# BAB II
# LANDASAN TEORI

## 2.1 Pengertian Sistem

Menurut Abdul Kadir (2003 : 54) pada dasarnya, sistem adalah kumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Jika dalam sebuah sistem terdapat elemen yang tidak memberikan manfaat dalam mencapai tujuan yang sama, maka elemen tersebut dapat dipastikan bukanlah bagian dari sistem.

Menurut Jogiyanto (2008 : 34) Sistem adalah kumpulan dari komponen dan prosedur-prosedur yang saling berhubungan satu dengan yang lainnya membentuk satu kesatuan untuk mencapai tujuan tertentu.

Pengertian sistem yang menekankan pada prosedur didefinisikan oleh *L. ACKOF* Sistem adalah setiap kesatuan secara konseptual atau fisik yang terdiri dari bagian-bagian dalam keadaan saling tergantung satu sama lainnya.

Syarat-syarat sistem :

1. Sistem harus dibentuk untuk menyelesaikan masalah.
2. Elemen sistem harus mempunyai rencana yang ditetapkan.
3. Adanya hubungan diantara elemen sistem.
4. Unsur dasar dari proses (arus informasi, energi dan material) lebih penting dari pada elemen sistem.
5. Tujuan organisasi lebih penting dari pada tujuan elemen.

Sedangkan pengertian prosedur adalah :

" Suatu urutan-urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan apa (*what*) yang harus dikerjakan, siapa (*who*) yang mengerjakan, kapan (*when*) dikerjakan, dan bagaimana (*how*) mengerjakannya " (3,1).

### 2.1.1 Elemen Sistem

Menurut Ladjamudin (2005) 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 tidak terkendali

2. Masukan

   Masukan (*input*) sistem adalah segala sesuatu yang masuk kedalam sistem dan selanjutnya menjadi bahan untuk diproses. Masukan dapat berupa hal-hal berwujud( tampak secara fisik) maupun yang tidak tampak.

3. Keluaran

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

4. Proses

   Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna, misalnya berupa informasi dan produk, tetapi juga bisa berupa hal-hal yang tidak berguna, misalnya saja sisa pembuangan atau limbah.

5. Mekanisme Pengembalian Umpan Balik

Mekanisme pengendalian (control mechanism) diwujudkan dengan menggunakan umpan balik (feedback), yang mencuplik keluaran. Umpan balik ini digunakan untuk mengendalikan baik masukan ataupun proses.

### 2.1.2 Karakteristik Sistem

Menurut Jogiyanto (2005 : 3) Sistem mempunyai karakteristik yang tidak bisa dipisahkan antara satu karakteristik dengan karakteristik yang lain. beberapa karakteristik tersebut antara lain:

1. Komponen sistem

   Suatu sistem terdiri dari sejumlah komponen yang saling berintegrasi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponenkomponen sistem atau elemen-elemen sistem dapat berupa subsistem atau bagian-bagian dari sistem.

2. Batasan sistem

   Merupakan suatu daerah yang membatasi suatu sistem dengan sistem yang lain atau lingkungan luar, dengan batasan ini kita bisa mengetahui ruang lingkup suatu sistem.

3. Lingkungan luar sistem

   Apapun yang ada diluar batas sistem yang mempengaruhi oprasi suatu sistem.

4. Penghubung sistem

   Merupakan media penghubung antara satu sub sistem dengan sistem lainnya. Dengan penghubung ini akan mengalir data-data sub sistem dimana keluaran (output) dari sistem akan menjadi masukan (input)

untuk sub sistem yang lain, sehingga antara sistem dan sub sistem akan saling berinteraksi membentuk satu kesatuan.

5. Masukan sistem

   Merupakan energi yang dimasukan kedalam sistem, dimana masukan ini dapat berupa masukan perawatan (maintenance input) dan masukan sinyal (signal input).

6. Keluaran sistem

   Merupakan hasil dari energy yang diolah dan didefinisikan menjadi keluaran yang berguna dan mampu menjadikan masukan baru atau informasi yang dibutuhkan.

7. Pengolahan sistem

   Setiap sistem pasti mempunyai pengolahan data masukan untuk diolah menjadi sebuah informasi.

8. Sasaran sistem

   Merupakan penentu dari tujuan untuk menentukan masukan yang dibutuhkan dan keluar yang akan dihasilkan sebuah sistem.

### 2.1.3 Klasifikasi Sistem

Sistem diklasifikasikan berdasarkan jenisnya, diantaranya yaitu :

1) Sistem Abstrak : sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik (sistem teologia).
2) Sistem Fisik : merupakan sistem yang ada secara fisik (sistem komputer, sistem akutansi, sistem produksi, dll).
3) Sistem Alamiah ; sistem yang terjadi melalui proses alam. (sistem matahari, sistem luar angkasa, sistem reproduksi dll).

4) Sistem Buatan Manusia ; sistem yang dirancang oleh manusia, Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut human-machine sistem (contoh ; sistem informasi).
5) Sistem Tertentu (Deterministic sistem); beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan (contoh : sistem komputer).
6) Sistem Tak Tentu (probabilistic sistem) : sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.
7) Sistem Tertutup (close sistem) : sistem yang tidak berhubungan dan tidak terpengaruh dengan sistem luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur ada sistem yang benar-benar tertutup, yang ada hanyalah relatively closed sistem (secara relatif tertutup, tidak benar-benar tertutup).
8) Sistem Terbuka (open sistem) : sistem yang berhubungan dan terpangaruh dengan lingkungan luarnya.

## 2.2. Konsep dasar Informasi

### 2.2.1 Pengertian informasi

Menurut Azhar (2004: 40) Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang.

Menurut Jogiyanto (2005 : 8) Informasi diartikan sebagai data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

### 2.2.2 Kualitas Informasi

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

1. Relevan (*relevancy*), yaitu sejauh mana tingkat relevansi informasi tersebut terhadap kenyataan kejadian masa lalu, masa sekarang dan kejadian yang akan datang.
2. Akurat (*accuracy*), yaitu suatu informasi dikatakan berkualitas jika seluruh kebutuhan informasi telah tersampaiakan serta pesan yang disampaiakan sudah lengkap sesuai dengan yang diinginkan oleh user.
3. 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.
4. 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.

5. Efisien (*efficiency*), informasi yang berkualitas harus memiliki kalimat yang sederhana dan mudah dimengerti, tapi bisa memberikan makna yang mendalam.
6. Dapat dipercaya (*reliability*), informasi yang didapat harus dari sumber yang bisa dipercaya. Sumber tersebut juga harus sudah teruji tingkat kejujurannya.

## 2.3. Sistem Informasi

### 2.3.1. Pengertian Sistem Informasi

Menurut Jogiyanto (2005 : 11) yang dimaksud sistem informasi adalah suatu sistem 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.

Menurut Jogiyanto (2005:36), tujuan dari sistem informasi adalah menghasilkan informasi. Tugas dari sistem informasi adalah untuk melakukan siklus pengolahan data. Untuk melakukan siklus ini, maka sebagai suatu sistem diperlukan komponen-komponen tertentu.

Abdul Kadir (2003 : 70) juga menyebutkan dalam suatu sistem informasi terdapat komponen-komponen seperti :

1. Perangkat keras (*hardware*), yaitu mencakup pidanti-piranti fisik seperti komputer dan printer.
2. Perangkat lunak (*software*), yaitu sekumpulan instruksi yang memungkinkan perangkat keras untuk dapat memproses data.

3. Prosedur, yaitu sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.
4. Orang (*user*), yaitu pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan, dan penggunaan keluaran sistem informasi.
5. Basis data (*database*), yaitu sekumpulan table, hubungan, dan lain-lain yang berkaitan dengan penyimpanan data.

Jaringan komputer dan komunikasi data, yaitu sistem penghubung yang memungkinkan sesumber (*resources*) dipakai secara bersama atau diakses oleh sejumlah pemakai.

### 2.3.2 Komponen Sistem Informasi

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

1. Hardware dan software yang berfungsi sebagai mesin.
2. People dan procedures yang merupakan manusia dan tatacara menggunakan mesin.
3. Data merupakan jembatan penghubung antara manuia dan mesin agar terjadi suatu proses pengolahan data.

## 2.4 Pengenalan Object Oriented

Menurut Suhendar dan Hariman (2002:10) Object oriented merupakan paradigma baru dalam rekayasa software yang didasarkan pada objek dan kelas. Object oriented memandang software bagian per bagian, dan menggambarkan suatu bagian dalam satu objek. Satu objek dalam sebuah model merupakan suatu

fokus selama dalam proses analisis, desain dan implementasi dengan menekankan pada state, prilaku (behavior), dan interaksi objek-objek dalam model tersebut.

### 2.4.1 OOAD *(Object Oriented Analysis and Design)*

Object oriented analysis adalah metode analisis yang memeriksa keperluan requirements (syarat/keperluan yang harus dipenuhi suatu sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup permasalahan. Object oriented design adalah metode untuk mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem.

### 2.4.2 Konsep dasar dalam *Object Oriented Analysis and Design (*OOAD)

1. Abstraksi

   Abstraction adalah prinsip mengabaikan sejumlah aspek dari suatu objek yang tidak relevan dengan tujuan tertentu untuk lebih memfokuskan pada objek tersebut secara utuh.

2. Objek

   Objek (object) adalah benda, secara fisik ataupun konseptual. Hardware,software, dokumen dan manusia adalah beberapa contoh dari objek. Sebuah objek memiliki keadaan sesaat (state) dan prilaku (behavior). State adalah kondisi objek yang menggambarkan objek tersebut, sedangkan behaviour adalah suatu definisi tindakan dan reaksi suatu objek. Atribut

adalah nilai internal suatu objek yang mencerminkan antara lain karakteristik objek, kondisi sesaat, koneksi dengan objek lain, dan identitas.

Behavior suatu objek mendefinisikan bagaimana sebuah objek bertindak dan member reaksi. Behavior ditentukan oleh himpunan semua atau beberapa operasi yang dapat dilakukan dalam objek itu sendiri. Behavior dari suatu objek dicerminkan oleh interface, service, dan method dari objek tersebut.

3. Kelas

   Kelas (class) adalah definisi umum untuk himpunan objek sejenis. Kelas menetapkan spesifikasi prilaku (behavior) dan atribut objek-objek tersebut. Class adalah abstraksi dari entitas dalam dunia nyata. Objek adalah contoh (instance) dari sebuah kelas.

4. Pemodulan (encapsulation)

   Encapsulation adalah prinsip yang digunakan ketika membangun keseluruhan struktur program, setiap komponen dari program harus dibungkus atau tersembunyi dalam satu desain keputusan.

5. Penurunan (inheritance)

   Inheritance adalah mekanisme untuk mengekspresikan kesamaan diantara kelas, memudahkan definisi dari kelas yang sama.

6. Kebanyak Rupaan (polymorphism)

   Polymorphism menunjukan bahwa ada banyak objek yang berasal dari kelas yang berbeda dapat bereaksi pada pesan yang sama.

7. Association

Asosiasi adalah gabungan atau hubungan dari beberap ide. Merupakan landasan untuk, menentukan kompleksitas hubungan antara kelas objek.

## 2.5 UML (*Unified Modelling Language*)

### 2.5.1 Pengertian UML

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah system. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering).

Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi booch , metodologi coad , metodologi OOSE , metodologi OMT , metodologi shlaer-mellor , metodologi wirfs-brock , dsb. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita

bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan.

Dimulai pada bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan mempelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direlease draft pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh Object Management.

### 2.5.2 Diagram-Diagram dalam UML

1. *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.



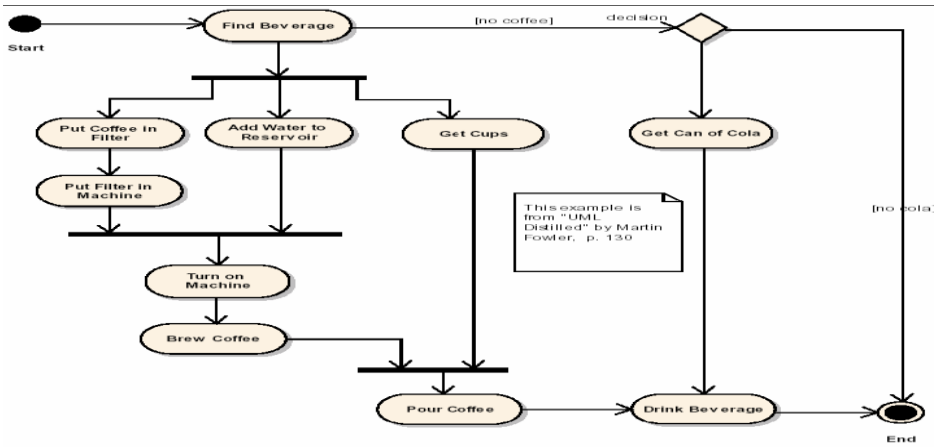

Gambar 2.1. *Use Case Diagram*

2. *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.




Gambar 2.2. *Activity Diagram*

3. *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*.

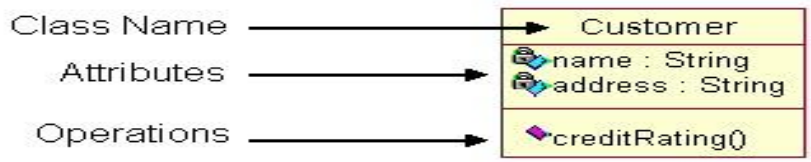
Hubungan Antar *Class* :

a. 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* menunjukkan arah *query* antar *class*.

b. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).

c. 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.

d. 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.

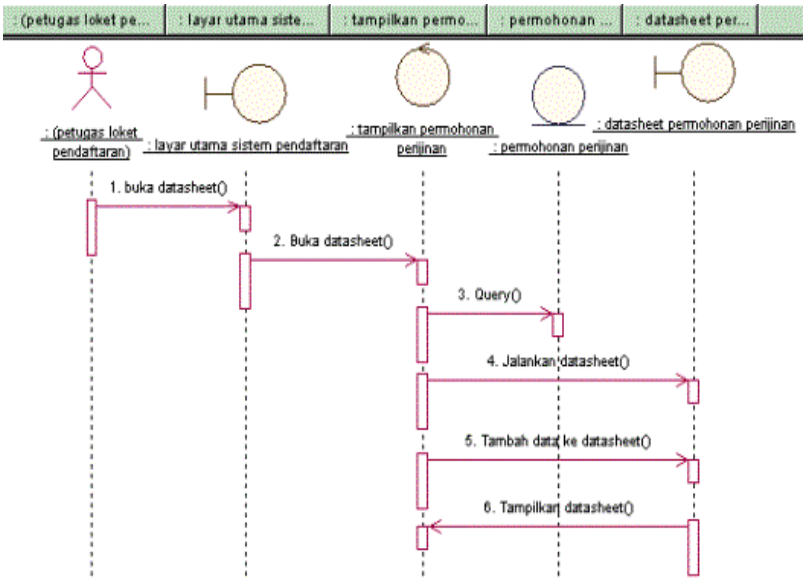



*Gambar 2.3. Bagian-bagian dari class*

4. *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

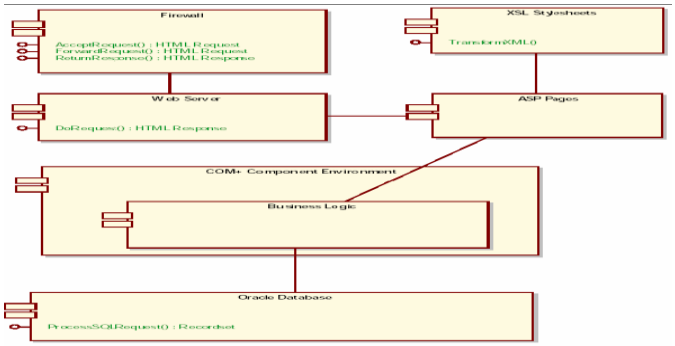
satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity*.




Gambar 2.4. *Sequence Diagram*

5. *Component Diagram*

Menggambarkan struktur fisik kode dari komponen. Komponen dapat berupa *source code*, komponen biner atau *executeable component*. Sebuah komponen berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view.*

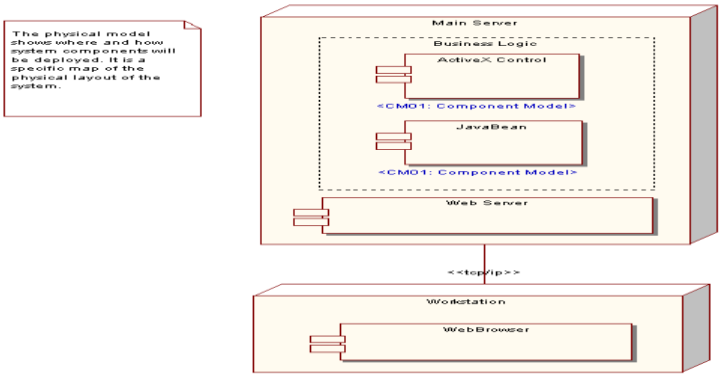



Gambar 2.5. *Component Diagram*

6. *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.




Gambar 2.6. *Deployment Diagram*

## 2.6. Tools Yang Mendukung UML

### 2.6.1. Rational Rose

Dalam penelitian ini, *Rational Rose* berfungsi sebagai *tool* untuk pemodelan sistem yang menggambarkan proses-proses yang ada pada sistem ini. *Rational Rose* adalah kakas (*tools)* pemodelan visual untuk pengembangan system 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 pengembang menulis kode-kode dalam bahasa pemrograman tertentu. Ia juga membantu analisis system dengan cara pengembang membuat diagram *use case* untuk melihat fungsionalitas sistem secara keseluruhan sesuai dengan harapan dan keinginan pengguna. Kemudian, ia juga menuntut pengembang untuk mengembangkan *Interaction* diagram untuk melihat bagaimana objek-objek saling bekerjasama dalam menyediakan fungsionalitas yang diperlukan (Adi Nugroho, 2005:20).

## 2.7. Definisi Permasalahan

### 2.7.1. Penjualan

Penjualan merupakan salah satu fungsi dari pemasaran atau merupakan bagian dari kegiatan pemasaran. Penjualan sangat penting dan menetukan karena suatu perusahaan untuk melakukan suatu penjualan yang baik harus mempunyai pemasaran yang baik pula.

Pengertian penjualan menurut Mulia Nasution.SE (2003) dari (http://www.itmaranatha.org/) adalah :

*"Pola hubungan antara bagian-bagian yang saling berkaitan untuk melakukan kegiatan yaitu memproses data penjualan sehingga dihasilkan data yang cepat, tepat, dan akurat yang memuaskan kedua belah pihak baik penjual maupun pembeli sehingga tujuan perusahaan tercapai."*

Jadi Sistem Informasi Penjualan adalah suatu sistem informasi yang mengorganisasikan serangkaian prosedur dan metode yang dirancang untuk menghasilkan, menganalisa, menyebarkan dan memperoleh informasi guna mendukung pengambilan keputusan mengenai penjualan.

### 2.7.2 Pemesanan

Pemesanan merupakan suatu proses kegiatan untuk mengajukan sesuatu permintaan kepada produsen penyedia barang atau jasa yang dilakukan olen individu atau kelompok untuk memenuhi kebutuhan akan barang atau jasa yang akhirnya akan meninbulkan suatu proses jual beli antara pihak konsumen dan produsen.

### 2.7.3 Reklame

"Reklame atau Periklanan (advertising) adalah bentuk-bentuk komunikasi atau persentasi non pribadi produk atau perusahaan yang di kendalikan oleh produsen untuk berkomunikasi dengan pelanggan".

Atas dasar tujuan yang hendak di capai dalam proses komunikasi, periklanan dapat di bedakan :

1. Periklanan infornatif yaitu promosi yang tujuannya memmbeberkan informasi dari produk perusahaan. Misalnya tentang deskripsi, kemampuan, cara kerja, keunggulan, kualitas, harga, saluran distribusi dan sebagainya. Umumnya di gunkan pada tahap perkenlan suatu produk baru. Periklanan informative berguna untuk membina permintaan primer yaitu permintaan terhadap suatu jenis barang tertentu.
2. Periklanan Persuasif yaitu periklanan yang tujuannya membujuk calon konsumen actual untuk melakukan pembelian atau pembelian ulang produk perusahaan. Tujuan periklanan persuasive adalahmenciptakan permintaan selektif, yaitu permintaan terhadap merk barang tertentu. Seringkali periklanan persuatif akan menjadi periklanan yang komperatif, yaitu periklanan yang membandingkan dengan produk pesaing. Perusahaan berusaha meyakinkan keunggulan produk perusahaan terhadap produk pesaing.
3. Periklanan yang mengingatkan kembali yaitu bentuk periklanan yang bertujuan untuk meningkatkan dan memelihara ingatan konsumen agar tetap setia terhadap produk perusahaan. Periklana ini sangat bermanfaat bagi produk yang berada apda tahap pendewasaannya. Yang di harapkan produsen adalah bahwa dalam bawah sadrnya, konsumen selalu teringat merk perusahaan untuk pemenuhan kebutuhannya sekarang maupun yang akan dating. Di samping itu, bentuk periklanan ini juga berusaha untuk memberikan keyakinan konsumen bahwa pemeliharaannya tetap.

## 2.8 Perangkat Lunak Pendukung

### 2.8.1 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 di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti Apache, di mana 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 di bawah lisensi GPL (General Public License). Di mana 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. Sebagai database server, MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase. Selain itu MySQL juga memiliki beberapa keistimewaan, antara lain :

1. Portability

   MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga dan masih banyak lagi.

2. Open Source

   MySQL didistribusikan secara open source (gratis), di bawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

3. Multiuser

   MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. Performance tuning

   MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. Column types

   MySQL memiliki tipe kolom yang sangat kompleks, seperti signed atau unsigned, integer, float, double, char, text, date, timestamp dan lain-lain.

6. Command dan functions

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam query.

7. Security

   MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host dan izin akses user dengan sistem perizinan yang mendetail serta password terenkripsi.

8. Scalability dan limits

   MySQL mampu menangani database dalam skala besar, dengan jumlah records lebih dari 50 juta dan 60 ribu tabel serta lima milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. Connectivity

   MySQL dapat melakukan koneksi dengan client menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).

10. Localisation

    MySQL dapat mendeteksi pesan kesalahan pada client dengan menggunakan lebih dari 20 bahasa. Meskipun demikian, bahasa Indonesia belum termasuk di dalamnya.

11. Interface

    MySQL memiliki interface (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).

12. Clients dan tools

    MySQL dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi database, dan pada setiap tool yang ada disertakan petunjuk online.

13. Struktur tabel

    MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani alter table, dibandingkan database lainnya semacam PostgreSQL ataupun Oracle.

### 2.8.2 PHP (*Hypertext Preprocessor*)

#### 2.8.2.1 Pengertian PHP

PHP (akronim dari PHP: Hypertext Preprocessor) adalah bahasa pemrograman yang berfungsi untuk membuat website dinamis maupun aplikasi web. Berbeda dengan HTML yang hanya bisa menampilkan konten statis, PHP bisa berinteraksi dengan database, file dan folder, sehingga membuat PHP bisa menampilkan konten yang dinamis dari sebuah website. Blog, Toko Online, CMS, Forum, dan Website Social Networking adalah contoh aplikasi web yang bisa dibuat oleh PHP. PHP adalah bahasa scripting, bukan bahasa tag-based seperti HTML. PHP termasuk bahasa yang cross-platform, ini artinya PHP bisa berjalan

pada sistem operasi yang berbeda-beda (Windows, Linux, ataupun Mac). Program PHP ditulis dalam file plain text (teks biasa) dan mempunyai akhiran ".php".

#### 2.8.2.2 Sejarah PHP

PHP ditulis (diciptakan) oleh Rasmus Lerdorf, seorang software engineer asal Greenland sekitar tahun 1995. Pada awalnya PHP digunakan Rasmus hanya sebagai pencatat jumlah pengunjung pada website pribadi beliau. Karena itu bahasa tersebut dinamakan Personal Home Page (PHP) Tools. Tetapi karena perkembangan nya yang cukup disukai oleh komunitas nya, maka beliau pun merilis bahasa PHP tersebut ke publik dengan lisensi open-source. Saat ini, PHP adalah server-side scripting yang paling banyak digunakan di website-website di seluruh dunia, dengan versi sudah mencapai versi 5 dan statistiknya terus bertambah (www.php.net/usage.php).

#### 2.8.2.3 Syarat Untuk Menjalankan PHP

Untuk dapat berjalan, PHP membutuhkan web server, yang bertugas untuk memproses file-file php dan mengirimkan hasil pemrosesan untuk ditampilkan di browser client. Oleh karena itu, PHP termasuk server-side scripting (script yang diproses di sisi server). Web server sendiri adalah software yang diinstall pada komputer lokal ataupun komputer lain yang berada di jaringan intranet / internet yang berfungsi untuk melayani permintaan-permintaan web dari client. Web server yang paling banyak digunakan saat ini untuk PHP adalah "Apache" (www.apache.org). Selain Apache, PHP juga memerlukan PHP binary (www.php.net) yang bisa dikonfigurasikan sebagai modul Apache atau pun

sebagai aplikasi CGI. Untuk media penyimpanan datanya (database server), PHP biasa menggunakan ”MySQL” (www.mysql.com).

Untuk menginstall dan mengkonfigurasi ketiga software tersebut (Apache, MySQL, PHP) agar dapat berjalan dan saling terhubung, memang cukup sulit. Maka dari itu dibuatlah paket software LAMP, XAMPP, MAMP, WAMP, dll yang tinggal kita install dalam satu kali installasi. Dalam satu kali installasi, sudah mencakup ketiga software tersebut dan sudah dikonfigurasi untuk keperluan lingkungan pengembangan aplikasi web. Sehingga, programmer web hanya tinggal menulis program PHP dan langsung menjalankan / mengetest program yang ditulis tersebut melalui web browser. Untuk mendapatkan paket software web server tersebut silakan download dari website yang bersangkutan (untuk XAMPP: www.apachefriends.org, dan untuk WampServer: www.wampserver.com/en/).

#### 2.8.2.4 Appache

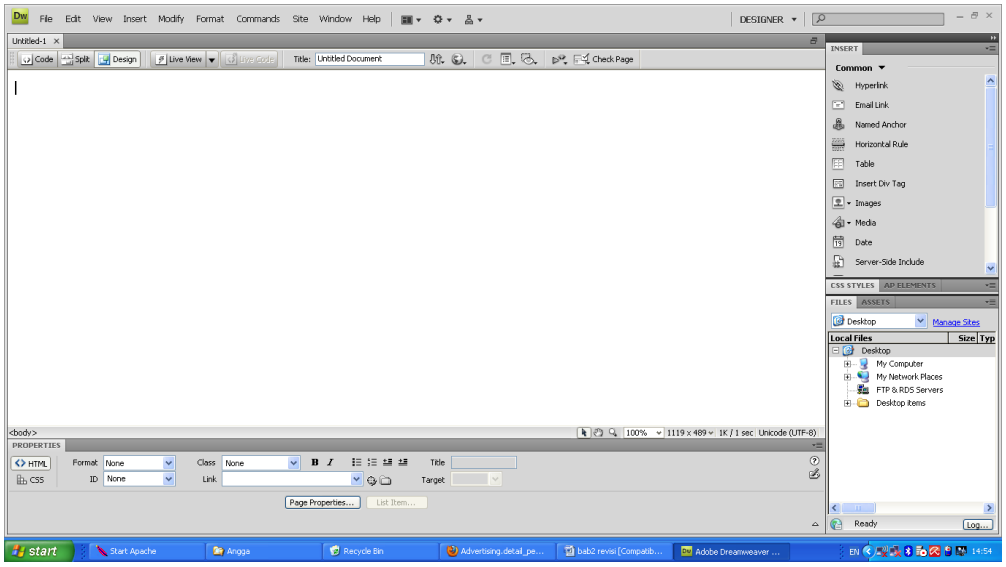
Apache menurut Bambang (2004) merupakan web server yang paling banyak di gunakan saat ini. Apache di gunakan karena factor kecepatan, kinerja yang stabil dan performansi. Apache sebagai web server mempunyai fungsi untuk melayani permintaan data adalam protocol http. Apache melayani permintaan data dalam bentuk atau format teks, gambar,suara dan animasi.

### 2.8.3 Adobe Dreamweaver CS 4

Menurut C. Widyo Hermawan, Sri Sulistiyani, Leo Agung dan Suci Nurasih (2009:1) Adobe Dreamweaver CS4 merupakan produk software Adobe

yang digunakan sebagai HTML editor professional untuk mendesain web secara visual dan dapat juga digunakan untuk mengelola situs atau halaman web. Selain itu Adobe Dreamweaver CS4 memberikan keleluasaan kepada anda untuk menggunakan sebagai media penulisan pemrograman web. Dalam perkembangannya Dreamweaver banyak digunakan para web desainer maupun web programmer. Fasilitas optimal dalam jendela design yang tersedia menjadikan program ini sebuah produk unggulan dalam memberikan kemudahan dalam mendesain web, tidak terkecuali bagi para web desainer pemula.

Kemampuan Dreamweaver untuk berinteraksi dengan beberapa bahasa pemrograman seperti : PHP, ASP, JavaScript, dan sebagainya, juga merupakan fasilitas pendukung maksimal kepada para desainer web yang menyertakan bahasa pemrograman web dalam pekerjaannya. Adobe Dreamweaver Creative Suite 4 (CS4), merupakan web komersial dikenal juga editor yang memungkinkan Anda untuk merancang, membangun dan mengelola website a kompleks. Editor adalah yang berarti bahwa Anda dapat membuat halaman web Anda secara visual dan apa pun yang Anda lihat pada layar saat desain adalah apa yang akan Anda dapatkan ketika situs Anda dimuat dalam web browser normal.




Gambar 2.7. Tampilan **Adobe Dreamweaver CS 4**

## 2.9. Pengenalan Jaringan Komputer

Berdasarkan kriterianya, jaringan komputer dibedakan menjadi 4 yaitu:

1. Berdasarkan distribusi sumber informasi/data.
   a. Jaringan terpusat

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

   b. Jaringan terdistribusi

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

2. Berdasarkan jangkauan geografis dibedakan menjadi:
   a. Jaringan LAN

Merupakan jaringan yang menghubungkan 2 komputer atau lebih dalam cakupan seperti laboratorium, kantor, serta dalam 1 warnet.

b. Jaringan MAN

Merupakan jaringan yang mencakup satu kota besar beserta daerah setempat. Contohnya jaringan telepon lokal, sistem telepon seluler, serta jaringan relay beberapa ISP internet.

c. Jaringan WAN

Merupakan jaringan dengan cakupan seluruh dunia. Contohnya jaringan PT. Telkom, PT. Indosat, serta jaringan GSM Seluler seperti Satelindo, Telkomsel, dan masih banyak lagi.

3. Berdasarkan peranan dan hubungan tiap komputer dalam memproses data.

a. Jaringan Client-Server

Pada jaringan ini terdapat 1 atau beberapa komputer server dan komputer client. Komputer yang akan menjadi komputer server maupun menjadi komputer client dan diubah-ubah melalui *software* jaringan pada protokolnya. Komputer client sebagai perantara untuk dapat mengakses data pada komputer server sedangkan komputer server menyediakan informasi yang diperlukan oleh komputer client.

b. Jaringan Peer-to-peer

Pada jaringan ini tidak ada komputer client maupun komputer server karena semua komputer dapat melakukan pengiriman maupun penerimaan informasi sehingga semua komputer berfungsi sebagai client sekaligus sebagai server.

4. Berdasarkan media transmisi data

   a. 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.

   b. Jaringan Nirkabel (*Wireless Network*)

      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.