

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Menurut Kusrini (2007:5) kata sistem mempunyai beberapa pengertian, tergantung dari sudut pandang mana kata tersebut didefinisikan. Secara garis besar ada dua kelompok pendekatan, yaitu:

1. Pendekatan sistem yang lebih menekankan pada elemen-elemen atau kelompoknya, yang dalam hal ini sistem itu didefinisikan sebagai “suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu aturan tertentu.
2. Pendekatan sistem sebagai jaringan kerja dari prosedur yang lebih menekankan urutan operasi di dalam sistem. Prosedur didefinisikan oleh Richard F. Neushl sebagai “urutan operasi kerja yang biasanya melibatkan beberapa orang di dalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi bisnis yang terjadi.

Secara sederhana suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu. (Tata Sutabri; 2005:2).

Menurut Prof. Dr. Mr. S. Prajudi Atmosudirdjo, suatu sistem terdiri atas objek-objek atau unsur-unsur atau komponen-komponen yang berkaitan dan berhubungan satu sama lain sedemikian rupa sehingga unsur-unsur tersebut merupakan suatu kesatuan pemrosesan atau pengolahan yang tertentu. (Tata Sutarbi; 2005:9-10).

II.1.1. Karakteristik Sistem.

Model umum sebuah sistem adalah input, proses, dan output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana sebab sebuah sistem dapat mempunyai beberapa masukan dan keluaran. Selain itu, sebuah sistem memiliki karakteristik atau sifat-sifat tertentu. Adapun karakteristik yang dimaksud adalah sebagai berikut:

1. **Komponen Sistem (*Component*)**

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

2. **Batasan Sistem (*Boundary*)**

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya.

3. **Lingkungan Luar Sistem (*Environment*)**

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem.

4. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau interface. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain.

5. Masukan Sistem (*Input*)

Energi yang masuk ke dalam sistem, berupa perawatan dan sinyal. Masukan perawatan adalah energy yang dimasukkan supaya sistem tersebut dapat berinteraksi.

6. Keluaran Sistem (*Output*)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna.

7. Pengolahan Sistem (*Process*)

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem (*Objectivitas*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. (Tata Sutabri; 2005:11-12)

II.1.2. Klasifikasi Sistem.

Sistem dapat diklasifikasikan dari beberapa sudut pandang diantaranya:

a. Sistem Abstrak dan Sistem Fisik.

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak nampak secara fisik. Sistem fisik merupakan sistem yang ada secara fisik.

b. Sistem Alamiah dan Sistem Buatan Manusia.

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia. Sedangkan sistem buatan manusia adalah sistem yang melibatkan interaksi manusia dengan mesin.

c. Sistem Deterministik dan Sistem Probabilistik.

Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem deterministik. Sedangkan sistem yang bersifat probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi.

d. Sistem Terbuka dan Sistem Tertutup.

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sedangkan sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya. (Tata Sutabri; 2005:13)

II.1.3. Daur Hidup Sistem (*System Life Cycle*).

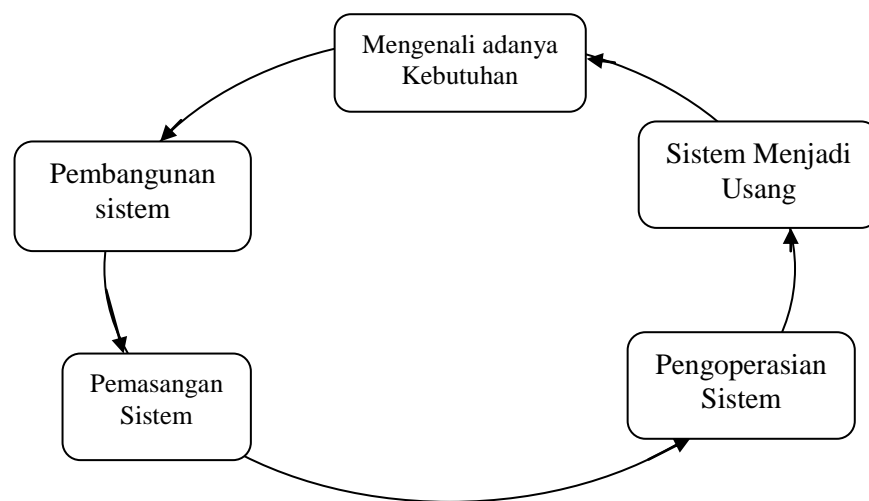
Daur hidup sistem merupakan proses *evolusioner* yang diikuti dalam menerapkan sistem atau subsistem informasi berbasis komputer. Siklus hidup sistem terdiri dari serangkaian tugas yang erat mengikuti langkah-langkah pendekatan sistem karena tugas-tugas tersebut mengikuti pola yang teratur dan dilakukan secara *top down*.

Beberapa tahapan dari daur hidup sistem adalah sebagai berikut:

- a. Mengenali adanya kebutuhan
- b. Pembangunan sistem

- c. Pemasangan sistem
- d. Pengoperasian sistem
- e. Sistem menjadi usang (Tata Sutabri; 2005:14-15)

Untuk menggambarkan daur hidup sistem, dapat dilihat pada gambar dibawah ini:



Gambar II.1 Daur Hidup Sistem

Sumber: (Tata Sutabri; 2005:15)

II.2. Informasi

Informasi adalah data yang sudah diolah menjadi sebuah bentuk yang berarti bagi pengguna, yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi. Data belum memiliki nilai sedangkan informasi sudah memiliki nilai. Informasi dikatakan bernilai bila manfaatnya lebih besar dibanding biaya untuk mendapatkannya. (Kusrini; 2007:7-8)

Informasi adalah data yang telah diklasifikasi atau diolah atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. (Tata Sutabri; 2005:23)

Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya untuk mendapatkan informasi tersebut. Nilai dan kualitas dari informasi sangat dipengaruhi atau oleh beberapa sifat, diantaranya:

1. Relevan (*Relevancy*)
2. Akurat (*Accurancy*)
3. Tepat waktu (*Time lines*)
4. Efisien (*Efficiency*)
5. Ekonomis (*Economy*)
6. Ketersediaan (*Availabilty*)
7. Dapat dipercaya (*reliability*)
8. Konsisten

Informasi akan bernilai sempurna, jika dalam pengambilan keputusannya diperoleh hasil yang optimal dan bukan hasil rata-rata dari masalah yang ada untuk dijadikan hasil optimal guna menghindari masalah yang akan mendatangkan kerugian.

II.3. Sistem Informasi

Defenisi umum sistem informasi adalah sebuah sistem yang terdiri atas rangkaian subsistem informasi terhadap pengolahan data untuk menghasilkan informasi yang berguna dalam pengambilan keputusan. (Kusrini, 2007:9)

Sistem informasi didefinisikan oleh Robert A. Laitch dan K. Roscoe Bavis sebagai berikut “Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan”. (Kusrini, 2007:8)

Dalam suatu sistem informasi terdapat komponen-komponen sebagai berikut :

1. Perangkat keras (*hardware*), mencakup berbagai peranti fisik seperti komputer dan printer.
2. Perangkat lunak (*software*) atau program, yaitu sekumpulan instruksi yang memungkinkan perangkat keras memproses data.
3. Prosedur, yaitu sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.
4. Orang, yaitu semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan dan penggunaan keluaran sistem informasi.
5. Basis data (*database*), yaitu sekumpulan table, hubungan dan lain-lain yang berkaitan dengan penyimpanan data.
6. Jaringan komputer dan komunikasi data, yaitu sistem penghubung yang memungkinkan sumber (*resources*) dipakai secara bersama atau diakses oleh sejumlah pemakai.

II.4. Sistem Informasi Geografis

SIG (*Geographic Information System*) Puskesmas merupakan suatu sistem (berbasis komputer) yang digunakan untuk menyimpan dan memanipulasi informasi-inforamsi geografis. SIG dirancang untuk mengumpulkan, menyimpan, dan menganalisis objek-objek dan fenomena-fenomena dimana lokasi geografis merupakan karakteristik yang penting atau kritis untuk dianalisis. Dengan demikian, SIG merupakan sistem komputer yang memiliki empat kemampuan berikut dalam menangani data yang bereferensi geografis:

- a. Masukan
- b. Keluaran
- c. Manajemen data (penyimpanan dan pemanggilan data)
- d. Analisis dan manipulasi data (Eddy Prahasta; 2009:1)

Sistem Informasi Geografis (SIG) adalah sebuah sistem atau teknologi berbasis komputer yang dibangun dengan tujuan untuk mengumpulkan, menyimpan, mengolah, dan menganalisa, serta menyajikan data-data dan informasi dari suatu objek atau fenomena yang berkaitan dengan letak atau keberadaannya di permukaan bumi.

Suatu komponen yang terdiri dari perangkat keras, perangkat lunak, data geografis dan sumberdaya manusia yang bekerja bersama secara efektif untuk memasukan, menyimpan, memperbaiki, memperbaharui, mengelola, memanipulasi, mengintegrasikan, menganalisa dan menampilkan data dalam suatu informasi berbasis geografis.

Pada dasarnya SIG dapat dirinci menjadi beberapa sub sistem yang saling berkaitan mencakup input data, manajemen data, pemrosesan atau analisis data, pelaporan (*output*), dan hasil analisa. (Ekadinata A; 2008: 2)

SIG mempunyai kemampuan untuk menghubungkan berbagai data pada suatu titik tertentu di bumi, menggabungkannya, menganalisa dan akhirnya memetakan hasilnya. Data yang akan diolah pada SIG merupakan *data spasial* yaitu sebuah data yang berorientasi geografis dan merupakan lokasi yang memiliki sistem koordinat tertentu, sebagai dasar referensinya. Sehingga aplikasi SIG dapat menjawab beberapa pertanyaan seperti; lokasi, kondisi, trend, pola dan pemodelan. Kemampuan inilah yang membedakan SIG dari sistem informasi lainnya.

II.4.1. Komponen-Komponen SIG

Komponen-komponen yang membangun SIG adalah perangkat lunak, perangkat keras, data, pengguna, dan aplikasi. SIG dalam pengelolaan sumber daya alam di lingkungan pemerintah lokal, sebagai contoh, memerlukan sistem yang mendukung yang mendukung tersedianya kelima komponen tersebut, sebagaimana di ilustrasikan oleh Gambar II.2 (Ekadinata A; 2008: 3).



Gambar II.2. Komponen Sistem Informasi Geografis

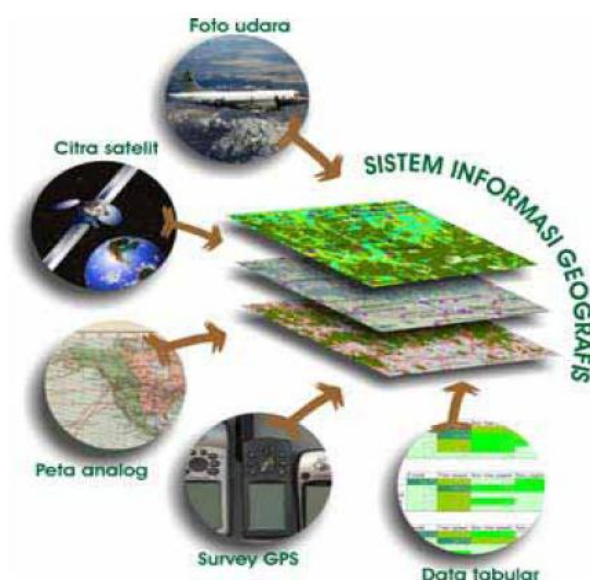
Sumber: (Ekadinata A; 2008:3)

II.4.2. Jenis dan Sumber Data Geografis

Data geografis pada dasarnya tersusun oleh dua komponen penting yaitu data spasial dan data atribut. Data spasial merepresentasikan posisi atau lokasi geografis dari suatu objek di permukaan bumi. Sedangkan data atribut memberikan deskripsi atau penjelasan suatu objek. Data atribut dapat berupa informasi numerik, foto, narasi, dan lain sebagainya, yang diperoleh dari data statistik, pengukuran lapangan dan sensus, dan lain-lain.

Data spasial dapat diperoleh dari berbagai sumber dalam berbagai format. Sumber data spasial antara lain mencakup: data grafis peta analog, foto udara, citra satelit, survey lapangan, pengukuran theodolit, pengukuran dengan

menggunakan *global positioning systems (GPS)* dan lain-lain. Adapun format data spasial, secara umum dapat dikategorikan dalam format digital dan format analog.



Gambar II.3: Sumber data dalam SIG

Sumber: (Ekadinata A; 2008:3)

II.5. Database

Database atau sering juga disebut *basis data* adalah sekumpulan informasi yang disimpan dalam komputer secara sistematis dan merupakan sumber informasi yang dapat diperiksa menggunakan suatu program komputer. Database berfungsi untuk menyimpan informasi atau data. Untuk mengelola database diperlukan *software* yang sering disebut dengan *DBMS (Database Management System)*. Dengan *DBMS* pengguna atau *user* dapat membuat, mengelola, mengontrol dan mengakses database dengan mudah, praktis dan efisien.

Database terdiri dari tabel yang didalamnya terdapat *field-field*, dan sebuah database bisa terdiri dari beberapa tabel. Dalam pembuatan database, Anda perhatikan hal-hal berikut :

- a. Setiap tabel dalam database, harus memiliki *field* (kolom) yang unik yang disebut *primary key*.
- b. Tabel dalam database tidak boleh ada *redudancy data* yaitu mengandung *record* ganda. Jika terdapat data yang sama, maka perlu dilihat kembali rancangan tabelnya.
- c. Pilih tipe data yang tepat, sehingga ukuran database seminimal mungkin.
(Madcoms; 2011 : 12).

II.6. MySQL

MySQL adalah salah satu program yang dapat digunakan sebagai *database*, dan merupakan salah satu software untuk database *server* yang banyak digunakan. *MySQL* bersifat *Open Source* dan menggunakan *SQL*. *MySQL* bisa dijalankan diberbagai *platform* misalnya : *Windows*, *Linux* dan lain sebagainya.

MySQL memiliki beberapa kelebihan, antara lain :

- a. *MySQL* dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah.
- b. *MySQL* memiliki kecepatan yang bagus dalam menangani *query* sederhana.
- c. *MySQL* memiliki operator dan fungsi secara penuh dan mendukung perintah *Select* dan *Where* dalam perintah *query*.

- d. *MySQL* memiliki keamanan yang bagus karena beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses user dengan system perijinan yang mendetail serta sandi terenkripsi.
- e. *MySQL* mampu menangani basis data dalam skala besar.
- f. *MySQL* dapat mendeteksi pesan kesalahan pada client dengan menggunakan lebih dari dua puluh bahasa. (Madcoms Madiun; 2011:140).

II.7. PHP (PHP Hypertext Preprocessor)

PHP (*PHP Hypertext Preprocessor*) adalah bahasa pemrograman yang berjalan dalam sebuah webserver dan berfungsi sebagai pengolah data pada sebuah server. Dengan menggunakan program *PHP*, sebuah website akan lebih interaktif dan dinamis. (Madcoms Madiun; 2011: 11)

PHP adalah script pemrograman yang terletak dan dieksekusi di server. Salah satunya adalah untuk menerima, mengolah, dan menampilkan data dari dan ke sebuah situs (Madcoms Madiun; 2011: 81).

II.7.1. Menulis Script program PHP

Suatu *script* akan dikenali sebagai *script PHP* bila diapit oleh tanda berikut ini:

<??>

Atau

<?php

?>

Perintah - perintah *PHP* dapat disisipkan dibagian manapun dalam *HTML*. (Madcoms; 2011: 86).

II.7.2. Membuat Variabel di PHP

Variabel adalah tempat untuk menyimpan data. Data yang disimpan sewaktu-waktu bisa saja dipanggil atau diganti dengan data lain. Mendefinisikan *variable* cukup dengan sebuah kata (akan menjadi nama *variable*) yang diawali dengan tanda \$, kemudian isikan nilai *variable* tersebut. Contoh:

\$bulan="Mei"; atau \$nomor= 2011;

Penulisan *variable* ada aturannya, yaitu: gunakan kata apa saja yang didahului oleh huruf dan selalu diberi tanda \$ di depan kata tersebut.

Contoh : \$bulan (Madcoms Madiun; 2011: 88).

II.8. Google Maps

Google Maps adalah layanan gratis yang diberikan oleh Google dan sangat populer. Google Maps adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah. Dengan kata lain, Google Maps merupakan suatu peta yang dapat dilihat dengan menggunakan suatu browser. Kita dapat menambahkan fitur Google Maps dalam web yang telah kita buat atau pada blog kita yang berbayar maupun gratis sekalipun dengan Google Maps API. Google Maps API adalah suatu library yang berbentuk JavaScript.

Cara membuat Google Maps untuk ditampilkan pada suatu web atau blog sangat mudah hanya dengan membutuhkan pengetahuan mengenai HTML serta JavaScript, serta koneksi Internet yang sangat stabil. Dengan menggunakan

Google Maps API, kita dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga kita dapat fokus hanya pada data-data yang akan ditampilkan. Dengan kata lain, kita hanya membuat suatu data sedangkan peta yang akan ditampilkan adalah milik Google sehingga kita tidak dipusingkan dengan membuat peta suatu lokasi, bahkan dunia. Dalam pembuatan program Google Map API menggunakan urutan sebagai berikut:

1. Memasukkan Maps API JavaScript ke dalam HTML kita.
2. Membuat element div dengan nama map_canvas untuk menampilkan peta.
3. Membuat beberapa objek literal untuk menyimpan property-property pada peta.
4. Menuliskan fungsi JavaScript untuk membuat objek peta.
5. Meng-inisiasi peta dalam tag body HTML dengan event onload.

Pada Google Maps API terdapat 4 jenis pilihan model peta yang disediakan oleh Google, diantaranya adalah:

1. ROADMAP, ini yang saya pilih, untuk menampilkan peta biasa 2 dimensi
2. SATELLITE, untuk menampilkan foto satelit
3. TERRAIN, untuk menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, contohnya akan menunjukkan gunung dan sungai
4. HYBRID, akan menunjukkan foto satelit yang di atasnya tergambar pula apa yang tampil pada ROADMAP (jalan dan nama kota). (Eddy Prahasta; 2006 :1)

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

UML adalah singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. Sebagai bahasa, berarti UML memiliki sintaks dan semantik. (Widodo Prabowo, 2011:6)

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk:

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

II.9.1 Diagram-Diagram UML

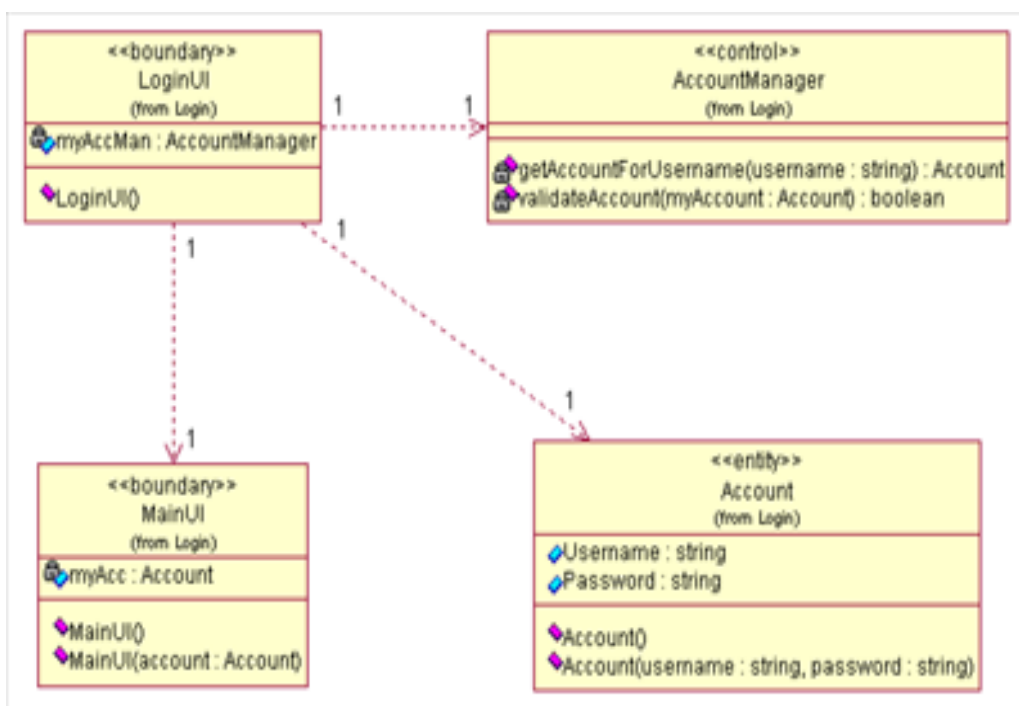
Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek. (Haviluddin; 2011:1)

Beberapa literature menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain:

1. Diagram Kelas (*Class diagram*)

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class diagram* membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. *Class* memiliki tiga area pokok:

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

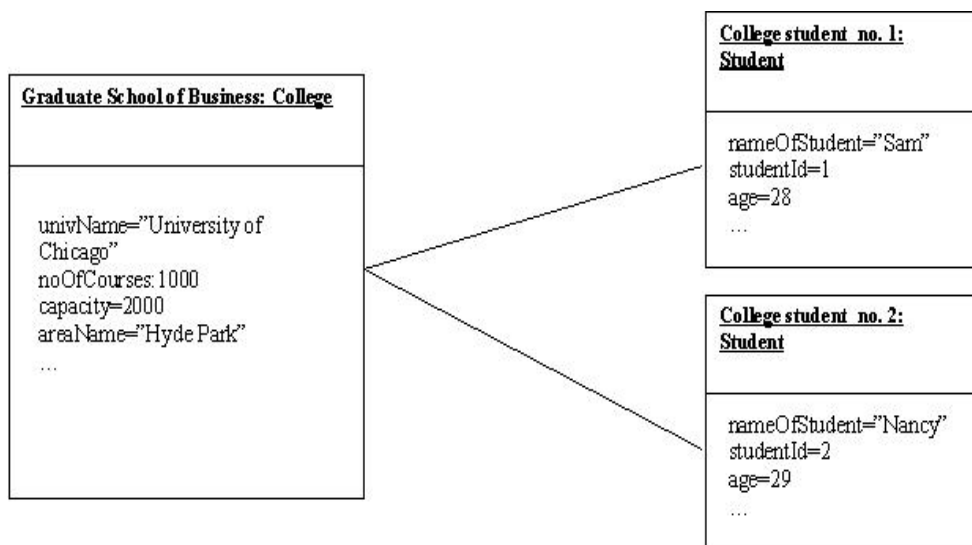


Gambar II.4: Notasi *Class Diagram*

Sumber: (Haviluddin; 2011:3)

2. Diagram Objek (*Object Diagram*)

Object diagram menggambarkan kejelasan kelas dan warisan dan kadang-kadang diambil ketika merencanakan kelas, atau untuk membantu pemangku kepentingan non-program yang mungkin menemukan diagram kelas terlalu abstrak. Berikut notasi *object diagram*.

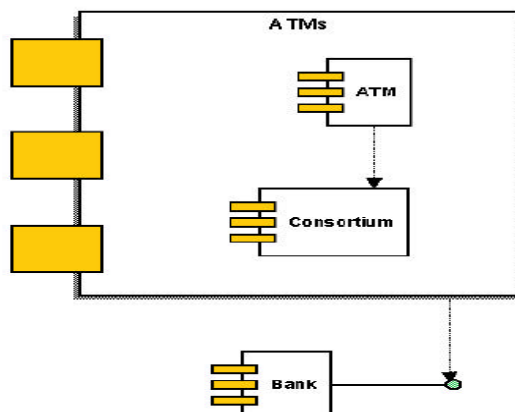


Gambar II.5: Notasi *object diagram*

Sumber: (Haviluddin; 2011:3)

3. Diagram Komponen (*Component diagram*)

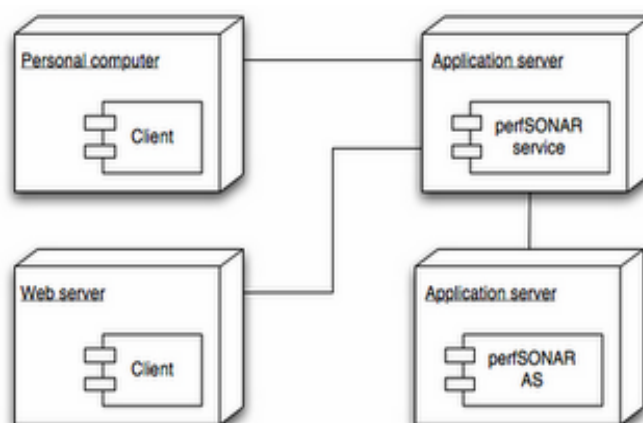
Component diagram menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini dilaksanakan.



Gambar II.6: Notasi *component diagram*
Sumber: (Haviluddin; 2011:3)

4. Deployment Diagram

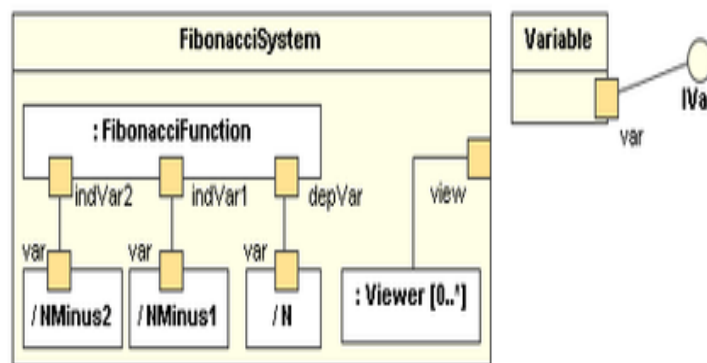
Deployment diagram memberikan gambaran dari arsitektur fisik perangkat lunak, perangkat keras, dan artefak dari sistem. *Deployment* diagram dapat dianggap sebagai ujung spektrum dari kasus penggunaan, menggambarkan bentuk fisik dari sistem yang bertentangan dengan gambar konseptual dari pengguna dan perangkat berinteraksi dengan sistem.



Gambar II.7: Notasi *deployment diagram*
Sumber: (Haviluddin; 2011:4)

5. Composite Structure Diagram

Sebuah diagram struktur komposit mirip dengan diagram kelas, tetapi menggambarkan bagian individu, bukan seluruh kelas. Kita dapat menambahkan konektor untuk menghubungkan dua atau lebih bagian dalam atau ketergantungan hubungan asosiasi.

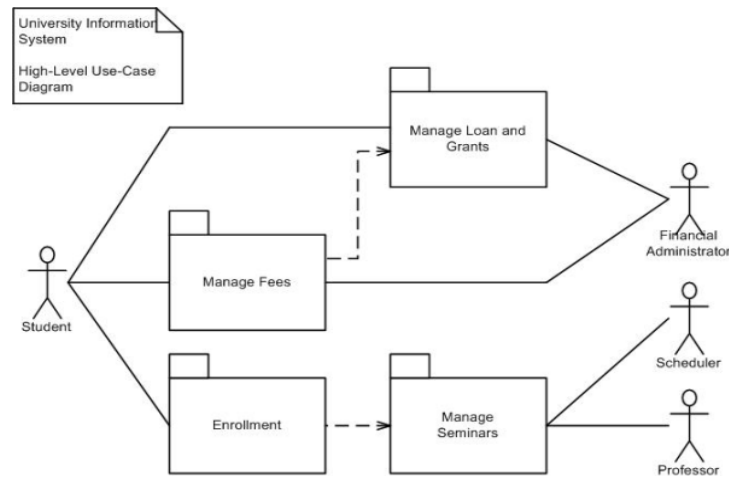


Gambar II.8: Notasi *composite diagram*

Sumber: (Haviluddin; 2011:4)

6. Diagram Paket (*Package diagram*)

Paket diagram biasanya digunakan untuk menggambarkan tingkat organisasi yang tinggi dari suatu proyek *software*. Atau dengan kata lain untuk menghasilkan diagram ketergantungan paket untuk setiap paket dalam Pohon Model.



Gambar II.9: Notasi *package diagram*

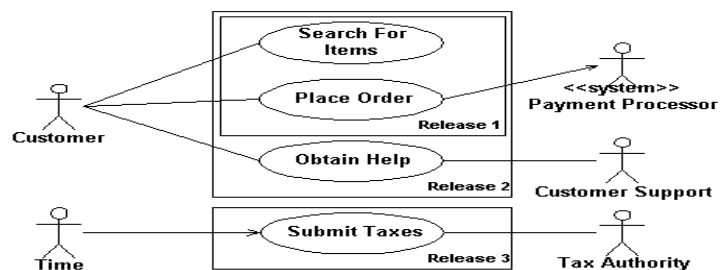
Sumber: (Haviluddin; 2011:4)

7. Use Case Diagram

Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case*.

Use Case memiliki dua istilah:

1. *System use case*; interaksi dengan sistem.
2. *Business use case*; interaksi bisnis dengan konsumen atau kejadian nyata.

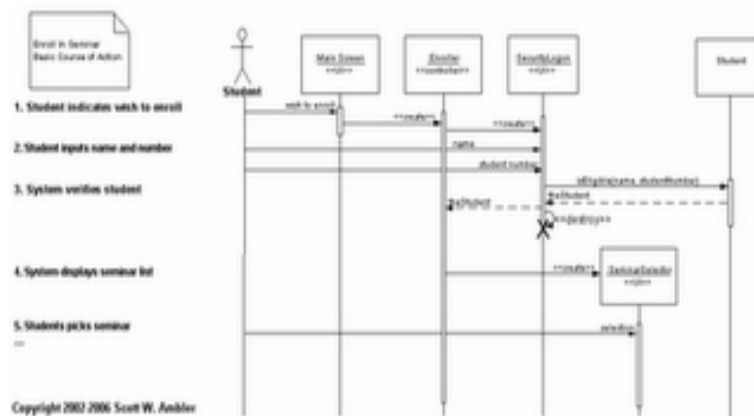


Gambar II.10: Notasi *use case diagram*

Sumber: (Haviluddin; 2011:4)

8. Sequence Diagram

Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence* diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case* diagram.

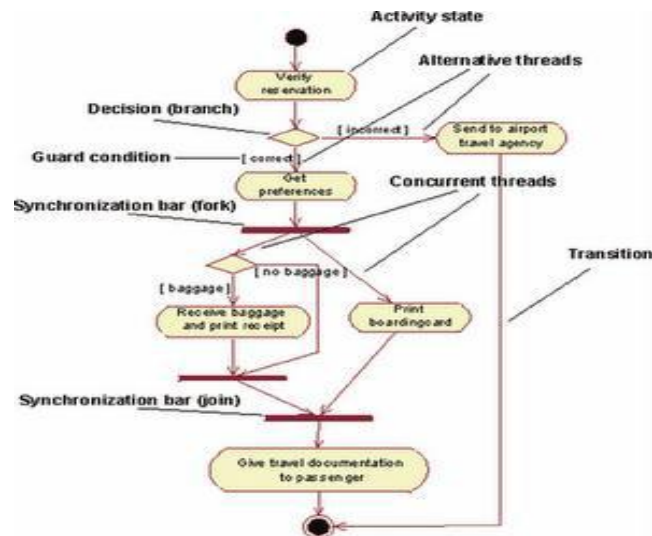


Gambar II.11 Notasi *sequence* diagram

Sumber: (Haviluddin; 2011:5)

9. Activity Diagram

Menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas.

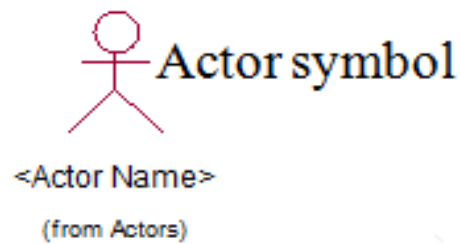


Gambar II.12: Notasi *activity* diagram

Sumber: (Haviluddin; 2011:1)

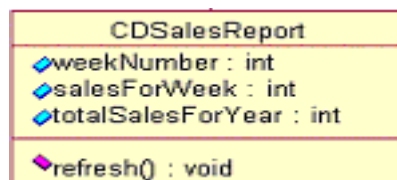
Untuk menggambarkan analisa dan desain diagram, UML memiliki seperangkat notasi yang akan digunakan ke dalam tiga kategori diatas yaitu struktur diagram, *behaviour* diagram dan *interaction* diagram. Berikut beberapa notasi dalam UML diantaranya :

1. *Actor*; menentukan peran yang dimainkan oleh *user* atau sistem lain yang berinteraksi dengan subjek. *Actor* adalah segala sesuatu yang berinteraksi langsung dengan sistem aplikasi komputer, seperti orang, benda atau lainnya. Tugas *actor* adalah memberikan informasi kepada sistem dan dapat memerintahkan sistem untuk melakukan sesuatu tugas.



Gambar II.13: Notasi *actor*
Sumber: (Haviluddin; 2011:6)

2. *Class diagram*; Notasi utama dan yang paling mendasar pada diagram UML adalah notasi untuk mempresentasikan suatu *class* beserta dengan atribut dan operasinya. *Class* adalah pembentuk utama dari sistem berorientasi objek.



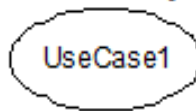
Gambar II.14: Notasi *class*
Sumber: (Haviluddin; 2011:6)

3. *Use Case* dan *use case specification*; *Use case* adalah deskripsi fungsi dari sebuah sistem perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem

dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut skenario.

Use case merupakan awal yang sangat baik untuk setiap fase pengembangan berbasis objek, design, testing, dan dokumentasi yang menggambarkan kebutuhan sistem dari sudut pandang di luar sistem.

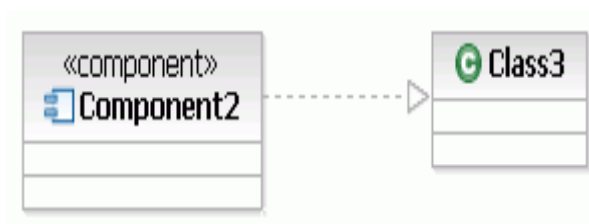
Use-case symbol



Gambar II.15: Notasi *use case*

Sumber: (Haviluddin; 2011:6)

4. *Realization*; *Realization* menunjukkan hubungan bahwa elemen yang ada di bagian tanpa panah akan merealisasikan apa yang dinyatakan oleh elemen yang ada di bagian dengan panah.



Gambar II.16: Notasi *realization*

Sumber: (Haviluddin; 2011:6)

II.9.2. Keuntungan dan Kelemahan UML

a. Keuntungan

Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.

Taylor (1992) menyatakan bahwa membangun *software* menggunakan pendekatan teknologi objek memberikan beberapa keuntungan, antara lain: memungkinkan penggunaan kembali objek yang ada (*reusable*), memungkinkan *software* yang baru dengan konstruksi yang lebih besar, *software* berorientasi objek secara umum lebih mudah dimodifikasi dan dirawat karena sebuah objek dapat dimodifikasi tanpa banyak berpengaruh pada objek yang lain.

b. Kelemahan

UML dipandang masih mempunyai kekurangan terutama dalam meng-*generate* kode program secara komplit. Hal ini karena kurangnya cara memodelkan aspek kelakuan *internal* perangkat lunak untuk dipetakan ke dalam kode program.

Seperti yang kita ketahui, diagram UML yang dapat menghasilkan kode hanyalah *diagram class*, namun itupun hanya sebatas kerangka kodenya saja dan tidak bisa meng-*generate* badan program-nya.

II.10. Dreamweaver

Dreamweaver adalah sebuah HTML (*Hypertext Markup Language*) editor professional untuk mendesain web secara visual dan mengelola situs atau halaman web. *Dreamweaver* merupakan software utama yang digunakan oleh web desainer

maupun *web programmer* dalam mengembangkan suatu situs web, Karena *Dreamweaver* mempunyai ruang kerja, fasilitas dan kemampuan yang mampu meningkatkan produktivitas dan efektifitas dalam desain maupun membangun suatu situs web. (Madcoms Madiun; 2011: 13).