dan Meminta Izin Akses Kamera pada *User* dengan Expo.

```

1 import React, { useState, useEffect } from 'react';
2 import { StyleSheet, Text, View, TouchableOpacity } from
  'react-native';
3 import { Camera } from 'expo-camera';
4 export default function App() {
5   const [hasPermission, setHasPermission] =
     useState(null);
6   const [type, setType] =
     useState(Camera.Constants.Type.back);
7   useEffect(() => {
8     (async () => {
```

```

9  const { status } = await
    Camera.requestCameraPermissionsAsync();
10 setHasPermission(status === 'granted');})());, []);
11 if (hasPermission === null) {
12 return <View />;}
13 if (hasPermission === false) {
14 return <Text>No access to camera</Text>;}
15 return (
16 <View style={styles.container}>
17 <Camera style={styles.camera} type={type}>
18 <View style={styles.buttonContainer}>
19 <TouchableOpacity style={styles.button} onPress={() => {
20 setType(type === Camera.Constants.Type.back ?
    Camera.Constants.Type.front :
    Camera.Constants.Type.back);}}>
21 <Text style={styles.text}> Flip </Text>
22 </TouchableOpacity>
23 </View>
24 </Camera>
25 </View>);}
26 const styles = StyleSheet.create({ ... });

```

2.7.2. Peta

Peta merupakan komponen pada aplikasi yang digunakan untuk menampilkan lokasi tempat wisata dan lokasi pengunjung secara visual di permukaan bumi. Komponen pada Expo untuk menampilkan *Google Maps* disebut dengan *MapView*. *MapView* membutuhkan adanya instalasi *library react-native-maps*. Kemudian, *library* tersebut dapat diakses pada kode program. **Kode 2.2** berikut ini menunjukkan cara akses peta pada Expo.

Kode Program 2.2. Menampilkan *Google Maps* dengan Expo.

```

1  import * as React from 'react';
2  import MapView from 'react-native-maps';
3  import { StyleSheet, Text, View, Dimensions } from
    'react-native';
4  export default function App() {
5  return (
6  <View style={styles.container}>
7  <MapView
8  style={styles.map}
9  initialRegion={{
10 latitude: 0.000000,
11 longitude: 109.333336,
12 latitudeDelta: 0.0922,
13 longitudeDelta: 0.0421,}}/>
14 </View>

```

```

15 );}
16 const styles = StyleSheet.create({
17   container: {
18     flex: 1,
19     backgroundColor: '#fff',
20     alignItems: 'center',
21     justifyContent: 'center', },
22   map: {
23     width: Dimensions.get('window').width,
24     height: Dimensions.get('window').height, }, }, });

```

2.8. Firebase

Firebase merupakan penyedia layanan *cloud* dan *backend* untuk pengembangan aplikasi di berbagai platform, seperti Android, iOS, web, C++, dan Unity. Adapun untuk menggunakan *console* Firebase, dapat diakses melalui link <https://console.firebase.google.com/>. Menurut Kurniawan dkk. (2007), Firebase memberikan kemudahan, serta meningkatkan keamanan dan kecepatan dalam pengembangan aplikasi karena *Application Programming Interface (API) backend* yang digunakan untuk berkomunikasi dengan *server* telah disediakan, sehingga tidak membutuhkan proses yang panjang seperti pembangunan aplikasi pada umumnya yang mengembangkan *REST API*.

Adapun beberapa layanan Firebase untuk mengembangkan aplikasi berdasarkan Website Firebase (2022), yang dibutuhkan untuk pembangunan aplikasi jejak digital adalah sebagai berikut ini:

2.8.1. *Firestore Authentication*

Firestore Authentication menyediakan layanan backend, *SDK* yang mudah digunakan, dan library *User Interface (UI)* siap pakai untuk mengautentikasi (mengetahui identitas) pengguna ke aplikasi, serta menyimpan data pengguna secara aman di *Firestore Cloud Storage*. *Firestore Authentication* mendukung autentikasi menggunakan sandi, nomor telepon, serta penyedia identitas gabungan yang populer seperti akun Google, Facebook, Twitter, dan lain-lain.

2.8.2. *Firestore Cloud Storage*

Firestore Cloud Storage menyediakan penyimpanan berbasis *cloud* yang berisi konten-konten buatan pengguna (*user*) aplikasi, seperti gambar, audio, video

dan konten lainnya. Penyimpanan ini diberikan secara gratis dengan kapasitas hingga 5 GB.

2.8.3. *Firestore Database*

Firestore Database merupakan salah satu layanan dari Firebase yang bertujuan untuk melakukan manajemen *database*, bersifat NoSQL dan keluarannya dalam bentuk JSON. Layanan ini sangat optimal untuk digunakan karena kemampuannya dalam melakukan proses komunikasi dengan *user* dengan sangat cepat.

2.8.4. *Cloud Messaging*

Cloud Messaging merupakan salah satu layanan Firebase yang memungkinkan agar aplikasi saling berkiriman pesan dan menerima pemberitahuan secara *real time*. Penerapan layanan ini cocok digunakan untuk fitur *chatting* dan *handle* notifikasi.

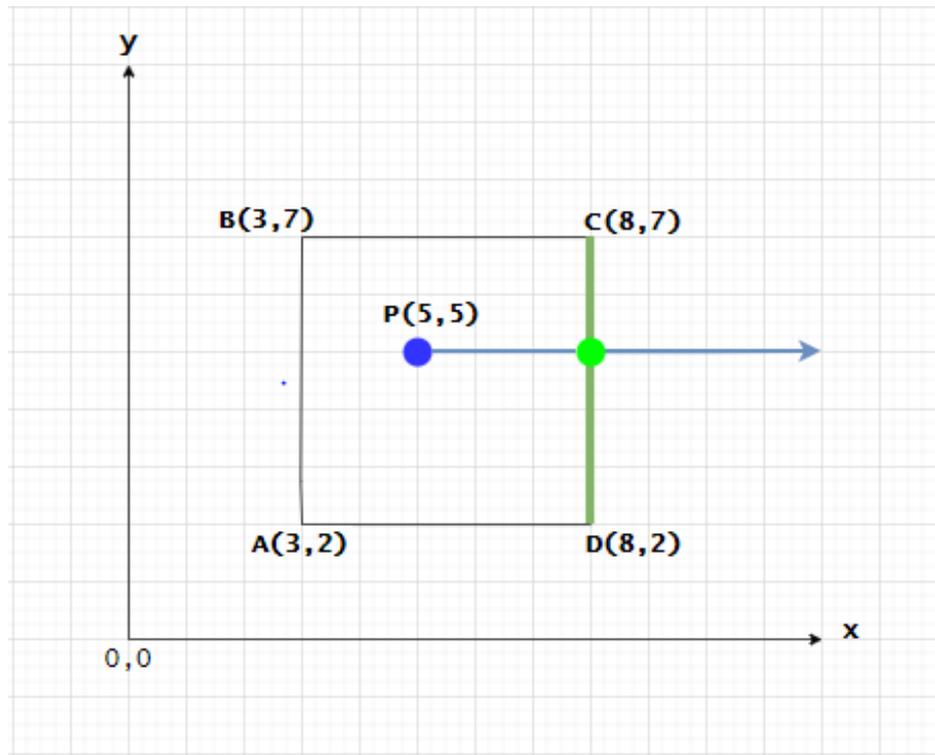
2.9. Algoritma Ray-Casting

Algoritma Ray-Casting merupakan algoritma yang dapat digunakan untuk memeriksa suatu titik berada di dalam atau di luar area yang berbentuk poligon. Poligon adalah sebuah bidang datar yang dibentuk dari garis lurus yang bergabung dan membentuk sirkuit atau rantai tertutup.

Menurut Website Geeksforgeeks (2022), untuk memeriksa suatu titik berada di dalam atau di luar poligon sesuai dengan algoritma Ray-Casting, maka secara sederhana dilakukan sebagai berikut:

1. Menggambar sebuah garis horizontal ke arah kanan dari titik yang diperiksa hingga melewati area poligon dan diperpanjang sampai tak terhingga.
2. Menghitung jumlah titik potong yang diciptakan dari perpotongan antara garis tersebut dengan garis poligon.
3. Jika jumlah titik potong menghasilkan bilangan ganjil, maka dapat dinyatakan bahwa titik yang diperiksa berada di dalam poligon.

Sedangkan, jika jumlah titik potong menghasilkan bilangan genap, maka dapat dinyatakan bahwa titik yang diperiksa berada di luar poligon.



Gambar 2.5. Ilustrasi Pemeriksaan Suatu Titik di Dalam atau di Luar Bidang Poligon dengan Menggunakan Algoritma *Ray Casting*.

Berdasarkan **Gambar 2.5** tersebut, untuk menentukan apakah titik P berada di dalam poligon, maka terdapat dua kondisi yang harus terpenuhi, yakni:

1. Berorientasi pada sumbu y, ordinat titik P berada di antara ordinat titik C dan ordinat titik D, seperti yang ditunjukkan pada Persamaan 2.1 berikut:

$$Py < Cy) \neq (Py < Dy) \quad (2.1)$$

2. Berorientasi pada sumbu x, absis pada titik P berada disebelah kiri absis titik intersection, seperti yang ditunjukkan pada Persamaan 2.2 berikut:

$$Px < Intersection \quad (2.2)$$

Adapun rumus *intersection* ditunjukkan pada Persamaan 2.3 berikut ini:

$$x = \frac{(x_2 - x_1)(y - y_1)}{(y_2 - y_1)} + x_1 \quad (2.3)$$

Sehingga, jika mengacu pada **Gambar 2.5**, maka rumus *intersection* tersebut ditunjukkan pada Persamaan 2.4 berikut ini:

$$Intersection = \frac{(Dx - Cx)(Py - Cy)}{Dy - Cy} + Cx \quad (2.4)$$

Berdasarkan **Gambar 2.5** dan mengacu pada Persamaan 2.1, Persamaan 2.2, Persamaan 2.3 dan Persamaan 2.4, maka algoritma Ray-Casting ditunjukkan pada **Kode Program 2.3** berikut ini.

Kode Program 2.3. Algoritma Ray-Casting untuk Memeriksa Suatu Titik Koordinat Berada di Dalam atau di Luar Poligon.

```

1  polygon = [[3 , 2], [3 , 7], [8, 7], [8,2]]
2  num = polygon.length
3  isInside = False
4  for (i = 0, j = num - 1; i < num; j = i) do
5  x1 = polygon[i][0]
6  y1 = polygon[i][1]
7  x2 = polygon[j][0]
8  y2 = polygon[j][1]
9  Intersection = ((x2 - x1) (y - y1)/y2 - y1) + x1
10 if ((y < y1) != (y < y2)) AND (x < Intersection) then
11 isInside = not(isInside)

```

2.10. *Exchangable Image File Format Header (EXIF Header)*

Exchangeable Image File Format (EXIF) merupakan spesifikasi industri yang diterbitkan untuk format *file* gambar yang digunakan oleh kamera digital. *Exchangeable Image File Format (EXIF) Headers* merupakan suatu tempat penyimpanan informasi yang menyertai sebuah *file* foto yang dapat dibaca oleh perangkat lunak manajemen foto. Informasi yang dapat disimpan diantaranya waktu dan lokasi ketika foto itu diambil, penggunaan lampu flash, orientasi dari kamera (*landscape* atau *portrait*), dan detail kamera lainnya. Penambahan informasi lokasi pada *file* foto dikenal dengan *geotagging*.

Geotagging adalah sebuah proses penambahan informasi posisi atau lokasi yang didapatkan dari *GPS* yang terdiri atas data *latitude* dan *longitude* dalam sebuah foto digital. Dengan adanya fitur *geotagging* dalam informasi sebuah foto maka lokasi pengambilan foto tersebut dapat dengan mudah diketahui. Terdapat tiga metode yang dapat dilakukan untuk melakukan *geotagging* pada suatu media, antara lain:

1. *Geocoding manual*, merupakan metode dimana informasi mengenai lokasi ditambahkan secara manual dengan menginputkan koordinat tertentu atau memilih lokasi pada saat melakukan *upload* suatu media ke dalam internet. Tingkat akurasi dari metode *geotagging* ini tergantung pada *tools* yang digunakan ataupun *GPS receiver* untuk mendapatkan koordinat.
2. Kamera digital dengan fitur *GPS*, yang mana pada beberapa kamera digital telah dilengkapi dengan fitur *GPS* secara langsung, sehingga hasil foto maupun video akan secara otomatis memiliki informasi lokasi pada *EXIF Header*.
3. Kamera digital dengan *GPS* terpisah, yang mana pada kamera digital yang belum dilengkapi dengan fitur *GPS*, *geotagging* dilakukan dengan sinkronisasi kamera dan *GPS* melalui sebuah *software*. Koordinat yang terekam oleh *GPS* pada waktu yang bersamaan dengan waktu pengambilan foto atau video akan ditambahkan sebagai informasi lokasi pada *EXIF Header* foto.

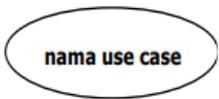
2.11. Perancangan Perangkat Lunak

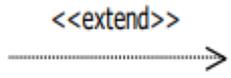
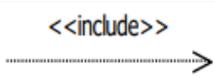
Perancangan perangkat lunak adalah pemodelan visual untuk membuat spesifikasi, menggambarkan, membangun dan menuliskan dokumentasi dari sistem perangkat lunak. Perancangan perangkat lunak yang akan dilakukan pada penelitian ini terdielei atas perancangan arsitektur aplikasi, serta perancangan fungsional sistem dan perancangan skenario aplikasi.

Menurut UPT Perpustakaan dan- Undip Press (2022), perancangan arsitektur sistem menghasilkan gambaran atau deskripsi terinci tentang struktur sebuah sistem informasi. Deskripsi ini bergantung pada karakteristik sistem yang dibangun, namun secara umum biasanya ada semacam kerangka besar yang bersifat generik untuk dijadikan panduan. Di dalam kerangka deskripsi ini terdapat penjelasan tentang aspek-aspek fundamental dari sebuah sistem dan panduan atau rekomendasi cara pengembangan atau membangun sistem tersebut.

Adapun perancangan fungsional sistem dilakukan dengan membuat diagram *Use Case* sedangkan perancangan skenario aplikasi dilakukan dengan membuat tabel-tabel yang menjelaskan urutan aktivitas dari aktor atau pengguna sistem pada setiap *Use Case*. Diagram *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dengan kata lain, diagram *Use Case* digunakan untuk mengetahui fungsi-fungsi apa saja yang terdapat di dalam sistem dan siapa saja yang berhak mengakses fungsi tersebut. Berikut ini simbol-simbol notasi pada diagram *Use Case* yang dapat dilihat pada **Tabel 2.1** berikut ini.

Tabel 2.1. Deskripsi Notasi Diagram *Use Case*.

No	Gambar	Nama	Keterangan
1		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi aktor.

No	Gambar	Nama	Keterangan
2		Aktor	Menspesifikasikan sekumpulan peran yang pengguna mainkan ketika berinteraksi dengan <i>Use Case</i>
3		<i>Extend</i>	Menspesifikasikan bahwa <i>Use Case</i> target memperluas perilaku dari <i>Use Case</i> sumber pada suatu titik yang diberikan
4		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya
5		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
6		<i>Include</i>	Menspesifikasikan bahwa <i>Use Case</i> sumber secara eksplisit.

2.12. Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan metode pengujian untuk menentukan kualitas dari perangkat lunak apakah sesuai dengan kebutuhan yang ditentukan (Subhiyakto dan Utomo, 2016). Pengujian perangkat lunak merupakan sebuah elemen kritis dari penjaminan mutu perangkat lunak (*software quality assurance*) dan merepresentasikan *review* secara keseluruhan terhadap hasil-hasil yang diperoleh dari analisis kebutuhan, perancangan dan implementasi perangkat lunak (Kurniawan, 2007). Dengan pengujian perangkat lunak diharapkan dapat meminimalisir kesalahan (*error*) pada sebuah *software* dan sebagai pengukuran kualitas dari software tersebut (Pratama dkk., 2020).

Adapun pengujian yang digunakan dalam penelitian ini adalah Pengujian Metode Validasi Lokasi. Pengujian metode adalah suatu proses untuk menguji apakah metode yang dipilih mampu menyelesaikan masalah suatu penelitian.

Adapun dalam penelitian ini, metode yang akan diuji adalah metode validasi lokasi pengunjung tempat wisata. Pengujian metode validasi lokasi berarti proses membuktikan bahwa dengan metode yang dipilih, yakni algoritma *Ray Casting*, *Location Based Services*, dan penggunaan *EXIF Header* pada aplikasi yang dibangun akan dapat merekam dan memeriksa bahwa data titik lokasi pengunjung benar berada di dalam area tempat wisata sehingga informasi jejak digital pengunjung tempat wisata tersebut dapat disimpan dan ditampilkan pada aplikasi. Namun, jika titik koordinat lokasi pada informasi jejak digital yang diunggah pada aplikasi tidak berada pada area tempat wisata, maka informasi tersebut dianggap tidak *valid* dan tidak dapat diunggah di aplikasi.

2.13. Kajian Terkait

Adapun penelitian-penelitian yang pernah dilakukan sebelumnya dengan menggunakan *Location Based Services* dan *geotagging* adalah sebagai berikut:

1. Aplikasi *Location Based Service (LBS)* Objek Wisata Tanggamus Berbasis Android oleh Hartanto dan Yuniarthe (2020), yang menghasilkan sistem informasi yang mampu memberikan detail lokasi wisata dan dapat memperkenalkan lokasi-lokasi pariwisata yang ada di Kabupaten Tanggamus.
2. Pembangunan Aplikasi Social *Geotagging* Destinasi Wisata Berbasis Android oleh Defitiria dkk. (2018), yang menghasilkan aplikasi berbasis Android yang menggunakan fitur *geotagging* untuk mendapatkan lokasi dari foto wisata dan rumus *haversine* untuk pencarian lokasi terdekat. Aplikasi yang dihasilkan berfungsi dengan baik.
3. Rekomendasi Tempat Wisata Di Kabupaten Grobogan oleh Perbawa dkk. (2021), yang menghasilkan sistem informasi wisata di kabupaten Grobogan berbasis *web* dan dapat merekomendasikan tempat wisata di Grobogan berdasarkan radius terdekat. Berdasarkan uji validitas antara aplikasi dengan Google Maps, maka didapatkan selisih sebesar 0,221 km, sehingga bisa dikatakan cukup *valid*, sedangkan uji validitas antara aplikasi dengan pengujian riil di lapangan bisa dikatakan cukup *valid* dengan rata-rata selisih sebesar 0,358 km.