

BAB II

TINJAUAN PUSTAKA

II.1. Konsep Sistem Informasi Geografis

Sistem informasi geografis (*geographic information system* atau *GIS*) yang selanjutnya akan disebut sistem informasi geografis merupakan sistem informasi berbasis komputer yang digunakan untuk mengolah dan menyimpan data atau informasi geografis. Keunggulan utama dari sistem informasi geografis adalah sistem informasi geografis memungkinkan kita untuk melihat, memahami, menanyakan, menginterpretasi dan menampilkan data spasial dalam banyak cara, yang memperlihatkan hubungan, pola dan *trend* secara spasial, dalam bentuk peta, *globe*, laporan dan grafik. sistem informasi geografis mampu membantu dalam pemecahan masalah dengan cara menampilkan data menggunakan cara yang mudah dipahami dan hasilnya mudah disebarluaskan (Bramantiyo Marjuki ; 2014 : 1).

sistem informasi geografis sebagai suatu sistem yang mengorganisir *hardware*, *software*, dan data, serta dapat mendayagunakan sistem penyimpanan, pengolahan, maupun analisis data secara simultan, sehingga dapat diperoleh informasi yang berkaitan dengan aspek keruangan. Bahwa sistem informasi geografis merupakan manajemen data spasial dan non-spasial yang berbasis komputer dengan tiga karakteristik dasar, yaitu : mempunyai fenomena aktual (variabel data *non-lokasi*) yang berhubungan dengan topik permasalahan di lokasi

bersangkutan, merupakan suatu kejadian di suatu lokasi dan mempunyai dimensi waktu.

II.1.1. Komponen Sistem Informasi Geografis

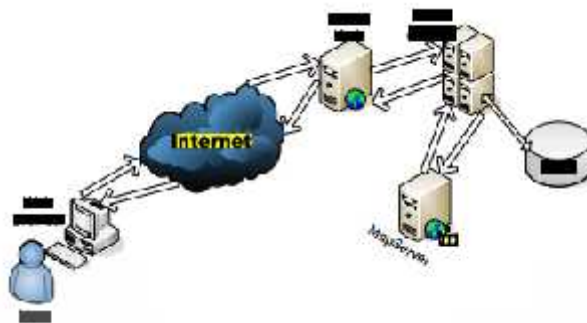
Menurut Bramantiyo Marjuki (2014 : 2) Sebagai salah satu jenis sistem informasi, SIG mempunyai sub sistem atau komponen yang bekerja secara bersama untuk menghasilkan fungsionalitas sistem informasi geografis. Komponen sistem informasi geografis terdiri dari *hardware*, *software*, data dan metode.

1. *Hardware* atau perangkat keras merupakan media tempat pelaksanaan proses sistem informasi geografis. *Hardware* yang diperlukan dalam sebuah sistem informasi geografis meliputi perangkat keras untuk masukan data, penyimpanan data, pengolahan dan analisa data dan pembuatan keluaran.
2. *Software* atau perangkat lunak merupakan alat pelaksana pekerjaan sistem informasi geografis. *Software* standar sistem informasi geografis harus mempunyai kapabilitas data *input*, penyimpanan, manajemen data, transformasi dan konversi data, analisa dan pembuatan keluaran.
3. Data atau representasi dari sebuah obyek atau fenomena adalah bahan yang dianalisa di dalam sistem informasi geografis. sistem informasi geografis memerlukan sebuah jenis data yang spesifik agar dapat memberikan keluaran seperti fungsionalitasnya. Data yang digunakan dalam sistem informasi geografis adalah data geospasial atau data yang berefrensi geografis (mempunyai informasi lokasi).

4. Metode adalah cara bagaimana data diolah untuk menjadi sebuah informasi. Metode meliputi aspek pemasukan data ke dalam sistem, bagaimana data dikelola dan disimpan, bagaimana data dianalisis, dan bagaimana informasi ditampilkan.

II.1.2. Arsitektur Sistem Informasi Geografis

Web sistem informasi geografis bisa dikatakan adalah sebuah *web* mapping yang berarti pemetaan internet, tetapi bukan memetakan internet. *Web* mapping memanfaatkan fungsi interaktifitas yang ada pada aplikasi sistem informasi geografis dalam bentuk *web*. Bentuk umum arsitektur berbasis peta di *web* dapat dilihat pada Gambar berikut :



Gambar II.1. Arsitektur Aplikasi SIG Berbasis Web
(Sumber : Fie Jannatin Aliyah ; 2012 : 8)

Interaksi antara *user* dengan *server* berdasar request dan respon. *Web* browser pada *user* mengirim request ke *server web*. Karena *server web* tidak memiliki kemampuan pemrosesan 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* PHP (Fie Jannatin Aliyah ; 2012 : 7).

II.2. Internet

II.2.1. Pengertian Internet

Internet (*Inter-Network*) adalah sebutan untuk sekumpulan jaringan komputer yang menghubungkan situs akademik, pemerintahan, komersial, organisasi, maupun perorangan. Internet menyediakan akses untuk layanan telekomunikasi dan sumber daya informasi untuk jutaan pemakainya yang tersebar di seluruh dunia. Layanan internet meliputi komunikasi langsung (*email, chat*), diskusi (*Usenet News, email, milis*), sumber daya informasi yang terdistribusi (*World Wide Web, Gopher*), *remote login* dan lalu lintas *file* (*Telnet, FTP*), dan aneka layanan lainnya. Jaringan yang membentuk internet bekerja berdasarkan suatu set protokol standar yang digunakan untuk menghubungkan jaringan komputer dan mengamati lalu lintas dalam jaringan. Protokol ini mengatur format data yang diijinkan, penanganan kesalahan (*error handling*), lalu lintas pesan, dan standar komunikasi lainnya (Ramadhani, 2013).

Protokol standar pada internet dikenal sebagai TCP atau IP (*Transmission Control Protocol* atau *Internet Protocol*). Protokol ini memiliki kemampuan untuk bekerja di atas segala jenis komputer, tanpa terpengaruh oleh perbedaan perangkat keras maupun sistem operasi yang digunakan. Sebuah sistem komputer yang terhubung secara langsung ke jaringan memiliki nama domain dan alamat IP (*Internet Protocol*) dalam bentuk numerik dengan format tertentu sebagai pengenal. Internet juga memiliki *gateway* ke jaringan dan layanan yang berbasis protokol lainnya (Ramadhani, 2013).

II.3. Website

II.3.1. Website Statis

Umumnya *web* statis cocok untuk usaha kecil dan menengah, juga untuk kelompok usaha yang menghasilkan produk atau jasa yang tetap selama satu tahun atau sangat jarang melakukan *update* dalam setahun. *Web* statis masih dapat dibagi lagi dalam tiga kategori desain yang berbeda, yaitu :

a. Full desain standar HTML

Umumnya formatnya lebih statis, walaupun ada sedikit efek animasi biasanya dilakukan hanya untuk beberapa bagian tertentu seperti menu atau foto *gallery*. Umumnya *web* dalam format desain ini lebih ringan dan cepat dibuka dan bila dibutuhkan dapat dirubah sewaktu-waktu (dengan catatan yang melakukan perubahan harus memahami *coding* dan perintah-perintah HTML).

b. Full desain animatif dengan menggunakan *Flash*

Biasanya *web* dalam format ini tampil lebih menarik, namun juga sangat berat dan lambat untuk dibuka, pengunjung harus memiliki koneksi yang cukup baik untuk dapat membuka *website* jenis ini. Masalahnya adalah ketika isi sebuah *web* jenis ini perlu di *update* maka *script Flash* harus diperbaiki dan di *compile* ulang yang kemudian hasilnya diletakkan kembali ke dalam *website*.

c. Kombinasi desain HTML dengan Flash Solusi terbaik untuk desain jenis statis adalah dengan menggabungkan desain standar HTML dengan teknologi desain Flash di bagian tertentu untuk memberikan efek“dinamis”. Biasanya animasi Flash digunakan untuk *Heading* atau *Header* maupun *banner* (Saputra, 2015)

II.3.2. Website Dinamis

Web dinamis digunakan oleh perusahaan yang yang sewaktu-waktu produknya dapat berubah dan bertambah dengan cepat, sering mengadakan pameran, sering mengadakan program *promo* atau *sale*, ataupun perlu dengan segera menginformasikan perkembangan perusahaan dan berita-berita terkini maka sebuah *web* yang dirancang dengan teliti serta menggunakan *Content Management Systems* merupakan jawaban yang tepat bagi *website* dinamis. Dengan adanya CMS perubahan atau penambahan informasi dalam *web* tidak lagi perlu di sentralisasi ke bagian tertentu (EDP misalnya), tanggung jawab *update website* dapat didelegasikan dengan mudah kepada departemen-departemen terkait, misalnya untuk meng*update* informasi berita terkini atau *event* dapat diserahkan ke bagian *Customer Service*, *update* produk dan harga dapat didelegasikan kepada bagian penjualan dan seterusnya (Saputra, 2015).

Intinya hampir dari seluruh bagian atau isi *website* dapat dikelola, dikelompokkan, ditambah dan dirubah sewaktu-waktu dengan mudah dan personil yang ditraining juga tidak perlu memiliki pengetahuan komputer khusus untuk dapat menggunakannya (Saputra, 2015).

II.3.3. Web Server

Web server merupakan software yang memberikan layanan data yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal

dengan browser *web* dan mengirimkan kembali hasilnya dalam bentuk halaman - halaman *web* yang umumnya berbentuk dokumen HTML.

Cara kerja dari *web server* sebenarnya sangat mudah kita pahami. Contoh paling mudah seperti berikut; kita akan membuka sebuah halaman *website*, yang biasanya berupa URL `http:` atau `www.wikipedia.org` atau `home.htm`. Kita akan mengetikkan URL tersebut di peramban atau browser kemudian menekan tombol enter, tanpa kita ketahui proses yang terjadi di belakang layar atau di dalam browser itu sendiri, maka akan muncullah halaman *website* di layar monitor komputer kita. Proses yang akan terjadi pada browser adalah browser akan membentuk koneksi dengan *web server*, meminta halaman *website* dan menerimanya. *Web server* kemudian mengecek permintaan tersebut apakah tersedia atau tidak.

Apabila tersedia, maka *web server* akan mengirimkan data kepada browser. Apabila permintaan tidak ditemukan atau terjadi *error* maka *web server* akan mengirimkan pesan *error* kepada *browser*. Pembentukan koneksi, permintaan data, penerimaan data dari browser ke *web server* diatur dalam sebuah kode RFC2616. RFC2616 mencantumkan status *web server* dalam bentuk kombinasi tiga angka yang memiliki arti berbeda-beda. Status ini muncul di peramban saat kita mengakses *web server* tertentu (Suhesti, 2014).

II.4. Metode Dijkstra

Menurut Penelitian Fitria (2013 : 614) menyatakan bahwa Algoritma yang ditemukan oleh *Dijkstra* untuk mencari path terpendek merupakan algoritma

yang lebih efisien dibandingkan algoritma *Warshall*, meskipun implementasinya juga lebih sukar. Misalkan G adalah graf berarah berlabel dengan titik-titik $V(G) = \{v_1, v_2, \dots, v_n\}$ dan path terpendek yang dicari adalah dari v_1 ke v_n . Algoritma *Dijkstra* dimulai dari titik v_1 . dalam iterasinya, algoritma akan mencari satu titik yang jumlah bobotnya dari titik 1 terkecil. Titik-titik yang terpilih dipisahkan dan titik-titik tersebut tidak diperhatikan lagi dalam iterasi berikutnya.

Misalkan :

$$V(G) = \{v_1, v_2, \dots, v_n\}$$

L = Himpunan titik-titik $V(G)$ yang sudah terpilih dalam jalur path terpendek.

$D(j)$ = Jumlah bobot path terkecil dari v_1 ke v_j .

$w(i,j)$ = Bobot garis dari titik v_i ke v_j .

$w^*(1,j)$ = Jumlah bobot path terkecil dari v_1 ke v_j

Secara formal, algoritma *Dijkstra* untuk mencari path terpendek adalah sebagai berikut :

1. $L = \{ \}$;

$$V = \{v_2, v_3, \dots, v_n\}.$$

2. Untuk $i = 2, \dots, n$, lakukan $D(i) = w(1, i)$

3. Selama $v_n \notin L$ lakukan:

a. Pilih titik $v_k \in V - L$ dengan $D(k)$ terkecil.

$$L = L \cup \{v_k\}$$

b. Untuk setiap $v_j \in V - L$ lakukan:

Jika $D(j) > D(k) + W(k,j)$ maka ganti $D(j)$ dengan $D(k) + W(k,j)$

4. Untuk setiap $v_j \in V$, $w^*(1, j) = D(j)$

Menurut algoritma di atas, path terpendek dari titik v_1 ke v_n adalah melalui titik-titik dalam L secara berurutan, dan jumlah bobot path terkecilnya adalah $D(n)$. Algoritma *Dijkstra* dinyatakan dalam *pseudo-code* berikut ini :

procedure *Dijkstra* (*input* m:matriks, a:simpul awal)

(

Mencari lintasan terpendek dari simpul awal a ke semua simpul lainnya

Masukan : matriks ketetanggaan (m) dari graf berbobot G dan simpul awal a

Keluaran : lintasan terpendek dari a ke semua simpul lainnya

)

Deklarasi

s_1, s_2, \dots, s_n :integer (tabel integer)

d_1, d_2, \dots, d_n :integer (tabel integer)

i, j, k: integer

Algoritma

(langkah 0 (Inisialisasi:)

for i = 1 to n do

$s_i = 0$

$d_i = m_{ai}$

endfor

(langkah 1 :)

Sa = 1 (karena simpul a adalah simpul asal lintasan terpendek, jadi simpul a sudah pasti terpilih

dalam lintasan terpendek)

Da o o (tidak ada lintasan terpendek dari simpul a ke a)

(langkah 2, 3, ..., n-1:)

For k = 2 to n-1 do

J = simpul dengan $s_j = 0$ dan d_j minimal

$S_j = S_j \cup \{j\}$ {simpul j sudah terpilih ke dalam lintasan terpendek}

{perbaharui tabel d}

For semua simpul I dengan $s_i = 0$ do

If $d_j + m_{ji} < d_i$ then

$d_i = d_j + m_{ji}$

Endif

Endfor

Endfor

)

II.4.1. Langkah – Langkah Metode *Dijkstra*

1. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga terhadap node lain (belum terisi)
2. Set semua node “Belum terjamah” dan set node awal sebagai “Node keberangkatan”
3. Dari node keberangkatan, pertimbangkan node tetangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan. Sebagai contoh, jika titik keberangkatan A ke B memiliki bobot jarak 6 dan dari B ke

node C berjarak 2, maka jarak ke C melewati B menjadi $6+2=8$. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.

4. Saat kita selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang telah terjamah sebagai “Node terjamah”. Node terjamah tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
5. Set “Node belum terjamah” dengan jarak terkecil (dari node keberangkatan) sebagai “Node Keberangkatan” selanjutnya dan lanjutkan dengan kembali ke step 3 (Hasbi Asyadiq ; 2012 : 1).

II.5. Open Street Map

OpenStreetMap (OSM) adalah alat untuk membuat serta berbagi informasi dalam peta. Siapapun dapat berkontribusi di OSM dan ribuan orang akan menambahkan proyek baru di OSM setiap harinya. Pengguna membuat peta di komputer mereka bukan di kertas, tetapi seperti yang nanti akan kita pelajari di panduan ini, menggambar peta di komputer tidak terlalu berbeda seperti menggambar di kertas. Kita tetap menggunakan simbol-simbol seperti garis untuk menggambarkan jalan atau persegi untuk bangunan. Tetapi, hal yang paling penting dari OSM adalah peta yang anda buat akan disimpan di internet dan siapapun dapat mengaksesnya kapanpun dan dimanapun dengan Gratis (Humanitarian ; 2016 : 6).

II.6. Pengertian PHP

PHP adalah sebuah bahasa pemrograman scripting untuk membuat halaman *web* yang dinamis. Walaupun dikenal sebagai bahasa untuk membuat halaman *web*, tapi php sebenarnya juga dapat digunakan untuk membuat aplikasi command line dan juga GUI. *Website* yang dibuat menggunakan PHP memerlukan software bernama *webserver* tempat pemrosesan kode PHP dilakukan. *Server web* yang memiliki software PHP parser akan memproses *input* berupa kode PHP dan menghasilkan *output* berupa halaman *web*. PHP bersifat terbuka dan *multiplatform*, karenanya dapat dijalankan di banyak merek *web server* (seperti apache dan IIS) (Zaki, 2014)

Berikut contoh kode PHP:

```
<?php
    Echo "Apa Kabar";
?>
```

Kode PHP juga dapat dikombinasikan dengan kode HTML, seperti contoh berikut:

```
<html>
<body>
    <?php
        Echo "Apa Kabar";
    ?>
    < atau body>
< atau html>
```

II.7. Pengertian Database

Secara sederhana *database* (basis data atau pangkalan data) dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Pengertian akses dapat mencakup pemerolehan data maupun manipulasi data seperti menambah serta menghapus data. Dengan memanfaatkan komputer, data dapat disimpan dalam media pengingat yang disebut *harddisk*. Dengan menggunakan media ini, keperluan kertas untuk menyimpan data dapat dikurangi. Selain itu, data menjadi lebih cepat untuk diakses terutama jika dikemas dalam bentuk *database* (Agustinus Mujilan ; 2012 : 23).

II.8. Pengertian MySQL

MySQL merupakan salah satu sistem *database* yang sangat handal karena menggunakan sistem SQL. Pada awalnya SQL berfungsi sebagai bahasa penghubung antara program *database* dengan bahasa pemrograman yang kita gunakan. Dengan adanya SQL maka para pemrogram jaringan dan aplikasi tidak mengalami kesulitan sama sekali di dalam menghubungkan aplikasi yang mereka buat. Setelah itu SQL dikembangkan lagi menjadi sistem *database* dengan munculnya MySQL. MySQL merupakan *database* yang sangat cepat, beberapa *user* dapat menggunakan secara bersamaan dan lebih lengkap dari SQL. MySQL merupakan salah satu software gratis yang dapat didownload melalui situsnya. MySQL merupakan sistem manajemen *database*, relasional sistem *database* dan software *open source*. (Sakur, 2015)

II.9. *Unified Modeling Language (UML)*

Menurut Evi Triandini (2012 : 15) UML menyediakan beberapa notasi dan artifak standar yang bisa digunakan sebagai alat komunikasi bagi para pelaku dalam proses analisis dan disain. Artifak di dalam UML didefinisikan sebagai informasi dalam berbagai bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat lunak. Contohnya adalah *source code* yang dihasilkan oleh proses pemrograman. Yang perlu diperhatikan untuk menjaga konsistensi antar artifak selama proses analisis dan disain adalah bahwa setiap perubahan yang terjadi pada satu artifak harus juga dilakukan pada artifak sebelumnya.

1. *Use Case Diagram*

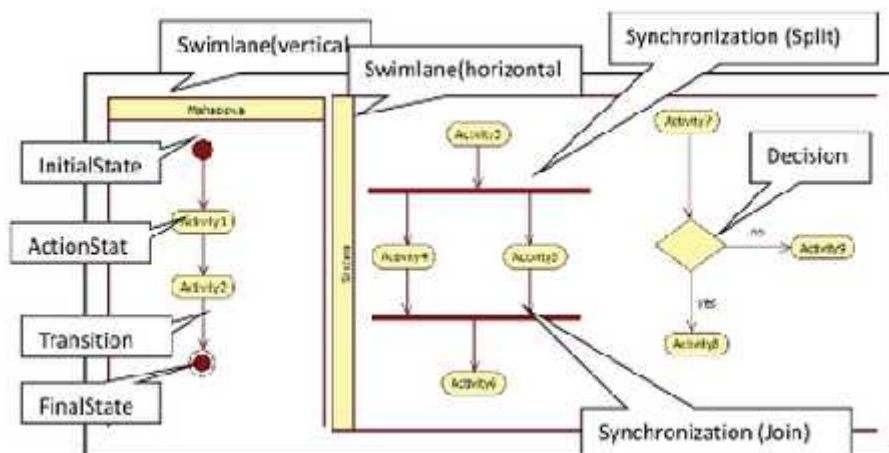
Use case adalah sebuah kegiatan yang dilakukan oleh sistem, biasanya dalam menanggapi permintaan dari pengguna sistem.



Gambar II.1. Notasi *Use Case Diagram*
(Sumber : Evi triandini ; 2012 : 17)

2. *Activity Diagram*

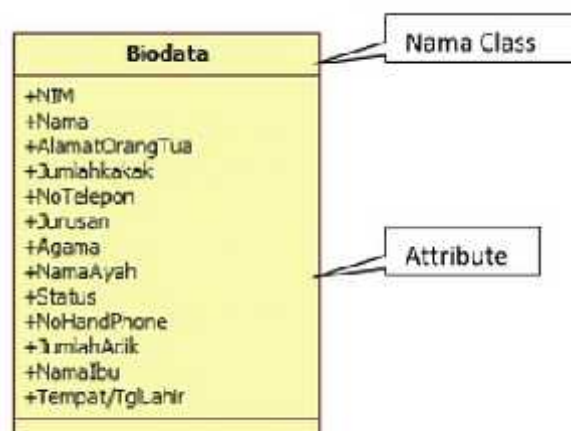
Activity diagram adalah sebuah diagram alur kerja yang menjelaskan berbagai kegiatan pengguna (atau sistem), orang yang melakukan masing-masing aktivitas, dan aliran sekuensial dari aktivitas-aktivitas tersebut.



Gambar II.2. Notasi *Activity* Diagram
(Sumber : Evi triandini ; 2012 : 37)

3. *Class* Diagram

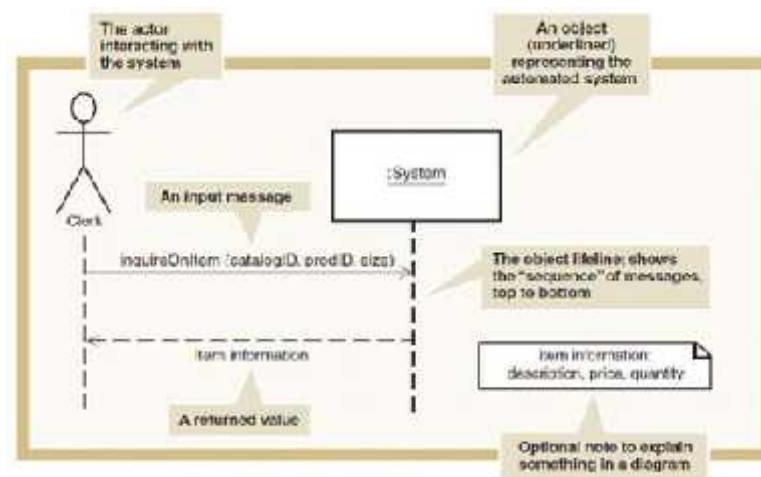
Class diagram bertujuan untuk mendokumentasikan dan menggambarkan kelas-kelas dalam pemrograman yang nantinya akan dibangun.



Gambar II.3. Notasi *Class* Diagram
(Sumber : Evi triandini ; 2012 : 49)

4. *Sequence* Diagram

Sequence diagram adalah diagram yang digunakan untuk mendefinisikan *input* dan *output* serta urutan interaksi antara pengguna dan sistem untuk sebuah *usecase*.



Gambar II.4. Notasi *Sequence* Diagram
(Sumber : Evi triandini ; 2012 : 71)