

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Untuk menyelesaikan masalah-masalah dalam penyusunan skripsi ini dibutuhkan tinjauan pustaka yang berisi teori-teori atau konsep-konsep yang digunakan sebagai kajian dan acuan bagi penulis.

A. Konsep Dasar Sistem Informasi Geografis

Sistem Informasi Geografis (SIG) atau juga dikenal sebagai Geographic Information System (GIS) pertama pada tahun 1960 yang bertujuan untuk menyelesaikan permasalahan geografis. 40 tahun kemudian GIS berkembang tidak hanya bertujuan untuk menyelesaikan permasalahan geografi saja tetapi sudah merambah ke berbagai bidang seperti analisis penyakit epidemik (demam berdarah) dan analisis kejahatan (kerusuhan) termasuk analisis kepariwisataan.

1. Definisi Sistem

Sistem secara umum dapat didefinisikan sebagai sekumpulan objek yang saling memiliki keterkaitan. Sistem adalah sekumpulan komponen (sub-sistem) yang saling berhubungan satu sama lainnya dan bekerjasama secara harmonis untuk mencapai suatu tujuan.

2. Definisi Informasi

Informasi adalah suatu data yang telah diproses sehingga dapat mengurangi ketidakjelasan tentang keadaan atau suatu kejadian. Sedangkan kata data itu sendiri

adalah fakta atau kenyataan yang sebenarnya. Informasi juga dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian (event) yang nyata (fact) yang digunakan untuk pengambilan keputusan.

3. Definisi Sistem Informasi Geografis

Aronoff dalam Prahasta (2005) mendefinisikan Sistem Informasi Geografis (SIG) sebagai sistem berbasis komputer yang digunakan untuk menyimpan dan memanipulasi informasi-informasi geografi. SIG dirancang untuk mengumpulkan, menyimpan, dan menganalisis obyek-obyek dan fenomena dimana lokasi geografis merupakan karakteristik yang penting atau kritis untuk dianalisis. Secara umum proses SIG terdiri atas tiga subsistem, yaitu: subsistem masukan data, subsistem manipulasi dan analisis data, dan subsistem penyajian data.

4. Definisi Pariwisata

Pariwisata atau turisme adalah suatu perjalanan yang dilakukan untuk rekreasi atau liburan dan juga persiapan yang dilakukan untuk aktivitas ini. Seorang wisatawan atau turis adalah seseorang yang melakukan perjalanan paling tidak sejauh 80 km (50 mil) dari rumahnya dengan tujuan rekreasi, merupakan definisi oleh Organisasi Pariwisata Dunia.

Definisi yang lebih lengkap, turisme adalah industri jasa. Mereka menangani jasa mulai dari transportasi, jasa keramahan, tempat tinggal, makanan, minuman dan jasa bersangkutan lainnya seperti bank, asuransi, keamanan dll. Dan juga

menawarkan tempat istirahat, budaya, pelarian, petualangan, pengalaman baru dan berbeda lainnya.

Banyak negara bergantung banyak dari industri pariwisata ini sebagai sumber pajak dan pendapatan untuk perusahaan yang menjual jasa kepada wisatawan. Oleh karena itu pengembangan industri pariwisata ini adalah salah satu strategi yang dipakai oleh Organisasi Non-Pemerintah untuk mempromosikan wilayah tertentu sebagai daerah wisata untuk meningkatkan perdagangan melalui penjualan barang dan jasa kepada orang non-lokal.

Menurut Undang Undang No. 10/2009 tentang Kepariwisata, yang dimaksud dengan pariwisata adalah berbagai macam kegiatan wisata yang didukung oleh berbagai fasilitas serta layanan yang disediakan masyarakat, pengusaha, Pemerintah dan Pemerintah Daerah.

5. Definisi *Application Programming Interface (API)*

API atau *application programming interface* adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk sistem operasi tertentu. Menurut *web Google Maps for Bussines*, *Google Maps API* adalah kumpulan API yang memungkinkan pengguna menghamparkan data pada *Google Maps* yang disesuaikan. Pengguna dapat membuat aplikasi *web* dan seluler menarik dengan platform pemetaan canggih dari *Google* termasuk basis data citra satelit, pemandangan jalan, profil ketinggian, petunjuk arah mengemudi, peta dengan sentuhan gaya, demografi, analisis, dan tempat yang luas.

6. Definisi *Google Maps*

Google Maps merupakan salah satu fasilitas dari *Google* yang menyediakan layanan pemetaan suatu daerah. Pemetaan tersebut dilengkapi dengan berbagai kemampuan dan mudah digunakan. Kelengkapan lain pendukung peta tersebut seperti layanan informs bisnis, jasa, layanan public, jalan, lokasi, dan lain-lain (Febrian, 2008). Fitur-fitur yang terdapat dalam *Google Maps* adalah: Integrated business search results, Dragable Maps, Satellite Imagery. Digunakan untuk melihat foto dari satelit. *Detailed directions*, *Keyboard shortcut*, dan *Scroll wheel zooming*. (Febrian, 2008). 2.4.3 *Google Maps API*.

B. *Unified Modelling Language (UML)*

Pendapat tentang UML berbeda satu sama lain. Hal ini dikarenakan sejarahnya sendiri dan oleh perbedaan persepsi tentang apa yang membuat sebuah proses rancang-bangun perangkat lunak efektif. Sukanto dan shalahuddin (2013:133) mendefinisikan bahwa “UML adalah salah satu standar bahasa yang banyak digunakan didunia industri untuk mendefinisikan *requirement*, membuat analisa dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi obyek.

Menurut (Hend, 2006) “*Unified Modelling Language (UML)* adalah bahasa yang telah menjadi standar untuk visualisasi, menetapkan, membangun, dan mendokumentasikan artifak suatu sistem perangkat lunak”.

UML mempunyai dua tipe diagram, strukturan dan *behavioural*. Strukturan diagram menggambarkan bagian statik dari sistem. *Behavioural* diagram menggambarkan

bagian dinamik dari sistem. *Behavioural* diagram diklasifikasikan lebih lanjut ke dalam interaksi dan *state diagram*.

1. Struktural

- a. *Class diagram* menggambarkan hubungan antar objek.
- b. *Use case diagram* digunakan untuk mengorganisasikan *use case* dan *behaviours*.
- c. *Component diagram* adalah komponen dan hubungan yang mengilustrasikan implementasi sistem.
- d. *Deployment diagram* adalah konfigurasi waktu kerja dari node dan objek yang memiliki node.

2. *Behavioural*

- a. *Statechart diagram* menunjukkan urutan kondisi dari objek yang mengalami interaksi beserta respon dan tindakannya.
- b. *Sequence diagram* menggambarkan waktu urutan *message* dan *objek lifeline*.
- c. *Collaboration diagram* menggambarkan urutan *message* dan organisasi objek dalam interaksi.
- d. *Activity diagram* menggambarkan arus dari aktifitas, difokuskan pada operasi yang dilewatkan antar objek.

Pada penelitian dan pengembangan aplikasi sistem ini, tipe UML yang digunakan adalah :

1. *Use Case Diagram*

Menurut Maimunah (2010;30), “ Use case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna”. Use case bekerja dengan cara mendeskripsikan tipikal interaksi antara pengguna sebuah sistem (aktor) dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai.

Use case diagram secara grafis menggambarkan interaksi antara sistem, sistem eksternal dan pengguna. Dengan kata lain use case diagram secara grafis mendeskripsikan siapa yang akan menggunakan sistem dan dalam cara apa pengguna (user) mengharapkan interaksi dengan sistem itu. Use case secara naratif digunakan untuk secara tekstual menggambarkan sekuensi langkah-langkah dari setiap interaksi.

2. *Activity diagram.*

Activity diagram menggambarkan berbagai alir aktivasi dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, delicion yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Aktiviti diagram merupakan state diagram khusus, dimana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *aktiviti diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan

aktivitas. *Decesion* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. Activity diagram dapat dibagi menjadi beberapa *object swimlane* unruk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

3. *Componen Diagram*

Componen diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) diantaranya komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi juga dari komponen-komponen yang lebih kecil. Komponen dapat berupa interface, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

4. *Deployment Diagram*

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, dimana komponen akan terletak (pada mesin, servver atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal. Sebuah *node* adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirment* dapat juga didefinisikan dalam diagram ini.

C. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh System Analysts dalam tahap analisis persyaratan proyek pengembangan system. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain database relasional yang mendasari sistem informasi yang dikembangkan. Rosa dan Shalahuddin (2013:53) “ERD adalah bentuk paling awal dalam melakukan perancangan basis data relasional. Jika menggunakan OODBMS maka perancangan ERD tidak perlu dilakukan”.

1. Komponen *ERD*

Komponen dalam *Entity Relationship Diagram* adalah sebagai berikut:

a. Entitas

Adalah segala sesuatu yang dapat digambarkan oleh data. Entitas juga dapat diartikan sebagai individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain.

b. Atribut

Atribut merupakan pendeskripsian karakteristik dari entitas. Atribut digambarkan dalam bentuk lingkaran atau *elips*. Atribut yang menjadi kunci entitas atau *key* diberi garis bawah.

c. *Relationship*

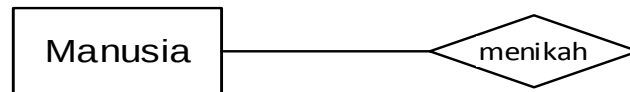
Relasi menunjukkan adanya hubungan diantara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Dalam hal ini di gambarkan dengan garis lurus.

2. Derajat *Relationship*

Derajat dari *relationship* menjelaskan jumlah *entity* yang berpartisipasi dalam suatu *relationship*.

a. *Unary* (Derajat Satu)

Adalah satu buah *relationship* menghubungkan satu buah *entity*.



Sumber: Rosa dan Shalahuddin (2013:52)

Gambar II. 1

Contoh *Unary Relationship*

b. *Binary* (Derajat Dua)

Adalah satu buah *relationship* yang menghubungkan dua buah *entity*.



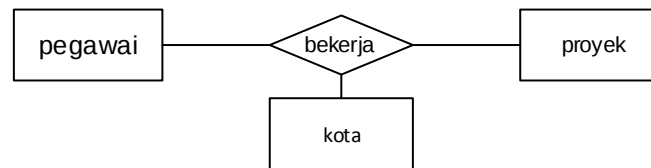
Sumber: Rosa dan Shalahuddin (2013:52)

Gambar II. 2

Contoh *Binary Relationship*

c. *Ternary* (Derajat Tiga)

Adalah satu buah *relationship* menghubungkan tiga buah *entity*.



Sumber: Rosa dan Shalahuddin (2013:52)

Gambar II. 3

Contoh *Ternary Relationship*

Dalam ERD terdapat *Cardinality Ratio Constraint* yang menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Jenis-jenis *Cardinality* adalah sebagai berikut:

1. *One To One* (1: 1)

Yaitu perbandingan antara *entity* pertama dengan *entity* kedua berbanding satu berbanding satu.

2. *One To Many* (1: M)

Yaitu perbandingan antara *entity* pertama dengan *entity* kedua berbanding satu berbanding banyak.

3. *Many To One* (M: 1)

Yaitu perbandingan antara *entity* pertama dengan *entity* kedua berbanding banyak berbanding satu.

4. *Many To Many* (M: N)

Yaitu perbandingan antara *entity* pertama dengan *entity* kedua berbanding banyak berbanding banyak.

2.2. Penelitian Terkait

Sistem informasi wisata adalah sistem informasi yang dibangun untuk memenuhi kebutuhan informasi wisatawan. Informasi dibedakan menjadi tiga bagian yaitu informasi sebelum perjalanan, informasi saat perjalanan dan informasi setelah perjalanan. Sebuah sistem informasi wisata yang baik harus dapat menyediakan informasi dari keseluruhan bagian tersebut. Dalam bagian pertama wisatawan merencanakan perjalanan, tempat yang dikunjungi, apa yang ingin dilakukan, dan informasi yang ingin diperoleh. Untuk penyediaan informasi, Sistem Informasi Geografis (SIG) merupakan alat bantu yang tepat untuk menangani masalah ini. SIG menggunakan Google Maps API memiliki kemampuan untuk menyediakan informasi dan menampilkan dalam bentuk peta. SIG dapat menyediakan informasi seperti objek wisata, restoran, akomodasi, dan lainnya, jalan yang menghubungkan tempat-tempat tersebut, maupun deskripsi tentang tempat dan jalan. Dengan kemampuan SIG

menggunakan Google Maps API, wisatawan mempunyai panduan yang cepat dan akurat untuk membantu perjalanan. (Minarni dkk ; 2016 : 31).

Menurut (Antonio dkk ; 2013 : 125) Industri pariwisata ini akan berkembang baik, bila pertumbuhan pengunjung wisatawan yang terus meningkat akan memberi kontribusi pendapatan ekonomi yang semakin meningkat. Pertumbuhan bisa dijamin dengan beberapa faktor yang mampu menjamin industri pariwisata yaitu ketersediaan informasi tentang pariwisata yang mudah diakses dari berbagai tempat melalui internet. Hal ini bisa terealisasi melalui teknologi sistem informasi geografis berbasis web.