

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Teori Dasar**

Teori dasar ialah bagian dari pustaka yang akan digunakan dalam perancangan aplikasi yang memuat tentang teori- teori dasar yang memiliki fungsi sebagai kerangka utama dalam perancangan aplikasi ini. Teori dasar akan menjelaskan bagian dari kerangka yang akan digunakan dalam menyusun aplikasi ini, dengan tujuan agar setiap kerangka yang digunakan memiliki fungsi dan penggunaan yang benar. Adapun teori dasar dalam penelitian ini yaitu:

##### **2.1.1 Aplikasi**

Aplikasi adalah perangkat lunak untuk mengolah data yang memiliki aturan dan ketentuan bahasa *pemrograman* yang dapat membantu dan menyelesaikan tugas-tugas yang diberikan oleh user (KBBI, 2015). Menurut sumber lain, aplikasi merupakan alat bantu yang dapat mempermudah dan mempercepat proses penyelesaian, dan tidak menjadi beban bagi user (Fathurrohman & Ardhiansyah, 2020). Berdasarkan pendapat diatas maka penulis mendefinisikan pengertian dari aplikasi adalah perangkat lunak yang dapat membantu untuk mempermudah dan mempercepat penyelesaian tugas dari pengguna yang memiliki ketentuan bahasa pemrograman.

Menurut pendapat (Fathurrohman & Ardhiansyah, 2020) aplikasi dapat dikategorikan menjadi 3 kelompok, yaitu:

- a) Aplikasi *Desktop*, yaitu aplikasi yang hanya dijalankan di komputer atau laptop.
- b) Aplikasi *web*, yaitu aplikasi yang dijalankan di komputer dan terkoneksi dengan *internet*.
- c) Aplikasi *Mobile*, yaitu aplikasi yang dijalankan diperangkat *mobile*, dan aplikasi ini yang paling banyak digunakan.

Berdasarkan fungsinya, aplikasi dapat dikelompokkan menjadi 5 kelompok (Fathurrohman & Ardhiansyah, 2020) , yaitu:

- 1) *System Software*, yaitu aplikasi yang berfungsi untuk mengelola proses internal dalam komputer.
- 2) *Real Time Software*, yaitu aplikasi yang berfungsi untuk untuk mengamati kejadian di lokasi nyata secara langsung.
- 3) *Engineering and Scientific Software*, yaitu aplikasi yang berfungsi untuk menyelesaikan masalah yang sifatnya nonalgoritmik.
- 4) *Web Based Software*, yaitu aplikasi yang berfungsi untuk menghubungkan pengguna dengan internet secara langsung.
- 5) *Personal Computer Software*, yaitu aplikasi yang mengelola *input* menjadi *output* dan digunakan pada perangkat komputer secara resmi dan pribadi.

### **2.1.2 Pemandu**

Pemandu atau *guide* adalah orang yang menemani, membimbing, menunjukkan dan memberikan pengetahuan tentang perjalanan dan objek wisata yang sedang dikunjungi oleh wisatawan (Habibullah, Mulyanto, & Sofya, 2020).

Dalam hal ini pemandu yang akan dibangun yaitu berbasis aplikasi, dimana pemandu yang dimaksud bukanlah orang yang akan memberikan informasi atau pelayanan selama berwisata, tetapi pemandu yang dimaksud yaitu aplikasi yang akan menyediakan kebutuhan dan petunjuk tentang objek wisata dan akan memberikan rute untuk mengakses objek wisata.

### **2.1.3 Wisata**

Pariwisata merupakan kegiatan berwisata yang dilengkapi dengan fasilitas serta layanan yang disediakan oleh masyarakat setempat, pemerintah, dan pengusaha (UU, 2009). Pariwisata ialah perjalanan yang dilakukan dengan waktu yang singkat ketempat lain, yang sifatnya bukan untuk tinggal disana dan bukan untuk bekerja, tetapi untuk tujuan berkunjung, refreshing, dan untuk liburan. (Habibullah et al., 2020). Wisata adalah kegiatan perjalanan yang dilakukan seseorang keluar rumah yang sifatnya untuk berkunjung dan menikmati kegiatannya dengan mengeluarkan uang untuk menikmati dan memenuhi keinginannya, (Smitch, 2014). Kegiatan wisata biasanya dilakukan untuk melihat, menikmati suatu tempat atau ingin menyaksikan suatu pertunjukan dan bahkan untuk menikmati kuliner suatu daerah. Namun berwisata biasanya dilakukan untuk menikmati alam yang unik atau jarang di saksikan dalam kehidupan sehari-hari. Setiap daerah biasanya mempunyai tempat tertentu yang biasanya dikunjungi oleh wisatawan karena adanya daya tarik

tertentu dari daerah tersebut. Salah satunya Danau Toba, yang memiliki banyak objek wisata yang layak dikunjungi saat liburan. Tempat wisata yang sering dikunjungi wisatawan yang di Danau Toba yaitu:

#### 1. Wisata Budaya

Wisata budaya ialah kegiatan wisata yang bertujuan untuk melihat, mempelajari dan menikmati kebudayaan yang ada ditempat wisata. Danau Toba memiliki banyak objek wisata budaya yang menarik perhatian setiap wisatawan yang berkunjung. Pertunjukan kebudayaan di Danau Toba dilakukan untuk memperkenalkan dan melestarikan budaya Batak Toba, agar orang muda yang berasal dari Danau Toba tidak melupakan kebudayaan yang ada di daerah tersebut. Tempat wisata budaya yang ada di Danau Toba yaitu berada di Sianjur Mula-Mula, di Tomok, dan salah satu Museum Batak atau sering disebut museum TB Silalahi , dan masih banyak lagi daerah di danau toba yang memiliki tempat wisata budaya.





**Gambar 2. 1** Wisata Budaya Batak

Sumber: (<https://www.visitsamosir.com>)

## 2. Wisata Alam

Wisata Alam merupakan perjalanan wisata yang dilakukan untuk menikmati alam sekitar yang bukan ciptaan manusia, dan bertujuan untuk menjelajahi, mengagumi, menikmati keaneka ragaman flora dan fauna, maupun jenis alamiah lainnya. Kawasan danau Toba mempunyai banyak sekali wisata alam yang menjadi daya tarik tersendiri bagi setiap orang yang melihatnya. Wisata alam di Danau Toba sangat unik, dan jarang kita jumpai di Dunia, contohnya Danau diatas Danau, Batu

Gantung, Permandian Air Panas, Bukit Indah Simarjarunjung, Pulau Tao, dan masih banyak lagi wisata alam yang memang benar-benar alamiah sekali.





**Gambar 2. 2** Wisata Alam di Danau Toba

Sumber:(<https://www.visitsamosir.com>)

### 3. Wisata Kuliner

Wisata Kuliner merupakan kegiatan berwisata atau berkunjung untuk menikmati masakan khas dari suatu daerah yang dikunjungi. Kuliner dari setiap daerah pasti akan berpengaruh terhadap tumbuhnya daya tarik untuk mengunjungi tempat tersebut. Kuliner dari setiap daerah juga penyumbang pendapatan ekonomi masyarakat di daerah tersebut. Danau Toba memiliki masakan khas tersendiri, yang menjadi cirikhas dan daya tarik bagi wisatawan agar berwisata kesana. Masakan khas yang disajikan di Samosir diolah secara alamiah serta menggunakan bumbu alami yang tidak mengandung bahan pengawet, serta cita rasa yang khas dari daerah Batak Toba. Wisata kuliner yang dapat kita jumpai di Samosir yaitu Naniura, Natinombur, Arsik, Dali ni Horbo, Napinadar, Ombus-ombus, Mie Gomak, Pohul-pohul, dan jenis makanan nasional lainnya.



**Gambar 2. 3** Jenis Makanan khas dari Danau Toba

Sumber: (<https://www.visitsamosir.com>)

#### 2.1.4 Web

Website merupakan halaman halaman situs internet yang berisi file, gambar, video yang sifatnya digital dan diletakkan di host yang dapat diakses melalui internet, (Irfan & Ebi, 2018). Menurut (Ahmad & Baharom, 2012) web adalah aplikasi yang berisi dokumen multimedia, yang menerapkan protocol HTTP (Hypertext Transfer Protocol) yang dapat diakses melalui perangkat lunak yang terhubung dengan internet.

Website terdiri dari 3 macam, yaitu:

- 1) *Website Statis*, website yang sifatnya menetap, dan perubahan hanya bisa dilakukan dengan manual. Contohnya: *Website* perusahaan.
- 2) *Website Dinamis*, Website yang strukturnya dapat di update, dan kode *website* dapat di edit. Contohnya: Situs *E-Commerce*.

- 3) *Website Interaktif*, *Website* yang menyediakan layanan untuk berinteraksi dengan orang lain secara online. Contohnya: situs media sosial, seperti *Facebook*, *Whatsap*.

## **2.2 Teori Khusus**

### **2.2.1 Rancang Bangun**

Rancang adalah tahapan atau langkah-langkah untuk menuangkan hasil analisa untuk mendeskripsikan secara detail bagaimana suatu sistem itu dapat diimplementasikan. Bangun adalah kegiatan untuk membuat sesuatu yang baru atau memrevisi sistem yang sudah ada melalui analisa dan perancangan secara keseluruhan. Rancang bangun adalah langkah-langkah untuk membuat sesuatu yang baru atau merevisi sistem yang sudah ada berdasarkan perancangan secara lengkap (Pressman, 2015). Menurut pendapat lain, rancang bangun merupakan proses untuk menyusun dan membuat struktur suatu aplikasi yang belum ada pada suatu organisasi atau objek tertentu (Dendi & Endang, 2018).

Kesimpulan yang didapat dari pengertian diatas bahwa Rancang bangun adalah tahapan atau struktur yang dibuat untuk menciptakan suatu sistem yang baru atau memperbaiki sistem yang sudah ada melalui metode perancangan dan analisa secara keseluruhan dengan tujuan untuk memecahkan permasalahan yang ada.

### 2.2.2 *Agile*

Menurut (Irfan & Ebi, 2018) *Agile* merupakan metode pengembangan perangkat lunak dilakukan secara cepat, terupdate, terstruktur karena setiap tahapannya diselesaikan berdasarkan permasalahan yang dijumpai dan diselesaikan dengan benar. Metode *Agile* berfokus pada kepuasan pelanggan, interaksi pengguna dengan pengembang, dan memberikan layanan terbaru kepada pengguna. (Irfan & Ebi, 2018) Mengemukakan prinsip utama dari metode *Agile* yaitu:

1. Mengutamakan kepuasan pengguna, dengan menghasilkan produk lebih cepat dan berkelanjutan.
2. Menerima masukan dari pengguna.
3. Dapat menyelesaikan software dengan baik.
4. Rekan bisnis dan developer harus bekerjasama dengan baik selama proyek IT berlangsung.
5. Mengembangkan proyek pada orang-orang yang termotivasi.
6. Dapat menyesuaikan dengan lingkungan kebutuhan pengguna.
7. Komunikasi secara langsung antara pengguna dengan pengembang, yang menghasilkan interaksi yang efektif dan efisien.
8. Keunggulan teknis merupakan hal utama untuk meningkatkan *agility*.
9. Metode ini bersifat pengembangan berkelanjutan karena disesuaikan dengan kebutuhan pengguna.
10. Bersifat sederhana dan mengoptimalkan sumber daya yang ada.
11. Tahapan dan struktur *software* terbaik akan muncul dari tim yang mampu mengorganisir diri sendiri.

12. Para perancang senantiasa berupaya menyelaraskan cara kerja yang efektif dengan pola kerja mereka.

Menurut (Irfan & Ebi, 2018) metode *Agile* terdiri dari 4 jenis, yaitu :

1. *Extreme Programming (XP)*

Dalam metode ini terdapat beberapa fase yang dikembangkan yaitu *planning, design, coding, dan testing*. Keunggulan metode ini yaitu lebih meningkatkan kepuasan pelanggan dan pengembang proyek IT. *Feedback* akan cepat di tangani, dan apabila ada kekurangan akan segera ditangani.

2. *Dynamic System Developmet Method (DSSM)*

Dalam metode ini terdapat 5 fase yang dikembangkan yaitu *feasibility study, functional model iteration, design and build iteration, dan implementation phase*. Keunggulan dari metode ini yaitu dapat membangun software lebih cepat.

3. *Scrum*

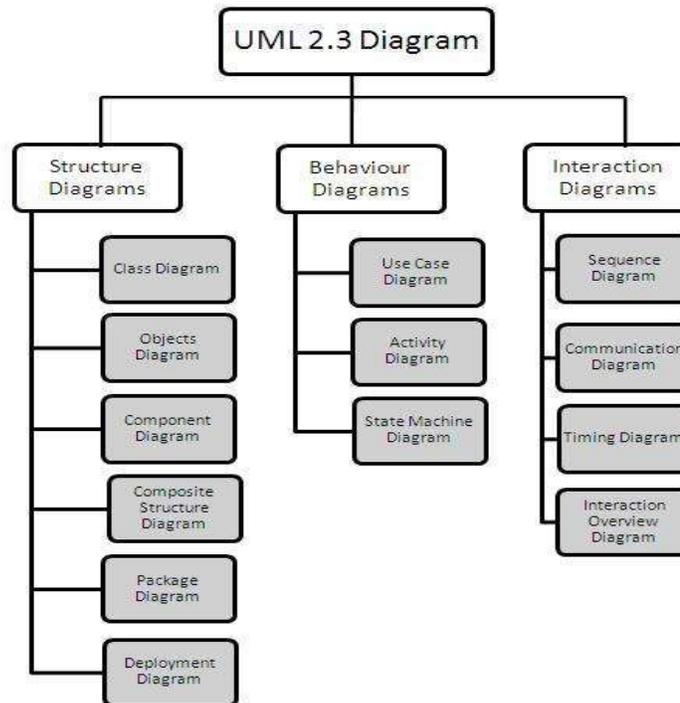
Metode ini didasarkan pada kolaborasi tim, *incremental product* dan proses iterasi. Kelebihan dari metode ini yaitu, membantu efisiensi perusahaan dan biaya *overhead* yang minim, dan metode ini dapat bekerja pada berbagai jenis teknologi dan bahasa pemrograman untuk meningkatkan sebuah website.

#### 4. *Adaptive Software Development (ASD)*

Aktifitas utama dalam metode ini yaitu *speculation*, *collaboration*, dan *learning*. Metode ini dapat meningkatkan kerja tim dan kualitas perangkat lunak, serta menghemat biaya produksi dan mengurangi resiko kegagalan implementasi *software*.

#### 2.2.3 *UML*

UML(*Unified Modeling Language*) adalah standar bahasa pemodelan dalam mengembangkan perangkat lunak yang menggunakan teknik pemrograman berbasis objek (A.S, Rosa. & Shalahuddin, 2018). UML dibuat agar dapat memenuhi kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari perangkat lunak atau sistem yang akan dibangun yang memakai diagram dan teks pendukung.



**Gambar 2. 4** Struktur UML2.3

Sumber: (docplayer.info)

### 2.2.3.1 *Class Diagram*

Menurut (A.S, Rosa. & Shalahuddin, 2018) *Class Diagram* atau kelas diagram memberikan penjelasan atau makna dari setiap kelas yang telah ditentukan untuk mempermudah dalam menyusun suatu sistem. Kelas memiliki dua komponen yaitu atribut dan operasi atau metode.

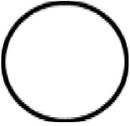
1. Atribut ialah variable yang ada didalam sebuah kelas yang telah dibuat.
2. Operasi atau metode ialah penjelasan atau fungsi dari suatu kelas yang telah dibuat.

Diagram kelas dibuat untuk mempermudah pembuat program dalam menentukan setiap bagian atau kelas agar sistem yang akan dibangun sesuai dengan rencana pembuatan sistem yang telah dirancang terlebih dahulu. Tahapan penyusunan kelas pada sebuah sistem yang akan digambarkan sangat mempengaruhi alur dari perancangan sistem. Tahapan penyusunan yang baik dan benar dalam membangun sebuah sistem yaitu:

- 1) *Class Main* (kelas yang dijalankan diawal saat sistem dioperasikan).
- 2) *View* (kelas yang mengatur tampilan sistem).
- 3) *Controller* (kelas yang akan menangani proses bisnis, dan definisi yang harus diambil dari penjabaran *use case*).
- 4) *Model* (kelas yang menyimpulkan dan memberikan penjelasan dari setiap kelas agar dapat disimpan kedalam basis data).

Menyusun setiap kelas dalam sebuah sistem yang akan dibangun tentunya harus memiliki metode dan prinsip yang baik dan tepat, supaya sistem yang dibangun dapat digunakan dengan benar. Menyusun kelas dalam perancangan sistem harus memperhatikan metode *cohesion* dan *coupli*. *Cohesion* merupakan hubungan instruksi dalam sebuah metode yang terkait satu sama lain. *Cuopling* merupakan hubungan instruksi dalam metode yang digunakan dalam suatu kelas. Penyusunan kelas yang tepat dan benar, harus memiliki metode dimana kadar *cohesion* hars lebih kuat dibandingkan dengan kadar *coupling*.

**Tabel 2. 1** Lambang dalam *Class Diagram*

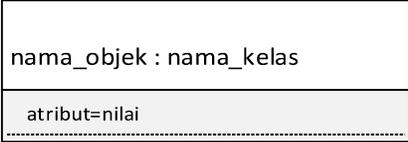
Lambang	Pemaparan
<p>Kelas</p> 	Kelas dalam suatu sistem
<p>antarmuka / <i>interface</i></p>  <p>nama_interface</p>	konsep <i>interface</i> dalam sisten yang menerapkan objek
<p>asosiasi / <i>association</i></p> 	Hubungan antar kelas yang bersifat umum
<p>asosiasi berarah / <i>directed association</i></p> 	Hubungan antar kelas yang berada dikelas lain
<p>Generalisasi</p> 	Hubungan antar kelas yang sifatnya umum ke khusus
<p>kebergantungan / <i>dependency</i></p> 	Hubungan antar kelas yang saling membutuhkan
<p>agregasi / <i>aggregation</i></p> 	Hubungan antarkelas dengan sistem seluruh bagian ( <i>whole-part</i> )

Sumber: (Data olahan penulis, 2021)

### 2.2.3.2 Object Diagram

Menurut (A.S, Rosa. & Shalahuddin, 2018) Diagram objek memberikan gambaran dari sistem yang akan dibangun, baik dalam memberikan nama objek dan proses yang dilalui objek tersebut. Diagram objek akan memberikan penjelasan dari isi setiap atribut yang ada dalam tiap kelas. Hubungan *link* dalam diagram objek merupakan sistem yang akan menghubungkan apabila ditemukan objek yang digunakan oleh objek lain.

**Tabel 2. 2** Lambang dalam *Objek Diagram*

Lambang	Pemaparan
<p>Objek</p> 	Objek yang berjalan dalam kelas ketika sistem sedang aktif
<p>Link</p> 	Hubungan sesama objek

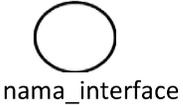
Sumber: (Data olahan penulis, 2021)

### 2.2.3.3 Component Diagram

Diagram komponen akan menunjukkan bagaimana hubungan antara komponen yang terdapat pada suatu sistem yang akan dibangun, serta akan berfokus pada komponen sistem yang akan digunakan yang memiliki peranan penting selama penyusunan sistem tersebut (A.S, Rosa. & Shalahuddin, 2018). Bagian utama yang ditemukan dalam suatu sistem ialah:

- 1) Komponen *user interface*, yaitu bagian yang mengatur tampilan sistem yang dibangun.
- 2) Komponen *business processing*, komponen yang akan mengatur fungsi bisnis.
- 3) Komponen data, komponen yang akan memanipulasi data.
- 4) Komponen *security*, komponen yang bertanggung jawab atas keamanan sistem.

**Tabel 2. 3** Lambang dalam *Component Diagram*

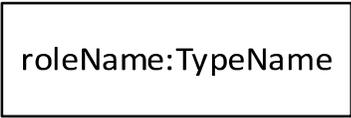
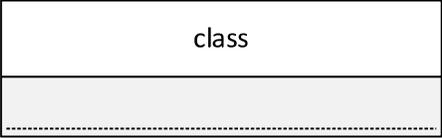
Lambang	Pemaparan
Package	Package ialah sampul dari beberapa komponen
Komponen	Bagian dari suatu sistem
Kebergantungan / <i>dependency</i> 	Menunjukkan hubungan komponen dalam satu sistem
Antarmuka / <i>interface</i> 	Sebagai antarmuka komponen untuk mewakili komponen lain agar tidak mengakses langsung komponen
<i>Link</i> 	Hubungan sesama komponen

Sumber: (Data olahan penulis, 2021)

### 2.2.3.4 Composite Structure Diagram

*Composite Structure Diagram* mulai diterapkan pada UML versi 2.0, yang berfungsi memberikan gambaran desain dari bagian yang saling terhubung dan menjelaskan desain tersebut ketika sistem dijalankan dari *instance* yang saling terhubung.

**Tabel 2. 4** Lambang dalam *Composite Structure Diagram*

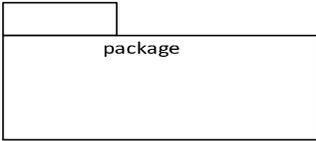
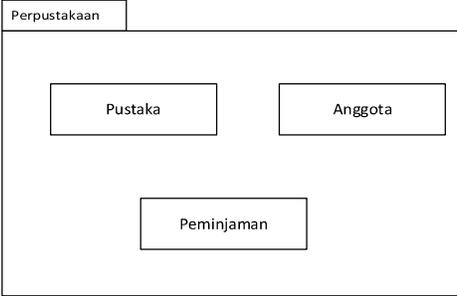
Lambang	Pemaparan
Property 	roleName: indikasi/ petunjuk sebuah <i>property</i> Typename: menunjukkan tipe kelas yang dimiliki <i>Property</i>
Connector 	<i>Connector</i> : media koneksi yang digunakan buah <i>instance</i> .
Port 	Port: media yang digunakan untuk mewakili data internal dalam sebuah sistem.
Class 	Kelas: bagian yang akan dijelaskan strukturnya dalam sebuah sistem

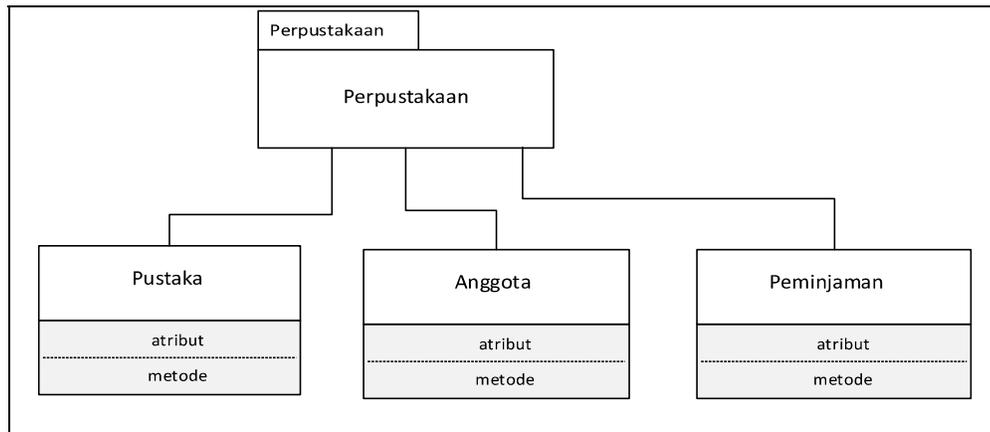
Sumber: (Data olahan penulis, 2021)

### 2.2.3.5 Package Diagram

*Package Diagram* memiliki peranan sebagai penyedia langkah untuk menyatukan elemen- elemen yang memiliki keterkaitan dalam diagram UML. Setiap elemen yang ada dalam UML yang memiliki relasi dan keterkaitan antara satu dan lainnya akan dapat dikelompokkan menggunakan *package diagram*.

**Tabel 2. 5** Lambang dalam *Package Diagram*

Lambang	Pemaparan
	<p><i>Package</i>: sampul yang digunakan untuk mewakili elemen dalam UML.</p>
<p>Bagian dalam <i>package</i> yang disimpulkan dalam sebuah <i>package</i></p> 	
<p>Bagian <i>package</i> yang dikelompokkan dalam suatu sampul <i>package</i></p>	



Sumber: (Data olahan penulis, 2021)

### 2.2.3.6 Deployment Diagram

*Deployment Diagram* berperan sebagai penunjuk pengkonfigurasi komponen pada saat aplikasi dijalankan. *Deployment Diagram* berfungsi untuk memodelkan sistem penambahan (*embedded system*) yang mewakili rancangan (*device, node, dan hardware*), bentuk *client/server*, metode terdistribusi murni, dan rekayasa ulang aplikasi (A.S, Rosa. & Shalahuddin, 2018).

**Tabel 2. 6** Lambang dalam *Deployment Diagram*

Lambang	Pemaparan
Package	Sampul yang digunakan untuk membungkus <i>node</i>
<i>Node</i>	Gambaran dari perangkat yang digunakan dan tidak dibuat sendiri, dan apabila memiliki komponen pendukung maka harus sesuai dengan rancangan awal
Kebergantungan/ <i>Dependency</i>	Hubungan antar <i>node</i> yang menunjukkan tahapan berikutnya

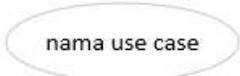
	
<i>Link</i> 	Sebagai penghubung dalam <i>node</i> yang sedang digunakan

Sumber: (Data olahan penulis, 2021)

### 2.2.3.7 Use Case Diagram

*Use Case Diagram* berfungsi dalam memberikan keterangan atau deskripsi dari interaksi diantara beberapa pelaku dengan sistem informasi yang akan dirancang (A.S, Rosa. & Shalahuddin, 2018). *Use Case Diagram* juga akan memberikan keterangan dari kegunaan yang ada dalam suatu sistem serta orang yang dapat mengakses *tools* tersebut. *Use Case Diagram* akan mempertimbangkan penamaan setiap kelas, apakah sudah memenuhi syarat dari pendefinisian kelas yang baik.

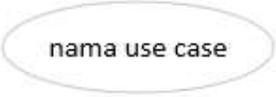
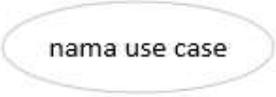
**Tabel 2. 7** Lambang dalam *Use Case Diagram*

Lambang	Pemaparan
<i>Use case</i> 	Sarana yang disediakan oleh sistem untuk mendukung interaksi unit dan aktor
Aktor / <i>actor</i> 	Pelaku, satuan, dan komponen lain dalam sistem memiliki bagian fungsi tertentu
Asosiasi / <i>association</i> 	Interaksi yang dilakukan oleh aktor dengan <i>use case</i>
Ekstensi / <i>extend</i>	Relasi yang ditambahkan untuk mendukung kelengkapan fungsi <i>Use</i>

	<i>case</i> , yang umumnya dapat berdiri sendiri
Generalisasi / <i>generalization</i> 	Hubungan yang dimiliki dua <i>use case</i> yang sifatnya tergolong umum atau khusus
Menggunakan / <i>include</i> / <i>uses</i> 	Relasi yang ditambahkan yang memiliki karakter terikat dengan <i>use case</i> agar dapat menjalankan fungsinya

Sumber: (Data olahan penulis, 2021)

**Tabel 2. 8** Hubungan *use case diagram*

Penghubung	Keterangan
Ekstensi / <i>extend</i> 	Komponen yang dapat digunakan untuk mewakili beberapa metode menjadi sebuah <i>use case</i> induk
Generalisasi / <i>generalization</i> 	<i>Use case</i> umum dapat digunakan dalam perwakilan untuk membuat sebuah <i>use case</i> dengan prinsip <i>use case</i> khusus sebagai pendukung
<i>use case</i> yang berdiri sendiri 	Prinsip yang diterapkan dalam suatu proses dengan menyelesaikannya dengan diri sendiri
Nilai <i>use case</i> yang berdiri sendiri 	Kelas yang fungsinya kurang maksimal karena hanya menggunakan satu metode

Sumber: (Data olahan penulis, 2021)

### 2.2.3.8 Activity Diagram

*Activity Diagram* digunakan untuk memberikan gambaran atau tahapan dari setiap proses yang dijalankan, dan tidak menggambarkan aktivitas dari aktor, tetapi berfokus pada kegiatan yang ada dalam sistem. *Activity Diagram* juga akan menjelaskan beberapa hal berikut ini:

- 1) Tahapan kegiatan aktivitas, yaitu langkah- langkah yang sedang dijalankan.
- 2) Tahapan struktur tampilan sistem yang dibuat sebagai rancangan tampilan yang terstruktur.
- 3) Urutan perancangan pengujian dalam semua aktivitas yang perlu didefinisikan berdasarkan pengujinya.
- 4) Urutan perancangan menu yang akan digunakan pada perangkat lunak.

**Tabel 2. 9** Lambang dalam *Activity Diagram*

Lambang	Pemaparan
status awal 	Kondisi pertam saat sistem akan dijalankan
aktivitas 	Menggambarkan kesibukan yang terjadi dalam sistem
percabangan / <i>decision</i> 	Asosiasi sebagai pemersatu dari beberapa aktivitas yang dijalankan
penggabungan / <i>join</i> 	Titik akhir sebelum status akhir untuk mengumpulkan setiap aktivitas agar menjadi satu.

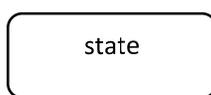
status akhir 	Tahap terakhir dari semua aktivitas komponen dalam sistem
<i>swimlane</i> 	Batasan organisasi bisnis yang memiliki peranan tanggung jawab dari aktivitas komponen.

Sumber: (Data olahan penulis, 2020)

### 2.2.3.9 State Machine Diagram

*State Machine Diagram* berfungsi menampilkan perubahan status atau *transisi* status dari suatu proses, dan perubahan itu akan dapat ditampilkan menjadi suatu graf berarah (A.S, Rosa. & Shalahuddin, 2018). *State Machine Diagram* digunakan untuk mendefinisikan interaksi yang ada pada objek, dan biasanya digunakan untuk menggambarkan tahapan proses pengguna dengan sistem yang dijalankan.

**Tabel 2. 10** Lambang dalam *State Machine Diagram*

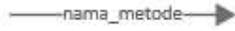
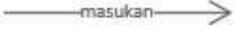
Lambang	Pemaparan
Start/ Status Awal ( <i>initial State</i> ) 	Kondisi pertama ketika sistem dimulai
End / Status akhir ( <i>Final State</i> ) 	Kondisi akhir setiap aktivitas yang dijalankan
Event 	Proses yang membuat adanya perubahan status dalam sistem
State 	Kondisi yang mewakili aktivitas yang terjadi

Sumber: (Data olahan penulis, 2021)

### 2.2.3.10 Sequence Diagram

*Sequence Diagram* akan memberikan gambaran dari kelakuan objek berdasarkan proses serta pesan yang dikirim dan diterima oleh objek satu dengan yang lain. *Sequence Diagram* terlebih dahulu harus memahami objek yang ikut serta dalam *use case* serta metode yang dijalankan setiap kelas yang dianggap menjadi suatu objek.

**Tabel 2. 11** Lambang dalam *Sequence Diagram*

Lambang	Pemaparan
Aktor 	Pelaku, satuan, dan komponen lain dalam sistem memiliki bagian fungsi tertentu
Garis hidup / <i>lifetime</i> 	Mewakili jalannya aktivitas objek
Objek 	Setiap bagian yang terlibat dalam komunikasi pesan
Waktu aktif 	Keadaan yang dibaca ketika objek dapat melakukan proses dan semua aktivitas yang terjadi dalam sebuah komponen
Pesan tipe <i>create</i> 	Menggambarkan hubungan satu komponen berdasarkan arah panah yang mewakili tujuan yang dilakukan komponen tersebut
Pesan tipe <i>call</i> 	Menggambarkan perintah yang dikirimkan ke komponen tujuan dengan skema berdasarkan arah panah
Pesan tipe <i>send</i> 	Bentuk interaksi yang dikirimkan kekomponen lain, yang berisi pesan,

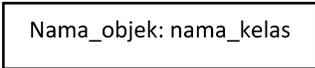
	dengan memperhatikan arah panah yang dibuat
Pesan tipe <i>return</i> 	Gambaran dari hasil pesan yang telah dikirimkan, yang berisi balasan atau hasil dari perintah yang dijalankan, yang mengarah pada komponen pengirim sebelumnya
Pesan tipe <i>destroy</i> 	Gambaran aktivitas terakhir yang dialami dalam sebuah interaksi, yang menggambarkan keadaan selesai

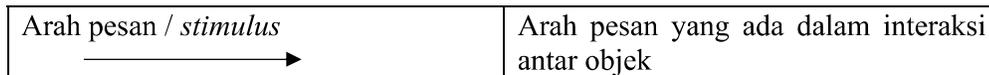
Sumber: (Data olahan penulis, 2021)

### 2.2.3.11 *Communication Diagram*

*Communication Diagram* berfungsi untuk memberikan penjelasan interaksi, berupa urutan pengiriman pesan, dan setiap pesan yang diterima dari diagram kelas, diagram sekuen, dan diagram *use case* yang akan menjelaskan susunan statis dan sifat dinamis dari suatu sistem (A.S, Rosa. & Shalahuddin, 2018). *Communication Diagram* akan menuliskan desain/alur yang digunakan oleh objek secara menyeluruh, dan biasanya dapat disalin dari diagram sekuen. Penomoran metode yang dituliskan harus berurutan berdasarkan desain/alur diantara sesama objek atapu berdasarkan objek itu sendiri.

**Tabel 2. 12** Lambang dalam *Communication Diagram*

Lambang	Pemaparan
Objek 	Interaksi yang dilakukan objek
<i>Link</i> 	Relasi antar objek yang saling berhubungan dengan objek tersebut



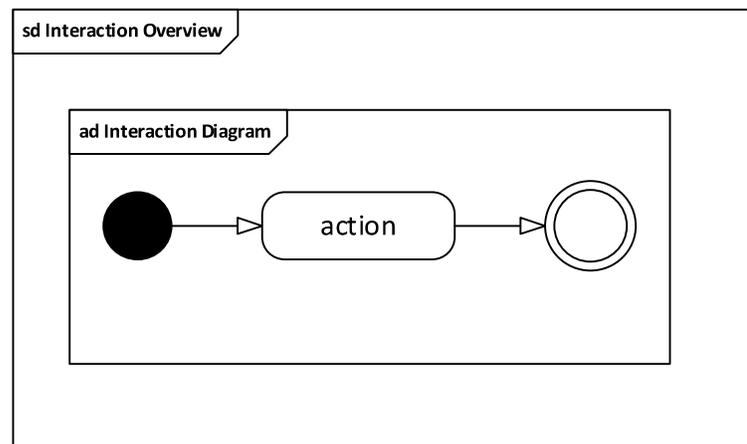
Sumber: (Data olahan penulis, 2021)

### 2.2.3.12 *Timing Diagram*

*Timing Diagram* berfungsi memberikan gambaran tingkah laku sistem dalam kurun waktu tertentu, dan menjelaskan proses yang dilalui alat *digital* karena menjelaskan dengan kata-kata akan lebih sulit dipahami dibandingkan penggambaran secara visual.

### 2.2.3.13 *Interaction Overview Diagram*

*Interaction Overview Diagram* berfungsi untuk menjelaskan setiap interaksi yang ada dalam setiap diagram yang saling terhubung. *Interaction Overview Diagram* akan melibatkan diagram sekuen, diagram komunikasi, dan timing diagram untuk mendefinisikan setiap interaksi yang terjadi pada saat adanya message diantara objek.



**Gambar 2. 5** Contoh Interaction Element

Sumber: (Data olahan penulis, 2021)

#### **2.2.4 Google Maps API**

*Google maps* ialah gambar dunia yang akan digunakan untuk melihat suatu daerah dengan menggunakan *browser*. Kita bisa menambah fitur dalam *maps* sesuai dengan keinginan dan kebutuhan kita dengan menggunakan suatu aplikasi, yang disebut dengan *Google Maps API*. *Google Maps API* adalah sebuah *library* yang berbentuk *Javascript* yang bertujuan untuk memodifikasi *Google maps* sesuai dengan kebutuhan yang memiliki keunggulan untuk mengambil gambar atau peta secara statis (Rismayani, 2016).

#### **2.2.5 Database**

*Database* merupakan kumpulan data dan informasi yang tersimpan diluar komputer, yang dapat diolah dan diperiksa secara sistematis dengan menggunakan *software* tertentu untuk memanipulasinya (Linda, 2020). Menurut pendapat lain (Kurniawan, 2016) *Database* ialah kumpulan data dan uraian data yang saling terikat secara logika, dan dibuat berdasarkan kebutuhan informasi organisasi. Kesimpulan yang didapatkan oleh penulis ialah *Database* merupakan kumpulan data dan informasi yang tersimpan dalam suatu sistem yang saling berhubungan dan dapat diolah serta dirancang untuk memenuhi kebutuhan tertentu.

(Enterprise, 2017) *MySql* merupakan contoh dari *Database* yang sering digunakan dan sangat tepat disatukan dengan bahasa pemrograman *PHP*, untuk mendukung pengolahan basis data yang diproses secara tepat dan cepat secara otomatis. *MySql* berbasis *sql* (*Structure Query Language*) yang biasanya digunakan untuk mendukung bahasa *Database*. Perintah yang sering digunakan dalam *MySql*, yaitu:

1. *Select* (Menampilkan data dari suatu tabel)

```
SELECT * FROM pelanggan WHERE id_pelanggan NOT IN('P0001');
```

2. *Insert* (Menambahkan data kedalam tabel)

Pada menu insert terdapat 7 menu, yaitu:

1. INSERT INTO (field1, field2, ...) VALUES (...)
2. INSERT ... SELECT ...
3. INSERT IGNORE ...
4. INSERT DELAYED ...
5. INSERT LOW PRIORITY | HIGH PRIORITY ...
6. INSERT ... ON DUPLICATE KEY UPDATE
7. REPLACE INTO ...

3. *Update* (Mengubah suatu data)

```
UPDATE pelanggan SET nama_pelanggan = 'Aris Web' WHERE  
id_pelanggan = 'P0001';
```

4. *Delete* (Menghapus data)

```
DELETE FROM mahasiswa  
WHERE jurusan = 'Teknik Informatika';
```



**Gambar 2.5** Logo MySQL

Sumber: (<https://www.mysql.com>)

### **2.2.6 HTML (Hypertext Markup Language)**

Menurut (Saputra, 2012) *HTML (Hypertext Markup Language)* ialah sebuah bentuk bahasa pemrograman yang mengandung tag didalamnya untuk menyatakan kode yang dapat dibaca oleh browser supaya bisa ditampilkan dengan benar. HTML berfungsi untuk membuat halaman web, menampilkan informasi didalam browser, dan membuat link menuju halaman web lain dengan kode tertentu. (Saputra, 2012)

Struktur dasar dalam *HTML* yaitu :

1. Elemen *HTML*

Elemen ini merupakan elemen dasar apabila akan memulai suatu dokumen *html*. Contoh *tag*-nya adalah: `<html></html>`

2. Elemen *Head*

Elemen ini berfungsi untuk menulis keterangan tentang dokumen web yang akan ditampilkan. Contoh penulisan elemen ini dalam suatu program yaitu:

```
<html>
```

```
<head>
```

```
</head>
```

```
</html>
```

### 3. Elemen *Title*

Elemen ini terdapat didalam elemen *head* yang digunakan untuk menuliskan judul pada *caption browser web* tentang topik dari dokumen yang akan ditampilkan. Contoh struktur penggunaannya yaitu:

```
<html>
```

```
<head>
```

```
  <title> Judul Tulisan </title>
```

```
</head>
```

```
</html>
```

### 4. Elemen *Body*

Elemen ini merupakan bagian utama dari dokumen yang kita buat, karena pada bagian ini kita akan meletakkan informasi atau dokumen yang akan ditampilkan pada *web browser*. Contoh struktur penulisannya:

```
<html>
```

```
<head>
```

```
  <title> Judul Tulisan </title>
```

```
</head>
```

```
  <body>
```

### Isi dari konten yang akan ditampilkan

`</body>`

`</html>`

Menurut (Saputra, 2012) html telah mengalami perubahan dan perkembangan seiring dengan berkembangnya ilmu pengetahuan. Dalam perkembangannya *html* telah berkembang sesuai dengan versi masing-masing, yaitu:

1. *HTML* 1.0 (1991)

*HTML* yang pertama kali keluar, dan fitur yang digunakan sangat terbatas. Dalam versi ini kita hanya bisa menggunakan fitur untuk *font*, menggunakan *table*, dan mengubah *background* halaman.

2. *HTML* 2.0 (1995)

Dalam versi ini fitur yang digunakan telah bertambah yaitu konsep standarisasi yang didirikan oleh *W3C (World Wide Web Consortium)*.

3. *HTML* 3.2 (1997)

Dalam versi ini telah didukung suatu elemen dan fitur yang baru, yaitu diterapkannya *CSS (Cascading Style Sheet)*, yang bertujuan untuk membantu mendesain *web*.

4. *HTML* 4.01 (1999)

Pada versi ini telah mengalami perkembangan yaitu ditambahkan konsep *cascading style sheet external*. Dimana perintah dalam *css external*

ini dipanggil dari *HTML* menggunakan perintah tertentu, dan pada versi ini juga ditambahkan kemampuan scripting untuk elemen multimedia.

5. *XHTML* 1.0 (2000)

Awalnya versi ini bertujuan untuk mengganti *HTML* dengan menyediakan halaman web berkualitas tinggi. Pada versi ini semua elemen dalam *HTML* 4.01 telah digunakan dan ditambah dengan sintaks *XML*.

6. *HTML* 5 (2014)

*HTML* 5 merupakan versi terbaru yang digunakan sampai saat ini dengan berbagai keunggulan dan kelebihan dibandingkan versi *HTML* sebelumnya. *HTML* 5 mampu menyederhanakan kode- kode *HTML* terdahulu menjadi lebih ringkas.



**Gambar 2. 6** Logo *HTML*

Sumber: (<https://www.html.com>)

### 2.2.7 *PHP (Personal Home Page)*

Menurut (Fahrozi & Bronson Harahap, 2018) *PHP (Personal Home Page)* ialah bahasa pemrograman yang didesain dalam pengembangan website, dan ditanamkan atau dimasukkan dalam HTML ,serta dieksekusi didalam server sehingga menghasilkan situs web yang dinamis . Salah satu kelebihan *PHP* yaitu kemampuannya dalam melakukan koneksi keberbagai software sistem menjemen basis data/*Database Management System (DBSM)*. Dan hampir semua aplikasi yang berbasis web dapat disusun menggunakan *PHP*.

(Saputra, 2012)Ada beberapa aturan dasar dalam penulisan variable didalam pemrograman *PHP*, yaitu:

1. Penulisan variable ditulis menggunakan awalan simbol Dollar (\$).
2. Karakter pertama setelah simbol dollar harus ditulis dengan huruf.
3. Setelah menggunakan simbol dollar dan karakter huruf maka setelah itu bisa ditulis meenggunakan angka.

Menurut (Saputra, 2012) keunggulan dari *PHP* dibandingkan dengan bahasa pemrograman lainnya, yaitu:

1. *PHP* tidak melakukan kompilasi dalam penggunaannya.
2. *PHP* mempunyai server pendukung yang lebih banyak dibandingkan Bahasa pemrograman biasa.
3. Pengembangan *PHP* lebih mudah, karena didukung tenaga ahli dan *developer* yang banyak.

4. Mudah untuk memahaminya karena banyak sumber yang menjadi referensi.
5. *PHP* bersifat *open source* yang dapat dijalankan diberbagai jenis mesin dan dijalankan secara *runtime*.
6. *PHP* bersifat stabil.

### 2.2.8 Framework Yii 2

Yii adalah kerangka kerja yang berbasis *php* yang menyediakan *reusability* maksimal dalam pemrograman *web* yang mendukung meningkatkan kecepatan pengembangan secara signifikan, yang memiliki konsep MVC (*Model View Controller*). Proyek Yii dimulai sejak 1 Januari 2008 oleh Qiang Xue, seorang *programmer* China dengan memakai bahasa kerja Prado. Yii 1.0 resmi dirilis pada 3 Desember 2008, dan telah mencapai versi V1.1.12 yang dirilis pada 19 Agustus 2012. Pada 12 Oktober 2014 Yii2.0 berhasil dirilis tanpa mengurangi kemudahan dan *ekstensibilitas* asli dari Yii dengan memperbaharui teknologi dan sistem yang baru untuk meningkatkan kemampuan Yii 2.0.

Yii menggunakan pola desain *Model- View- Controller* (MVC) yang memiliki tujuan dalam melepaskan logika bisnis dari perancangan antar muka, untuk mempermudah *developer* dalam membuat perubahan pada setiap komponen yang akan diubah dengan tidak mengubah komponen lainnya.



**Gambar 2. 7** Logo *frame work* *Yii*

Sumber: (<https://www.yiiframework.com>)

### **2.2.9** *Text Editor*

*Text editor* merupakan sebuah perangkat lunak aplikasi komputer yang dapat digunakan dalam membangun, dan mengolah file teks yang ada pada halaman *web* menjadi naskah sederhana. *Text Editor* dapat digunakan dalam pembuatan program komputer dan mengubah kode program yang ada di halaman *web* dengan menggunakan bahasa *pemrograman* tertentu. Saat ini banyak jenis dari *Text editor* yang sering digunakan dan memiliki kelebihan dan kekurangan masing-masing, seperti Notepad++, Sublime Text, Visual Code, Atom dan lain-lain. Penulis menggunakan menggunakan sublime Text 3 dalam merancang aplikais ini, karena aplikasi ini memiliki banyak keunggulan dibandingkan dengan *Text Editor* lainnya.



**Gambar 2. 8** Logo Sublime Text

Sumber: (Sublimetext.com)

### **2.3 Penelitian Terdahulu**

- 1. Dedy Hidayat Kusuma, Moh. Nur Shodiq, Sistem Rekomendasi Destinasi Pariwisata Menggunakan Metode Hibrid Case Based Reasoning dan Location Based Service Sebagai Pemandu Wisatawan di Banyuwangi, Jurnal INTENSIF, Vol.1, No.1, Februari 2017.**

Permasalahan yang dijumpai oleh peneliti dalam penelitian ini yaitu, dengan kekayaan objek wisata yang dimiliki oleh Kabupaten Banyuwangi, dan juga jaraknya yang berdekatan dengan pulau Bali, seharusnya wisatawan akan lebih banyak berkunjung ke daerah tersebut, namun karena kurangnya sistem yang mampu memberikan rekomendasi wisata dan informasi tempat wisata yang membuat wisatawan kebingungan dalam memilih tempat wisata yang akan dikunjungi dan kurangnya rute yang tersedia secara online. Dalam

penelitian tersebut peneliti menggunakan metode case-based reasoning (CBR) dan location based. Sistem yang dirancang nantinya akan diakses wisatawan dalam bentuk aplikasi *mobile*. Dimana dalam aplikasi tersebut tersedia menu jenis wisata, kategori tarif, moda transportasi, aktifitas wisata, dan koordinat GPS wisatawan, sedangkan luaran sistem berupa obyek wisata yang direkomendasikan. Hasil dari penelitian ini yaitu, sebuah aplikasi *mobile* berbasis *android* yang mampu memberikan rekomendasi destinasi pariwisata dan pengaksesan tempat wisata melalui *GPS* berdasarkan pilihan objek wisata yang dipilih oleh wisatawan.

**2. Alvendo Wahyu Aranski, Hery Sanjaya, ANALISIS POTENSI WISATA MENGGUNAKAN LOGIKA FUZZY, Jurnal Ilmiah Informatika (JIF), Vol.6, No.1, Maret 2018.**

Dalam observasi yang dilakukan, peneliti menemukan masalah yang nantinya akan diberikan solusi. Karena posisi Kota Batam yang berpotensi sangat banyak untuk menambah pendapatan negara, baik dari segi industri, pariwisata, perdagangan, alih kapal, maka sangat diperlukan analisis untuk menentukan potensi yang paling mendominasi untuk segera dikembangkan. Pada penelitian ini peneliti menggunakan logika *Fuzzy* dengan metode *Mamdani* karena metode tersebut menghasilkan keputusan yang lebih adil serta lebih manusiawi. Pada penelitian ini, peneliti langsung kelapangan untuk mengambil data, dan menganalisa dengan sistem yang telah dibuat. Hasil dari

penelitian ini, penentuan potensi wisata yang ada di Kota Batam dapat dilakukan dengan baik, dan uji tingkat potensi yang dihasilkan menggunakan metode yang digunakan oleh peneliti telah sama jika dihitung secara manual.

3. **Fenty Ariani, Theresia Cathline Adelia, Yuthsi Aprilinda, Ayu Kartika Puspa, TEMU LAMPUNG BERASIS WEBSITE UNTUK MEMPERMUDAH PENCARIAN TEMPAT WISATA SERTA PEMANDU WISATA DI LAMPUNG, Jurnal Management Sistem Informasi dan Teknologi, Vol.9, No.2, Tahun 2019**

Dalam penelitian ini, masalah yang ditemukan oleh peneliti ialah, lokasi wisata yang ada di Provinsi Lampung sering kali tidak diketahui oleh wisatawan, dan sangat sulit untuk menemukan pemandu wisata, yang akan memberikan petunjuk dan informasi tentang objek wisata di Provinsi Lampung. Pada observasi ini, peneliti menerapkan 3 metode pengumpulan data yaitu Study Literatur, Study Pustaka, Study Lapangan. Dalam penelitian ini peneliti telah membuat sebuah *website* yang akan memberikan informasi dalam mencari lokasi wisata di Provinsi Lampung serta mencari pemandu wisata. Dalam *website* tersebut, para pemandu wisata juga dapat mempromosikan jasa mereka, sesuai dengan lokasi wisata yang mereka ketahui.

**4. Zakarial Anshory, Penerapan Algoritma Ant Colony Optimization Pada Aplikasi Pemandu Wisata Provinsi Sumatera Utara Berbasis Android, Journal of Computer System and Informatics (JoSYC), Volume 1, No. 2, Februari 2020**

Berdasarkan permasalahan yang ditemukan oleh peneliti yaitu, banyaknya tempat wisata yang tidak diketahui oleh wisatawan di Sumatera Utara, dan sulitnya mendapatkan informasi secara *online*, dan pemandu wisata yang ada di Sumatera Utara, maka peneliti berupaya untuk menyelesaikan penelitian ini dengan baik. Metode yang digunakan oleh peneliti yaitu, Algoritma Ant Colony Optimization, Ant Colony Optimization (ACO), karena metode ini menggunakan sistem yang dibuat oleh koloni semut yang mampu menemukan jalan terpendek dari sarang ke tempat mencari sumber makanan berdasarkan jejak kaki yang telah dilalui. Hasil yang diperoleh dari penelitian ini yaitu, pengaksesan rute terpendek ke tempat tujuan wisata dapat diterapkan dengan metode yang digunakan, informasi tentang objek wisata dan pemandu wisata dapat diakses lebih mudah dari aplikasi yang dirancang yang berbasis *Android*.

**5. Rizky Fathurrohman, Maulana Ardhiansyah, Aplikasi Pemandu Wisata Berbasis Web Menggunakan Model Extreme Programming (Studi Kasus: DPD Himpunan Pramuwisata Indonesia DKI Jakarta), Jurnal Ilmu Komputer & Informatika, Vol. 1, No. 1, Juli 2020**

Pada penelitian ini, masalah yang di temukan oleh peneliti yaitu, informasi tentang objek wisata yang belum pernah dikunjungi sebelumnya, dan informasi wisata yang ada di internet seringkali telah lama atau tidak di *update*, maka wisatawan lebih memilih menggunakan jasa pemandu wisata. Tetapi karena sulitnya menemukan pemandu wisata yang akan mendampingi wisatawan selama perjalanannya. Bahkan dengan adanya Himpunan Pramuwisata Indonesia (HPI), seringkali wisatawan kesulitan untuk mendapatkan pemandu wisata yang akan menemani perjalanannya. Pada penelitian ini peneliti menggunakan metode *Agile Software Development* yang berjenis *Extreme Programming (XP)* dan metode pembayaran yang dilakukan dalam penelitian ini ialah *Cash On Delivery (COD)*. Hasil dari penelitian yang dibuat ialah, wisatawan dapat memesan pemandu wisata dengan mudah tanpa harus pergi ke gedung HPI, akurasi dan efisiensi wisatawan dalam mendapatkan informasi destinasi wisata dengan didampingi pemandu wisata.

6. **Raheleh Hassannia, Ali Vatankhah Barenji, Zhi Li 2 and Habib Alipour 1, Web-Based Recommendation System for Smart Tourism: Multiagent Technology**

Peneleitian ini dilatarbelakangi oleh masalah yaitu adanya agen dan hotel yang tidak memiliki pembaruan data otonom untuk operasi mereka, adanya kekurangan platform pengguna akhir yang sesuai dan konfigurasi waktu nyata pada tur paket, adanya kekurangan sistem pemfilteran rekomendasi yang sesuai untuk merekomendasikan paket wisata. Hasil dari penelitian ini ialah adanya aplikasi rekomendasi cerdas berbasis web agent yang menggabungkan data *real-time* dan sistem filter berbasis *hybrid* kedalam industri periwisata pintar untuk merekomendasikan paket wisata berdasarkan permintaan pelanggan.

7. **Juc'elia Giacomelli Beux, Ericles Andrei Bellei, La'is Andressa Brock, Ana Carolina Bertoletti De Marchi, Carlos Amaral H'olbig, Agile Design Process with User-Centered Design and User Experience in Web Interfaces: A Systematic Literature Review, LATIN AMERICAN JOURNAL OF COMPUTING, VOL. 5, NO. 2, NOVEMBER 2018**

Penelitian ini dilatar belakang oleh, banyaknya aplikasi yang dijumpai, dengan beragam jenis metode pengembangan yang tidak memberikan nilai kepuasan kepada *user*, kurangnya pengaturan tampilan aplikasi, informasi yang tidak ter-*update* secara *real-time* yang mengurangi tingkat kepuasan dan penggunaan aplikasi yang dimiliki oleh suatu

perusahaan. Dalam penelitian ini, para peneliti berusaha menganalisis, dan berhasil membuat sebuah kesimpulan bahwa metode yang sebaiknya digunakan dalam merancang suatu aplikasi ialah metode *Agile* dengan jenis *Scrum* dan *XP* karena dengan jenis tersebut, sebuah perusahaan akan dapat mempertimbangkan semua jenis aktifitas dalam perusahaan dan membuat desain yang sangat bagus dan terstruktur yang akan meningkatkan kepuasan pengguna, serta penanganan masalah serta *feedback* yang cepat.

#### **2.4 Kerangka Berfikir**

Kerangka pemikiran merupakan garis besar dari proses yang dilalui oleh peneliti dalam melakukan penelitian. Kerangka pemikirian dibuat untuk membantu peneliti dalam menyelesaikan tahapan dalam penyusunan aplikasi agar tahap demi tahap diselesaikan dengan benar. Berikut ini merupakan kerangka pemikiran yang digunakan oleh peneliti:



**Gambar 2. 9** Kerangka Pemikiran

Sumber: (Data olahan penulis,2021)

Dari kerangka berfikir diatas maka dapat dijelaskan sebagai berikut:

1. Pengumpulan Data

Berdasarkan latar belakang yang diangkat dalam penelitian ini, maka pengumpulan data adalah Langkah pertama yang harus dilakukan agar peneliti dapat menyelesaikan permasalahan yang ada. Metode pengumpulan

data dilakukan dengan cara mengumpulkan informasi dari internet, jurnal, buku, dan pengalaman penulis.

## 2. Pengumpulan software dan Metode yang diterapkan

Tahap kedua yang dilakukan peneliti dalam observasi ini yaitu mengumpulkan software serta metode yang akan diterapkan untuk menyelesaikan masalah yang dijumpai. Pada penelitian ini, penulis membutuhkan beberapa aplikasi dan software pendukung untuk merancang aplikasi yang telah di rencanakan. Metode yang digunakan yaitu metode *Agile* karena metode ini merupakan metode yang sangat *dinamis* dan mengutamakan kepuasan pengguna aplikasi yang di rancang.

## 3. Perancangan Aplikasi

Tahap selanjutnya yang dilakukan dalam penelitian ini yaitu merancang aplikasi mulai dari dasar pembuatan sebuah *website*. Setelah data dikumpulkan dan juga *software* pendukung telah dipersiapkan maka peneliti akan merancang aplikasi sesuai dengan data yang ada, dan menerapkan metode yang telah ditetapkan. Pada tahap ini peneliti juga merancang bagian *interface*, beserta menu yang akan ditampilkan dihalaman utama aplikasi ketika dibuka oleh pengguna.

## 4. Input data pada Aplikasi

Pada tahap ini, semua data yang sudah dikumpulkan dan disimpan di *database* akan gabungan dengan menu akses yang ada pada menu tampilan aplikasi. Pada tahap ini semua bentuk tampilan dan isi dari aplikasi digabungkan untuk memberikan pelayanan terbaik kepada pengguna, serta

penyelesaian tahap perancangan aplikasi. Tahap ini merupakan tahap terakhir sebelum aplikasi tersebut di uji coba.

#### 5. Uji Coba

Pada tahap ini aplikasi telah selesai dirancang, namun tidak langsung di publikasikan karena harus di uji terlebih dahulu, untuk menghindari adanya ketidaknyamanan yang ditemukan ketika menggunakan aplikasi tersebut. Tahap uji coba biasanya dilakukan dengan beberapa percobaan untuk memastikan kelayakan aplikasi tersebut. Apabila ditemukan kekurangan atau kesulitan dalam menggunakan aplikasi tersebut, maka harus dilakukan perbaikan atau *revisi*. Namun jika aplikasi tersebut telah sesuai dengan rancangan dan permintaan kebutuhan, maka aplikasi tersebut dapat digunakan oleh masyarakat.