

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Penelitian yang terkait dengan metode yang digunakan penulisan skripsi ini adalah penelitian yang dilakukan oleh Ryan Herwan Dwi Putra (2015) dengan judul penelitian Penerapan Metode Haversine Formula Pada Sistem Informasi Geografis Pengukuran Luas Tanah. Hasil pengujian akurasi yang dilakukan terhadap dua titik wilayah tanah menunjukkan bahwa sistem dengan memanfaatkan metode haversine formula menghasilkan selisih panjang sebesar 3.33% terhadap pengukuran menggunakan GPS dan 7.33% terhadap pengukuran manual. Hasil pengujian akurasi juga menunjukkan bahwa sistem menghasilkan selisih luas sebesar 3.923% terhadap pengukuran menggunakan GPS dan 7.846% terhadap pengukuran manual. Berdasarkan kuesioner yang dibagikan kepada 10 responden pegawai BPN dan melalui pengujian UAT didapatkan hasil bahwa sistem ini dinilai positif dan berhasil. Kesimpulannya sistem ini membantu pegawai BPN dalam pengukuran luas tanah dan memudahkan untuk mendapatkan informasi mengenai luas tanah yang telah diukur oleh pegawai BPN.

Penelitian lain yang terkait dengan metode yang digunakan penulisan skripsi ini adalah penelitian yang dilakukan oleh Fadhoelur Rohman (2013) dengan judul penelitian Mobile Fasilitas Untuk Pengguna Jalan Berbasis Android. Aplikasi mobile phone yang dibuat dapat diakses pada telepon genggam dengan sistem operasi Android minimal 2.3. Aplikasi dapat menampilkan fasilitas

umum yaitu SPBU sebanyak 35, Kantor Polisi sebanyak 11, dan Puskesmas/Rumah Sakit sebanyak 9.

Penelitian lain yang terkait dengan metode yang digunakan penulisan skripsi ini adalah penelitian yang dilakukan oleh Wahyuni Eka Sari (2013) dengan judul penelitian Penerapan *Jquery Mobile* dan *PHP Data Object* Pada Aplikasi Pencarian Lokasi Tempat Ibadah di Yogyakarta. Aplikasi Pencarian Lokasi Tempat Ibadah ini menggunakan layanan berbasis lokasi yang menggabungkan antara proses dari layanan *mobile* dengan posisi geografis dari penggunanya. Posisi target dapat diimplementasikan pada peta virtual yaitu *google maps*. *Google maps* memanfaatkan teknologi GPS untuk mengetahui lokasi user. Aplikasi yang dihasilkan adalah aplikasi berbasis web mobile dengan menggunakan bahasa pemrograman HTML5, CSS3, *PHP Data Object* dan *database MySQL*, serta *framework JQuery Mobile*. Perancangan aplikasi menggunakan UML (*Unified Modelling Language*).

II.2. Landasan Teori

II.2.1.Sistem

Menurut Sulindawati (2010 : 375) Sistem merupakan sekumpulan elemen-elemen yang saling terintegrasi serta melaksanakan fungsinya masing-masing untuk mencapai tujuan yang telah ditetapkan. Karakteristik sistem terdiri dari:

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-

komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. Batasan Sistem

Batasan merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung Sistem

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya.

5. Masukan Sistem

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

6. Keluaran Sistem

Keluaran sistem adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem

Suatu sistem mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya.

II.2.2. Informasi

Menurut Sulindawati (2010 : 3) Informasi adalah data yang diolah menjadi suatu bentuk yang berarti bagi penerimaannya dan bermanfaat dalam pengambilan keputusan pada saat sekarang atau yang akan datang. Informasi juga merupakan fakta-fakta atau data yang telah diproses sedemikian rupa atau mengalami proses transformasi data sehingga berubah bentuk menjadi informasi. Kualitas dari suatu informasi tergantung pada tiga hal yaitu :

1. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bisa atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan (*noise*) yang dapat merubah atau merusak informasi tersebut.
2. Tepat pada waktunya, berarti informasi yang pada penerima tidak boleh terlambat. Informasi yang sudah using tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan di dalam pengambilan keputusan. Bila pengambilan keputusan tersebut, maka dapat berakibat fatal untuk organisasi.

3. Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya. relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

II.2.3. Sistem Informasi Geografis

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

Istilah SIG juga didefinisikan sebagai sistem informasi khusus yang mengelola data yang memiliki informasi spasial (bereferensi keruangan). Dalam arti yang lebih sempit, SIG adalah sistem komputer yang memiliki kemampuan untuk membangun, menyimpan, mengelola dan menampilkan informasi bereferensi geografis, misalnya data yang diidentifikasi menurut lokasinya, dalam sebuah *database*. Para praktisi juga memasukkan orang yang membangun dan mengoperasikan SIG dan data sebagai bagian dari sistem ini. Teknologi SIG dapat digunakan untuk berbagai kepentingan, seperti investigasi ilmiah, pengelolaan

sumber daya, perencanaan pembangunan, kartografi dan perencanaan rute (Edhy Sutanta ; 2010 : 38).

Adapun beberapa jenis data yang digunakan dalam SIG yaitu meliputi:

1. Data Spasial

Menurut Mohd. Ichsan (2012 : 51), Sebagian besar data yang akan ditangani dalam SIG merupakan data spasial, data yang berorientasi geografis. Data ini memiliki sistem koordinat tertentu sebagai dasar referensinya dan mempunyai dua bagian penting yang berbeda dari data lain, yaitu informasi lokasi (spasial) dan informasi deskriptif (atribut) yang dijelaskan berikut ini:

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

2. Format Data Spasial

Menurut Mohd. Ichsan (2012 : 51), Secara sederhana format dalam bahasa komputer berarti bentuk dan kode penyimpanan data yang berbeda antara *file* satu dengan lainnya. Dalam SIG, data spasial dapat direpresentasikan dalam dua format, yaitu :

a. Data vektor

Data vektor merupakan bentuk bumi yang direpresentasikan ke dalam kumpulan garis, area (daerah yang dibatasi oleh garis yang berawal dan

berakhir pada titik yang sama), titik dan *nodes* (titik perpotongan antara dua buah garis).

b. Data *raster*

Data *raster* (disebut juga dengan sel *grid*) adalah data yang dihasilkan dari sistem penginderaan jauh. Pada data *raster*, obyek geografis direpresentasikan sebagai struktur sel *grid* yang disebut dengan *pixel* (*picture element*).

3. Data Vektor

Data Vektor merupakan bentuk bumi yang dipresentasikan ke dalam kumpulan garis, area (daerah yang dibatasi oleh garis yang berawal dan berakhir pada titik yang sama), titik dan *nodes* (merupakan titik perpotongan antara dua buah garis). Keuntungan utama dari format data vektor adalah ketepatan dalam merepresentasikan fitur titik, batasan dan garis lurus. Hal ini sangat berguna untuk analisa yang membutuhkan kepetatan posisi, misalnya pada basis data batas-batas kedaster. Contoh pengguna lainnya adalah untuk mendefinisikan hubungan spasial dari beberapa fitur. Kelemahan data vektor yang utama adalah ketidakmampuannya dalam mengakomodasi perubahan *gradual* (Bramantiyo Marjuki ; 2014 : 5).

4. Data *Raster*

Data *raster* (atau disebut juga dengan sei *grid*) adalah data yang dihasilkan dari sistem penginderaan jauh. Pada data *raster*, obyek geografis direpresentasikan sebagai struktur sel *grid* yang disebut dengan *pixel* (*picture element*). Pada data *raster*, resolusi (definisi visual) tergantung pada ukuran *pixel*-nya. Dengan kata lain, resolusi *pixel* menggambarkan ukuran sebenarnya di

permukaan bumi yang diwakili oleh setiap *pixel* pada citra. Semakin kecil ukuran permukaan bumi yang direpresentasikan oleh satu sel, semakin tinggi resolusinya. Data *raster* sangat baik untuk merepresentasikan batas-batas yang berubah secara *gradual*, seperti jenis tanah, kelembaban tanah, vegetasi, suhu tanah dan sebagainya. Keterbatasan utama dari data *raster* adalah besarnya ukuran *file*, semakin tinggi resolusi *grid*nya semakin besar pula ukuran *filenya* dan sangat tergantung pada kapasitas perangkat keras yang tersedia (Bramantiyo Marjuki ; 2014 : 5).

II.2.4. Starbucks Coffee

Starbucks dimulai sejak tahun 1971, dimana saat itu ada tiga orang akademis yaitu Jerry Baldwin seorang guru bahasa Inggris, Zev Siegel seorang guru sejarah, dan Gordon Bowker membuka suatu toko yaitu *Starbucks Coffee, Tea and Spice di Pikes Place Market, Seattle*. Tiga partner ini berbagi kecintaan akan kopi terbaik, dan teh yang eksotik. Masing-masing orang menginvestasikan dananya sebesar \$1,350 dan meminjam kepada bank sebesar \$5000 untuk membuka tokonya di *Pikes Place Market, Seattle*. Logo dari perusahaan pertama kali di desain oleh seorang seniman, dimana bergambar seorang putri duyung berbuntut dua dan dikelilingi dengan lingkaran bertuliskan nama perusahaan. Di Indonesia, *Starbucks Coffee* dibuka pertama kali pada tanggal 17 Mei 2002 di Plaza Indonesia, Jakarta. *Starbucks Coffee* di Indonesia dikelola oleh PT. Sari Coffe Indonesia. Dalam perkembangannya *Starbucks Coffee* di Indonesia telah memiliki banyak gerai baru yang tersebar pada kota-kota besar di Indonesia

seperti Surabaya, Bandung, Medan, Yogyakarta, Bali, Semarang, Malang. Selain memfokuskan pada menu olahan kopi, di Indonesia *Starbucks* juga menyediakan makanan ringan seperti roti dan *cake* sebagai menu pelengkap. Berbagai *merchandise* seperti *mug*, *tumbler*, *pitcher* hingga termos mini yang dapat digunakan untuk berpartisipasi dalam program *Starbucks on the go* tersedia pada setiap gerai-gerai *Starbucks Coffee* di Indonesia. *Starbucks Coffee* memiliki komitmen yang cukup tinggi dalam program penyelamatan lingkungan selama beberapa tahun terakhir ini. Dimulai pada bulan Maret 2006 *Starbucks Coffee* meluncurkan cangkir kopi yang 10% dari bahan bakunya terbuat dari materi daur ulang. Dapat diketahui bahwa bahan baku cangkir kopi berasal dari bubur kayu sehingga dengan mengurangi 10% materi bahan tersebut *Starbucks* bisa menyelamatkan 300.000 pohon setiap tahunnya. Program *Go Green* lainnya yang diluncurkan oleh *Starbucks Coffee* adalah pemberian diskon sebesar 10 sen bagi pengunjung yang membawa cangkir kopi sendiri ketika menikmati suguhan kopi dari *Starbucks Coffee*. Dengan melibatkan pengunjung dalam gerakan reuse atau pemakaian kembali cangkir kopi bekas pakai, perusahaan yang memiliki situs resmi beralamat di <http://www.Starbucks.com/> ini dapat mengurangi 109 truk sampah cangkir kopi setiap tahunnya. Hal tersebut adalah sebuah fakta yang sangat menggembirakan terutama bagi penggemar kopi yang kini bisa menikmati sajian kopi dari *Starbucks Coffee* dengan lebih ramah lingkungan. Program semacam ini juga turut diterapkan pada *Starbucks* Indonesia, yang dikenal dengan program *tumbler Starbucks on the go*. (Devina Susilo ; 2014 : 12).

II.2.5. Metode Haversine

Posisi di bumi dapat direpresentasikan dengan posisi garis lintang (*latitude*) dan bujur (*longitude*). Untuk menentukan jarak antara dua titik di bumi berdasarkan letak garis lintang dan bujur, ada beberapa rumusan yang digunakan. Semua rumusan yang digunakan berdasarkan bentuk bumi yang bulat (*spherical earth*) dengan menghilangkan faktor bahwa bumi itu sedikit elips (*elipsodial factor*).

$$\Delta\text{lat} = \text{lat2} - \text{lat1} \quad (1)$$

$$\Delta\text{long} = \text{long2} - \text{long1} \quad (2)$$

$$a = \sin^2(\Delta\text{lat}/2) + \cos(\text{lat1}) \cdot \cos(\text{lat2}) \cdot \sin^2(\Delta\text{long}/2) \quad (3)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a}) \quad (4)$$

$$d = R \cdot c \quad (5)$$

Keterangan :

R = jari-jari bumi sebesar 6371(km)

Δlat = besaran perubahan latitude

Δlong = besaran perubahan longitude

c = kalkulasi perpotongan sumbu

d = jarak (km) (Wahyuni Eka Sari ; 2013 : 72).

II.2.6. Android

Android adalah sistem operasi disematkan pada *gadget*, baik itu *handphone*, *tablet*, juga sekarang sudah merambah ke kamera digital dan jam tangan. Saat ini *gadget* berbasis android, baik itu *tablet* atau *handphone*, begitu

digandrungi. Selain harganya yang semakin terjangkau, juga banyak varian spesifikasi yang bisa dipilih sesuai kebutuhan.

Perkembangan android sangat cepat. Di awal tahun 2002 ada 200 juta pengguna aktif *android*, dan *google play* mampu menampung 400.000 aplikasi yang siap digunakan, dan total mencapai 10 triliun kali aplikasi yang sudah di *download* lewat *android market*, pertumbuhan yang luar biasa. Jumlah ini diyakini akan terus bertambah seiring waktu dan perkembangan teknologi.

Memasuki tahun 2013, versi android yang terbaru adalah 4.x ada versi 4.0 *Ice Cream Sandwich* (ICS) ada juga versi 4.1 dan 4.2 yang diberi nama *Jelly Bean*. Namun, di pasar indonesia versi *Jelly Bean* masih jarang ditemukan, bahkan sampai sekarang *Acer Iconia A500* belum juga merilis *update* resmi dari ICS ke *Jelly Bean* (Agus Wahadyo ; 2013 : 3).

II.2.7. Database

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

II.2.8. MySQL

MySQL merupakan salah satu sistem *database* yang sangat handal karena menggunakan sistem SQL. Pada awalnya SQL berfungsi sebagai bahasa penghubung antara program *database* dengan bahasa pemrograman yang kita gunakan. Dengan adanya SQL maka para pemrogram jaringan dan aplikasi tidak mengalami kesulitan sama sekali di dalam menghubungkan aplikasi yang mereka buat. Setelah itu SQL dikembangkan lagi menjadi sistem *database* dengan munculnya MySQL. MySQL merupakan *database* yang sangat cepat, beberapa user dapat menggunakan secara bersamaan dan lebih lengkap dari SQL. MySQL merupakan salah satu *software* gratis yang dapat di-download melalui situsnya. MySQL merupakan sistem manajemen *database*, relasional sistem *database* dan *software open source*. (Stendy B. Sakur ; 2015 : 58)

II.2.9. Teknik Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional.

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang telah ditemukan.

Adapun bentuk-bentuk Normalisasi yaitu :

1. Bentuk Normal Pertama (1NF)

Contoh yang kita gunakan di sini adalah sebuah perusahaan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri. Di bawah ini terdapat Bentuk Normal Pertama yang dapat dilihat pada tabel II.1.

Tabel II.1. Contoh Tabel Bentuk Normal Pertama (1NF)

p#	Status	Kota	b#	Qty
p1	20	Yogyakarta	b1	300
p1	20	Yogyakarta	b2	200
p1	20	Yogyakarta	b3	400
p1	20	Yogyakarta	b4	200
p1	20	Yogyakarta	b5	100
p1	20	Yogyakarta	b6	100
p2	10	Medan	b1	300
p2	10	Medan	b2	400
p3	10	Medan	b2	200
p4	20	Yogyakarta	b2	200
p4	20	Yogyakarta	b4	300
p4	20	Yogyakarta	b5	400

(Janner Simarmata ; 2010 : 78)

2. Bentuk Normal Kedua (2NF)

Definisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada seluruh kolom yang membentuk kunci utama. Di bawah ini terdapat Bentuk Normal Kedua yang dapat dilihat pada tabel II.2.

Tabel II.2. Contoh Tabel Bentuk Normal Kedua (2NF)

Pemasok2			Barang		
p#	Status	Kota	p#	b#	Qty
P1	20	Yogyakarta	p1	b1	300

P2	10	Medan
P3	10	Medan
P4	20	Yogyakarta
P5	30	Bandung

p1	b2	200
p1	b3	400
p1	b4	200
p1	b5	100
p1	b6	100
p2	b1	300
p2	b2	400
p3	b2	200
p4	b2	200
p4	b4	300
p4	b5	400

(Janner Simarmata ; 2010 : 78)

3. Bentuk Normal Ketiga (3NF)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara definisi, sebuah tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya. Di bawah ini terdapat tabel Bentuk Normal Ketiga yang dapat dilihat pada tabel II.3.

Tabel II.3. Contoh Tabel Bentuk Normal Ketiga (3NF)

Pemasok Kota		Kota Status	
p#	Kota	Kota	Status
P1	Yogyakarta	Yogyakarta	20
P2	Medan	Medan	10
P3	Medan	Yogyakarta	20
P4	Yogyakarta	Bandung	30
P5	Bandung		

(Janner Simarmata ; 2010 : 78)

4. Bentuk Normal Boyce-Code (BCNF)

Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3NF sudah cukup karena sangat

jarang entitas yang berada pada 3NF bukan merupakan 4NF dan 5NF. Di bawah ini terdapat Bentuk Normal BCNF yang dapat dilihat pada tabel II.4.

Tabel II.4. Contoh Tabel Bentuk Normal BCNF

Pegawai Proyek		Pegawai Ahli	
peg#	Pry#	Peg#	Ahli
1211	P1	1211	Analisis
1211	P3	1211	Perancangan
		1211	Pemrograman

(Janner Simarmata ; 2010 : 78)

5. Bentuk Normal Tahap Keempat

Sebuah tabel relasional berada pada bentuk normal keempat (4NF) jika dia dalam BCNF dan semua ketergantungan multivalued merupakan ketergantungan fungsional. Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan multivalued (MVD). Di bawah ini terdapat Bentuk Normal Keempat yang dapat terlihat pada tabel II.5.

Tabel II.5. Contoh Tabel Bentuk Normal Keempat (4NF)

peg#	Pry#	Ahli
1211	11	Perancangan
1211	28	Pemrograman

(Janner Simarmata ; 2010 : 78)

6. Bentuk Normal Kelima (5NF)

Sebuah tabel berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan

gabungan (*join dependence*) (Janner Simarmata ; 2010 : 76). Di bawah ini terdapat Bentuk Normal Kelima dapat dilihat pada tabel II.6.

Tabel II.6. Contoh Tabel Bentuk Normal Kelima

peg#	Pry#	Ahli
1211	11	Perancangan
1211	28	Pemrograman

(Janner Simarmata : 2010 : 78)

II.2.10. UML (*Unified Modeling Language*)

Menurut Windu Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

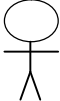
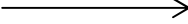
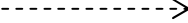
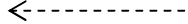
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat

dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Di bawah ini terdapat Simbol-simbol yang digunakan dalam *use case* diagram yang terdapat pada tabel II.7.

Tabel II.7. Simbol Use Case

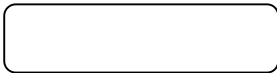
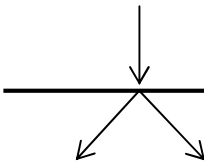
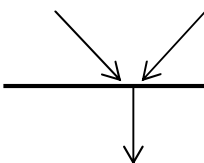
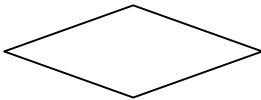
Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Windu Gata ; 2013 : 4)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Di bawah ini terdapat Simbol-simbol yang digunakan dalam *activity diagram* yang terdapat pada tabel II.8.

Tabel II.8. Simbol *Activity Diagram*

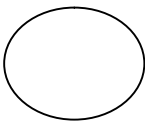
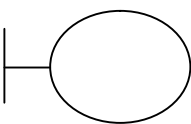
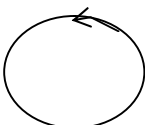
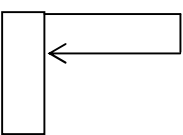
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Windu Gata ; 2013 : 6)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Di bawah ini terdapat Simbol-simbol yang digunakan dalam *sequence diagram* yang dapat dilihat pada tabel II.9.

Tabel II.9. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Windu Gata ; 2013 : 7)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti. Di bawah ini terdapat tabel *Multiplicity Class Diagram* yang dapat dilihat pada tabel II.10.

Tabel II.10. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Windu Gata ; 2013 : 9)