BAB II
LANDASAN TEORI

2.1 Pengertian Sistem

Suatu sistem terdiri dari sejumlah elemen yang saling berinteraksi, artinya saling bekerja sama untuk mencapai suatu tujuan. suatu sistem betapapun kecilnya selalu mengandung elemen-elemen, dimana elemen-elemen dapat berupa subsistem-subsistem atau bagian dari sistem yang mempunyai sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

Dalam keseharian secara sadar atau tidak manusia terlibat dalam sebuah sistem, seperti sistem pernafasan, sistem pencernaan, dan lain sebagainya. Dibawah ini terdapat beberapa konsep dasar di dalam menentukan definisi sebuah sistem.

Pengertian sistem menurut JOG[5] adalah sebagai berikut :

> *"Suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu"*.

Berdasarkan pengertian diatas, dapat disimpulkan bahwa pengertian sistem adalah kumpulan dari elemen-elemen yang saling terkait dan bekerja sama untuk mencapai suatu tujuan tertentu.

2.2 Pengertian Informasi

Informasi sangat diperlukan dalam menunjang suatu sistem. Tanpa adanya informasi, sebuah sistem tidak dapat berjalan secara optimal dan tanpa sistem pula

pihak manajemen suatu perusahaan tidak akan dapat mengambil sebuah keputusan.

Menurut DAV[3] pengertian informasi adalah sebagai berikut :

> *"Data yang telah diolah menjadi suatu bentuk yang penting bagi si penerima dan mempunyai nilai yang nyata atau yang dapat dirasakan dalam keputusan yang sekarang atau keputusan yang akan datang"*.

Berdasarkan pengertian diatas, dapat disimpulkan bahwa pengertian informasi adalah data yang telah diolah sehingga mempunyai nilai guna dan dapat menunjang keputusan pada saat sekarang maupun keputusan yang akan datang.

2.3 Pengertian Sistem Informasi

Sebuah informasi dapat dihasilkan melalui proses dari sebuah sistem. Sistem informasi adalah sebuah sistem yang bertujuan untuk menghasilkan informasi yang diperlukan.

Menurut JOG[5] pengertian sistem informasi adalah :

> *"Suatu sistem didalam organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan"*.

Berdasarkan pengertian diatas, maka dapat disimpulkan bahwa pengertian sistem informasi adalah elemen-elemen yang saling terkait dan bekerja sama untuk menghasilkan informasi bagi suatu organisasi atau perusahaan.

2.4 Metode Pendekatan Sistem

Pada metode pengembangan sistem ini akan menggunakan *Object Oriented* (OO) yang merupakan paradigma baru dalam rekayasa *software* yang didasarkan

pada objek dan kelas. *Object oriented* merupakan metodologi terbaik dalam rekayasa *software* dan mencakup bidang aplikasi yang sangat luas. Karena luasnya cakupan *object oriented* maka terdapat beberapa hal yang membingungkan berkenaan dengan istilah dan konsep *object oriented*. Istilah dan konsep yang berkenaan dengan dengan *Object Oriented* ini adalah:

1. *Object Orinted Analysis* adalah metode analisis yang memeriksa *requirement* (syarat atau keperluan yang harus dipenuhi suatu sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam uang lingkup permasalahan.
2. *Object Orinted Design* adalah metode untuk mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek sistem atau subsistem.

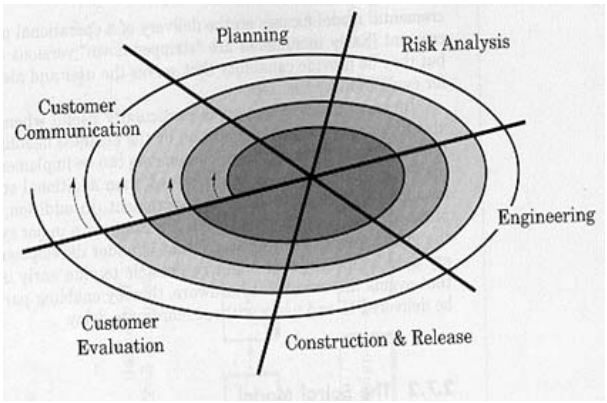
   Object Oriented mendapatkan pemahaman dasar dalam pemodelan dengan *Unified Modelling Language* (UML) dan *Rational Rose.*

2.5 Model Proses

Pada model proses ini penulis ingin merancang suatu sistem informasi yang menghasilkan perangkat lunak yang berkualitas dan tepat waktu dengan metode Spiral. Model ini mengambil fitur penting dari model *waterfall* dan *prototyping*, dengan menambah elemen baru yaitu analisa resiko (*risk analysis*). Model ini memiliki 6 aktivitas penting, yaitu:

1. *Customer Communication*; komunikasi antara pengembang dengan pelanggan.
2. *Planning*; penentuan tujuan, alternatif dan batasan.
3. *Risk Analysis*; analisa alternatif dan identifikasi atau pemecahan resiko.

4. *Engineering*; pengembangan level berikutnya dari produk.
5. *Construction and Release*; pengujian, instalasi, dan menyediakan *support* termasuk dengan pelatihan pada *user* dan pembuatan dokumentasi.
6. *Customer Evaluation*; penilaian terhadap hasil *engineering*.



Gambar 2.1 Metode Spiral
(*Sumber : Pressman, Software Engineering*)

Bentuk Spiral memberikan gambaran bahwa makin iterasinya membesar, maka menunjukkan makin lengkapnya versi dari perangkat lunak yang digunakan. Selama awal sirkuit, objektif, alternatif dan batasan didefinisikan serta resiko diidentifikasi dan dianalisa. Jika analisa resiko menunjukkan ada ketidakpastian terhadap kebutuhan, maka *prototyping* harus dibuat pada kuadran *engineering*. Simulasi dan pemodelan lain dapat digunakan untuk mendefinisikan masalah dan memperbaiki kebutuhan.

Pelanggan mengevaluasi hasil *engineering* (kuadran *customer evaluation*) dan membuat usulan untuk perbaikan. Berdasarkan masukan dari pelanggan, fase berikutnya adalah *planning* dan analisis resiko. Setelah analisis resiko, selalu

diperiksa apakah proyek diteruskan atau tidak, jika resiko terlalu besar, maka proyek dapat dihentikan.

Kelebihan metode Spiral ini adalah pendekatan yang paling realistik untuk sistem skala besar. Metode ini menggunakan pendekatan evolusioner, sehingga pelanggan dan pengembang dapat mengerti dan bereaksi terhadap suatu resiko yang mungkin terjadi. Model ini membutuhkan konsiderasi langsung terhadap resiko teknis, sehingga diharapkan dapat mengurangi terjadinya resiko yang lebih besar. Pada setiap fase evolusi, bisa digunakan *prototyping*.

Kekurangan dari metode Spiral ini mungkin akan agak sulit untuk meyakinkan pelanggan besar, bahwa pendekatan evolusioner ini dapat diatur. Hal ini membutuhkan keahlian tersendiri. Selain itu, jika resiko utama tidak ditemukan, maka masalah bisa muncul kemudian. Sehingga membutuhkan kemampuan manajemen dan perkiraan resiko (*risk assessment*) yang cukup tinggi.

2.6 Pengenalan *Unified Modeling Language* (UML)

2.6.1 Pengertian *Unified Modeling Language* (UML)

Unified Modeling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam

bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk *modeling* aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

2.6.2 Bagian-bagian UML

Bagian-bagian utama dari UML adalah *view*, *diagram*, *model element*, dan *general mechanism*. Di bawah ini adalah penjelasan mengenai bagian-bagian tersebut :

2.6.2.1 *View*

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat sebuah grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram.

Beberapa jenis *view* dalam UML antara lain :

a. *Use Case View*

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan *external actors*. *Actor* yang berinteraksi dengan sistem dapat berupa *user* atau sistem lainnya.

View ini digambarkan dalam *use case diagrams* dan kadang-kadang dengan *activity diagrams*. *View* ini digunakan terutama untuk pelanggan, perancang, pengembang, dan penguji sistem.

b. *Logical View*

Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (*class*, *object*, dan *relationship*) dan kolaborasi dinamis yang terjadi ketika *object* mengirim pesan ke *object* lain dalam suatu fungsi tertentu.

View ini digambarkan dalam *class diagrams* untuk struktur statis dan dalam *state*, *sequence*, *collaboration*, dan *activity diagram* untuk model dinamisnya.

c. *Component View*

Mendeksripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari *code module* diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan informasi administratif lainnya.

d. *Concurrency View*

Membagi sistem kedalam proses dan prosesor. *View* ini digambarkan dalam diagram dinamis (*state, sequence, collaboration,* dan *activity diagrams*) dan diagram implementasi (*component* dan *deployment diagrams*) serta digunakan untuk pengembang, pengintegrasi, dan penguji.

e. *Deployment View*

Mendeskripsikan fisik dari sistem, seperti komputer dan perangkat (*nodes*) dan bagaimana hubungannya dengan lainnya. *View* ini digambarkan dalam *deployment diagrams* dan digunakan untuk pengembang, pengintegrasi, dan penguji.

2.6.2.2 Diagram

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem.

Sebuah diagram merupakan bagian dari suatu *view* tertentu. Adapun jenis diagram antara lain :

a. *Use Case Diagram*

Menggambarkan sejumlah *external actors* dan hubungannya ke *use case* yang diberikan oleh sistem.

b. *Class Diagram*

Menggambarkan struktur statis class di dalam sistem. *Class* mempresentasikan sesuatu yang ditangani oleh sistem. Sebuah sistem biasanya mempunyai beberapa *class diagram*.

c. *State Diagram*

Mengambarkan semua *state* (kondisi) yang dimiliki oleh suatu *object* dari suatu class dan keadaan yang menyebabkan *state* berubah. Kejadian dapat berupa *object* lain yang mengirim pesan.

d. *Sequence Diagram*

Menggambarkan kolaborasi dinamis antara sejumlah *object*. Kegunaannya untuk menunjukkan rangakaian pesan yang dikirim antara object juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

e. *Collaboration Diagram*

Menggambarkan kolaborasi dinamis seperti *sequence diagrams*. Dalam menunjukkan pertukaran pesan, *collaboration diagram* menggambarkan *object* dan hubungannya (mengacu ke konteks).

f. *Activity Diagram*

Menggambarkan rangkaian aliran dari aktifitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

g. *Component Diagram*

Menggambarkan struktur fisik kode dari *component*. *Component* dapat berupa *source code*, *component biner*, atau *executable component*.

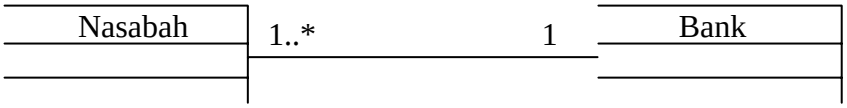
h. *Deployment Diagram*

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem. Diagram ini menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya.

2.6.3 Indikator *Multiplicity*

Indikator *Multiplicity* ditentukan untuk *class*, *multiplicity* menentukan banyaknya *object* yang terlibat dalam relasi. *Multiplicity* menentukan banyaknya *object* yang terhubung satu dengan yang lainnya. Indikator *multiplicity* terdapat pada masing-masing akhir garis relasi, baik pada asosiasi maupun agregasi. Beberapa contoh *multiplicity* berserta gambar adalah :

1	Tepat waktu
0..*	Nol atau lebih
1..*	Satu atau lebih
0..1	Nol atau satu
5..8	range 5 s.d. 8
4..6,9	range 4 s.d. 6 dan 9



Gambar 2.2

Indikator *Multiplicity*

2.7 Basis Data

Basis data adalah sekumpulan informasi yang diatur dalam cara tertentu sehingga sebuah program komputer dapat dengan cepat memilih data yang diinginkan. Basis data dapat diibaratkan sebagai sistem pengarsipan elektronis.

Basis data tradisional terdiri dari :

- *field*, yaitu item tertentu dari informasi
- *record*, yaitu sekumpulan *field*
- *file*, yaitu sekumpulan *record*.

Untuk mengakses informasi dari basis data diperlukan *Database Management System* (DBMS) yang merupakan sekumpulan program yang memungkinkan pengguna memasukkan, mengatur, atau memilih data dari basis data.

2.8 Objek Yang Berhubungan Dengan *Web*

Dibawah ini adalah beberapa objek yang biasanya berhubungan atau terdapat dalam *internet* :

a. *World Wide Web* (WWW)

World Wide Web atau lebih dikenal dengan web adalah suatu kumpulan dari informasi-informasi yang terdapat pada komputer yang terkoneksi ke *internet* dari seluruh penjuru dunia. Informasi yang ada dapat berupa teks, grafik atau gambar, suara dan juga video. Perbedaan antara *web* dengan berbagai tipe bentuk *internet* adalah bahwa *web* memakai suatu pendekatan untuk merangkai materi-materi yang kita

kenal dengan *Hyperlink*. Jika *link*nya berupa teks maka teks tersebut dinamai *Hyperteks*.

b. *Browser*

Browser adalah antarmuka antara pemakai dan *World Wide Web* (WWW) yang menginterpretasikan *hypertext link* dan digunakan untuk melihat dan memandu dari satu simpul *internet* ke yang lain.

c. *Homepage*

Homepage adalah halaman pertama sebuah situs *web*. *Homepage* akan memandu pengguna membuka halaman-halaman lain pada situs tersebut dengan mengikuti informasi atau *link* yang ada.

d. *Hypertext Transfer Protocol* (HTTP)

Hypertext transfer protocol adalah bahasa yang terstruktur yang dirancang untuk menandai dokumen yang dikirim pada *internet* dengan mengikuti aturan-aturan penulisan *tags*. Protokol HTTP bersifat *request – response*, yaitu *client* menyampaikan *request* ke *server* dan *server* memberikan *response* yang sesuai dengan *request* tersebut.

2.9 *Internet*

Pada dasarnya *internet* adalah suatu jaringan komputer yang besar, dimana masing-masing komputer saling berhubungan dan saling berbagi TCP/IP *protocol*. Istilah *internet* berasal dari kata *internetworking* yang digunakan untuk menunjukan usaha menghubungkan komputer dari sistem operasi yang berbeda,

termasuk didalamnya sistem jaringan yang berbeda. *Internet* berfungsi sebagai pengantar informasi yang berada pada komputer lain.

Internet merupakan jaringan dari jaringan, dengan *internet* dibangun dari kesimpulan jaringan. Jadi, penjelasan jaringan merupakan penjalanan tentang dasar dibangunnya *internet*. Jaringan dibangun dari dua atau lebih komputer yang terhubung agar masing-masing komputer yang terhubung dapat saling berbagi informasi bahkan berkomunikasi secara elektronik.

Dibawah ini adalah fasilitas-fasilitas yang sering terdapat didalam *internet,* yaitu :

a. *E-mail*

   E-mail (*electronic mail*) adalah pesan elektronik yang dikirim dari komputer seorang pengguna ke komputer lainnya. *E-mail* dapat dikirimkan melalui *Local Area Network* (LAN) atau *internet.*

b. *File Transfer Protocol* (FTP)

   FTP adalah protokol *internet* yang digunakan untuk menyalin *file* antar komputer.

c. *Telnet*

   Telnet adalah protokol yang digunakan untuk menghubungi *host* yang berada jauh dari sebuah terminal. Koneksi ke *host* dari sebuah terminal yang menggunakan telnet dilakukan melalui *internet.*

2.10 Perangkat Lunak Pendukung

2.10.1 *Hypertext Markup Language* (HTML)

Hypertext Markup Language (HTML) adalah sekumpulan perintah yang terformat yang digunakan untuk membuat halaman dokumen *web*. Ketika membuka sebuah halaman *web*, maka *browser* akan menginterpretasikan perintah HTML pada halaman tersebut dan kedalam bentuk teks dan grafik.

Struktur sebuah dokumen HTML pada dasarnya dibagi menjadi dua bagian besar, yaitu *header* dan *body*. Masing-masing ditandai oleh pasangan container tag <head> dan <body>. Bagian *head* berisi judul dokumen dan informasi dasar lainnya, sedangkan bagian *body* adalah data dokumennya.

2.10.2 PHP

PHP adalah singkatan dari *Personal Home Page Tools*, adalah sebuah bahasa pemograman yang dibundel dengan HTML yang dijalankan di sisi *server*. Sebagian besar perintahnya berasal dari bahasa C, Java dan Perl dengan beberapa tambahan khusus fungsi PHP.

PHP mulai diperkenalkan pada tahun 1984 oleh Rasmus Lerdorf. Dia menggunakan PHP untuk membuat situs pribadinya yang dapat mencatat siapa saja yang mengakses situs pribadinya tersebut.

Yang membedakan PHP dengan bahasa pemograman lain adalah adanya *tag* penentu, yaitu diawali dengan “<?” atau “<?PHP” dan diakhiri dengan “?>”. Jadi, kita bebas menempatkan skrip PHP dimanapun dalam dokumen HTML yang telah kita buat.

2.10.3 MySQL *Database*

2.10.3.1 Sejarah Singkat MySQL

SQL (*Structured Query Language*) adalah bahasa *query* yang distandarisasi untuk meminta informasi dari sebuah basis data. Versi awal SQL adalah SEQUEL (*Structured English Query Language*) yang dirancang oleh IBM pada tahun 1974 dan 1975.

SQL pertama kali diperkenalkan sebagai sistem basis data komersial pada tahun 1979 oleh Oracle Corporation. Awalnya SQL merupakan bahasa *query* untuk sistem manajemen basis data yang berjalan pada *minicomputer* dan *mainframe*. Namun sekarang SQL juga dapat digunakan pada sistem manajemen basisdata PC karena mendukung basis data tersebar (*distributed database*).

2.10.3.2 Keistimewaan MySQL

SQL mempunyai beberapa kelebihan-kelebihan untuk *programmer* dan *database administrator* yaitu :

- dapat mengubah struktur basis data
- dapat mengubah pengaturan keamanan sistem
- memberikan akses kepada pengguna untuk mengakses sebuah basis data atau *table*
- memperoleh informasi dari basis data
- memutakhirkan isi basis data.

Perintah-perintah dalam SQL dikelompokkan menjadi lima (5) macam :

1. *Data Definition Language* (DDL)

   Adalah perintah SQL yang menjelaskan objek dari basis data. Dengan kata lain DDL digunakan untuk mendefinisikan kerangka basis data. Perintah-perintahnya seperti *create, alter, drop,* dan lain-lain.

2. *Data Manipulation Language* (DML)

   Adalah perintah SQL yang digunakan untuk memanipulasi atau mengoperasikan isi basis data. SQL menyediakan 4 perintah DML yaitu *select, insert, delete*, dan *update*.

3. *Security*

   Adalah perintah SQL yang digunakan untuk menjamin keamanan data, antara lain terdiri atas *grant* dan *revoke*.

4. *Integrity*

   Adalah perintah SQL yang digunakan untuk menjaga kesatuan data. Perintahnya antara lain *recover table*.

5. *Auxiliary*

   Adalah perintah-perintah tambahan atau pelengkap seperti *rename* dan *unload*.

2.10.4 Apache

Apache merupakan *web server* yang paling banyak digunakan di *internet*. Ini disebabkan oleh beberapa faktor seperti kecepatan, performansi, dan tentu saja harganya yang gratis.

Apache merupakan turunan dari *web server* yang dikeluarkan oleh NCSA, yaitu NCSA HTTPD pada sekitar tahun 1999-an. Pada dasarnya apache adalah "*A Patchy* (*patch*)" dan pengganti dari NCSA HTTPD.

2.10.5 Macromedia Dreamweaver MX

Macromedia Dreamweaver MX adalah HTML editor profesional yang berfungsi mendesain, melakukan *coding*, dan mengembangkan *website* yang paling terkenal di dunia *web*. Macromedia adalah produsen *software website* yang paling besar di dunia. Selain Dreamweaver, macromedia juga mengeluarkan Macromedia Flash, Fireworks, Coldfusion, Freehand, Director, dan lain sebagainya.

Salah satu kelebihan Dreamweaver adalah ruang kerja Dreamweaver beserta *tool* yang tersedia dapat digunakan dengan mudah. Sehingga anda bisa membangun suatu website dengan cepat. Selain itu juga mempunyai integrasi dengan produk Macromedia lainnya seperti Flash dan Fireworks. Flash sudah sangat terkenal sebagai program untuk membuat animasi yang berbasiskan *web*.

Fungsi coding Dreamweaver tidak hanya mendukung coding HTML, tapi juga CSS (*Cascading Style Sheet*), Javascript, Coldfusion, ASP (*Active Server Page*), JSP (*Java Server Page*), dan Dreamweaver juga memungkinkan anda membangun website dengan server yang berhubungannya CFML (Colfusion Markup Language), ASP, net dan PHP.

2.11 Pengertian Sistem Informasi Pariwisata

Pengertian pariwisata menurut POE[6] adalah :

"Perjalanan dari suatu tempat ke tempat lain, bersifat sementara, dilakukan perorangan maupun kelompok, sebagai usaha untuk mencari keseimbangan atau keserasian dan kebahagian dengan lingkungan hidup dalam dimensi sosial, budaya, alam, dan ilmu".

Berdasarkan pengertian di atas, dapat disimpulkan bahwa pengertian sistem informasi pariwisata adalah kumpulan elemen-elemen yang saling terkait dan bekerja sama untuk menampilkan informasi mengenai kepariwisataan, dalam hal ini mengenai objek wisata, budaya, dan fasilitas pendukung lainnya.