

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Sistem Informasi Geografis

Menurut Prahasta (2009) Sistem Informasi Geografis (SIG) merupakan gabungan dari tiga unsur pokok: sistem, informasi, geografis. SIG merupakan salah satu informasi dengan tambahan unsur Geografis, atau SIG merupakan suatu sistem yang menekankan pada unsur “informasi geografis”.

##### 1. Sistem

Secara umum, sistem dapat didefinisikan sebagai sekumpulan objek, ide, berikut saling keterkaitannya antar relasi di dalam mencapai suatu tujuan (atau sasaran bersama tertentu). Atau, dengan kata lain, sistem dapat disebutkan sebagai kumpulan komponen (sub-sistem fisik maupun non-fisik/logika) yang saling berhubungan satu sama lainnya dan bekerja sama secara harmonis untuk mencapai suatu tujuan (Prahasta, 2009).

##### 2. Informasi

Menurut John dalam (Prahasta, 2009) Informasi adalah data yang telah ditempatkan pada konteks yang penuh arti oleh penerimanya.

##### 3. Geografis

Istilah “Geografis” merupakan bagian dari spasial (keruangan). Kedua istilah ini sering digunakan secara bergantian atau bahkan tertukar satu sama lainnya hingga muncullah istilah yang ketiga, geospasial. Ketiga istilah ini mengandung pengertian yang kurang lebih serupa di dalam konteks SIG. Penggunaan kata “Geografis” mengandung pengertian suatu persoalan atau hal mengenai (wilayah di permukaan) bumi, baik permukaan dua dimensi atau tiga dimensi.

Dengan demikian istilah “informasi geografis” mengandung pengertian informasi mengenai tempat-tempat yang terletak di permukaan bumi, pengetahuan mengenai posisi dimana suatu objek terletak di permukaan bumi, atau informasi mengenai keterangan-keterangan (atribut) objek penting yang terdapat di permukaan bumi yang posisinya diberikan atau diketahui.

## 2.2 Pariwisata

Menurut Undang-undang Nomor 9 Tahun 1990 bahwa pariwisata adalah segala suatu yang berhubungan dengan wisata, termasuk didalamnya perusahaan obyek dan daya tarik wisata serta usaha-usaha yang berkaitan dibidang tersebut. Istilah pariwisata berhubungan erat dengan pengertian perjalanan pariwisata, yaitu sebagai suatu perubahan tempat tinggal sementara seseorang diluar tempat tinggalnya karena suatu alasan bukan untuk melakukan kegiatan yang menghasilkan upah.

Menurut Ruomei (2002) pariwisata memiliki hubungan yang erat dengan ruang dan geografi, dengan menggunakan sistem informasi geografis dapat meningkatkan pemasaran perhotelan dan pariwisata, dan sistem informasi geografis sangat efektif digunakan dalam pemasaran khususnya bidang pariwisata.

## 2.3 Google Maps

*Google Maps* adalah layanan pemetaan yang di dikembangkan oleh *Google*. Layanan ini memberikan citra untuk satelit, petajalan panorama 360°, kondisi lalu lintas dan perencanaan rute untuk bepergian dengan berjalan kaki, bersepeda, mobil atau angkutan umum. *Google Maps* juga tersedia untuk setiap *platform* yaitu *Android*, *iOS*, *browser web*, dan melalui layanan *web HTTP (Google Developer)*.

## 2.4 Android

*Android* merupakan sistem operasi yang dikembangkan untuk perangkat *mobile* berbasis *Linux*. Pada awalnya sistem operasi ini dikembangkan oleh *Android Inc*, yang kemudian dibeli oleh *Google* pada tahun 2005. Dalam usaha mengembangkan *Android*, Pada tahun 2007 dibentuklah *Open Handser Alliance (OHA)*, Sebuah *konsorsium* dari beberapa perusahaan, yaitu *Texas Instruments*, *Boardcom Corporation*, *Google*, *HTC*, *Intel*, *LG*, *Marvell Technology Grups*, *Motorolla*, *Nvidia*, *Qualcomm*, *Samsung Electronics*, *Sprint Nextel* dan *T-Mobile* dengan tujuan untuk mengembangkan standar terbuka untuk perangkat *mobile*. Pada tanggal 9 Desember 2008, diumumkan bahwa 14 anggota baru akan

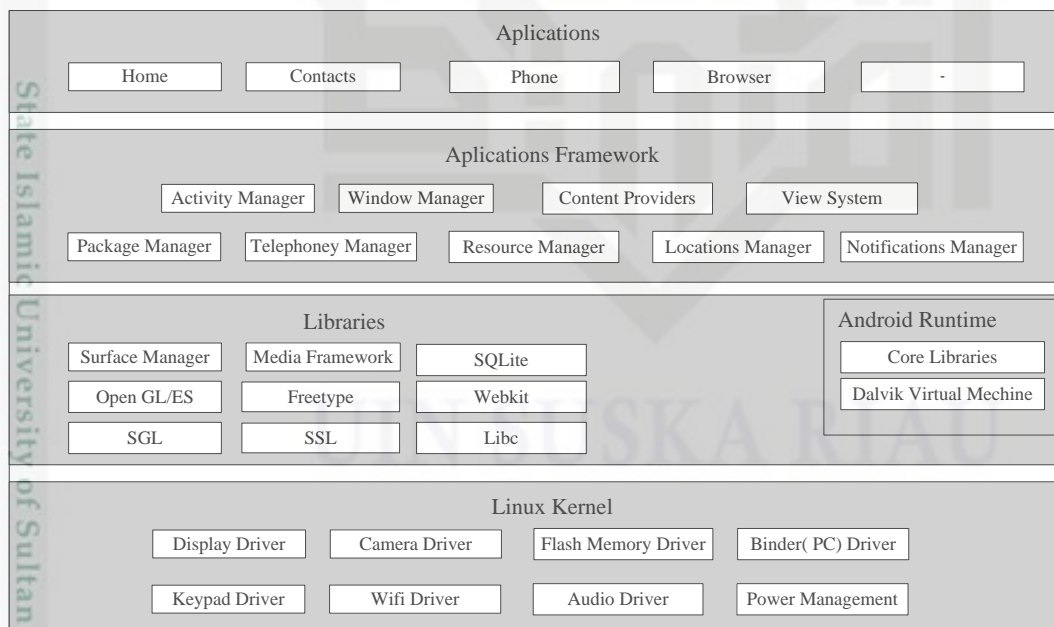
bergabung proyek *Communications, Asustek Computer Inc, Garmin Ltd, Softbank, Sony Erection, Toshiba Corp dan Vedafone Grups Pplc* (Hermawan, 2011).

### 2.4.1 Fitur dan Arsitektur *Android*

Fitur yang tersedia pada *Android* adalah:

1. *Framework* aplikasi memungkinkan penggunaan dan pemindahan dari komponen yang tersedia.
2. *Dalvik Virtual Machine, Virtual Machine* yang dioptimalkan untuk perangkat *mobile*.
3. Grafik 2D dan Grafik 3D yang didasarkan pada *library Open GL*.
4. *SQLite* untuk penyimpanan data.
5. Media seperti audio, video dan berbagai format gambar.
6. *GSM, Bluetooth, EDGE, 3G dan WiFi*.
7. Kamera, *Global Positioning System (GPS), Compass dan accelerometer*.
8. Lingkungan pengembangan yang kaya, termasuk *emulator*, peralatan *debugging* dan *plugin* untuk *Eclipse IDE*.

Sistem operasi *Android* dibangun berdasarkan *kernel linux* dan memiliki arsitektur yang dapat dilihat pada Gambar 2.1.



Gambar 2.1 Arsitektur *Android*

Sumber: (Hermawan, 2011)

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

a. *Applications*

Lapisan ini adalah lapisan aplikasi, serangkaian aplikasi akan terdapat pada perangkat *mobile*. Aplikasi ini yang telah terdapat pada *Android* termasuk kalender, kontak, SMS dan lain sebagainya. Aplikasi-aplikasi ini ditulis dengan bahasa pemrograman *Java*.

b. *Application Framework*

Pengembangan aplikasi memiliki akses penuh ke *Android* sama dengan aplikasi inti yang telah tersedia. Pengembang dapat dengan mudah mengakses informasi lokasi, mengatur alarm menambahkan pemberitahuan ke *status bar* dan lain sebagainya. Arsitektur aplikasi ini dirancang untuk menyederhanakan penggunaan kembali komponen, aplikasi apa pun dapat memublikasikan kemampuan dan aplikasi lain dapat menggunakan kemampuan mereka sesuai batasan keamanan. Dasar dari aplikasi adalah seperangkat layanan sistem yaitu berbagai *View* yang digunakan untuk mengembangkan *user interface*, *Content Provider* yang memungkinkan aplikasi berbagai data, *Resource Manager* menyediakan akses bukan kode seperti grafik, *string* dan *layout*, *Notification Manager* yang akan membuat aplikasi dapat menampilkan tanda pada *status bar* dan *Activity Manager* yang berguna mengatur daur hidup dari aplikasi.

c. *Libraries*

Satu set *libraries* dalam bahasa C/C++ yang digunakan oleh berbagai komponen pada sistem *Android*.

d. *Android Runtime*

Satu set *libraries* inti yang menyediakan sebagian besar fungsi yang tersedia di *libraries* inti dari bahasa pemrograman *Java*. Setiap aplikasi akan berjalan sebagai proses sendiri pada *Dalvik Virtual Machine* (VM).

e. *Linux Kernel*

*Android* bergantung pada *linux* versi 2.6 untuk layanan sistem inti seperti keamanan, manajemen memori, manajemen proses, *network*

**Hak Cipta Dilindungi Undang-Undang**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

*stack* dan modal *driver*. *Kernel* juga bertindak sebagai antara *hardware* dan seluruh *software*.

### 2.4.2 Versi *Android*

Banyak *smartphone* dan PC tablet menggunakan sistem operasi dengan versi yang berbeda. Semakin versi tinggi, fiturnya semakin canggih dan banyak. telepon pertama yang memakai operasi *android* adalah *HTC Dream* yang dirilis pada tanggal 22 oktober 2008 dapat dilihat pada Tabel 2.1 (Istiyanto, 2013).

Tabel 2.1 Versi *Android*

No	Versi <i>Android</i>	Keterangan
1	<i>Android</i> Versi 1.1	Pertama kali dirilis oleh <i>Google</i> pada tanggal 9 Maret 2009. Versi ini memiliki fitur jam, alarm, <i>voice search</i> (pencari suara), pengirim pesan dengan <i>Gmail</i> dan pemberitahuan.
2	<i>Android</i> Versi 1.5	Merupakan versi yang dirilis <i>Google</i> pada pertengahan Mei 2009. Fitur-fitur yang dimiliki versi ini adalah merekam dan menonton video dengan modus kamera, mengupload video ke <i>youtube</i> , meng- <i>upload</i> gambar <i>picasa</i> dari telepon, <i>bluetooth</i> A2DP, terhubung langsung ke <i>headset bluetooth</i> , animasi layar, <i>keyboard</i> pada layar yang disesuaikan.
3	<i>Android</i> Versi 1.6	Dirilis pada September 2009 dengan fitur proses pencarian yang lebih baik dibanding versi sebelumnya, menggunakan baterai indikator dan <i>control applet</i> VPN, galeri memilih foto yang dihapus, kamera, <i>camcorder</i> , CDMA/EVDO, 802.1x, <i>gestures</i> , <i>text-to-speech engine</i> dan <i>dial contact</i>
4	<i>Android</i> Versi 2.0/2.1	Merupakan versi yang dirilis <i>Google</i> pada tanggal 3 Desember 2009, memiliki fitur mengoptimalkan <i>hardware</i> versi sebelumnya, <i>Google maps</i> 3.1.2, perubah UI, <i>browser</i> baru, HTML5, daftar kontak baru, <i>flash</i> untuk kamera 3.2 MP, <i>digital zoom</i> , <i>bluetooth</i> 2.1.

Tabel 2.1 Versi *Android* (Lanjutan)

No	Versi <i>Android</i>	Keterangan
5	<i>Android</i> Versi 2.2	Merupakan versi yang dirilis <i>Google</i> pada Mei 2010. Versi pertama memiliki fitur dapat menghapus komponen, DVM dioptimalkan, <i>graphic</i> 2D dan 3D, <i>SQLite</i> , media audio dan video berbagai format, GSM, <i>bluetooth</i> , EDGE, 3G, <i>Wifi</i> , <i>camera</i> , <i>Global Positioning System</i> (GPS), kompas, <i>accelormeter</i> (tergantung <i>hardware</i> ).
6	<i>Android</i> Versi 2.3	Merupakan versi yang dirilis <i>Google</i> pada Desember 2010, memiliki fitur SIP-based VoIP, NFC ( <i>Near Field Communications</i> ), <i>Gyroscope</i> dan Sensor, <i>Multiple Cameras Support</i> , <i>Mixable Audio Effect</i> dan <i>download manager</i> .
7	<i>Android</i> Versi 3.0/3.2	Merupakan versi yang dirilis <i>Google</i> khusus untuk PC <i>tablet</i> . Beberapa <i>smartphone</i> yang tidak dapat menggunakan kamera versi ini memang banyak dipakai bukan untuk <i>voice call</i> , memiliki fitur aksesories API terbuka, USB <i>host</i> API, <i>keyboard extenal</i> dan perangkat penunjuk, <i>joystick</i> dan <i>gamepads</i> , <i>Wifi</i> , perluasan daftar terbaru <i>apps</i> , layar <i>home widget resizeable</i> .
8	<i>Android</i> Versi 4.0	<i>Android Ice Cream Sandwich</i> atau biasa dikenal dengan nama <i>Android</i> 4.0 ICS adalah versi terbaru <i>Android</i> yang sangat mendukung baik untuk <i>smartphone</i> , <i>tablet</i> , dan lainnya. <i>Android</i> ini rilis pada 19 October 2011. Yang baru dalam <i>android</i> ini adalah perubahan <i>interface</i> dari <i>android</i> sebelumnya, antara lain pengoptimalan <i>multitasking</i> , variasi layar beranda yang bisa disesuaikan dan interaktivitas mendalam serta cara baru yang ampuh untuk berkomunikasi dan berbagi konten.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Tabel 2.1 Versi *Android* (Lanjutan)

No	Versi <i>Android</i>	Keterangan
9	<i>Android</i> Versi 4.1-4.3	<i>Android</i> 4.1 <i>Jelly Bean</i> diumumkan pada 27 Juni 2012 pada konferensi <i>Google I/O</i> yang secara resmi dikenalkan ke publik sekitar Oktober 2012. Fitur baru yang terapat di versi ini adalah meningkatkan kemudahan dan keindahan tampilan dari <i>Ice Cream Sandwich</i> dan memperkenalkan pengalaman pencarian <i>Google</i> yang baru di <i>Android</i> . <i>Android</i> 4.2 <i>Jelly Bean</i> juga menawarkan peningkatan kecepatan dan kemudahan <i>Android</i> 4.1 serta mencakup semua fitur baru seperti <i>Photo Sphere</i> dan desain baru aplikasi kamera, <i>keyboard Gesture Typing</i> , <i>google Now</i> dan lainnya.
10	<i>Android</i> Versi 4.4	Awalnya <i>android</i> versi ini di isukan bernama <i>Key Lime Pie</i> . Namun pada tanggal Oktober 2013 <i>Google</i> merilis kitkat sebagai generasi <i>android</i> berikutnya. <i>Android</i> versi ini memiliki banyak fitur dan semakin memanjakan para pengguna <i>android</i> . Diantaranya: <i>Immersive mode</i> , Akses kontak langsung dari aplikasi telepon, <i>Google now launcher</i> , dan pastinya memiliki <i>interface</i> UI yang baru.
11	<i>Android</i> Versi 5.0	<i>Android</i> 5.0 merupakan versi paling baru dari sistem operasi <i>android</i> , <i>Android</i> 5.0 sendiri dianggap membawa <i>update</i> yang fantastis, banyak perubahan yang disertakan <i>Google</i> di dalamnya.
12	<i>Android</i> Versi 6.0	<i>Android Marshmallow</i> memberikan dukungan asli untuk pengenalan sidik jari, memungkinkan penggunaan sidik jari untuk membuka perangkat dan <i>otentikasi Play Store</i> dan pembelian <i>Android Pay</i> , API standar juga tersedia untuk melaksanakan <i>otentikasi</i> berbasis sidik jari dalam aplikasi lain. <i>Android Marshmallow</i> mendukung <i>USB Type-C</i> , termasuk kemampuan untuk menginstruksikan perangkat untuk mengisi daya perangkat lain melalui <i>USB</i> .

Sumber: (Istiyanto, 2013).

## 2.5 Web

*World wide web consortium* (W3C) adalah komunitas Internasional dimana anggota organisasi melakukan pekerjaan umum bersama untuk mengembangkan standar *web*. Dipimpin oleh *web* penemu Tim Bernes-Lee dan CEO Jeffrey Jaffe, misi W3C adalah untuk memimpin *web* secara maksimal.

Standar W3C mendefinisikan *web platform* terbuka untuk pengembangan aplikasi yang memiliki potensi belum pernah terjadi sebelumnya untuk memungkinkan pengembang untuk membangun pengalaman interaktif yang kaya, didukung oleh data toko besar yang tersedia pada perangkat apapun. Meskipun batas-batas *platform* terus berkembang, pemimpin industri berbicara hampir serempak tentang bagaimana HTML5 akan menjadi landasan untuk platform ini. Tapi kekuatan *platform* bergantung pada lebih banyak teknologi yang W3C dan mitranya ciptakan, termasuk CSS, SVG, WOFF, *web* tumpukan Sematic, XML dan berbagai API (W3C).

## 2.6 Geocoding Api

*Geocoding* adalah proses konversi alamat (seperti “1600 Amphitheatre Parkway, Mountain View, Ca”) menjadi koordinat geografis seperti Garis Lintang 32,423021 dan Garis Bujur -122,083739, yang bisa digunakan untuk menempatkan *marker* pada peta atau memosisikan peta (*Google Developer*).

*Geocoding* terbalik adalah proses mengonversi koordinat menjadi alamat yang bisa dibaca orang.

## 2.7 Directions Api

Terdapat 3 mode dalam *directions api* (*Google Developer*), yaitu:

### 1. Mode Transportasi

Saat menghitung arah, bisa menetapkan mode transportasi yang akan digunakan. Secara *default*, arah dihitung sebagai arah *driving*. Mode perjalanan berikut ini telah didukung:

- Driving* (*default*) menunjukkan arah mengemudi standar yang menggunakan jaringan jalan raya.



Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

- b. *Walking* meminta arah berjalan melalui jalur pejalan kaki dan trotoar bila tersedia.
- c. *Bicyling* meminta arah bersepeda melalui rute angkutan umum (bila tersedia).
- d. *Transit* meminta arah melalui rute angkutan umum (bila tersedia). Jika mengatur mode ke *transit* kita juga bisa menetapkan *departure time* atau *arrival time*. Jika tidak ada waktu yang ditetapkan, *default departure time* adalah sekarang (yaitu waktu keberangkatan *default* adalah saat ini).

2. Mode Titik Jalan

Saat menghitung rute menggunakan *Google Maps directions api*, anda juga bisa menetapkan titik jalan untuk arah mengemudi, berjalan atau bersepeda (titik jalan tidak tersedia untuk arah angkutan umum). Titik jalan memungkinkan anda menghitung rute melalui lokasi tambahan, dimana rute yang dikembalikan menyertakan persinggahan disetiap titik jalan. Titik jalan ditetapkan dalam parameter *waypoints* dan terdiri dari satu atau beberapa alamat atau lokasi yang dipisah dengan karakter pipa.

3. Waktu Tempuh

Menetapkan asumsi yang akan digunakan saat menghitung waktu dalam lalu lintas. Peraturan ini mempengaruhi nilai yang dikembangkan dibidang *duration in traffic* dalam *respons*, yang berisi prediksi waktu dalam lalu lintas berdasarkan rata-rata *historis*. Parameter *traffic model* hanya bisa ditetapkan untuk arah mengemudi yang permintaannya menyertakan *departure time*, dan hanya jika permintaan menyertakan kunci api atau id *client Google Maps api for work*. Nilai yang tersedia untuk parameter ini adalah:

- a. *Best guest (default)* menunjukkan *duration in traffic* yang dikembalikan harus berupa perkiraan waktu tempuh terbaik berdasarkan informasi riwayat kondisi lalu lintas dan lalu lintas saat ini. Lalu lintas saat ini menjadi kian penting bila *departure time* semakin dekat ke waktu sekarang.

- b. Optimistik menunjukan *duration in traffic* yang dikembangkan lebih lama dari waktu tempuh sesungguhnya dihari-hari biasa, meskipun hari-hari tertentu dengan kondisi lalu lintas yang buruk mungkin melebihi saat ini.

## 2.8 MySQL

MySQL adalah sebuah aplikasi *Relational Database Management Server* (RDBMS) yang sangat cepat dan kokoh serta bersifat *open source*. MySQL merupakan salah satu jenis *database server* yang banyak digunakan di dunia maya, yang menggunakan *SQL* sebagai bahasa dasar untuk mengakses *database*. MySQL dapat digunakan pada berbagai *platform* sistem operasi. Keunggulan MySQL dalam mengolah *database* adalah:

1. Kecepatan.
2. Mudah digunakan.
3. *Open Source*.
4. Kapabilitas.
5. Biaya murah.
6. Keamanan.
7. Lintas *Platform*.
8. Minim "bug".

## 2.9 Hypertext Preprocessor (PHP)

PHP yang merupakan bahasa pemrograman berbasis *web* yang memiliki kemampuan untuk memproses data dinamis. PHP dikatakan sebagai sebuah *server-side embedded script language* artinya *sintaks-sintaks* dan perintah yang kita berikan akan sepenuhnya dijalankan oleh *server* tetapi disertakan pada halaman HTML biasa.

Aplikasi-aplikasi yang dibangun oleh PHP pada umumnya akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan di *server*. Pada prinsipnya *server* akan bekerja apabila ada permintaan dari *client*. Dalam hal ini *client* menggunakan kode-kode PHP untuk mengirimkan permintaan ke *server*.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Cara kerja PHP yang akan kita bahas pada buku ini adalah PHP sebagai bahasa pemrograman untuk mengembangkan aplikasi berbasis *web*. Karena selain untuk *web programming* PHP juga dapat digunakan untuk mengembangkan aplikasi berbasis *desktop* dan *Command Line Interface (CLI)*. Berikut Ini adalah keunggulan PHP:

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. Dalam sisi pengembangan lebih mudah, karena banyaknya forum dan *developer* yang siap membantu dalam pengembangan.
3. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
4. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (*Linux, Unix, Macintosh, Windows*) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

## 2.10 Unified Modelling Language (UML)

### 2.10.1 Pengertian UML

UML adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis *Object-Oriented*. UML juga merupakan sistem notasi yang membantu pemodelan sistem menggunakan konsep berorientasi objek.

UML adalah sebuah alat bantu yang sangat handal di dunia pengembangan sistem berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan *visual* yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain (Nugroho, 2005).

UML dibangun atas model 4+1 view, model ini didasarkan pada fakta struktur sebuah sistem dideskripsikan dalam 5 *view*, dimana salah satunya adalah *Use Case View*. Dalam proyek pengembangan sistem apapun, fokus utama dalam

**Hak Cipta Dilindungi Undang-Undang**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

analisis dan perancangan adalah model. Hal ini berlaku umum tidak hanya untuk perangkat lunak. Dengan model kita dapat merepresentasikan sesuatu karena:

1. Model mudah dan cepat untuk dibuat.
2. Model bisa digunakan sebagai simulasi untuk mempelajari lebih detail tentang sesuatu.
3. Model bisa dikembangkan sejalan dengan pemahaman kita tentang sesuatu.
4. Model bisa mewakili sesuatu yang nyata maupun tidak nyata.


**2.10.2 Diagram-Diagram Dalam UML**

UML mempunyai sejumlah elemen grafis yang bisa dikombinasikan menjadi *diagram*. Karena ini merupakan sebuah bahasa, UML mempunyai aturan untuk menggabungkan dan mengkombinasikan elemen-elemen tersebut. Dalam membangun suatu model perangkat lunak dengan UML, digunakan bentuk-bentuk *diagram* atau simbol untuk merepresentasikan elemen-elemen dalam sistem. Bentuk *diagram* yang digunakan untuk merepresentasikannya adalah sebagai berikut:

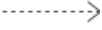

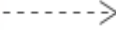





1. *Use Case Diagram*

*Use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case diagram* mendiskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case diagram* digunakan untuk mengetahui fungsi apa saja yang ada dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu, notasi *use case diagram* dapat di lihat pada Tabel 2.2 (Shalahuddin, 2015).

Tabel 2.2 Notasi *Use Case Diagram*

Gambar	Nama	Keterangan
	Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .


Tabel 2.2 Notasi *Use Case Diagram* (Lanjutan)

Gambar	Nama	Keterangan
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
	<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> ).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya ( <i>sinergi</i> ).

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



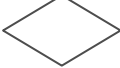

Tabel 2.2 Notasi *Use Case Diagram* (Lanjutan)

Gambar	Nama	Keterangan
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.


## 2. Activity Diagram

*Diagram* aktivitas menggambarkan aliran kerja (*workflow*) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak, yang perlu diperhatikan disini adalah *diagram* aktivitas menggambarkan aktivitas sistem, bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. *Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses *paralel* yang mungkin terjadi pada beberapa eksekusi, *notasi activity diagram* dapat dilihat pada Tabel 2.3 (Shalahuddin, 2015).

Tabel 2.3 Notasi *Activity Diagram*

Simbol	Nama	Keterangan
	Status awal	Status awal aktivitas sistem, sebuah <i>diagram</i> aktivitas memiliki status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

Tabel 2.3 Notasi *Activity Diagram* (Lanjutan)


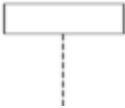
Simbol	Nama	Keterangan
	Status akhir	Status akhir yang dilakukan sistem, sebuah <i>diagram</i> aktivitas memiliki sebuah status akhir.

### 3. *Sequence Diagram*



*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Maka dari itu untuk menggambarkan *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* serta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

*Sequence diagram* mendokumentasikan komunikasi/interaksi antar kelas-kelas. *Diagram* ini menunjukkan sejumlah obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek didalam *use case*. Perlu diingat bahwa di dalam diagram ini, kelas-kelas dan aktor-aktor diletakkan dibagian atas diagram dengan urutan dari kiri ke kanan dengan garis *lifeline* yang diletakkan secara *vertikal* terhadap kelas dan *actor*, notasi *sequence diagram* dapat dilihat pada Tabel 2.4 (Shalahuddin, 2015).

Tabel 2.4 Notasi *Sequence Diagram*

Simbol	Nama	Keterangan
	<i>Actor</i>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi dan mendapat manfaat dari sistem. Ditempatkan di bagian atas diagram
	<i>LifeLine</i>	Menyatakan hidup suatu objek
<div style="border: 1px solid black; padding: 2px; display: inline-block;">Nama Objek:nama kelas</div>	<i>Object</i>	Menyatakan objek yang berinteraksi pesan

Tabel 2.4 Notasi *Sequence Diagram* (Lanjutan)

SIMBOL	NAMA	KETERANGAN
	<i>Message</i>	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
Pesan tipe create 	<i>Message</i>	Objek yang lain, arah panah mengarah pada objek yang di buat

#### 4. *Class Diagram*



*Class Diagram* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah obyek dan merupakan inti dari pengembangan dan desain berorientasi obyek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi), *notasi class diagram* dapat dilihat pada Tabel 2.5 (Shalahuddin, 2015).

Tabel 2.5 Notasi *Class Diagram*

SIMBOL	NAMA	KETERANGAN
	<i>Class</i>	<i>Class</i> adalah blok- blok pembangun pada pemrograman berorientasi obyek. Sebuah <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian.
	<i>Assosiation</i>	Sebuah asosiasi merupakan sebuah <i>relationship</i> paling umum antara dua <i>class</i> , dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i> .
	<i>Composition</i>	Sebuah <i>relationship composition</i> digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid.
	<i>Dependency</i>	Kadangkala sebuah <i>class</i> menggunakan <i>class</i> yang lain. Penggunaan <i>dependency</i> di gunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain.

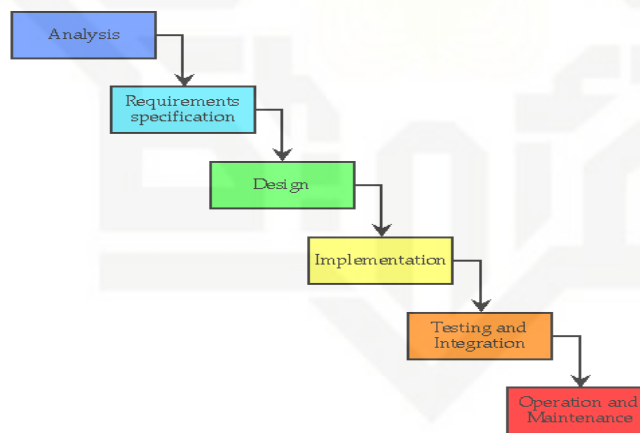


Tabel 2.5 Notasi *Class Diagram* (Lanjutan)

SIMBOL	NAMA	KETERANGAN
	<i>Aggregation</i>	Mengindikasikan keseluruhan bagian nya <i>relationship</i> dan biasanya disebut sebagai relasi mempunyai sebuah atau bagian dari.
	<i>Generalization</i>	Sebuah relasi <i>generalization</i> sepadan dengan sebuah relasi <i>inheritance</i> pada konsep berorientasi obyek.

### 2.11 Metode Pengembangan *Waterfall*

Model *Waterfall* merupakan salah satu metode dalam *Systems Development Life Cycle* (SDLC). Menurut Azhari (2010) metode ini merupakan metode dengan model sekuensial, sehingga penyelesaian satu set kegiatan menyebabkan dimulainya aktivitas berikutnya. Fokus terhadap masing-masing *fase* dapat dilakukan maksimal karena tidak adanya pengerjaan yang sifatnya *paralel*. Tahapan *waterfall* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Tahapan Metode *Waterfall*

#### 1. *Requirement Analysis*

Seluruh kebutuhan *software* didapatkan pada fase ini, termasuk didalamnya kegunaan *software* yang diharapkan pengguna dan batasan *software*. Informasi ini biasanya dapat diperoleh melalui wawancara, survey atau diskusi. Informasi tersebut dianalisis untuk mendapatkan

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

dokumentasi kebutuhan pengguna untuk digunakan pada tahap selanjutnya.

2. *System Design*

Tahap ini dilakukan sebelum melakukan tahapan *coding system*. Tahap ini bertujuan untuk memberikan gambaran seperti apa sistem yang akan dibuat dan bagaimana *interface* untuk setiap kegiatannya. Tahap ini membantu dalam menspesifikasikan kebutuhan *hardware* dan sistem serta mendefinisikan arsitektur sistem secara keseluruhan.

3. *Implementation*

Pada tahap ini dilakukan tahapan pemrograman. Pembuatan *software* dipecah menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap berikutnya. Selain itu dalam tahap ini juga dilakukan pemeriksaan terhadap modul yang dibuat, apakah telah memenuhi fungsi yang diinginkan atau belum.

4. *Integration and Testing*

Pada tahap ini dilakukan penggabungan modul-modul yang telah dibuat dan dilakukan pengujian untuk mengetahui apakah *software* telah sesuai atau belum sesuai dengan desainnya.

5. *Operation and Maintenance*

Ini merupakan tahap terakhir dalam model *waterfall*. *Software* yang sudah jadi, dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

**2.12 Pengujian Sistem**

Pengujian *software* sangat diperlukan untuk memastikan *software/aplikasi* yang sudah/sedang dibuat dapat berjalan sesuai dengan *fungsi* yang diharapkan. Pengembang atau penguji *software* harus menyiapkan sesi khusus untuk menguji program yang sudah dibuat agar kesalahan ataupun kekurangan dapat dideteksi sejak awal dan dikoreksi secepatnya. Pengujian atau *testing* sendiri merupakan elemen kritis dari jaminan kualitas perangkat lunak dan merupakan bagian yang tidak terpisah dari siklus

hidup pengembangan *software* seperti halnya analisis, desain, dan pengkodean. (Mustaqbal, 2015).

Pengujian *software* haruslah dilakukan dalam proses rekayasa perangkat lunak atau *software engineering*. Sejumlah strategi pengujian *software* telah diusulkan dalam literatur. Semuanya menyediakan template untuk pengujian bagi pembuat *software*. Dalam hal ini semuanya harus memiliki karakteristik umum berupa:

1. *Testing* dimulai pada level modul dan bekerja keluar ke arah integrasi pada sistem berbasis komputer.
2. Teknik *testing* yang berbeda sesuai dengan poin-poin yang berbeda pada waktunya.
3. *Testing* diadakan oleh pembuat/pengembang *software* dan untuk proyek yang besar oleh *group testing* yang *independent*.
4. *Testing* dan *Debugging* adalah aktivitas yang berbeda tetapi *debugging* harus diakomodasikan pada setiap strategi *testing*.

Langkah-langkah pengujian *software* ada 4, yaitu:

- a. *Unit testing*-testing per unit yaitu mencoba alur yang spesifik pada struktur modul kontrol untuk memastikan pelengkapan secara penuh dan pendeteksian *error* secara maksimum.
- b. *Integration testing*-testing perpenggabungan unit yaitu pengalamatan dari isu-isu yang diasosiasikan dengan masalah ganda pada verifikasi dan konstruksi program.
- c. *High-order test* yaitu terjadi ketika *software* telah selesai diintegrasikan atau dibangun menjadi satu, tidak terpisah-pisah.
- d. *Validation test* yaitu menyediakan jaminan akhir bahwa *software* memenuhi semua kebutuhan fungsional, kepribadian dan performa.

### 2.12.1 Pengujian *Blackbox*

*Blackbox* berfokus pada spesifikasi fungsional dari perangkat lunak. *Tester* dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. *Blackbox* bukanlah solusi alternatif dari *Whitebox*

*Testing* tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *Whitebox testing* (Mustaqbal, 2015).

*Blackbox testing* cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

*Blackbox testing* cenderung diaplikasikan selama tahap akhir pengujian.

Pengujian *blackbox* memperhatikan struktur kontrol, sehingga perhatian berfokus pada domain informasi. Saat ini terdapat banyak metoda atau teknik untuk melaksanakan *Blackbox Testing*, antara lain:

1. *Equivalence Partitioning*.
2. *Boundary Value Analysis/Limit Testing*.
3. *Comparison Testing*.
4. *Sample Testing*.
5. *Robustness Testing*.
6. *Behavior Testing*.
7. *Requirement Testing*.
8. *Performance Testing*.
9. Uji Ketahanan (*Endurance Testing*).
10. Uji Sebab-Akibat (*Cause-Effect Relationship Testing*).

### 2.12.2 Pengujian *User Acceptance Testing* (UAT)

UAT merupakan pengujian yang dilakukan oleh *end-user* dimana *user* tersebut adalah karyawan perusahaan yang langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi yang ada telah berjalan sesuai dengan kebutuhan/fungsinya.

Menurut Lewis, setelah dilakukan *system testing*, *acceptance testing* menyatakan bahwa sistem *software* memenuhi persyaratan. *Acceptance testing* merupakan pengujian yang dilakukan oleh pengguna yang menggunakan teknik pengujian *blackbox* untuk menguji sistem terhadap spesifikasinya. Pengguna akhir

bertanggung jawab untuk memastikan semua fungsionalitas yang relevan telah diuji. *Acceptance testing* biasanya berusaha menunjukkan bahwa sistem telah memenuhi persyaratan-persyaratan tertentu. Pada pengembangan *software* dan *hardware* komersial, *acceptance test* biasanya disebut juga *alpha tests* (yang dilakukan oleh pengguna *in-house*) dan *beta tests* (yang dilakukan oleh pengguna yang sedang menggunakan atau akan menggunakan sistem tersebut). *Acceptance testing* mencakup data, *environment* dan skenario yang sama atau hampir sama pada saat *live* yang biasanya berfokus pada skenario penggunaan produk tertentu.

Dari definisi di atas, *user acceptance testing* adalah pengujian yang dilakukan oleh pengguna dari sistem tersebut untuk memastikan fungsi-fungsi yang ada pada sistem tersebut telah berjalan dengan baik dan sesuai dengan kebutuhan pengguna.

### 2.13 Astah Comunity

Astah Community merupakan perangkat lunak yang penulis gunakan untuk membuat model UML. Astah dibuat oleh perusahaan Jepang bernama Change Vision. Astah terasa “ringan” bahkan untuk pemula sekalipun, hal ini dikarenakan antar-mukanya yang *user-friendly* sehingga memungkinkan pengguna membuat *diagram* UML dengan mudah dan cepat.

Astah Community merupakan *tool* gratis dengan fitur-fitur dasar, dilengkapi dengan fitur mencetak *diagram*, mengimpor/mengekspor ke/dari program java. Astah Community yang dulunya bernama Astah Jude telah digunakan sedikitnya 120.000 orang diseluruh dunia pada tahun 2006.

### 2.14 Provinsi Riau

Provinsi Riau secara geografis, geoekonomi dan geopolitik terletak pada jalur yang strategis baik pada masa kini dan masa yang akan datang karena terletak pada wilayah jalur perdagangan regional maupun internasional di kawasan ASEAN melalui kerjasama IMT-GT dan IMS-GT. Wilayah Provinsi Riau terletak antara 01°05'00” Lintang Selatan sampai 02°25'00” Lintang Utara dan 100°00'00” sampai 105°05'00” Bujur Timur.

**Hak Cipta Dilindungi Undang-Undang**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

### 2.14.1 Profil Dinas Pariwisata Provinsi Riau

Dinas Pariwisata Provinsi Riau dahulunya berdasarkan Peraturan Daerah Provinsi Riau Nomor 53 Tahun 2009 bernama Dinas Kebudayaan dan Pariwisata Provinsi Riau. Sesuai dengan perkembangan tentang penataan organisasi Pemerintah Daerah maka disusun Peraturan Daerah Provinsi Riau Nomor 2 Tahun 2014 Tentang Organisasi Dinas Daerah Provinsi Riau (Lembaran Daerah Provinsi Riau Tahun 2014 Nomor 2), dimana berdasarkan Peraturan Daerah tersebut diatur uraian tugas Dinas Pariwisata Provinsi Riau yang dituangkan melalui Peraturan Gubernur Riau Nomor Tahun 2014 tentang Rincian Tugas, Fungsi dan Tata Kerja Dinas Pariwisata Provinsi Riau.

### 2.14.2 Visi dan Misi Dinas Pariwisata Provinsi Riau

#### 1. Visi

Terwujudnya Provinsi Riau sebagai daerah tujuan wisata yang berdaya saing, didukung oleh budaya melayu dan ekonomi kreatif untuk kesejahteraan masyarakat.

#### 2. Misi

- a. Mengembangkan daerah tujuan wisata yang berdaya saing dan berkelanjutan yang didukung oleh kebudayaan melayu sebagai kekayaan dan kearifan lokal.
- b. Meningkatkan kualitas sumber daya manusia aparatur bidang pariwisata dan ekonomi kreatif yang didukung oleh sarana dan prasarana, serta penguasaan terhadap teknologi.
- c. Mendorong peningkatan peluang usaha dan kesempatan kerja melalui fasilitasi pengembangan usaha dan peningkatan jejaring, serta penghargaan terhadap hak kekayaan intelektual dan prestasi di bidang pariwisata dan ekonomi kreatif.
- d. Melaksanakan pemasaran pariwisata yang lebih strategis, di dalam dan di luar negeri, dengan memanfaatkan *event* promosi pariwisata, media, jaringan pemasaran pariwisata dan perkembangan teknologi informasi.

- e. Meningkatkan peran serta pemangku kepentingan (*stakeholders*), dan meningkatkan kerjasama dengan berbagai pihak dalam rangka pembangunan pariwisata daerah.
- f. Melaksanakan beberapa kebijakan khusus di bidang pariwisata dalam rangka pengembangan desa-desa wisata, *event-event* pariwisata daerah, serta mewujudkan Provinsi Riau sebagai destinasi wisata syariah, dan mewujudkan Kota Pekanbaru sebagai destinasi wisata *meeting, incentive, conference and exhibition* (MICE).

### 2.14.3 Data Kunjungan Wisatawan di Provinsi Riau

Data kunjungan wisatawan yang berkunjung di Provinsi Riau pada tahun 2015 dan 2016 dapat dilihat pada Tabel 2.6 dan Tabel 2.7.

Tabel 2.6 Kunjungan Wisatawan 2015

Bulan	Sultan Syarif kasim	Dumai dan bengkalis	Wisnus
Januari	2.606	954	100.000
Februari	2.043	1.206	350.000
Maret	2.300	902	125.000
April	2.112	881	350.000
Mei	2.503	334	175.000
Juni	1.954	1.492	450.000
Juli	2.660	2.717	365
Agustus	2.488	1.256	650
September	1.752	4.622	432.651
Oktober	462	8.378	243.880
November	2.211	3.255	656.961
Desember	3.362	1.988	589.000
Jumlah	2.6453	27.994	4.487.592
		54.437	

Tabel 2.7 Kunjungan Wisatawan 2016

Bulan	Sultan syarif kasim II	Dumai	Selat panjang	Bengkalis	Wisnus
Januari	1.770	339	-	419	85063
Februari	2.257	1.107	11.000	638	375.281
Maret	2.461	903	-	614	225.076
April	1.953	600	-	576	550.244
Mei	2.747	630	-	626	275.657
Juni	1.900	756	1.560	543	545.270
Juli	2.868	3.048	-	1.082	262.328
Agustus	2.621	621	-	1.082	775.118
September	3.051	957	-	574	472.465
Oktober	3.514	680	-	573	545.890
November	3.280	920	-	728	786.841
Desember		1.872	-	872	925.680

Tabel 2.7 Kunjungan Wisatawan 2016 (Lanjutan)

Bulan	Sultan syarif kasim II	Dumai	Selat Panjang	Bengkalis	Wisnus
Jumlah	28.422	12.433	12.560	8.327	5.827.913
				61.742	

Sumber: (Dinas Pariwisata Provinsi Riau, 2016)

#### 2.14.4 Nama Objek Wisata Unggulan di Provinsi Riau

Berdasarkan data dari Dinas Pariwisata Provinsi Riau terdapat 16 objek wisata unggulan yang ada di Provinsi Riau, yaitu:

- a. Pekanbaru : Danau Bandar Kayangan Lembah Sari
- b. Dumai : Danau Bunga Tujuh
- c. Kampar : Candi Muara Takus
- d. Rohul : Air Panas Kaiti, Benteng Tujuh Lapis
- e. Kuansing : Air Terjun Tujuh Tingkat
- f. Indragiri Hulu : Danau Raja
- g. Indragiri Hilir : Pantai Solop
- h. Siak : Istana Siak
- i. Pelalawan : Bono, Taman Nasional Tesso Nilo
- j. Rokan Hilir : Pulau Jemur, Pulau Pedamaran
- k. Bengkalis : Pulau Rukat Utara
- l. Kep. Meranti : Tasik Nambus, Pulau Rangsang

#### 2.15 Penelitian Terdahulu

Penelitian terdahulu yang digunakan dalam penelitian ini dapat dilihat pada Tabel 2.8.

Tabel 2.8 Penelitian Terdahulu

Tahun	Penulis	Hasil
2017	Muhammad Syafrizal	Penelitian yang sedang berjalan dengan judul sistem informasi geografis pariwisata kabupaten Siak berbasis <i>android</i> . Kelebihan dari sistem ini adalah <i>user</i> dapat menerima notifikasi <i>event</i> terbaru dari sistem.



Tabel 2.8 Penelitian Terdahulu (Lanjutan)

Tahun	Penulis	Hasil
2017	Imam Muttaqin, Inggih Permana, Febi Nur Salisah	Didalam penelitiannya mengenai aplikasi <i>navigasi</i> objek wisata Kabupaten Lingga berbasis <i>mobile</i> , hasil dari penelitian ini adalah aplikasi <i>navigasi</i> yang dapat digunakan sebagai penunjuk arah pencarian objek wisata di Kabupaten Lingga.
2016	Muhammad Fatkhur Rahman , Yuli Christyono, and Imam Santoso	Di dalam penelitiannya mengenai rancang bangun aplikasi mytrip untuk pencarian lokasi wisata di wilayah Semarang dalam perangkat <i>android</i> . Hasil dari penelitian ini adalah menampilkan <i>list</i> tempat wisata berdasarkan jenis tempat dan lokasi tempat wisata yang ada disemarang, serta <i>side</i> menu atau filter menu sesuai keinginan.
2015	Fajar Dwi Hernawan, Moehammad Awaluddin, Andri Suprayogi	Di dalam penelitiannya mengenai pembuatan aplikasi peta wisata di Salatiga berbasis <i>mobile</i> GIS memanfaatkan <i>smartphone android</i> hasil dari penelitian ini adalah aplikasi yang digunakan sebagai media pencari rute angkutan kota dan objek wisata yang ada di Salatiga.
2015	Rizki Putra Agrarian, Andri Suprayogi, Bambang Darmo Yuwono	Di dalam penelitiannya mengenai pembuatan aplikasi <i>mobile</i> GIS berbasis <i>android</i> untuk informasi pariwisata di Kabupaten Gunung Kidul. Hasil dari penelitian ini adalah dapat menampilkan informasi berupa foto, lokasi objek wisata.
2014	Ida Bagus Made Yogie Adnyana, Rissal Efendi	Di dalam penelitiannya mengenai sistem informasi geografis persebaran lokasi obyek pariwisata berbasis <i>web</i> dan <i>mobile android</i> . Kelebihan dari sistem ini adalah fitur <i>terrain maps</i> dan <i>satellite maps</i> .

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Tabel 2.8 Penelitian Terdahulu (Lanjutan)

Tahun	Penulis	Hasil
2014	Muhammad Rifqi Andikasani, M. Awaluddin, Andri Suprayogi	Hasil dari penelitian ini adalah dapat menampilkan posisi <i>user</i> , arah kemudi, video preview objek wisata, foto objek wisata dan fungsi panggilan telepon.
2011	Donel Dahesri	Hasil dari penelitian ini adalah dapat menampilkan peta <i>Google maps</i> Kabupaten Kuantan Singingi disertai titik koordinat dan video secara visualisasi tentang keadaan objek.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.