BAB 2

TINJAUAN PUSTAKA

2.1 Profil Tempat Penelitian

Dalam melaksanakan penelitian diperlukan tempat yang dapat menjadi narasumber guna kebutuhan penelitian.

2.1.1 Sejarah Instansi

Pusat Vulkanologi dan Mitigasi Bencana Geologi adalah satu unit dilingkungan Badan Geologi, Departemen Energi dan Sumberdaya Mineral yang dibentuk berdasarkan Peraturan Menteri Energi dan Sumberdaya Mineral Nomor 0030 tahun 2005 tentang Organisasi dan Tata Kerja Departemen Energi dan Sumberdaya Mineral. Di dunia internasional *agency* ini lebih dikenal dengan sebutan *Volcanological Survey of Indonesia* (*VSI*).

Organisasi ini terbentuk setelah beberapa kali berganti nama yang berawal setelah meletusnya Gunung Kelut di Jawa Timur tahun 1919 yang menimbulkan korban manusia lebih dari 5000 orang. Pada tanggal 16 September 1920 dibentuk *Vulkaan Bewakings Dients* (Dinas Penjagaan Gunungapi) di bawah naungan *Dients Van Het Mijnwezen* dan pada tahun 1922 diresmikan menjadi *Volcanologische Onderzoek* (*VO*), yang tahun 1939 di dunia internasional dikenal sebagai *Volcanological Survey*. Sejak tahun 1920 - 1941, *Volcanologische Onderzoek* ini telah membangun beberapa pos penjagaan gunungapi, yaitu Pos Gunung Krakatau di Pulau Panjang, Pos Gunung Tangkubanparahu, Pos Gunung Papandayan, Pos Kawah Kamojang, Pos Gunung Merapi (Babadan, Krinjing, Plawangan, Ngepos), Pos Gunung Kelut, Pos Gunung Semeru dan Pos Kawah

Ijen. Pada saat pendudukan Jepang, kegiatan penjagaan gunungapi ditangani oleh Kazan Chosabu selama periode 1942-1945.

Setelah Indonesia merdeka dibentuk Dinas Gunung Berapi (DGB) di bawah Jawatan Pertambangan, kemudian 1966 dirubah menjadi Urusan Vulkanologi di bawah Direktorat Geologi dan selanjutnya pada tahun 1976 berubah lagi menjadi Sub Direktorat Vulkanologi di bawah Direktorat Geologi, Departemen Pertambangan. Berdasarkan Keputusan Menteri Pertambangan dan Energi No. 734 Tahun 1978 terbentuklah Direktorat Vulkanologi di bawah Direktorat Jenderal Pertambangan Umum, Departemen Pertambangan dan Energi. Perkembangan organisasi Departemen Pertambangan dan Energi berdasarkan Keputusan Menteri Pertambangan dan Energi Nomor 1092 Tahun 1984 dan Keputusan Menteri Pertambangan dan Energi Nomor 1748 Tahun 1992 terbentuk Direktorat Vulkanologi di bawah Direktorat Jenderal Geologi dan Sumberdaya Mineral.

Sejak tahun 2001 sampai saat ini, berdasarkan Keputusan Menteri Energi dan Sumberdaya Mineral Nomor 1915 Tahun 2001, urusan gunungapi, gerakan tanah, gempabumi, tsunami, erosi dan sedimentasi ditangani oleh Direktorat Vulkanologi dan Mitigasi Bencana Geologi. Tugas utamanya adalah melaksanakan perumusan kebijaksanaan, standardisasi, bimbingan teknis dan evaluasi bidang vulkanologi dan mitigasi bencana alam geologi. Tujuan dibentuknya Direktorat Vulkanologi dan Mitigasi Bencana Geologi adalah secara umum mencakup pengelolaan informasi potensi kegunungapian dan pengelolaan mitigasi bencana alam geologi, sedangkan missi yang diemban adalah meminimalkan korban jiwa manusia dan kerugian harta benda dari bencana

geologi. Setelah bergabung dengan Badan Geologi, Direktorat Vulkanologi dan Mitigasi Bencana Geologi berubah nama institusinya menjadi Pusat Vulkanologi dan Mitigasi Bencana Geologi (PVMBG), *Centre of Volcanology and Geological Hazard Mitigation*. Di dunia Internasional kantor kami ini lebih dikenal dengan sebutan *Volcanological Survey of Indonesia* (VSI).

2.1.2 Logo Instansi

Pada gambar 2.1 merupakan logo instansi.



Gambar 2.1 Logo instansi pusat Vulkanologi dan Mitigasi Bencana Geologi

2.1.3 Badan Hukum Instansi

Pusat Vulakanologi dan Mitigasi Bencana Geologi merupakan Lembaga Milik Pemerintah Dibawah Departemen Energi dan Sumber Daya Mineral Badan Geologi Pusat Vulkanologi dan Mitigasi Bencana Geologi (*Surat Keputusan Menteri No.52 thn 2006 tgl.20 Oktober 2006*).

2.1.4 Struktur Organisasi dan *Job Description*

Struktur organisasi Pusat Vulkanologi dan Mitigasi Bencana Geologi seperti pada gambar 2.2 sebagai berikut :



Gambar 2.2 Struktur organisasi pusat Vulkanologi dan Mitigasi Bencana Geologi

Tugas Pokok Tiap Bidang

1. Bidang Pengamatan dan Penyelidikan Gunungapi

   Bidang Pengamatan dan Penyelidikan Gunungapi mempunyai tugas melaksanakan pengamatan dan penyelidikan gunung api.

   b.1.a Subbidang Pengamatan Gunungapi

   Subbidang pengamatan gunungapi mempunyai tugas melakukan pengumpulan bahan, penelaahan, pelaksanaan dan evaluasi atas pengamatan gunungapi, penetapan status, peringatan serta rekomendasi penanggulangannya.

   b.1.b Subbidang Penyelidikan Gunungapi

   Subbidang Penyelidikan Gunungapi mempunyai tugas melakukan pengumpulan bahan, penelaahan, pelaksanaan, dan evaluasi penyelidikan gunungapi.

2. Bidang Pengamatan Gempabumi dan Gerakan Tanah

   Bidang Pengamatan Gempabumi dan Gerakan Tanah mempunyai tugas melaksanakan pengamatan gempa bumi dan gerakan tanah.

   b.2.a Subbidang Pengamatan Gempabumi

   Subbidang Pengamatan Gempabumi mempunyai tugas melakukan pengumpulan bahan, penelaahan, pelaksanaan, dan evaluasi mitigasi geempabumi.

   b.2.b Subbidang Pengamatan Gerakan Tanah

   Subbidang Pengamatan Gerakan Tanah mempunyai tugas melakukan pengumpulan bahan, penelaahan, pelaksanaan, dan evaluasi atas gerakan tanah.

3. Bidang Evaluasi Potensi Bencana

   Bidang Evaluasi Potensi Bencana mempunyai tugas melaksanakan evaluasi potensi bencana geologi

   b.3.a Subbidang Evaluasi Bencana Gunung Api

   Subbidang Evaluasi Bencana Gunung Api mempunyai tugas melakukan pengumpulan bahan, penelaahan penyiapan, serta pelaksanaan atas perencanaan program, akuntabilitas kinerja dan evaluasi potensi bencana gunungapi.

   b.3.b Subbidang Evaluasi Bencana Geologi

   Subbidang Evaluasi Bencana Geologi mempunyai tugas melakukan pengumpulan bahan penelaahan, penyiapan, serta pelaksanaan atas perencanaan program, akuntabilitas kinerja dan evaluasi potensi bencana geologi.

4. BPPTK (Balai Penyelidikan dan Pengembangan teknologi Kegunungapian)

   BPPTK mempunyai tugas melaksanakan penyelidikan Gunungapi Merapi, pengembangan metode, analisis, teknologi dan instrumentasi serta pengelolaan sarana dan prasarana laboratorium kegunungapian dan mitigasi bencana geologi.

2.2 Landasan Teori

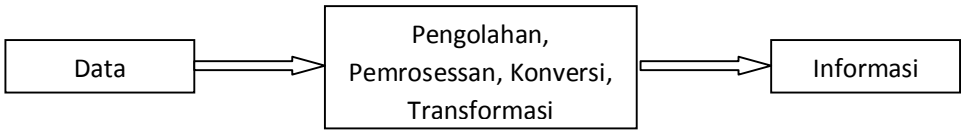
Teori-teori yang mendukung dari pengembangan sistem ini.

2.2.1 Konsep Dasar Informasi

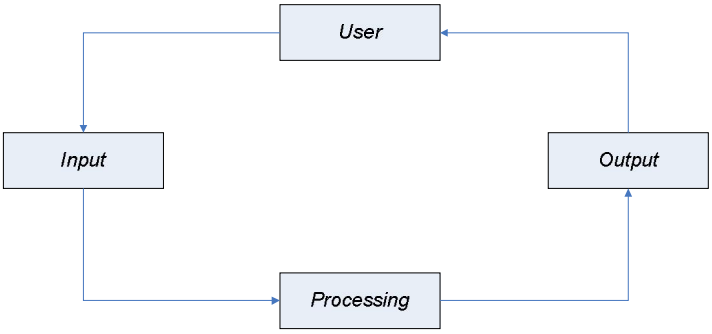
Informasi merupakan hasil dari pengolahan data dan ini merupakan hal terpenting dari sistem informasi.

2.2.1.1 Definisi Informasi

Istilah "data" dan "informasi" sering kali digunakan secara bergantian dan saling tertukar, meskipun kedua istilah ini sebenarnya berbeda. Data merupakan bahasa *mathematical*, dan atau simbol-simbol pengganti lain yang telah disepakati secara umum di dalam (usaha) menggambarkan suatu objek, manusia, peristiwa, aktivitas, konsep, atau objek-objek penting lainnya. Singkatnya, informasi adalah data yang telah ditempatkan pada konteks yang penuh arti oleh penerimanya [John83].



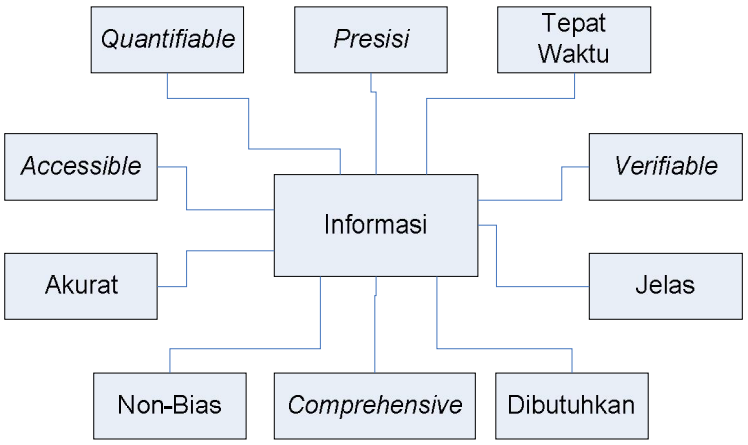
Gambar 2.3 : Relasi (umum) antara data dan informasi



Gambar 2.4 : *Input*, *Processing*, *Output* dan umpan baliknya

2.2.1.2 Atribut Informasi

Banyak atribut atau kualitas-kualitas yang berkaitan dengan konsep informasi dapat membantu (proses) perancangan di dalam (usaha) mengidentifikasi dan mendeskripsikan kebutuhan informasi yang spesifik. Untuk ilustrasinya lihat gambar 2.5.



Gambar 2.5 : Atribut-atribut informasi

Keterangan gambar :

- Akurasi : derajat kebebasan informasi dari kesalahan.
- Presisi : ukuran detail yang digunakan di dalam penyediaan informasi.

- Tepat waktu : penerimaan informasi masih dalam jangkauan waktu yang dibutuhkan oleh si penerima, tidak kadaluarsa atau terlambat.
- Jelas : derajat kebebasan informasi dari keraguan.
- Dibutuhkan : tingkat relevansi informasi yang bersangkutan dengan kebutuhan pengguna.
- *Quantifiable* : tingkat atau kemampuan dalam menyatakan informasi dalam bentuk numerik.
- *Verifiable* : tingkat kesepakatan atau kesamaan nilai sebagai hasil pengujian informasi yang sama oleh berbagai pengguna.
- *Accessible* : tingkat kemudahan dan kecepatan dalam memperoleh informasi yang bersangkutan.
- *Non-bias* : derajat perubahan yang sengaja dibuat untuk mengubah atau memodifikasi informasi dengan tujuan mempengaruhi para penerimanya.
- *Comprehensive* : tingkat kelengkapan informasi.

2.2.2 Konsep Dasar Sistem

Dalam sub-bab ini akan dijelaskan mengenai konsep dasar sistem.

2.2.2.1 Definisi Sistem

Secara umum, *sistem* dapat didefinisikan sebagai sekumpulan objek, ide, berikut saling keterkaitannya di dalam mencapai suatu tujuan atau dengan kata lain, sistem dapat disebutkan sebagai kumpulan komponen (subsistem fisik maupun non-fisik/logika) yang saling berhubungan satu sama lainnya dan bekerja sama secara harmonis untuk mencapai suatu tujuan.

2.2.3 Konsep Dasar Sistem Informasi

Dalam sub-bab ini akan dijelaskan mengenai konsep dasar sistem informasi.

2.2.3.1 Definisi Sistem Informasi

Pada saat ini tentu saja hampir semua organisasi telah memiliki sistem informasi. Sistem informasi ini merupakan sebuah entitas (kesatuan) formal yang terdiri dari berbagai sumber daya fisik maupun logika. Dari organisasi ke organisasi, sumber daya ini disusun atau distrukturkan dengan berbagai cara karena suatu organisasi dan sistem informasi terkait merupakan sumberdaya yang bersifat dinamis.

2.2.3.2 Fungsionalitas Sistem Informasi

Fungsi dari sistem informasi yaitu mengambil, mengolah, menyimpan dan menyampaikan informasi (komunikasi) yang diperlukan di dalam mengoperasikan seluruh aktivitas organisasi yang bersangkutan.

2.2.3.3 Tujuan Sistem Informasi

Tujuan sistem informasi adalah untuk menyediakan dan mensistematikkan informasi yang merefleksikan seluruh kejadian atau kegiatan yang diperlukan untuk mengendalikan operasi suatu organisasi.

2.2.4 Konsep Dasar Sistem Informasi Geografis

Era komputerisasi telah membuka wawasan dan paradigma baru di dalam proses pengambilan keputusan berikut penyebaran informasi terkait. Sehubungan dengan hal ini, data yang merepresentasikan "dunia nyata" dapat disimpan kemudian diproses sedemikian rupa sehingga akhirnya disajikan dalam bentuk-bentuk yang lebih sederhana (bersifat elementer tetapi tetap sesuai dengan kebutuhan).

2.2.4.1 Definisi Sistem Informasi Geografis

Sejak pertengahan 1970-an, telah dikembangkan sistem-sistem yang secara khusus dibuat untuk menangani masalah informasi yang bereferensi geografis dalam berbagai cara dan bentuk. Masalah-masalah ini antara lain mencakup :

a) Pengorganisasian data dan atau informasi
b) Menempatkan informasi pada lokasi (geografis) tertentu.
c) Melakukan komputasi-komputasi, memberikan ilustrasi mengenai keterhubungan antara suatu objek spasial dengan objek-objek spasial yang lain (koneksinya), beserta beberapa analisis spasial terkait.

Sebutan umum bagi sistem-sistem berfungsionalitas seperti ini adalah SIG (Sistem Informasi Geografis). Di dalam beberapa literatur terkadang SIG juga dipandang sebagai hasil dari "perkawinan" antara sistem komputer untuk bidang

Kartografi (CAC) dan atau sistem komputer untuk bidang perancangan (CAD) dengan teknologi pengelolaan basis data.

Pada awalnya, data atau objek-objek geografi hanya disajikan di atas peta dengan menggunakan bentuk simbol, variasi ukuran, pola garis dan kombinasi warna. Kemudian, elemen-elemen geometri ini dideskripsikan di dalam legendanya misalnya, garis hitam tebal untuk jalan utama, garis hitam tipis untuk jalan sekunder dan jalan-jalan yang berikutnya. Lebih dari itu, berbagai *layer* atau data semacam ini juga dapat di-*overlay*-kan berdasarkan kesamaan sistem koordinatnya (planimetris). Oleh karena, sebuah peta dapat menjadi media yang sangat efektif baik sebagai alat presentasi maupun sebagai bank tempat penyimpanan data atau objek geografis.

Walaupun peta sudah dianggap efektif untuk memenuhi beberapa kebutuhan, media seperti ini juga dapat dipandang masih mengandung beberapa "kelemahan" atau keterbatasan. Dalam kaitan produksi peta, data hasil survey atau informasi yang tersimpan akan diproses dan akhirnya dipresentasikan dengan suatu cara tertentu untuk memenuhi tujuan tertentu pula. Oleh karena itu, adalah tidak mudah untuk merubah bentuk presentasi (yang sudah muncul di atas peta-peta jadi). Sebuah peta (*hardcopy)* selalu menyediakan gambar atau simbol unsur-unsur geografi dengan bentuk, pola, ukuran dan warna yang tetap atau statis meskipun oleh para penggunanya diperlukan untuk memenuhi berbagai kebutuhan.

Di dalam konteks yang lebih luas, peta juga merupakan salah satu aset publik yang sangat berharga. Survey-survey pemetaan yang telah dilakukan di berbagai negara telah mengindikasikan bahwa nilai keuntungan atas penggunaan

peta akan meningkat jauh bahkan beberapa kali lipat dari biaya produksi peta itu sendiri. Meskipun demikian, jika dibandingkan dengan fungsionalitas peta-peta seperti ini (dalam bentuk analog atau *hardcopy*), SIG memiliki beberapa keunggulan *inherent* karena penyimpanan data dan presentasinya dipisahkan secara tegas atau dibedakan dengan jelas. Dengan demikian, data (basis data spasial) yang dimiliki oleh SIG dapat dipresentasikan dalam berbagai cara dan bentuk (dinamis).

2.2.4.2 Komponen Sistem Informasi Geografis

SIG merupakan salah satu sistem yang kompleks dan pada umumnya juga terintegrasi dengan lingkungan sistem komputer lainnya di tingkat fungsional dan jaringan (*network).* Jika diuraikan, SIG sebagai sistem terdiri dari beberapa komponen sebagai berikut:

a) **Perangkat keras**. Pada sat ini SIG sudah tersedia bagi berbagai *platform* perangkat keras, mulai dari kelas PC *desktop*, *workstations,* hingga *multi-user host* yang bahkan dapat digunakan oleh banyak orang secara bersamaan dalam jaringan komputer yang luas, tersebar, berkemampuan tinggi, memiliki ruang penyimpanan yang besar dan mempunyai kapasitas memori (RAM) yang besar. Walaupun demikian, fungsionalitas SIG tidak terikat dengan secara ketat pada karakteristik-karakteristik fisik perangkat kerasnya sehingga keterbatasan memori pada suatu PC-pun dapat diatasi. Adapun perangkat keras yang sering digunakan untuk perangkat lunak SIG adalah komputer (PC), *mouse,* monitor (beserta VGA-*card* drafik) yang beresolusi tinggi, *digitizer, printer, plotter, receiver* GPS dan *scanner.*

b) **Perangkat lunak.** Dari sudut pandang yang lain, SIG bisa juga merupakan sistem perangkat lunak yang tersusun secara modular di mana sistem basis datanya memegang peranan kunci. Pada kasus perangkat SIG tertentu, setiap sub-sistem diimplementasikan dengan menggunakan perangkat lunak yang terdiri dari beberapa modul hingga tidak mengherankan jika ada perangkat SIG yang tediri dari ratusan modul program(*.exe) yang masing-masing dapat dieksekusi tersendiri.

c) **Data & informasi geografi.** SIG dapat mengumpulkan dan menyimpan data atau informasi yang diperlukan baik secara tidak langsung (dengan cara meng-*import*-nya dari format-format perangkat lunak SIG yang lain) maupun secara langsung dengan cara melakukan dijitasi data spasialnya dari peta analog dan kemudian memasukkan data atributnya dari tabel-tabel atau laporan dengan menggunakan *keyboard*.

d) **Manajemen.** Suatu proyek SIG akan berhasil jika dikelola dengan baik dan dikerjakan oleh orang-orang memiliki keahlian yang tepat pada semua tingkatan.

2.2.5 Konsep Dasar Geodesi

Objek atau entitas yang memiliki *properties* geometrik, sering kali disebut sebagai unsur-unsur spasial, di dalam SIG perlu memiliki referensi geografis. Untuk memahami lebih lanjut mengenai representasi data spasial beserta referensi terkait perlu dibahas konsep-konsep geodesi yang mendukungnya. Dalam sub-bab ini akan mengulas konsep-konsep geodesi dalam kaitan kontribusinya sebagai alat representasi dan referensi koordinat-koordinat di dalam data spasial.

2.2.5.1 Definisi Geodesi

Menurut definisi klasik dari F.R. Helmert, geodesi adalah "sains pengukuran dan pemetaan permukaan bumi" (dicuplik dari pustaka [1]). Oleh karena itu, sesuai dengan definisi ini, disiplin ilmu geodesi termasuk ke dalam bidang *geosciences*, selain bisa juga dimasukkan ke dalam kelompok *engineering sciencesm* sedangkan menurut sumber pustaka [1], geodesi merupakan salah satu cabang ilmu matematika terpakai (aplikasi atau terapan) yang bermaksud menentukan bentuk dan ukuran bumi, menentukan posisi (koordinat) titik-titik, panjang dan arah garis di permukaan bumi, dan mempelajari medan gaya berat bumi.

2.2.5.2 Sistem Proyeksi Peta

Sebagaimana telah diketahui secara umum, bahwa hasil (utama) suatu proses survey dan pemetaan adalah peta itu sendiri. Peta meupakan suatu representasi konvensional (miniatur) dari unsur-unsur diksi dari (alamiah dan buatan manusia) dari sebagian atau bahkan keseluruhan permukaan bumi di atas media bidang datar dengan dengan skala tertentu. Meskipun demikian, permukaan bumi ini secara keseluruhan merupakan permukaan yang memiliki bentuk melengkung hingga hingga sama sekali tidak memungkinkan untuk dapat dibentangkan menjadi bidang datar sempurna tanpa mengalami perubahan atau kerusakan .

Persyaratan-persyaratan geometrik yang harus dipenuhi oleh suatu peta sehingga menjadi peta yang ideal adalah :

1) Jarak-jarak antara titik-titik yang terletak di atas peta harus sesuai dengan jarak-jarak realitasnya.

2) Luas area suatu unsur yang direpresentasikan di atas peta harus sesuai dengan luas yang sebenarnya.
3) Sudut atau arah suatu garis yang direpresentasikan di atas peta harus sesuai dengan arah yang sebenarnya.
4) Bentuk suatu unsur yang direpresentasikan di atas peta harus sesuai dengan bentuk yang sebenarnya.

2.2.5.3 Jenis Proyeksi Peta

Hingga saat ini, sudah banyak proyeksi peta yang pernah dibuat oleh manusia. Walaupun demikian, proyeksi-proyeksi peta ini dapat dikelompokkan menurut jenis-jenis berikut :

1) Menurut bidang proyeksi yang digunakan :
   a. Proyeksi *azimuthal,* menggunakan bidang datar sebagai bidang proyeksi.
   b. Proyeksi kerucut (*conic*), menggunakan bidang kerucut (dapat didatarkan tanpa mengalami perubahan dan kerusakan) sebagai bidang proyeksi.
   c. Proyeksi selinder (*cylindrical*), menggunakan bidang selinder (dapat didatarkan tanpa mengalami perubahan dan kerusakan) sebagai bidang proyeksi.
2) Menurut kedudukan garis karakteristik atau kedudukan bidang proyeksi proyeksi terhadap bidang datum yang digunakan:
   a. Proyeksi normal, garis karakteristik berimpit dengan sumbu bumi.
   b. Proyeksi miring, garis karakteristik membentuk sudut terhadap sumbu bumi.
   c. Proyeksi transversal atau ekuatorial, garis karakteristik tegak lurus terhadap sumbu bumi.

3) Menurut ciri-ciri asli yang tetap dipertahankan:
   a. Proyeksi ekuidistan (jarak di atas peta sama dengan jarak di permukaan bumi).
   b. Proyeksi konform (sudut dan arah di atas peta sama dengan sudut dan arah di permukaan bumi)
   c. Proyeksi ekuivalen (luas di atas peta) sama dengan luas di permukaan bumi.
4) Menurut karakteristik singgungan antara bidang proyeksi dengan bidang datumnya:
   a. Proyeksi menyinggung
   b. Proyeksi memotong
   c. Proyeksi baik yang tidak menyinggung maupun tidak memotong (hampir tidak pernah ada).

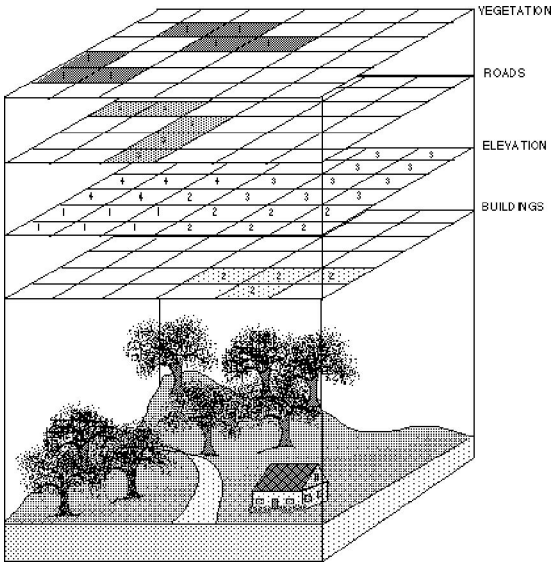
2.2.6 Model Data Spasial SIG

Data spasial yaitu data yang memiliki referensi ruang kebumian (*georeference*) dimana berbagai data atribut terletak dalam berbagai unit spasial. Dalam pemanfaatannya, data spasial semakin meningkat setelah adanya teknologi pemetaan digital dan pemanfaatannya pada Sistem Informasi Geografis (SIG). Format data spasial dapat berupa vektor (*Polygon, line, points*) maupun *raster*. Pada sub-bab ini akan dijelaskan format data spasial tersebut.

2.2.6.1 Model Data *Raster*

Model data raster bertugas untuk menampilkan, menempatkan dan menyimpan *content* data spasial dengan menggunakan struktur (semacam) matriks atau susunan piksel-piksel yang membentuk suatu *grid* (segi-empat). Akurasi

spasial model data ini sangat bergantung pada resolusi spasial atau ukuran pikselnya di permukaan bumi.



Gambar 2.6 Contoh tampilan permukaan bumi & *layer* model data *raster*

Raster memiliki beberapa karakteristik yang dapat membedakannya satu sama lain. Karakteristik ini bisa mencakup sebagai berikut:

a. Resolusi

   Resolusi (spasial) dapat didefinisikan sebagai dimensi linier minimum dari satuan terkecil *geographic space* yang dapat direkam.

b. Orientasi

   Orientasi dibuat untuk merepresentasikan arah utara pada sistem *grid*, yang paling sering dilakukan adalah menginpitkan arah utara *grid* ini dengan arah utara yang sebenarnya di titik asal sistem koordinat *grid* yang bersangkutan.

c. *Zone*

Setiap *zone* "*layer"* peta raster merupakan sekumpulan lokasi-lokasi yang memperlihatkan nilai-nilai (bisa diasumsikan sebagai Id atau nomor pengenal yang direpresentasikan oleh nilai-nilai pikselnya) yang sama. Contoh *zone* yagn dimaksud adalah persil-persil tanah milik, batas-batas administrasi, danau atau pulau, jenis tanah dan vegetasi dan lain sejenisnya.

d. Domain nilai piksel

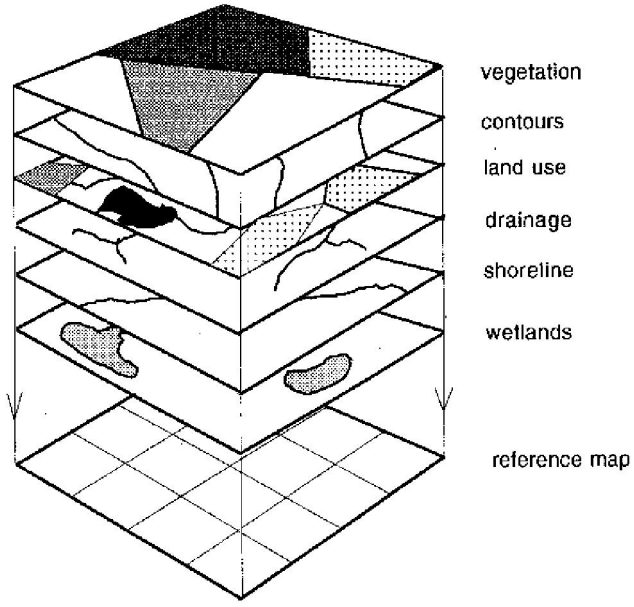
Nilai, dalam konteks data *raster*, adalah *item* informasi (atribut) yang disimpan di dalam sebuah *layer* untuk setiap pikselnya. Piksel-piksel di dalam *zone* atau area yang sejenis memiliki nilai yang sama. Pada umumnya, nilai sebuah piksel sebuah data *raster* dikuantisasikan ke dalam *domain* bilangan bulat dengan panjang 8 *bit* (atau 1 *byte*), kebanyakan raster produk sensor satelit.

e. Koordinat Piksel atau Lokasi Unsur

Pada umumnya, lokasi di dalam model data *raster*, secara langsung dapat diidentifikasikan dengan menggunakan pasangan koordinat lokalnya kolom dan baris (x,y). Meskipun demikian, posisi-posisi koordinat geografis (geodetik) yang sebenarnya (di permukaan bumi) dari beberapa piksel yang terletak di sudut-sudut citra *raster* (atau di titik-titik yang mudah dikenali seperti halnya persimpangan jalan atau anak sungai) juga diketahui (diamati di lapangan atau dihitung) melalui proses pengikatan memerlukan beberapa titik kontrol (GCP – *Ground Control Point*).

2.2.6.2 Model Data Vektor

Model data vektor dapat menampilkan, menempatkan dan menyimpan data spasial dengan menggunakan titik-titik, garis-garis atau kurva, atau poligon beserta atribut-atributnya. 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 merupakan sekumpulan titik-titik terurut yang saling terhubung. Sedangkan luasan atau poligon juga disimpan sebagai sekumpulan dafar titik-titik, tetapi dengan catatan bahwa titik awal dan titik akhir geometri poligon memiliki nilai koordinat yang sama (poligon tertutup sempurna).



Gambar 2.7 Contoh tampilan *layer* dalam model data vektor

2.2.6.3 Area atau Poligon Sederhana

Cara yang paling sederhana dalam merepresentasikan suatu unsur yang berbentuk poligon adalah dengan menggunakan komponen *chain* (*arc*), yaitu merepresentasikan setiap poligon sebagai sekumpulan koordinat (x,y). Kemudian

menguntungkan, karena sifat-sifatnya yang mudah dan sederhana, cara ini masih memiliki beberapa kelemahan sebagai berikut :

a. Garis-garis yang terletak diperbatasan dan membentuk dua poligon yang bersebelahan harus di-*sampling*, didijitasi dan direkam dua kali.
b. Tidak tersedianya informasi mengenai hubungan ketetanggaan.
c. Penggambaran unsur-unsur spasial poligon yang bertingkat seperti halnya unsur "pulau" yang terdapat di dalam unsur "danau" yang juga terdapat di dalam unsur "pulau" yang lebih besar adalah tidak memungkinkan untuk dilakukan kecuali hanya sebagai bangunan grafis sementara.
d. Tidak memiliki cara yang mudah untuk memeriksa apakah topologi batas-batasnya sudah benar, lengkap, tidak lengkap atau tidak dapat diterima dan aneh.

2.2.6.4 Model Data *Spaghety*

Model data vektor yang telah dibahas di muka, dikenal sebagai model data *spaghety*. Pada vektor ini, lembaran pada kertas seolah-olah ditranslasikan garis demi garis kedalam sejumlah koordinat (x,y) dalam format dijital. Sebuah titik dikodekan sebagai pasangan koordinat (x,y) tunggal, sebuah garis dikodekan sebagai *list* atau *string* (*chain* atau *arc*) pasangan-pasangan koordinat (x,y), sementara unsur spasial berbentuk area atau luasan dikodekan sebagai geometri poligon dan direkam sebgai pasangan-pasangan koordinat *close loop* yang mendefinisikan batas-batasnya.

2.3 Metode Perancangan Sistem

Dalam merancang suatu perangkat lunak dibutuhkan alat pembantu dalam memodelkannya, dikarenakan perangkat lunak ini berbasis *Object Oriented*

Programme maka dalam memodelkannya menggunakan UML (*Unified Modeling Language*).

2.3.1 UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk menvisualisasi, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*Object Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem *blue print,* yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database* dan komponen-komponen yang diperlukan dalam sistem *software*.

UML sebagai sebuah bahasa yang memberikan *vocabulary* dan tatanan penulisan kata-kata dalam *'MS Word'* untuk kegunaan komunikasi. Sebuah bahasa model adalah sebuah bahasa yang mempunyai *vocabulary* dan konsep tatanan / aturan penulisan serta secara fisik mempresentasikan dari sebuah sistem. Seperti halnya UML adalah sebuah bahasa standard untuk pengembangan sebuah *software* yang dapat menyampaikan bagaimana membuat dan membentuk model-model, tetapi tidak menyampaikan apa dan kapan model yang seharusnya dibuat yang merupakan salah satu proses implementasi pengembangan *software.*

UML tidak hanya merupakan sebuah bahasa pemograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemograman, seperti JAVA, C++, Visual Basic, atau bahkan dihubungkan secara langsung ke dalam sebuah *object-oriented database*. Begitu juga mengenai pendokumentasian dapat dilakukan seperti; *requirements*, arsitektur, *design, source code, project plan, tests*, dan *prototypes*. Untuk dapat memahami UML membutuhkan bentuk

konsep dari sebuah bahasa model, dan mempelajari 3 (tiga) elemen utama dari UML seperti *building block*, aturan-aturan yang menyatakan bagaimana *building block* diletakkan secara bersamaan, dan beberapa mekanisme umum (*common*).

2.3.2 Diagram UML

Dalam pemodelan UML terdapat beberapa diagram yang masing-masingnya memiliki fungsi atau kegunaan masing-masing.

2.3.2.1 *Use Case Diagram*

Use case class digunakan untuk memodelkan dan menyatakan unit fungsi/layanan yang disediakan oleh sistem (subsistem atau *class*) ke pemakai. *Use case* dapat dilingkupi dengan batasan sistem yang diberi label nama sistem. *Use case* adalah sesuatu yang menyediakan hasil yang dapat diukur ke pemakai atau sistem eksternal.

Use cases adalah interaksi atau dialog antara sistem dan *actor*, termasuk pertukaran pesan dan tindakan yang dilakukan oleh sistem. *Use case* diprakarsai oleh *actor* dan mungkin melibatkan peran *actor* lain. *Use case* harus menyediakan nilai minimal kepada satu *actor*. *Use case* bisa memiliki perluasan yang mendefinisikan tindakan khusus dalam interaksi atau *usecase* lain mungkin disisipkan. *Use case class* memiliki objek *usecase* yang disebut skenario. Skenario menyatakan urutan pesan dan tindakan tunggal.

Komponen pembentuk *usecase diagram* :

1. *Actor*

Pada dasarnya *actor* bukanlah bagian dari *use case diagram*, namun untuk dapat terciptanya suatu *use case diagram* diperlukan beberapa *actor*. *Actor* tersebut mempresentasikan seseorang atau sesuatu (seperti

perangkat, sistem lain) yang berinteraksi dengan sistem. Sebuah *actor*mungkin hanya memberikan informasi masuk pada sistem, hanya menerima informasi dari sistem atau keduanya menerima, dan memberi informasi pada sistem. A*ctor* hanya berinteraksi dengan *use case,* tetapi tidak memiliki kontrol atas *use case*. *Actor* digambarkan dengan *stick man*. *Actor* dapat digambarkan secara secara umum atau spesifik, dimana untuk membedakannya kita dapat menggunakan *relationship.*

2. *Use Case*

Use case adalah gambaran fungsionalitas dari suatu sistem, sehingga *customer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.

Cara menentukan *use case* dalam suatu sistem:

a. Pola perilaku perangkat lunak aplikasi.

b. Gambaran tugas dari sebuah *actor*.

c. Sistem atau "benda" yang memberikan sesuatu yang bernilai kepada *actor*.

d. Apa yang dikerjakan oleh suatu perangkat lunak (*bukan bagaimana cara mengerjakannya).

Relasi dalam u*se case*

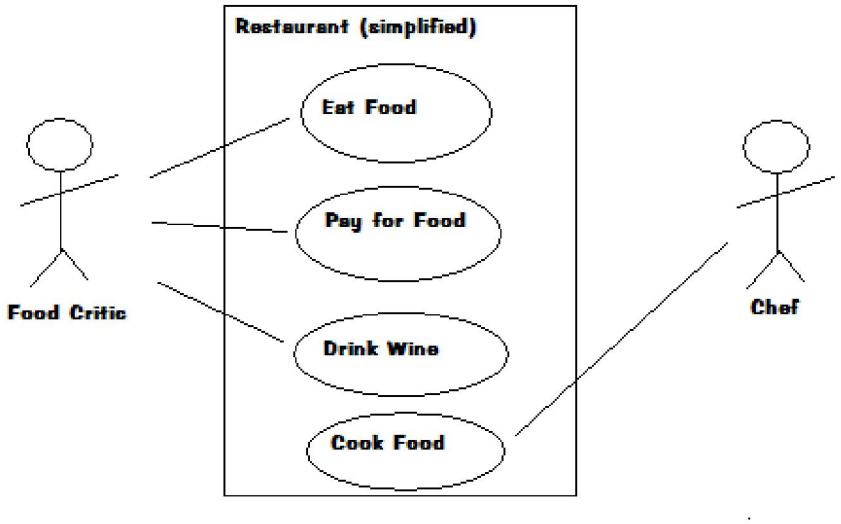
Ada beberapa relasi yang terdapat pada *use case diagram*:

1. *Association*, menghubungkan *link* antar elemen.
2. *Generalization*, disebut juga *inheritance* (pewarisan), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.

3. *Dependency*, sebuah elemen bergantung dalam beberapa cara ke elemen lainnya.
4. *Aggregation*, bentuk *assosiation* dimana sebuah elemen berisi elemen lainnya.

Tipe relasi/ *stereotype* yang mungkin terjadi pada *use case diagram*:

1. <<include>> , yaitu kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya.
2. <<extends>>, kelakuan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan alarm.
3. <<communicates>>, mungkin ditambahkan untuk asosiasi yang menunjukkan asosiasinya adalah *communicates association* . Ini merupakan pilihan selama asosiasi hanya tipe *relationship* yang dibolehkan antara *actor* dan *use case*.



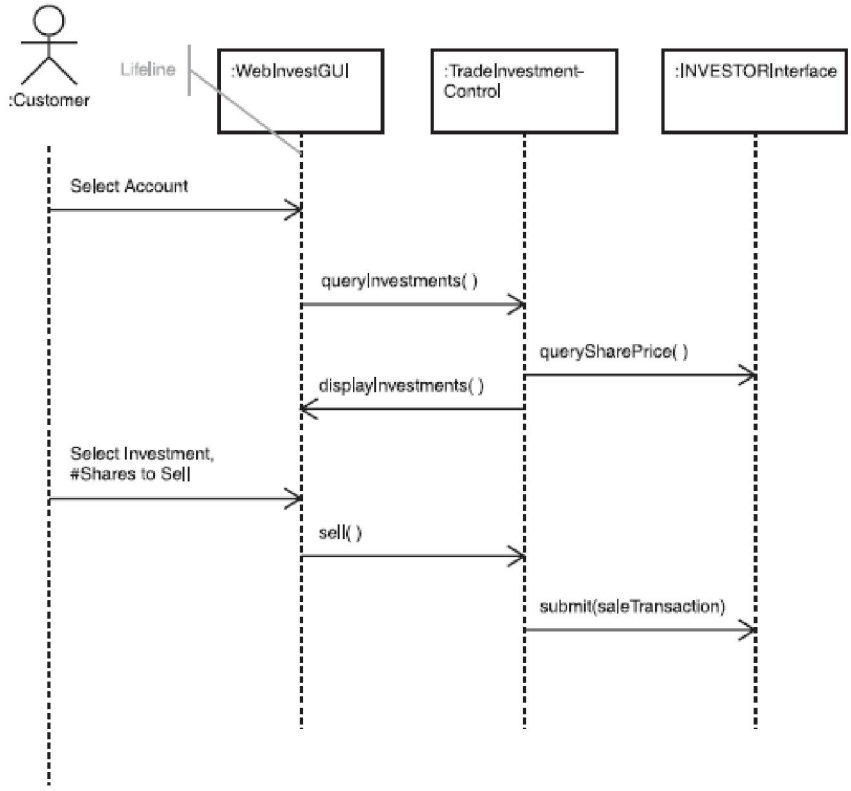
Gambar 2.8 contoh *use case* diagram

2.3.2.2 *Sequence Diagram*

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek-obyek yang melakukan suatu tugas atau aksi tertentu. Obyek-obyek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya ditaruh di paling kiri dari diagram.

Pada diagram ini, dimensi vertikal merepresentasikan waktu. Bagian paling atas dari diagram menjadi titik awal dan waktu berjalan ke bawah sampai dengan bagian dasar dari diagram. Garis Vertical, disebut *lifeline*, dilekatkan pada setiap obyek atau aktor. Kemudian, *lifeline* tersebut digambarkan menjadi kotak ketika obyek melakukan suatu operasi, kotak tersebut disebut *activation box*. Obyek dikatakan mempunyai *live activation* pada saat tersebut.

Pesan yang dipertukarkan antar obyek digambarkan sebagai sebuah anak panah antara *activation box* pengirim dan penerima. Kemudian di atasnya diberikan label pesan. Salah satu contoh *sequence* diagram digambarkan pada gambar 2.11.



Gambar 2.9 Contoh *sequence* diagram

Pada contoh *sequence diagram* di atas digambarkan contoh *use case* investasi perdagangan. Pada diagram tersebut obyek yang berinteraksi adalah *user*, *user interface* sistem, dan *interface* terhadap sistem eksternal.

Pada diagram tersebut terlihat aliran secara umum, yakni :

1. *User* memilih *account investment*.
2. Kemudian, sistem akan mengirimkan pesan pada sistem investor untuk melakukan *query* harga saham dari investasi pada *account user*.
3. Sistem akan menampilkan harga saham pada *account* investasi *user*.

4. *User* memilih investasi dan jumlah saham yang akan dijual.

5. Sistem akan mengirimkan pesan kepada sistem investor untuk menyampaikan permintaan untuk menjual saham yang telah ditentukan oleh *user*.

2.3.2.3 *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *class diagram* menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok :

1. Nama (dan *stereotype*)
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

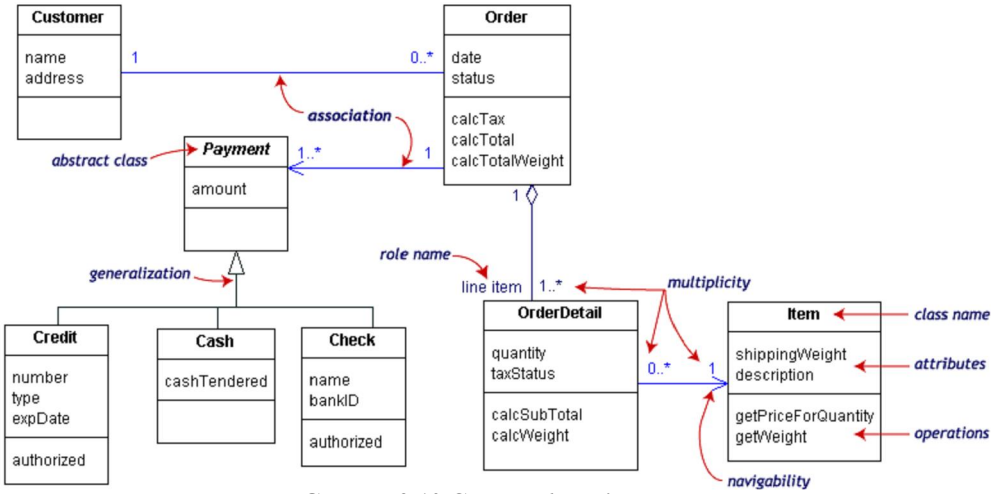
- *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
- *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
- *Public*, dapat dipanggil oleh siapa saja

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class.* Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*.

Sesuai dengan perkembangan *class* model, *class* dapat dikelompokkan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package*.

Hubungan Antar *Class*

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2. Agregasi, yaitu hubungan yang menyatakan bagian ("terdiri atas..").
3. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.



Gambar 2.10 Contoh *class* diagram

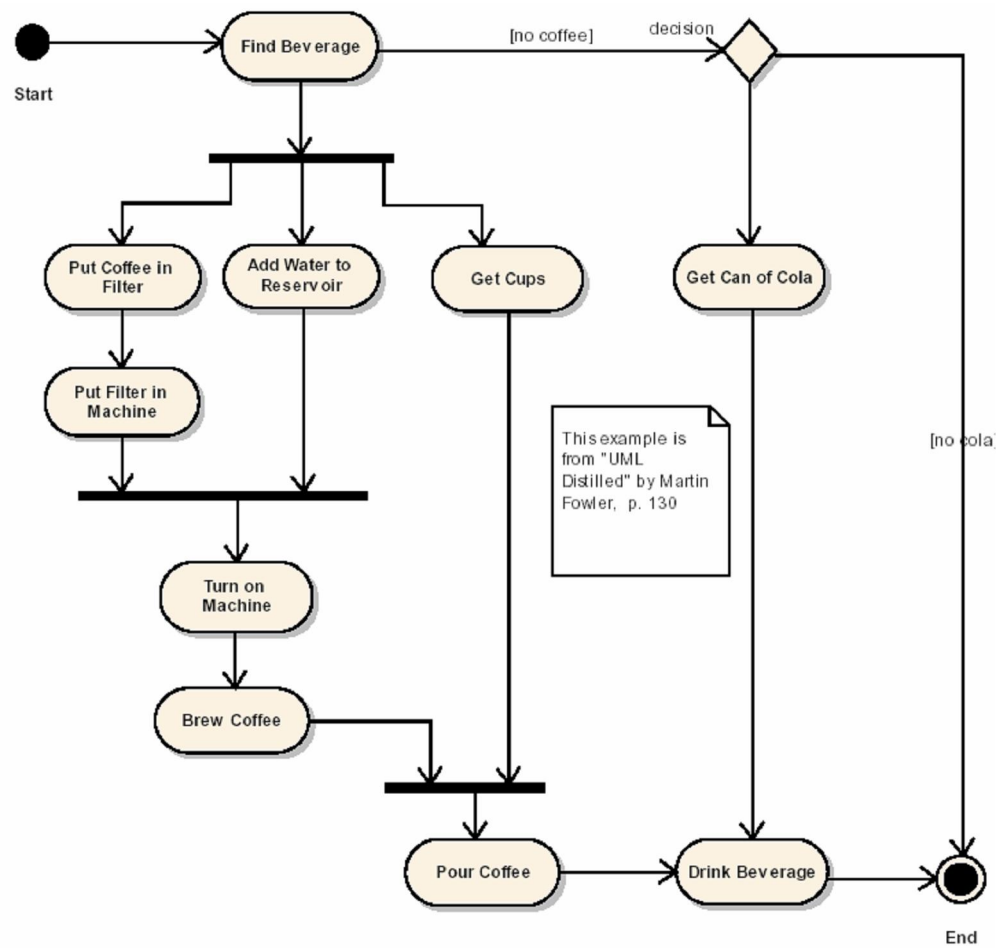
2.3.2.4 *Activity Diagram*

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 2.11 Contoh *activity* diagram

2.3.2.5 *Deployment Diagram*

Deployment diagram menunjukkan susunan fisik sebuah sistem, menunjukkan bagian perangkat lunak yang berjalan pada perangkat keras. Hal utama dalam diagram tersebut adalah pusat-pusat yang dihubungkan oleh jalur

komunikasi. Sebuah pusat adalah sebuah titik yang dapat mengumpulkan beberapa perangkat lunak. Pusat-pusat mempunyai dua bentuk. Sebuah alat adalah perangkat keras, alat ini dapat berupa sebuah komputer atau perangkat keras yang lebih sederhana yang terhubung pada sebuah sistem. Sebuah lingkungan eksekusi merupakan perangkat lunak yang mengumpulkan atau terdiri dari perangkat lunak lain, contohnya adalah sebuah sistem operasi atau sebuah *process* kontainer.