BAB II

TINJAUAN PUSTAKA

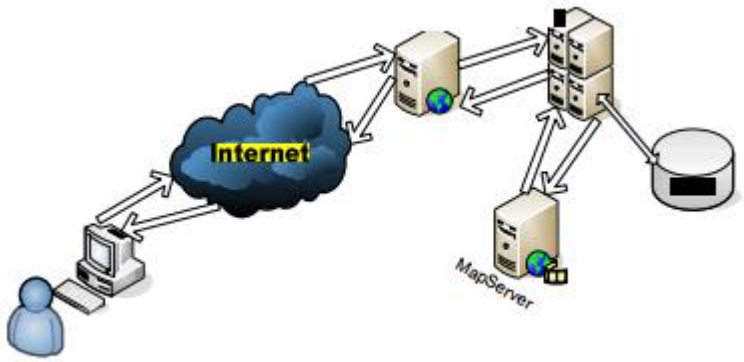
II.1. Sistem Informasi Geografis

Menurut (BramantiyoMarjuki ; 2014 : 1)Sistem Informasi Geografis (*GIS*) yang selanjutnya akan disebut SIG merupakan sistem informasi berbasis komputer yang digunakan untuk mengolah dan menyimpan data atau informasi geografis. Keunggulan utama dari SIG adalah SIG memungkinkan kita untuk melihat, memahami, menanyakan, menginterprestasi dan menampilkan data spasial dalam banyak cara, yang memperlihatkan hubungan, poladan trend secara spasial, dalam bentuk peta, globe, laporan dan grafik. SIG mampu membantu dalam pemecahan masalah dengan cara menampilkan data menggunakan cara yang mudah dipahami dan hasilnya mudah disebar luaskan.

Menurut (Hersa Farida Qoriani ; 2012 : 2) konsep *GIS* telah diperkenalkan di Indonesia sejak pertengahan tahun 1980-an, dan kini telah dimanfaatkan di berbagai bidang baik negeri maupun swasta. Kemampuan dasar dari *GIS* adalah mengintegrasikan berbagai operasi *basis data* seperti *quIery*, menganalisisnya, dan menyimpan serta menampilkannya dalam bentuk pemetaan berdasarkan letak geografisnya. Inilah yang membedakan *GIS* dengan sistem informasi lain. Komponen *GIS* terdiri atas *hardware, software*, data, dan *user*. Dengan adanya *GIS* diharapkan tersedia informasi yang cepat, benar dan akurat tantang keadaan di lingkungannya.

II.1.1. Arsitektur Sistem Informasi Geografis

Web GIS bisa dikatakan adalah sebuah *web mapping* yang berarti pemetaan internet, tetapi bukan memetakan internet. *Web mapping* memanfaatkan fungsi interaktifitas yang ada pada aplikasi SIG dalam bentuk *web*. Bentuk umum arsitektur berbasis peta di web dapat dilihat pada Gambar II.1. berikut :



Gambar II.1 Arsitektur GIS
(*Sumber : Fie Jannatin Aliyah ; 2012 : 8*)

Pada Gambar II.1, 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 ; 2012 : 8)

II.1.2. *Data Spasial*

Sebagian besar data yang akan ditangani dalam SIG merupakan *data spasial*, data yang berorientasi geografis. Data ini memiliki sistem koordinat tertentu sebagai dasar referensinya dan mempunyai dua bagian penting yang

berbeda dari data lain, yaitu informasi lokasi (*spasial*) dan informasi deskriptif (atribut) yang dijelaskan berikut ini:

1. Informasi lokasi (*spasial*), berkaitan dengan suatu koordinat baik koordinat geografi (lintang dan bujur) dan koordinat XYZ, termasuk diantaranya informasi data dan proyeksi.
2. Informasi deskriptif (atribut) atau informasi nonspasial, suatu lokasi yang memiliki beberapa keterangan yang berkaitan dengannya. Contoh jenis vegetasi, populasi, luasan, kode pos, dan sebagainya.

II.1.3. *Format Data Spasial*

Secara sederhana *format* dalam bahasa komputer berarti bentuk dan kode penyimpanan data yang berbeda antara *file* satu dengan lainnya. Dalam SIG, *data spasial* dapat direpresentasikan dalam dua *format*, yaitu:

a. Data vektor

   Data vektor merupakan bentuk bumi yang direpresentasikan ke dalam kumpulan garis, area (daerah yang dibatasi oleh garis yang berawal dan berakhir pada titik yang sama), titik dan nodes (titik perpotongan antara dua buah garis).

b. Data raster

   Data raster (disebut juga dengan sel grid) adalah data yang dihasilkan dari sistem penginderaan jauh. Pada data raster, obyek geografis direpresentasikan sebagai struktur sel *grid* yang disebut dengan *pixe*l (*picture element*) (Mohd. Ichsan ; 2012 : 51).

II.2. Metode *Haversine*

Menurut (WahyuniEka Sari ; 2013 : 72) posisi di bumi dapat direpresentasikan dengan posisi garis lintang (*latitude*) dan bujur (*longitude*). Untuk menentukan jarak antara dua titik di bumi berdasarkan letak garis lintang dan bujur, ada beberapa rumusan yang digunakan. Semua rumusan yang digunakan berdasarkan bentuk bumi yang bulat (*sphericalearth*) dengan menghilangkan faktor bahwa bumi itu sedikit elips (*elipsodialfactor*).

$\Delta lat = lat2 - lat1$

$\Delta long = long2 - long1$

$a = \sin2(\Delta lat/2) + \cos(lat1).\cos(lat2).\sin2(\Delta long/2)$

$c = 2.atan2(\sqrt{a}, \sqrt{(1-a)})$

$d = R.c$

Keterangan :

R = jari-jari bumi sebesar 6371(km)

Δlat = besaran perubahan latitude

Δlong = besaran perubahan longitude

c = kalkulasi perpotongan sumbu

d = jarak (km)

II.4. Pengertian PHP

Menurut (Ali Zaki ; 2015 : 2) *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* 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 d*an *IIS (Internet Information Service).*

```php
private function getData() {
    $data = new stdClass();

    $results = $this->mysql->query( 'SELECT * FROM hit_counter' );
    if ( $results->rowCount() === 0 ) {


        $data->total = 0;
        $data->unique = 0;

        $stmt = $this->mysql->prepare( 'INSERT INTO hit_counter( `total_hits`, `unique_hits` ) VALUES(:total, :unique)' );
        $stmt->bindParam( ':total', $data->total );
        $stmt->bindParam( ':unique', $data->unique );
        $stmt->execute();

    } else {
        $rows = $results->fetchAll( PDO::FETCH_OBJ );
        $data->total = $rows[0]->total_hits;
        $data->unique = $rows[0]->unique_hits;
    }

    return $data;
}

private function isNewVisitor() {
    return !array_key_exists( 'visited', $_SESSION ) && $_SESSION['visited'] !== true;
}

private function visit() {

    $this->mysql->query( "UPDATE hit_counter SET total_hits = total_hits + 1" );

    if ( $this->isNewVisitor() ) {
        $this->mysql->query( "UPDATE hit_counter SET unique_hits = unique_hits + 1");
        $_SESSION['visited'] = true;
    }
```

II.5. Pengertian Database

Menurut (AgustinusMujilan ; 2012 : 23) secara sederhana *database*(basis data/pangkalan data) dapatdiungkapkan sebagai suatu pengorganisasian data dengan bantuankomputer yang memungkinkan data dapat diakses dengan mudah dancepat. Pengertian akses dapat mencakup pemerolehandata maupun pemanipulasian data seperti menambah sertamenghapus data. Dengan memanfaatkan komputer, data dapatdisimpan dalam media pengingat yang

disebut *harddisk. Dengan*menggunakan media ini, keperluan kertas untuk menyimpan datadapat dikurangi. Selain itu, data menjadi lebih cepat untuk diaksesterutama jika dikemas dalam bentuk *database*.

II.6. Pengertian MySQL

Menurut (Stendy B. Sakur ; 2015 : 57) *MySQL* merupakan salahsatu 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 sangatcepat, beberapa user dapat menggunakan secara bersamaan dan lebih lengkap dari *SQL*. *MySQL* merupakan salah satu *software* gratis yang dapat di-*download* melaluisitusnya. *MySQL* merupakan sistem manajemen *database*, relasional sistem *database* dan *software open source*.

II.7. UML (*Unified Modeling Language*)

Menurut (Sri Dharwiyanti 2013 : 2) *Unified Modelling Language (UML)* adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. *UML* menawarkan sebuah standar untuk merancang model sebuah sistem.

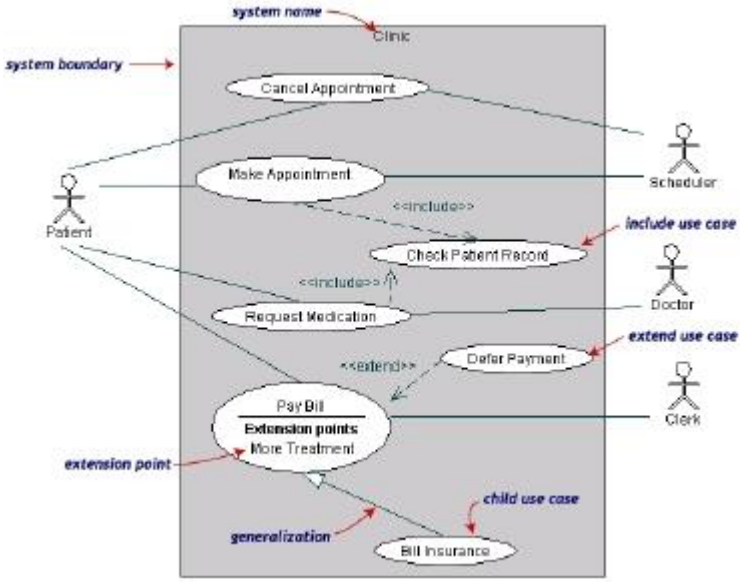
1. *Use Case* Diagram

Menurut (Sri Dharwiyanti 2013 : 4) *Use case diagram* menggambarkan

fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login kesistem, meng-*create* sebuah daftarbelanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan system untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem.

Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagiand ari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *usecase* yang meng-*include*di eksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*.

Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

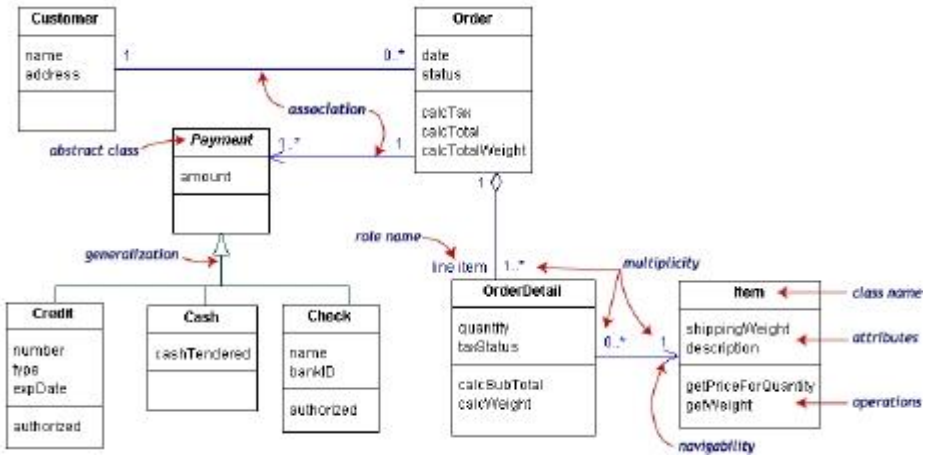


Gambar II.1. *Usecase* Diagram
(Sumber : Sri Dharwiyanti ; 2013 : 5)

2. *Class* Diagram

Menurut (Sri Dharwiyanti 2013 : 5) *Class Diagram* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layana nuntuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

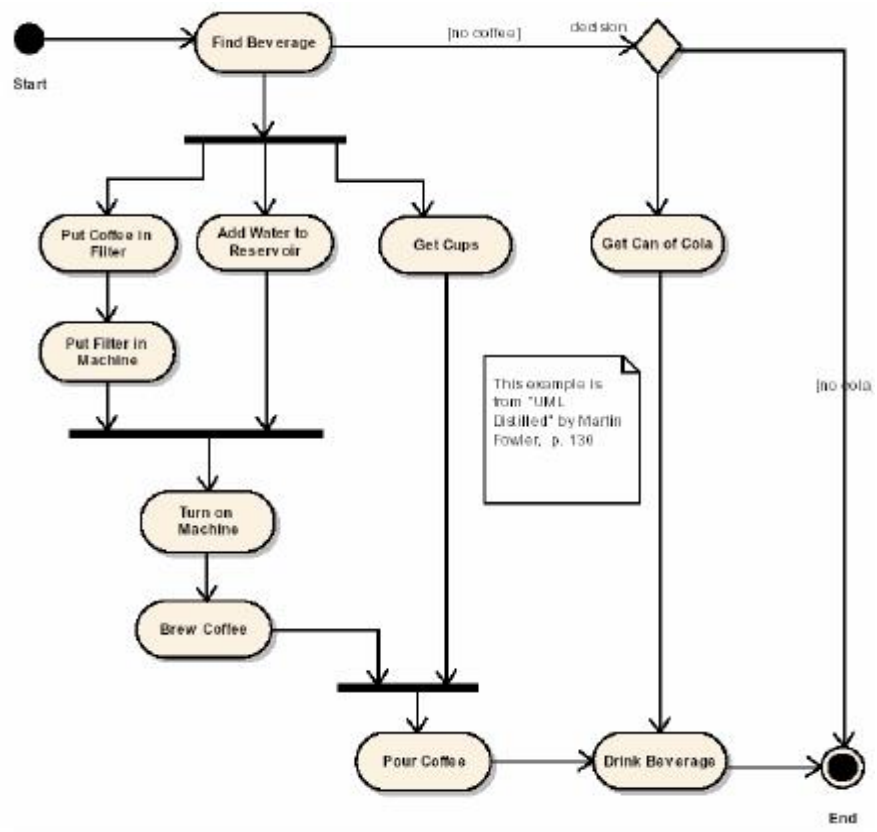


Gambar II.2. *Class* Diagram
(Sumber : Sri Dharwiyanti ; 2013 : 6)

3. *Activity* Diagram

Menurut (Sri Dharwiyanti 2013 : 7) *Activity diagrams* menggambarkan berbagaialir 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 manase bagian 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, tetapil ebihmenggambarkan proses-proses danjalur-jaluraktivitasdari level atassecaraumum.

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



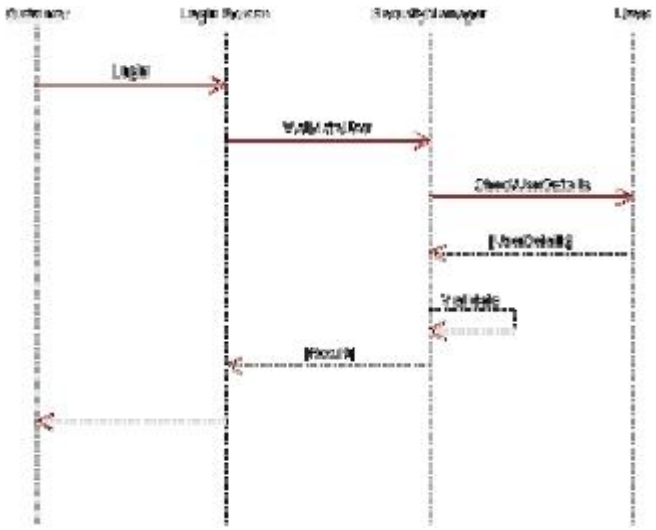
Gambar II.3. *Activity*Diagram
(Sumber : Sri Dharwiyanti ; 2013 : 8)

4. *Sequence* Diagram

Menurut (Sri Dharwiyanti 2013 : 8) *Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah

Yang dilakukan sebagai respons dari sebuah *even t*untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadisecara internal dan *outpu t*apa yang dihasilkan.

Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objeklainnya. Pada *fase* desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*.*Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*.



Gambar II.4. *Sequence*Diagram
(Sumber : Sri Dharwiyanti ; 2013 : 9)