

BAB II

LANDASAN TEORI

2.1 Pengertian Sistem

(Jogiyanto HM, 2003) Sistem (*System*) dapat didefinisikan dengan pendekatan prosedur dan dengan pendekatan komponen. Dengan pendekatan prosedur, sistem dapat didefinisikan sebagai kumpulan dari prosedur-prosedur yang mempunyai tujuan tertentu. Contoh sistem yang didefinisikan dengan pendekatan prosedur ini adalah sistem akuntansi. Sistem ini didefinisikan sebagai kumpulan dari prosedur-prosedur penerimaan kas, pengeluaran kas, penjualan, pembelian, dan buku besar.

Dengan pendekatan komponen, sistem dapat didefinisikan sebagai kumpulan komponen yang saling berhubungan satu dengan yang lainnya, membentuk satu kesatuan untuk mencapai tujuan tertentu. Contoh dari sistem yang didefinisikan dengan pendekatan ini misalnya adalah sistem komputer yang didefinisikan sebagai kumpulan dari perangkat keras dan perangkat lunak.

2.2 Pengertian Sistem Informasi

(Abdul Kadir, 2014:08) Sesungguhnya, yang dimaksud dengan sistem informasi tidak harus melibatkan komputer. Sistem informasi yang menggunakan komputer biasa disebut sistem informasi berbasis komputer. Dalam praktik, istilah sistem informasi lebih sering dipakai tanpa embel-embel berbasis komputer walaupun dalam kenyataannya komputer merupakan bagian yang penting. Ada beragam definisi sistem informasi sebagai berikut:

- a. Alter (1992), Sistem informasi adalah kombinasi antar prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.

- b. Bodnar dan Hopwood (1993), Sistem informasi adalah kumpulan perangkat lunak dan keras yang dirancang untuk mentransformasikan data kedalam bentuk informasi yang berguna.
- c. Gelinas, Oram, dan Wiggins (1990), Sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan, dan mengolah data serta menyediakan informasi keluaran kepada pemakai.
- d. Hall (2001), Sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi dan didistribusikan kepada pemakai.
- e. Turban, McLean, Wetherbe (1999), Sebuah sistem informasi mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan yang spesifik.
- f. Wilkinson (1992), Sistem informasi adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan (input) menjadi keluaran (informasi), guna mencapai sasaran-sasaran perusahaan.

2.3 Pengertian Sistem Informasi Geografis

(Dr. Indarto, 2013:3) Sistem Informasi Geografis (SIG) atau lebih dikenal dengan istilah *Geographical Information System (GIS)* didefinisikan sebagai suatu alat atau media untuk memasukan, menyimpan, mengambil, memanipulasi, menganalisis dan menampilkan data-data beratribut geografis (data geospasial) yang berguna untuk *review* mendukung proses pengambilan keputusan dalam

perencanaan dan manajemen sumber daya alam, lingkungan, transportasi, masalah perkotaan dan administratif (*Burrough. 1986*).

SIG juga didefinisikan sebagai suatu sistem *database* dimana hampir semua data terstruktur secara geospasial dan adanya suatu prosedur yang bekerja untuk memberikan informasi tentang suatu objek *database* geospasial didalam *database* tersebut (*Smith et at., 1987*).

Arnof (1989) mendefinisikan SIG sebagai sebuah sistem komputer yang menyediakan empat kemampuan utama untuk menangani data yang telah tergeoreferensi, meliputi proses pemasukan data, manajemen data (menyimpan dan pemanggilan kembali), manipulasi dan analisis data, dan proses penampilan data.

2.3.1 Komponen Sistem Informasi Geografis

Pada prinsipnya SIG mempunyai 3 komponen utama yakni: Sistem Komputer, Data dan Pengguna (*user*). Jadi, SIG merupakan satu kesatuan sistem termasuk: Perangkat keras (*hardware*), Data, Perangkat lunak (*software*) dan pengguna yang mengaplikasikan SIG untuk menyelesaikan suatu permasalahan dalam bidang tertentu.

2.3.2 Fungsi Utama Sistem Informasi Geografis

Pada prinsipnya hampir semua bidang kehidupan membutuhkan dari suatu SIG. untuk Membantu permasalahan. Pengguna (*user*) akan menentukan informasi yang dibutuhkan SIG dan standar yang dibutuhkan. Memilih cara memperbaharui yang paling efisien dan menganalisa output implementasi ke suatu permasalahan.

2.3.2.1 Mengoleksi Data

Data yang digunakan di dalam SIG sering berasal dari berbagai tipe dan disimpan dengan cara yang berbeda. menyediakan alat dan metode untuk mengintegrasikan data-data yang berbeda tersebut ke dalam sebuah format, sehingga data-data tersebut mudah untuk dibandingkan dan dianalisa.

2.3.2.2 Memperbarui dan mengelola *database*

Setelah data dikoleksi dan di integrasikan, SIG seharusnya mampu untuk menyediakan fasilitas untuk menambahkan dan memelihara data. Manajemen data yang efektif memiliki arti yang cukup luas, yang mencakup aspek keamanan, integrasi, penyimpanan dan pencarian data, serta kemampuan untuk pemeliharaan.

2.3.2.3 Analisa Geografis

Integrasi dan konversi data merupakan salah satu bagian dari tahap pemasukan data didalam SIG. langkah yang dibutuhkan selanjutnya adalah interpretasi dan analisa koleksi informasi tersebut secara kuantitatif dan kualitatif. Sebagai contoh, citra satelit dapat membantu pakar pertanian untuk memperkirakan pertumbuhan tanaman perhektar dalam satuan luas tertentu.

2.3.2.4 Menampilkan/Mempresentasikan Hasil

Salah satu aspek yang menarik dari Teknologi SIG adalah bahwa informasi yang beragam dapat ditampilkan sekaligus dalam suatu bidang yang sama.

2.4 Metode Pengembangan Sistem

(Adi Nugroho, 2010:1) USDP (*Unified Software Development Process*) salah satu metode pengembangan sistem / perangkat lunak yang menggunakan UML (*Unified Modeling Language*) sebagai *tool* utamanya dengan tahapan yaitu :

2.3.1 Perencanaan (*Planning*)

Studi tentang kebutuhan pengguna (*user's specification*), studi-studi kelayakan (*feasibility study*) baik secara teknis maupun secara teknologi serta penjadwalan pengembangan suatu proyek sistem informasi dan atau perangkat lunak.

2.3.2 Analisis (*Analysis*)

Tahap menggali permasalahan yang muncul pada pengguna dengan mendekomposisi dan merealisasikan *use case* diagram lebih lanjut, mengenali komponen-komponen sistem, objek-objek, hubungan antar objek, dan sebagainya.

2.3.3 Perancangan (*Design*)

Mencari solusi permasalahan yang di dapat dari tahap analisis, pada tahap ini dibagi menjadi dua yaitu :

1. Tahap perancangan yang lebih menekankan pada *platform* apa hasil dari tahap analisis yang akan di implementasikan.
2. Tahap perancangan yang dimana melakukan penghalusan (*refinement*) kelas-kelas yang di dapat pada tahap analisis serta menambahkan dan memodifikasi kelas-kelas yang akan lebih mengefisienkan serta mengefektifkan sistem/perangkat lunak yang akan dikembangkan.

2.3.4 Implementasi

melakukan penyesuaian setting perangkat lunak agar bisa dipakai di sisi pengguna (misal, install dan setting database di server pengguna, penyesuaian setting IP) dan melakukan perbaikan coding yang ditemukan selama beta testing.

2.3.5 Pengujian (*Testing*)

Pada tahap ini digunakan untuk menentukan apakah sistem yang di buat sudah sesuai dengan kebutuhan pengguna atau belum. Jika belum, proses selanjutnya adalah bersifat *interaktif* yaitu kembali ke tahap sebelumnya.

2.5 UML (*Unified Modeling Language*)

(Adi Nugroho, 2010:6) Bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan yang kompleks sehingga lebih mudah untuk dipelajari dan dipahami.

Tujuan pemodelan dalam kerangka pengembangan sistem adalah sebagai sarana analisis, pemahaman, visualisasi, dan komunikasi antar tim pengembang yang beranggotakan beberapa/banyak anggota. Beberapa diagram dalam UML yang akan digunakan dalam membantu pengembangan sistem yaitu :

2.5.1 Use Case Diagram

Merupakan unit koheren dari fungsionalitas sistem yang tampak dari luar dan diekspresikan sebagai urutan pesan-pesan yang dipertukarkan unit-unit sistem dengan satu atau lebih *actor* yang sda diluar sistem. Kegunaan *use*

case sesungguhnya adalah untuk mendefinisikan suatu bagian perilaku sistem yang bersifat koheren tanpa perlu menyiapkan struktur internal sistem yang sedang dikembangkan. Definisi *use case* di dalamnya mencakup semua perilaku yang ada dalam sistem yang sedang kita kembangkan. Simbol dan keterangan *use case diagram* seperti pada tabel 2.1.

Tabel 2.1 Simbol dan Keterangan *Use Case Diagram*

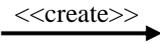
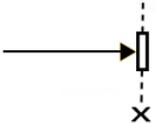
SIMBOL	KETERANGAN
<p data-bbox="501 846 580 875"><i>Actor</i></p>  <p data-bbox="496 1043 585 1068">NewClass</p>	<p data-bbox="700 857 1278 1003">Mewakili peran orang, <i>system</i> yang lain atau alat ketika berkomunikasi dengan <i>use case</i>.</p>
<p data-bbox="485 1111 600 1140"><i>Use case</i></p>  <p data-bbox="485 1258 600 1283">NewUseCase</p>	<p data-bbox="700 1097 1278 1182">Abstraksi dari interaksi antara <i>system</i> dan <i>actor</i>.</p>
<p data-bbox="461 1357 624 1386">Generalisasi</p> 	<p data-bbox="700 1344 1278 1429">Menunjukkan spesialisasi <i>actor</i> untuk dapat berpartisipasi dalam <i>use case</i>.</p>
<p data-bbox="496 1547 588 1576"><i>Extend</i></p> 	<p data-bbox="700 1534 1278 1677">Mempesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.</p>
<p data-bbox="493 1744 592 1774"><i>Include</i></p> 	<p data-bbox="700 1731 1278 1874">Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.</p>

2.5.2 *Sequence Diagram*

Sequence Diagram menekankan pada urutan waktu penerimaan *message*, kita menjumpai garis hidup objek yaitu garis tegas *vertical* yang mencerminkan ekstensi sebuah objek sepanjang periode waktu. Kebanyakan objek yang hadir pada *interaction diagram* akan eksis sepanjang durasi tertentu dari interaksi, sehingga objek itu diletakkan di atas diagram dengan ‘garis hidup’ digambarkan dari atas hingga ke bagian bawah diagram. Simbol dan keterangan *sequence diagram* seperti pada tabel 2.2.

Tabel 2.2 Simbol dan Keterangan *Sequence Diagram*

SIMBOL	KETERANGAN
Objek 	Berpartisipasi secara berurutan dengan mengirimkan atau menerima pesan dan ditempatkan di bagian atas diagram.
Garis hidup objek 	Menandakan kehidupan obyek selama urutan dan diakhiri tanda X pada titik dimana kelas tidak lagi berinteraksi.
Waktu aktif 	menandakan ketika suatu objek mengirim atau menerima pesan.
Pesan 	Objek mengirim satu pesan ke objek lainnya.

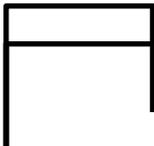
<p><i>Create</i></p> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
<p><i>Destroy</i></p> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah objek yang diakhiri.

2.5.3 Activity Diagram

Activity Diagram atau Diagram Aktivitas menggambarkan alur aktivitas dalam aplikasi, menjelaskan proses masing-masing alur berawal dan proses aplikasi berakhir. Diagram aktivitas juga menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Simbol dan keterangan *activity diagram* seperti pada tabel 2.3.

Tabel 2.3 Simbol dan Keterangan *Activity Diagram*

SIMBOL	KETERANGAN
<p>Status Awal</p> 	Status awal aktivitas <i>system</i> , sebuah diagram aktivitas memiliki sebuah status awal.
<p>Aktivitas</p> 	Aktivitas yang dilakukan <i>system</i> , aktivitas biasanya diawali dengan kata kerja.
<p>Perancangan</p> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.

<p>Status akhir</p> 	<p>Status akhir yang dilakukan <i>system</i>, sebuah diagram aktivitas memiliki sebuah status akhir.</p>
<p>Action</p> 	<p>Memperlihatkan masing-masing kelas antar muka saling berinteraksi satu sama lain.</p>
<p>Fork Node</p> 	<p>Satu aliran pada tahap tertentu berubah menjadi beberapa aliran.</p>
<p>Swimlane</p> 	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.</p>

2.5.4 Class Diagram

Class Diagram atau Diagram Kelas merupakan diagram yang memodelkan sekumpulan kelas, *interface*, kolaborasi dan relasinya. Diagram kelas digambarkan dengan bentuk kotak. Simbol dan keterangan *class diagram* seperti pada tabel 2.4.

Tabel 2.4 Simbol dan Keterangan *Class Diagram*

SIMBOL	KETERANGAN
<p>Class</p> 	<p>Himpunan dari objek-objek yang berbagai atribut serta operasi yang sama.</p>
<p>Nary Association</p> 	<p>Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.</p>

<i>Generalization</i> <hr/>	Hubungan dimana objek anak (<i>descendent</i>) berbagai perilaku dan struktur data dari objek yang ada di atasnya objek induk(<i>oncestor</i>).
<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.

2.6 Google Maps

Menurut *Wikipedia*, *Google Maps* adalah layanan pemetaan *web* yang dikembangkan oleh *Google*. Layanan ini memberikan citra satelit, peta jalan, panorama 360°, kondisi lalu lintas, dan perencanaan rute untuk berpergian dengan berjalan kaki, mobil, sepeda (versi beta), atau angkutan umum. *Google Maps* dimulai sebagai program *desktop* C++, dirancang oleh Lars dan Jens Eilstrup Rasmussen pada *Where 2 Technologies*. Pada Oktober 2004, perusahaan ini diakuisisi oleh *Google*, yang diubah menjadi sebuah aplikasi *web*. Setelah akuisisi tambahan dari perusahaan visualisasi data geospasial dan analisis lalu lintas, *Google Maps* diluncurkan pada Februari 2005. Layanan ini menggunakan *Javascript*, *XML*, dan *AJAX*. *Google Maps* untuk seluler dirilis pada bulan September 2008. Pada Agustus 2013, *Google Maps* bertekad untuk menjadi aplikasi yang paling populer di dunia untuk ponsel cerdas, dengan lebih dari 54% dari pemilik ponsel cerdas di seluruh dunia menggunakannya setidaknya sekali.

2.6.1 Google Maps API

Google Maps menawarkan *API* yang memungkinkan peta untuk dimasukkan pada situs *web* pihak ketiga, dan menawarkan penunjuk lokasi

untuk bisnis perkotaan dan organisasi lainnya di berbagai negara di seluruh dunia. *Google Map Maker* memungkinkan pengguna untuk bersama-sama mengembangkan dan memperbarui pemetaan layanan di seluruh dunia.

2.7 PHP

(Abdul Kadir, 2001) Menurut Dokumen Resmi *PHP*, *PHP* singkatan dari *PHP Hypertext preprocessor*, merupakan bahasa berbentuk scrip yang ditempatkan didalam *server* dan diproses di *server*. Hasilnyalah yang dikirim ke *clien*, tempat pemakai menggunakan *browser*.

Secara khusus *PHP* dirancang untuk membentuk *web* dimanis. Artinya ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnya anda biasa menampilkan isi *database* ke halaman *Web*. Pada prinsipnya, *PHP* mempunyai fungsi yang sama dengan Scrip-scrip seperti *ASP (Active Server Page)*, *Cold Fusion*, ataupun *Perl*.

PHP bersifat bebas pakai, artinya tidak perlu membayar apapun untuk menggunakan perangkat lunak ini. Anda dapat mendownloadnya melalui situs www.php.net

2.8 HTML (Hyper Text Markup Language)

(Bethar Ir, Husni I Pohan, 2005) *HTML* adalah kependekan dari *Hyper Text Markup Language*. Dokumen *HTML* adalah file *text* murni yang dapat dibuat dengan *editor text* sembarang. Dokumen ini dikenal sebagai *web page*. Dokumen *HTML* merupakan dokumen yang disajikan dalam *browser web surfer*. Dokumen ini umumnya berisi informasi atau pun *interface* aplikasi di dalam internet. Ada dua cara untuk membuat sebuah *web page*: dengan *HTML editor* atau dengan *editor teks* biasa (misalnya *notepad*).

2.9 MySQL

(Bunafit Nugroho, 2004:1) *MySQL* adalah sebuah program *database* server yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi *user*, serta menggunakan perintah standar *SQL* (*Structured query language*).

MySQL merupakan sebuah *database server* yang *free*, artinya kita bebas menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya.

Database MySQL, biasa kita baca **mai-es-ki-el** merupakan suatu perangkat lunak *database* yang berbentuk *database* relasional atau dalam bahasa basis data biasa kita sebut dengan *Relation Database Management system* (*RDBMS*) yang menggunakan suatu bahasa permintaan bernama *SQL*. *MySQL* memiliki beberapa kelebihan, diantaranya adalah:

- *MySQL* sebagai *Database Management System* (*DBMS*)
- *MySQL* sebagai *Relation Database Management System* (*RDBMS*)
- *MySQL* adalah sebuah software *database* yang *opensource*, artinya program ini bersifat *free* atau bebas digunakan oleh siapa saja tanpa harus membeli dan membayar lisensi kepada pembuatnya.

2.10 XAMPP

Menurut *Wikipedia*, *XAMPP* ialah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program. Yang mempunyai fungsi sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri dari program *MySQL database*, *Apache HTTP Server*, dan penerjemah ditulis dalam bahasa pemrograman *PHP* dan *Perl*.

Nama *XAMPP* merupakan singkatan dari X (empat sistem operasi), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia di bawah *GNU* (*General Public*

License) dan bebas, adalah mudah untuk menggunakan *web server* yang dapat melayani tampilan halaman *web* yang dinamis.

2.11 Sublime Text

Menurut *Wikipedia*, *Sublime text* merupakan sebuah teks editor berbasis *Python* yang mempunyai banyak fitur – fitur. *Sublime text* juga mendukung banyak bahasa pemrograman diantaranya *C++*, *C*, *CSS*, *PHP*, *HTML*, *Javascript*, *ASP*, dan masih banyak lagi. *Sublime text* mempunyai beberapa keunggulan – keunggulan yang dapat membantu pengguna dalam membuat sebuah *web development*. Keunggulan-keunggulan yang dimiliki *Sublime Text* :

1. Multiple Selection

Multiple selection mempunyai fungsi untuk membuat perubahan pada sebuah kode pada waktu yang sama dan dalam baris yang berbeda. *Multiple selection* ini juga merupakan salah satu fitur unggulan dari *Sublime Text 3*. Kita dapat meletakkan kursor pada kode yang akan di ubah / edit, lalu tekan *Ctrl + click* atau blok kode yang akan diubah kemudian *Ctrl + D* setelah itu kita dapat merubah kode secara bersamaan.

2. Drag & Drop

Dalam teks editor ini pengguna dapat menyeret dan melepas file teks ke dalam editor yang akan membuka *tab* baru secara otomatis.

3. Multi Platform

Sublime text juga mempunyai keunggulan dalam bagian platform. *Sublime text* sendiri sudah tersedia dalam berbagai platform sistem operasi, yaitu *Windows*, *Linux*, dan *MacOS*.

2.12 Aplikasi Sistem Informasi Geografis Tentang Kos-Kosan

Menggunakan *Google Maps Api 3*.

Jurnal ini merupakan referensi dalam pembuatan Skripsi ini. Didalam jurnal ini diterangkan bagaimana cara membangun Sistem Informasi Geografis Kos-Kosan menggunakan *Google Maps Api*. Bahan bahan yang digunakan didalam jurnal ini juga tidak jauh berbeda dengan bahan-bahan yang digunakan dalam penelitian ini, mulai dari *PHP, MySQL, HTML* dan lain lain.