BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

2.1.1 Pengertian Sistem

Dalam bukunya Sistem adalah sekelompok elemen-elemen yang saling terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. Raymond McLeod (2004: 9).

2.1.2 Elemen-elemen Sistem

Sebuah sistem tidak memiliki kombinasi elemen yang sama tetapi memiliki susunan dasar yang sama yang terdiri dari (*Input, Output*, Transformasi, Mekanisme Pengendalian dan Tujuan). Raymond McLeod (2004: 9). Sebagaimana yang dijelaskan berikut :

Sumber daya *input* diubah menjadi sumber daya *output*. Sumber daya mengalir dari elemen *input*, melalui elemen transformasi ke elemen output. Suatu mekanisme pengendalian memantau proses transformasi untuk meyakinkan bahwa sistem tersebut memenuhi tujuannya. Mekanisme pengendalian ini dihubungkan pada arus sumber daya dengan memakai suatu lingkaran umpan balik (*feedback loop*) yang mendapatkan informasi dari *output* sistem dan menyediakan informasi bagi mekanisme pengendalian. Mekanisme pengendalian membandingkan sinyal-sinyal umpan balik dengan tujuan dan mengarahkan sinyal pada elemen *input* jika sistem operasi memang perlu diubah.

2.1.3 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu mempunyai komponen-komponen, batas sistem, lingkungan luar sistem, penghubung, masukan, keluaran, pengolah, dan sasaran atau tujuan.

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi yang artinya saling bekerjasama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap subsistem mempunyai karakteristik dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batas Sistem (*Boundary*)

Batas Sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini dapat dipandang sebagai satu kesatuan dan menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar dari suatu sistem adalah apapun diluar batas sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan juga merugikan. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus

ditahan dan dikendalikan, jika tidak maka akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung Sistem (*Interface*)

Merupakan suatu media yang menghubungkan antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya. Keluaran dari satu subsistem akan menjadi masukan untuk subsistem lainnya melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

5. Masukan Sistem (*Input*)

Masukan (input) adalah energi yang dimasukkan ke dalam sistem yang berupa masukan perawatan (*maintenance input*) agar sistem tersebut dapat beroperasi dan masukan sinyal yang diproses untuk menghasilkan keluaran. Sebagai contoh di dalan sistem komputer, program adalah maintenance input yang digunakan untuk mengoperasikan komputer dan data adalah signal input untuk diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

Merupakan hasil dari energi yang telah diproses dan dapat diklasifikasikan antara lain menjadi output yang langsung digunakan, *output* yang bisa dijadikan input untuk subsistem yang lain atau kepada supra sistem serta output yang bagi sistem tidak digunakan lagi (sisa pembuangan) tapi bagian sistem lain sangat berguna.

7. Pengolah Sistem (*Process*)

Suatu sistem dapat mempunyai bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan memproses masukan menjadi keluaran. Sistem akuntansi akan mengolah data-data transaksi menjadi laporan-laporan keuangan dan laporan-laporan yang dibutuhkan oleh manajemen.

8. Sasaran dan Tujuan

Suatu sistem dikatakan berhasil jika mengenai sasaran atau tujuannya, kalau sistem tidak mempunyai sasaran maka sistem tidak akan ada. Sasaran sangat berpengaruh pada masukan dan keluaran yang dihasilkan.

2.1.4 Klasifikasi Sistem

Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lainnya. Karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi yang ada didalam sistem tersebut. Oleh karena itu sistem dapat diklasifikasikan kedalam beberapa sudut pandang. Seperti yang akan dijelaskan sebagai berikut:

a. Sistem diklasifikasikan sebagai sistem alamiah dan sistem buatan manusia. Sistem alamiah adalah sistem yang terjadi karena proses alam tidak dibuat oleh manusia (ditentukan dan tunduk kepada kehendak sang pencipta alam). Misalnya sistem perputaran bumi. Sedangkan sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan

mesin disebut dengan human-machine system atau ada yang menyebut dengan *machine system*.

b. Sistem diklasifkasikan sebagai sistem tertutup dan sistem terbuka. Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak berpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan pihak luarnya. Sistem terbuka adalah sistem yang berhubungan dan tepengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lain.

2.2 Konsep Dasar Informasi

2.2.1 Pengertian Informasi

Menurut Gorgon. B. Davis (1985) dalam buku Al-bahra (2005:8) mendefinisikan informasi sebagai data yang telah diolah menjadi bentuk yang lebih berarti dan berguna bagi penerimanya untuk mengambil keputusan masa kini maupun yang akan datang.

Dalam buku yang sama, Al-Bahra (2005:9), Raymond McLeod (1995) mendefinisikan informasi sebagai data yang telah diolah menjadi bentuk yang lebih berarti bagi penerimanya.

Sumber informasi adalah data. Data adalah kenyataan yang menggambarkan kejadian-kejadian dan kesatuan nyata. Kejadian adalah sesuatu yang terjadi pada saat tertentu. Informasi diperoleh setelah data-data mentah diproses atau diolah. Agar informasi yang dihasilkan menjadi

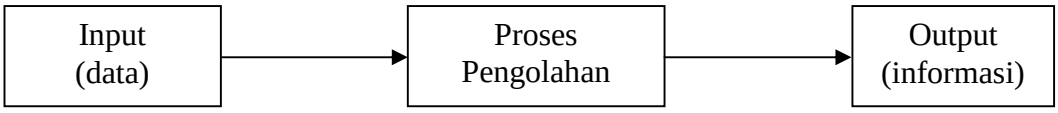
lebih berharga, maka harus memenuhi kriteria sebagai berikut. Al-Bahra (2005:9)

1. Informasi harus akurat, seningga mendukung pihak manajemen dalam mengambil keputusan.
2. Informasi harus relevan, benar-benar terasa manfaatnya bagi yang membutuhkan.
3. Informasi harus tepat waktu, sehingga tidak ada keterlambatan pada saat dibutuhkan.

Kegunaan Informasi adalah untuk mengurangi ketidakpastian didalam proses pengambilan keputusan tentang suatu keadaan. Informasi yang digunakan didalam suatu sistem informasi memiliki nilai manfaat yang lebih efektif.

2.2.2 Siklus Informasi

Untuk memperoleh informasi yang bermanfaat bagi penerimanya, maka perlu dijelaskan bagaimana siklus yang terjadi atau dibutuhkan dalam menghasilkan informasi. Siklus informasi atau siklus pengolahan data adalah sebagai berikut. Al-Bahra (2005:11).



Gambar 2.1 Siklus Informasi

[Sumber : Al-bahra bin Ladjamudin. 2005. *Analisis dan Desain Sistem Informasi.* Graha Ilmu. Yogyakarta.]

2.2.3 Kualitas Informasi

Kualitas informasi sangat dipengaruhi atau ditentukan oleh hal-hal sebagai berikut. Al-Bahra (2005:11)

a. Relevan (*relevancy*), yaitu sejauh mana tingkat relevansi informasi tersebut terhadap kenyataan kejadian masa lalu, masa sekarang dan kejadian yang akan datang.
b. Akurat (*accuracy*), yaitu suatu informasi dikatakan berkualitas jika seluruh kebutuhan informasi telah tersampaikan serta pesan yang disampaikan sudah lengkap sesuai dengan yang diinginkan oleh user.
c. Tepat Waktu (*timelines*), informasi yang datang pada penerima tidak boleh terlambat. Suatu informasi harus sesuai dengan keadaan saat itu. Keterlambatan suatu informasi bisa berakibat fatal bagi suatu organisasi atau pemakainya hal ini dikarenakan informasi merupakan landasan dalam pengambilan keputusan.
d. Ekonomis (*economy*), informasi yang dihasilkan harus mempunyai daya jual yang tinggi dan biaya operasional yang harus dikeluarkan untuk menghasilkan informasi tersebut harus minimal, informasi tersebut juga mapu memberikan dampak yang luas terhadap laju pertumbuhan ekonomi dan teknologi informasi.
e. Efisien (*efficiency*), informasi yang berkualitas harus memiliki kalimat yang sederhana dan mudah dimengerti, tapi bisa memberikan makna yang mendalam.

f. Dapat dipercaya (*reliability*), informasi yang didapat harus dari sumber yang bisa dipercaya. Sumber tersebut juga harus sudah teruji tingkat kejujurannya.

2.2.4 Nilai Informasi

Manfaat informasi adalah untuk membantu memberi kejelasan dari suatu ketidakpastian atau untuk mengurangi ketidakpastian tersebut, sehingga dapat membuat sesuatu keputusan dengan kepastian yang lebih baik dan menguntungkan. (Zulkifly Amsyah : 314).

2.3 Konsep Dasar Sistem Informasi

2.3.1 Pengertian Sistem Informasi

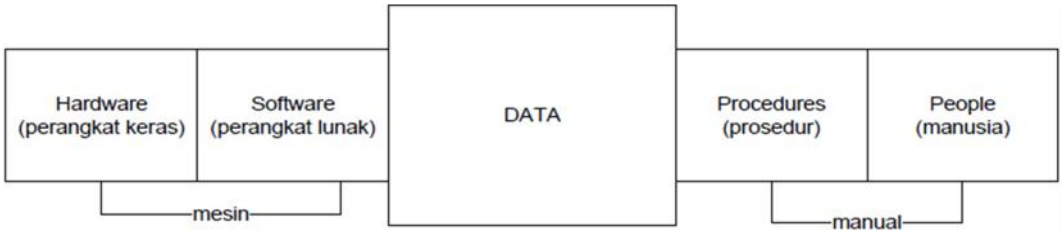
Pengertian sistem informasi yang terdapat pada buku Al-Bahra (2005:13), sistem informasi dapat didefinisikan sebagai berikut:

a. Suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.
b. Sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan / atau untuk mengendalikan organisasi.
c. Suatu sistem dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

2.3.2 Komponen Sistem Informasi

Adapun beberapa komponen sistem informasi dalam buku Al-bahra (2005:14) dapat diklasifikasikan sebagai berikut:

1. *Hardware* dan *software* yang berfungsi sebagai mesin. *Hardware* merupakan adalah semua bagian fisik komputer. Sedangkan *software* merupakan kumpulan dari perintah / fungsi yang ditulis dengan aturan tertentu untuk memerintahkan komputer dalam melaksanakan tugas tertentu.
2. *People* dan *procedures* yang merupakan manusia dan tatacara menggunakan mesin. *People* (manusia) adalah mereka yang terlibat dalam kegiatan sistem informasi seperti *operator*, pemimpin sistem informasi dan sebagainya. Sedangkan *procedures* (prosedur) merupakan tatacara yang menghubungkan berbagai perintah, dan aturan yang akan menentukan rancangan dan penggunaan sistem informasi.
3. Data merupakan komponen dasar dari informasi yang akan diproses lebih lanjut untuk menghasilkan informasi dan merupakan penghubung antara manusia dan mesin agar terjadi suatu proses pengolahan data.



Gambar 2.2 Lima Komponen Sistem Informasi

[Sumber : Al-bahra bin Ladjamudin. 2005. *Analisis dan Desain Sistem Informasi.* Graha Ilmu. Yogyakarta.]

2.3.3 Kegiatan Sistem Informasi

Adapun kegiatan sistem informasi yang akan dipaparkan berdasarkan buku Al-Bahra (2005:22) yaitu:

1. *Input*

   Menggambarkan suatu kegiatan untuk menyediakan data untuk diproses.

2. Proses

   Menggambarkan bagaimana suatu data diproses untuk menghasilkan suatu informasi yang bernilai tambah.

3. *Output*

   Suatu kegiatan untuk menghasilkan laporan dari proses diatas tersebut.

4. Penyimpanan

   Suatu kegiatan untuk memelihara dan menyimpan data.

5. *Control*

   Suatu aktivitas untuk menjamin bahwa sistem informasi tersebut berjalan sesuai dengan yang diharapkan.

2.4 Pengertian Sistem Informasi Akademik

Menurut Kamus Besar Bahasa Indonesia Edisi Kedua, Akademi adalah "*Lembaga pendidikan tinggi kurang lebih 3 tahun lamannya yang mendidik tenaga profesi"*. Sedangkan Akademik sendiri adalah kondisi dimana orang-orang bisa menyampaikan dan menerima gagasan pemikiran dan ilmu pengetahuan sekaligus dapat mengujinya secara bebas, jujur dan leluasa.

Maka Sistem Informasi Akademik adalah : Sistem yang memberikan layanan informasi yang berupa data dalam hal yang berhubungan dengan Akademik. Dimana dalam hal ini pelayanan yang di berikan yaitu seperti : penyimpanan data untuk siswa baru, penentuan kelas, penentuan jadwal pelajaran, pembuatan jadwal mengajar, pembagian wali kelas, proses penilaian.

2.5 Metodologi Berorientasi Objek

Metodologi ini diperkenalksan sekitar tahun 1990 sebagai pelengkap untuk pemrograman yang terlebih dahulu telah mengadopsi metode berorientasi objek. Beberapa alat dan teknik yang dapat digunakan antara lain *dynamic* dan *static object-oriented model*, *state transisition diagram* dan *case scenario*.

Fokus utama metodologi ini pada objek, dengan melihat suatu sistem terdiri dari objek yang saling berhubungan dengan beberapa cara untuk mencapai suatu tujuan. Objek dapat digambarkan sebagai benda,

orang, tempat dan sebaginya yang mempunyai atribut dan *method*. Obyek dalam „*software analysis & design*“ adalah sesuatu berupa konsep (*concept*), benda (*thing*), dan sesuatu yang membedakannya dengan lingkungannya. Secara sederhana obyek adalah mobil, manusia, alarm dan lain-lainnya. Tapi obyek dapat pula merupakan sesuatu yang abstrak yang hidup didalam sistem seperti tabel, *database*, *event*, *system messages*. Obyek dikenali dari keadaannya dan juga operasinya. Sebagai contoh sebuah mobil dikenali dari warnanya, bentuknya, sedangkan manusia dari suaranya. Ciri-ciri ini yang akan membedakan obyek tersebut dari obyek lainnya.

Object Oriented Programming merupakan cara atau metode baru dalam membuat program. Langkah membuat program dengan teknik OOP, pertama kali mengetahui objeknya, kemudian atribut (*field/data*) dan perilakunya (fungsi). Berbeda dengan pemrograman prosedural (konvensional), membuat program memakai aspek *input*, proses, dan *output*. OOP adalah cara berpikir, pandangan, atau paradigma baru untuk membuat program/merancang sistem dengan memperhatikan objek, cirri objek, dan perilakunya. OOP ini sangat berbeda dengan program prosedural yang fokusnya aspek input, proses, dan output.

2.5.1 Pengertian *Object Oriented Analysis and Design*

Menurut Mathiassen *Object-Oriented Analysis and Design* (OOAD) adalah metode untuk menganalisa dan merancang sistem dengan

pendekatan berorientasi object. *Object* diartikan sebagai suatu entitas yang memiliki identitas, *state*, dan *behavior*.

Pada analisa, identitas sebuah *object* menjelaskan bagaimana seorang *user* membedakannya dari *object* lain, dan *behavior object* digambarkan melalui event yang dilakukannya. Sedangkan pada perancangan, identitas sebuah *object* digambarkan dengan cara bagaimana *object* lain mengenalinya sehingga dapat diakses, dan *behavior object* digambarkan dengan *operation* yang dapat dilakukan *object* tersebut yang dapat mempengaruhi *object* lain dalam sistem.

2.5.2 Prinsip Umum OOAD

1. *Model the context* : Sistem yang bermanfaat sesuai dengan konteks OOAD. *Emphasize the architecture* : Merupakan arsitektur yang mudah dipahami yang memfasilitasi kolaborasi antara *designer* dan *programmer*.
2. *Reuse Patters* : Dibangun berdasarkan gagasan-gagasan yang kuat dan komponen *pretested* memperbaiki kualitas sistem dan produktivitas dari proses *development*.
3. *Tailor the method to suit specific projects* : Setiap usaha *devlopment* masing-masing mempunyai tantangan yang unik. OOA&D harus disesuaikan dengan kebutuhankebutuhan yang khusus dari situasi analisis dan desain yang diberikan.

2.5.3 Keuntungan OOAD

1. Menurut konsep umum yang dapat digunakan untuk memodelkan hampir semua fenomena dan dapat dinyatakan dalam bahasa umum (natural language).
   a. *Noun* menjadi *object* atau *class.*
   b. *Verb* menjadi *behaviour*.
   c. *Adjective* menjadi *attribute*.
2. Memberikan informasi yang jelas tentang *context* dari sistem.
3. Mengurangi biaya *maintainance*.
4. Memudahkan untuk mencari hal yang akan diubah.
5. Membuat perubahan menjadi lokal tidak berpengaruh pada modul yang lain.

2.5.4 Karakteristik OOAD

Dalam pendekatan berorientasi objek ada 4 pilar utama yang harus dipahamai dalam pendekatan berorientasi objek yaitu karakteristik. Karakteristik (ciri) suatu program termasuk OOAD/OOP, apabila terdapat abstraksi, pembungkusan (*encapsulation*), *polymorphisme*, dan turunan (*inheritance*).

1. *Abstraction*

   Kemampuan untuk menjadikan dalam bentuk yang lebih sederhana. Hal ini juga dikenal dalam metodologi pendekatan struktur yaitu dekomposisi seperti menyerderhanakan suatu sistem dalam bentuk *Context* Diagram.

2. *Encapsulation*

   Merupakan suatu karakteristik OOAD dimana program terbungkus (jadi satu) data dan perilaku, artinya lebih memperhatikan aspek internal daripada aspek eksternal. Contoh: dalam program terdapat tombol *button close* didalamnya ada *method system.exit*(0) untuk keluar dari sistem java. Berbeda dengan metodologi terdahulu, metodologi ini menggabungkan atribut dan fungsi / proses kedalam suatu objek yang disebut dengan encapsulation. Setiap objek dapat "menyembunyikan" kompleksitasnya dan berhubungan dengan objek lain dengan mengirim "pesan / *message*" yang dapat dikenal dan diproses oleh objek penerima. Contoh: Pada dunia nyata, seorang ibu rumah tangga menanak nasi dengan menggunakan *rice cooker*, ibu tersebut menggunakannya hanya dengan menekan tombol. Tanpa harus tahu bagaimana proses itu sebenarnya terjadi. Disini terdapat penyembunyian informasi milik *rice cooker*, sehingga tidak perlu diketahui seorang ibu. Dengan demikian menanak nasi oleh si ibu menjadi sesuatu yang menjadi dasar bagi konsep *information hiding*.

3. *Polymorphism*e

   Dengan kata lain suatu mekanisme yang memungkinkan suatu objek memiliki semua atau sebagian definisi dari objek induk. Menurut Bambang Hariyanto (2007:67) Polymorphism berasal dari kata Poly yang artinya banyak dan morph yang artinya bentuk. Jadi *polymorphism* adalah kemampuan suatu atribut atau method dapat

berubah dalam berbagai bentuk dalam implementasi. Contoh Pada obyek mobil, walaupun minibus dan truk merupakan jenis obyek mobil yang sama, namun memiliki juga perbedaan. Misalnya suara truk lebih keras dari pada minibus, hal ini juga berlaku pada obyek anak (*child*) melakukan metoda yang sama dengan algoritma berbeda dari obyek induknya. Hal ini yang disebut *polymorphism*, teknik atau konsep dasar lainnya adalah ruang lingkup/pembatasan. Artinya setiap obyek mempunyai ruang lingkup kelas, atribut, dan metoda yang dibatasi.

4. *Inheritance*

   Merupakan suatu karakteristik OOAD di mana suatu kelas (*parent/base class*) dapat diturunkan ke kelas lain (*child/derived class*), sehingga kelas anak dapat memiliki data atau perilaku kelas orangtuanya. Contoh dengan beberapa buah mobil yang mempunyai kegunaan yang berbeda-beda. Ada mobil bak terbuka seperti truk, bak tertutup seperti sedan dan minibus. Walaupun demikian obyek-obyek ini memiliki kesamaan yaitu teridentifikasi sebagai obyek mobil, obyek ini dikatakan obyek induk (*parent)*. Sedangkan minibus obyek anak (*child*), berarti semua operasi yang berlaku pada mobil berlaku pada minibus.

2.5.5 Ciri-ciri Object Oriented Programming (OOP)

Menurut Bambang Hariyanto dalam bukunya yang berjudul Rekayasa Perangkat Lunak Berorientasi Objek, bahwa *Object Oriented* memiliki cirri-ciri sebagai berikut: Bambang Hariyanto (2007:67)

1. Objek

   Bentuk baik yang nyata atau tidak, seperti manusia, hewan, benda, konsep, aliran, dan lain-lain. Objek merupakan inisiasi (turunan langsung) dari suatu kelas.

2. Kelas

   Kumpulan objek yang memiliki kemiripan perilaku (method), ciri atau karakteristik (*property*). Contoh objek orang dari kelas manusia, potongan sebagai berikut: Manusia orang1=*new* manusia(“Parto”);

3. Method

   Perilaku dari objek atau kelas tertentu. Merupakan perwujudan aksi atau tindakan dari dunia nyata di dalam pemrograman komputer.

4. Konstruktor

   Suatu fungsi yang dideklarasikan atau didefinisikan di dalam kelas, konstruktor harus mempunyai nama yang sama dengan fungsinya. Konstruktor dijalankan bersamaan dengan terciptanya kelas tersebut. Dalam suatu kelas bias terdapat lebih dari satu konstruktor. Konstruktor seperti method tetapi tidak mengembalikan nilai dan dapat didefinisikan tanpa parameter atau memakainya.

5. De-konstruktor

   Fungsi yang dideklarasikan dalam kelas, nama sama dengan nama fungsinya. Tetapi dijalankan bersamaan dengan dimusnahkannya kelas tersebut.

6. Karakteristik / properties

   Ciri yang dimiliki oleh suatu objek, karakteristik ini juga sebagai pembeda objek satu dengan objek lainnya dalam kelas yang sama (konsep individu).

7. Variabel

   Tempat menampung data sementara, dalam pemrograman objek biasanya disebut data, sedangkan dalam pemrograman prosedural sering disebut dengan variabel.

8. Data

   Istilah lain dari variabel dalam OOP. Dalam pemrograman java bisa juga disebut *field*, data *member* atau *instance variable*.

9. Hak akses (*access attribute*)

   Hak akses digunakan untuk dapat menentukan data *member* mana yang dapat digunakan oleh kelas lain, dan mana yang tidak dapat digunakan. Hak akses ini sangat penting dalam membuat program turunan kelas.

   a. *Public*

      Data *member* atau *variable* dapat diakses dari kelas mana saja.

   b. *Protected*

      Dapat mengakses data member dari kelas dalam *package* yang sama dan subkelasnya.

c. *Private*

Kelas yang data membernya memakai *private* hanya dapat digunakan oleh kelas bersangkutan, tidak dapat digunakan kelas lain.

d. Tidak disebutkan

Data member dapat diakses dari kelas dalam *package* yang sama.

2.6 Bahasa Pemodelan UML (*Unified Model Language)*

Notasi UML dibuat sebagai kolaborasi dari Graddy Booch, James Rumbough, Rebbecca Wirfs-Brock, Peter Yourdon, dan lain-lain. Jacobson menulis tentang pendefinisian persyaratan-persyaratan system yang disebut *use case*. Juga mengembangkan sebuah metode untuk perancangan sistem yang disebut *Object-Oriented Software Engineering* (OOSE) yang berfokus pada analisis. Booch, Rumbough, dan Jacobson biasa disebut dengan tiga sekawan (tree amigos). Semuanya bekerja di *Rational Software Corporation* dan berfokus pada standarisasi dan perbaikan ulang UML. Simbol UML mirip dengan Booch, notasi OMT, dan juga ada kemirioan dengan notasi lanilla.

Penggabungan beberapa metode menjadi UML dimulai tahun 1993. setiap orang dari *Rational* mulai menggabungkan idenya dengan metode-metode lanilla. Pada akhir tahun 1995 *Unified Method* versi 8.0 diperkenalkan. *Unified Method* diperbaiki dan diubah menjadi UML pada tahun 1996, UML 1.0 disahkan dan diberikan pada *Object Technology*

Group (OTG) pada tahun 1997, dan pada tahun itu juga beberapa perusahaan pengembangan utama perangkat lunak mulai mengadopsinya. Pada tahun yang sama OMG merilis UML 1.1 sebagai standar industri. UML menyediakan beberapa diagram visual yang menunjukkan berbagai aspek dalam sistem, ada beberapa diagram yang disediakan dalam UML, antara lain :

a. Diagram *Use Case* menyajikan interaksi antara *use case* dan aktor. Dimana, aktor dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. *Use Case* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai.
b. Diagram Aktivitas menggambarkan aliran fungsionalitas sistem. Pada tahap pemodelan bisnis, diagram aktivitas dapat digunakan untuk menunjukkan aliran kerja bisnis. Dapat juga digunakan untuk menggambarkan aliran kejadian dalam *use case*.
c. Diagram Sekuensial digunakan untuk menunjukkan aliran fungsionalitas dalam *use case*. Setiap objek yang terlibat dalam diagram *use case* digambarkan dengan garis putus-putus vertikal, kemudian *message* yang dikirim oleh objek digambarkan dengan garis *horizontal* secara kronologis dari atas ke bawah.
d. Diagram Kolaborasi menunjukkan informasi yang sama persisi dengan diagram sekuensial. Pada diagram sekuensial, keseluruhan interaksi berdasarkan urutan waktu, tetapi pada diagram kolaborasi, interaksi

antar objek atau aktor ditunjukkan dengan arah panah tanpa keterangan waktu.

e. Diagram Kelas menunjukkan interaksi antar kelas dalam sistem. Diagram kelas dibangun berdasarkan diagram *use case* dan diagram sekuensial yang telah dibuat sebelumnya. Kelas memiliki tiga area pokok yaitu :

   Nama (*stereotype*), Atribut dan Metode

   Atribut dan metode dapat memiliki salah satu sifat berikut :

   1. *Private,* tidak dapat dipanggil dari luar kelas yang bersangkutan.
   2. *Protected,* hanya dapat dipanggil oleh kelas yang bersangkutan dan anak-anak yang mewarisinya.
   3. *Public,* dapat dipanggil oleh siapa saja.

f. Diagram Statechart menyediakan sebuah cara untuk memodelkan bermacam-macam keadaan yang mungkin dialami oleh sebuah objek. Jika dalam diagram kelas menunjukkan gambaran statis kelas-kelas dan relasinya, diagram *statechart* digunakan untuk memodelkan tingkah laku dinamik sistem.

g. Diagram komponen menunjukkan model secara fisik komponen perangkat lunak pada sistem dan hubungannya antar mereka. Ada dua tipe komponen dalam diagram yaitu komponen *excurable* dan kode pustaka.

h. Diagram *deployment* menampilkan rancangan fisik jaringan dimana berbagai komponen akan terdapat di sana.

2.7 Jaringan Komputer

Jaringan komputer adalah sebuah sistem yang terdiri atas komputer-komputer yang didesain untuk dapat berbagi sumber daya (*printer, CPU*), berkomunikasi dan dapat mengakses informasi. Tujuan dari jaringan komputer adalah agar dapat mencapai tujuannya, setiap bagian dari jaringan komputer dapat meminta dan memberikan layanan. Pihak yang meminta atau menerima layanan disebut klien dan yang memberikan atau mengirim layanan disebut peladen. Disain ini disebut juga dengan sistem *client-server*, dan digunakan pada hampir seluruh aplikasi jaringan komputer.

2.7.1 Klasifikasi Jaringan komputer

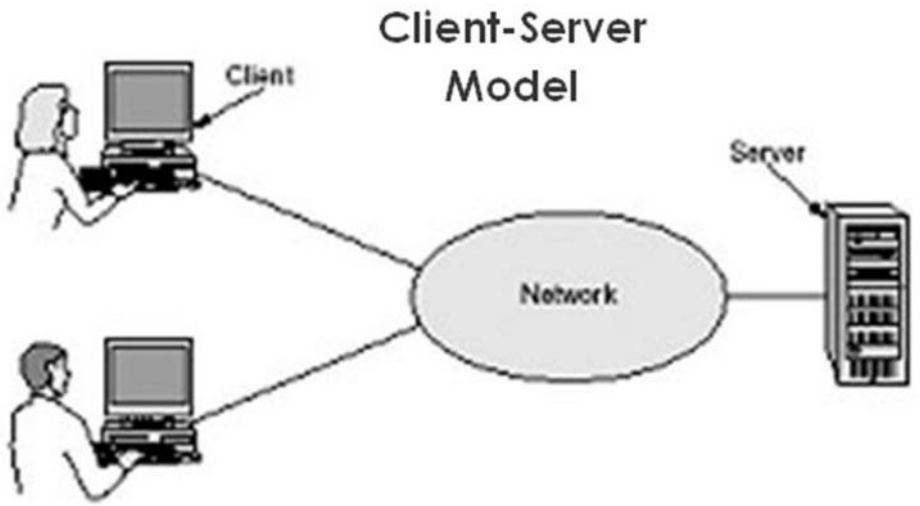
Klasifikasi jaringan komputer terbagi menjadi :

1. Berdasarkan geografisnya, jaringan komputer terbagi menjadi Jaringan wilayah lokal atau *Local Area Network* (LAN), Jaringan wilayah metropolitan atau *Metropolitan Area Network* (MAN) dan Jaringan wilayah luas atau *Wide Area Network* (WAN). LAN merupakan jaringan milik pribadi di dalam sebuah gedung atau tempat yang berukuran sampai beberapa 1 - 10 kilometer. LAN seringkali digunakan untuk menghubungkan komputer-komputer pribadi dan stasiun kerja (*workstation*) dalam kantor suatu perusahaan atau pabrik-pabrik untuk memakai bersama sumberdaya (misalnya pencetak (*printer*) dan saling bertukar informasi. Sedangkan Jaringan wilayah metropolitan merupakan perluasan jaringan LAN sehingga mencakup

satu kota yang cukup luas, terdiri atas puluhan gedung yang berjarak 10 - 50 kilometer. Kabel transmisi yang digunakan adalah kabel serat optik (*Coaxial Cable*). Jaringan wilayah luas Merupakan jaringan antarkota, antar propinsi, antar negara, bahkan antar benua. araknya bisa mencakup seluruh dunia, misalnya jaringan yang menghubungkan semua bank di Indonesia, atau jaringan yang menghubungkan semua kantor Perwakilan Indonesia di seluruh dunia. Media transmisi utama adalah komunikasi lewat satelit, tetapi banyak yang mengandalkan koneksi serat optik antar negara.

2. Berdasarkan fungsi, terbagi menjadi Jaringan Klien-server (*Client-server*) dan Jaringan Ujung ke ujung (*Peer-to-peer*). Jaringan klien-server pada dasaranya ada satu komputer yang disiapkan menjadi peladen (server) dari komputer lainnya yang sebagai klien (*client*). Semua permintaan layanan sumberdaya dari komputer klien harus dilewatkan ke komputer peladen, komputer peladen ini yang akan mengatur pelayanannya. Apabila komunikasi permintaan layanan sangat sibuk bahkan bisa disiapkan lebih dari satu komputer menjadi peladen, sehingga ada pembagian tugas, misalnya *file-server, print-server, database server* dan sebagainya. Tentu saja konfigurasi komputer peladen biasanya lebih dari konfigurasi komputer klien baik dari segi kapasitas memori, kapasitas cakram keras (*harddisk*), maupun kecepatan prosessornya. Sedangkan jaringan ujung ke ujung itu ditunjukkan dengan komputer-komputer saling mendukung, sehingga

setiap komputer dapat meminta pemakaian bersama sumberdaya dari komputer lainnya, demikian pula harus siap melayani permintaan dari komputer lainnya. Model jaringan ini biasanya hanya bisa diterapkan pada jumlah komputer yang tidak terlalu banyak, maksimum 25, karena komunikasi akan menjadi rumit dan macet bilamana komputer terlalu banyak.



Gambar 2.3 Model Jaringan *Client-Server*

[Sumber : http://id.wikipedia.org/wiki/Jaringan_komputer]

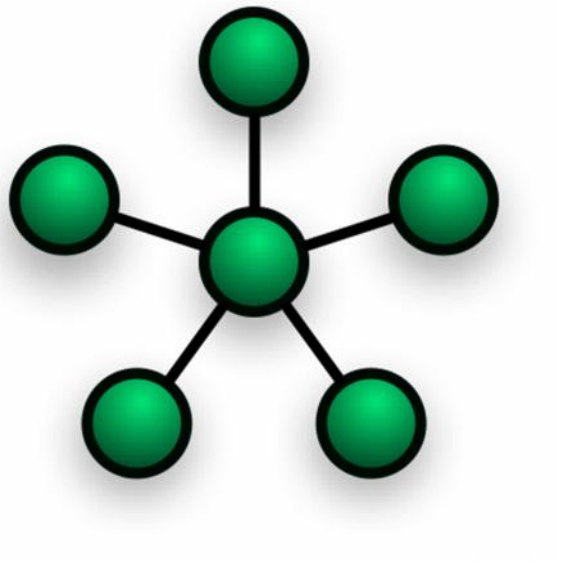
3. Berdasarkan topologi jaringan, jaringan komputer dapat dibedakan atas:
    1. Topologi bus merupakan topologi yang banyak dipergunakan pada masa penggunaan kabel sepaksi menjamur. Dengan menggunakan T-*Connector* (dengan terminator 50ohm pada ujung *network*),

maka komputer atau perangkat jaringan lainnya bisa dengan mudah dihubungkan satu sama lain. Ciri-ciri dari topologi bus adalah :

a. Teknologi lama, dihubungkan dengan satu kabel dalam satu baris
b. Tidak membutuhkan peralatan aktif untuk menghubungkan terminal/komputer
c. Sangat berpengaruh pada unjuk kerja komunikasi antar komputer, karena hanya bisa digunakan oleh satu komputer
d. Kabel "*cut*" dan digunakan konektor BNC tipe T
e. Diujung kabel dipasang 50 ohm konektor
f. Jika kabel putus maka komputer lain tidak dapat berkomunikasi dengan lain
g. Susah melakukan pelacakan masalah
h. *Discontinue Support*.

2. Topologi bintang merupakan bentuk topologi jaringan yang berupa konvergensi dari node tengah ke setiap node atau pengguna. Topologi jaringan bintang termasuk topologi jaringan dengan biaya menengah. Kelebihan dari topologi bintang adalah :
   a. Kerusakan pada satu saluran hanya akan memengaruhi jaringan pada saluran tersebut dan station yang terpaut.
   b. Tingkat keamanan termasuk tinggi.
   c. Tahan terhadap lalu lintas jaringan yang sibuk.

d. Penambahan dan pengurangan station dapat dilakukan dengan mudah.
e. Akses Kontrol terpusat.
f. Kemudahan deteksi dan isolasi kesalahan/kerusakan pengelolaan jaringan.
g. Paling fleksibel.



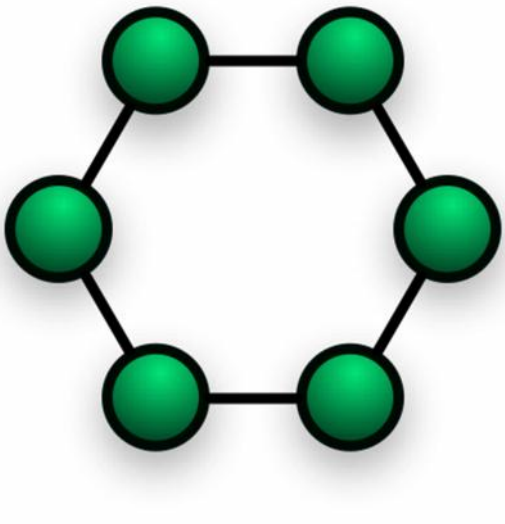
Gambar 2.4 Topologi Bintang

[Sumber : http://id.wikipedia.org/wiki/Topologi_bintang]

3. Topologi Cincin adalah topologi jaringan berbentuk rangkaian titik yang masing-masing terhubung ke dua titik lainnya, sehingga membentuk jalur melingkar membentuk cincin. Pada topologi cincin, komunikasi data dapat terganggu jika satu titik mengalami gangguan. Jaringan FDDI mengantisipasi kelemahan ini dengan

mengirim data searah jarum jam dan berlawanan dengan arah jarum jam secara bersamaan. Kelebihan topologi cincin adalah :

a. Hemat kabel
b. Tidak akan terjadi tabrakan pengiriman data (*collision*), karena pada satu waktu hanya satu node yang dapat mengirimkan data.



Gambar 2.5 Topologi Cincin

[Sumber : http://id.wikipedia.org/wiki/Topologi_cincin]

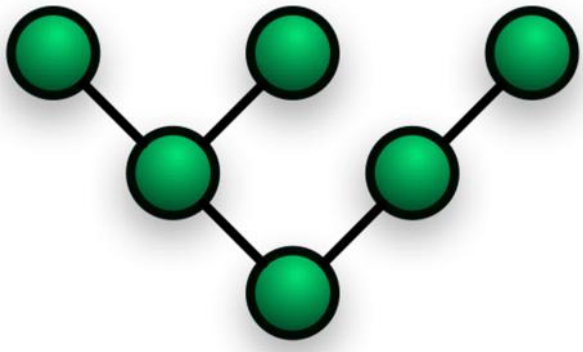
4. Topologi Jala atau topologi mesh adalah suatu bentuk hubungan antar perangkat dimana setiap perangkat terhubung secara langsung ke perangkat lainnya yang ada di dalam jaringan. Akibatnya, dalam topologi mesh setiap perangkat dapat berkomunikasi langsung dengan perangkat yang dituju (*dedicated links*).

   Dengan demikian maksimal banyaknya koneksi antar perangkat pada jaringan bertopologi mesh ini dapat dihitung yaitu sebanyak

n(n-1)/2. Selain itu karena setiap perangkat dapat terhubung dengan perangkat lainnya yang ada di dalam jaringan maka setiap perangkat harus memiliki sebanyak n-1 *Port Input/Output* (I/O *ports*).

5. Topologi Pohon adalah kombinasi karakteristik antara topologi bintang dan topologi bus. Topologi ini terdiri atas kumpulan topologi bintang yang dihubungkan dalam satu topologi bus sebagai jalur tulang punggung atau *backbone*. Komputer-komputer dihubungkan ke hub, sedangkan hub lain di hubungkan sebagai jalur tulang punggung.

   Topologi jaringan ini disebut juga sebagai topologi jaringan bertingkat. Topologi ini biasanya digunakan untuk interkoneksi antar sentral dengan hirarki yang berbeda. Untuk hirarki yang lebih rendah digambarkan pada lokasi yang rendah dan semakin keatas mempunyai hirarki semakin tinggi. Topologi jaringan jenis ini cocok digunakan pada sistem jaringan komputer.



Gambar 2.6 Topologi Pohon

[Sumber : http://id.wikipedia.org/wiki/Topologi_pohon]

6. *Linear topology* biasa disebut dengan topologi bus beruntut, tata letak ini termasuk tata letak umum. Satu kabel utama menghubungkan tiap titik sambungan (komputer) yang dihubungkan dengan penyambung yang disebut dengan Penyambung-T dan pada ujungnya harus diakhiri dengan sebuah penamat (terminator). Penyambung yang digunakan berjenis BNC (*British Naval Connector*: Penyambung Bahari Britania), sebenarnya BNC adalah nama penyambung bukan nama kabelnya, kabel yang digunakan adalah RG 58 (Kabel Sepaksi Thinnet). Pemasangan dari topologi bus beruntut ini sangat sederhana dan murah tetapi sebanyaknya hanya dapat terdiri dari 5-7 komputer.

4. Berdasarkan distribusi sumber informasi atau data

    1. Jaringan terpusat

        Jaringan ini terdiri dari komputer klien dan peladen yang mana komputer klien yang berfungsi sebagai perantara untuk mengakses sumber informasi/data yang berasal dari satu komputer peladen.

    2. Jaringan terdistribusi

        Merupakan perpaduan beberapa jaringan terpusat sehingga terdapat beberapa komputer peladen yang saling berhubungan dengan klien membentuk sistem jaringan tertentu.

5. Berdasarkan media trasmisi data

    1. Jaringan Berkabel (*Wired Network*)

        Pada jaringan ini, untuk menghubungkan satu komputer dengan komputer lain diperlukan penghubung berupa kabel jaringan. Kabel jaringan berfungsi dalam mengirim informasi dalam bentuk sinyal listrik antar komputer jaringan.

    2. Jaringan nirkabel (*Wi-Fi*)

        Merupakan jaringan dengan medium berupa gelombang elektromagnetik. Pada jaringan ini tidak diperlukan kabel untuk menghubungkan antar komputer karena menggunakan gelombang elektromagnetik yang akan mengirimkan sinyal informasi antar komputer jaringan.

2.7.2 Manfaat Jaringan Komputer

Banyak Manfaat atau keuntungan yang bisa kita dapatkan dengan dibangunnya sebuah Jaringan Komputer diantaranya yaitu :

1. Berbagi sumber daya komputer.
2. Peningkatan kehandalan.
3. Penghematan.

Sumber daya komputer yang dibagi bersama mencakup CPU, alokasi memori, penyimpanan data/ disk, pencetak/ printer, dan program/ utilitas. CPU/memori yang sedang diam (*idle*) dapat "dikaryakan" oleh sistem lain yang sedang sibuk. *Disk* yang semula tersebar di semua komputer dengan utilitas yang sama kini dapat digabungkan menjadi unit disk yang besar dan dibagi bersama. berbagi sumber daya komputer.

2.8 Perangkat Lunak Pendukung

Berikut adalah perangkat lunak pendukung untuk merancang dan membangun aplikasi Sistem Informasi Akademik ini.

2.8.1 Netbeans

NetBeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra (dan terus bertambah!). *Sun Microsystems* mendirikan proyek kode terbuka NetBeans pada bulan Juni 2000 dan terus menjadi sponsor utama. Saat ini terdapat dua produk : *NetBeans* IDE dan *NetBeans Platform* JAVA.

Sejarah NetBeans

Pengembangan *NetBeans* diawali dari *Xelifi*, sebuah proyek mahasiswa tahun 1997 di bawah bimbingan Fakultas matematika dan Fisika Universitas Charles, Praha. Sebuah perusahaan kemudian dibentuk untuk proyek tersebut dan menghasilkan versi komersial *NetBeans* IDE hingga kemudian dibeli oleh *Sun Microsistem* pada tahun 1999. *Sun* kemudian menjadikan *NetBeans open source* pada bulan Juni tahun 2000. Sejak itu komunitas *NetBeans* terus berkembang. *NetBeans* mengacu pada dua hal, yakni platform untuk pengembangan aplikasi *desktop* java, dan sebuah *Integrated Development Kit* (IDE) yang dibangun menggunakan platform *NetBeans*.

Platfom *NetBeans* memungkinkan aplikasi dibangun dari sekumpulan komponen perangkat lunak moduler yang disebut 'modul'. Sebuah modul adalah suatu arsip Java (*Java archive*) yang memuat kelas-kelas Java untuk berinteraksi dengan *NetBeans Open API* dan *file manifestasi* yang mengidentifikasinya sebagai modul. Aplikasi yang dibangun dengan modul-modul dapat dikembangkan dengan menambahkan modul-modul baru. Karena modul dapat dikembangkan secara independen, aplikasi berbasis platform *NetBeans* dapat dengan mudah dikembangkan oleh pihak ketiga secara mudah dan powerfull .

Netbeans IDE

NetBeans IDE adalah IDE *open source* yang ditulis sepenuhnya dengan bahasa Java menggunakan platform *NetBeans*. *NetBeans* IDE

mendukung pengambangan semua tipe aplikasi Java (J2SE, web, EJB, dan aplikasi mobile). Fitur lainnya adalah sistem proyek berbasis Ant, control versi, dan refactoring .

Modularitas : Semua fungsi IDE disediakan oleh modul-modul. Tiap modul menyediakan fungsi yang didefinisikan dengan baik, seperti dukungan untuk bahasa pemrograman Java, editing, atau dukungan bagi CVS. *NetBeans* memuat semua modul yang diperlukan dalam pengembangan Java dalam sekali download, memungkinkan pengguna untuk mulai bekerja sesegera mungkin. Modul-modul juga mengijinkan *NetBeans* untuk dikembangkan. Fitur-fitur baru, seperti dukungan bahsa pemrograman lain, dapat ditambahkan dengan menginstal modul tambahan. Sebagai contoh, Sun Studio, Sun Java Studio Enterprise, dan Sun Java Studio Creator dari Sun Microsistem semuanya berbasis NetBeans IDE

2.8.2 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data *SQL* (bahasa Inggris: database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. *MySQL* AB membuat *MySQL* tersedia sebagai perangkat lunak gratis dibawah lisensi *GNU General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL Tidak sama dengan proyek-proyek seperti *Apache,* dimana perangkat lunak dikembangkan

oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, *MySQL* dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia *MySQL* AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan *MySQL* AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan *MySQL*, namun tidak boleh dijadikan produk turunan yang bersifat closed source atau komersial. *MySQL* sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu *SQL* (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya.

2.8.3 XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache* HTTP *Server*, MySQL *database*, dan penerjemah bahasa

yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Untuk mendapatkanya dapat men-download langsung dari web resminya.

2.8.4 Rational Rose

Menurut Adi Nugroho (2005:20) dalam bukunya yang berjudul *Rational Rose* untuk Pemodelan Berorientasi Objek, menerangkan bahwa *Rational Rose* adalah kakas (*tools*) pemodelan visual untuk pengembangan sistem berbasis objek yang sangat handal untuk digunakan sebagai bantuan bagi para pengembang dalam melakukan analisis dan perancangan sistem.

Rational Rose digunakan untuk melakukan pemodelan sistem sebelum pengembangan menulis kode-kode dalam bahasa pemrograman tertentu. *Rational Rose* mendukung pemodelan bisnis yang membantu para pengembang untuk memahami sistem secara komprehensif. *Rational Rose* juga membantu analisis sistem dengan cara pengembang membuat diagram use case untuk melihat fungsionalitas sistem secara keseluruhan sesuai dengan harapan dan keinginan pengguna, juga menuntut pengembang untuk mengembangkan *Interaction* Diagram untuk melihat bagaimana objek-objek saling bekerjasama dalam menyediakan fungsionalitas yang diperlukan.

Dalam *Rational Rose,* pemodelan adalah cara melihat sistem dari berbagai sudut pandang mencakup semua program yang dikenal dalam UML, aktor-aktor yang terlibat dalam sistem, *use-case*, objek-objek, kelas-kelas, komponen-komponen, serta simpul-simpul penyebaran (*deployment node*).

2.9 Konsep Sistem Pembelajaran

Pada Lembaga Kursus Bahasa Asing FCC (*Foreign Language Course and Club*) memiliki tiga program belajar bahasa asing yaitu program dua bulan dengan target aktif percakapan bahasa Inggris, program delapan bulan dengan target aktif, fasih dan bisa mengajar bahasa Inggris, lalu program lima bulan bahasa Perancis.

Untuk program kursus bahasa Inggris peserta didik atau member akan mendapatkan garansi belajar apabila target yang telah ditetapkan tidak tercapai dengan syarat jumlah kehadiran baik yaitu member tidak boleh tidak mengikuti pembelajaran lebih dari empat kali pertemuan baik itu program delapan bulan maupun program dua bulan. Dan pengajar untuk program kursus bahasa Inggris adalah pengajar local berkebangsaan Indonesia.

Sedangkan untuk program kursus bahasa Perancis pihak Lembaga belum menetapkan garansi belajar dikarenakan pembelajaran itu sendiri dilakukan oleh *native speaker* berkebangsaan Perancis.

2.9.1 Program 2 Bulan

Program belajar bahasa Inggris dengan waktu dua bulan dikhususkan untuk orang-orang yang telah mempunyai dasar-dasar bahasa Inggris dengan pertemuan tiga kali dalam satu minggu dan target yang akan dicapai adalah hanya Aktif Percakapan bahasa Inggris dengan jumlah pertemuan sebanyak dua puluh empat kali pertemuan. Jumlah pembayaran untuk program dua bulan ini adalah Rp. 2.300.000,- dan pembayaran hanya dilakukan sebanyak satu kali atau tunai.

2.9.2 Program 8 Bulan

Program belajar bahasa Inggris dengan waktu delapan bulan dikhususkan untuk orang-orang memang belum mempunyai dasar-dasar bahasa Inggris dengan pertemuan dua kali satu minggu dan target yang akan dicapai adalah Aktif Percakapan, Fasih dan Bisa Mengajar bahasa Inggris dan jumlah pertemuan sebanyak enam puluh empat kali pertemuan. Jumlah pembayaran untuk program delapan bulan ini adalah Rp. 5.400.000,- dan pembayaran dapat dilakukan dengan tiga cara bayar yaitu :

1. Tunai

   Untuk pembayaran secara tunai, pembayaran dilakukan pada saat pendaftaran atau jika pada saat proses pembelajaran pertama kalinya. Dan khusus untuk pembayaran tunai member mendapatkan potongan dan pembayaran menjadi Rp. 4.800.000,-.

2. Cicil Reguler

   Untuk pembayaran secara cicil regular, pembayaran dilakukan pada tiga bulan pertama. Untuk jumlah cicilannya yaitu pada cicilan pertama member harus membayar cicilan sebesar Rp. 2.300.000,- cicilan kedua sebesar Rp. 2.100.000,- dan cicilan ketiga sebesar Rp. 1.000.000,-.

3. Cicil Autodebet

   Untuk pembayaran dengan cara bayar autodebet, member melakukan pembayaran sebanyak dua belas kali setiap tanggal 1 dan jumlah pembayaran sebesar Rp. 450.000,-.

2.9.3 Program 5 Bulan Bahasa Perancis

Program belajar bahasa Perancis dengan waktu lima bulan dikhususkan untuk orang-orang memang belum mempunyai dasar-dasar bahasa Perancis dengan pertemuan dua kali satu minggu dan target yang akan dicapai adalah Aktif Percakapan dan jumlah pertemuan sebanyak empat puluh kali pertemuan. Jumlah pembayaran untuk program lima bulan bahasa Perancis ini adalah Rp. 4.300.000,- dan pembayaran dapat dilakukan dengan dua cara bayar yaitu :

1. Tunai

   Untuk pembayaran secara tunai, pembayaran dilakukan pada saat pendaftaran atau jika pada saat proses pembelajaran pertama kalinya. Dan khusus untuk pembayaran tunai member

mendapatkan potongan dan pembayaran menjadi Rp. 3.800.000,-.

2. Cicil Reguler

   Untuk pembayaran secara cicil regular, pembayaran dilakukan pada tiga bulan pertama. Untuk jumlah cicilannya yaitu pada cicilan pertama member harus membayar cicilan sebesar Rp. 1.800.000,- cicilan kedua sebesar Rp. 1.500.000,- dan cicilan ketiga sebesar Rp. 1.000.000,-.