

BAB II

LANDASAN TEORI

2.1 Analisis dan Perancangan Sistem

2.1.1 Analisis Sistem

Menurut Whitten (2004) analisis sistem adalah sebuah teknik pemecahan masalah yang memecahkan sebuah sistem menjadi komponen-komponen untuk tujuan pembelajaran bagaimana komponen-komponen tersebut bekerja dan berinteraksi untuk mencapai tujuannya.

Menurut Jogiyanto (1999), di dalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analisis sistem, yaitu:

- a. *Identify* yaitu mengidentifikasi masalah.
- b. *Understand* yaitu memahami kerja dari sistem yang ada.
- c. *Analyze* yaitu menganalisis sistem.
- d. *Report* yaitu membuat laporan hasil analisis.

2.1.2 Perancangan Sistem

Menurut Whitten (2004) perancangan sistem adalah teknik komplementer pemecahan masalah (yang bekerjasama dengan sistem analisis) yang menyusun kembali komponen-komponen sebuah sistem kembali ke sistem yang utuh dengan harapan menghasilkan sistem yang lebih baik. Teknik ini dapat melibatkan penjumlahan, penghapusan dan perubahan komponen-komponen terhadap sistem sebelumnya.

Rancangan sistem adalah proses mengidentifikasi proses-proses dan data yang diperlukan oleh sistem baru. Jika sistem yang dirancang adalah sistem

berbasis komputer, perancangan dapat menyertakan spesifikasi jenis peralatan yang digunakan (McLeod, 2001).

Rancangan sistem terdiri dari dua kelompok, yaitu:

a. Sistem konseptual

Perancangan dibuat berdasarkan kebutuhan user dan dibuat kerangka kerja untuk penerapannya.

b. Sistem fisik

Perancangan dibuat berdasarkan rancangan, kemudian dibuat spesifikasi secara terperinci, yang nantinya dapat dipergunakan untuk pembuatan dan pengetesan program.

2.2 Internet

Menurut Turban (2005), internet merupakan sistem jaringan komputer dan jaringan dari banyak jaringan yang meliputi seluruh dunia. Internet menggunakan standar *Internet Protocol Suite* (TCP/IP). Internet mengandung banyak layanan dan sumber daya informasi, sebagai contoh *hypertext document* yang digunakan pada *World Wide Web* (WWW) dan kemampuannya dalam support surat elektronik (email).

Komunikasi secara tradisional sekarang mulai tergantikan oleh internet fungsinya, contoh :

- a. Telepon yang digantikan oleh VoIP (*Voice over Internet Protocol*).
- b. Koran, buku dan dokumen lain yang digantikan dengan halaman web, blog, dan bahkan feed.
- c. Rapat yang dahulu harus datang sekarang menggunakan *Video Conference*.
- d. Surat yang tergantikan dengan email.

Kelebihan internet di era sekarang ini :

1. Biaya yang relatif murah (dan lebih murah bila dibandingkan dengan cara tradisional).
2. Menghilangkan batasan jarak dan waktu.
Informasi yang didapatkan *real time*.

2.3 World Wide Web (WWW)

Menurut Turban (2005), *World Wide Web* adalah aplikasi yang digunakan dalam internet yang berfungsi sebagai transportasi data yang diterima sebagai start untuk menyimpan, menerima, *formatting* dan menampilkan informasi melalui *client-server architecture*.

Web dibagi menjadi 2 yaitu web statis dan web dinamis.

1. Web statis

Web statis adalah web yang contentnya dikirimkan ke user sama dengan yang disimpan di server. Pada web ini sama sekali tidak ada perubahan, berbandingterbalik dengan web dinamis yang dihasilkan dari aplikasi web server.

2. Web dinamis

Web dinamis adalah web yang contentnya dihasilkan dari hasil output dari web server. Tidak seperti web statis yang contentnya tidak dapat berubah-ubah, web dinamis dapat berubah-ubah sesuai dengan informasi terakhir yang ada di server. Web dinamis dibagi menjadi dua yaitu :

a. Server side

Web dinamis dengan metode server side berjalan dengan kode program berjalan di server. Contoh : PHP, ASP, JSP, dan lain-lain. Server side memiliki kelebihan yaitu kode program yang tidak diketahui oleh pengguna. Sedangkan kelemahannya adalah kinerja server yang berat.

b. Client Side

Web dinamis dengan metode *clientside* berjalan dengan kode program berjalan di *client*. Contoh : Javascript. *Clientside* memiliki kelebihan yaitu kode program dieksekusi di komputer pengguna sehingga mengurangi beban kerja server. Sedangkan kelemahannya adalah kode program dapat dibaca oleh pengguna.

2.4 Pedagang Kaki Lima

Pedagang Kaki lima merupakan salah satu usaha dalam perdagangan atau salah satu wujud sektor informal. Pedagang kaki lima adalah orang yang dengan modal yang relatif sedikit berusaha di bidang produksi dan penjualan barang-barang (jasa-jasa) untuk memenuhi kebutuhan kelompok tertentu di dalam masyarakat, usaha tersebut dilaksanakan pada tempat-tempat yang dianggap strategis dalam suasana lingkungan informal (Haryono, 1989). Dalam bidang perdagangan, PKL dapat dikategorikan dalam kategori terakhir yaitu popular atau community sector. Kegiatan usaha Pedagang Kaki Lima merupakan bidang usaha informal, dapat dikatakan tidak resmi atau ilegal dan merupakan kegiatan usaha yang sederhana. Dilihat dari kriteria Operasional yang ada sekarang, pengertian PKL terbagi 2 macam, yaitu:

1. PKL Tersentra

Yaitu Pedagang kaki Lima yang dalam usahanya sehari-hari menempati lokasi yang telah sesuai atau diijinkan oleh pemerintah kota. Bila di kota Surabaya diijinkan oleh Pemkot Surabaya dan tempatnya sudah berbentuk *foodcourt*.

2. PKL Binaan

Yaitu Pedagang Kaki Lima yang dalam usahanya sehari-hari menempati lokasi yang telah sesuai dan menggunakan tenda-tenda sebagai tempat dagangannya, namun keberadaannya selalu diawasi, dibina dan diarahkan untuk menjadi PKL yang baik.

Terdapat 2 sisi yang berbeda dalam keberadaan PKL ini yang selalu mengundang perdebatan yaitu sisi positif dan negatif. Yang negatif yaitu bahwa dengan keberadaan PKL ini dapat merusak atau merubah tatanan keruangan kota, perubahan fungsi tempat atau ruang publik kota, merusak citra kota sehingga menjadikan pola struktur kawasan kota yang sudah direncanakan menjadi berubah. Sedangkan sisi positifnya adalah keberadaan PKL mempunyai fungsi sosial dan ekonomi, yaitu :

- a. Membuka lapangan kerja dan usaha baru
- b. Meningkatkan penghasilan bagi rakyat kecil
- c. Terciptanya nodes atau kawasan komersial
- d. Memberikan *income* bagi pemerintah dengan adanya retribusi
- e. Menciptakan kontak sosial antar masyarakat Penyebaran PKL dipengaruhi oleh sifat dan jenis komoditi yang diperdagangkan menurut kebutuhan konsumen dan kebutuhan PKL.

Kebutuhan Konsumen Mengakibatkan kecenderungan untuk berkelompok bagi PKL dan biasanya sudah menjadi kebiasaan sehari-hari, misalnya : pasar pagi yang hanya berusaha pada jam-jam tertentu di pagi hari.

Di Amerika, pedagang kaki lima semacam ini disebut dengan *Hawkers* yang memiliki pengertian orang-orang yang menawarkan barang dan jasa untuk dijual di tempat umum, terutama di pinggir jalan dan trotoar. (McGee dan Yeung, 1977:25). *Hawkers center* atau juga dapat disebut *food center* yang ada di Singapore dan di negara lainnya sama dengan *food court* yang telah banyak di Indonesia. *Hawkers* yang berada di negara-negara maju seperti Singapore sudah tertata rapi dan menempati di pusat kota. Untuk pemetaan *hawkers* dinegara maju dilakukan oleh jasa-jasa travel, dan pemerintah setempat juga mendukung karena dengan adanya pemetaan *hawkers* menarik para turis, salah satu jasa travel yang memetakan *hawkers* adalah streetdirectory.com, australianexplorer.com, dan tripadvisor.com.

2.5 Sistem Informasi

Konsep dari Sistem Informasi terbagi atas dua pengertian. Yang pertama adalah sistem, dan yang kedua adalah sistem informasi itu sendiri.

2.5.1 Sistem

Definisi sistem dapat dibagi menjadi dua pendekatan, yaitu pendekatan secara prosedur dan pendekatan secara komponen. Berdasarkan pendekatan prosedur, sistem didefinisikan sebagai kumpulan dari beberapa prosedur yang mempunyai tujuan tertentu. Sedangkan berdasarkan pendekatan komponen, sistem merupakan kumpulan dari komponen-komponen yang saling berkaitan untuk mencapai tujuan tertentu.

Dalam perkembangan sistem yang ada, sistem dibedakan menjadi dua jenis, yaitu sistem terbuka dan sistem tertutup. Sistem terbuka merupakan sistem yang dihubungkan dengan arus sumber daya luar dan tidak mempunyai elemen pengendali. Sedangkan sistem tertutup tidak mempunyai elemen pengontrol dan dihubungkan pada lingkungan sekitarnya. (Herlambang, 2005).

2.5.2 Sistem Informasi

Data adalah fakta-fakta atau kejadian-kejadian yang dapat berupa angka-angka atau kode-kode tertentu. Data masih belum mempunyai arti bagi penggunanya. Untuk dapat mempunyai arti data diolah sedemikian rupa sehingga dapat digunakan oleh penggunanya. Hasil pengolahan data inilah yang disebut sebagai informasi. Secara ringkas, Informasi adalah data yang telah diolah dan mempunyai arti bagi penggunanya. Sehingga sistem informasi dapat didefinisikan sebagai prosedur-prosedur yang digunakan untuk mengolah data sehingga dapat digunakan oleh penggunanya (Herlambang, 2005).

2.6 Peta

Peta merupakan gambaran wilayah geografis, yang pada umumnya bagian permukaan bumi. Peta dapat disajikan dengan berbagai cara berbeda, dari peta konvensional yang tercetak sampai peta digital. Peta dapat menunjukkan banyak informasi penting, misal sungai, gunung, hutan, laut, batas-batas negara dan lain lain.

Peta adalah suatu representasi konvensional (miniatur) dari unsur-unsur fisik (alamiah dan buatan manusia) dari sebagian atau keseluruhan permukaan bumi di

atas media bidang datar dengan skala tertentu (Prahasta, 2002). Adapun persyaratan geometrik yang harus dipenuhi oleh suatu peta sehingga menjadi peta yang ideal adalah:

- a. Jarak antara titik-titik yang terletak di atas peta harus sesuai dengan jarak aslinya di permukaan bumi (dengan memperhatikan faktor skala tertentu).
- b. Luas suatu unsur yang direpresentasikan harus sesuai dengan luas sebenarnya (juga dengan mempertimbangkan skalanya).
- c. Sudut atau arah suatu garis yang direpresentasikan di atas peta harus sesuai dengan arah yang sebenarnya (seperti di permukaan bumi).
- d. Bentuk suatu unsur yang direpresentasikan di atas peta harus sesuai dengan bentuk yang sebenarnya. Pada kenyataannya di lapangan merupakan hal yang tidak mungkin menggambarkan sebuah peta dapat memenuhi semua kriteria di atas, karena permukaan bumi itu sebenarnya melengkung.

Sehingga pada saat melakukan proyeksi dari bentuk permukaan bumi yang melengkung tersebut ke dalam bidang datar(kertas) akan terjadi distorsi. Oleh karena itu maka akan ada kriteria yang tidak terpenuhi, prioritas kriteria dalam melakukan proyeksi peta tergantung dari penggunaan peta tersebut di lapangan.

Jenis peta dapat di golongan menjadi beberapa golongan, antara lain:

A. Jenis peta berdasarkan skala

Jenis peta ini berdasarkan besar skala, antara lain:

- a. Peta kadaster, berskala 1:100 sampai 1:5000, menggambarkan peta-peta tanah dan peta sertifikat tanah.
- b. Peta skala besar, berskala 1:5000 sampai 1:250000, menggambarkan wilayah-wilayah yang relatif sempit, seperti kelurahan dan kecamatan.

- c. Peta skala sedang, berskala 1:250000 sampai 1:500000, menggambarkan wilayah-wilayah yang lebih tidak terlalu luas, seperti propinsi, daerah regional dan pulau.
- d. Peta skala kecil, berskala 1:500000 samapi 1:1000000, menggambarkan wilayah-wilayah yang cukup luas, misalnya negara.
- e. Peta skala geografis, berskala lebih dari 1:1000000, menggambarkan sekumpulan negara, benua, atau dunia.

B. Jenis Peta Berdasarkan bentuknya

Jenis Peta ini berdasarkan bentuk, antara lain:

- a. Peta timbul, peta jenis ini menggambarkan bentuk bumi yang sebenarnya, misalnya peta relief.
- b. Peta datar (peta biasa), peta umumnya yang dibuat pada bidang datar, misalnya kertas, kain atau kanvas.
- c. Peta digital, peta digital adalah peta yang datanya terdapat pada suatu pita magnetik atau disket, sedangkan pengolahan dan penyajian datanya menggunakan komputer. Peta digital dapat ditayangkan melalui monitor komputer atau layar televisi. Peta digital ini hadir seiring perkembangan teknonolgi komputer dan peralatan digital lainnya.

C. Jenis Peta Berdasarkan Informasi

Jenis peta ini berdasarkan informasi yang disajikan, antara lain:

1. Peta Umum/ Peta Ikhtisiar

Peta umum merupakan peta yang menggambarkan topologi daerah ataupun batas-batas administrasi suatu wilayah/ Negara yang biasa digunakan untuk bermacam-macam tujuan.

2. Peta Khusus/ Peta Tematik

Peta tematik merupakan peta yang menampilkan hubungan keruangan, kenampakan tertentu di permukaan bumi dalam bentuk atribut tunggal atau hubungan atribut seperti geologi, geografis, pertanahan dan sebagainya.

Contohnya:

a. Peta Geologi

Peta yang menggambarkan struktur batuan dan sifat-sifatnya yang dapat mempengaruhi bentuk-bentuk permukaan tanah.

b. Peta Air Tanah

Peta yang menggambarkan lokasi atau sebaran air tanah di suatu tempat/ daerah.

c. Peta Irigasi

Peta yang menggambarkan tentang aliran sungai, bendungan air dan saluran irigasi.

d. Peta Transportasi

Peta yang menggambarkan peta lalu lintas baik di darat, laut maupun udara.

e. Peta Lokasi

Peta yang menggambarkan keadaan tinggi rendahnya permukaan bumi.

f. Peta Arkeologi

Peta yang menggambarkan penyebaran letak benda-benda atau peninggalan purba.

2.7 Sistem Informasi Geografis

Sistem Informasi Geografis adalah sistem berbasis komputer yang digunakan untuk menyimpan, memanipulasi dan menganalisis informasi geografi (Paryono, 1994). Adapun pengertian lain tentang sistem informasi geografis merupakan suatu kesatuan formal yang terdiri dari berbagai sumber daya fisik dan logika yang berkaitan dengan objek-objek yang terdapat dipermukaan bumi (Prahasta, 2001). Dengan demikian, pengertian terhadap ketiga unsur-unsur pokok ini akan sangat membantu dalam memahami SIG. Dengan melihat unsur-unsur pokoknya, maka jelas SIG dengan tambahan unsur "Geografis". SIG merupakan suatu sistem yang menekankan pada unsur "informasi geografis"

Istilah "Informasi Geografis" mengandung pengertian informasi mengenai tempat-tempat yang terletak di permukaan bumi, pengetahuan mengenai posisi dimana suatu objek terletak di permukaan bumi, dan informasi mengenai keterangan-keterangan (atribut) yang terdapat di permukaan bumi yang posisinya diketahui.

Menurut Abidin(2007) Pemetaan dapat didefinisikan sebagai suatu proses terpadu yang mencakup pengumpulan, pengolahan dan visualisasi dari data spasial (keruangan). Data spasial umumnya didefinisikan sebagai data keruangan yang terkait dengan permukaan Bumi (termasuk dasar laut) serta obyek, fenomena dan proses yang berada, terjadi atau berlangsung di atasnya. Produk suatu proses pemetaan adalah suatu informasi spasial yang dapat divisualisasikan dalam bentuk

atlas (kertas maupun elektronik), peta (kertas maupun digital), basis data digital maupun Sistem Informasi Geografis (SIG).

Dengan memperhatikan pengertian sistem informasi, maka SIG merupakan suatu kesatuan formal yang terdiri dari berbagai sumberdaya fisik dan logika yang berkenaan dengan objek-objek yang terdapat di permukaan bumi. (Prahasta, 2001).

2.7.1 Subsistem Sistem Informasi Geografis

Dari definisi SIG dapat diuraikan menjadi beberapa subsistem berikut (Prahasta, 2001):

1. Data Input

Subsistem ini bertugas untuk mengumpulkan dan mempersiapkan data spasial dan atribut dari berbagai sumber. Subsistem ini pula yang bertanggungjawab dalam mengkonversi atau mentransformasikan format-format data-data aslinya ke dalam format yang dapat digunakan oleh SIG.

2. Data Output

Subsistem ini menampilkan atau menghasilkan keluaran seluruh atau sebagian basisdata baik dalam bentuk *softcopy* maupun bentuk *hardcopy* seperti: tabel, grafik, peta, dan lain-lain.

3. Data Management

Subsistem ini mengorganisasikan baik data spasial maupun atribut ke dalam sebuah basis data sedemikian rupa sehingga mudah dipanggil, di-*update*, dan di-*edit*.

4. Data Manipulation & Analisis

Subsistem ini menentukan informasi-informasi yang dapat dihasilkan oleh SIG. Selain itu, subsistem ini juga melakukan manipulasi dan permodelan data untuk menghasilkan informasi yang diharapkan.

2.7.2 Komponen Sistem Informasi Geografis

SIG merupakan sistem kompleks yang biasanya, terintegrasi dengan lingkungan sistem-sistem komputer yang lain di tingkat fungsional dan jaringan sistem SIG terdiri dari beberapa komponen berikut (Prahasta, 2001):

1. Perangkat keras

Pada saat ini SIG tersedia untuk berbagai *Platform* perangkat keras mulai dari *PC desktop*, *workstations*, hingga *multiuser host* yang dapat digunakan oleh banya orang secara bersamaan dalam jaringan komputer yang luas, berkemampuan tinggi, memiliki ruang penyimpanan (*harddisk*) yang besar, dan mempunyai kapasitas memori (RAM) yang besar. Walaupun demikian, fungsionalitas SIG tidak terikat secara ketat terhadap karakteristik-karakteristik fisik perangkat keras ini sehingga keterbatasan memori pada PC-pun dapat diatasi. Adapun perangkat keras yang sering digunakan untuk SIG adalah komputer (PC), *mouse*, *digitizer*, *printer*, *plotter*, dan *scanner*.

2. Perangkat lunak

Bila dipandang dari sisi lain, SIG juga merupakan sistem perangkat lunak yang tersusun secara modular dimana basisdata memegang peranan kunci. Setiap subsistem (telah dibahas di atas) diimplementasikan dengan menggunakan perangkat lunak yang terdiri dari beberapa modul, hingga tidak

mengerahkan jika ada perangkat SIG yang terdiri dari ratusan modul program (exe) yang masing-masing dapat dieksekusi.

3. Data & informasi geografi

SIG dapat mengumpulkan dan menyimpan data dan informasi yang diperlukan baik secara tidak langsung dengan cara meng-*import*-nya dari perangkat-perangkat lunak SIG yang lain maupun secara langsung dengan cara mendijitasi data spasialnya dari peta dan memasukkan data atributnya dari tabel-tabel dan laporan dengan menggunakan keyboard.

4. Manajemen

Suatu proyek Sistem Informasi Geografis akan berhasil jika di kelola dengan baik dan dikerjakan oleh orang yang memiliki keahlian yang tepat.

Manajemen sering disebut juga sumber daya manusia atau *brainware*.

2.8 PHP

PHP (*Personal Home Page tools*) adalah skrip bersifat server-side yang ditambahkan ke dalam HTML (*HyperText Markup Language*). Skrip ini akan membuat suatu aplikasi dapat diintegrasikan ke dalam HTML sehingga suatu halaman web tidak lagi bersifat statis, namun menjadi bersifat dinamis. Sifat server-side berarti pengerjaan skrip akan dilakukan di server, baru kemudian hasilnya dikirim ke browser (Kurniawan, 2002:1). Keunggulan dari sifatnya yang server-side tersebut antara lain :

1. Tidak diperlukan kompatibilitas browser atau harus menggunakan browser tertentu, karena serverlah yang akan mengerjakan skrip PHP. Hasil yang dikirimkan kembali ke browser umumnya bersifat teks atau gambar saja sehingga pasti dikenal oleh browser apa pun.

2. Dapat memanfaatkan sumber-sumber aplikasi yang dimiliki oleh server, misalnya koneksi ke database.
3. Skrip tidak dapat “diintip” dengan menggunakan fasilitas *view HTML sourcecode*.

Kelebihan PHP dapat melakukan semua aplikasi program CGI (*Common Gateway Interface*), seperti mengambil nilai form, menghasilkan halaman web yang dinamis, serta mengirim dan menerima cookie. PHP juga dapat berkomunikasi dengan layanan-layanan yang menggunakan protokol IMAP, SNMP, NNTP POP3, HTTP, dan lain-lain. Namun kelebihan yang paling signifikan adalah kemampuannya untuk melakukan koneksi dengan berbagai macam database.

2.9 MySQL

MySQL adalah *Relational Database Management Sistem* (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial (Dwi Prasetyo, 2003).

Berikut ini beberapa keistimewaan yang dimiliki oleh MySQL (Dwi Prasetyo, 2003) :

1. *Portability* : dapat berjalan stabil pada berbagai sistem operasi (windows, Linux, Mac OS dan lain-lain).
2. *Open Source* : didistribusikan secara gratis, di bawah lisensi GPL sehingga dapat di pergunakan secara Cuma-Cuma tanpa dipungut biaya.

3. *Multiuser* : dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance Tuning* : memiliki kecepatan yang menakjubkan dalam menangani query sederhana.
5. *Column Types* : memiliki tipe kolom yang sangat kompleks, seperti signed/unsigned integer, float, double, char, varchar, text, blob, date, time, datetime, timestamp, year, set serta enum.
6. *Command dan Functions* : memiliki operator dan fungsi secara penuh yang mendukung perintah SELECT dan WHERE dalam query.
7. *Security* : memiliki beberapa lapisan sekuritas seperti level subnetmark, nama host, dan izin akses user dengan system perizinan yang mendetail serta password terenkripsi.
8. *Scalability dan Limits* : mampu menangani database dalam skala besar, dengan jumlah records lebih dari 50 juta dan 60 ribu table serta 5 miliar baris. Selain itu, batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. *Connectivity* : dapat melakukan koneksi dengan client menggunakan protokol TCP/IP, Unix socket (Unix), atau Named Pipes (NT).
10. *Localisation* : dapat mendeteksi pesan kesalahan (error code) pada client dengan menggunakan lebih dari dua puluh bahasa.
11. *Interface* : memiliki interface terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

12. *Lients* dan *Tools* : dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi database, dan pada setiap tool yang ada disertakan petunjuk online.

Struktur tabel : memiliki struktur tabel yang lebih lebih fleksibel dalam menangani ALTER TABLE, dibandingkan database lainnya semacam PostgreSQL ataupun Oracle.

2.10 Google Map Api

Google Maps adalah sebuah jasa peta globe virtual gratis dan online disediakan oleh Google dapat ditemukan di <http://maps.google.com/>. Google Map menawarkan peta yang dapat diseret dan gambar satelit untuk seluruh dunia.

Google Map API merupakan aplikasi interface yang dapat diakses lewat javascript agar Google Map dapat ditampilkan pada halaman web yang sedang kita bangun.

Google map adalah aplikasi yang mirip dengan Google Earth, namun ditampilkan dengan antarmuka web. Karena menggunakan thin client browser maka pengalaman yang akan diperoleh dari Google map tidak akan penuh ketika menggunakan Google Earth (Zaki,2010:153).

2.11 UML (*Unified Modeling Language*)

Unified modelling language (UML) adalah bahasa standar yang digunakan untuk menjelaskan dan memvisualisasikan artifak dari proses analisis dan disain berorientasi objek. UML menyediakan standar pada notasi dan diagram yang bisa

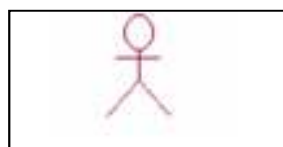
digunakan untuk memodelkan suatu sistem. Ada beberapa diagram yang disediakan dalam UML (Sholih, 2006), antara lain:

- a. Diagram *use case* (*use case diagram*)
- b. Diagram aktivitas (*activity diagram*)
- c. Diagram sekuensial (*sequence diagram*)
- d. Diagram kolaborasi (*collaboration diagram*)
- e. Diagram kelas (*class diagram*)
- f. Diagram *statechart* (*statechart diagram*)
- g. Diagram komponen (*component diagram*)
- h. Diagram *deployment* (*deployment diagram*)

Terdapat notasi pada UML, antara lain :

A. Actor

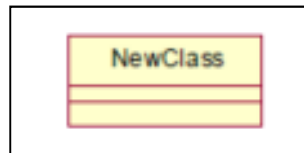
Actor adalah segala sesuatu yang berinteraksi dengan sistem aplikasi komputer. Jadi *actor* ini bisa berupa orang, perangkat keras, atau mungkin juga objek lain dalam sistem yang sama. Biasanya yang dilakukan oleh *actor* adalah memberikan informasi pada sistem dan memerintahkan sistem untuk melakukan sesuatu.



Gambar 2.1 Notasi Actor

B. Class

Class merupakan pembentuk utama dari sistem berorientasi objek karena *class* menunjukkan kumpulan objek yang memiliki atribut dan operasi yang sama. *Class* digunakan untuk mengimplementasikan interface.



Gambar 2.2 Notasi Class

C. Interface

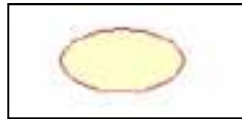
Interface merupakan kumpulan operasi tanpa implementasi dari suatu *class*. Implementasi operasi dalam *interface* dijabarkan oleh operasi dalam *class*. Oleh karena itu keberadaan *interface* selalu disertai oleh *class* yang mengimplementasikan operasinya. *Interface* ini merupakan salah satu cara mewujudkan prinsip enkapsulasi dalam objek.



Gambar 2.3 Notasi Interface

D. Use case

Use case menjelaskan urutan kegiatan yang dilakukan *actor* dan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan namun *use case* hanya menjelaskan apa yang dilakukan oleh *actor* dan sistem, bukan bagaimana *actor* dan sistem melakukan kegiatan tersebut.



Gambar 2.4 Notasi Use Case

E. Interaction

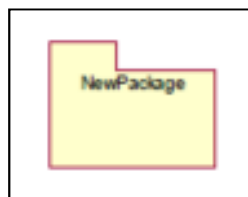
Interaction digunakan untuk menunjukkan baik aliran pesan atau informasi antar objek. Biasanya *interaction* ini dilengkapi juga dengan teks bernama *operation signature* yang tersusun dari nama operasi, parameter yang dikirim dan tipe parameter yang dikembalikan.



Gambar 2.5 Notasi Interaction

F. Package

Package adalah kontainer atau wadah konseptual yang digunakan untuk mengelompokkan elemen-elemen dari sistem yang sedang dibangun, sehingga bisa dibuat model yang lebih sederhana. Tujuannya adalah untuk mempermudah penglihatan (*visibility*) dari model yang sedang dibangun.



Gambar 2.6 Notasi Package

G. Note

Note digunakan untuk memberikan keterangan dan komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model. Note ini bisa ditempelkan ke semua elemen notasi yang lain.



Gambar 2.7 Notasi Note

H. Dependency

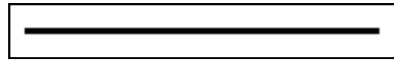
Dependency merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain. Elemen yang ada di bagian tanda panah adalah elemen yang tergantung pada elemen yang ada di bagian tanpa tanda panah.



Gambar 2.8 Notasi Dependency

I. Association

Association menggambarkan navigasi antar *class* (*Navigation*), beberapa banyak objek lain yang bisa berhubungan dengan satu objek (*Multiplicity* antar *class*), dan apakah suatu *class* menjadi bagian dari *class* lainnya (*Aggregation*).



Gambar 2.9 Notasi Association

J. Generalization

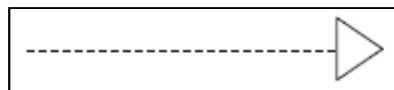
Generalization menunjukkan hubungan antara elemen yang lebih umum ke elemen yang lebih spesifik. Dengan *generalization*, *class* yang lebih spesifik (*subclass*) akan menurunkan atribut dan operasi dari *class* yang lebih umum (*superclass*), atau “*subclass is a superclass*”. Dengan menggunakan notasi *generalization* ini konsep *inheritance* dari prinsip hirarki dimodelkan



Gambar 2.10 Notasi Generalization

K. Realization

Realization menunjukkan hubungan bahwa elemen yang ada di bagian tanpa panah akan merealisasikan apa yang dinyatakan oleh elemen yang ada di bagian dengan panah. Misalnya merealisasikan *package*, *component* merealisasikan *class* atau *interface*.



Gambar 2.11 Notasi Realization

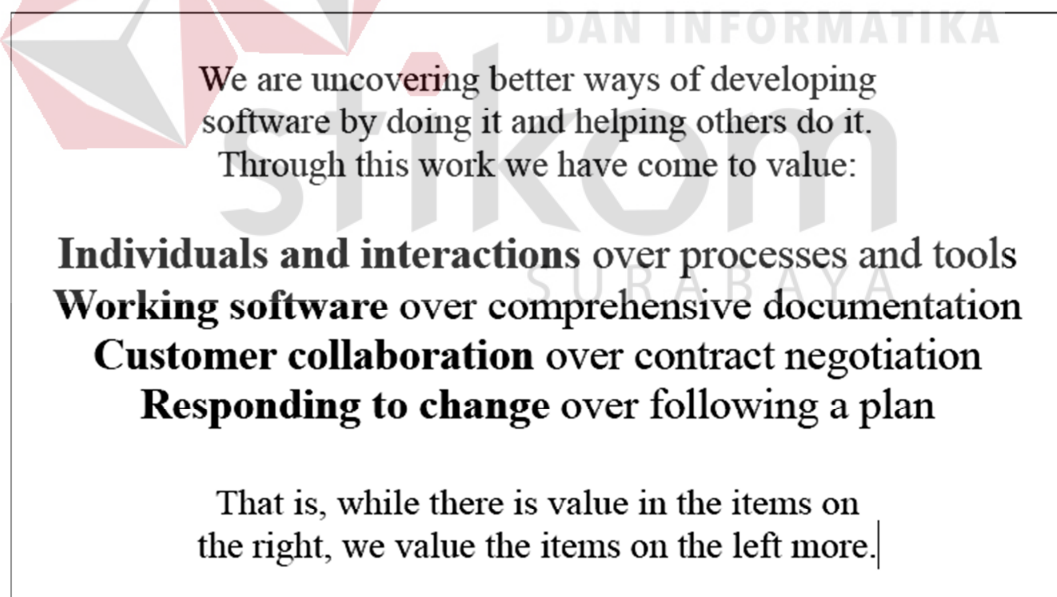
2.12 Metode Pengembangan

2.12.1 Pengembangan Perangkat Lunak Secara *Agile*

Secara umum pengembangan perangkat lunak menggunakan *agile* menawarkan pendekatan secara profesional yang meliputi aspek manusia,

organisasi, dan teknologi dalam proses pengembangannya. Secara khusus, gambaran utama dari pengembangan secara *agile* yaitu: yang pertama adalah dengan mendeskripsikan *Agile Manifesto* dan implementasinya, dan yang kedua adalah mengimplementasikan salah satu metode *agile* yang dapat membuat tim menyelesaikan tugas pengembangan dengan kualitas tinggi (Hazzan, 2008).

Agile Manifesto merupakan hasil formulasi dari tujuh belas pengembangan perangkat lunak yang dilakukan pada bulan Februari tahun 2001 bertempat di Wasatch Mountain, Utah. Penerapan *Agile Manifesto* dapat dilakukan dengan menggunakan salah satu metode *agile*. Beberapa metode *agile* diantaranya: *Extreme Programming*, *SCRUM*, *DSDM*, *Adaptive Software Development*, *Crystal*, *Feature-Driven Development* dan lainnya. Isi dari *Agile Manifesto* dapat dilihat pada gambar 2.12



Gambar 2.12 *Agile Manifesto*

2.12.2 *Agile Modeling*

Agile Modeling (AM) merupakan salah satu penerapan metode *agile*. Menurut Ambler(2012) *Agile Modeling* (AM) merupakan metodologi berbasis praktek untuk pemodelan dan dokumentasi yang efektif dari sistem perangkat lunak. Secara singkat AM adalah kumpulan dari nilai-nilai, prinsip-prinsip, dan praktek-praktek untuk pemodelan perangkat lunak yang dapat diaplikasikan pada proyek pengembangan perangkat lunak secara efektif dan dalam waktu singkat. Nilai-nilai yang ada pada *Agile Modeling* adalah sebagai berikut:

1. **Komunikasi:** penting untuk memiliki komunikasi yang efektif antara tim pengembang dan stackholder.
2. **Kesederhanaan:** berusaha untuk mengembangkan solusi sesederhana mungkin.
3. **Umpan balik:** dapatkan umpan balik lebih cepat dan lebih sering.
4. **Keberanian:** keberanian untuk mencoba teknik baru dan tetap pada pendirian.
5. **Kerendahan hati:** berusaha sadar bahwa anda tidak selalu mengetahui semua dengan demikian pihak lain dapat membantu menambahkan nilai pada pengembangan proyek.

2.12.3 *Agile Model-Driven Development (AMDD)*

Menurut Ambler (2012) AMDD merupakan versi *agile* dari Model Driven Development(MDD). MDD adalah sebuah pendekatan dalam pengembangan perangkat lunak yang mana pembuatan model secara ekstensif (menyeluruh) dilakukan sebelum mulai menulis kode. Perbedaan dengan AMDD adalah pada AMDD pembuatan model tidak dilakukan secara ekstensif melainkan cukup

membuat model yang dapat membuat pengembangan segera dijalankan. Langkah-langkah yang dilakukan dalam pengembangan menggunakan *Agile Model Driven Development* yaitu: *Envisioning*, *Iteration Modelling*, *Model Storming* dan implementasi via *Test Driven Development (TDD)*. Langkah-langkah akan dijelaskan, yaitu:

1. *Envisioning*

Proses *envisioning* dilakukan untuk mendapatkan gambaran umum dari sistem yang akan dibuat. Yang nantinya di *Envisioning* ini berisi *user stories* dan *use case* aplikasi.

2. Iterasi Pemodelan

Pada tahap ini pengembang *agile* harus menyusun estimasi jadwal dan pekerjaan yang akan dilakukan pada setiap iterasi. Untuk melakukan estimasi yang tepat maka pengembang harus memahami pekerjaan seperti apa untuk mengimplementasikannya, dan itulah tujuan pemodelan pada tahap ini.

Pemodelan yang akan dibuat dapat dirancang berdasarkan informasi yang tersedia dari sketsa antar muka dan *user stories* yang telah dibuat. Berdasarkan informasi tersebut tim dapat melakukan estimasi dan menentukan prioritas fitur-fitur mana yang akan diselesaikan lebih dahulu. Seperti tahap *agile* yang lain pemodelan pada tahap ini tidak perlu terlalu detail karena akan lebih disempurnakan saat *model storming* atau *Test Driven Development (TDD)*.

3. *Model Storming*

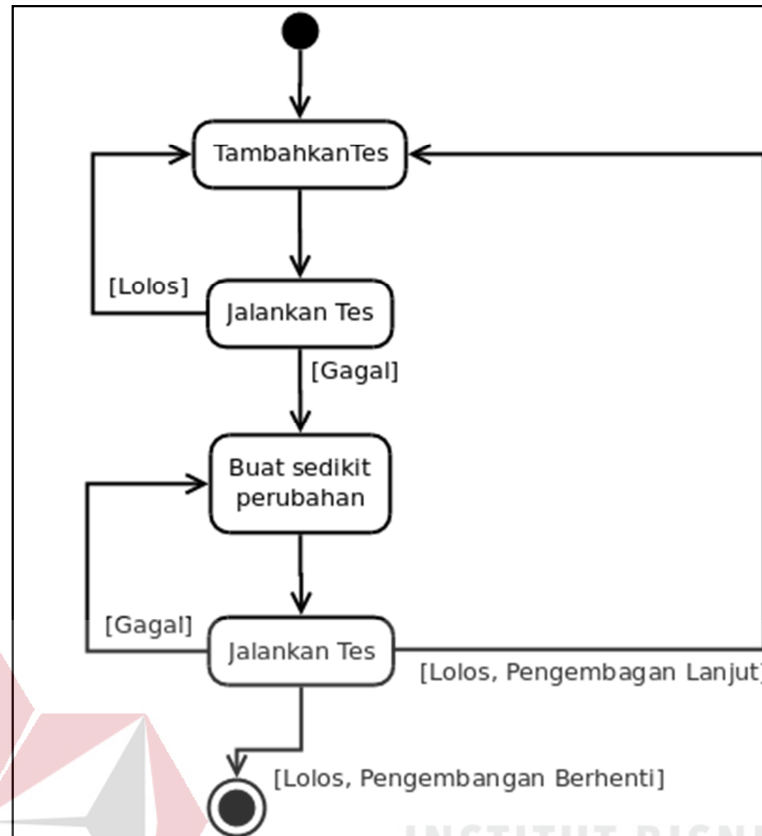
Menurut Ambler (2012) *model storming* adalah *Just In Time (JIT) modeling* yang artinya pengembang mengidentifikasi masalah yang akan diselesaikan, jika dalam tim maka pengembang mengajak rekan yang dapat membantu, tim

tersebut kemudian mengeksplorasi masalah dan kemudian kembali masing-masing melanjutkan pekerjaan seperti sebelumnya. Ambler (2012) menjelaskan bahwa *model storming* merupakan kegiatan yang mendadak dan berlangsung dalam hitungan menit, rata-rata lima sampai sepuluh menit. Pada sesi *model storming* terdapat dua tahap yaitu analisis *Model Storming* dan desain *Model Storming*.

4. Implementasi via Test Driven Development

Spesifikasi yang telah dibuat pada *iteration modeling* dan *model storming* dikerjakan menggunakan pendekatan TDD. Menurut (Beck 2003; Astel 2003) dalam Ambler (2012) Test-Driven Development (TDD) adalah sebuah pendekatan evolusioner dalam pengembangan yang mengkombinasikan *test-first development* dimana anda menulis sebuah tes sebelum anda sepenuhnya menulis kode yang akan diperuntukkan untuk memenuhi tes dan *refactoring* tersebut. Menurut Fowler (1999) dalam Ambler (2012) refactoring adalah sebuah cara dalam melakukan restrukturisasi (perubahan) pada kode.

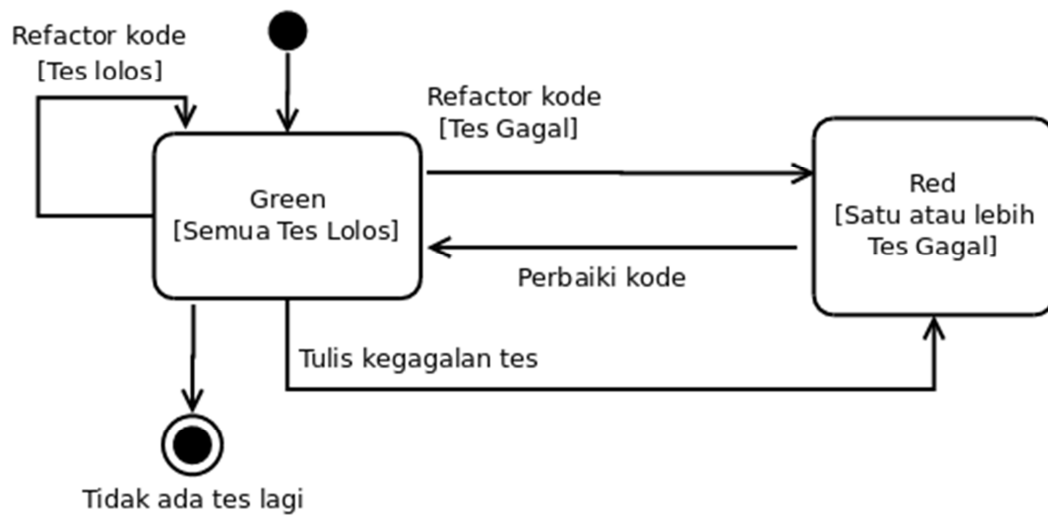
Secara singkat Ambler (2012) mendeskripsikan TDD sebagai kombinasi dari refactoring dan Test-First Development (TFD). Langkah-langkah pada TFD adalah menambahkan sebuah tes secara cepat, dibuat cukup untuk membuat sebuah kode akan gagal. Langkah berikutnya adalah menjalankan tes untuk memastikan apakah kode gagal atau tidak. Jika gagal maka ubah (refactor) bagian kode tersebut hingga dapat lolos. Jika menambahkan fungsionalitas baru pada kode maka buat dulu tes dari fungsionalitas tersebut kemudian jalankan tes, refactor bagian kode hingga lolos tes. Lakukan berulang kali, ilustrasi dari alur tersebut ditunjukkan pada gambar 2.13 pada halaman 32.



Gambar 2.13 Alur Pengembangan TDD (Ambler, 2011)

Menurut Beck (2002) tujuan utama TDD adalah untuk menciptakan kode yang bersih dan bekerja. Beck (2002) mengemukakan sebuah istilah yang disebut *red/green/refactor*. Maksud dari istilah tersebut adalah:

1. *Red* – Tulis tes kecil yang tidak dapat bekerja, atau bahkan mungkin tidak dapat di-compile.
2. *Green* – Buatlah tes berhasil secepatnya, perbaiki semua kesalahan yang diperlukan selama proses tersebut.
3. *Refactor* – Eliminasi semua duplikasi yang tercipta dalam proses membuat tes itu berhasil.



Gambar 2.14 Alur red/green/refactor pada TDD (Ambler, 2011)

Menurut Ambler (2012) keuntungan yang signifikan pada TDD adalah membantu pengembang mengambil beberapa langkah kecil saat menulis *software*. Karena hal tersebut terbukti lebih produktif daripada mencoba menulis kode langsung dalam jumlah yang banyak. Manfaat lain adalah kesalahan akan lebih mudah ditemukan dan diperbaiki.