**BAB II**

**KAJIAN PUSTAKA DAN KERANGKA PIKIR**

1. **Kajian Pustaka**

Kajian pustaka adalah suatu kegiatan penelitian yang bertujuan melakukan  kajian secara sungguh-sungguh tentang teori-teori dan konsep-konsep yang berkaitan dengan topik yang akan diteliti sebagai dasar dalam melangkah pada tahap penelitian selanjutnya. Teori dan konsep yang dikaji digunakan untuk memperjelas dan mempertajam ruang lingkup dan konstruk variabel yang akan di teliti, sebagai dasar perumusan hipotesis dan penyusunan instrumen penelitian, dan sebagai dasar dalam membahas hasil penelitian untuk digunakan untuk memberikan saran dalam upaya pemecahan topik permasalahan.

Pada bab ini akan dijelaskan teori-teori serta konsep definisi yang berkaitan dengan penelitian pengembangan sistem informasi geografis.

**Visualisasi**

Menurut Shneiderman (1998), visualisasi merupakan penggunaan komputer pendukung, penggambaran data visual interaktif untuk memperkuat pengamatan. Dan Informasi berarti item-item, *entity*-*entity*, hal-hal yang tidak memiliki korespondensi fisik secara langsung. Dengan kata lain visualisai informasi itu sendiri berarti rekayasa dalam pembuatan gambar, diagram, grafik atau animasi untuk penampilan suatu informasi. Secara umum, visualisasi dalam bentuk gambar baik yang bersifat abstrak maupun nyata telah dikenal sejak awal dari peradaban manusia.

Pengertian visualisasi yang dimaksud dalam penelitian ini adalah segala bentuk rekayasa data menjadi bentuk grafik dan peta tematik dengan tujuan untuk interpretasi informasi.

**Sistem Informasi Geografis**

Sistem Informasi Geografis (SIG) adalah sistem informasi yang berdasar pada data keruangan dan merepresentasikan objek di bumi. Dalam SIG sendiri teknologi informasi merupakan perangkat yang membantu dalam menyimpan data, memproses data, menganalisa data, mengelola data dan menyajikan informasi. SIG merupakan sistem yang terkomputerisasi yang menolong dalam me-*maintain* data tentang lingkungan dalam bidang geografis (De Bay, 2002). SIG selalu memiliki relasi dengan disiplin keilmuan Geografi, hal tersebut memiliki hubungan dengan disiplin yang berkenaan dengan yang ada di permukaan bumi, termasuk di dalamnya adalah perencanaan dan arsitektur wilayah (Longley, 2001).

Data dalam SIG terdiri atas dua komponen yaitu data spasial yang berhubungan dengan geometri bentuk keruangan dan data *attribute* yang memberikan informasi tentang bentuk keruangannya (Chang, 2002). Menurut pendapat Peter A. Burrough (1998), SIG adalah sekumpulan fungsi-fungsi terorganisasi yang menyediakan tenaga-tenaga profesional yang berpengalaman untuk keperluan penyimpanan, *retrieval*, manipulasi dan penayangan hasil yang didasarkan atas data berbasis geografis. Aronoff (1989) menyatakan bahwa SIG adalah sekumpulan komponen yang dilakukan secara manual atau berbasis komputer yang merupakan prosedur-prosedur yang digunakan untuk keperluan *store* dan pemanipulasian data bereferensi geografis. Menurut pendapat tersebut dapat dipahami bahwa, isi aktifitas pada bidang SIG merupakan integrasi dari beragam bidang keilmuan yang didasarkan pada peruntukan aktifitas SIG tersebut dilakukan. Implementasi dari pelaksanaan kegiatan tersebut tidak selalu mengacu pada penyertaan komputer sebagai salah satu elemen pada sistem informasi.

**Data Spasial**

Data spasial adalah data yang bereferensi geografis atas representasi objek di bumi. Data spasial pada umumnya berdasarkan peta yang berisikan interprestasi dan proyeksi seluruh fenomena yang berada di bumi. Fenomena tersebut berupa fenomena alamiah dan buatan manusia. Pada awalnya, semua data dan informasi yang ada di peta merupakan representasi dari objek di muka bumi.

Sesuai dengan perkembangan, peta tidak hanya merepresentasikan objek-objek yang ada di muka bumi, tetapi berkembang menjadi representasi objek diatas muka bumi (di udara) dan dibawah permukaan bumi. Data spasial memiliki dua jenis tipe yaitu vektor dan raster. Model data vektor menampilkan, menempatkan, dan menyimpan data spasial dengan menggunakan titik-titik, garis-garis atau kurva, atau poligon beserta atribut-atributnya. Model data Raster menampilkan, dan menyimpan data spasial dengan menggunakan struktur matriks atau piksel–piksel yang membentuk *grid*. Pemanfaatan kedua model data spasial ini menyesuaikan dengan peruntukan dan kebutuhannya.

**Data Vektor**

Model data vektor adalah yang dapat menampilkan, menempatkan, dan menyimpan data spasial dengan menggunakan titik-titik, garis atau kurva dan poligon beserta atribut-atributnya (Prahasta, 2001). Bentuk-bentuk dasar representasi data spasial ini, di dalam sistem model data vektor, didefinisikan oleh sistem koordinat kartesian dua dimensi (x, y).

Di dalam model data spasial vektor, garis-garis atau kurva (busur atau *arcs*) merupakan sekumpulan titik-titik terurut yang dihubungkan (Prahasta, 2001). Poligon akan terbentuk penuh jika titik awal dan titik akhir poligon memiliki nilai koordinat yang sama dengan titik awal. Sedangkan bentuk poligon disimpan sebagai suatu kumpulan *list* yang saling terkait secara dinamis dengan menggunakan *pointer*/titik.

**Data Raster**

Objek di permukaan bumi disajikan sebagai elemen matriks atau sel-sel grid yang homogen. Model data Raster menampilkan, menempatkan dan menyimpan data spasial dengan menggunakan struktur matriks atau piksel-piksel yang membentuk *grid* (Prahasta, 2001). Tingkat ketelitian model data raster sangat bergantung pada resolusi atau ukuran pikselnya terhadap objek di permukaan bumi. *Entity* spasial raster disimpan di dalam layers yang secara fungsionalitas di relasikan dengan unsur–unsur petanya (Prahasta, 2001).

Satuan elemen data raster biasa disebut dengan pixel, elemen tersebut merupakan ekstrasi dari suatu citra yang disimpan sebagai digital number (DN) (De Bay, 2000). Meninjau struktur model data raster identik dengan bentuk matriks. Pada model data raster, matriks atau array diurutkan menurut koordinat kolom (x) dan barisnya (y) (Prahasta, 2001).

**Pemrosesan Spasial**

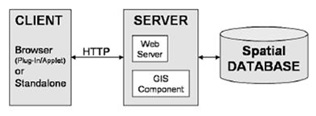
Pengelolaan, pemrosesan dan analisis data spasial biasanya bergantung dengan model datanya. Pengelolaan, pemrosesan dan analisis data spasial memanfaatkan pemodelan SIG yang berdasar pada kebutuhan dan analitiknya. Analitik yang berlaku pada pemrosesan data spasial seperti *overlay, clip, intersect, buffer, query, union, merge*; yang mana dapat dipilih ataupun dikombinasikan.

Pemrosesan data spasial seperti dapat dilakukan dengan teknik yang disebut dengan geoprocessing (ESRI, 2002), pemrosesan tersebut antara lain:

1. *overlay* adalah merupakan perpaduan dua layer data spasial;
2. *clip* adalah perpotongan suatu area berdasar area lain sebagai referensi;
3. *intersection* adalah perpotongan dua area yang memiliki kesamaan karakteristik dan kriteria;
4. *buffer* adalah menambahkan area di sekitar objek spasial tertentu;
5. *query* adalah seleksi data berdasar pada kriteria tertentu;
6. *union* adalah penggabungan/kombinasi dua area spasial beserta atributnya yang berbeda menjadi satu;
7. *merge* adalah penggabungan dua data berbeda terhadap *feature* spasial;
8. *dissolve* adalah menggabungkan beberapa nilai berbeda berdasar pada atribut tertentu.

**SIG Berbasis *Web***

SIG berbasis *web* bisa dikatakan adalah sebuah *web* *mapping* yang berarti pemetaan *internet*, tetapi bukan memetakan *internet*, dan tidak berarti hanya menampilkan peta (yang berupa gambar yang statis) ke dalam sebuah situs *internet*. Jika hanya menampilkan peta statis pada sebuah situs, tidak perbedaan antara *web* *mapping* dengan peta yang ada pada media tradisional lainya. *Web* *mapping* memanfaatkan fungsi interaktivitas yang ada pada aplikasi GIS dalam bentuk *web*.



Gambar 1. Interaksi antara klien dengan *server*

Pada gambar di atas, interaksi antara klien dengan *server* berdasar skenario permintaan dan respon. *Web* *browser* di sisi klien mengirimkan permintaan ke *web* *server*. Karena *web* *server* tidak memiliki kemampuan pemrosesan peta, permintaan yang berkaitan dengan pemrosesan peta akan diteruskan oleh *web* *server* ke *server* aplikasi dan *Mapserver*. Hasil pemrosesan akan dikembalikan lagi melalui *web* *server*, terbungkus dalam bentuk berkas *HTML* atau *applet*.

Kemampuan inilah yang membedakan SIG dengan sistem informasi lainnya yang membuatnya menjadi berguna untuk berbagai kalangan untuk menjelaskan kejadian, merencanakan strategi, dan memprediksi apa yang akan terjadi. Kemampuan SIG antara lain memetakan letak, memetakan kuantitas, memetakan kerapatan (*densities*), memetakan perubahan, dan memetakan apa yang ada di dalam dan di luar suatu area.

**Peta**

Peta merupakan gambaran wilayah geografis, bagian permukaan bumi yang disajikan dalam berbagai cara yang berbeda, mulai dari peta konvensional yang tercetak hingga peta digital yang tampil di layar komputer. Peta dapat digambarkan dengan berbagai gaya, masing-masing menunjukkan permukaan yang berbeda untuk subjek yang sama untuk memvisualisasikan dunia dengan mudah, informatif dan fungsional.

Peta berbasis komputer (digital) lebih serba guna dan dinamis karena bisa menunjukkan banyak *view* yang berbeda dengan subjek yang sama. Peta ini juga memungkinkan perubahan skala, animasi gabungan, gambar, suara, dan bisa terhubung ke sumber informasi tambahan melalui *internet*. Peta digital dapat di*update* ke peta tematik baru dan bisa menambahkan detail informasi geografi lainnya. (Denny Carter, Irma Agtrisari, 2003).

* Peta Tematik

Peta Tematik adalah peta yang menyajikan tema tertentu dan untuk kepentingan tertentu (land status, penduduk, transportasi dll.) dengan menggunakan peta rupa bumi yang telah disederhanakan sebagai dasar untuk meletakkan informasi tematiknya.

* Peta *Choropleth*

Peta *choropleth* adalah peta yang menggunakan arsiran/gradasi warna untuk menunjukkan tingkat variabilitas di suatu daerah. Peta *choropleth* diciptakan pertama kali oleh Baron Pierre Charles Dupin pada tahun 1826 untuk memetakan distribusi penduduk buta huruf di Perancis dengan memberikan gradasi warna dari hitam ke putih (Friendly, 2009).

Pemberian gradasi warna dalam peta choropleth dapat dilakukan dengan beberapa metode klasifikasi. Setiap metode menghasilkan peta choropleth yang mempunyai ciri khas dan keunikan masing-masing. Adapun metode yang dapat digunakan adalah sebagai berikut.

1. *Equal Interval* : Metode klasifikasi data dengan pembagian nilai variabel ke dalam beberapa kelas yang memuat interval nilai yang sama.
2. *Natural Breaks* : Metode klasifikasi yang dirancang untuk menentukan susunan terbaik dari nilai -nilai ke dalam kelas yang berbeda. Hal ini dilakukan dengan meminimalkan variasi dalam kelas dan memaksimalkan variasi antarkelas.
3. *Quantile :* Metode klasifikasi yang mendistribusikan sekelompok nilai atribut ke dalam kelas-kelas yang mengandung jumlah observasi yang sama.
4. *Arithmetic progression :* Metode klasifikasi yang membagi kelas dengan jarak antar kelas membentuk sebuah deret aritmatika.
5. *geometric progression :* Metode klasifikasi yang membagi kelas dengan jarak antar kelas membentuk sebuah deret geometri.
6. *standard deviation :* Metode klasifikasi yang mengunakan rata-rata dan standar deviasi sebagai batas antar kelas.

***Geoserver***

*Geoserver* adalah sebuah perangkat lunak open source yang dibangun dengan menggunakan java yang memungkinkan pengguna untuk menampilkan dan memanipulasi data geospasial. Geo*server* dirancang untuk *interoperability* yaitu menerbitkan data dari semua sumber data spasial dengan menggunakan standar terbuka.

Sebagai *project* berbasis komunitas, geo*server* dikembangkan, diuji, dan didukung oleh berbagai kelompok individu dan organisasi dari seluruh dunia. *Geoserver* adalah implementasi dari Open Geospatial Consortium (OGC) *Web Feature Service* (WFS) dan *Web Coverage Service* (WCS) standar, serta *high performance Web Map Service* (WMS).

***SHP (Shapefile)***

Format data *SHP* atau *shapefile* merupakan format data vektor yang terkenal untuk *software* Sistem Informasi Geografis (SIG). *SHP* adalah format data vektor yang digunakan untuk menyimpan lokasi, bentuk, dan atribut dari fitur geografis. Format data *SHP* disimpan dalam satu set *file* terkait dan berisi dalam satu kelas fitur. Format data vektor ini berisi tentang data referensi geografis yang didefinisikan sebagai objek tunggal seperti jalan, sungai, *landmark*, dan kode pos. Data fitur dan atribut akan disimpan dalam satu SHP. Banyak aplikasi SIG yang bersifat *opensource* ataupun proprietary dapat bekerja dengan shapefile. *File* ini meyimpan data geometri seperti *point*, *line* dan poligon sehingga dapat di-*render* pada *map* SIG.

Shapefile merupakan kumpulan beberapa file dengan tiga ekstensi utama yang *mandatory*/wajib yaitu \*.shp, \*.shx, \*.dbf serta beberapa tambahan/opsional *file* yang lain. Satu set *shapefile* ditunjukkan dengan nama *file* yang sama dengan ekstensi yang berbeda, misal jateng.shp, jateng.shx dan jateng.dbf serta beberapa *file* opsional lain.

* .shp merupakan *shape* format, menyimpan data *feature geometry*;
* .shx merupakan *shape* *index* format, *index* dari *fetaure* *geometry* sehingga memudahkan atau mempercepat proses pencarian;
* .dbf adalah attribute format, berisi *table* *attribute* dari tiap *feature* dalam *dBase IV format*.

Ukuran data *SHP* dan file komponen *DBF* tidak dapat melebihi 2 GB (atau 231 *bit*) sekitar 70 juta fitur titik yang terbaik. Jumlah maksimum fitur untuk jenis geometri lainnya bervariasi tergantung pada jumlah simpul yang digunakan. Kurang mendukung untuk nama *field* *Unicode* atau tempat penyimpanan *field*, panjang maksimum nama *field* adalah 10 karakter, dan jumlah maksimum dari *field* adalah 255.

*Shapefile* adalah format data vektor geospasial untuk *software* SIG yang dikembangakan oleh *ESRI* (*Environmental System Research Institute*) dengan spesifikasi yang terbuka untuk kepentingan *interoperabilitas* antar *software* SIG.

*Software* yang bisa mengolah (*input*) format data *SHP* antara lain *ArcGIS*, *ArcView*, *MapInfo*, *ERDAS*, *Global Mapper*.

***HTML (HyperText Mark-up Language)***

*HTML* (*HyperText Mark-up Language*) adalah bahasa dasar untuk *web scripting* bersifat *client side* yang memungkinkan untuk menampilkan informasi dalam bentuk teks, grafik, serta multimedia dan juga untuk menghubungkan antar tampilan *web page* atau yang lebih dikenal dengan *Hyperlink*.

Tidak diperlukan suatu program editor khusus untuk menggunakan kode perintah-perintah *HTML*, dapat menggunakan *Notepad*, *Notepad*++ ataupun editor lainnya yang berbasis *GUI* (*Graphical User Interface*) seperti *Microsoft Front Page*, *Dreamweaver CS3* dan sebagainya. Sebagai seorang pengembang aplikasi *web* maka kemampuan penguasaan kode-kode *HTML* sangat diperlukan, karena *HTML* sebagai dasar/tumpuan dari suatu aplikasi *web*.

***JavaScript***

*JavaScript* adalah bahasa *scripting* yang digunakan oleh milyaran halaman *web*, dan mampu bekerja di banyak *browser* ternama seperti *Internet Explorer*, *Firefox*, *Chrome*, *Opera* dan *Safari*.

*JavaScript* merupakan implementasi dari standar *Script Language* *ECMA*. *ECMA*-262 adalah standar resmi *JavaScript*.

Bahasa ini pertama kali diimplementasikan oleh *Netscape Communications Corp pada Netscape Navigator 2 beta* (1995). *JavaScript* berbeda dengan bahasa *Java* (dikembangkan pada 1990-an di Sun Microsystems). Namun dua bahasa tersebut dapat saling bekerja sama dengan baik. Program *JavaScript* sisi *client* , atau *script,* dapat ditanamkan langsung dalam sumber *HTML* halaman *web.*

***PHP (Hypertext Preprocessor)***

*PHP* adalah Bahasa pemograman yang memungkinkan para *web developer* untuk membuat aplikasi *web* yang dinamis dengan cepat. *PHP* juga merupakan salah satu bahasa *script* yang tersedia secara bebas dan masih memungkinkan untuk dikembangkan lebih lanjut.

*PHP* dapat berjalan pada semua jenis sistem operasi, antara lain pada *Linux* dan *varian Unix* (HP-UX, Solaris dan OpenBSD), pada *Microsoft Windows*, *Mac* dan masih banyak lagi. Selain itu *PHP* juga dapat berjalan pada beberapa jenis *web-server* antara lain *Apache*, *Microsoft IIS*, *personal webserver*, *Netscape* dan *Iplanet* *Server*, *Caudium*, *Xitami*, *Omnihttpd* dan masih banyak lagi.

*Output* yang dihasilkan *PHP* bukan hanya *HTML*, namun juga dalam bentuk gambar, *file PDF*, serta gambar animasi menggunakan *Libswf* dan *Ming*. *Output* yang lain dengan jenis teks dapat berupa file *XHTML* dan *XML*.

***Yii Framework***

*Yii* adalah *framework* (kerangka kerja) *PHP* berbasis-komponen, berkinerja tinggi untuk pengembangan aplikasi *Web* berskala-besar. *Yii* menyediakan *reusability* maksimum dalam pemrograman *Web* dan mampu meningkatkan kecepatan pengembangan secara signifikan.

*Yii* merupakan kepanjangan dari “*Yes It Is*”. *Yes it is* memiliki makna bahwa *Yii* mampu dan tepat anda pilih untuk mengerjakan *project*. *Yii* merupakan *free* *open* *source* *PHP* *framework* terbaru berbasis komponen dengan performa tinggi untuk mengembangkan aplikasi *web* bersekala besar. Ia menyediakan resuabilitas maksimum dalam pemrograman *Web* dan bisa mengakselerasi proses pengembangan secara signifikan. Nama *Yii* (dieja sebagai/i:/) singkatan dari *easy*, *efficient* dan *extensible* (mudah, efisien, dan bisa diperluas).

Fitur yang ada pada *Yii* menggunakan pola *MVC*, pola standar pengembangan aplikasi yang memisahkan tampilan, *logic program*, dan *model-*nya. Menggunakan *database abstraction layer* seperti *Data Access Objects (DAO)* dan *Active Record* untuk memudahkan interaksi antar *database*. Terintegrasi dengan *JQuery Javascript Framework*. Walaupun menggunakan *Jquery* sebagai *internal javascript library*, namun *Yii* dapat menggunakan librari lain tanpa terjadi bentrok. Mendukung *Internationalization (I18N)* and *localization (L10N)* untuk memudahkan pengembangan aplikasi dalam multi bahasa dan lokasi seperti penggunaan waktu dan tanggal. Memiliki layer *cache* untuk *cache* data, halaman, sebagian, dan keseluruhan aplikasi sehingga dapat meningkatan performa dengan beragam pilihan media *cache*. Penggunaan media *cache* seperti database, *APC*, mem*cache*, dan sebagainya mudah diatur tanpa melakukan perubahan besar pada kode. Fitur penanganan *eror* dan *logging*, sehingga memudahkan pengembangan dalam melakukan *debuging* aplikasi dalam masa pengembangan aplikasi. Penggunaan tema, memudahkan pengembangan aplikasi dalam merancang tampilan aplikasi. *Console*, penggunaan perintah pada console untuk melakukan beragam perintah otomatis seperti men*generate* struktur dasar aplikasi, model, *crud*, dan sebagainya. Dukungan *Authentication* dan *authorization* internal sehingga memudahkan pengembangan aplikasi dengan fitur autentikasi. Widget, semacam kontrol yang memiliki fungsi seperti *auto* *complete* *datapicker* *table* dan lain-lain. Menggunakan *Jquery* sebagai *javascript* *client* *side*. *Form* *input* dan validasi memudahkan pengembangan untuk bekerja dengan *form* pada aplikasi dan melakukan validasi *input* dari *form*. *Modular* dan mudah ditambahkan dengan dukungan extensi dan komponen tambahan sehingga beragam fitur tambahan dengan mudah dimasukkan.

Kelebihan *Framework* *Yii* ada beberapa keuntungan yang dapat peneliti ambil dalam penggunaan *framework PHP Yii* berikut. Keuntungannya antara lain adalah:

1. *Yii* adalah salah satu *framework* yang sangat ringan dan dilengkapi dengan solusi *caching* yang memuaskan.
2. *Yii* sangat cocok untuk pengembangan aplikasi dengan lalu lintas-tinggi, seperti portal, *forum*, sistem manajemen konten (*CMS*), dan sistem e-*commerce*.
3. *Yii* didokumentasikan dengan jelas, efisien, dan kaya-fitur.

*Yii* didesain dengan hati-hati dari awal agar sesuai untuk pengembangan aplikasi *Web* secara serius. *Yii* bukan berasal dari produk pada beberapa projek maupun konglomerasi pekerjaan pihak-ketiga. *Yii* adalah hasil dari pengalaman kaya para pembuat pada pengembangan aplikasi *Web* dan investigasi serta refleksi kerangka kerja pemrograman *Web* paling populer dan aplikasi. *Yii* adalah kerangka kerja pemrograman umum *Web* yang bisa dipakai untuk mengembangkan semua jenis aplikasi *Web*.

*Yii* saat ini memiliki dua versi mayor, yakni: *Yii 1.1* dan *Yii 2.0*. *Yii* versi 1.1 yang merupakan versi lama, saat ini hanya dalam tahap pemeliharaan semata. Sedangkan *Yii* 2.0 adalah hasil penulisan ulang dari *Yii* 1.1 dengan mengadopsi teknologi dan protokol terbaru seperti *Composer, PSR, namespace, trait* dan sebagainya. *Yii* versi 2.0 adalah generasi terbaru *Yii* dan akan menerima setiap usaha pengembangan utama (sekarang). Panduan ini ditujukan untuk *Yii* versi 2.0.

*Yii* 2.0 membutuhkan php versi 5.4.0 atau yang lebih tinggi. Untuk mengunakan *Yii* diperlukan pengetahuan dasar tentang pemrograman *object*-*oriented* (OOP), mengingat *Yii* adalah murni merupakan *framework* berbasis *OOP*.

***GeoJSON***

*JSON* (*Javascript Object Notation*) adalah format pertukaran data yang ringan dan mudah digunakan dibandingkan dengan *XML*. *JSON* lebih mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. *JSON* merupakan bagian dari bahasa pemrograman *Javascript* tetapi memiliki format teks yang tidak bergantung pada bahasa pemrograman tertentu.

Sama halnya dengan *JSON*, *GeoJSON* juga merupakan format pertukaran data, namun dikhususkan untuk struktur data geografis. *GeoJSON* dapat merepresentasikan jenis data geometri tertentu, suatu atribut, ataupun kumpulan dari atribut. Jenis data geometri yang didukung oleh Geo*JSON* antara lain *Point*, *LineString*, *Poligon*, dan *Geometry* *Collection*. Sedangkan atribut dalam struktur Geo*JSON* berisi objek geometri beserta *properties* lainnya dari objek tersebut.

***MySQL***

*SQL* (*Structured Query Language*) merupakan sebuah bahasa relasional yang berisi pernyataan yang digunakan untuk memasukkan, mengubah, menghapus, memilih dan melindungi data (Prihatna, 2005). *SQL* bukan *database* aplikasi, tetapi lebih berarti dengan suatu bahasa yang digunakan untuk mengajukan pertanyaan ke dalam database berupa pengguna *SQL*.

Database sistem yang memiliki konsep sama dengan *SQL*, adalah *Postgres* dan My*SQL*, dimana *database* tersebut bisa didapatkan gratis atau dengan harga yang murah. *MySQL* adalah *server* *multithreaded*, sehingga sangat memungkinkan *daemon* untuk meng*-handle* permintaan layanan secara stimultan. Model koneksi dengan *protocol TCP-IP* membuat akses ke *server* *database* lebih cepat jika dibandingkan dngan menggunakan *mapping* *drive*.

***Flowchart***

*Flowchart* adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowchart* juga merupakan simbol yang merepresentasikan proses-proses dalam suatu aplikasi. *Flowchart* menolong analis dan programmer untuk memecahkan masalah ke dalam segmen-segmen yang lebih kecil dan menolong dalam menganalisa alternatif-alternatif dalam pengoperasian.

1. *Flow Direction Symbols*

Yaitu, simbol yang dipakai untuk menghubungkan antara simbol yang satu dengan simbol lainnya atau disebut juga *connecting line.*

Tabel 1. *Flow Direction Symbols*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| (1) | (2) | (3) | (4) |
| 1 | C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-30162.pngC:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-8064.png  C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-22166.png  C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-23398.png | *Arus / Flow* | Penghubung antara prosedur / proses |
| Tabel 1. *Flow Direction Symbols* (lanjutan) | | | |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| (1) | (2) | (3) | (4) |
| 2 | C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-15213.png | *Connector* | Simbol keluar / masuk prosedur atau proses dalam lembar / halaman yang sama |
| 3 | C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-18917.png | *Off-line Connector* | Simbol keluar / masuk prosedur atau proses dalam lembar / halaman yang lain |

1. *Processing Symbols*

Merupakan simbol yang menunjukkan jenis operasi pengolahan dalam suatu prosedur.

Tabel 2. *Processing Symbols*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| (1) | (2) | (3) | (4) |
| 1 | C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-25859.png | Process | Simbol yang menunjukkan pengolahan yang dilakukan  Komputer |
| 2 | C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-2446.png | *Decision* | Simbol untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban / aksi |
| Tabel 2. *Processing Symbols* (lanjutan) | | | |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| (1) | (2) | (3) | (4) |
| 3 | C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-7661.png | *Predefined Process* | Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan didalam storage |
| 4 | C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-28606.png | Terminal | Simbol untuk permulaan atau akhir darti suatu program |
| 5 |  | *Manual Input* | Simbol untuk pemasukan data secara manual on-line keyboard |

1. *Input Output Symbols*

Simbol yang dipakai untuk menyatakan jenis peralatan yang digunakan sebagai media input atau output. Pada tabel 3 di bawah ini akan dijelaskan simbol-simbol dari *input-output.*

Tabel 3. *Input Output Symbols*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| (1) | (2) | (3) | (4) |
| 1 | C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-28963.png | *Input*-*Output* | Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya |
| 2 | C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-22186.png | *Document* | Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output di cetak dikertas |
| 3 | C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-24398.png | *Disk and On-line Storage* | Simbol untuk menyatakan input berasal dari disk atau output di simpan ke disk |

***Ishikawa* Diagram (*Fishbone* Diagram)**

Diagram *Fishbone* atau *Ishikawa* merupakan sebuah alat grafis yang digunakan untuk mengidentifikasi, mengeksplorasi dan menggambarkan suatu masalah, sebab dan akibat dari masalah itu. Sering disebut diagram sebab-akibat atau diagram tulang ikan (*fishbone* diagram) karena menyerupai tulang ikan.

Berikut ini merupakan contoh gambar dari diagram *Fishbone* yang dapat dilihat pada gambar 2 dibawah ini.

Material

Metode

Manusia

Teknologi

Masalah

Gambar 2. Contoh *Ishikawa* Diagram

***Unified Modeling Language (UML)***

*UML* adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek untuk memvisualisasikan, membuat spesifikasi, dan mendokumentasikan suatu aplikasi (Whitten, Jeffrey dan Bentley, 2007). Berikut ini ialah penjelasan beberapa diagram *UML* yang akan digunakan.

1. *Use case* Diagram

*Use case* adalah abstraksi dari interaksi antara sistem dan aktor. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana system akan terlihat di mata *user*. Sedangkan *Use case* diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan *client*.

Tabel 4. Simbol *Use Case* Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| (1) | (2) | (3) | (4) |
| 2 | C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-10845.png | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*). |
| 3 | C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-24408.png | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 4 | C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-20926.png | *Include* | Menspesifikasikan bahwa *Use case* sumber secara *eksplisit*. |
| 5 | C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-16506.png | *Extend* | Menspesifikasikan bahwa *Use case* target memperluas perilaku dari *Use case* sumber pada suatu titik yang diberikan. |
| 6 | C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-29865.png | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7 | C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-8289.png | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
|  |  |  |  |
| Tabel 4. Simbol *Use Case* Diagram (lanjutan) | | | |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| (1) | (2) | (3) | (4) |
| 8 |  | *Use Case* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 9 |  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| 10 | C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-4630.png | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |

1. *Activity* Diagram

Pada dasarnya diagram *activity* sering digunakan oleh *flowchart*. Diagram ini berhubungan dengan diagram *statechart*. Diagram *statechart* berfokus pada objek dalam suatu proses (atau proses menjadi suatu objek), diagram *activity* berfokus pada aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal. Dengan kata lain, diagram ini menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain. Proses berawal dari lingkaran start hitam pada bagian atas dan berakhir dipusat lingkaran setop hitam/putih pada bagian bawah. Aktivitas digambarkan dalam bentuk kotak persegi.

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| (1) | (2) | (3) | (4) |
| 2 | **C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-1268.png** | *Action* | State dari sistem yang mencerminkan eksekusi dari suatu aksi |
| 3 | **C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-7258.png** | *Initial Node* | Bagaimana objek dibentuk atau diawali. |
| 4 | **C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-2839.png** | *Actifity Final Node* | Bagaimana objek dibentuk dan dihancurkan |
| 5 | **C:\Users\HAIDIR~1\AppData\Local\Temp\ksohtml\wps_clip_image-32580.png** | *Fork Node* | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |

Tabel 5. Simbol *Activity* Diagram

Diagram *activity* dapat dibagi menjadi beberapa jalur kelompok yang menunjukkan objek mana yang bertanggung jawab untuk suatu aktifitas. Peralihan tunggal *(single transition*) timbul dari setiap adanya *activity* (aktifitas), yang saling menghubungi pada aktifitas berikutnya. Sebuah transition (transisi) dapat membuat cabang ke dua atau lebih percabangan exclusivetransition (transisi eksklusif). *Label Guard Expression* (ada di dalam [ ]) yang menerangkan *output* (keluaran) dari percabangan. Percabangan akan menghasilkan bentuk menyerupai bentuk intan. *Transition* bisa bercabang menjadi beberapa aktifitas paralel yang disebut *Fork*. *Fork* beserta join (gabungan dari hasil *output* *fork*) dalam diagram berbentuk *solid* bar (batang penuh).

***Entity Relationship Diagram (ERD)***

Entity relationship diagram (ERD) adalah sebuah data model yang menggunakan beberapa notasi untuk menggambarkan data dengan istilah entitas dan relationships yang ada. Sebuah entitas merupakan sebuah class untuk menyimpan data dan antarsatu atau lebih entitas dihubungkan dengan suatu relationship (Whitten, Jeffrey dan Bentley, 2007).

1. **Penelitian Terkait**

Adapun penelitian terkait tentang penggunaan teknologi SIG yang telah dirangkum penulis adalah sebagai berikut:

1. Yunofri (2009) dalam Sitem Informasi Geografis Berbasis *Web* dengan Atribut Dinamis.

Penelitian ini bertujuan untuk menghasilkan aplikasi yang dapat menampilkan peta tematik dan diagram dari data yang dimasukan. Penelitian ini menggunakan map *server* untuk menyimpan data spasial yang akan dipakai untuk membuat peta tematik. Penelitian ini terfokus pada penyajian data, peta tematik yang dibuat hanya sebatas perbedaan degradasi warna untuk sebaran data dan satu projek hanya menampilkan satu tema (misal Pertumbuhan hasil ternak di Provinsi Jawa Timur).

1. Disna (2012) dalam Sistem Aplikasi Visualisasi Data ke Peta Tematik berbasis *WebGIS*.

Penelitian ini menghasilkan suatu aplikasi yang dapat membuat aplikasi *web*GIS secara otomatis. Aplikasi ini men-*generate* data dari Microsoft excel menjadi suatu peta tematik. Alur pembuatan *webGIS* dengan aplikasi adalah data agregat dari satu atau beberapa variabel di-*import* ke dalam aplikasi lalu di-*generate* langsung menjadi peta tematik, tetapi untuk melihat hasilnya harus diatur dahulu skala dan degradasi warnanya, sehingga peta tematik yang dihasilkan tidak memiliki tampilan default. Aplikasi masih bersifat statis karena data atribut dari setiap projek tidak bias ditambah atau dikurang secara langung dan harus membuat proyek baru untuk mengubah variabel-variabel yang ingin ditampilkan dalam setiap proyek dan peta tematik yang ditampilkan dalam satu level wilayah.

1. Anisa Aizatin (2013) dalam *Enhanced* *GIS*: Pengembangan Visualisasi Data ke Dalam Peta Tematik.

Penelitian ini merupakan pengembangan dari penelitian yang dilakukan oleh Disna (2012). Data atribut yang akan ditampilkan menjadi peta tematik dapat diubah baik ditambah, maupun diubah setiap variabelnya. *webGIS* yang dibuat pun memiliki tampilan awal atau *default.* Masih terdapat banyak kekurangan dalam aplikasi ini yaitu diantaranya adalah:

* Dalam pembuatan peta tematik atau import masih membutuhkan tahapan yang cukup panjang sehingga membutuhkan usaha lebih yaitu saat pemilihan variabel dan wilayah.
* Dalam sekali import data, aplikasi ini hanya mampu mengeksekusi satu variabel terpilih saja.
* Untuk data jenis kategori masih belum terfasilitasi.

Sistem yang akan dibuat oleh penulis adalah hasil pengebangan dari penelitian yang telah dilakukan oleh Anisa Aizatin (2013). Teknologi yang digunakan adalah *Geoserver* dengan bantuan *Leaflet* untuk pendukung peta dasar. Pengembangan yang menjadi pembeda antara penelitian ini dan penelitian sebelumnya adalah:

1. Menggunakan konsep *backend* dan *frontend*, sehingga desain maupun koding terpisah antara *user admin* dan *user* pengunjung.
2. Desain dari sisi user lebih dinamis, sehingga user bisa mengubah tampilan peta sesuai dengan kebutuhannya tanpa mengubah *database*.
3. Meningkatkan kemudahan dari sisi admin yaitu dalam sekali *import* data yang masuk bisa lebih dari satu variabel sesuai dengan kebutuhan.
4. Ditambahakan grafik dan beberapa informasi statistik.
5. **Kerangka Pikir**

Badan Pusat Statistik (BPS) sebagai penyedia data untuk semua terus berusaha untuk meningkatkan kualitas publikasinya. Namun masih ada saja pengguna data atau masyarakat yang kurang paham maksud dari publikasi BPS tersebut. Hal ini bisa saja terjadi karena kurangnya pengetahuan pengguna data atau mungkin data yang disajikan terlalu rumit sehingga butuh *effort* lebih untuk memahaminya.

Berdasarkan hasil dari survei kepuasan konsumen tahun 2014 yang telah dilakukan BPS ditemukan bahwa pengguna data cenderung masih kesulitan dalam memahami data yang dipublikasi BPS serta mengakses datanya pun cukup kesusahan.

Sehubungan dengan semakin meningkatnya ilmu teknologi sekarang ini tentu BPS dapat memanfaatkan kondisi tersebut. Berbagai cara publikasi telah dilakukan, salah satunya adalah dengan menggunakan SIG berbasis *web*. Namun SIG yang ada sekarang ini masih statis dan dibuatnya pun cenderung menunggu publikasi-publikasi besar seperti contoh sensus penduduk. Oleh karena itu dibutuhkan sebuah SIG berbasi *web* yang terpusat dan lebih dinamis variabel dan wilayahnya sehingga pengguna data dapat dengan mudah memilih sesuai dengan kebutuhan.

Berdasarakan masalah tersebut diatas maka peneliti mengusulkan untuk mengembangkan SIG berbasis *web* yang telah ada. Pengembangan yang diusulkan peneliti adalah penambahan grafik, variabel yang lebih dinamis dan dapat mengakases SIG dengan berbagai level peta. Alur kerangka pikir dapat di liat pada gambar 3 di bawah ini.

**Solusi**

Mengembangkan SIG berbasis WEB

* Degradasi warna
* Grafik
* Variabel yang dinamis

**Pembangunan Sistem**

1. Perancangan
2. Implementasi

**Evaluasi**

● Kritik ● Saran

**Masalah**

1. Kebutuhan informasi yang mudah dipahami
2. Kebutuhan akses data yang mudah
3. SIG BPS yang masih statis

**Pengujian**

1. *Black Box*
2. *White Box*
3. *SUS*

Gambar 3. Alur Kerangka Pikir

Dari solusi permasalahan diatas kemudian selanjutnya peneliti melakukan pembangunan sistem dengan tahapan perancangan kemudian dilakukan implementasi. Setelah diimplementasi lalu aplikasi tersebut dilakukan pengujian dengan metode uji *black box, white box,* dan *SUS (System Usability Scale).* Terakhir adalah tahap evaluasi yaitu kritik dan saran untuk perbaikan kedepannya.