BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Sistem merupakan suatu sistem terdiri dari elemen-elemen atau variabel yang berhubungan melalui berbagai bentuk interaksi dan bekerjasama untuk mencapai tujuan-tujuan tertentu. Elemen-elemen tersebut mempunyai sifat atau karakteristik, untuk menjalankan tugas dan fungsinya elemen-elemen tersebut tidak dapat dipisahkan antara satu dengan yang lainya, karena saling berhubungan dan saling mempengaruhi proses secara keseluruhan. Jika dalam suatu sistem terdapat elemen yang tidak memberikan manfaat atau kontribusi terhadap tercapainya tujuan, maka elemen tersebut bukanlah bagian dari sistem.

2.1.1. Pengertian Sistem

Menurut Jerry FitzGerald et al, pendekatan sistem yang lebih menekankan pada prosedur didefinisikan bahwa suatu sistem adalah suatu jaringan kerja dari prosedur – prosedur yang saling berhubungan, berkumpul bersama – sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu. Jogianto (2005:1).

Menurut Jogianto (2005:2), pendekatan sistem yang lebih menekankan pada elemen atau komponennya mendefinisikan bahwa sistem adalah kumpulan dari elemen – elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

Dari kedua pengertian di atas dapat disimpulkan bahwa pengertian sistem adalah kumpulan elemen-elemen atau jaringan kerja dan prosedur-prosedur yang saling berinteraksi satu sama lain untuk mencapai tujuan atau sasaran tertentu.

2.1.2. Elemen Sistem

Berikut elemen-elemen dari sistem tersebut :

a. 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.

b. 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 jasa pelanggan).

c. Keluaran

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

d. Proses

Proses (*procces*) 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, proses dapat berupa bahan mentah. Pada rumah sakit, proses dapat berupa aktivitas pembedahan pasien.

e. 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.

2.1.3. Karakteristik Sistem

Karakteristik sistem adalah sistem yang mempunyai komponen-komponen, batas sistem, lingkungan sistem, penghubung, masukan, keluaran, pengolah dan sasaran. Menurut Jogianto (2005:3), Suatu sistem mempunyai karakteristik atau sifat–sifat yang tertentu, yaitu:

1. **Komponen Sistem (*components*)**

Suatu sistem terdiri dari sejumlah kumpulan komponen yang saling berinteraksi atau bekerja sama satu sama lain untuk membentuk satu kesatuan. Setiap komponen atau subsistem tersebut memiliki sifat-sifat dan fungsi tertentu yang mempengaruhi proses sistem secara keseluruhan.

2. Batas Sistem (*Boundary*)

Batas sistem (*Boundary*) yaitu daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya . Dan batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas sistem ini merupakan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (e*nvironment*)

Lingkungan luar (e*nvironment*) dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem yang harus tetap dijaga dan dipelihara. Sedangkan lingkungan luar sistem yang bersifat merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung Sistem (*Interface*)

Penghubung (*Interface*) 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. Keluaran (*output*) dari satu subsistem akan menjadi masukan (*input*) untuk subsistem yang lain dengan melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

5. Masukan Sistem (*Input*)

Masukan (*Input*) merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan

sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh di dalam sistem komputer, program adalah *maintaenance input* yang digunakan untuk mengoperasikan komputer dan data adalah signal *input* untuk diolah menjadi informasi.

6. Keluaran sistem (*Output*)

Keluaran (*Output*) merupakan hasil dari energi yang diolah. Keluaran ini dapat berupa keluaran yang berguna atau keluaran yang tidak berguna. Sebagai contoh panas yang dikeluarkan dari sistem komputer merupakan keluaran yang tidak berguna dan merupakan sisa pembuangan. Sedangkan informasi adalah keluaran yang berguna dan dibutuhkan.

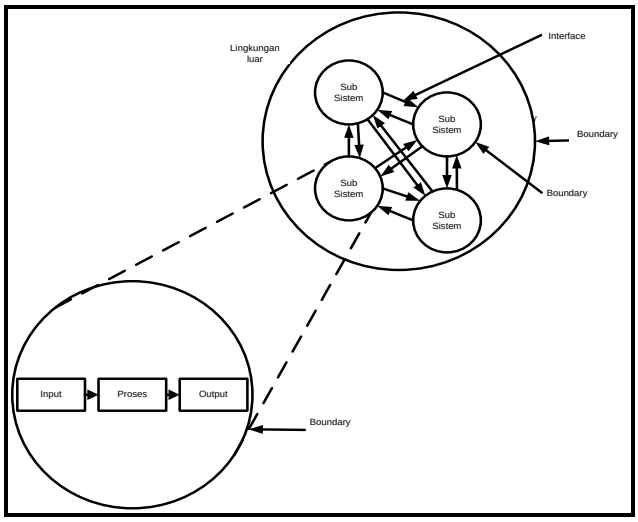
7. Pengolahan Sistem (*System Processing*)

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. Sistem akuntansi akan mengolah data-data transaksi menjadi laporan-laporan keuangan dan laporan-laporan yang dibutuhkan oleh manajemen.

8. Sasaran Sistem (*objective*) atau Tujuan (*goal*)

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

Karateristik suatu sistem dapat digambarkan seperti berikut:



Gambar 2.1 Karakteristik Sistem

[Sumber: Jogiyanto HM, 2005, *Analisis dan Desain Sistem Informasi*. Andi,Yogyakarta]

2.1.4. Klasifikasi Sistem

Menurut Jogiyanto (2005:3), Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantanya adalah sebagai berikut:

1. **Sistem Abstrak (*Abstract System*) dan Sistem Fisik (*physical system*).**

   Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem teologi. Adapun sistem fisik merupakan sistem yang ada secara fisik, misalnya sistem komputer.

2. **Sistem alamiah (*natural system*) dan sistem buatan manusia (*Human Made System*).**

   Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Misalnya sistem perputaran bumi. Adapun sistem buatan manusia adalah sistem yang dirancang oleh manusia. Misalnya, sistem buatan manusia yang melibatkan interaksi antara manusia dengan mesin yang disebut *human machine system.*

3. Sistem tertentu (*Deterministik System*) dan sistem tak tertentu (*Probabilistik System*).

Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Misalnya sistem komputer. Adapun sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat di prediksi karena mengandung unsur probabilitas.

4. Sistem tertutup (*Closed System*) dan Sistem terbuka (*Open System*).

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur dari pihak luarnya. Secara teoritis sistem tertutup ini ada, tetapi kenyataannnya tidak ada sistem yang sifatnya tertutup, yang ada hanyala *relatively closed system* (secara relatif tertutup, tidak benar - benar tertutup). Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya. Karena sistemnya terbuka dan terpengaruh lingkungan luarnya, maka suatu sistem harus mempunyai suatu sistem pengendalian yang baik.

2.2. Konsep Dasar Informasi

Sumber informasi adalah data. Data merupakan fakta atau kenyataan yang menggambarkan suatu kejadian-kejadian atau event nyata, kemudian dirumuskan ke dalam sekelompok simbol atau lambang-lambang yang teratur yang menunjukkan kualitas, tindakan atau hal-hal lain.

Konsep dasar informasi dapat dilihat dari pengertian informasi, kualitas informasi, serta nilai informasi yang secara lebih lengkap sebagai berikut :

2.2.1. Pengertian Informasi

Menurut Jogiyanto (2005 : 8) informasi adalah Data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

Menurut Davis dalam buku Pengenalan Sistem Informasi yang ditulis oleh Abdul Kadir (2002:31) informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang.

Dari kedua pengertian di atas dapat disimpulkan bahwa pengertian informasi adalah hasil dari pengolahan data yang memberi sesuatu yang berarti seperti pengetahuan dan bisa bermamfaat dalam pengambilan keputusan.

2.2.2. Kualitas Informasi

Kualitas dari suatu informasi (*quality of information*) diukur berdasarkan beberapa hal sebagai berikut:

1. **Akurat (*Accuracy*)**

Suatu informasi dapat berkualitas jika informasi bebas dari kesalahan–kesalahan dan tidak bias atau menyesatkan. Informasi harus akurat karena dari sumber informasi sampai penerima informasi kemungkinan banyak terjadi gangguan (*noise*) yang dapat merubah atau merusak informasi tersebut.

2. **Tepat waktu (*Timelines*)**

Informasi yang datang pada penerima tidak boleh terlambat. Sebab informasi yang sudah usang tidak punya nilai lagi.

3. **Relevan (*Relevancy*)**

Relevansi berarti informasi mempunyai manfaat untuk pemakainya. Informasi yang berkualitas akan mampu menunjukan relevansi.

4. Ekonomis (*Economy*)

Informasi yang dihasilkan mempunyai manfaat yang lebih besar dibandingkan dengan biaya mendapatkannya.

5. Efisiensi (*Efficiency*)

Informasi akan memiliki kualitas yang baik jika informasi tersebut memiliki efisiensi, yang berarti bahwa informasi tersebut tepat guna bagi pemakainya.

6. Dapat Dipercaya (Reliability)

Informasi berasal dari sumber yang dapat dipercaya. Sumber tersebut telah teruji tingkat kejujurannya.

2.2.3. Nilai Informasi

Nilai dari informasi (*value of information*) ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Informasi yang digunakan didalam suatu sistem informasi umumnya digunakan untuk beberapa kegunaan. Sebagian besar informasi tidak dapat ditaksir dari keuntungannya dengan satuan nilai uang, tetapi dapat ditaksir dengan nilai efektivitasnya.

2.3. Pengertian Sistem Informasi

Informasi merupakan hal yang sangat penting dalam pengambilan keputusan, permasalahannya adalah dimana informasi tersebut didapat. Informasi dapat diperoleh dari sistem informasi.

Robert A Leitch dan K. Roscoe Davis mendefinisikan 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. Jogianto (2005:11)

Menurut Hall dalam buku Pengenalan Sistem Informasi yang ditulis oleh Abdul Kadir (2003 : 11) mendefinisikan sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses, menjadi informasi, dan didistribusikan kepada pemakai.

Berdasarkan dua definisi diatas maka dapat didefinisikan bahwa sistem informasi adalah mencakup sejumlah komponen ada sesuatu yang diproses dan dimaksudkan untuk mencapai suatu tujuan tertentu.

2.4. Komponen Sistem Informasi

John Burch dan Gary Grudnitski dalam buku Analisis dan Desain Sistem Informasi yang ditulis oleh Jogianto (2005:12) mengemukakan bahwa sistem Informasi terdiri dari komponen-komponen yang disebutnya dengan istilah blok bangunan (*building block*), yaitu blok masukan (*input block*), blok model *(model block*), blok keluaran (*output block*), blok teknologi (*technology block*), blok basis data (*database block*), dan blok kendali (*controls block*).

Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya.

a. Blok masukan merupakan input, mewakili data yang masuk kedalam sistem Informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukan, yang dapat berupa dokumen-dokumen.
b. Blok keluaran adalah produk sistem informasi berupa keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta pemakai sistem.

c. Blok model, blok ini merupakan kombinasi prosedur dan logika yang akan memanipulasi data input dan data yang tersimpan didalam basis data dengan cara tertentu untuk menghasilkan keluaran yang diinginkan.
d. Blok teknologi, blok ini merupakan kotak alat dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan data mengakses data, menghasilkan data mengirimkan keluaran, dan membantu sistem secara keseluruhan. Teknologi ini terdiri dari tiga bagian yaitu, teknisi (*brainware*), perangkat lunak (*sofware*), dan perangkat keras((*hardware*).
e. Blok basis data, basis data (*database*), merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan didalam basis data untuk keperluan penyediaan informasi lebih lanjut.
f. Blok kendali, digunakan untuk menghindari kerusakan sistem informasi seperti bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, kegagalan sistem itu sendiri dan sebagainya.

2.5. Alat Bantu yang digunakan

Adapun alat-alat bantu yang digunakan dalam penelitian ini akan dijelaskan seperti di bawah ini :

2.5.1. Konsep Perancangan Berorientasi Objek

Teknologi objek menganalogikan sistem aplikasi seperti kehidupan nyata yang didominasi oleh objek. Didalam membangun sistem berorientasi objek akan

menjadi lebih baik apabila langkah awalnya didahului dengan proses analisis dan perancangan yang berorientasi obke. Tujuannya adalah mempermudah programmer didalam mendesain program dalam bentuk objek-objek dan hubungan antar objek tersebut untuk kemudian dimodelkan dalam sistem nyata. Suatu perusahaan software yaitu Rational Software, telah membentuk konsarium dengan berbagai organisasi untuk meresmikan pemakaian *Unified Modelling Language* (UML) sebagai bahasa standar dalam *Object Oriented Analysist Design* (OOAD).

2.5.1.1. Pengertian UML

Unified Modeling Language **(UML)** merupakan sistem arsitektur yang bekerja dalam OOAD (*Object-Oriented Analysis/Design*) dengan satu bahasa yang konsisten untuk menentukan, visualisasi, mengkontruksi, dan mendokumentasikan *artifact* (sepotong informasi yang digunakan atau dihasilkan dalam suatu proses rekayasa software, dapat berupa model, deskripsi, atau software) yang terdapat dalam sistem software.UML merupakan bahasa pemodelan yang paling sukses dari tiga metode OO yang telah ada sebelumnya, yaitu Booch, OMT (*Object Modeling Technique*), dan OOSE (*Object-Oriented Software Engineering*).

2.5.1.2. Tujuan UML

Tujuan UML yaitu diantaranya:

1. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.

2. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
3. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.
4. Untuk membuat suatu model, UML memiliki diagram grafis sebagai berikut:

   a) *Business Use Case model*

   b) *Activity Diagram*

   c) *Use Case model*

   d) *Behavior diagram : Sequence diagram*

   e) *Implementation diagram : Component diagram, Deployment diagram*

   f) *Generate Code*

Diagram diagram tersebut diberi nama berdasarkan sudut pandang yang berbeda-beda terhadap sistem dalam proses analisis atau rekayasa. Dibuatnya berbagai jenis diagram diatas karena :

1. Setiap sistem yang kompleks selalu paling baik jika didekati melalui himpunan berbagai sudut pandang yang kecil yang satu sama lain hampir saling bebas (*independent*). Sudut pandang tunggal senantiasa tidak mencukupi untuk melihat isi item yang besar dan kompleks.
2. Diagram yang berbeda-beda tersebut dapat menyatakan tingkatan yang berbeda-beda dalam proses rekayasa.
3. Diagram-diagram tersebut dibuat agar model yang dibuat semakin mendekati realitas.

Diagram-diagram ini ditambah dengan kemampuan dokumentasi perupakan *artifacts* utama UML. *Data-flow diagram* dan tipe diagram lain yang

tidak terdapat dalam UML tidak termasuk dalam paradigma *object-oriented*. *Activity diagram* dan *collaboration diagram* yang terdapat dalam UML menggantikan *data-flow diagram*. *Activity diagram* juga sangat bermanfaat untuk membuat *workflow*.

2.5.1.3. Diagram-Diagram Dalam UML

Adapun diagram – diagram dalam UML mempunyai beberapa diagram antara lain yaitu:

a. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng- *create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.*Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di- *include* akan dipanggil setiap kali *use case* yang meng- *include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat

dihindari dengan cara menarik keluar fungsionalitas yang *common* . Sebuah *use case* juga dapat meng- *extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

b. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan stereotype)

2. Atribut

3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. *Private,* tidak dapat dipanggil dari luar *class* yang bersangkutan

2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
3. *Public*, dapat dipanggil oleh siapa saja

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda.*Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class.* Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*. Sesuai dengan perkembangan *class* model, *class* dapat dikelompokkan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package*.

Adapun hubungan antar *class*, sebagai berikut :

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

c. Activity Diagram

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di- *trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state* , standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork*dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

d. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display* , dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men- *trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal.*Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class* .*Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message.

Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary, controller* dan *persistent entity.*

e. Collaboration Diagram

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram,* tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence*

number , di mana *message* dari level tertinggi memiliki nomor 1. Messages dari level yang sama memiliki prefiks yang sama.

g. Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code* , baik berisi *source code* maupun *binary code* , baik *library* maupun *executable* , baik yang muncul pada *compile time, link time* , maupun *run time* . Umumnya komponen terbentuk dari beberapa *class* dan/atau *package* , tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

h. Deployment Diagram

Deployment/physical diagram menggambarkan detail bagaimana komponen di- *deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal Sebuah *node* adalah server, *workstation* , atau piranti keras lain yang digunakan untuk men- *deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.

2.5.2. Pengertian StarUML

StarUML adalah sebuah proyek open source untuk mengembangkan cepat, fleksibel, extensible, featureful, dan bebas-tersedia UML / platform MDA berjalan

pada platform Win32.Tujuan dari proyek StarUML adalah untuk membangun sebuah alat pemodelan perangkat lunak dan juga platform yang menarik adalah pengganti alat UML komersial seperti Rational Rose.

2.6. Perangkat Lunak Pendukung

Perangkat lunak pada penelitian ini adalah Java Netbeans 7.0, Xampp, MySQL. Berikut ini adalah penjelasan dari kedua perangkat pendukung tersebut.

2.6.1. Java Netbeans IDE 7.0

The NetBeans IDE 7.0 adalah sebuah lingkungan pengembangan sebuah *tools* untuk programmer menulis, mengkompilasi, mencari kesalahan dan menyebarkan program. Netbeans IDE ditulis dalam Java namun dapat mendukung bahasa pemrograman lain. Terdapat banyak modul untuk memperluas Netbeans IDE. Netbeans IDE adalah sebuah produk bebas dengan tanpa batasan bagaimana digunakan. (dikutip dari : www.netbean.org)

NetBeans mengacu pada dua hal, yakni platform untuk pengembangan aplikasi desktop java, dan sebuah *Integrated Development Environment* (IDE) yang dibangun menggunakan platform NetBeans. Platform 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 powerful.

Lingkungan Pengembangan yang terintegrasi pada netbean IDE, memudahkan pengguna untuk membuat beragam aplikasi dengan mudah. Adapun komponen – komponen Neatbeans adalah sebagai berikut:

1. **Menu Bar**

   Letak *Menu Bar* pada Palette yaitu pada *Swing Menus*, untuk menggunakan *button* ini klik pada *button* tersebut. Fungsi daripada menu bar ini untuk menciptakan tab-tab menu yang nantinya dipergunakan untuk perintah menu yang diinginkan.

2. **Toolbar**

   ToolBar berfungsi sebagai tempat untuk menampilkan *tool-tool* yang bertujuan mempermudah user dalam mengoperasikan beberapa *tool-tool* yang mungkin sering digunakan.

3. **Project Explorer**

   Project Explorer akan digunakan untuk melihat file – file yang berupa *class* java yang terbentuk melalui Neatbeans.

4. **Panel Inspector**

   Panel inspector akan tampil apabila mengaktifkan dokumen yang mengandung *container* atau pemograman grafis (GUI). Pada panel ini menampilkan komponen yang digunakan oleh file yang bersangkutan, seperti *container*, komponen kontrol, komponen menu, komponen border dan lainnya.

5. Jendela Utama

Jendela utama adalah sebuah objek yang digunakan untuk menempatkan objek-objek yang dipilih dari kontrol komponen.

6. Kontrol Komponen

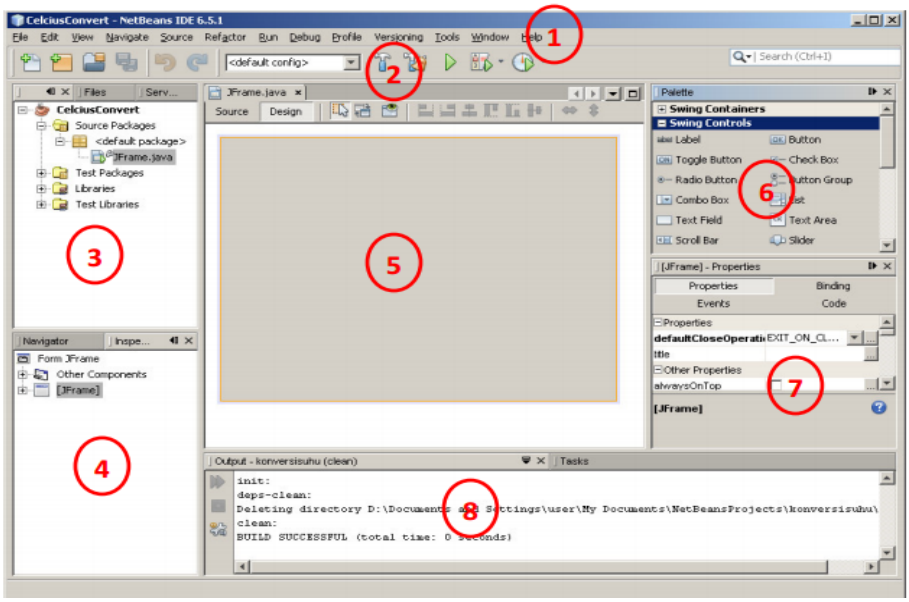
Kontrol komponen merupakan jendela yang menyediakan *tool-tool* untuk mendesign form berbasis grafis (GUI). *Tool* ini dibagi menjadi beberapa kategori, dimana setiap kategori menyediakan *tool* - *tool* GUI Builder sesuai dengan kategorinya. untuk menggunakannya, tinggal menyeret *tool-tool* kedalam area desain.

7. Jendela Properties

Jendela Properties berfungsi menampilkan properties komponen yang aktif untuk mengatur properties yang dimiliki oleh suatu komponen.

8. Jendela keluaran (*debugging*)

Terdiri dari jendela *Debugger* dan *Output*. Jendela *Debugger* mempunyai tab panel untuk menset *breakpoint*, memonitor *thread*, dan mengawasi nilai variabel-variabel. Jendela *Output* menampilkan pesan dari *debugger*. Jika ada file yang terbuka, *Debugging workspace* juga mengandung S*ource Editor.*



Gambar 2.2 Komponen Neatbeans

(Sumber: http://www.slideshare.net/kindiyanuar/dasar-dasar-netbeans/)

2.6.2. Xampp

XAMPP merupakan pengembangan dari LAMP (Linux Apache, MySQL, PHP and PERL), XAMPP ini merupakan *project non-profit* yang di kembangkan oleh Apache Friends yang didirikan Kai 'Oswalad' Seidler dan Kay Vogelgesang pada tahun 2002, project mereka ini bertujuan mempromosikan pengunaan Apache web server.

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* dan bebas, merupakan *web server* yang

mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Untuk mendapatkanya dapat mendownload langsung dari web resminya.

Mengenal bagian XAMPP yang biasa digunakan pada umumnya:

1. htdoc adalah folder tempat meletakkan berkas-berkas yang akan dijalankan, seperti berkas PHP, HTML dan skrip lain.
2. PhpMyAdmin merupakan bagian untuk mengelola basis data MySQL yang ada dikomputer. Untuk membukanya, buka browser lalu ketikkan alamat http://localhost/phpMyAdmin, maka akan muncul halaman phpMyAdmin.
3. Kontrol Panel yang berfungsi untuk mengelola layanan (*service)* XAMPP. Seperti menghentikan (*stop*) layanan, ataupun memulai (*start).*

2.6.3. MySQL

MySQL merupakan aplikasi *database server*. Perkembangannya disebut SQL yang merupakan kepanjangan dari *Structured Query Language*. SQL merupakan bahasa terstruktur yang khusus digunakan untuk mengolah *databas*e. SQL pertama kali didefinisikan oleh American National Standards Institute (ANSI) pada tahun 1986. MySQL adalah sebuah sistem manajemen *database* yang bersifat *open source*. MySQL adalah pasangan serasi dari PHP. MySQL dibuat dan dikembangkan oleh MySQL AB yang berada di Swedia. MySQL dapat digunakan untuk membuat dan mengola *database* beserta isinya. Kita dapat memanfaatkan MySQL untuk menambahkan, mengubah dan menghapus data yang berada dalam *database.*

MySQL merupakan sisitem manajemen *database* yang bersifat *at relational* artinya data-data yang dikelola dalam *database* akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan menjadi jauh lebih cepat. MySQL dapat digunakan untuk mengelola *database* mulai dari yang kecil sampai dengan yang sangat besar. MySQL juga dapat menjalankan perintah-perintah *Structured Query Language* (SQL) untuk mengelola *database-database* yang ada di dalamnya. Hingga kini, MySQL sudah berkembang hingga versi 5. MySQL 5 sudah mendukung trigger untuk memudahkan pengelolaan tabel dalam *database.*

2.7. Arsitektur Aplikasi

Arsitektur aplikasi meliputi pengertian jaringan komputer, tipe-tipe jaringan, jenis topologi jaringan, dan pengertian *client server*.

2.7.1. Pengertian Jaringan Komputer

Menurut Abdul Kadir (2003:346) yang dimaksud dengan jaringan komputer (*computer network*) adalah hubungan dua buah simpul (umumnya berupa komputer) atau lebih yang bertujuan utamanya adalah untuk melakukan pertukaran data.

2.7.2. Tipe-tipe Jaringan Komputer

Menurut Budhi Irawan (2005:19) terdapat (4) empat kategori utama jaringan komputer yaitu:

1. **LAN (*Local Area Network*)**

LAN digunakan untuk menghubungkan komputer yang berada di dalam suatu area kecil. Jarak antar komputer yang dihubungkannya mencapai 5 sampai 10 km, kecepatannya mulai 10 Mpbs sampai 100 Mbps.

2. **MAN (*Metropolitan Area Network*)**

MAN merupakan suatu jaringan yang cakupannya meliputi suatu kota. MAN menghubungkan LAN-LAN yang lokasinya berjauhan. Jangkauan MAN bisa mencapai 10 km sampai beberapa ratus km. Suatu MAN biasanya bekerja pada kecepatan 1,5 sampai 150 Mbps.

3. **WAN (*Wide Area Network*)**

WAN dirancang untuk menghubungkan komputer-komputer yang terletak pada suatu cakupan geografis yang luas, seperti hubungan dari satu kota ke kota lain di dalam suatu negara. Cakupan WAN bisa meliputi 100 km sampai 1.000 km, dan kecepatan antar kota bisa bervariasi anatara 1,5 Mbps sampai 2,4 Gbps. Dalam WAN, biaya untuk peralatan transmisi sangat tinggi, dan biasanya jaringan WAN dimiliki dan dioperasikan sebagai suatu jaringan publik.

4. **GAN (*Global Area Network*)**

GAN merupakan suatu jaringan yang menghubungkan negara-negara di seluruh dunia. Kecepatan GAN bervariasi mulai dari 1,5 Mbps sampai 100 Gbps dan cakupannya mencakapi ribuan kilometer. Contoh sangat baik dari GAN adalah *Internet*.

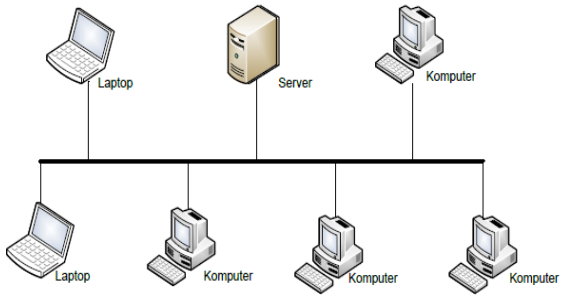
2.7.3. Topologi Jaringan Komputer

Menurut Budhi Irawan (2005:25). Topologi secara fisik dari suatu jaringan

lokal adalah merujuk kepada konfigurasi kabel, komputer, dan perangkat lainnya. Berikut adalah macam-macam topologi fisik yang digunakan di dalam jaringan lokal dintaranya :

1. ***Linear Bus* (Garis Lurus)**

Topologi *linear* bus terdiri dari satu jalur kabel utama dimana masing-masing ujungnya diberikan sebuah terminator. Semua nodes pada jaringan (*file server*, *workstation*, dan perangkat lainnya) terkoneksi sebuah kabel utama (*backbone*). Jaringan-jaringan *Ethernet* dan *local talk* menggunakan topologi *linear* ini.

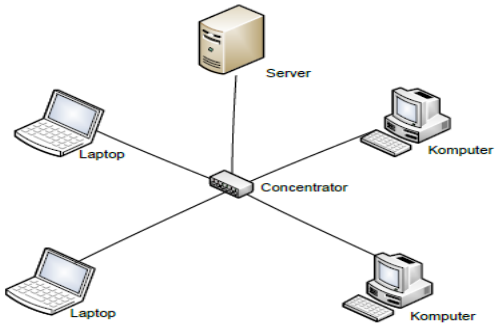


Gambar 2.3 Topologi Linear Bus

(Sumber: Budhi Irawan.2005. *Jaringan Komputer*. Graha Ilmu. Yogyakarta)

2. ***Star* (Bintang)**

Pada topologi *star*, setiap *nodes* (*file server*, *workstation*, dan perangkat lainnya) terkoneksi ke jaringan melalui sebuah *concentrator*.

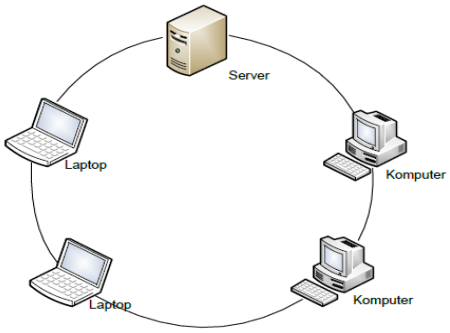


Gambar 2.4 Topologi *star*

(Sumber: Budhi Irawan.2005. *Jaringan Komputer*. Graha Ilmu. Yogyakarta)

3. *Ring* (Cincin)

Topologi *ring* menggunakan teknik konfigurasi yang sama dengan topologi *star* tetapi pada topologi ini terlihat bahwa jalur media transmisi menyerupai suatu lingkaran tertutup.



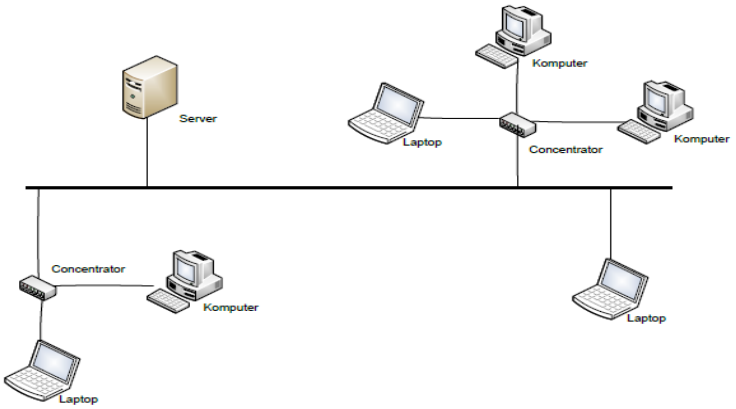
Gambar 2.5 Topologi *Ring*

(Sumber: Budhi Irawan.2005. *Jaringan Komputer*. Graha Ilmu. Yogyakarta)

4. *Tree* (Pohon)

Topologi model ini merupakan perpaduan antara topologi *linear bus* dan *star*, yang terdiri dari kelompok-kelompok dari *workstation* dengan konfigurasi *star* yang terkoneksi ke kabel utama yang menggunakan topologi *linear bus*. Topologi

ini memungkinkan untuk perkembangan jaringan yang telah ada dan memungkinkan untuk mengkonfigurasi jaringan sesuai dengan kebutuhan.



Gambar 2.6 Topologi *Tree*

(Sumber: Budhi Irawan.2005. *Jaringan Komputer*. Graha Ilmu. Yogyakarta)

2.7.4. Pengertian Client Server

Menurut Abdul Kadir (2003:81-82), *Client* adalah sembarang sistem atau proses yang melakukan sesuatu permintaan data atau layanan ke *server*. *Server* adalah sistem atau proses yang menyediakan data atau layanan yang diminta oleh *client*. Secara fisik, sebuah *server* dapat berupa komputer (*mainframe, mini computer, workstation,* ataupun PC) atau piranti yang lain (misalnya printer).

Client mempunyai kemampuan untuk melakukan proses sendiri. Ketika sebuah *client* meminta suatu data ke *server*, *server* akan segera menanggapinya dengan memberikan data yang diminta *client* bersangkutan. Setelah data diterima, *client* segera melakukan pemrosesan.

Keuntungan arsitektur *client server* menurut Abdul Kadir dalam bukunya yang berjudul "Pengenalan Sistem Informasi" adalah sebagai berikut:

Tabel 2.1 Keuntungan Client Server

Fitur	Keuntungan
Jaringan mesin-mesin yang kecil tetapi berdaya guna.	Jika sebuah mesin macet, bisnis tetap berjalan.
Kumpulan komputer dengan ribuan MIPS (*Million Instructions Per second*).	Sistem memberikan kekuatan dalam melaksanakan suatu tugas tanpa memonopoli sumber-sumber daya. Pemakai akhir diberi hak untuk bekerja secara lokal.
Beberapa *workstation* sangat handal seperti mainframe, tetapi dengan biaya 90% lebih rendah.	Dengan memberikan kekuatan yang lebih untuk biaya yang kecil, sistem menawarkan keluwesan untuk melakukan pembelian pada hal – hal lain atau untuk meningkatkan keuntungan.

Tabel 2.1. Keuntungan Client Server (Lanj)

Fitur	Keuntungan
Sistem terbuka.	Anda bisa memilih perangkat keras, perangkat lunak, dan layanan dari berbagai vendor.
Sistem tumbuh dengan mudah dan dapat diperluas secara tak terbatas.	Sangatlah mudah untuk memperbarui sistem anda saat kebutuhan anda berubah

Lingkungan operasi klien yang bersifat individual	Anda dapat mencampur dan mencocokan platform komputer yang sesuai dengan kebutuhan masing – masing departmen dan pemakai.

Kekurangan arsitektur *client server*:

A. Mahal
B. Membutuhkan investasi untuk *dedicated file server.*
C. Perbaikan (jaringan besar membutuhkan seorang staf untuk mengatur agar sistem berjalan secara efisien).
D. Ketergantungan
E. Ketika *server* jatuh, mengakibatkan keseluruhan operasi pada *network* akan jatuh pula.

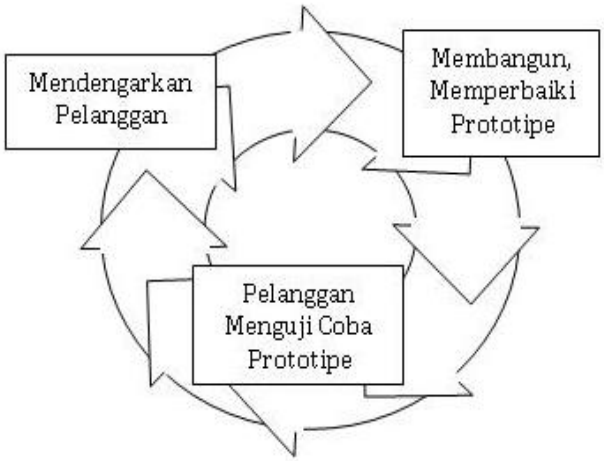
2.8. Metode Pengembangan Sistem

Metode Pengembangan adalah sebuah cara yang tersistem atau teratur yang bertujuan untuk melakukan analisa pengembangan suatu sistem agar sistem tersebut dapat memenuhi kebutuhan. Terdapat beberapa metode pengembangan sistem diantaranya metode *Prototype.*

2.8.1. Metode *Prototype*

Paradigma *prototype* dimulai dengan pengumpulan kebutuhan, pengembang dan pelanggan bertemu dan mendefinisikan obyektif keseluruhan dari perangkat lunak, mengidentifikasi segala kebutuhan yang diketahui, dan area garis besar dimana definisi lebih jauh merupakan keharusan kemudian dilakukan perancangan kilat. Roger S. Pressman (2002 : 40).

Terdapat 3 tahapan dalam metode *prototype* yaitu mendengarkan pelanggan, merancang dan membuat *prototype* sistem, kemudian melakukan uji coba dan evaluasi *prototype* yang telah dibuat.



Gambar 2.7. Paradigma *Prototype*

(Sumber : http://aldinobahtiar.wordpress.com/membangun-software-menggunakan-prototipe/)

1. **Mendengarkan Pelanggan**

Pada tahapan ini dilakukan pengumpulan kebutuhan dari sistem dengan cara mendengarkan keluhan dari pelanggan. Untuk membuat suatu sistem yang sesuai kebutuhan, maka harus diketahui terlebih dahulu bagaimana sistem yang sedang berjalan untuk kemudian mengetahui masalah yang terjadi.

2. **Merancang dan Membuat *Prototype***

Pada tahapan ini, dilakukan perancangan dan pembuatan *prototype* sistem. *Prototype* yang dibuat disesuaikan dengan kebutuhan sistem yang telah didefinisikan sebelumnya dari keluhan pelanggan atau pengguna.

3. Uji coba

Pada tahapan ini, *prototype* dari sistem di uji coba oleh pelanggan atau pengguna. Kemudian dilakukan evaluasi kekurangan-kekurangan dari kebutuhan pelanggan. Pengembang kemudian kembali mendengarkan keluhan dari pelanggan untuk memperbaiki *prototype* yang ada.

2.9. Metode Pendekatan Sistem

Terdapat dua metode pendekatan sistem yang saat ini sering digunakan yaitu dengan menggunakan metode pendekatan berorientasi objek (*object – oriented*) dan pendekatan terstruktur. Baik metode terstruktur ataupun metode berorientasi objek mempunyai fungsi yang sama yaitu membantu menganalisis dan merancang suatu sistem.

Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata. Ada banyak cara untuk mengabstraksikan dan memodelkan objek - objek tersebut, mulai dan abstraksi objek, kelas, hubungan antar kelas sampai abstraksi sistem.

Metode terstruktur lebih mengacu pada dokumen atau data yang berjalan dalam sistem.

Untuk pemilihan metode mana yang akan dipakai tergantung pada cara yang dilakukan dalam menganalisis, bila analisis lebih menekankan pada dokumen atau data maka lebih cocok menggunakan metode terstruktur dan bila analisis lebih

menekan pada objek dan kelas maka lebih cocok menggunakan pendekatan metode berorientasi objek.

2.10. Pengertian Kasus yang Dianalisis

Pada sub bab ini akan dijelaskan beberapa pengertian mengenai penjelasan dari kasus yang dianalisis :

2.10.1. Tentang Kenaikan Pangkat

Pangkat adalah kedudukan yang menunjukkan tingkatan seseorang Pegawai, berdasarkan jabatannya dalam rangkaian susunan kepegawaian dan digunakan sebagai dasar penggajian. Kenaikan pangkat adalah penghargaan yang diberikan atas prestasi kerja dan pengabdian Pegawai Luar Intansi, seperti Pegawai PMI terhadap Negara, serta sebagai dorongan kepada pegawai untuk lebih meningkatkan prestasi kerja dan pengabdiannya. Agar kenaikan pangkat dapat dirasakan sebagai penghargaan, maka kenaikan pangkat harus diberikan tepat pada waktunya dan tepat kepada orangnya. Susunan Pangkat dan Golongan Ruang Pegawai. Susunan pangkat serta golongan ruang Pegawai PMI sebagai berikut:

No,Pangkat,Golongan Ruang :

a. Juru Muda, Ia
b. Juru Muda Tingkat 1, Ib
c. Juru, Ic
d. Juru Tingkat 1, Id
e. Pengatur Muda, IIa

f. Pengatur Muda Tingkat 1, IIb
g. Pengatur, IIc
h. Pengatur Tingkat 1, IId
i. Penata Muda, IIIa
j. Penata Muda Tingkat 1, IIIb
k. Penata, IIIc
l. Penata Tingkat 1, IIId
m. Pembina, IVa
n. Pembina Tingkat 1, IVb
o. Pembina Utama Muda, IVc
p. Pembina Utama Madya, IVd
q. Pembina Utama, IVe

Setiap pegawai baru yang dilantik atau diputuskan sebagai Pegawai PMI baik di pusat maupun cabang akan diberikan Nomor Anggota Pegawai atau NAP yang berjumlah 5 digit angka, golongan dan pangkat sesuai dengan tingkat pendidikan yang diakui sebagai mana berikut di bawah ini :

a. Pegawai baru lulusan SD atau sederajat = I/a
b. Pegawai baru lulusan SMP atau sederajat = I/c
c. Pegawai baru lulusan SMA atau sederajat = II/a
d. Pegawai baru lulusan D1/D2 atau sederajat = II/b
e. Pegawai baru lulusan D3 atau sederajat = II/c
f. Pegawai baru lulusan S1 atau sederajat = III/a
g. Pegawai baru lulusan S2 sederajad/S1 Kedokteran/S1 Apoteker = III/b

h. Pegawai baru lulusan S3 atau sederajat = III/c

Periode kenaikan pangkat Pegawai PMI ditetapkan tanggal 1 April dan 1 Oktober setiap tahun, kecuali kenaikan pangkat anumerta dan kenaikan pangkat pengabdian. Masa kerja untuk kenaikan pangkat pertama Pegawai PMI dihitung sejak pengangkatan sebagai Calon Pegawai PMI. Kenaikan pangkat dilaksanakan berdasarkan sistem kenaikan pangkat regular (berkala) dan sistem kenaikan pangkat pilihan (golongan). Kenaikan pangkat berkala diberikan setelah 2 tahun pegawai menjadi pegawai tetap dan memenuhi persyaratan penilaian kepribadian >= cukup. Kenaikan pangkat golongan ini diberikan sekurang-kurangnya telah 4 tahun dalam pangkat terakhir dan pangkat tertingginya ditentukan oleh pendidikan tertinggi yang dimilikinya/pengajuan makalah. Kenaikan pangkat golongan juga diberikan kepada Pegawai PMI yang melaksanakan tugas belajar dan sebelumnya tidak menduduki jabatan struktural atau jabatan fungsional tertentu, dan dipekerjakan atau diperbantukan secara penuh di luar instansi induk dan tidak menduduki jabatan pimpinan yang telah ditetapkan persamaan eselonnya atau jabatan fungsional tertentu. Pegawai tidak bisa langsung melakukan kenaikan golongan sebelum ia telah melakukan kenaikan pangkat berkala terlebih dahulu.

Kenaikan pangkat golongan tertinggi diberikan kepada Pegawai PMI sampai dengan pangkat:

a. Pengatur Muda golongan ruang II/a, bagi yang memiliki Surat Tanda Tamat Belajar Sekolah Dasar.

b. Pengatur golongan ruang II/c, bagi yang memiliki Surat Tanda Tamat Belajar Sekolah Lanjutan Tingkat Pertama.
c. Pengatur Tingkat I golongan ruang II/d, bagi yang memiliki Surat Tanda Tamat Belajar Sekolah Lanjutan Kejuruan Tingkat Pertama.
d. Penata Muda Tingkat I golongan ruang III/b, bagi yang memiliki Surat Tanda Tamat Belajar Sekolah Lanjutan Tingkat Atas, Sekolah Lanjutan Kejuruan Tingkat Atas 3 Tahun, Sekolah Lanjutan Kejuruan Tingkat Atas 4 Tahun, Ijazah Diploma I, atau Ijazah Diploma II.
e. Penata golongan ruang III/c, bagi yang memiliki Ijazah Sekolah Guru Pendidikan Luar Biasa, Ijazah Diploma III, Ijazah Sarjana Muda, Ijazah Akademi atau Ijazah Bakaloreat.
f. Penata Tingkat I golongan ruang III/d, bagi yang memiliki Ijazah Sarjana (SI), atau Ijazah Diploma IV.
g. Pembina golongan ruang IV/a, bagi yang memiliki Ijazah Dokter, Ijazah Apoteker, Ijazah Magister (S2), atau ijazah lain yang setara
h. Diangkat menjadi Pejabat Negara;
i. Memperoleh surat tanda tamat belajar atau ijazah;
j. Melaksanakan tugas belajar dan sebelumnya menduduki jabatan struktural atau jabatan fungsional tertentu;
k. Telah selesai mengikuti dan lulus tugas belajar; dan
l. Dipekerjakan atau diperbantukan secara penuh di luar instansi induknya yang diangkat dalam jabatan pimpinan yang telah ditetapkan persamaan eselonnya atau jabatan fungsional tertentu.

Kenaikan pangkat golongan bagi Pegawai PMI yang menduduki jabatan struktural, jabatan fungsional tertentu, atau jabatan tertentu yang pengangkatannya ditetapkan dengan Keputusan Presiden, diberikan dalam batas jenjang pangkat yang ditentukan untuk jabatan yang bersangkutan. Kenaikan pangkat bagi Pegawai PMI yang menduduki jabatan struktural Pegawai PMI yang menduduki jabatan struktural dapat diberikan kenaikan pangkat pilihan apabila:

- Telah 4 tahun dalam pangkat terakhir.
- Daftar penilaian pelaksanaan pekerjaan setiap unsurnya sekurang-kurangnya bernilai baik dalam 2 tahun terakhir,
- Lulus ujian dinas bagi kenaikan pangkat yang akan pindah golongan, kecuali telah dibebaskan karena pendidikan/pendidikan dan pelatihan yang telah diikuti,
- Tidak akan melampaui pangkat atasannya,
- Belum mencapai pangkat tertinggi yang ditetapkan bagi jabatannya.

2.10.2. Tentang Gaji

Gaji merupakan salah satu unsur yang penting yang dapat mempengaruhi kinerja karyawan, sebab gaji adalah alat untuk memenuhi berbagai kebutuhan pegawai, sehingga dengan gaji yang diberikan pegawai akan termotivasi untuk bekerja lebih giat. Besarnya gaji di PMI dapat dilihat dari pangkat & golongannya, semakin tinggi pangkat & golongannya, maka semakin tinggi pula jumlah gaji yang diperoleh pegawai PMI tersebut. Komponen gaji bersih di PMI terdiri dari Gaji pokok & tunjangan-tunjangan lainnya, seperti tunjangan

istri/suami, anak, beras, sub. assuransi, khusus, struktural, kompensasi, dan insentif. Besarnya tunjungan istri/suami diperoleh dari 10% gaji pokok, dengan ketentuan bila suami/istrinya tidak bekerja di PMI. Besarnya tunjangan anak sebesar 2% dari gaji pokok dikali jumlah tanggungan anak, dengan maksimal 2 anak. Besarnya tunjangan beras yaitu jumlah tanggungan anak & istri/suami dikali harga beras perkilo/orang Rp. 6.000,00. Besarnya tunjangan sub. assuransi diperoleh dari 5% gaji pokoknya. Besarnya tunjangan khusus diperoleh dari 25% gaji pokoknya. Tunjangan Struktural hanya berlaku bagi yang menjabat sebagai kepala saja, dengan besar nominal, seperti tabel berikut :

Tabel 2.2. Tunjangan Struktural

No.	Jabatan	Besarnya nominal tunjangan
1.	Kepala Markas	Rp. 500.000,00
2.	Kepala Bagian/Bidang	Rp. 450.000,00
3.	Kepala Sub.bagian/seksi	Rp. 400.000,00
4.	Kepala Urusan	Rp. 300.000,00

Sumber : PMI

Kemudian, besarnya tunjangan kompensasi diperoleh karena pengecualian tertentu, tergantung kebijakan atasan, berlaku untuk yang menduduki jabatan kepala, besarnya tunjangan tergantung dari pegawai tersebut memakai fasilitas

kantor atau tidak & jumlah nominalnya perbulan bisa berbeda. Sedangkan tunjangan insentif diperoleh karena golongannya, seperti tabel berikut ini :

Tabel 2.3. Tunjangan Insentif

Pangkat/Golongan	Besarnya Tunjangan Insentif
IV/c	Rp. 210.000,00
IV/b	Rp. 195.000,00
IV/a	Rp. 180.000,00
III/d	Rp. 165.000,00
III/c	Rp. 150.000,00
III/b	Rp. 135.000,00
III/a	Rp. 120.000,00
II/d	Rp. 105.000,00
II/c	Rp. 90.000,00
II/b	Rp. 75.000,00
II/a	Rp. 60.000,00

Sumber : PMI