BAB II

LANDASAN TEORI

2.1 Pengertian Sistem

Pada bab ini, mengkaji lebih dalam mengenai landasan teori, mulai daripara ahli sampai dengan acuan atau pedoman yang dipakai oleh penulis, salah satunya adalah :

Sistem menurut Zulkifli (2005:4) sistem adalah himpunan sesuatu "benda" nyata atau abstrak *(a set of things)* yang terdiri dari bagian-bagian atau komponen-komponen yang saling berkaitan, berhubungan, berketergantungan dan saling mendukung, yang secara keseluruhan bersatu dalam satu kesatuan *(unity)* untuk mencapai tujuan tertentu secara efisien dan efektif.

Sistem menurut Jogiyanto (2005:1) suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaian suatu sasaran yang tertentu.

Berdasarkan pengertian di atas, penulis berpedoman pada pengertian yang dikemukakan oleh Zulkiflikarena sistem merupakan prosedur logis dan rasional untuk merancang suatu rangkaian komponen yang berhubungan satu dengan yang lainnya dengan maksud untuk berfungsi sebagai suatu kesatuan dalam usaha mencapai suatu tujuan yang telah ditentukan.

2.1.1 Elemen Sistem

Ada beberapa elemen yang membentuk sebuah sistem, yaitu :

1. Tujuan

Setiap sistem memiliki tujuan (*Goal*), entah hanya satu atau mungkin banyak.Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem.Tanpa tujuan, sistem menjadi tak terarah dan tak terkendali.Tentu saja, tujuan antara satu sistem dengan sistem yang lain berbeda.

2. Masukan

Masukan (*input*) sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan yang diproses. Masukan dapat berupa hal-hal yang berwujud (tampak secara fisik) maupun yang tidak tampak.Contoh masukan yang berwujud adalah bahan mentah, sedangkan contoh yang tidak berwujud adalah informasi (misalnya permintaan barang oleh pemohon).

3. Proses

Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna dan lebih bernilai, misalnya berupa informasi dan produk, tetapi juga bisa berupa hal-hal yang tidak berguna, misalnya saja sisa pembuangan atau limbah. Pada pabrik kimia, prosesdapat berupa bahan mentah. Pada Direktorat Aerostructure proses dapat berupa aktivitas perawatan komputer.

4. Keluaran

Keluaran (*output*) merupakan hasil dari pemrosesan. Pada sistem informasi, keluaran bisa berupa suatu informasi, saran, cetakan laporan, dan sebagainya.

5. Batas

Batas (*boundary*) sistem adalah pemisah antara sistem dan daerah di luar sistem (lingkungan).Batas sistem menentukan konfigurasi, ruang lingkup, atau kemampuan sistem. Sebagai contoh, tim sepakbola mempunyai aturan permainan dan keterbatasan kemampuan pemain. Pertumbuhan sebuah toko kelontong dipengaruhi oleh pembelian pelanggan, gerakan pesaing dan keterbatasan dana dari bank. Tentu saja batas sebuah sistem dapat dikurangi atau dimodifikasi sehingga akan mengubah perilaku system.

6. Mekanisme pengendalian dan umpan balik

Mekanisme pengendalian (*control mechanism*) diwujudkan dengan menggunakan umpan balik (*feedback*), yang mencuplik keluaran. Umpan balik ini digunakan untuk mengendalikan baik masukan maupun proses. Tujuannya adalah untuk mengatur agar sistem berjalan sesuai dengan tujuan.

7. Lingkungan

Lingkungan adalah segala sesuatu yang berada diluar sistem.Lingkungan bisa berpengaruh terhadap operasi sistem dalam arti bisa merugikan atau menguntungkan sistem itu sendiri. Lingkungan yang merugikan tentu saja harus ditahan dan dikendalikan supaya tidak mengganggu kelangsungan operasi sistem,

sedangkan yang menguntungkan tetap harus terus dijaga, karena akan memacu terhadap kelangsungan hidup sistem.

2.1.2 Karakterisitik Sistem

Karakteristik sistem mencakup beberapa hal, yaitu :

1. Organisasi

   Organisasi di dalamnya mencakup struktur organisasi dan fungsi organisasi. Organisasi tidak akan berjalan tanpa adanya fungsi dari setiap bagian maupun sub bagian.

2. Interaksi

   Interaksi menjelaskan tentang Saling keterhubungan antara bagian yang satu dengan lainnya.

3. Interdependensi

   Bagian yang satu mempunyai ketergantungan dengan bagian yang lainnya.

4. Integrasi

   Suatu keterpaduan antara subsistem-subsistem untuk mencapai tujuan.

5. Tujuan pokok

   Pemusatan tujuan yang sama dari masing-masing subsistem.

2.1.3 Klasifikasi Sistem

1. Deterministik Sistem

   Sistem dimana operasi-operasi (input/output) yang terjadi didalamnya dapatditentukan/ diketahui dengan pasti.

2. Probabilistik Sistem

Sistem yang input dan prosesnya dapat didefinisikan, tetapi output yang dihasilkan tidak dapat ditentukan dengan pasti.

3. Open Sistem

Sistem yang mengalami pertukaran energi, materi atau informasi dengan lingkungannya. Sistem ini cenderung memiliki sifat adaptasi, dapat menyesuaikan diri dengan lingkungannya sehingga dapat meneruskan eksistensinya.

4. Close Sistem

Sistem fisik di mana proses yang terjadi tidak mengalami pertukaran materi, energi atau informasi dengan lingkungan di luar sistem tersebut.

5. Relatively Closed Sistem

Sistem yang tertutup tetapi tidak tertutup sama sekali untuk menerima pengaruh-pengaruh lain. Sistem ini dalam operasinya dapat menerima pengaruh dari luar yang sudah didefinisikan dalam batas-batas tertentu.

6. Artificial Sistem

Sistem yang meniru kejadian dalam alam.Sistem ini dibentuk berdasarkan kejadian di alam di mana manusia tidak mampu melakukannya. Dengan kata lain tiruan yang ada di alam.

7. Natural Sistem

Sistem yang dibentuk dari kejadian dalam alam.

8. Manned Sistem

Sistem penjelasan tingkah laku yang meliputi keikut sertaan manusia.

2.2 Pengertian Informasi

Informasi merupakan kumpulan data-data yang sangat bermanfaat. Di bawah ini adalah pengertian informasi menurut para ahli, antara lain adalah :

Informasi menurut Zulkifli (2005 : 2) informasi adalah data yang sudah diolah, dibentuk, atau dimanipulasi sesuai dengan keperluan tertentu.

Informasi menurut McLeod (2001 : 12) informasi adalah data yang telah diproses atau data yang memiliki arti.

2.3 Pengertian Sistem Informasi

Sistem informasi merupakan kerangka kerja yang mengkoordinasikan sumber daya (manusia dan komputer) untuk mengubah masukan (input) menjadi keluaran (informasi) guna mencapai sasaran-sasaran perusahaan. Sistem informasi tidak harus melibatkan komputer, sistem informasi yang menggunakan komputer biasa disebut sistem informasi berbasis komputer, tetapi dalam prakteknya sistem informasi lebih sering dikait-kaitkan dengan komputer.Di bawah ini adalah pengertian Sistem Informasi menurut para ahli yang digunakan oleh penulis sebagai pedoman..

Sistem informasi menurut Jogiyanto (2005 : 11) mendefinisikan bahwa sistem informasi adalah suatu sistem di dalam 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.

Sistem informasi adalah kumpulan antara sub – sub sistem yang saling berhubungan yang membentuk suatu komponen yang didalamnya mencakup input – proses -output yang berhubungan dengan pengolahan informasi (data yang telah diolah sehingga berguna bagi user)

Penulis berpedoman pada pendapat yang dikemukakan oleh Jogiyanto (2005 : 11) karena sistem informasi merupakan suatu kesatuan yang memadukan antar sistem dan informasi yang dapat diolah dan diproses yang dapat mendukung dalam pengambilan keputusan.

2.4 Pengertian Distribusi

Distribusi adalah suatu proses penyampaian barang atau jasa dari produsen ke konsumen dan para pemakai, sewaktu dan dimana barang atau jasa tersebut diperlukan. Proses distribusi tersebut pada dasarnya menciptakan faedah (utility) waktu, tempat, dan pengalihan hak milik. Dalam menciptakan ketiga faedah tersebut, terdapat dua aspek penting yang terlibat didalamnya, yaitu :

1. Lembaga yang berfungsi sebagai saluran distribusi (Channel of distribution/marketing channel).
2. Aktivitas yang menyalurkan arus fisik barang (Physical distribution).

2.4.1 Fungsi Distribusi

Fungsi distribusi dilakukan oleh badan usaha atau perorangan sejak pengumpulan barang dengan jalan membelinya dari produsen untuk disalurkan ke konsumen, berdasarkan hal tersebut maka fungsi distribusi terbagi atas:

1. Fungsi pertukaran dimana kegiatan pemasaran atau jual beli barang atau jasa yang meliputi pembelian, penjualan, dan pengambilan resiko (untuk mengatasi resiko bisa dilakukan dengan menciptakan situasi dan kondisi pergudangan yang baik, mengasuransikan barang dagangan yang akan dan sedang dilakukan).
2. Fungsi penyediaan fisik berkaitan dengan menyediakan barang dagangan dalam jumlah yang tepat mencakup masalah pengumpulan, penyimpanan, pemilahan, dan pengangkutan.
3. Fungsi penunjang merupakan fungsi yang berkaitan dengan upaya memberikan fasilitas kepada fungsi-fungsi lain agar kegiatan distribusi dapat berjalan dengan lancar, fungsi ini meliputi pelayanan, pembelanjaan, penyebaran informasi, dan koordinasi.

2.5 Pengertian Persediaan Barang

Menurut Freddy Rangkuti (2004 : 1-2) pengertian dari persediaan adalahsebagai suatu aktiva yang meliputi barang-barang milik perusahaan denganmaksud untuk dijual dalam suatu periode usaha tertentu, atau persediaan barang-barangyang masih dalam pengerjaan / proses produksi, ataupun persediaan bahanbaku yang menunggu penggunaanya dalam suatu proses produksi. Jadi,persediaan merupakan bahan-bahan, bagian yang disediakan, dan bahan-bahan dalam proses yang terdapat dalam perusahaan untuk proses produksi, sertabarang-barang jadi atau produk yang disediakan untuk memenuhi permintaan dari konsumen atau pelanggan setiap waktu.

2.5.1 Fungsi persediaan

Persediaan memberikan berbagai manfaat yang ada pada umumnya yaitu mempermudah atau memperlancar jalannya operasi perusahaan yang harus dilakukan secara berturut-turut untuk memproduksi barang-barang serta menyampaikannya kepada para pelanggan atau konsumen. Hal tersebut sejalan dengan pendapat yang dikemukakan oleh Freddy Rangkuti (2004 : 7), bahwa persediaan yang diadakan mulai dari bahan baku sampai barang jadi berguna untuk :

1. Menghilangkan resiko keterlambatan datangnya barang;
2. Menghilangkan resiko barang yang rusak;
3. Mempertahankan stabilitas operasi perusahaan;
4. Mencapai penggunaan mesin yang optimal;
5. Memberikan pelayanan yang sebaik-baiknya bagi konsumen;

Persediaan merupakan salah satu unsur paling aktif dalam operasi perusahaan yang secara terus-menerus diperoleh, diubah, yang kemudian dijual kembali.

Persediaan merupakan kegiatan pemenuhan kebutuhan. Pengertian persediaan menurut Kamus Besar Bahasa Indonesia, persediaan berasal dari kata “sedia” dan ditambahkan awalan pe- dan akhiran –an sehingga mempunyai arti “Persediaan adalah proses menjadikan sesuatu yang tadinya tidak ada menjadi ada”.

Berdasarkan pemikiran tersebut diatas maka dapat diambil kesimpulan bahwa yang dimaksud dengan Sistem Informasi Persediaan Barang adalah

pola hubungan yang berkaitan untuk melakukan kegiatan memproses data kebutuhan barang dan jasa untuk produksi atau pendukung produksi yang tidak terpenuhi dan proses pembeliannya sehingga kebutuhan tersebut terpenuhi dan dihasilkan data yang cepat, tepat dan akurat, sehingga tujuan perusahaan dapat tercapai.

2.6 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.

2.6.1 Pengertian *Object Oriented Analysisand 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.

2.6.2 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.

2.6.3 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.

2.6.4 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.

2.6.5 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), cirri 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. *Publi*c

      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. *Privat*e

      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.

2.7 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.

2.8 Perangkat Lunak Pendukung

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

2.8.1 JAVA

Java adalah sebuah teknologi yang diperkenalkan oleh Sun Microsystems pada pertengahan tahun 1990. Menurut definisi dari Sun, Java adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan. Kita lebih menyukai menyebut Java sebagai sebuah teknologi dibanding hanya sebuah bahasa pemrograman, karena Java lebih lengkap dibanding sebuah bahasa pemrograman konvensional. Teknologi Java memiliki tiga komponen penting, yaitu:

1. *Programming-language specification*
2. *Application-programming interface*
3. *Virtual-machine specification*

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.