

BAB II

TINJAUAN PUSTAKA

II.1 Pengertian Komputer

Pengertian Komputer Dalam bahasa Inggris disebut *to compute* yang berarti orang yang sedang menghitung, sedangkan dalam bahasa Latin disebut dengan *computare* yang berarti menghitung. Komputer seperti yang telah kita ketahui merupakan sebuah alat elektronik yang mampu memiliki banyak fungsi dan mampu melakukan banyak tugas. Sebelumnya pada masa Perang Dunia ke-2 komputer adalah suatu mesin mekanis yang berfungsi untuk melakukan perhitungan operasi aritmatika. Selain itu komputer dapat didefinisikan sebagai sekumpulan alat elektronik yang saling terkoordinasi satu sama lain sehingga dapat menerima data, kemudian mengolah data, dan pada akhirnya akan menghasilkan suatu keluaran yang berupa informasi (Input > Proses > Output).

Menurut *Oxford English Dictionary* (OED2) edisi ke-2 “Computer sebagai suatu perangkat yang digunakan untuk menghitung atau mengendalikan operasi-operasi yang dinyatakan dalam bentuk numerik atau logika”. Menurut Sanders 1985 “Computer adalah sistem elektronik untuk memanipulasi data yang cepat dan tepat serta dirancang dan diorganisasikan agar secara otomatis menerima dan menyimpan data input, memprosesnya dan menghasilkan output berdasarkan instruksi-instruksi yang telah tersimpan dalam memori”. Komputer pada saat ini merupakan suatu kebutuhan bagi setiap orang, komputer menjadi suatu bagian yang penting bagi setiap kehidupan manusia karena lewat komputer kita bisa

melakukan berbagai aktivitas pekerjaan sehingga menjadi lebih ringan dan untuk mempersingkat waktu pekerjaan tersebut. Komputer tidak hanya digunakan oleh orang dewasa saja, namun anak-anak pun juga bisa menggunakannya, mereka bisa belajar sambil bermain. Komputer terdiri atas 2 bagian besar yaitu perangkat lunak (software) dan perangkat keras (hardware) yg merupakan komponen-komponen suatu komputer.

II.1.1 Perangkat Keras (*hardware*)

Perangkat keras komputer (computer hardware) adalah komponen-komponen fisik yang membentuk satu kesatuan sistem Personal Computer (PC).

Adapun pengelompokan hardware agar lebih mudah sesuai dengan fungsinya pada komputer adalah sebagai berikut :

1. Input Device

Input device merupakan perangkat komputer yang memiliki fungsi sebagai input atau masukan, baik itu perintah maupun koneksi kedalam sistem komputer. Ada beberapa perangkat yang masuk kategori input device diantaranya keyboard, mouse, touchpad, touch screen, light pen, joy stick, games paddle, barcode, digitizer, scanner, sensor dan voice recognizer.

2. Central Processing Unit (CPU)

Central Processing Unit (CPU) atau yang lebih dikenal dengan nama Prosesor adalah perangkat keras komputer yang sangat penting dalam memahami dan melaksanakan perintah dari perangkat lunak.

3. Output Device

Output device merupakan perangkat komputer yang memberikan output/keluaran, baik berupa tampilan visual, suara, maupun tampilan cetak, dan lain sebagainya. Adapun yang termasuk output device adalah *Monitor/ Lcd Monitor, Speaker, Printer.*

II.1.2 Perangkat lunak (*Software*)

Perangkat lunak atau piranti lunak adalah program komputer yang berfungsi sebagai sarana interaksi antara pengguna dan perangkat keras. Perangkat lunak dapat juga dikatakan sebagai penerjemah perintah-perintah yang dijalankan pengguna komputer untuk diteruskan ke atau diproses oleh perangkat keras. Sistem software berkomunikasi dengan I/O dan mengatur penggunaan memori secara efisien. Sistem Software dapat dibedakan menjadi :

1. Sistem Operasi (*Operating System*)

Sistem Operasi merupakan perangkat lunak yang bertugas mengontrol dan melakukan manajemen perangkat keras / Hardware serta operasi-operasi dasar sistem, termasuk menjalankan perangkat lunak aplikasi. Komputer tidak dapat menjalankan perangkat lunak aplikasi jika tidak memiliki sistem operasi di dalamnya. contohnya : Windows, Mac Os, BeOS, DOS, UNIX, Novell Netware yang menjalankan sistem operasi network berbasis DR-DOS, FreeBSE (SO turunan UNIX), atau Windows dengan aplikasi Citrix Metaframe.

2. Bahasa Pemrograman

Bahasa pemrograman merupakan instruksi dan aturan yang tertuang dalam bentuk kode-kode yang diberikan pada komputer untuk melaksanakan suatu tugas. Manusia memberikan instruksi yang dimengerti oleh komputer melalui bahasa pemrograman, dan komputer akan menerjemahkannya.

II.1.3 Perangkat Pemikir (*Brainware*)

Brainware adalah manusia yang terlibat dalam mengoperasikan serta mengatur sistem di dalam komputer. Diartikan juga sebagai perangkat intelektual yang mengoperasikan dan mengeksplorasi kemampuan dari Hardware maupun Software. Manusia merupakan suatu elemen dari sistem komputer. Manusia adalah yang merancang bagaimana suatu mesin dapat bekerja sesuai dengan hasil yang diinginkannya. Di dalam sistem informasi adalah aspek manusia yang menangani proses komputerisasi .

II.2 Pengertian Sistem

Kata Sistem awalnya berasal dari bahasa Yunani (*sustēma*) dan bahasa Latin (*systema*). Dalam definisi yang paling umum, sebuah sistem adalah sekumpulan objek/benda yang memiliki hubungan diantara mereka. Biar lebih jelas, mari kita lihat contoh berikut ini. Menurut Fat pengertian sistem adalah “Sistem adalah suatu himpunan suatu “benda” nyata atau abstrak (a set of thing) yang terdiri dari bagian–bagian atau komponen-komponen yang saling berkaitan, berhubungan, berketergantungan, saling mendukung, yang secara keseluruhan

bersatu dalam satu kesatuan (Unity) untuk mencapai tujuan tertentu secara efisien dan efektif". *Pengertian Sistem Menurut Jogianto (2005: 2)* mengemukakan bahwa sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. sistem ini menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata adalah suatu objek nyata, seperti tempat, benda, dan orang-orang yang betul-betul ada dan terjadi. Adapun Karakteristik Sistem Jogianto (2005: 3) mengemukakan sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu :

1. Komponen

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu mempengaruhi proses sistem secara keseluruhan.

2. Batasan sistem

Batasan sistem (boundary) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. batasan suatu sistem menunjukkan ruang lingkup dari sistem tersebut.

3. Lingkungan Luar Sistem.

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

tersebut. Lingkungan luar yang menguntungkan berupa energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung Sistem

Penghubung (interfance) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

5. Masukan Sistem

Masukan (input) sistem adalah energi yang masukan kedalam sistem. masukan dapat berupa masukan perawatan (maintenance input), dan masukan sinyal (signal input).

6. Keluaran Sistem

Keluaran (output) sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi.

8. Sasaran Sistem

Sebuah sistem sudah tentu mempunyai sasaran ataupun tujuan. Dengan adanya sasaran sistem, maka kita dapat menentukan masukan yang dibutuhkan sistem dan keluaran apa yang akan dihasilkan sistem tersebut dapat dikatakan berhasil apabila mencapai/mengenai sasaran atau pun tujuan.

II.2.1 Pengambilan Keputusan

Keputusan adalah hasil pemecahan masalah yang dihadapinya dengan tegas. Hal ini berkaitan dengan jawaban atas pertanyaan-pertanyaan mengenai ‘apa yang harus dilakukan dan seterusnya mengenai unsur-unsur perencanaan. Dapat juga dikatakan bahwa keputusan itu sesungguhnya merupakan hasil proses pemikiran yang berupa pemilihan satu diantara beberapa alternatif yang dapat digunakan untuk memecahkan masalah yang dihadapinya.

Menurut Siagian pengambilan keputusan adalah “suatu pendekatan terhadap hakikat suatu masalah, pengumpulan fakta-fakta dan data, penentuan yang matang dari alternatif yang dihadapi dan pengambilan tindakan yang menurut perhitungan merupakan tindakan yang paling tepat”.

Maka dapat ditarik kesimpulan bahwa keputusan itu diambil dengan sengaja, tidak secara kebetulan, dan tidak boleh sembarangan. Masalahnya terlebih dahulu harus diketahui dan dirumuskan dengan jelas, sedangkan pemecahannya harus didasarkan pemilihan alternatif terbaik dari alternatif yang ada.

II.2.2 Pengenalan Fuzzy Topsis

Teori *fuzzy* pertama kali dibangun dengan menganut prinsip teori himpunan. Dalam himpunan konvensional (*crisp*), elemen dari semesta adalah anggota atau bukan anggota dari himpunan. Dengan demikian, keanggotaan dari himpunan adalah tetap. Perbedaan mendasar dari himpunan *crisp* dan *fuzzy* adalah bahwa himpunan *crisp* selalu memiliki fungsi keanggotaan yang unik, sedangkan setiap himpunan *fuzzy* memiliki nilai keanggotaan yang terbatas dari fungsi keanggotaan yang mewakilinya. Hal itu memungkinkan *fuzzy* dapat diatur secara maksimum dalam situasi yang diberikan. *Fuzzy* membagi data-data ke dalam derajat keanggotaan, yaitu sesuatu yang dapat menjadi sebagian benar dan sebagian salah dalam waktu yang bersamaan.

Salah satu metode yang digunakan untuk menangani permasalahan, adalah Technique for Order Performance by Similarity to Ideal Solution (TOPSIS). TOPSIS adalah salah satu metode pengambilan keputusan multikriteria yang pertama kali diperkenalkan oleh Yoon dan Hwang (1981). TOPSIS menggunakan prinsip bahwa alternatif yang terpilih harus mempunyai jarak terdekat dari solusi ideal positif dan jarak terpanjang (terjauh) dari solusi ideal negatif dari sudut pandang geometris dengan menggunakan jarak Euclidean (jarak antara dua titik) untuk menentukan kedekatan relative dari suatu alternative dengan solusi optimal. Solusi ideal positif didefinisikan sebagai jumlah dari seluruh nilai terbaik yang dapat dicapai untuk setiap atribut, sedangkan solusi negatif-ideal terdiri dari seluruh nilai terburuk yang dicapai untuk setiap atribut. TOPSIS mempertimbangkan keduanya, jarak terhadap solusi ideal positif dan jarak

terhadap solusi ideal negative dengan mengambil kedekatan relative terhadap solusi ideal positif.

Berdasarkan perbandingan terhadap jarak relatifnya, susunan prioritas alternative bias dicapai. Metode ini banyak digunakan untuk menyelesaikan pengambilan keputusan secara praktis. Hal ini disebabkan konsepnya sederhana dan mudah dipahami, komputasinya efisien, dan memiliki kemampuan mengukur kinerja relative dari alternative – alternative keputusan.

II.2.3 Algoritma Topsis

Adapun langkah-langkah algoritma dari Topsis ini adalah sebagai berikut :

1. Rangking Tiap Alternatif

Topsis membutuhkan ranking kinerja setiap alternatif A_i pada setiap kriteria C_j yang ternormalisasi yaitu :

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \quad \text{dengan } i=1,2,\dots,m; \text{ dan } j=1,2,\dots,n;$$

2. Matriks keputusan ternormalisasi terbobot

$$y_{ij} = w_j r_{ij} \quad \text{dengan } i=1,2,\dots,m \text{ dan } j=1,2,\dots,n$$

3. Solusi Ideal Positif Dan Negatif

Solusi ideal positif A^+ dan solusi ideal negatif A^- dapat ditentukan berdasarkan ranking bobot ternormalisasi (y_{ij}) sebagai berikut :

$$A^+ = (y_1^+, y_2^+, \dots, y_n^+); \text{ dimana :}$$

$$A^- = (y_1^-, y_2^-, \dots, y_n^-);$$

y_j^+ adalah : - max y_{ij} , jika j adalah atribut keuntungan
- min y_{ij} , jika j adalah atribut biaya
 y_j^- adalah : - min y_{ij} , jika j adalah atribut keuntungan
- max y_{ij} , jika j adalah atribut biaya

4. Jarak Dengan Solusi Ideal

Jarak adalah alternatif A_i dengan solusi ideal positif dirumuskan sebagai :

$$D_i^+ = \sqrt{\sum_{j=1}^n (y_i^+ - y_{ij}^+)^2} ; i=1,2,\dots,m$$

Jarak adalah alternatif A_i dengan solusi ideal negatif dirumuskan sebagai :

$$D_i^- = \sqrt{\sum_{j=1}^n (y_{ij}^- - y_i^-)^2} ; i=1,2,\dots,m$$

5. Nilai Preferensi Untuk Setiap Alternatif

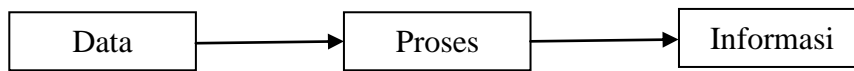
Nilai preferensi untuk setiap alternatif (V_i) diberikan sebagai :

$$V_i = \frac{D_i^-}{D_i^- + D_i^+} \quad i=1,2,\dots,m$$

Nilai V_i yang lebih besar menunjukkan bahwa alternatif A_i lebih dipilih.

II.3 Siklus Pengolahan Data

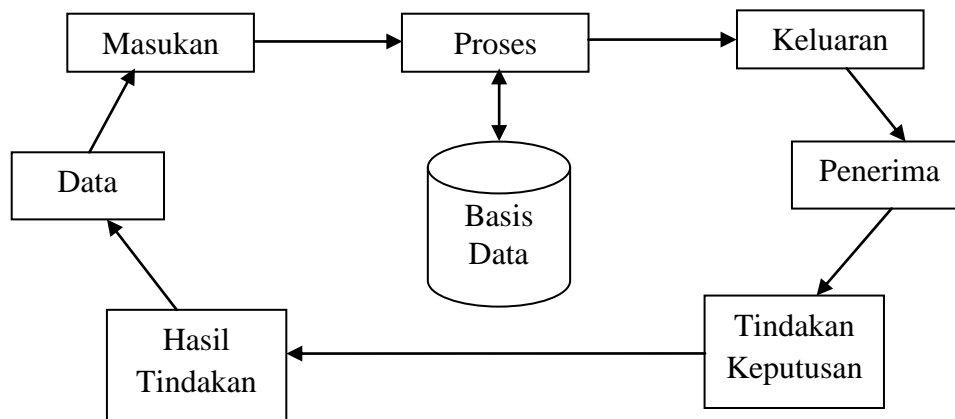
Suatu proses pengolahan data terdiri dari tiga tahapan dasar yang disebut dengan siklus pengolahan data (Data Processing Cycle) yaitu input, processing and output.



Gambar II.1 Siklus Pengolahan Data

Sumber : Abdul Kadir (2003:31)

Berikut ini gambar siklus informasi yang menggambarkan pengolahan data menjadi informasi dan pemakaian informasi untuk mengambil keputusan.



Gambar II.2 Siklus Informasi

Sumber : Abdul Kadir (2003:32)

II.4 Pengenalan Microsoft Visual Basic 6.0

Microsoft Visual Basic 6.0 (VB6) merupakan salah tu aplikasi pemograman visual yang dibuat oleh Microsft. Visual Basic 6.0 berjalan dalam sistem operasi windows dan tergabung dalam suite aplikasi Microsoft Visual Studio 6.0 yang dikkeluar pada akhir 1998. Setelah itu muncul versi-versi lanjutan dari Visual Basic, yaitu Visual Basic 3,4,5 dan 6. Pada Visual Basic 4, dukungan terhadap aplikasi 32 bit mulai diberikan.

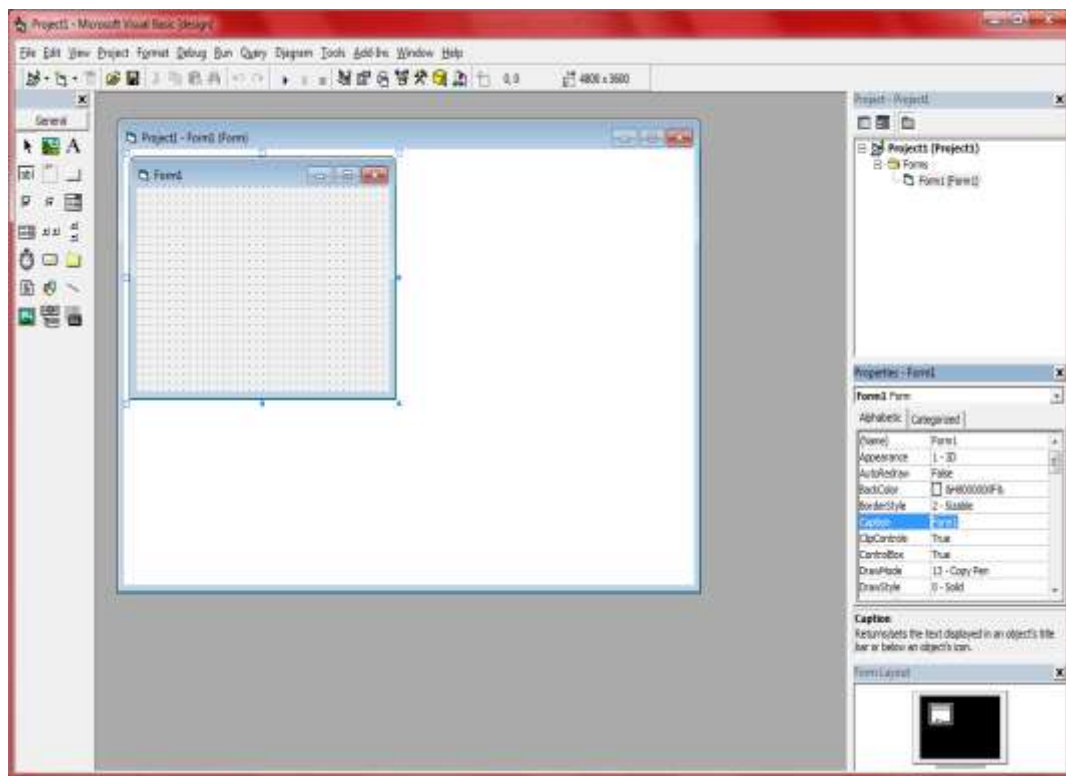
Visual Basic 6.0 tiga buah edisi yaitu :

- a. Standar Edition , merupakan produk dasar
- b. Profesional Edition, berisi tambahan Microsoft Jet Data Access Engine dan pembuatan server OLE Automation.
- c. Enterprise Edition, merupakan edisi untuk membuat program aplikasi client-server.

Visual Basic menyediakan berbagai macam perangkat yang dapat digunakan untuk membuat program aplikasi baik aplikasi kecil dan sederhana untuk keperluan sendiri, hingga aplikasi untuk sistem enterprise yang besar dan rumit, atau bahkan aplikasi yang dijalankan melalui internet.

Visual Basic 6.0 memanfaatkan pendekatan visual/GUI (General User Interface) dalam proses penggunaannya. Dengan pendekatan GUI, proses pembuatan aplikasi menjadi lebih dan nyaman.

Basis bahasa pemrograman yang digunakan dalam VB6 adalah bahasa BASIC (Beginner All-Purpose Symbolic Instruction Code). Bahasa BASIC merupakan salah satu bahasa pemrograman tingkat tinggi yang sederhana dan mudah untuk dipelajari. VB6 memiliki keunggulan dalam hal pengaksesan terhadap beberapa pustaka yang dimiliki oleh sistem operasi Windows. Para program dapat memanfaatkan Windows API (Application Programming Interface) untuk membuat program aplikasi yang lebih kompleks dan powerful.



Gambar II.3 Antarmuka Visual Basic 6.0

Sumber : Arif ramadhan (2041:9)

Ruang kerja Visual Basic 6.0 memiliki komponen-komponen utama sebagai berikut :

a. Title Bar

Title Bar menunjukkan nama project. Project adalah Program aplikasi yang sedang dibuat.

b. Menu Bar

Menu Bar adalah menu-menu utama yang dimiliki visual basic 6.0. Menu-menu tersebut antara lain menu File, Edit, View, Project, Run, dan lain sebagainya.

c. Form

Form adalah tempat yang digunakan untuk merancang aplikasi yang sedang dibuat.

d. Project Explorer Windows

Project Explorer Windows adalah jendela untuk menampilkan project-project, form-form atau modul-modul yang terlibat ddalam proses pembuatan aplikasi.

e. Properties Windows

Properties Windows adalah jendela untuk menampilkan dan mengubah properti yang dimiliki sebuah objek.

f. Toolbox

Toolbox terdiri atas beberapa class objek yang dapat digunakan dalam proses pembuatan aplikasi.

g. Form Layout Window

Form Layout Window adalah jendela yang menunjukkan posisi relatif form terhadap layar monitor.

II.5 Pengenalan SQL Server

SQL Server adalah salah satu program yang dapat digunakan sebagai database dan merupakan salah satu software untuk databse server yang banyak digunakan. SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. SQL Server merupakan sebuah aplikasi

pengolahan data yang berdasarkan pada bahasa Structured Query Language (SQL). Bahasa tersebut merupakan bahasa standar yang digunakan untuk mengolah, menyimpan, memanipulasi data dari sebuah database pada server. SQL Server bersifat open source dan menggunakan SQL. SQL Server bisa dijalankan diberbagai platform dan dapat berjalan secara stabil pada berbagai sitem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Agima, dan lain sebagainya. SQL Server memiliki operator dan fungsi secara penuh dan mendukung perintah *select* dan *where* dalam perintah *query*. SQL Server mampu menanganai basis data dalam skala besar dalam jumlah record lebih dari 50 juta dan 60 ribu tabel serta krang lebih 5 milyar baris dan batas indeks yang dapat ditampung mencapai 32 indeks disetiap tabelnya. SQL Server dapat melakukan koneksi dengan client menggunakan *protocol TCP/IP*, *Unix socket (UNIX)*, atau *named Pipes (NT)*. SQL Server didistribusikan secara open source di bawah lisensi GPL sehingga dapat digunakan secara gratis.



Gambar II.6 Tampilan Ms.SQL Server 2008

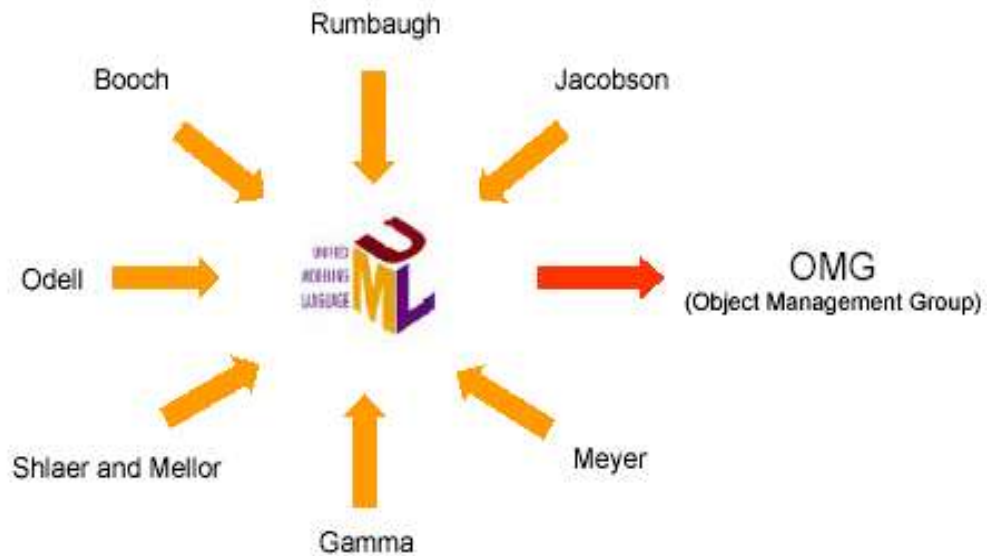
Sumber : Dianne Siebold (2005:5)

II.6 Pengenalan UML (Unified Modeling Language)

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML (Unified Modeling Language) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (Object-Oriented).

UML adalah salah satu tool atau model untuk merancang pengembangan software yang berbasis object oriented. UML sebagai sebuah bahasa yang memberikan vocabulary dan tataana 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.

UML tidak hanya merupakan sebuah bahasa pemrograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemrograman, 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).



Gambar II.7 UML (Unified Modeling Language)

Sumber : Julius Hermawan (2005:7)

II.6.1 Notasi UML (Unified Modeling Language)

Ada 10 jenis notasi yang dimiliki oleh UML (Unified Modeling Language) yaitu :

1. Actor



Gambar II.8 Notasi Actor

Sumber : Julius Hermawan (2005:3)

Actor menggambarkan segala pengguna software aplikasi (user). Actor memberikan suatu gambaran jelas tentang apa yang harus dikerjakan

software aplikasi. Sebagai contoh sebuah actor dapat memberikan input kedalam dan menerima informasi dari software aplikasi.

2. Use Case

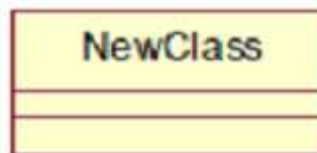


Gambar II.9 Notasi Use Case

Sumber : Julius Hermawan (2005:3)

Use case menjelaskan urutan kegiatan yang dilakukan actor dan sistem untuk mencapai suatu tujuan tertentu.

3. Class



Gambar II.10 Notasi Class

Sumber : Julius Hermawan (2005:3)

Class merupakan pembentuk utama dari sistem berorientasi obyek, karena class menunjukkan kumpulan obyek yang memiliki atribut dan operasi yang sama. Class digunakan untuk mengimplementasikan interface. Class digunakan untuk mengabstraksikan elemen-elemen dari sistem yang sedang dibangun.

4. Interface



Gambar II.11 Notasi Interface

Sumber : Julius Hermawan (2005:3)

Interface merupakan kumpulan operasi tanpa implementasi dari suatu class. Implementasi operasi dalam interface dijabarkan oleh operasi didalam class. Oleh karena itu keberadaan interface selalu disertai oleh class yang mengimplementasikan operasinya. Interface ini merupakan salah satu cara mewujudkan prinsip enkapsulasi dalam obyek.

5. Interaction



Gambar II.12 Notasi Interaction

Sumber : Julius Hermawan (2005:3)

Interaction digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek. Biasanya interaction ini dilengkapi juga dengan teks bernama operation signature yang tersusun dari nama operasi, parameter yang dikirim dan tipe parameter yang dikembalikan.

6. Note



Gambar II.13 Notasi Note

Sumber : Julius Hermawan (2005:3)

Note digunakan untuk memberikan keterangan atau komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model. Note ini bisa disertakan ke semua elemen notasi yang lain.

7. Dependency



Gambar II.14 Notasi Dependency

Sumber : Julius Hermawan (2005:3)

Dependency merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain. Elemen yang ada di bagian tanda panah adalah elemen yang tergantung pada elemen yang ada dibagian tanpa tanda panah.

8. Association

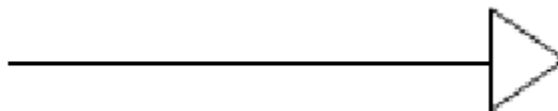


Gambar II.15 Notasi Association

Sumber : Julius Hermawan (2005:3)

Association menggambarkan navigasi antar class (navigation), berapa banyak obyek lain yang bisa berhubungan dengan satu obyek (multiplicity antar class) dan apakah suatu class menjadi bagian dari class lainnya (aggregation). Navigation dilambangkan dengan penambahan tanda panah di akhir garis. Bidirectional navigation menunjukkan bahwa dengan mengetahui salah satu class bisa didapatkan informasi dari class lainnya.

9. Generalization

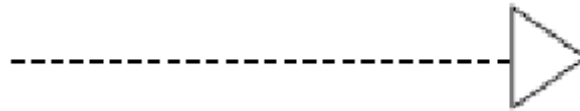


Gambar II.16 Notasi Generalization

Sumber : Julius Hermawan (2005:3)

Generalization menunjukkan hubungan antara elemen yang lebih umum ke elemen yang lebih spesifik. Dengan generalization, class yang lebih spesifik (subclass) akan menurunkan atribut dan operasi dari class yang lebih umum (superclass) atau subclass is superclass . Dengan menggunakan notasi generalization ini, konsep inheritance dari prinsip hirarki dapat dimodelkan.

10. Realization



Gambar II.17 Notasi Realization

Sumber : Julius Hermawan (2005:3)

Realization menunjukkan hubungan bahwa elemen yang ada di bagian tanpa panah akan merealisasikan apa yang dinyatakan oleh elemen yang ada di bagian dengan panah. Misalnya class merealisasikan package, component merealisasikan class atau interface.