

BAB II

TINJAUAN PUSTAKA

II.1 Pengertian Sistem Informasi

II.1.1. Pengertian Sistem

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, terkumpul bersama-sama untuk melakukan suatu kegiatan atau tujuan tertentu (Yakub : 2012 : 1).

II.1.2. Elemen-elemen Sistem

Ada beberapa elemen yang membentuk sebuah sistem yaitu tujuan, masukan, proses, keluaran, batas, mekanisme pengendalian dan umpan balik serta lingkungan.

1. Tujuan

Tujuan ini menjadi motivasi yang mengarahkan pada sistem, karena tanpa tujuan yang jelas sistem menjadi tak terarah dan tak terkendali.

2. Masukan (*input*) sistem

Masukan (*input*) sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk diproses. Masukan dapat berupa hal-hal berwujud maupun yang tidak berwujud. Masukan berwujud adalah bahan mentah, sedangkan yang tidak berwujud adalah informasi.

3. Proses

Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna dan lebih bernilai.

4. Keluaran (*output*)

Keluaran (*output*) merupakan hasil dari pemrosesan sistem dan keluaran dapat menjadi masukan untuk subsistem lain.

5. Batas (*boundary*) sistem

Batas (*boundary*) sistem adalah pemisah antara sistem dan daerah diluar sistem. Batas sistem menentukan konfigurasi, ruang lingkup atau kemampuan sistem

6. Mekanisme pengendalian dan umpan balik

Mekanisme pengendalian diwujudkan dengan menggunakan umpan balik (*feedback*), sedangkan umpan balik ini digunakan untuk mengendalikan masukan maupun proses. Tujuannya untuk mengatur agar sistem berjalan sesuai dengan tujuan.

7. Lingkungan

Lingkungan adalah segala sesuatu yang berada diluar sistem.

II.1.3. Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang. Klasifikasi sistem tersebut diantaranya yaitu sistem abstrak (*abstract system*), sistem fisik (*physical system*), sistem tertentu (*deterministic system*), sistem tak tentu (*probabilistic system*), sistem tertutup (*close system*) dan sistem terbuka (*open system*).

1. Sistem tak tentu (*probabilistic system*), adalah suatu sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

Sistem arisan merupakan contoh *probabilistic system* karena sistem arisan tidak dapat diprediksi dengan pasti.

2. Sistem abstrak (*abstract system*), adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sistem teologia yang berisi gagasan tentang hubungan manusia dengan Tuhan merupakan contoh *abstract system*.
3. Sistem fisik (*physical system*), adalah sistem yang ada secara fisik. Sistem komputer, sistem akuntansi, sistem produksi, sistem sekolah dan sistem transportasi merupakan contoh *physical system*.
4. Sistem tertentu (*determinic system*), adalah sistem yang beroperasi dengan tingkah laku yang dapat diprediksi, interaksi antara bagian dapat dideteksi dengan pasti sehingga keluarannya dapat diramalkan. Sistem komputer sudah diprogramkan merupakan contoh *determinic system* karena program komputer dapat diprediksi dengan pasti.
5. Sistem tertutup (*close system*), sistem yang tidak bertukar materi, informasi atau energi dengan lingkungan. Sistem ini tidak berinteraksi dan tidak dipengaruhi oleh lingkungan, misalnya : reaksi kimia dalam tabung yang terisolasi.
6. Sistem terbuka (*open system*), adalah sistem yang berhubungan dengan lingkungan dan dipengaruhi oleh lingkungan. Sistem perdagangan merupakan contoh *open system* karena dapat dipengaruhi oleh lingkungan.

II.2. Informasi

Informasi adalah data yang diolah menjadi bentuk lebih berguna dan lebih berarti bagi yang menerimanya. Informasi juga disebut data yang diproses atau data yang memiliki arti. Informasi merupakan data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan. Informasi dapat berupa data mentah, data tersusun, kapasitas sebuah saluran informasi, dan sebagainya (Yakub; 2012 : 8).

II.3. Konsep Dasar Sistem Informasi

Sistem Informasi merupakan kombinasi teratur dari orang-orang, perangkat keras, perangkat lunak, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi. Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan data transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi serta menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Sistem informasi juga dapat didefinisikan sebagai suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk menyajikan informasi. Sistem informasi merupakan sistem pembangkit informasi, kemudian dengan integrasi yang dimiliki antar subsistem, maka sistem informasi akan mampu menyediakan informasi yang berkualitas, tepat, cepat dan akurat sesuai dengan manajemen yang membutuhkannya. Sistem informasi juga merupakan suatu kumpulan dari komponen-komponen dalam organisasi yang berhubungan

denga proses penciptaan dan aliran informasi. Pada lingkungan berbasis komputer, sistem informasi menggunakan perangkat keras dan lunak komputer, jaringan telekomunikasi, manajemen basis data, dan berbagai bentuk teknologi informasi yang lain dengan tujuan untuk mengubah sumber data menjadi berbagai macam informasi yang dibutuhkan oleh pemakai (Yakub : 2012 : 17-18).

II.3.1. Sistem Informasi

Menurut Edhy Sutanta (2011:16) Sistem Informasi dapat dipahami sebagai sekumpulan sub sistem yang saling berhubungan, berkumpul bersama-sama dan membentuk satu kesatuan, saling berinteraksi dan bekerja sama antara bagian satu dengan yang lainnya dengan cara-cara tertentu untuk melakukan fungsi pengolahan data, menerima masukan (*input*) berupa data-data, kemudian mengolahnya (*processing*), dan menghasilkan keluaran (*output*) berupa informasi sebagai dasar bagi pengambilan keputusan yang berguna dan mempunyai nilai nyata yang dapat dirasakan akibatnya baik pada saat itu juga maupun dimasa mendatang, mendukung kegiatan operasional, manajerial, dan strategis organisasi, dengan memanfaatkan berbagai sumber daya yang ada dan tersedia bagi fungsi tersebut guna mencapai tujuan.

II.4. Pengertian Sistem Informasi Akuntansi (SIA)

Sistem Informasi Akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan. (Anastasia Diana, Lilis Setiawati : 2011 : 4-7)

SIA (Sistem Informasi Akuntansi) pada umumnya memiliki manfaat dan tujuan sebagai berikut :

1. Mengamankan harta ataupun kekayaan perusahaan. Harta dan kekayaan disini meliputi kas perusahaan, persediaan barang dagangan, termasuk aset tetap perusahaan.
2. Menghasilkan beragam informasi untuk pengambilan keputusan
3. menghasilkan informasi untuk pihak eksternal
4. Menghasilkan informasi untuk penilaian kinerja karyawan atau divisi
5. Menyediakan data masa lalu untuk kepentingan audit atau pemeriksaan
6. Menghasilkan informasi untuk penyusunan dan evaluasi anggaran perusahaan
7. menghasilkan informasi yang diperlukan dalam kegiatan perencanaan dan pengendalian.

II.5. Basis Data (Database)

Menurut Andri Kristanto (2008 : 79) manajemen sistem basis data atau data base manajemen system dapat didefinisikan sebagai berikut :

“Perangkat lunak yang didesain untuk membantu dalam hal pemeliharaan dan utilitas kumpulan data dalam jumlah besar“. DBMS dapat menjadi alternatif penggunaan secara khusus untuk aplikasi, misalnya penyimpanan data dalam file dan menulis kode aplikasi yang spesifik untuk pengaturannya.

Data dalam sebuah database disusun berdasarkan sistem hirarki yang unik, yaitu :

1. Database, merupakan kumpulan file yang saling terkait satu sama lain.
2. File, yaitu kumpulan dari record yang saling terkait dan memiliki format field yang sama dan sejenis.
3. Record, yaitu kumpulan field yang menggambarkan suatu unit dan individu tertentu.
4. Field, yaitu atribut dari record yang menunjukkan suatu item dari data seperti nama, alamat, dan lain sebagainya.
5. Byte, yaitu atribut dari field yang berupa huruf yang membentuk nilai dari sebuah field. Huruf tersebut dapat berupa numerik maupun abjad atau karakter khusus.
6. Bit, yaitu bagian terkecil dari data secara keseluruhan, yaitu berupa karakter nol atau satu yang merupakan komponen pembentuk byte.

II.6. MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris : *Database Management System*) atau DBMS yang multi *thread*, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi *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, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL adalah : *David Axmark*, *Allan Larsson*, dan *Michael "Monty" Widenius*

II.7. PHP

PHP merupakan singkatan dari "*Hypertext Preprocessor*" merupakan sebuah bahasa scripting server side, artinya dijalankan di server, kemudian outputnya dikirim ke client (browser).

PHP digunakan untuk membuat tampilan web menjadi lebih dinamis, dengan PHP bisa menampilkan atau menjalankan beberapa file dalam 1 file dengan cara di *include* atau *require*. PHP dapat berinteraksi dengan beberapa database walaupun dengan kelengkapan yang berbeda yaitu seperti : DBM, FilePro (Personic, Inc), Informix, Ingres, InterBase, Microsoft Access, MSSQL, MySQL, Oracle, PostgreSQL, dan Sybase (Madcom Madiun : 2013 : 309).

II.7.1. Adobe Dreamweaver Cs6

Adobe Dreamweaver Cs6 adalah perangkat lunak terkemuka untuk desain web yang menhedikan kemampuan visual *intuitif* termasuk pada tingkat kode, yang dapat digunakan untuk membuat dan mengedit website HTML serta aplikasi mobile seperti smartpone, tablet, dan perangkat lunak lainnya. Dengan adanya fitur layout *Fluid Grid* yang dirancang khusus untuk memungkinkan lintas platform, maka akan membuat layout menjadi *adaptif* atau dapat menyesuaikan

dengan browser yang dipakai. Anda juga dapat melihat hasil sementara desain dengan fasilitas *Multiscreen Preview* yang telah mengalami banyak perubahan dan peningkatan fungsionalitasnya (Madcom Madiun : 2013 : 1).

II.7.2. Aplikasi Web

Web server adalah software yang menjadi tulang belakang dari *world wide web* (www). Web server menunggu permintaan dari client yang menggunakan browser seperti Internet Explorer, Mozilla, dan program browser lainnya. Jika ada permintaan dari browser, maka web server akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke browser. Data ini mempunyai format yang standar, disebut dengan format SGML (*Standar General Markup Language*).

Data yang berupa format ini kemudian akan ditampilkan oleh browser sesuai dengan kemampuan Browser tersebut. Contohnya, bila data yang dikirim berupa gambar, browser yang hanya mampu menampilkan teks (misalnya lynx) tidak akan mampu menampilkan gambar tersebut, dan jika ada akan menampilkan alternatifnya. Web server, untuk berkomunikasi dengan client-nya (web browser) mempunyai protokol sendiri, yaitu HTTP (*Hypertext Transfer Protocol*) (Effendi, 2009)

II.7.3. World Wide Web (WWW)

World Wide web (www) adalah suatu file yang asal mulanya harus dibuat dengan bahasa HTML (*Hyper Text Markup Language*). Akan tetapi, pada awal

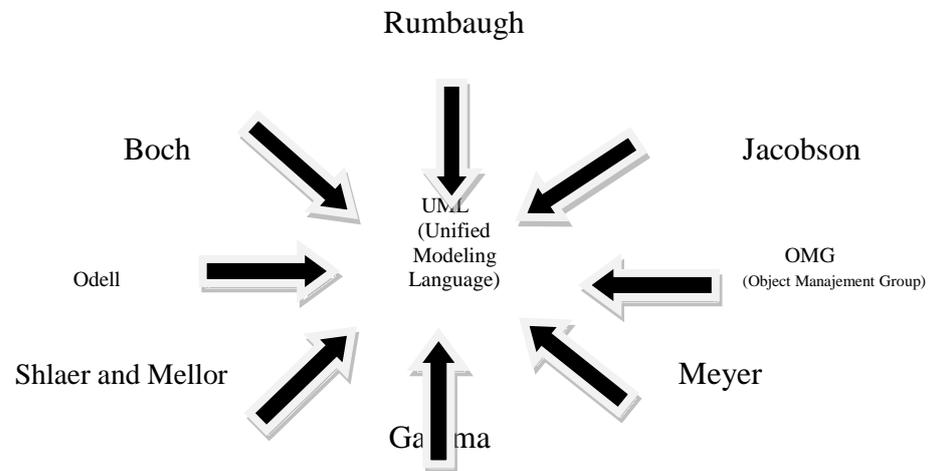
tahun 2002 ini sudah bisa dibuat berbagai macam perangkat lunak misalnya MS Word, FrontPage, Dreamweaver dan lain-lain. File WWW ini biasanya disebut situs web (web site) dan dapat diakses dari seluruh penjuru dunia manapun yang sudah terhubung ke Internet, File WWW mempunyai nama yang sudah diataur dengan standar baku Internasional (Eko Nugroho, 2008 : 49).

II.8. Model UML

UML(*Unified Modelling Language*) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

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), Jum Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering) (Sugiarti Yuni, S.T.M.Kom : 2013 : 34).

METODOLOGI PEMODELAN BERORIENTASI OBJECT



Gambar II.1 : Metodologi Pemodelan Berorientasi Object

Sumber : Yuni Sugiarti S.T.M.Kom (2013 : 35)

II.8.1. Digaram Dasar dalam UML

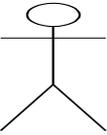
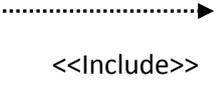
II.8.1.1. Use Case Diagram

Use Case Diagram atau diagram use case merupakan pemodelan untuk menggambarkan kelakuan (behavior) sistem yang akan dibuat. Diagram use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Dengan pengertian yang cepat, diagram use case digunakan untuk mengetahui fungsi apa saja yang ada dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Terdapat beberapa simbol dalam menggambarkan diagram use case, yaitu use case, aktor dan relasi. Hal yang perlu diingat mengenai diagram use case adalah diagram use bukan menggambarkan tampilan antar muka (user interface), arsitektur dari sistem, kebutuhan

nonfungsional, dan tujuan performansi (Yuni Sugiarti, S.T.M.Kom : 2013 : 41).

Simbol-simbol use case diagram dapat dilihat pada tabel II.1. sebagai berikut :

Tabel II.1. Simbol-Simbol Pada Use Case Diagram

Simbol	Deskripsi
Use Case	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case
 Actor	Orang, proses, atau sistem lain yang berinteraksi sebagai dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
Asosiasi / Association 	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor
 Extend	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu, miri dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan, arah panah menunjuk use case yang dituju.
 Include	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini. Ada dua sudut pandang yang cukup besar mengenai include di use case, include berarti use case yang ditambahkan akan selalu dipanggil saat use case tambahan dijalankan

Sumber : Yuni Sugiarti, S.T.M.Kom (2013 : 42)

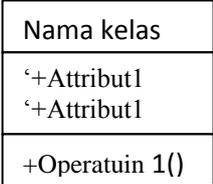
II.8.1.2. Class Diagram

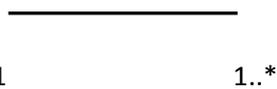
Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas, atribut mendeskripsikan properti dengan sebaris teks didalam kotak kelas tersebut. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Diagram kelas mendeskripsikan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat diantara mereka. Diagram juga menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.

Diagram kelas menggambarkan struktur dan deskripsi kelas, package, dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. Kelas memiliki tiga area pokok : nama, atribut dan operasi (Yuni Sugiarti, S.T.M.Kom : 2013 : 57).

Tabel II.2. Simbol-Simbol Pada Class Diagram

Simbol	Deskripsi
package 	Package merupakan sebuah bungkusan dari satu atau lebih kelas
Operasi 	Kelas pada struktur sistem

Antarmuka/interface 	Sama dengan konsep interface dalam pemrograman berorientasi objek
Asosiasi / Association 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah/directed asosiasi 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi spesialisasi (umum khusus)
Kebergantungan/Defedency 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi 	Relasi antar kelas dengan makna semua-bagian (whole-part)

Sumber : Yuni Sugiarti, S.T.M.Kom (2013 : 59)

II.8.1.3. Sequence Diagram

Diagram Sekuence menggambarkan kelakuan/perilaku objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekueces maka harus diketahui objek-objek yang terlibat dalam sebuah case beserta metode-metode yang dimiliki kelas yang di instansiasi menjadi objek itu.

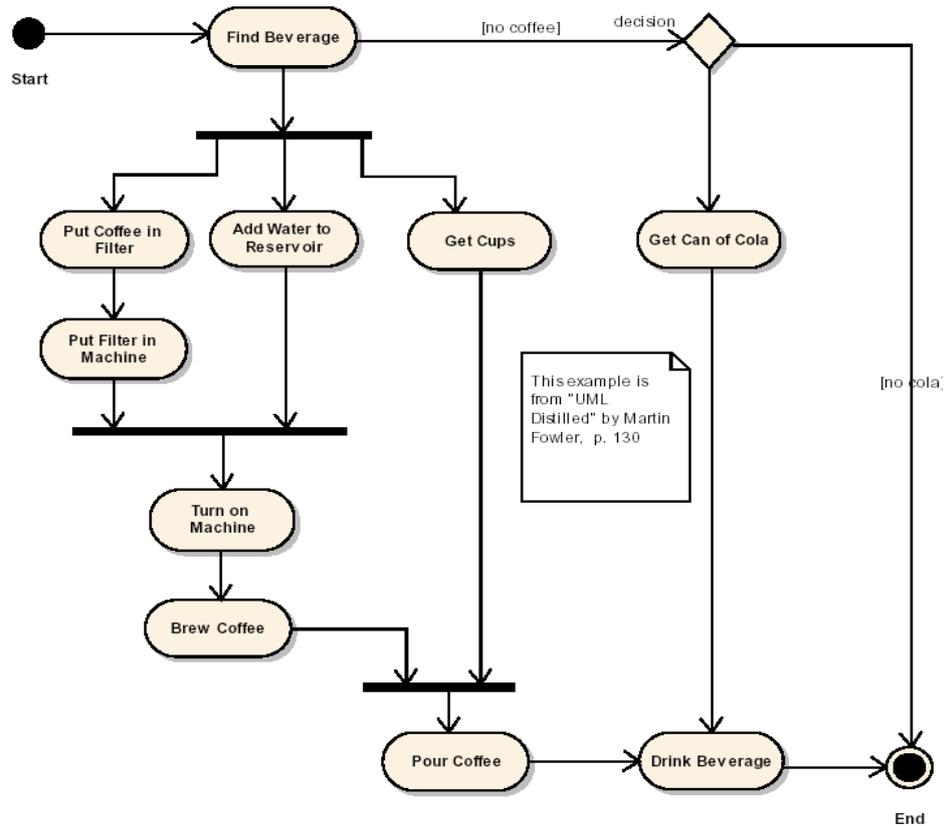
Diagram sekuece memiliki ciri yang berbeda dengan diagram interaksi pada diagram kolaborasi sebagai berikut :

1. Pada diagram sekuece terdapat garis objek
2. Terdapat pokus kendali (*Focus of Control*) , berupa empat persegi panjang ramping dan tinggi yang menampilkan aksi atau objek secara langsung atau sepanjang sub ordinat (Yuni Sugiarti, S.T.M.Kom : 2013 : 69-70).

II.8.1.4. Activity Diagram

Digram aktivitas atau *Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukakn soleh sistem. Diagram aktivitas mendukung perilaku paralel.

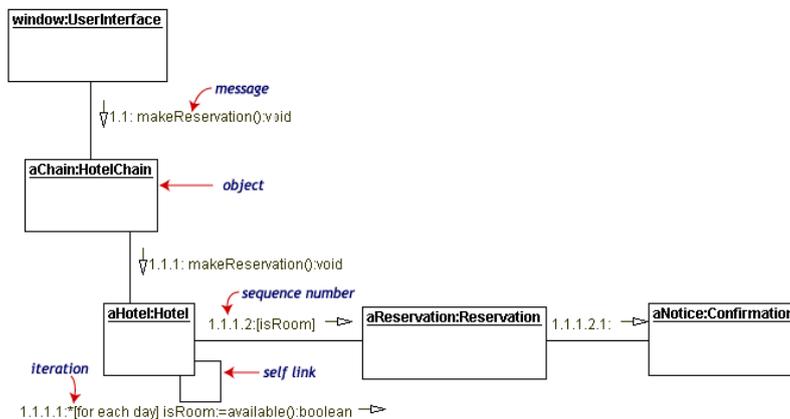
Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi (Yuni Sugiarti, S.T.M.Kom : 2013 : 75).



Gambar. II.2 : Activity Diagram Tanpa Swimlane
Sumber : Yuni Sugiarti, S.T.M.Kom (2013 : 76)

II.8.1.4.1. Collaboration Diagram

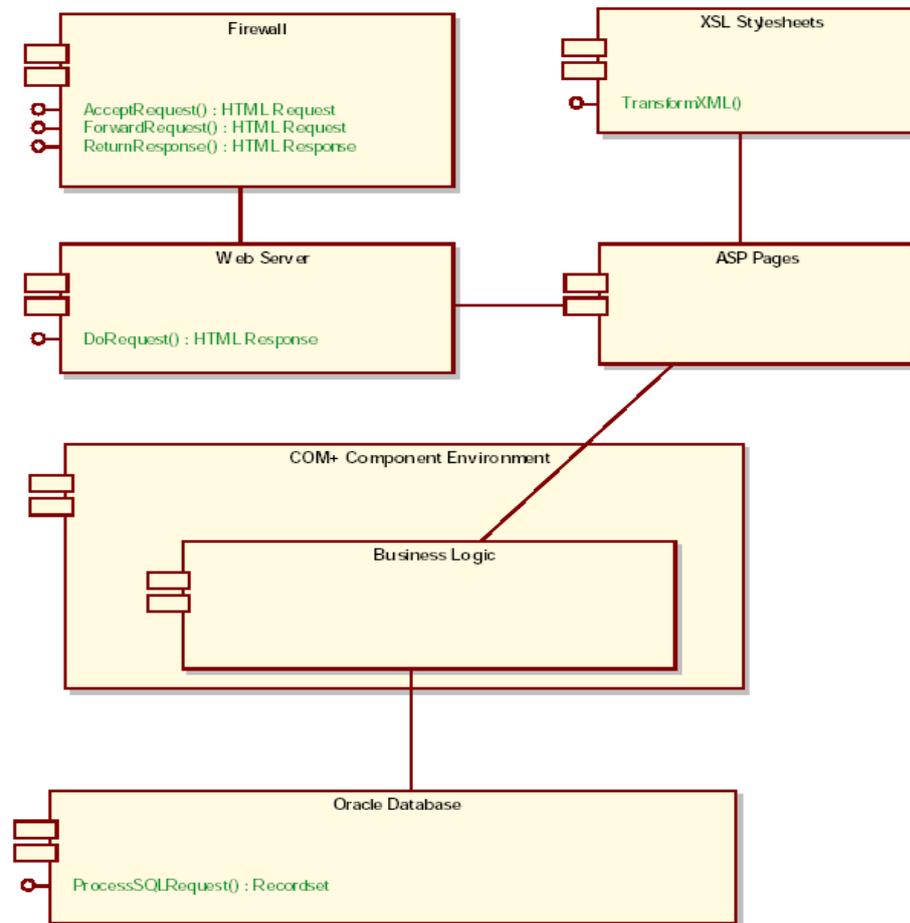
Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*.



Gambar. II.3 : Collaboration Diagram
Sumber : Yuni Sugiarti, S.T.M.Kom (2013 : 79)

II.8.1.4.2. Component Diagram

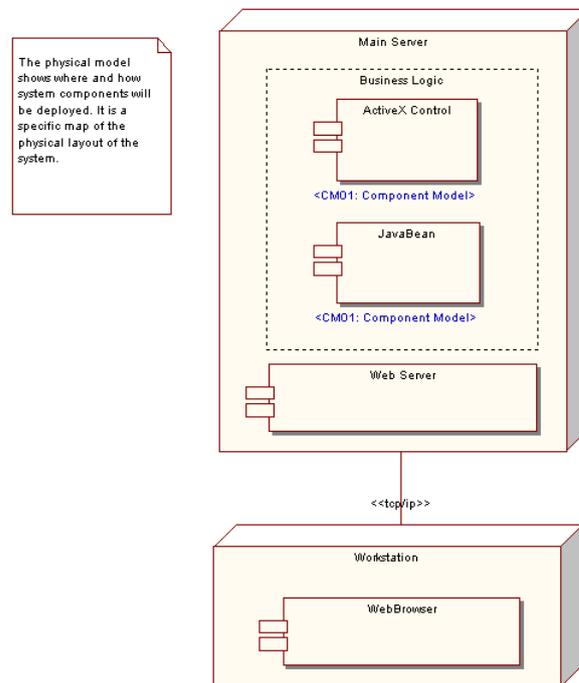
Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile 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.



Gambar. II.4 : Component Diagram
Sumber : Yuni Sugiarti, S.T.M.Kom (2013 : 80)

II.8.1.4.3. 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 fisik.



Gambar. II.5 : Deployment diagram
Sumber : Yuni Sugiarti, S.T.M.Kom (2013 : 81)

II.9. Pengertian Pajak Pertambahan Nilai (PPN)

Pajak pertambahan nilai barang dan jasa dipungut berdasarkan Undang-Undang No. 8 Tahun 1983 yang dikeluarkan pada tanggal 31 Desember 1983. Undang-Undang ini disebut juga Undang-Undang Pajak Pertambahan Nilai 1984 dan mulai berlaku efektif pada tanggal 1 April 1985. Pengunduran saat berlakunya Undang-Undang No. 8 Tahun 1983 atau Undang-Undang Pajak Pertambahan Nilai 1984 diatur dengan Peraturan Pemerintah Pengganti Undang-Undang No. 1 Tahun 1984. Undang-Undang No. 8 Tahun 1983 telah diubah dengan Undang-Undang No. 11 Tahun 1994 yang dikeluarkan pada tanggal 9 Nopember 1994 dan mulai berlaku sejak tanggal 8 Januari 1995 dan pada tanggal 2 Agustus 2000 telah

II.9.1. Mekanisme Penghitungan Pajak Pertambahan Nilai

Mekanisme penghitungan Pajak Pertambahan Nilai adalah sebagai berikut:

Beli = $400 \times \text{PPN } 10\% = 40$ (Pajak Masukan)

Jula = $750 \times \text{PPN } 10\% = 75$ (Pajak Keluaran)

Jumlah Pajak Keluaran = 75

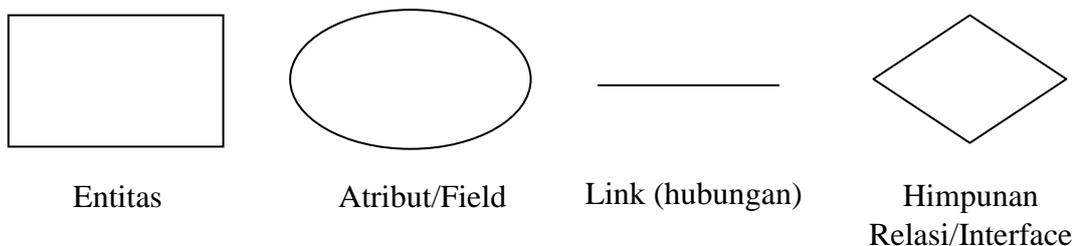
Jumlah Pajak Masukan = 40

PPN Masih harus dibayar = 35

PPN yang masih harus dibayar sebesar 35 adalah berasal dari 10 % dari jumlah pertambahan nilai sebesar 350. Jadi jelasnya, Pertambahan Nilai itu sama dengan Harga Jual dikurangi Harga beli atau harga perolehan suatu barang (Suprihatiningsih : 2012 : 103).

II.10. Entity Relationship Diagram (ERD)

Entity Relationship Diagram atau ERD merupakan salah satu alat (tool) berbentuk grafis yang populer untuk *desain database*. Tool ini relatif lebih mudah dibandingkan dengan Normalisasi. Kebanyakan sistem analis memakai alat ini, tetapi yang jadi masalah, kalau kita cermati secara seksama, tool ini mencapai 2NF (Ir. Yuniar Supardi : 2010 : 448).



Gambar. II.6 : Bentuk Simbol ERD
(Sumber : Ir. Yuniar Supardi (2010 : 448))

II.11. Teknik Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional.

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang telah ditemukan.

II.11.1. Bentuk-bentuk Normalisasi

1. Bentuk normal tahap pertama (1nd Normal Form)

Contoh yang kita gunakan di sini adalah sebuah perusahaan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri.

2. Bentuk normal tahap kedua (2nd normal form)

Definisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada seluruh kolom yang membentuk kunci utama.

3. Bentuk normal tahap ketiga (3rd normal form)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara definisi, sebuah tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya.

4. Boyce Code Normal Form (BCNF)

Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3NF sudah cukup karena sangat jarang entitas yang berada pada 3NF bukan merupakan 4NF dan 5NF.

5. Bentuk Normal Tahap Keempat dan Kelima

Sebuah tabel relasional berada pada bentuk normal keempat (4NF) jika dia dalam BCNF dan semua ketergantungan multivalued merupakan ketergantungan fungsional. Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan multivalued (MVD). Sebuah tabel berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join dependence*) (Janner Simarmata : 2010 : 76).

II.12. Kamus Data

Dalam tahapan ini unsur-unsur data yang diperlukan ditentukan untuk selanjutnya dijelaskan lagi dikamus data (*data dictionary*). Pengertian kamus data itu sendiri adalah suatu ensiklopedik dari informasi yang berkaitan dengan data perusahaan atau dapat juga kita katakana bahwa komputer (*computer-based catalog or directory*) yang berisi data perubahan (*metadata*) yang berkenaan dengan tahapan penjelasan data ini adalah sistem kamus data (*data dictionary system/DDS*) dan bahasa pendeskripsi data (*data description language/DDL*).

Sistem kamus data berbentuk perangkat lunak yang fungsinya adalah penciptaan dan pemerilahaan serta penyediaan kamus data agar dapat digunakan. Kamus data dapat berbentuk kertas ataupun arsip (*file*) komputer. DDS dapat kita peroleh dalam paket perangkat lunak terpisah ataupun dalam bentuk modul seperti yang ada dalam DBMS (*database management system*) dan CASE (teknik perangkat lunak tambahan komputer / *computer aided software engineering*) (Ian Soomerville : 2010 : 344).