

BAB II

TINJAUAN PUSTAKA

II.1. Penyebab Kemacetan

Selama ini diketahui ada 3 elemen dasar dari moda jalan raya, yaitu: manusia, kendaraan dan lingkungan. Untuk perkotaan sendiri sistem sirkulasi kota sebagai perangkat fisik kota terdiri atas beberapa aspek yang mencakup pola, struktur, dan perlengkapan jalan, aspek lalu lintas dan tempat parkir. Secara rinci aspek-aspek tersebut, sebagai berikut:

1. pola jaringan jalan, seperti : grid, radial, lingkaran, cul-de-sac, dan gabungan di antaranya.
2. Struktur jalan, terdiri atas: badan jalan, bahu jalan (sirkulasi pejalan kaki, tempat perlengkapan jalan, utilitas dan penghijauan).
3. Perlengkapan jalan, penerangan jalan, rambu lalu-lintas, pemberhentian bus, kotak reklame dan lain sebagainya.
4. Aspek lalu-lintas, terdiri dari arah lalu-lintas, kecepatan dan kepadatan serta moda dan alat angkutan.
5. Tempat parkir; merupakan ruang transisi dari sistem sirkulasi dengan bangunan atau kawasan tertentu.

Keberadaan tempat parkir sangat menentukan, karena ini menyangkut permasalahan kelancaran sistem sirkulasi dan penyediaan ruang untuk kendaraan. Kemacetan dapat muncul suatu waktu apabila terdapat pelanggaran pada salah

satu aspek yang telah disebutkan di atas yang dapat mengakibatkan masalah masalah lain sebagai efek samping dari kemacetan itu sendiri. Di samping itu ada beberapa faktor lain yang dapat menyebabkan kemacetan itu terjadi. (*Andy A.M. Malik ; 2011: 20*)

II.2. Efek Dari Kemacetan

Di kota besar kemacetan tak terhindarkan setiap harinya, walaupun jalan tol telah dibangun sebagai alternatif lain untuk menghindari kemacetan namun pada kenyataannya kemacetan tetap saja terjadi. Kemacetan tak memberi efek positif sedikitpun bagi masyarakat, malahan beberapa efek negatif yang merugikan masyarakat sangat rentan terjadi. Kemacetan dapat membawa dampak buruk terhadap kualitas udara sekitar yang dikotori oleh asap kendaraan bermotor, terlebih lagi untuk kendaraan-kendaraan tua yang belum diuji emisi namun masih berkeliaran di kota yang memberi suplai besar akan racun karbon monoksida (CO) di udara sehingga mungkin saja di waktu-waktu kedepan karbon monoksida (CO) yang tak dapat dinetralisir oleh udara dapat naik dan menutupi langit sehingga disinyalir dapat terjadi pemanasan global (Global Warming) yang dapat merugikan masyarakat.

Kebisingan menjadi efek lain yang muncul pada saat kemacetan terjadi dimana suara mesin dan knalpot mobil yang berlebihan dapat merusak gendang telinga kita semua, belum lagi ditambah oleh bunyi sound system yang marak dalam selang beberapa tahun terakhir ini. Desibel yang muncul akibat suara/bunyi tersebut dapat mempengaruhi jiwa dan perasaan kita masing-masing, dan tak

jarang beberapa orang bisa menderita stress saat terjadi kemacetan. Salah satu efek lain yang dapat muncul sewaktu-waktu adalah kriminalitas yang berupa tindak kejahatan pemalakkan, pencurian, dan pengrusakkan kendaraan bermotor oleh oknum-oknum yang tidak bertanggung jawab, walaupun belum separah kota-kota besar namun perlu diwaspadai lebih dini akan tindak kriminalitas tersebut. Tak disangka, tapi itulah kenyataan yang muncul akibat kemacetan walaupun mungkin di suatu hari akan ada akibat lain yang akan muncul tapi uraian di atas telah cukup jelas dalam memaparkan akan efek-efek kemacetan pada lingkup suatu kota. (*Andy A.M. Malik ; 2011 : 24*)

II.3. Sistem Informasi

Sistem Informasi (SI) adalah kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi itu untuk mendukung operasi dan manajemen. Dalam arti yang sangat luas, istilah sistem informasi yang sering digunakan merujuk kepada interaksi antara orang, proses algoritmik, data, dan teknologi. Dalam pengertian ini, istilah ini digunakan untuk merujuk tidak hanya pada penggunaan organisasi Teknologi Informasi dan Komunikasi (TIK), tetapi juga untuk cara di mana orang berinteraksi dengan teknologi ini dalam mendukung proses bisnis.

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (Building Block), dimana masing-masing blok ini saling berintegrasi satu sama lainnya membentuk satu kesatuan untuk mencapai tujuannya. (*Anisya ; 2013 : 50*)

Adapun blok-blok tersebut adalah sebagai berikut :

1. Blok masukan (*Input Blok*)

Meliputi metode-metode dan media untuk menangkap data yang akan dimasukan, dapat berupa dokumen-dokumen dasar.

2. Blok Model (*Model Block*)

Terdiri dari kombinasi prosedur, logika dan model matematik yang berfungsi memanipulasi data untuk keluaran tertentu.

3. Blok Keluaran (*Output Block*)

Berupa keluaran dokumen dan informasi yang berkualitas

4. Blok Teknologi

Untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran serta membantu pengendalian dari sistem secara keseluruhan.

5. Blok Basis Data (*Database Block*)

Merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya, tersimpan didalam perangkat keras komputer dan perangkat lunak untuk memanipulasi.

6. Blok Kendali (*Controls Block*)

Meliputi masalah pengendalian yang berfungsi mencegah dan menangani kesalahan atau kegagalan sistem.

II.4. Sistem Informasi Geografis

Geographical information system (GIS) merupakan komputer yang berbasis pada sistem informasi yang digunakan untuk memberikan bentuk digital dan analisa terhadap permukaan geografi bumi. Definisi GIS selalu berubah karena GIS merupakan bidang kajian ilmu dan teknologi yang relatif masih baru. (Solehuddin Sitorus, 2013, 2)

Beberapa definisi dari GIS adalah:

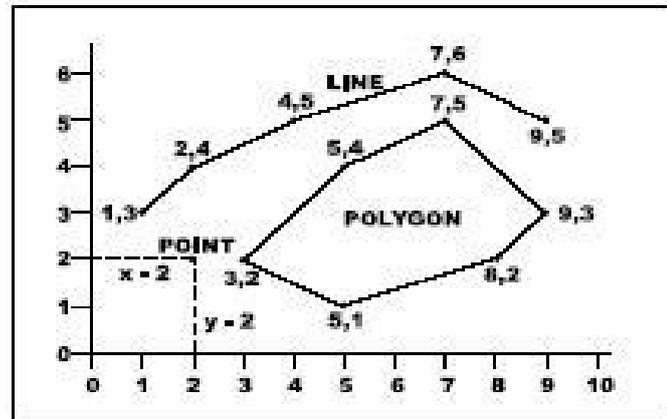
1. Definisi GIS (Rhind, 1988): *GIS is a computer system for collecting, checking, integrating and analyzing information related to the surface of the earth.*
2. Definisi GIS yang dianggap lebih memadai (Marble & Peuquet, 1983) and (Parker, 1988; Ozemoy et al., 1981; Burrough, 1986) *GIS deals with space-time data and often but not necessarily, employs computer hardware and software.*
3. Purwadhi, 1994: SIG merupakan suatu sistem yang mengorganisir perangkat keras (*hardware*), perangkat lunak (*software*), dan data, serta dapat mendaya-gunakan sistem penyimpanan, pengolahan, maupun analisis data secara simultan, sehingga dapat diperoleh informasi yang berkaitan dengan aspek keruangan. SIG merupakan manajemen data spasial dan nonspasialn yang berbasis komputer dengan tiga karakteristik dasar, yaitu:
 - a. mempunyai fenomena aktual (*variabel data non-lokasi*) yang berhubungan dengan topik permasalahan di lokasi bersangkutan.

- b. merupakan suatu kejadian di suatu lokasi.
- c. mempunyai dimensi waktu.

II.5. Data Spasial

Data spasial yaitu yang menyimpan kenampakan- kenampakan permukaan bumi, seperti jalan,sungai,pemukiman jenis penggunaan tanah, jenis tanah dan lain-lain.Model data spasial dibedakan menjadi dua yaitu , model data raster dan model data vektor. (*Solehuddin Sitorus, 2013, 2*)

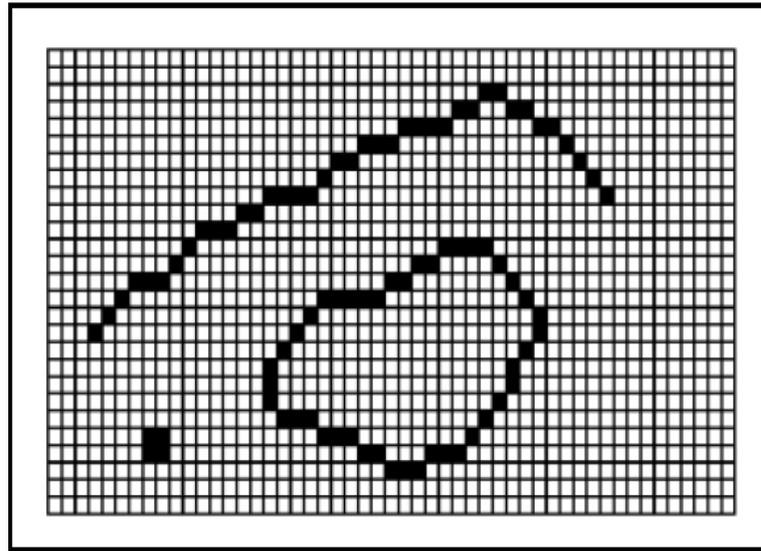
1. Model data vektor diwakili oleh simbol – symbol yaitu titik,garis,area dan permukaan dan dapat dilihat pada Gambar.
 - a. Data titik (node/point) merupakan sepasang koordinat (X,Y) tanpa dimensi
 - b. Data Garis (Arc/Line) merupakan pasangan– pasanagn koordinat dimana titik awal dan titik akhir (X ,Y1:X2,Y2).
 - c. Data luasan/area(polygon) merupakan kumpulan pasangan–pasangan koordinat dimana titik awal sama dengan titik akhir (X ,Y1=X2,Y2). atau loop, disebut berdimensi dan mempunyai dimensi ukuran panjang dan luas.
 - d. Data permukaan (surface) merupakan area dengan besaran (X,Y,Z) disebut berdimensi 3, dan mempunyai ukuran panjang luas dan ketinggian.



Gambar II.1. Layer Vektor
Sumber : (Solehuddin Sitorus, 2013, 2)

2. Model data Raster

Model data raster merupakan data sangat sederhana, dimana setiap informasi disimpan dalam petak petak bujursangkar (grid), yang membentuk sebuah bidang. Petak-petak bujur sangkar itu disebut dengan pixel (picture element). Posisi sebuah pixel dinyatakan dengan baris ke- m dan kolom ke-n. Data yang disimpan dalam format ini data hasil scanning , seperti gambar digital (citra dengan format BMP, JPG dan lain – lain), citra satelit digital (landsat , SPOT dan lain – lain) dan dapat dilihat pada gambar.



Gambar II.2. Layer Raster
Sumber : (Solehuddin Sitorus, 2013, 2)

II.6. Animasi 3 Dimensi

Objek 3D adalah representasi dari data geometrik 3 dimensi sebagai hasil dari pemrosesan dan pemberian efek cahaya terhadap grafika komputer 2d. Hasil ini kadang kala ditampilkan secara waktu nyata (*real time*) untuk keperluan simulasi. Secara umum prinsip yang dipakai adalah mirip dengan objek 2d, dalam hal penggunaan algoritma, grafika vektor, model frame kawat (*wire frame model*), dan grafika rasternya.

Objek 3D sering disebut sebagai model 3D. Namun, model 3D ini lebih menekankan pada representasi matematis untuk objek 3 dimensi. Data matematis ini belum bisa dikatakan sebagai gambar grafis hingga saat ditampilkan secara visual pada layar komputer atau printer. Proses penampilan suatu model matematis ke bentuk citra 2d biasanya dikenal dengan proses 3D rendering (*Angga Prasetio Romadhon ; 2013 : 222*)

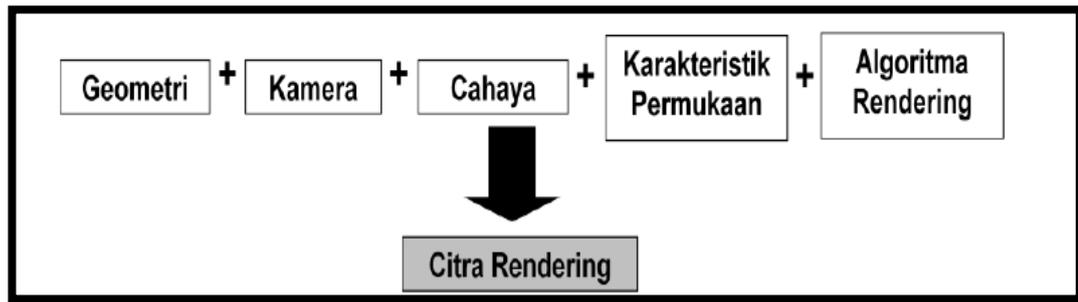
II.6.1. Konsep Dasar Rendering Objek 3D

Rendering merupakan sebuah proses untuk menghasilkan sebuah citra 2D dari data 3D. Proses ini bertujuan untuk memberikan visualisasi pada user mengenai data 3D tersebut melalui monitor atau pencetak yang hanya dapat menampilkan data 2D.

Gambar yang dibuat melalui proses rendering digital adalah gambar digital atau raster image, jenis gambar yang sama dengan yang biasa kita lihat sehari-hari pada desktop komputer atau wallpaper. Gambar digital tersebut dibuat melalui proses rendering digital sebagai langkah besar terakhir sebelum disusun menjadi animasi. Animasi sebagai tujuan akhir biasa digunakan dalam film, video game, permainan komputer, simulator, dan untuk efek khusus di televisi. Masing-masing menggunakan proses rendering digital yang menggunakan fitur dan teknik berbeda untuk mencapai hasil yang diinginkan. (*Angga Prasetio Romadhon ; 2013 : 223*)

Secara umum, proses untuk menghasilkan rendering dua dimensi dari objek-objek 3D melibatkan 5 komponen utama :

1. Geometri
2. Kamera
3. Cahaya
4. Karakteristik Permukaan
5. Algoritma Rendering

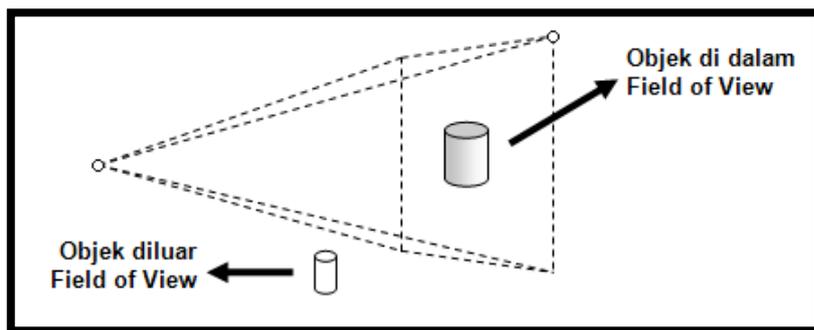


Gambar II.3. Skema Proses Rendering
Sumber : Angga Prasetio Romadhon (2013 : 223)

II.6.2. Kamera

Dalam grafika 3D, sudut pandang (point of view) adalah bagian dari kamera. Kamera dalam grafika 3D biasanya tidak didefinisikan secara fisik, namun hanya untuk menentukan sudut pandang kita pada sebuah world, sehingga sering disebut virtual camera.

Pada kamera, dikenal field of view yaitu daerah yang terlihat oleh sebuah kamera. Field of view pada grafika 3D berbentuk piramid, karena layar monitor sebuah komputer berbentuk segiempat. Objek-objek yang berada dalam field of view ini akan terlihat dari layar monitor, sedang objek-objek yang berada di luar field of view ini tidak terlihat pada layar monitor. Field of view ini sangat penting dalam pemilihan objek yang akan diproses dalam rendering. Objek-objek diluar field of view biasanya tidak akan diperhitungkan, sehingga perhitungan dalam proses rendering, tidak perlu dilakukan pada seluruh objek. (*Angga Prasetio Romadhon ; 2013 : 223*)



Gambar II.4. Sudut *Field Of View*
Sumber : Angga Prasetio Romadhon (2013 : 224)

II.6.3. Cahaya

Sumber cahaya pada grafika 3D merupakan sebuah objek yang penting, karena dengan cahaya ini sebuah world dapat terlihat dan dapat dilakukan proses rendering. Sumber cahaya ini juga membuat sebuah world menjadi lebih realistis dengan adanya bayangan dari objek-objek 3D yang ada. (*Angga Prasetio Romadhon ; 2013 : 224*)

Sebuah sumber cahaya memiliki jenis. Pada grafika 3D dikenal beberapa macam sumber cahaya, yaitu :

1. *Point light*

Memancar ke segala arah, namun intensitas cahaya yang diterima objek bergantung dari posisi sumber cahaya. Tipe ini mirip seperti lampu pijar dalam dunia nyata.

2. *Spotlight*

Memancarkan cahaya ke daerah tertentu dalam bentuk kerucut. Sumber cahaya terletak pada puncak kerucut. Hanya objek-objek yang terletak pada daerah kerucut tersebut yang akan nampak.

3. *Ambient light*

Cahaya latar/alam. Cahaya ini diterima dengan intensitas yang sama oleh setiap permukaan pada benda. Cahaya latar tersebut dimodelkan mengikuti apa yang terjadi di alam, dalam keadaan tanpa sumber cahaya sekalipun, benda masih dapat dilihat.

4. *Directional light*

Memancarkan cahaya dengan intensitas sama ke suatu arah tertentu. Letak tidak mempengaruhi intensitas cahayanya. Tipe ini dapat menimbulkan efek seolah-olah sumber cahaya berada sangat jauh dari objek .

5. *Parallel point*

Sama halnya dengan *directional light*, hanya pencahayaan ini memiliki arah dan posisi.

II.7. Website

Website (Situs *Web*) merupakan kumpulan dari halaman-halaman *web* yang berhubungan dengan *file-file* lain yang terkait. Dalam sebuah *website* terdapat suatu halaman yang dikenal dengan sebutan *home page*. *Home page* adalah sebuah halaman yang pertama kali dilihat ketika seseorang mengunjungi *website*. Dari *home page*, pengunjung dapat mengklik *hyperlink* untuk pindah kehalaman lain yang terdapat dalam *website* tersebut. (Yoni Widhiarso ; 2013 : 3)

Secara umum *website* mempunyai fungsi sebagai berikut :

1. Fungsi Komunikasi

Beberapa fasilitas yang memberikan fungsi komunikasi, seperti : *chatting, web base email* dan lain-lain.

2. Fungsi Informasi

Fungsi informasi *website* seperti : *News, Profile, Library*, referensi dan lain-lain.

3. Fungsi *Intertainment*

Website mempunyai fungsi hiburan. Misalnya *web-web* yang menyediakan *game on-line, music on-line* dan lain-lain.

II.8. Aplikasi Web

Aplikasi Web adalah sebuah sistem informasi yang mendukung interaksi pengguna melalui antarmuka berbasis Web. Fitur-fitur aplikasi web biasanya berupa *data persistence*, mendukung transaksi dan komposisi halaman web dinamis yang dapat dipertimbangkan sebagai hibridisasi, antara hipermedia dan sistem informasi.

Aplikasi Web adalah bagian dari client-side yang dapat dijalankan oleh browser web. Client-side mempunyai tanggung jawab untuk pengekseskuan proses bisnis.

Interaksi Web dibagi ke dalam tiga langkah, yaitu

1. Permintaan

Pengguna mengirimkan permintaan ke server Web, biasanya via halaman Web yang di tampilkan pada browser Web.

2. Pemrosesan

Server Web menerima permintaan yang dikirimkan oleh pengguna, kemudian memproses permintaan tersebut.

3. Jawaban

Browser menampilkan hasil dari permintaan pada jendela browser.

Halaman Web bias terdiri dari beberapa jenis informasi grafis (tekstual dan multimedia). Kebanyakan komponen grafis dihasilkan dengan tool khusus, menggunakan manipulasi langsung dan editor WYSIWYG. (*Janner Simarmata ; 2010 : 56*)

Halaman Web juga dapat dibuat dengan menggunakan berbagai program agar dapat menampilkan suatu informasi di dalam browser (misalnya, Java atau PHP). Pembangunan aplikasi Web membutuhkan beberapa kualifikasi yang berbeda. Biasanya, para pekerja dalam pembangunan Web akan memegang peranan berikut :

1. Pemasaran, Untuk menetapkan target pengunjung web dan konten untuk diserahkan.
2. Perancang grafis, untuk menetapkan tampilan visual (meliputi tata letak halaman, huruf, warna, gambar, dan film).
3. Integrator HTML, untuk mengembangkan halaman HTML.
4. Pemograman, untuk menulis program (di dalam Java, PHP atau bahasa pemograman lainnya, yang dapat dikombinasikan dengan HTML).
5. Penulis konten, untuk membuat aplikasi dengan informasi agar bernilai tambah.

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

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*). Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: *metodologi booch*, *metodologi coad*, *metodologi OOSE*, *metodologi OMT*, *metodologi shlaer-mellor*, *metodologi wirfs-brock*, dsb. Masa itu terkenal dengan

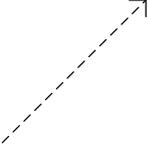
masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan. Dimulai pada bulan Oktober 1994 *Booch, Rumbaugh dan Jacobson*, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan memelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direlease draft pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh Object Management Group. Tahun 1997 UML versi 1.1 muncul, dan saat ini versi terbaru adalah versi 1.5 yang dirilis bulan Maret 2003. *Booch, Rumbaugh dan Jacobson* menyusun tiga buku serial tentang UML pada tahun 1999. Sejak saat itulah UML telah menjelma menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek.

II.9.1. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem 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 bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-include akan dipanggil setiap kali *use case* yang meng-include dieksekusi 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. (Yuni Sugiarti ; 2013 : 41)

Tabel II.1. Simbol-simbol Use Case Diagram

Simbol	Notasi	Keterangan
	<i>Actor</i>	Pengguna sistem atau yang berinteraksi langsung dengan sistem, bisa manusia, aplikasi, ataupun objek lain.
	<i>Use Case</i>	Digambarkan dengan lingkaran elips dengan nama <i>use case</i> -nya tertulis ditengah lingkaran.
	<i>Assotiation</i>	Digambarkan dengan sebuah garis yang berfungsi menghubungkan <i>actor</i> dengan <i>use case</i> .
	<i>Defendency</i>	<i>Depenciency</i> merupakan relasi yang menunjukkan bahwa perubahan pada salah satu <i>elemen</i> memberi pengaruh pada <i>elemen</i> lain.

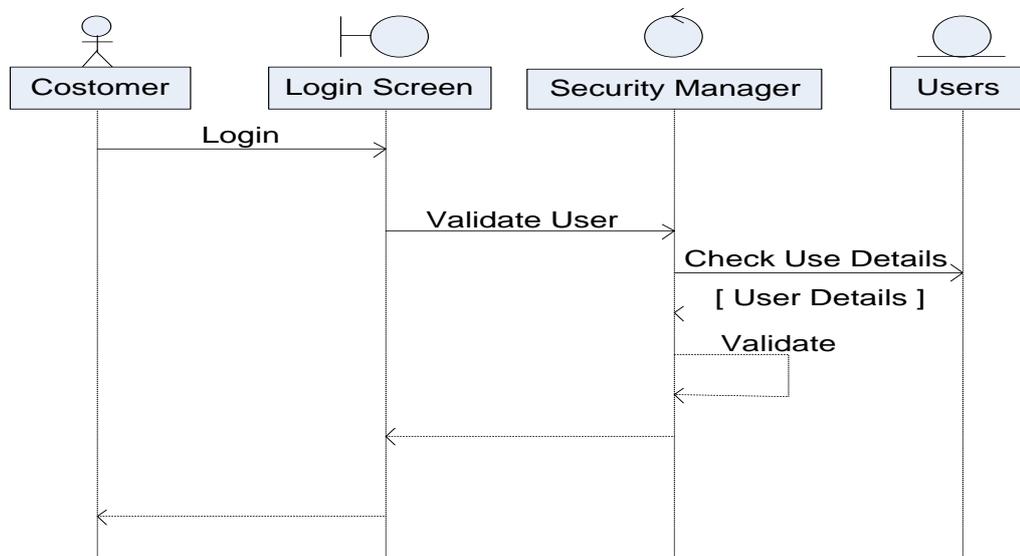
<p style="text-align: center;"> <<extended>> -----> </p>	<p><i>Extended</i></p>	<p><i>Extended</i> menunjukkan bahwa suatu bagian dari <i>elemen</i> digaris tanpa panah bisa disisipkan kedalam <i>elemen</i> yang ada digaris dengan panah.</p>
--	------------------------	---

Sumber : Yuni Sugiarti (2013 : 74)

II.9.2. Sequence Diagram

Diagram *Sequence* menggambarkan kelakuan/prilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya diagram *sequence* yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram *sequence* sehingga semakin banyak *use case* yang didefinisikan maka diagram *sequence* yang harus dibuat juga semakin banyak.



Gambar II.5. Contoh Sequence Diagram

Sumber : Yuni Sugiarti (2013 : 63)

II.9.3. Activity Diagram

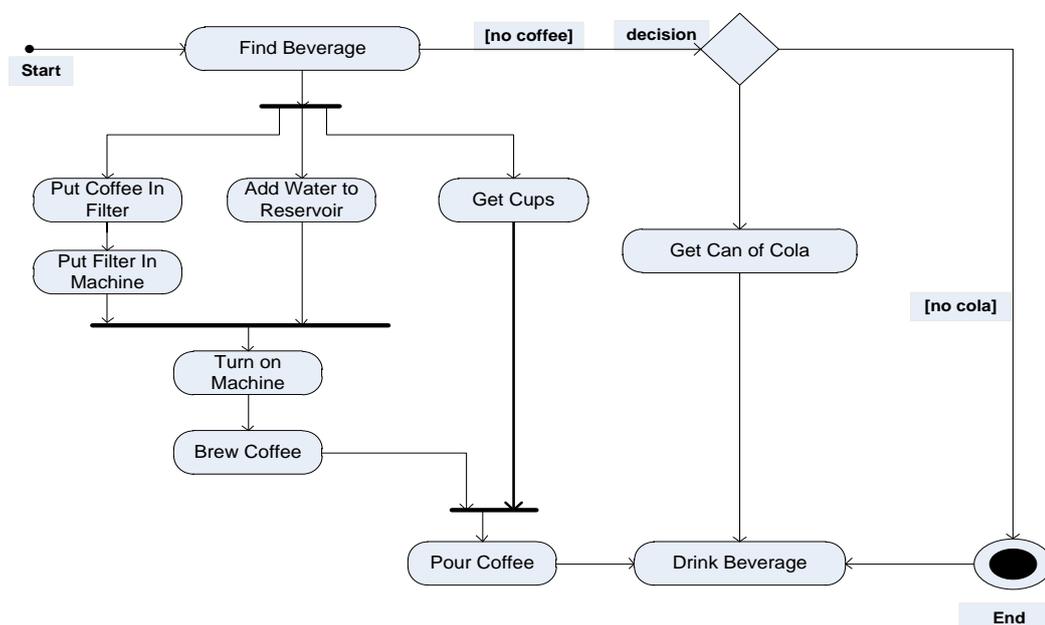
Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

Activity diagram dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.



Gambar II.6. Activity Diagram

Sumber : Yuni Sugiarti (2013 : 76)

II.10. *Quantum GIS*

Quantum GIS merupakan salah satu perangkat lunak *open source* di bawah proyek resmi dari *Open Source Geospatial Foundation* (OSGeo) yang dapat dijalankan dalam sistem operasi Windows, Mac OSX, Linux dan Unix. Aplikasi ini menawarkan pengolahan data geospasial dengan berbagai format dan fungsionalitas vektor, raster dan database. Untuk keperluan analisis spasial, aplikasi ini telah cukup lengkap karena telah terintegrasi dengan perangkat lunak GRASS. Pemanfaatan perangkat lunak Quantum GIS ini dapat digunakan sebagai pilihan alternatif dari software SIG komersial seperti ArcView maupun ArcGIS. Quantum GIS dapat diakses melalui situs resmi yang beralamatkan www.qgis.org. (*Saddam Hussein ; 2012 : 93*)

II.11. *PHP*

PHP singkatan dari PHP : *Hypertext Preprocessor* yaitu bahasa pemrograman web server-side yang bersifat open source. PHP merupakan script yang terintegrasi dengan HTML dan berada pada server (Server side HTML embedded scripting). PHP adalah script yang digunakan untuk membuat halaman website yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh client. Mekanisme ini menyebabkan informasi yang diterima client selalu yang terbaru/ up to date. Semua script PHP dieksekusi pada server dimana script tersebut dijalankan. (*Anhar,ST ; 2013 : 3*).

PHP merupakan sebuah bahasa pemrograman web yang memiliki sintak atau aturan dalam menuliskan script atau kode-kodenya. Untuk menjelaskan cara penulisan kode PHP, bisa kita lihat pada empat macam cara penulisan kode PHP, yaitu :

1. `<? echo ("ini adalah script PHP\n"); ?>`
2. `<? php echo ("ini juga script PHP\n"); ?>`
3. `<script language="php">`
`echo ("LATihan menulis script PHP");`
`</script>`
4. `<% echo ("kalau yang ini mirip ASP"); %>`. (Anhar,ST ; 2013 : 23-24).

II.12. MySQL

MySQL *database server* adalah RDBMS (*Relasional Database Management System*) yang dapat menangani data yang bervolume besar. Meskipun begitu, tidak menurut *resource* yang besar. MySQL adalah *database* yang paling populer diantara *database-database* yang lain. (Wahana Komputer ; 2010 ; 5)

MySQL adalah program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan multi *user*. MySQL memiliki dua bentuk lisensi, yaitu *free software* dan *shareware*. Penulis sendiri dalam menjelaskan buku ini menggunakan MySQL yang *free software* karena bebas menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensi, yang berada di bawah lisensi GNU/GPL (*general public license*), yang dapat anda download pada alamat resminya <http://www.mysql.com>. MySQL sudah

cukup lama dikembangkan, beberapa fase penting dalam pengembangan MySQL adalah sebagai berikut :

1. MySQL diliris pertama kali secara internal pada 23 Mei 1995.
2. Versi *Windows* diliris pada 8 Januari 1998 untuk *windows 95* dan *Windows NT*.
3. Versi 3.23 : beta dari Juni 2000, dan diliris pada Januari 2001.
4. Versi 4,1 : beta dari bulan Juni 2004, diliris pada bulan Oktober 2004 (*R-Trees* dan *B-Trees*, *Subqueries*, *Prepared Statements*).
5. Versi 5.0 : beta dari bulan Maret 2005, diliris pada Oktober 2005. (*Cursor*, *Stored Procedure*, *Trigger*, *Views XA Transaction*).
6. Versi 5.1 : diliris 27 November 2008 (*Event Scheduler*, *Partitioning*, *Plug-In API*, *Row-Based Replication*, *Server Log Table*).

II.13. Perangkat Lunak 3D

Perangkat lunak modelling 3D yang dimaksud disini adalah perangkat lunak untuk membuat objek tiga dimensi untuk nantinya dimasukkan dalam peta tiga dimensi yang telah dibuat. Perangkat lunak modelling 3D telah banyak tersedia dalam bentuk berbayar ataupun gratis. Berikut adalah sedikit ulasan mengenai perangkat lunak modelling tiga dimensi yang digunakan dalam pengerjaan penelitian ini. *Autodesk 3DS Max* Sebuah perangkat lunak keluaran Autodesk yang digunakan untuk melakukan modelling tiga dimensi, animasi, hingga rendering. *3D Studio Max* dapat melakukan export hasil modelling yang

dapat diterima oleh Unreal Development Kit seperti .ASE, .FBX dan .DAE.
(Berta Sihite ; 2013 : 398)

II.14. 3D Max

Sejak pertamakali dirilis, *3D Studio Max* menjadi pemimpin aplikasi pembangunan animasi tiga dimensi. Sejak versi ke empat, *Discreet*, produsen *3D Studio Max*, berusaha untuk meluaskan area fungsinya sehingga dapat digunakan untuk membuat animasi bagi *web* atau *film*. Versi terbaru, yaitu versi 13, sudah mengarah kepada perluasan fungsi tersebut. Hal ini ditunjukkan dengan adanya pengembangan pada *polymodeling*, *mapping* dan beberapa revisi pada *tool* untuk animasi. Namun dari fitur yang ada, fitur yang paling menarik dari 3D Studio max adalah *reactor*. *Reactor* ini terintegrasi dengan *inface* dari *3D max* dan menyediakan *tool* untuk membuat simulasi. *3D Studio Max* ini sering digunakan untuk membuat model-model rumah atau furniture. Selain itu, banyak pula digunakan di dalam seni digital atau pembuatan *game*.(Leno Sambodo ; 2013 : 3)