

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Informasi Geografis

Geographic Information System (GIS), merupakan suatu sistem (berbasis komputer) yang digunakan untuk menyimpan dan memanipulasi informasi-informasi geografis. SIG dirancang untuk mengumpulkan, menyimpan, dan menganalisis objek-objek dan fenomena-fenomena dimana lokasi geografis merupakan karakteristik yang penting atau kritis untuk dianalisis. SIG merupakan sistem komputer yang memiliki empat kemampuan berikut dalam menangani data yang bereferensi geografis (Sumber : Eddy Prahasta: 2009: 1):

- a. Masukan
- b. Keluaran
- c. Manajemen data (penyimpanan dan pemanggilan data)
- d. Analisis dan manipulasi data (Sumber: Eddy Prahasta: 2009: 1)

II.2. MapServer

Map server adalah aplikasi *freeware* dan *open source* yang memungkinkan kita menampilkan data spasial (peta) di web. Aplikasi ini pertama kali dikembangkan di Universitas Minesotta, Amerika Serikat untuk proyek *ForNet* (sebuah proyek untuk manajemen sumber daya alam) yang disponsori NASA (*National Aeronautics And Space Administration*). Support NASA dilanjutkan dengan proyek *TerraSIP* untuk manajemen data lahan. Saat ini, karena sifatnya

yang terbuka (*open source*), pengembangan MapServer dilakukan oleh pengembangan dari berbagai Negara.

Pengembangan MapServer menggunakan berbagai aplikasi *open source* atau *freeware* seperti Shapelib (<http://shapelib.maptools.org>) untuk baca/tulis format data *Shapefile*, FreeType (<http://www.freetype.org>) untuk merender karakter, GDAL/OGR (<http://www.remotesensing.org/gdal>) untuk baca/tulis berbagai format data *vektor* maupun juga data *raster*, dan Proj.4 (<http://www.remotesensing.org/proj>) untuk menanggapi beragam proyeksi peta.

Pada bentuk paling dasar, Mapserver berupa sebuah program CGI (*Common Gateway Interface*). Program tersebut akan dieksekusi di *web server*, dan berdasarkan beberapa parameter tertentu (terutama konfigurasi dalam bentuk file *.MAP) akan menghasilkan data yang kemudian akan dikirim ke *web browser*, baik dalam bentuk gambar peta ataupun bentuk lain (Ruslan Nurydin; 2005: 3).

II.2.1. Arsitektur MapServer

Interaksi antara klien dengan server berdasar scenario *request* dan respon. Web *browser* di sisi klien mengirim *request* ke server web. Karena server web tidak memiliki kemampuan pemrosesab peta, maka *request* berkaitan dengan pemrosesan peta akan diteruskan oleh server web ke server aplikasi dan MapServer. Hasil pemrosesan akan dikembalikan lagi melalui server web, terbungkus dalam bentuk file HTML atau applet.

Arsitektur aplikasi pemetaan di web dibagi menjadi dua pendekatan sebagai berikut :

a. Pendekatan *Thin Client*

Pendekatan ini memfokuskan diri pada sisi server. Hampir semua proses dan analisis data dilakukan berdasarkan permintaan (*request*) di sisi server. Data hasil pemrosesan kemudian dikirimkan ke klien dalam format standar HTML, yang di dalamnya terdapat file gambar dalam format standar (misalnya GIF, PNG atau JPG) sehingga dapat dilihat menggunakan sembarang *web browser*. Kelemahan utama pendekatan ini menyangkut keterbatasan pilihan interaksi dengan pengguna yang kurang fleksibel.

b. Pendekatan *Thick Client*

Pada pendekatan ini, pemrosesan data dilakukan di sisi klien menggunakan beberapa teknologi seperti control ActiveX atau applet. Kontrol ActiveX atau applet akan dijalankan di klien untuk memungkinkan *web browser* dengan kemampuan standar. Dengan adanya pemrosesan di klien, maka transfer data antara klien dengan web server akan berkurang.

MapServer menggunakan pendekatan *thin client*. Semua pemrosesan dilakukan di sisi server. Informasi peta dikirimkan ke *web browser* di sisi klien dalam bentuk file gambar (JPG, PNG, GIF atau TIFF). Untungnya, saat ini kelemahan pendekatan *thin client* dalam hal interaksi dengan pengguna sudah jauh berkurang dengan adanya *framework* aplikasi seperti *Chameleon* atau *CartoWeb* (Ruslan Nurydin; 2005: 8).

II.3. ArcView

ArcView merupakan salah satu perangkat lunak desktop Sistem Informasi Geografis dan pemetaan yang telah dikembangkan oleh ESRI (*Environmental*

Systems Research Institute). Dengan ArcView, pengguna dapat memiliki kemampuan-kemampuan untuk melakukan visualisasi, meng-explore, menjawab query (baik basisdata spasial maupun non-spasial), menganalisis data secara geografis, dan sebagainya (Eddy Prahasta: 2009: 1).

II.4. UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah suatu alat Bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek (Munawar ; 2005 : 17). Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

Meskipun UML sudah banyak menyediakan diagram yang bisa membantu mendefinisikan suatu aplikasi, tidak berarti bahwa semua diagram tersebut akan bisa menjawab persoalan yang ada. Adapun tipe diagram UML yang ada seperti pada Tabel II.1.

Tabel II.1 Tipe Diagram UML

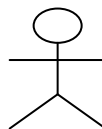
Diagram	Tujuan
Activity	Prilaku prosedural dan paralel
Class	Class, fitur dan relasinya
Communication	Interaksi diantara objek. Lebih menekankan kepada link
Component	Struktur dan koneksi dari komponen
Composite Structure	Dekomposisi sebuah class saat runtime
Deployment	Penyebaran/instalasi ke klien
Interaction Overview	Gabungan dari activity dan sequence diagram
Object	Contoh konfigurasi instance
Package	Struktur hierarki saat kompilasi
Sequence	Interaksi antara objek. Lebih menekankan pada urutan.
State Machine	Bagaimana event mengubah sebuah objek
Timing	Interaksi antar objek. Lebih menekankan pada waktu
Use Case	Bagaimana user berinteraksi dengan sebuah sistem

Sumber : " Pemodelan Visual dengan UML (Sumber: Munawar ; 2009 : 32) "

II.4.1. Notasi Dasar UML

1. Actor

Sebuah actor mencirikan suatu bagian *outside user* atau susunan yang berkaitan dengan user yang berinteraksi dengan sistem [Rumbaugh, Booch, dan Jacobson 1999]. Dalam model use case, aktor merupakan satu-satunya kesatuan eksternal yang berinteraksi dengan sistem (*Munawar ; 2009 : 41*). Berikut notasi actor pada gambar II.1 Berikut ini :

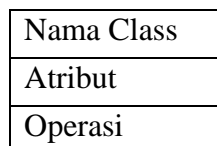


Gambar II.1 : Notasi Actor pada UML

Sumber : "Pemodelan Visual dengan UML (Munawar ; 2009 : 41) "

2. Class

Notasi utama dan yang paling mendasar pada diagram UML adalah notasi untuk mempresentasikan suatu class beserta dengan atribut dan operasinya. Class adalah pembentuk utama dari sistem yang berorientasi objek (Munawar ; 2009 : 41). pada gambar II.2 Berikut ini :

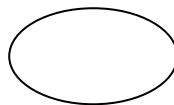


Gambar II.2 : Notasi Class pada UML

Sumber : "Pemodelan Visual dengan UML (Munawar ; 2009 : 41)"

3. Use Case

Use Case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah – langkah yang menerangkan antara pengguna dan sistem disebut *scenario* (Munawar ; 2009 : 42). Notasi *use case* dapat dilihat pada gambar II.3 :

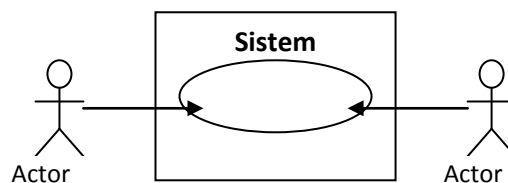


Gambar II.3 : Notasi Use Case pada UML

Sumber : "Pemodelan Visual dengan UML (Munawar ; 2009 : 42)"

4. Use Case Diagram

Use Case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* mempresentasikan sebuah interaksi antara actor dengan sistem. *Use Case* menggambarkan kata kerja seperti *Login* ke sistem, *maintenance user* dan sebagainya (Munawar ; 2009 : 43). Model use case seperti gambar 4 dan contoh use case diagram ditunjukkan pada Gambar II.4 :



Gambar II.4 : Model Use Case pada UML

Sumber : ” *Pemodelan visual dengan UML (Munawar ; 2009 : 43)*”

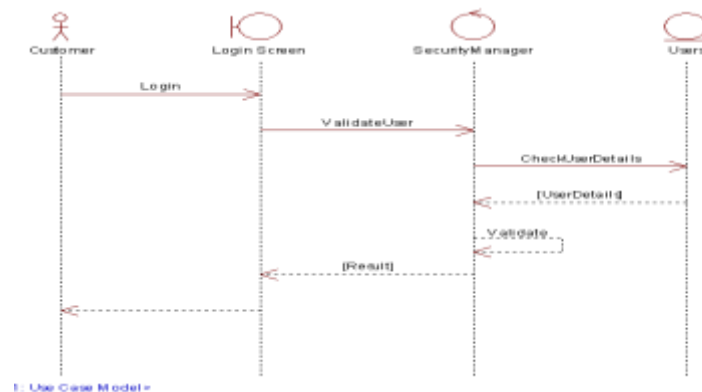
5. Sequence Diagram

Sequence diagram menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*: interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi. Pembuatan *sequence* diagram merupakan aktivitas yang paling kritikal dari proses disain karena artifak inilah yang menjadi pedoman dalam proses pemrograman nantinya dan berisi aliran kontrol dari program (Munawar ; 2009 : 3).

Sequence diagram biasanya tersusun dari elemen objek, *Interaction* dan *Message*. *Interaction* menghubungkan 2 Obyek dengan pesannya. Diagram ini

menjelaskan aspek dinamis dari sistem yang sedang dibangun. Di dalam *sequence diagram*, terdapat kelas *boundary*, *control* dan *entity*.

Contoh *sequence diagram* :



Gambar II.5 : Sequence Diagram

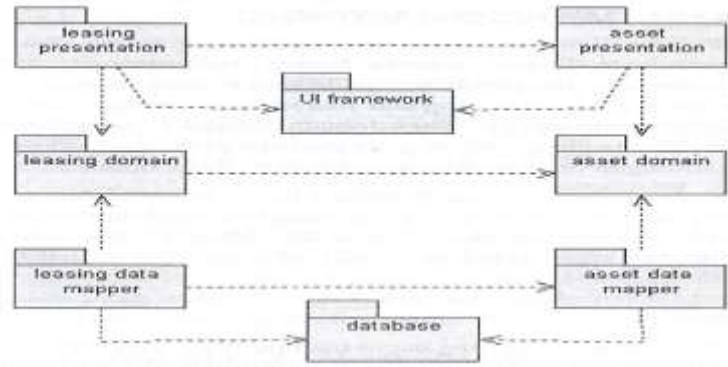
Sumber : ” Memahami Pengguna UML(Unified Modeling Language (Munawar ; 2009 : 3) ”

6. Class Diagram

Sama seperti *class*, maka class diagram merupakan diagram yang selalu ada di pemodelan sistem berorientasi objek. *Class diagram* menunjukkan hubungan antar *class* yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan (Munawar ; 2009 : 4).

menggambarkan navigasi antar *class*, berapa banyak objek lain bisa berhubungan dengan satu objek (*multiplicity* antar *class*), dan apakah satu *class* menjadi bagian dari *class* lainnya (*agregation*). *Class diagram* digunakan untuk menggambarkan disain statis dari sistem yang sedang dibangun.

Contoh *class diagram* :



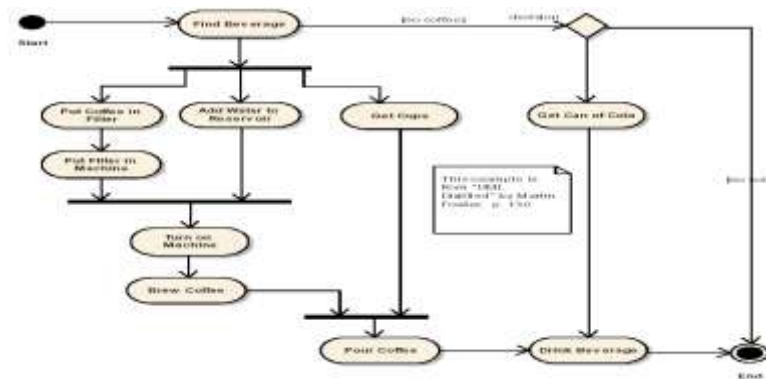
Gambar II.6 : Class Diagram

Sumber : " Memahami Pengguna UML(Unified Modeling Language
(Munawar ; 2009 : 4)"

7. Activity Diagram

Activity diagram adalah adalah tehnik untuk mendeskripsikan logika procedural, proses bisnis dan aliran kerja dalam banyak kasus. Activity diagram mempunyai peran seperti halnya flowchart, akan tetapi perbedaannya dengan flowchart adalah activity diagram bias mendukung perilaku parallel sedangkan flowchart tidak bisa. *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. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem dan interaksi antar subsistem secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas

dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, standart UML menggunakan segi empat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu, digambarkan dengan simbol belah ketupat. Untuk mengilustrasikan proses-proses paralel (*fork and join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu. Adapun contoh dari Activity Diagram dapat di lihat pada Gambar II.7.



Gambar II.7. :Activity Diagram

Sumber : " Memahami Pengguna UML(Unified Modeling Language (Munawar ; 2009 : 4) "

II.5. PHP (*Hypertext Preprocessor*)

Untuk membuat sebuah *website* yang dinamis dan mudah untuk *update* setiap saat dari *browser*, dibutuhkan sebuah program yang mampu mengolah data dari komputer *client* atau dari komputer *server* itu sendiri sehingga mudah dan nyaman untuk disajikan di *browser*.

Salah satu program yang dapat dijalankan di *server* dan cukup handal adalah PHP. PHP adalah salah satu bahasa pemrograman yang berjalan dalam sebuah *web server* dan berfungsi sebagai pengolah data pada sebuah server. Dengan menggunakan program PHP, sebuah *website* akan lebih interaktif dan dinamis. Data yang dikirim oleh pengunjung *website*/komputer *client* akan diolah dan disimpan pada *database web server* dan dapat ditampilkan kembali apabila diakses. Untuk menjalankan kode-kode program PHP ini, file harus *upload* ke dalam *server*.



PHP Version 5.2.6	
System	Windows NT 6SR0AD621 5.1 build 2600
Build Date	May 2 2008 18:01:20
Configure Command	fc:\opt\mologo\configure.js "--enable-mbstring-build" "--with-gd-shared" "--with-extra-include=C:\Program Files (x86)\Microsoft SDKs\Include\C:\PROGRAM-2\MICROS-3\VC98\ATL\INCLUDE;C:\PROGRAM-2\MICROS-2\VC98\INCLUDE;C:\PROGRAM-2\MICROS-2\VC98\MF\INCLUDE" "--with-extra-lib=C:\Program Files (x86)\Microsoft SDKs\LIB\C:\PROGRAM-2\MICROS-2\VC98\LIB;C:\PROGRAM-2\MICROS-2\VC98\MF\CLIB"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	C:\WINDOWS\php.ini
PHP API	20041225
PHP Extension	200000013
Zend Extension	220000510
Debug Build	no
Thread Safety	enabled
Zend Memory Manager	enabled
IPFilter Support	enabled
Registered PHP Streams	php, file, data, ftp, ftps, compress.zlib
Registered Stream Socket Transports	tcp, udp
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, string.rot13, ioncube.zlib.*
This program makes use of the Zend Scripting Language Engine: Zend Engine v2.2.0, Copyright (c) 1998-2008 Zend Technologies	

Gambar II.8. Tampilan phpinfo (Sumber: Bunafit Nugroho: 2009: 13)

II.6. MySQL

MySQL merupakan aplikasi *database* yang termasuk paling populer dalam lingkungan Linux, kepopuleran ini ditunjang karena performansi *query* dari databasenya yang sangat cepat dan jarang bermasalah. Saat ini, MySQL telah tersedia juga dalam lingkungan Windows, MySQL di lingkungan Windows diletakan pada direktori *c:\mysql\bin* adalah direktori yang berisi daftar modul *executable* dari MySQL.

PhpMyAdmin adalah aplikasi berbasis web yang dibuat dari pemrograman *PHP* dan *JavaScript*. *PhpMyAdmin* juga dapat disebut sebagai tools yang berguna untuk mengakses database MySQL Server dalam bentuk tampilan *web*. dengan adanya *phpMyAdmin*, semua pekerjaan menjadi mudah, karena tanpa harus mengerti perintah-perintah dasar *SQL* namun sudah dapat memanajemen database dan datan yang ada didalamnya (Sumber: Bunafit Nugroho: 2009: 13). Berikut ini adalah tampilan MySQL pada gambar II.9 di bawah ini.



Gambar II.9. MySQL diakses dengan *phpMyAdmin* di *Web-Browser*

(Sumber: Bunafit Nugroho: 2009: 13).