

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Penelitian ini akan membahas tentang bagaimana *user-centered design* (UCD) membantu dalam mengavaluasi sebuah antarmuka dengan melakukan *usability testing* kepada responden dengan kriteria tertentu, penelitian yang berkaitan dengan UCD dan *usability testing* sebelumnya sudah banyak dilakukan oleh peneliti lain. UCD sebagai istilah dan metode untuk yang berfokus pada desain untuk melibatkan pengguna dalam desain aplikasi. Pengguna yang terlibat menjadi partisipan dalam UCD memungkinkan untuk menyampaikan kebutuhan mereka, melakukan observasi dan mengikuti *usability testing* dan partisipan dapat menjadi rekan dalam proses desain. Banyak variasi metode yang telah dikembangkan untuk mendukung proses UCD termasuk *usability testing*, *usability engineering*, *heuristic evaluation*, *discount evaluation* dan *participatory design*. *Quick and dirty evaluation* juga menjadi penting untuk mendapatkan ide dan representatif pengguna tentang masukan pada desain awal (Abrams et al, 2004).

Perbandingan *adaptive map* dan *non-adaptive map* dan menghasilkan bahwa *adaptive map* memberikan pengguna keyakinan tinggi dalam mengoperasikan aplikasi dan pengguna tidak suka terinterupsi pada proses spasial kognitif dengan mendapatkan adaptasi konten pada *map* secara tiba-tiba (Kiefer, 2017). Penelitian ini menggunakan tahap metodologi yang hampir serupa yaitu dengan mengidentifikasi obyek yang akan diteliti dalam hal ini adalah aplikasi ArcGIS dan QGIS, lalu mempersiapkan desain skenario dan tugas *usability testing* hingga perangkat lunak dan perangkat keras yang mendukung *usability testing*.

2.2 User-Centered Design

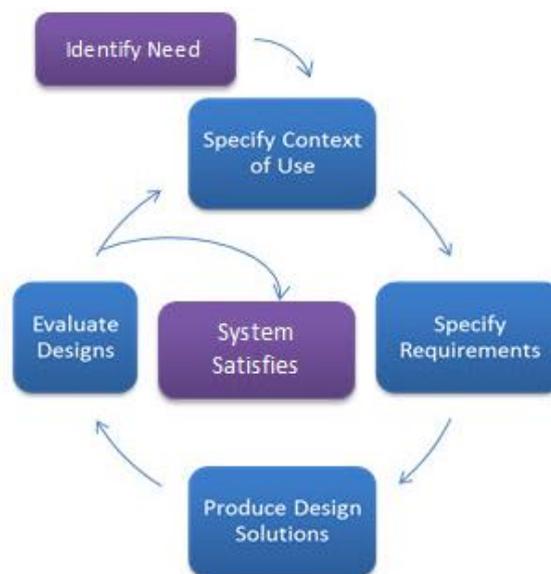
Desain pada benda-benda yang setiap hari ditemui tidak selalu intuitif dan sewaktu-waktu membuat pengguna frustrasi dan tidak dapat mengerjakan tugas yang sederhana. *User-centered design* (UCD) adalah metode untuk mendeskripsikan proses desain kepada *end-users* untuk memengaruhi bagaimana sebuah desain itu bekerja. Pengguna menjadi bagian penting dalam UCD. Contohnya, sebuah UCD berhubungan dengan kebutuhan pengguna dan melibatkan pengguna dalam proses desain, biasanya pada tahap pengumpulan kebutuhan dan *usability testing*. Metode UCD akan memberikan pengaruh yang sangat berdampak pada pengguna serta desain perangkat lunak dan akan sangat membantu desainer dalam tahap desain proses.

Istilah *user-centered design* pertama kali disebutkan di laboratory penelitian University of California San Diego (UCSD) pada tahun 1980 dan meluas setelah penerbitan buku yang berjudul "*User-centered System Design: New Perspectives on Human-Computer Interaction*" (Norman, 1986). Dia mengenali kebutuhan dan ketertarikan pengguna dan fokus pada *Usability* dari sebuah desain. Dia memberikan empat sugesti bagaimana desain itu seharusnya :

- Buatlah semudah mungkin untuk menentukan aksi apa yang memungkinkan untuk dilakukan pada momen tertentu
- Buatlah sesuatu yang terlihat, termasuk model konseptual sebuah sistem, aksi alternative, dan hasil dari aksi tersebut
- Buatlah semudah mungkin untuk mengevaluasi keadaan saat ini dari sebuah sistem
- Ikuti pemetaan natural diantara tujuan dan kebutuhan; antara aksi dan efek yang dihasilkan; antara informasi yang terlihat dan intepretasi keadaan sistem (Norman, 1988).

Rekomendasi ini menempatkan pengguna pada tengah-tengah sebuah desain. Peran dari desainer adalah untuk memfasilitasi tugas untuk pengguna dan memastikan pengguna dapat menggunakan produk yang dimaksud dengan usaha yang minimum untuk mempelajari bagaimana menggunakannya. Norman mencatat bahwa buku manual yang sangat panjang tidaklah *user-centered*. Dia menyarankan bahwa sebuah produk harus disertai pamflet kecil yang dapat dibaca dengan sangat cepat dan berdasarkan pengetahuan pengguna didunia.

2.2.1 Fase *User-Centered Design*



Gambar 2.1 Fase *User-Centered Design* (usability.gov, 2015)

UCD memiliki beberapa tahapan dalam sebuah penelitian pada Gambar 2.1, yaitu:

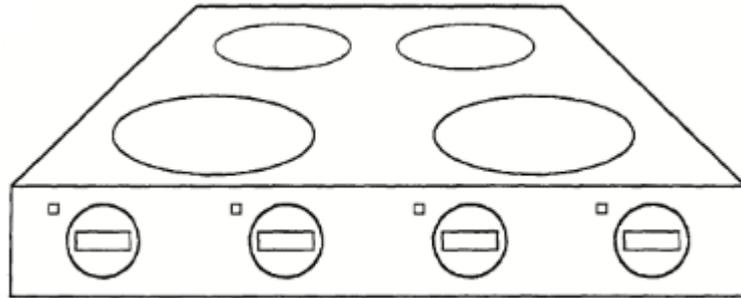
- *Specify the context of use*, mengidentifikasi orang-orang yang akan menggunakan produk, produk tersebut akan digunakan untuk apa, dan dalam kondisi apa mereka akan menggunakan produk tersebut. Hasil dari tahapan ini adalah kriteria dari pengguna yang akan diuji, dan jumlah pengguna yang akan menjadi responden pada penelitian

- *Specify requirements*, mengidentifikasi kebutuhan bisnis atau tujuan pengguna yang harus dipenuhi agar produk berhasil dalam memenuhi kebutuhan pengguna. Hasil dari tahapan ini adalah beberapa tugas yang akan diberikan yang harus diselesaikan oleh responden
- *Create design solutions*, membuat sebuah konsep kasar hingga desain yang lengkap
- *Evaluate designs*, melewati pengujian *usability* dengan beberapa pengguna. Hasil dari tahapan ini adalah nilai perbandingan dari aplikasi yang diteliti.

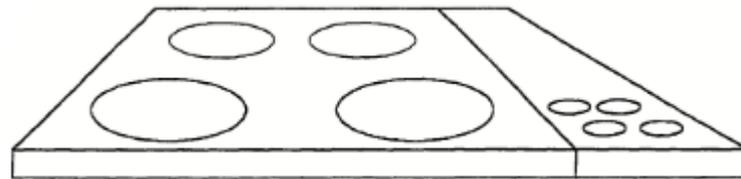
2.3 User Models

Pengguna memiliki model tentang bagaimana mereka menyelesaikan tugasnya dengan sebuah sistem, dan bagaimana sebuah sistem tepat membantu menyelesaikan tugas pengguna. Model ini ditentukan oleh prioritas pengalaman pengguna, pengetahuan pengguna, dan pendapat tentang tugas tersebut. Untuk menyelesaikan tugas tersebut pengguna harus sudah memiliki pengetahuan untuknya. Untuk sebuah bahan yang baru dapat dipahami, harus dapat diinterpretasikan dan diintegrasikan dengan pengetahuan yang sudah dimiliki (Norman, 1988). Apabila bahan baru dikenalkan dengan buruk, pengguna akan mencoba mengintegrasikan hal tersebut dengan pengetahuan yang dimiliki dan membuat penjelasan mereka sendiri. Contohnya, seorang siswa ditemukan menyalahkan dirinya sendiri ketika sebuah paket SIG yang dia kerjakan mengalami kerusakan karena bugs dari sebuah perangkat lunak. Dalam kata lain, siswa tersebut menyalahkan SIG dalam permasalahan yang dibuatnya sendiri. Kebiasaan ini terjadi pada layaknya pengguna komputer. Mereka cenderung selalu memikirkan tentang fungsi dari sebuah sistem yang mereka tidak mengerti. Salah satu tugas dari desainer antarmuka pengguna adalah membuat pengguna dapat menyelesaikan tugasnya sesuai dengan tujuannya masing-masing (Lanter, 1991).

Setiap pengguna memiliki konseptual model dari sebuah perangkat lunak yang berinteraksi dengannya. Model ini dibentuk dan ditularkan karena faktor internal atau eksternal. Faktor internal berhubungan dengan tujuan pengguna, ekspektasi, niat, pengalaman, praanggapan, pengetahuan yang lalu dan penjelasan. Faktor eksternal berhubungan dengan antarmuka sistem sebagai representasi sistem itu sendiri kepada pengguna untuk mengungkap dimensi lain dibalik sebuah antarmuka. Antarmuka sistem didesain untuk mencocokkan konseptual model pengguna yang memerlukan kebutuhan dokumentasi eksternal. Sistem yang tanpa memperhatikan pengguna biasanya memerlukan sebuah dokumentasi yang cukup banyak dibanding sebuah sistem yang memperhatikan penggunanya. Pelatihanpun saat ini berguna untuk mengubah model konseptual pengguna untuk mencocokkan sebuah sistem. *Support* berfungsi untuk membantu para pengguna untuk beradaptasi pada sebuah aspek sistem yang cukup sulit. Buku manual yang tebal, waktu pelatihan yang panjang, dan jumlah telpon pada *support* yang banyak menandakan kegagalan dalam sebuah antarmuka sistem dalam model konseptual pengguna.



Gambar 2.2 Desain Antarmuka Kompor 1 (Lanter, 1991)



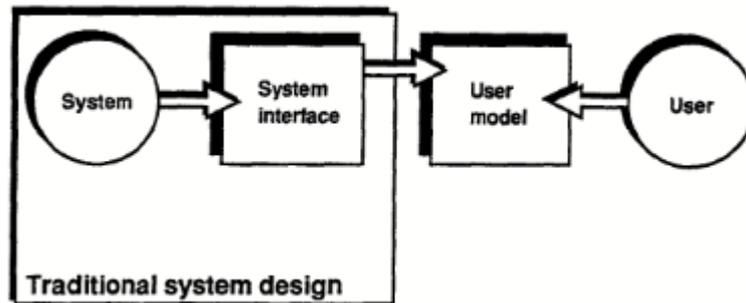
Gambar 2.3 Desain Antarmuka Kompor 2 (Lanter, 1991)

Pada Gambar 2.2 terdapat sebuah desain antarmuka kompor yang cukup tebal dengan tombol untuk menyalakan kompor berada dibagian depan kompor, dengan empat tombol yang sejajar, dan empat lubang api yang berada dibagian atas kompor, pada gambar ini memiliki desain antarmuka yang buruk, dikarenakan pengguna akan kebingungan ketika memilih untuk menyalakan api pada salah satu lubang api di kompor, pengguna akan terus mencoba ketika api yang dinyalakan tidak sesuai dengan keinginan pengguna, dalam kasus ini penempatan urutan tombol api dan lubang api sangat memengaruhi kemudahan dan kenyamanan pengguna dalam menggunakan kompor, berbeda dengan Gambar 2.3 yaitu desain antarmuka kompor dengan empat lubang api disisi kiri kompor dan empat tombol api di sebelah kanan kompor, desain antarmuka ini memudahkan pengguna dalam menggunakan kompor karena dapat dilihat dari tata letak lubang api dan tombol api yang memiliki bentuk posisi yang serupa. Perumpamaan yang ditunjukkan oleh Gambar 2.2 dan Gambar 2.3 dalam penelitian ini adalah tata letak menu pada sebuah antarmuka haruslah mudah dipahami dengan penempatan yang sesuai berdasarkan kegunaannya.

2.4 User-centered System Design

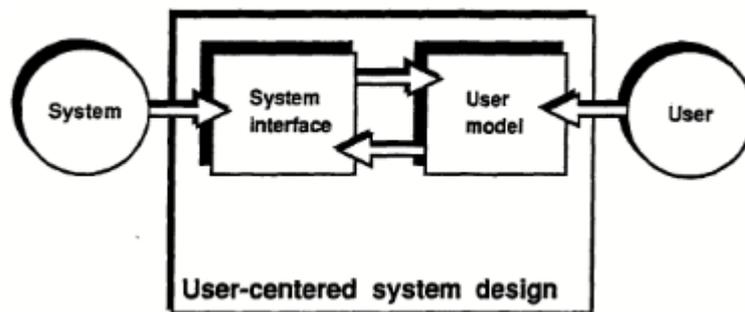
Dalam komputasi, pemetaan yang tidak wajar akan menghambat kegunaan yang buruk, hal ini menjadi tantangan yang besar untuk sebuah sistem informasi yang kompleks. Diagram konsep sebuah antarmuka sistem yang berhubungan dengan model pengguna ada pada Gambar 2.4. Ada dua cara pemetaan diantara antarmuka sistem dengan model pengguna. Sebuah desain akan menciptakan dampak yang kuat dalam pembelajaran terhadap sistem. Apabila pengguna menemukan ekspektasinya dan asumsinya bagaimana sistem bekerja maka pengguna hanya akan memerlukan pembelajaran terhadap sistem yang sedikit dalam mempelajari konsep yang baru dan cara untuk melakukan sesuatu. Apabila sistem tidak cocok dengan model pengguna yang ada, maka pembelajaran sangat

diperlukan bagi pengguna. Pengeluaran ingatan akan meningkat bagi pengguna agar harus mengingat sesuatu yang baru dan obyek yang tidak akrab dalam ingatan mereka atau keseharian mereka.



Gambar 2.4 Desain Sistem Tradisional (Lanter, 1991)

Desain antarmuka pengguna yang tradisional terdapat pada Gambar 2.4, dimana antarmuka pengguna didesain terfokus untuk menyelesaikan permasalahan bagaimana cara merepresentasikan yang terbaik sebuah fungsi pada sistem di antarmuka sistem.

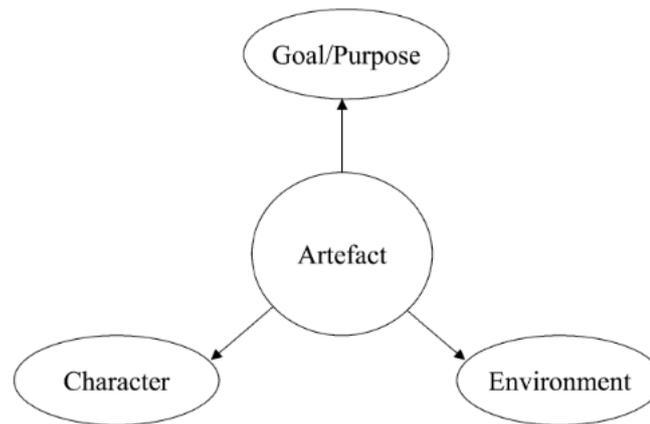


Gambar 2.5 User-centered System Design (Lanter, 1991)

Pada Gambar 2.5 terfokus pada *user-centered design* bagaimana memetakan sebuah antarmuka sistem dengan model pengguna yang ada dan bagaimana pengguna membentuk model pengguna itu sendiri. Bukan berarti antarmuka pengguna harus menjadi budak dari model pengguna, namun desainer harus bisa mengarahkan model pengguna ke sesuatu yang baru yang lebih informatif dan efisien untuk melakukan sesuatu hal yang dapat dikenalkan dan dipahami.

2.5 Artefak Desain

Artefak sebuah desain tidak selalu berupa *physical object* atau sesuatu yang bisa digunakan untuk melakukan fungsi tertentu seperti prototipe, namun artefak memiliki sifat lain yaitu berupa *intentional object*. Perbedaannya adalah *intentional object* memiliki makna hanya pada konteks yang dimaksudkan seperti apa (Kroes, 2009).



Gambar 2.6 Skema Representasi Artefak (Kroes, 2009)

Skema pada Gambar 2.6 menunjukkan bahwa sesuatu dapat disebut artefak ketika memiliki sebuah *goals/purpose*, *character*, dan *environment*. Sebuah skenario yang memiliki tujuan untuk mendukung dalam mendapatkan hasil dari *usability testing*, memiliki karakter dengan tugas dan fungsi yang dikerjakan didalamnya, dan lingkungan yang sudah ditentukan dapat disebut sebuah artefak.

2.6 Usability

Usability adalah sebuah atribut yang menilai seberapa mudah antarmuka pengguna dapat digunakan atau *usability* adalah sebuah pengukuran kualitas dimana pengguna mendapatkan pengalaman saat berinteraksi dengan sistem yang ada (Nielsen, 1993). *Usability* juga mengacu pada metode untuk meningkatkan kemudahan penggunaan dalam sebuah proses perancangan.

Usability dibagi menjadi tiga komponen, yaitu :

1. *Effectiveness*, dapatkah pengguna menyelesaikan tugasnya, dan mendapatkan tujuannya dengan penggunaan sebuah produk
2. *Efficiency*, seberapa besar usaha yang dilakukan pengguna untuk melakukan ini dalam satuan waktu
3. *Satisfaction*, apa yang dipikirkan oleh pengguna tentang kemudahan penggunaan sebuah produk (*International Organization for Standardization*, 1998).

Usability benar-benar murni fokus pada sebuah fungsi pada sebuah produk apakah dapat digunakan oleh pengguna dengan optimal atau tidak. Oleh karena itu pada penelitian ini aspek *usability* yang akan diuji dan dinilai adalah aspek *effectiveness*, *efficiency*, dan *satisfaction*.

2.6.1 Usability Metrics untuk Effectiveness

Effectiveness dapat dihitung dengan mengukur tingkat penyelesaian. Dengan menghitung dan memberikan nilai 1 jika responden dapat menyelesaikan tugas

yang diberikan, dan 0 jika responden tidak dapat menyelesaikan tugasnya (Mifsud, 2015).

Effectiveness dapat direpresentasikan sebagai presentasi menggunakan persamaan (2.1).

$$Effectiveness = \frac{\text{Jumlah tugas yang berhasil diselesaikan}}{\text{Total jumlah tugas yang dikerjakan}} \times 100\% \quad (2.1)$$

2.6.2 Usability Metrics Untuk Efficiency

Efficiency adalah pengukuran tentang waktu pengerjaan tugas yang dilakukan oleh responden (dalam detik atau menit) (Mifsud, 2015). Waktu yang dihabiskan untuk menyelesaikan tugas dapat dikalkulasikan dengan mengurangi waktu memulai pengerjaan dengan waktu berakhirnya pengerjaan seperti pada persamaan (2.2).

$$\text{Waktu Tugas} = \text{Waktu selesai} - \text{Waktu mulai} \quad (2.2)$$

Sedangkan untuk menghitung aspek *efficiency* dapat dilakukan kalkulasi seperti pada persamaan (2.3).

$$Time Based Efficiency = \left(\frac{N_{ij1} + N_{ij2} + N_{ijn}}{T_{ij1} + T_{ij2} + T_{ijn}} \right) = \text{goals/min} \quad (2.3)$$

Pada rumus diatas, dapat diartikan sebagai:

- n = Urutan responden hingga sejumlah N
- Nij = Hasil tugas responden (1 = berhasil, 0 = gagal)
- Tij = Waktu yang diperlukan responden dalam mengerjakan tugas hingga selesai / waktu responden hingga menyerah dalam mengerjakan tugas
- N = Total tugas yang diberikan
- R = Jumlah responden

2.7 Sistem Informasi Geografis

Sistem Informasi Geografis (SIG) adalah sistem informasi yang berdasar pada data keruangan dan merepresentasikan obyek di bumi. Dalam SIG sendiri teknologi informasi merupakan perangkat yang membantu dalam menyimpan

datas, memproses data, menganalisa data, mengelola data dan menyajikan informasi. SIG merupakan sistem yang terkomputerisasi yang menolong dalam memperbaiki 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 didalamnya 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 atribut yang memberikan informasi tentang bentuk keruangannya (Chang, 2002). 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 (Burrough, 1998). Pernyataan lainnya 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 aktivitas SIG tersebut dilakukan. Implementasi dari pelaksanaan kegiatan tersebut tidak selalu mengacu pada penyertaan komputer sebagai salah satu elemen pada sistem informasi (Aronoff, 1989).

2.8 Data Spasial

Data spasial adalah data yang bereferensi geografis atas representasi obyek di bumi. Data spasial pada umumnya berdasarkan peta yang berisikan interpretasi 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 obyek di muka bumi.

Sesuai dengan perkembangan, peta tidak hanya merepresentasikan obyek-obyek yang ada di muka bumi, tetapi berkembang menjadi representasi obyek diatas muka bumi (diudara) 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.

2.9 Data Vektor

Model data vektor adalah yang dapat menampilkan, menempatkan, dan menyimpan data spasial dengan menggunakan titik-titik, garis atau kurva dan polygon beserta atribut-atributnya (Prahasta, 2002). 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, 2002). 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 titik. Pada penelitian ini data vektor menjadi data yang digunakan untuk melakukan *geoprocessing* yang menjadi bagian yang sangat penting bagi perusahaan yang menggunakan SIG. Tidak terhitung jumlah tugas *geoprocessing* yang dilakukan dalam sehari. Fungsi *geoprocessing* yang pada umumnya sering dipakai adalah *clip*, *intersect*, *buffer*, *dissolve*, *merge*, dan *union* (ESRI, 2006).

2.9.1 Kelebihan Data Vektor

Data vektor memiliki beberapa kelebihan, yaitu :

- Data dapat direpresentasikan seperti resolusi aslinya dan tanpa digeneralisasikan
- Keluaran grafis memiliki estetika lebih seperti representasi katografi tradisional
- Kebanyakan data seperti peta dalam bentuk media cetak, dalam bentuk vektor tidak memerlukan konversi data
- Data lokasi geografis lebih baik
- Memungkinkan melakukan *encoding* topologi dengan mudah, dan hasilnya lebih efisien karena memerlukan informasi topologi seperti *proximity* dan analisis jaringan.

2.9.2 Kekurangan Data Vektor

Data vektor juga memiliki beberapa kekurangan, yaitu :

- Lokasi setiap vertex membutuhkan penyimpanan secara eksplisit
- Untuk analisis yang efektif, data vektor harus dikonversi terlebih dahulu ke struktur topologi. Proses intensif yang sering terjadi dan biasanya memerlukan pembersihan data yang luas. Karena topologi adalah status dan setiap perbaruan dan mengubah data vektor memerlukan pembangunan ulang dari topologi itu sendiri
- Algoritma untuk fungsi memanipulasi dan analisis sangat kompleks dan memungkinkan proses yang intensif. Sering kali terjadi keterbatasan fungsional untuk set data yang besar, contohnya adalah besarnya jumlah fitur
- Data yang berlanjut, seperti data elevasi, yang tidak efektif dalam merepresentasikan bentuk vektor. Biasanya generalisasi data substansial atau interpolasi diperlukan untuk lapisan data ini
- Analisis spasial dan *filtering* menggunakan poligon tidak memungkinkan.

2.10 QGIS

QGIS merupakan perangkat lunak pengolah Sistem Informasi Geografis (SIG) yang bersifat *Open Source* yang *user friendly*. QGIS dapat dijalankan diberbagai macam sistem operasi, contohnya pada Linux, Unix, Mac, OSX, Windows dan Android. QGIS memiliki banyak format dan fungsionalitas pada vektor, raster dan basisdata. Terutama fungsi yang akan digunakan pada penelitian ini adalah fungsi yang terdapat pada *geoprocessing tools* untuk mengolah data vektor seperti *clip*, *intersection*, *fixed distance buffer* atau *buffer*, *dissolve*, *merge*, dan *union*.

2.11 ArcGIS

ArcGIS *desktop* merupakan perangkat lunak pengolah SIG seperti QGIS, yang paling membedakan adalah sifatnya yang berbayar, tidak seperti QGIS yang bersifat *open source* dan gratis dan kedua aplikasi ini pun memiliki beberapa fungsi yang sama terutama pada *geoprocessing tools*.

2.12 Geoprocessing Tools

Geoprocessing adalah komponen yang paling kuat dalam SIG, *geoprocessing* dapat menentukan, mengelola, dan menganalisa informasi untuk dijadikan sebuah keputusan. *Geoprocessing* menjadi bagian yang sangat penting bagi perusahaan yang menggunakan SIG. Tidak terhitung jumlah tugas *geoprocessing* yang dilakukan dalam sehari. Fungsi *geoprocessing* yang pada umumnya sering dipakai adalah *clip*, *intersect*, *buffer*, *dissolve*, *merge*, dan *union* (ESRI, 2006).

Berikut ini adalah penjelasan tentang fungsi pada *geoprocessing* yaitu :

1. *Clip* adalah sebuah fungsi yang memiliki algoritma untuk memotong *layer* vektor menggunakan *polygon*. Hanya bagian dari fitur pada *input layer* yang berada didalam lapisan *polygon* yang akan dihasilkan pada *output layer*
2. *Intersection* adalah fungsi yang mengembalikan nilai geometri yang merepresentasikan bentuk dari dua potongan geometris. Algoritma pada fungsi *intersection* melakukan ekstraksi pada lapisan yang saling menumpuk pada *input layer* dan *intersect layer*. Obyek yang ada pada *intersection layer* akan menjalankan perintah pada atribut dari keduanya yaitu *input layer* dan *intersect layer*
3. *Buffer* adalah fungsi yang memiliki algoritma untuk menghitung *buffer area* untuk semua fitur pada *input layer*. Ukuran dari *buffer* untuk fitur tertentu ditentukan dengan sebuah atribut, sehingga setiap fitur dapat memiliki ukuran *buffer* yang berbeda-beda
4. *Dissolve* adalah fungsi yang memiliki algoritma untuk mengambil *polygon vector layer* ke geometri tunggal. Sebuah atribut dapat ditentukan untuk dilakukan *dissolve* pada *polygon* yang memiliki kelas yang sama (memiliki nilai yang sama untuk atribut yang ditentukan),

atau semua *polygon* dapat dilakukan *dissolve* dengan mempertimbangkan geometrisnya

5. *Union* adalah fungsi yang mengembalikan nilai geometri yang merepresentasikan satu kumpulan persatuan beberapa titik dari sebuah geometris. Algoritma *union* adalah membuat sebuah *layer* yang memuat semua fitur pada kedua *input layer*. Contohnya pada *polygon layer*, fitur yang terpisah tercipta untuk *layer* yang menumpuk dan *layer* yang tidak menumpuk. Atribut tabel dari *union layer* mengandung nilai atribut dari *input layer* untuk fitur yang tidak saling menumpuk, dan nilai atribut dari kedua *input layer* untuk fitur yang saling menumpuk
6. *Merge* adalah melakukan penggabungan data dengan tipe data yang sama menjadi kesatuan data yang baru. Fungsi ini dapat menggabungkan titik, garis, poligon atau tabel.

2.13 System Usability Scale (SUS)

SUS adalah sebuah kuesioner dengan skala Likert untuk mengukur tingkat *usability* dalam aspek *satisfaction*, SUS memiliki pernyataan yang jumlahnya konsisten yaitu 10 buah dengan setiap pernyataan memiliki 5 respon yang harus dijawab oleh responden dengan hati-hati setelah menggunakan aplikasi, dengan rentang sangat setuju hingga sangat tidak setuju. Jika responden merasa tidak dapat memilih respon pada kuesioner maka responden harus mengisi respon netral atau respon tengah dari rentang skala (Brooke, 1996). Pernyataan pada setiap nomor ganjil (1, 3, 5, 7, 9) merupakan pernyataan yang positif terhadap aplikasi, sedangkan pernyataan pada nomor genap (2, 4, 6, 8, 10) merupakan pernyataan negative terhadap aplikasi.

Penilaian kuesioner SUS memiliki caranya tersendiri yaitu (1) setiap pernyataan pada nomor ganjil bobot pada respon (r) yang dipilih dikurangi 1, sehingga dapat ditulis $(r - 1)$; (2) setiap pernyataan pada nomor genap kurangi 5 dengan bobot pada respon (r) yang dipilih, sehingga dapat ditulis $(5 - r)$; (3) nilai yang didapatkan dari setiap nomor ganjil dan genap ditambahkan secara keseluruhan lalu dikali 2,5 sebagai nilai total setiap responden; (4) seluruh nilai total responden dijumlahkan lalu dibagi sesuai dengan jumlah responden yang mengisi kuesioner; (5) hasil dari kuesioner SUS memiliki rentang nilai dari 0 hingga 100; (6) Nilai SUS dianggap diatas rata-rata jika berada diatas 68, dan dibawah rata-rata jika memiliki nilai dibawah 68.

SUS sudah menjadi standar industri, dengan referensi lebih dari 1300 artikel dan publikasi. Keuntungan menggunakan SUS adalah (1) skala yang sangat mudah untuk diberikan kepada peserta; (2) dapat digunakan untuk sampel yang kecil dengan hasil yang dapat diandalkan; (3) sangat efektif untuk membedakan sebuah sistem dapat digunakan atau tidak dapat digunakan (usability.gov, 2013).

Skala Likert dikembangkan untuk mengukur sikap atau pendapat responden pada suatu sistem, skala Likert mengukur tingkat kesepakatan atau ketidaksepakatan pada suatu pernyataan. Skala Likert memiliki 5 respon yang

berfungsi untuk mengekspresikan seberapa setuju atau tidak setuju responden tersebut dan mengasumsikan bahwa sebuah sikap dapat diukur (McLeod, 2008).