

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Teori sistem melahirkan konsep-konsep futuristik, antara lain yang terkenal adalah konsep sibernetika (*cybernetics*). Konsep atau dibidang kajian ilmiah ini berkaitan dengan upaya menerapkan berbagai ilmu yaitu ilmu perilaku, fisika, biologi, dan teknik. Oleh karena itu sibernetika biasanya berkaitan dengan usaha-usaha otomasi tugas-tugas yang dilakukan manusia, sehingga melahirkan studi-studi tentang robotika, kecerdasan buatan (*artificial intelegence*). Unsur-unsur yang mewakili suatu sistem secara umum adalah masukan (*input*), pengolahan (*processing*), dan keluaran (*output*). Di samping itu, suatu sistem senantiasa tidak lepas dari lingkungan sekitarnya, maka umpan balik (*feed-back*) dapat berasal dari *output*, juga bisa berasal dari lingkungan sistem yang dimaksud (Tata Sutabri, S. Kom, MM; 2005 : 2).

Dengan demikian defenisi ini akan mempunyai peranan yang penting di dalam pendekatan untuk mempelajari suatu sistem. Pendekatan sistem yang merupakan kumpulan elemen atau subsistem merupakan defenisi yang lebih luas. Defenisi ini lebih banyak diterima karena kenyataannya suatu sistem dapat terdiri dari beberapa subsistem atau sistem-sistem bagian. Sebagai contoh, sisem akutansi, penjualan, subsistem akutansi pembelian, subsistem akutansi penggajian, subsistem akutansi transaksi biaya, dan lain-lainya.

Pendekatan sistem yang menekankan komponen akan memudahkan mempelajari suatu sistem untuk tujuan analisis dan perancangan sistem. Suatu sistem mempunyai suatu sistem tertentu. Ada yang menyebutkan suatu sistem adalah untuk mempunyai tujuan (*goal*) dan ada yang menyebutkan untuk mencapai suatu sasaran (*objectives*) tujuan biasanya dihubungkan dengan ruang lingkup yang lebih luas dan sasaran dalam ruang lingkup yang lebih sempit. Bila merupakan suatu sistem yang utama, misalnya sistem bisnis, maka istilah *goal* lebih tepat diterapkan. Untuk sistem akuntansi atau sistem-sistem yang lain, yang merupakan bagian atau subsistem dari sistem bisnis, istilah *objectives* lebih tepat. Jadi tergantung dari ruang lingkup dari mana memandang sistem tersebut (Tata Sutabri, S. Kom, MM; 2005 : 10-11).

II.1.1. Karakteristik Sistem

Model umum sebuah sistem adalah input, proses, dan output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana sebab sebuah sistem dapat mempunyai beberapa masukan dan keluaran. Selain itu sebuah sistem memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut (Tata Sutabri, S. Kom, MM; 2005 :11) :

II.1.1.1. Komponen Sistem (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling berkerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem

secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar, yang disebut “supra sistem (Tata Sutabri, S. Kom, MM; 2005 :11).

II.1.1.2. Batas Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan (Tata Sutabri, S. Kom, MM; 2005 :11).

II.1.1.3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada di luar lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut operasi lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan yang menguntungkan merupakan bagi sistem tersebut. Dengan demikian, lingkungan luar tersebut harus dijaga dan dipelihara. Lingkungan luar yang merugikan harus dikendalikan. Kalau tidak maka akan mengganggu kelangsungan hidup sistem tersebut (Tata Sutabri, S. Kom, MM; 2005 :11).

II.1.1.4. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem lainnya disebut penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem lain. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut. Dengan demikian dapat terjadi suatu integrasi sistem untuk membentuk satu kesatuan (Tata Sutabri, S. Kom, MM; 2005 :12).

II.1.1.5. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Contoh di dalam suatu sistem unit komputer. “program” adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan “data” adalah *signal input* untuk diolah menjadi informasi (Tata Sutabri, S. Kom, MM; 2005 :12).

II.1.1.6. Keluaran Sistem (*Output*)

Yaitu hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain. Contoh, sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai masukan untuk pengambil keputusan atau hal-hal lain yang menjadi *input* bagi subsistem lain (Tata Sutabri, S. Kom, MM; 2005 :12).

II.1.1.7. Pengolah Sistem (*Proses*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Contoh, sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen (Tata Sutabri, S. Kom, MM; 2005 :12).

II.1.1.8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat *deterministik*. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan (Tata Sutabri, S. Kom, MM; 2005 :12).

II.1.2. Klasifikasi Sistem

Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda setiap kasus yang terjadi yang ada di dalam sistem tersebut. Oleh karena itu sistem dapat diklasifikasikan dari beberapa sudut pandangannya antara lain (Tata Sutabri, S. Kom, MM; 2005 :13).

II.1.2.1. Sistem Abstrak Dan Sistem Fisik

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem teologia, yaitu sistem yang berupa pemikiran hubungan antara manusia dengan tuhan, sedangkan sistem fisik merupakan sistem yang ada secara fisik, misalnya sistem komputer, sistem produksi, sistem penjualan, sistem administrasi personalia, dan lain sebagainya (Tata Sutabri, S. Kom, MM; 2005 :13).

II.1.2.2. Sistem Alamiah Dan Sistem Buatan Manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi, terjadinya siang malam, pergantian musim. Sedangkan sistem buatan manusia dengan mesin, merupakan melibatkan interaksi manusia dengan mesin, yang disebut "*human machine system*". Sistem informasi berbasis komputer merupakan contoh *human machine system* karena menyangkut penggunaan komputer yang berinteraksi dengan manusia (Tata Sutabri, S. Kom, MM; 2005 :13).

II.1.2.3. Sistem Deterministik Dan Sistem Probabilistik

Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem deterministic. Sistem komputer adalah contoh dari sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan. Sedangkan sistem bersifat probabilistik adalah sistem yang mana kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilistik (Tata Sutabri, S. Kom, MM; 2005 :13).

II.1.2.4. Sistem Terbuka Dan Sistem Tertutup

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa campur tangan pihak luar. Sedangkan sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya (Tata Sutabri, S. Kom, MM; 2005 :13).

II.1.3. Daur Hidup Sistem

Siklus hidup sistem (*system life cycle*) adalah merupakan proses evolusioner yang diikuti dalam menerapkan sistem atau subsistem informasi komputer. Siklus hidup sistem terdiri dari serangkaian tugas yang erat mengikuti langkah-langkah pendekatan sistem sistem karena tugas-tugas tersebut mengikuti pola yang teratur dan dilakukan secara *top down*. Siklus hidup sistem sering disebut sebagai pendekatan air terjun (*waterfall approach*) bagi pembangunan dan pengembangan sistem.

Pembangunan sistem hanyalah salah satu dari rangkaian daur hidup suatu sistem. Meskipun demikian, proses ini merupakan aspek yang sangat penting. Kita akan melihat beberapa fase/tahapan dari daur hidup suatu sistem (Tata Sutabri, S. Kom, MM; 2005 :14).

II.1.3.1. Mengenali Adanya Kebutuhan

Sebelum segala sesuatunya terjadi, timbul suatu kebutuhan atau problema yang harus dapat dikenali sebagaimana adanya. Kebutuhan dapat terjadi sebagai hasil perkembangan dari organisasi dan volume yang meningkat melebihi kapasitas dari sistem yang ada. Semua kebutuhan ini harus dapat didefinisikan dengan jelas. Tanpa adanya kejelasan dari kebutuhan yang ada, pembangunan sistem akan kehilangan arah dan efektifitasnya (Tata Sutabri, S. Kom, MM; 2005 :14).

II.1.3.2. Pembangunan Sistem

Suatu proses atau seperangkat prosedur yang harus diikuti untuk menganalisis kebutuhan yang timbul dan membangun suatu sistem untuk dapat memenuhi kebutuhan tersebut (Tata Sutabri, S. Kom, MM; 2005 :14).

II.1.3.3. Pemasangan Sistem

Setelah tahap pembangunan sistem selesai. Sistem kemudian akan dioperasikan. Pemasangan sistem merupakan tahap yang penting pula dalam daur hidup sistem. Peralihan dari tahap pembangunan menuju tahap operasional terjadi pemasangan sistem yang sebenarnya, yang akan merupakan langkah akhir dari suatu pembangunan sistem (Tata Sutabri, S. Kom, MM; 2005 :14).

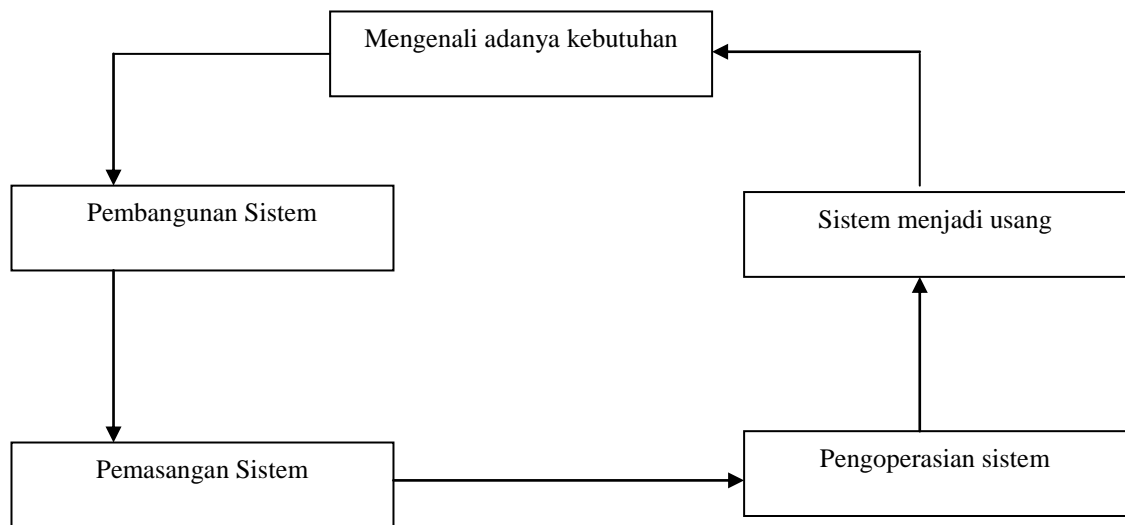
II.1.3.4. Pengoperasian Sistem

Program-program komputer dan prosedur-prosedur pengoperasian yang membentuk suatu sistem informasi semuanya bersifat statis. Sedangkan organisasi ditunjang oleh sistem informasi tadi. Ia selalu mengalami perubahan-perubahan itu karena pertumbuhan kegiatan bisnis, perubahan pengaturan, dan kebijaksanaan ataupun kemajuan teknologi. Untuk mengatasi perubahan-perubahan tersebut, sistem harus diperbaiki atau diperbaharui (Tata Sutabri, S. Kom, MM; 2005 :14).

II.1.3.5. Sistem Menjadi Usang

Kadang perubahan yang terjadi begitu drastis, sehingga tidak dapat diatasi hanya dengan melakukan perbaikan-perbaikan pada sistem yang berjalan. Tibalah saatnya secara ekonomis dan teknis sistem yang ada sudah tidak layak lagi untuk dioperasikan dan sistem yang baru perlu dibangun untuk menggantikannya(Tata Sutabri, S. Kom, MM; 2005 :15).

Sistem informasi kemudian akan melanjutkan daur hidupnya. Sistem dibangun untuk memenuhi kebutuhan yang muncul. Sistem beradaptasi terhadap perubahan-perubahan lingkungannya dinamis. Sampailah pada kondisi dimana sistem tersebut tidak dapat lagi beradaptasi dengan perubahan-perubahan yang ada atau secara ekonomis tidak layak lagi untuk dioperasikan. Sistem yang baru kemudian dibangun untuk menggantikannya. Untuk dapat menggambarkan daur hidup sistem ini, lihat pada gambar II.1. sebagai berikut (Tata Sutabri, S. Kom, MM; 2005 :15).



Gambar II.1. Daur Hidup Sistem

Sumber : Tata Sutabri, S.Kom, MM (2005 : 15)

II.2. Informasi

Informasi adalah data yang telah diklasifikasikan diolah atau diinterpretasi untuk digunakan dalam proses pengambil keputusan. Sistem pengolahan informasi mengolah data menjadi informasi atau tepatnya mengolah dari bentuk tak berguna menjadi berguna bagi penerimanya (Tata Sutabri, S. Kom, MM; 2005 :23).

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya (Prof. Dr. Jogiyanto HM, MBA, Akt; 2005 : 8).

II.3. Sistem Informasi

Sistem informasi adalah berupa suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan data transaksi harian yang mendukung

operasi yang bersifat manajerial dengan kegiatan strategi suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan (Tata Sutabri, S. Kom, MM ; 2005 :42).

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 (Prof. Dr. Jogiyanto HM, MBA, Akt; 2005 : 11).

II.3.1. Komponen Dan Jenis Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan (*building block*), yang terdiri dari blok masukan, blok model, blok keluaran, blok teknologi, blok basis data dan blok kendali. Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lain membentuk satu kesatuan untuk mencapai sasaran (Tata Sutabri, S. Kom, MM; 2005 :42).

II.3.1.1. Blok Masukan (*Input Block*)

Input mewakili data yang masuk ke dalam sistem informasi. *Input* di sini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar (Tata Sutabri, S. Kom, MM; 2005 :42).

II.3.1.2. Blok Model (*Model Block*)

Blok ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara

yang sudah tertentu untuk menghasilkan keluaran yang diinginkan (Tata Sutabri, S. Kom, MM; 2005 :42).

II.3.1.3. Blok Keluaran (*Output Block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta pemakai sistem (Tata Sutabri, S. Kom, MM; 2005 :42).

II.3.1.4. Blok Teknologi (*Technology Block*)

Teknologi merupakan “*tool box*” dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem keseluruhan. Teknologi sistem terdiri dari 3 (tiga) bagian yaitu teknisi teknologi (*brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*) (Tata Sutabri, S. Kom, MM; 2005 :42).

II.3.1.5. Blok Basis Data (*Database Block*)

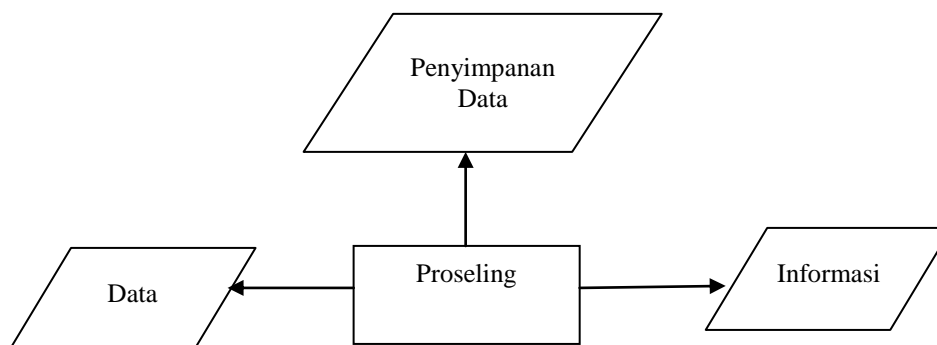
Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan lainnya, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan di basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi menggunakan perangkat lunak paket yang disebut DBMS (*database management system*) (Tata Sutabri, S. Kom, MM; 2005 :42).

II.3.1.6. Blok Kendali (*Control Block*)

Banyak hal yang dapat merusak sistem informasi, seperti bencana alam, api, temperature, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, ketidak efisienan, sabotase, dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah atau bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi (Tata Sutabri, S. Kom, MM; 2005 :43).

II.4. Data

Mengenai pengertian data, lebih jelas apa yang didefenisikan oleh Drs. Jhon J. Longkutoy dalam bukunya “ Pengenalan Komputer” sebagai berikut : isitilah data adalah suatu istilah majemuk yang berarti fakta atau bagian dari fakta yang mengandung arti yang dihubungkan dengan kenyataan simbol-simbol, gambar-gambar, angka-angka, huruf-huruf, atau simbol-simbol yang menunjukkan suatu ide, objek, kondisi, atau situasi dan lain-lain (Tata Sutabri, S. Kom, MM; 2005 : 16).



Gambar II.2. Pemrosesan Data

Sumber : Tata Sutabri, S. Kom, MM (2005 :16)

II.4.1. Pengolahan Data

Data merupakan bahan mentah untuk diolah, yang hasilnya kemudian menjadi informasi. Dengan kata lain, data yang telah diperoleh harus diukur dan dinilai baik buruknya, berguna atau tidak dalam hubungannya dengan tujuan yang akan dicapai. Pengolahan data terdiri dari kegiatan-kegiatan penyimpanan data dan penanganan data. Untuk lebih jelasnya akan diuraikan dibawah ini (Tata Sutabri, S. Kom, MM; 2005 :21).

II.4.1.1. Penyimpanan Data (*Data Storage*)

Penyimpanan data meliputi pekerjaan pengumpulan (*filig*), pencarian (*searching*), dan pemeliharaan (*maintenance*). Data disimpan dalam suatu tempat yang lazim dinamakan "*file*". *File* dapat berbentuk map, *ordner*, *disket*, *tape*, *hard disk*, dan lain sebagainya. Sebelum disimpan, suatu data diberi kode menurut jenis kepentingannya. Pengaturan dilakukan sedemikian rupa sehingga mudah mencarinya. Pengkodean memegang peranan penting. Kode yang salah data akan mengakibatkan data masuk ke dalam *file* juga salah, yang selanjutnya akan mengakibatkan kesulitan pencarian data tersebut apabila diperlukan. Jadi *file* diartikan suatu susunan data yang terbentuk dari sejumlah catatan (*record*) yang berhubungan satu sama lain (sejenis) mengenai suatu bidang dalam suatu unit usaha (Tata Sutabri, S. Kom, MM; 2005 :21).

Sistem yang umum dalam penyimpanan data (*filig*) ialah berdasarkan lembaga, perorangan, produksi, atau lain-lainya, tergantung dari sifat organisasi yang bersangkutan. Kadang-kadang dijumpai kesulitan apabila menghadapi suatu data dalam bentuk surat misalnya yang menyangkut ketiga klasifikasi tadi.

Metode yang terbaik adalah “referensi silang” (*cross reference*) antara *file* yang satu dengan *file* lain. Untuk memperoleh kemudahan dalam pencarian data (*searching*) di dalam *file*, maka *file* digai menjadi 2 (dua) jenis yaitu (Tata Sutabri, S. Kom, MM; 2005 :21) :

a. *File* Induk (*Master File*)

File induk ini berisi data-data permanen yang biasanya hanya dibentuk satu kali saja dan kemudian digunakan untuk pengolahan data selanjutnya.

Contoh : *file* kepegawaian, *file* gaji (Tata Sutabri, S. Kom, MM; 2005 :22)..

b. *File* Transaksi (*Detail File*)

File transaksi berisi data-data temporer untuk suatu periode untuk suatu bidang kegiatan atau suatu periode yang dihubungkan dengan suatu bidang kegiatan (Tata Sutabri, S. Kom, MM; 2005 :22).

Contoh : *file* lembur perminggu, *file* mutasi harian.

Pemeliharaan *file* (*file maintenance*) juga meliputi “peremajaan data” (*data updating*), yaitu kegiatan menambah catatan baru pada suatu data, mengadakan perbaikan, dan lain sebagainya. Misalnya, dalam hubungan dengan *file* kepegawaian, sudah tentu sebuah organisasi, entah itu perusahaan atau jawatan, akan menambah pegawainya. Sementara itu, ada pula pegawai yang pensiun atau berhenti bekerja atau putus hubungan dengan organisasi. Dengan demikian, data mengenai pegawai yang bersangkutan akan dikeluarkan dari *file* tersebut. Tidak jarang pula harus dilakukan perubahan data terhadap data seorang pegawai, misalnya kenaikan pangkat, kenaikan gaji berkala, menikah, pindah alamat, dan lain sebagainya (Tata Sutabri, S. Kom, MM; 2005 :22).

II.4.1.2. Penanganan Data (*Data Handling*)

Penanganan data meliputi berbagai kegiatan, seperti pemeriksaan (*verifying*), perbandingan (*comparing*), pemilihan (*sorting*), peringkasan (*extracting*), dan penggunaan (*manipulating*). Pemeriksaan data mencakup pengecekan data yang muncul pada berbagai daftar yang berkaitan atau yang datang dari berbagai sumber, untuk mengetahui berbagai sumber dan perbedaan atau ketidaksesuaian. Pemeriksaan ini dilakukan dengan kegiatan pemeliharaan *file* (*file maintenance*) (Tata Sutabri, S. Kom, MM; 2005 :22).

Pemilihan atau *sorting* dalam rangka kegiatan penanganan data mencakup pengaturan ke dalam suatu urutan yang teratur, misalnya daftar pegawai menurut pangkatnya, dari pangkat yang tertinggi sampai yang terendah atau daftar pelanggan dengan menyusun namanya menurut abjad dan lain sebagainya. Peringkasan merupakan kegiatan lain dalam penanganan data. Ini mencakup keterangan pilihan, misalnya daftar pegawai yang telah mengabdikan dirinya kepada organisasi/perusahaan lebih dari 10 tahun atau daftar yang memesan beberapa hasil produksi sekaligus dan lain-lain(Tata Sutabri, S. Kom, MM; 2005 :22).

Penggunaan data atau informasi "*data manipulation*" merupakan kegiatan untuk menghasilkan informasi. Kegiatan ini meliputi kompilasi tabel-tabel, statistik, ramalan mengenai perkembangan, dan lain sebagainya. Tujuan manipulasi ini adalah menyajikan informasi yang memadai mengenai apa yang terjadi pada waktu lampau guna menunjang manajemen, terutama membantu menyelidiki alternatif kegiatan mendatang. Jadi, hasil pengolahan data itu merupakan data

untuk disimpan bagi penggunaan di waktu yang akan datang, yakni informasi yang akan disampaikan kepada yang memerlukan atau mengambil keputusan mengenai suatu hal (Tata Sutabri, S. Kom, MM; 2005 :23).

II.5. Sistem Informasi Akuntansi

American Accounting Association mendefinisikan akuntansi sebagai proses mengidentifikasi, mengukur, dan melaporkan informasi ekonomi, untuk memungkinkan adanya penilaian dan keputusan yang jelas dan tegas bagi mereka yang menggunakan informasi tersebut. Definisi ini mengandung dua pengertian, yakni :

a. Kegiatan akuntansi

Bahwa akuntansi merupakan proses yang terdiri dari identifikasi, pengukuran dan pelaporan informasi ekonomi.

b. Kegunaan Akuntansi

Bahwa informasi ekonomi yang dihasilkan oleh akuntansi diharapkan berguna dalam penilaian dan pengambilan keputusan mengenai kesatuan usaha yang bersangkutan (Soemarso S.R; 2002 : 3).

Akuntansi merupakan bahasa dari bisnis. Setiap perusahaan menerapkannya sebagai alat komunikasi. Secara klasik akuntansi merupakan proses pencatatan (*recording*), pengelompokan (*classifying*), perangkuman (*summarizing*) dan pelaporan (*reporting*) dari kegiatan transaksi perusahaan. Tujuan akhir dari kegiatan akuntansi adalah penerbitan laporan-laporan keuangan. Laporan-laporan keuangan adalah merupakan suatu informasi. Sistem informasi yang berbasis pada

komputer sekarang dikenal dengan istilah sistem informasi akuntansi atau SIA (*accounting information system* atau AIS). Sistem informasi akuntansi (SIA) didefinisikan oleh Stephen A. Moscovice dan Mark G. Simkin sebagai berikut ini. SIA adalah merupakan komponen operasi yang mengumpulkan, mengklasifikasikan, memproses, menganalisis, mengkomunikasikan informasi pengambilan keputusan dengan orientasi financial yang relevan bagi pihak-pihak dalam perusahaan (secara prinsip adalah manajemen). Menurut Robert G. Murdick, Thomas C. Fuller dan Joel E. Ross : SIA adalah kumpulan kegiatan-kegiatan dari organisasi yang bertanggung jawab untuk menyediakan informasi keuangan dan informasi yang didapatkan dari transaksi data untuk tujuan pelaporan internal kepada manajer untuk digunakan dalam pengendalian dan perencanaan sekarang dan operasi masa depan serta pelaporan eksternal kepada pemegang saham, pemerintah dan pihak-pihak luar lainnya (Prof. Dr. Jogiyanto HM, MBA, Akt; 2005 : 17).

II.5.1. Persediaan Barang

Persediaan (*inventory*) digunakan untuk mengindikasikan :

1. Barang dagang yang disimpan untuk kemudian dijual dalam operasi bisnis perusahaan dan,
2. Bahan yang digunakan dalam proses produksi atau yang disimpan untuk tujuan itu. Dalam bab ini, fokus utama yang dibahas adalah persediaan barang dagang dan kemudian dijual kembali.

Beberapa biaya atau harga pokok yang harus dimasukkan dalam persediaan seperti telah diilustrasikan dalam bab-bab sebelumnya, harga pokok barang adalah

harga beli dikurangi diskon pembelian (jika ada). Biaya-biaya ini biasanya merupakan bagian terbesar dari biaya persediaan. Persediaan barang dagang juga mengandung biaya-biaya lain seperti transportasi, bea impor, dan biaya asuransi untuk perlindungan atas kehilangan perjalanan (Warren Reeve Fess ; 2006 :451).

Persediaan barang dagang (*merchandise inventory*) adalah barang-barang yang dimiliki perusahaan untuk dijual kembali. Untuk perusahaan pabrik termasuk dalam persediaan adalah barang-barang yang akan digunakan untuk proses produksi selanjutnya. Persediaan dalam perusahaan pabrik terdiri dari persediaan bahan baku, persediaan dalam proses dan persediaan barang jadi. Bab ini terutama akan membahas persediaan dalam perusahaan dagang merupakan bagian yang cukup berarti dari seluruh aktiva perusahaan. Di samping itu, transaksi yang berhubungan dengan persediaan dengan merupakan aktivitas yang paling sering terjadi.

Persediaan barang dagang pada umumnya dinilai pada harga terendah antara harga perolehan dan harga pasar atau nilai yang diharapkan dapat direalisasikan. Cara penilaian dan metode penetapan harga pokok harus diungkapkan dalam laporan keuangan (Soemarso S.R; 2002 : 384).

II.5.2. Metode *First –In, First-Out* (FIFO)

Sebagian besar perusahaan mengeluarkan barang sesuai dengan urutan pembeliannya. Hal ini terutama untuk barang-barang yang tidak tahan lama dan produk-produk yang modelnya cepat berubah. Sebagai contoh tokoh bahan pangan menyusun produk-produk susu dalam rak-rak berdasarkan tanggal kedaluwarsanya. Begitu juga toko pakaian memajang pakaian sesuai dengan

musim. Pada akhir musim, mereka biasanya memberikan diskon untuk menghabiskan pakaian-pakaian yang musimnya sudah lewat atau sudah ketinggalan mode. Jadi, metode fifo dapat dikatakan konsisten dengan arus *fisik* atau pergerakan barang dagang. Dalam hal ini, metode FIFO memberikan hasil yang sama diperoleh melalui pengidentifikasian biaya khusus setiap barang yang dijual yang ada dalam persediaan (Warren Reeve Fess ; 2006 :460).

Berdasarkan metode FIFO penentuan harga pokok penjualan untuk satuan periode tertentu dihitung berdasarkan rumus: Persediaan awal ditambah dengan pembelian bersih dikurangi dengan persediaan akhir. Angka untuk pembelian bersih diambil dari saldo akun yang bersangkutan di buku besar. Angka-angka kuantitas persediaan awal dan akhir saldo diperoleh dengan jalan melakukan perhitungan fisik. Harga pokok persediaan dihitung dengan mengalikan kuantitas pada harga pokok per unit (Soemarso S.R; 2002 : 387).

II.6. *Unified Modeling Language (UML)*

UML singkatan dari *Unified Modelling Language* yang berarti bahasa pemodelan standart. (Chonoles; 2003 : 6) mengatakan sebagai bahasa, berarti *UML* memiliki sintaks dan *semantic*. Ketika kita membuat model menggunakan konsep *UML* ada aturan –aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat harus berhubungan satu dengan lainnya harus mengikuti standart yang ada. *UML* bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dari sistem, bagaimana transaksinya? Bagaimana sistem mengatasi error yang terjadi?

Bagaimana keamanan terhadap sistem yang ada kita buat? Dan sebagainya dapat dijawab dengan *UML*.

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

UML telah diaplikasikan dalam investasi perbankan, lembaga kesehatan, departemen pertahanan, sistem terdistribusi, sistem pendukung alat kerja, retail, sales, dan supplier.

Blok pembangunan utama *UML* adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasikan objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. *UML* memungkinkan para anggota team untuk bekerja sama dalam mengaplikasikan beragam sistem. Intinya, *UML* merupakan alat komunikasi yang konsisten dalam mensupport para pengembang sistem saat ini. Sebagai perancang sistem mau tidak mau pasti menjumpai *UML*, baik kita sendiri yang membuat sekedar membaca diagram *UML* buatan orang lain (Prabowo Pudjo Widodo Dan Herlawati; 2011 : 6-7).

II.6.1. Diagram-Diagram UML

Beberapa literatur menyebutkan bahwa *UML* menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa yang digabung, misalnya diagram komunikasi, diagram urutan, dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain :

II.6.1.1. Diagram Kelas Bersifat Statis

Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi, serta relasi-relasi diagram. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas.

II.6.1.2. Diagram Paket (*Package Diagram*) Bersifat Statis

Diagram ini memperlihatkan kumpulan kelas-kelas merupakan bagian dari diagram komponen.

II.6.1.3. Diagram *Use Case* Bersifat Statis

Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.

II.6.1.4. Diagram Interaksi Dan *Sequence* (Urutan) Bersifat Dinamis

Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam waktu tertentu.

II.6.1.5. Diagram Komunikasi (*Communication Diagram*) Bersifat Dinamis

Diagram sebagai pengganti diagram kolaborasi *UML* yang menekankan organisasi *structural* dari objek-objek yang menerima serta mengirim pesan.

II.6.1.6. Diagram Statechart (*Statechart Diagram*) Bersifat Dinamis

Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*State*), transisi kejadian serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka (*interface*), kelas, kolaborasi dan terutam penting pada pemodelan sistem-sistem yang reaktif.

II.6.1.7. Diagram Aktivitas (*Activity Diagram*) Bersifat Dinamis

Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan member tekanan pada aliran kendali antar objek.

II.6.1.8. Diagram Komponen (*Component Diagram*) Bersifat Statis

Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan diagram kelas dimana komponen dipetakan kedalam satu atau lebih kelas-kelas. Antarmuka-antarmuka serta kolaborasi-kolaborasi.

II.6.1.9. Diagram Deployment (*Deployment Diagram*) Bersifat Statis

Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run time*). Memuat simpul-simpul beserta komponen-komponen yang ada di dalamnya. Diagram *Deployment* berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat

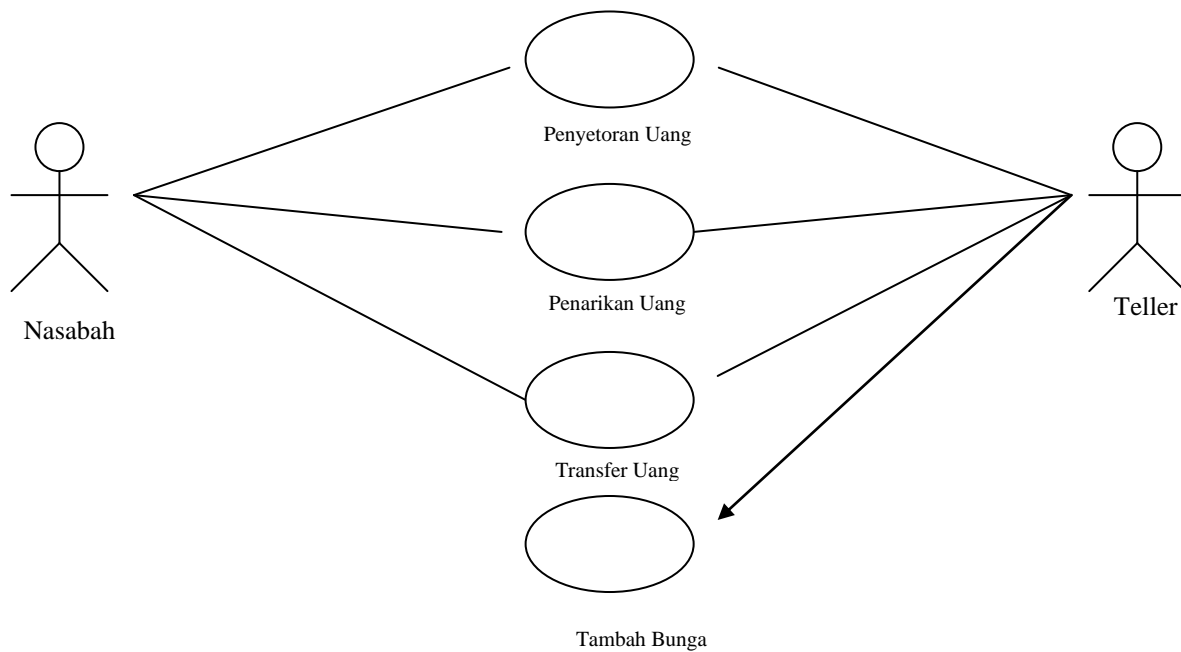
berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*).

Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai dengan kebutuhan. Pada *UML* dimungkinkan kita menggunakan diagram-diagram lainya misalnya *Data Flow Diagram*, *Entity Relationship Diagram* dan sebagainya (Prabowo Pudjo Widodo, Dan Herlawati; 2011 : 10-12).

II.6.2. Diagram Use Case (*Use Case Diagram*)

Use Case menggambarkan *external view* dari sistem yang akan kita buat modelnya. Menurut Pooley (2005:15) mengatakan bahwa model *use case* dapat dijabarkan dalam diagram, tetapi yang perlu diingat, diagram tidak indentik dengan model karena model lebih luas dari diagram.

Gambar di bawah ini merupakan salah satu contoh bentuk diagram *use case* (Prabowo Pudjo Widodo Dan Herlawati; 2011 : 16-17).



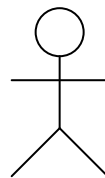
Gambar II.3. Diagram Use Case

Sumber : Prabowo Pudjo Widodo Dan Herlawati (2011:17)

Komponen pembentuk diagram *use case* adalah sebagai berikut.

II.6.2.1. Aktor (*actor*)

Menggambarkan pihak-pihak yang berperan dalam sistem. Menurut Chonoles (2003 :17) menyarankan sebelum membuat *use case* dan menentukan aktornya, agar mengidentifikasi siapa saja pihak yang terlibat dalam sistem kita. Pihak yang terlibat biasanya dinamakan *stakeholder*.

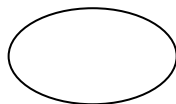


Gambar II.4. Aktor

Sumber : Prabowo Pudjo Widodo Dan Herlawati (2011:17)

II.6.2.2. Use Case

Aktivitas/ sarana yang disiapkan oleh bisnis/sistem. Menurut Pitone (2005 : 21) *use case* menggambarkan fungsi tertentu dalam suatu sistem berupa komponen kejadian atau kelas. Sedangkan menurut Whitten (2004 : 258) mengartikan *use case* sebagai urutan langkah-langkah yang secara tindakan saling terkait (skenario) baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal. *Use case* digambarkan dalam bentuk *ellips/oval*.



Gambar II.5. Simbol Use Case

Sumber : Prabowo Pudjo Widodo Dan Herlawati (2011:22)

Use case sangat menentukan karakteristik sistem yang kita buat, oleh karena itu Chonoles (2003:22-23) menawarkan cara untuk menghasilkan *use case* yang baik yakni :

a. Pilihlah Nama Yang Baik

Use case adalah sebuah *behaviour* (perilaku), jadi seharusnya dalam frase kata kerja. Untuk membuat namanya lebih detil tambahkan kata benda mengindikasikan dampak aksinya terhadap suatu kelas objek. Oleh karena itu diagram *use case* seharusnya berhubungan dengan diagram kelas.

b. Ilustrasikan Perilaku Dengan Lengkap

Use case dimulai dari inisiasi oleh aktor primer dan berakhir pada aktor dan menghasilkan tujuan. Jangan membuat *use case* kecuali anda mengetahui

tujuannya. Sebagai contoh memilih tempat tidur (*King Size, Queen Size*, atau dobel) saat tamu memesan tidak dapat dijadikan *use case* karena merupakan bagian dari *use case* pemesanan kamar dan tidak dapat berdiri sendiri (tidak mungkin tamu memesan kamar tidur jenis king tapi tidak memesan kamar hotel).

c. Identifikasi Perilaku Dengan Lengkap

Untuk mencapai tujuan dan menghasilkan nilai tertentu dari aktor, *use case* harus lengkap. Ketika memberi nama pada *use case*, pilihlah frasa kata kerja yang implikasinya hingga selesai. Misalnya gunakan frasa *reserve a room* (pemesanan kamar) dan jangan *reserving a room* (memesan kamar) karena memesan menggambarkan perilaku yang belum selesai.

d. Menyediakan Use Case Lawan (*Inverse*)

Kita biasanya membutuhkan *use case* yang membatalkan tujuan, misalnya pada *use case* pemesanan kamar, dibutuhkan pula *use case* pembatalan pesanan kamar.

e. Batasi Use Case Hingga Satu Perilaku Saja

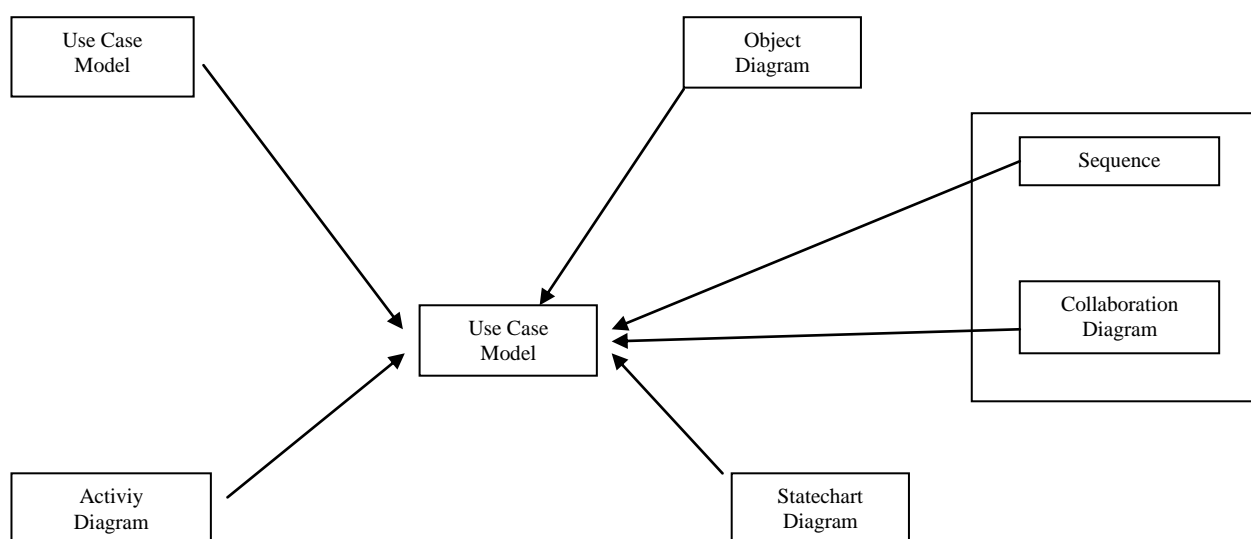
Kadang kita cenderung membuat *use case* yang lebih dari satu tujuan aktivitas. Guna menghindari kerancuan, jagalah *use case* kita hanya fokus pada satu hal. Misalnya, penggunaan *use case check in* dan *check out* dalam satu *use case* menghasilkan ketidakfokusan, karena memiliki dua perilaku yang berbeda.

II.6.2.3. Hubungan (*Link*)

Aktor mana saja yang terlibat dalam *use case* ini.

II.6.3. Diagram Kelas (*Class Diagram*)

Diagram kelas adalah inti dari proses pemodelan objek. Baik *forward engineering* maupun *reverse engineering* memanfaatkan diagram ini *forward engineering* adalah proses perubahan model menjadi kode program sedangkan *reverse engineering* sebaliknya merubah kode program menjadi model (Prabowo Pudji Widodo Dan Herlawati; 2011 : 37).



Gambar II.6. Hubungan Diagram Kelas Dengan Diagram UML lainnya

Sumber : Prabowo Pudjo Widodo Dan Herlawati (2011 : 38)

II.6.4. Diagram Aktivitas (*Activity Diagram*)

Diagram aktivitas lebih memfokuskan diri pada eksekusi dan alur sistem dari pada bagaimana sistem dirakit. Diagram ini tidak hanya memodelkan software melainkan memodelkan bisnis juga. Diagram aktivitas menunjukkan aktivitas sistem dalam kumpulan aksi-aksi. Ketika digunakan dalam pemodelan *software*, diagram aktivitas merepresentasikan pemanggilan suatu fungsi tertentu misalnya

call. Sedangkan bila digunakan dalam pemodelan bisnis, diagram ini menggambarkan aktivitas yang dipicu oleh kejadian-kejadian diluar seperti pemesanan atau kejadian-kejadian internal misalnya penggajian tiap jumat sore (Probowo Pudji Widodo, Dan Herlawati ;2011 : 143-145).

Aktivitas merupakan kumpulan aksi-aksi. Aksi-aksi nelakukan langka sekali saja tidak boleh dipecah menjadi beberapa langkah-langkah lagi. Contoh aksinya yaitu :

- a. Fungsi Matematika
- b. Pemanggilan Perilaku
- c. Pemrosesan Data

Ketika kita menggunakan diagram aktivitas untuk memodelkan perilaku suatu *classifier* dikatakan kontek dari aktivitas. Aktivitas dapat mengakses atribut dan operasi *classifier*, tiap objek yang terhubung dan parameter-parameter jika aktivitas memiliki hubungan dengan perilaku. Ketika digunakan dengan model proses bisnis, informasi itu biasanya disebut *process-relevant data*. Aktivitas diharapkan dapat digunakan ulang dalam suatu aplikasi, sedangkan aksi biasanya *specific* dan digunakan hanya untuk aktivitas tertentu.

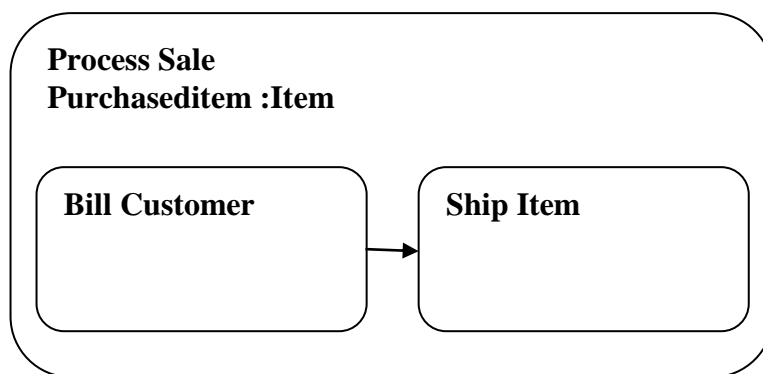
Aktivitas digambarkan dengan persegi panjang tumpul. Namanya ditulis di kiri atas. Parameter yang terlibat dalam aktivitas ditulis dibawahnya.



Gambar II.7. Aktivitas sederhana tanpa rincian

Sumber : Prabowo Pudjo Widodo Dan Herlawati (2011:145)

Detail aktivitas dapat dimasukkan di dalam kotak. Aksi diperlihatkan dengan symbol yang sama dengan aktivitas dan namanya diletakkan didalam persegi panjang.



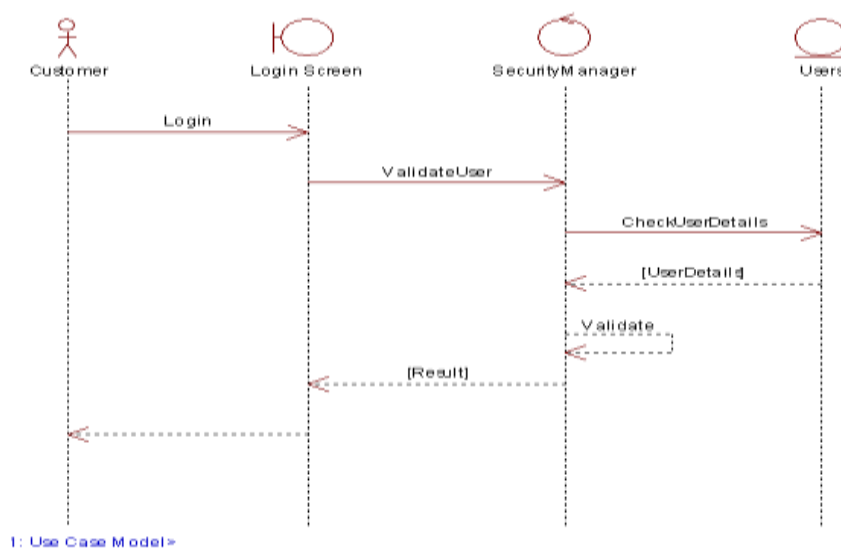
Gambar II.8. Aktivitas dengan detail rincian

Sumber : Prabowo Pudjo Widodo Dan Herlawati (2011:145)

II.6.5. Diagram Urutan (*Sequence Diagram*)

Menurut Douglas (2004 : 174) menyebutkan ada tiga diagram primer UML dalam memodelkan scenario interaksi, yaitu diagram urutan (*sequence diagram*), diagram waktu (*timing diagram*) dan diagram komunikasi (*communication diagram*).

Menurut Pilone (2005 : 174) menyatakan bahwa diagram yang paling banyak dipakai adalah diagram urutan. Gambar II.10. memperlihatkan contoh diagram urutan dengan notasi-notasinya yang akan dijelaskan nantinya (Prabowo Pudjo Widodo Dan Herlawati; 2011 : 174 – 175).



Gambar II.9. Diagram Urutan

Sumber : Prabowo Pudjo Widodo Dan Herlawati (2011:175)

II.7. Bahasa Pemrograman *Microsoft Visual Studio 2008*

Microsoft Visual Studio 2008 merupakan kelanjutan dari *Microsoft Visual Studio* sebelumnya, yaitu *Visual Studio .Net 2003* yang diproduksi oleh Microsoft. Pada bulan Februari 2002 *Microsoft* memproduksi teknologi. *Net Framework* versi 1.0, teknologi. *Net* ini didasarkan atas susunan berupa *Net Framework*, sehingga setiap produk baru yang terkait dengan teknologi. *Net* akan selalu berkembang mengikuti perkembangan. *Net Frameworknya*. Pada perkembangannya nantinya mungkin untuk membuat program dengan teknologi. *Net* memungkinkan para pengembang perangkat lunak akan dapat menggunakan lintas sistem operasi, yaitu dapat dikembangkan di sistem operasi windows juga dapat dijalankan pada sistem operasi lain, misalkan pada sistem operasi *Linux*, seperti yang telah dilakukan pada pemrograman *Java* oleh *Sun Microsystems*. Pada saat ini perusahaan-perusahaan sudah banyak mengupdate aplikasi lama yang

dibuat *Microsoft Visual Basic 6.0* ke teknologi .Net karena kelebihan-kelebihan yang ditawarkan, terutama memungkinkan pengembang perangkat lunak secara cepat mampu membuat program *robust*, serta berbasiskan integrasi ke internet yang dikenal dengan *XML Web Service* (Ketut Darmayuda ; 2009 : 1)

Untuk melihat tampilan visual studio 2008 dapat dilihat pada gambar II.11. sebagai berikut :



Gambar II.10. Tampilan Utama Visual Studio 2008

Sumber : Ketut Darmayuda (2009 : 12)

II.8. Microsoft SQL Server

SQL Server adalah salah satu produk *Relational Database Management Sistem* (RDBMS) populer saat ini. Fungsi utamanya adalah sebagai *database server* yang mengatur semua proses penyimpanan data dan transaksi suatu aplikasi. *SQL Server Versi 2008* memiliki *feature-feature* lengkap untuk membangun aplikasi mulai skala kecil sampai dengan tingkat *enterprise*. Untuk lebih jelasnya dapat dilihat pada Gambar II.10 (SmitDev Community; 2010 : 1).



Gambar II.11. *SQL Server 2008*

Sumber : SmitDev Community (2010 : 20)

II.8.1. Tingkatan Data *SQL Server 2008*

Ada 3 tingkatan data *SQL Server 2008* adalah sebagai berikut :

1. Level fisik (*Physcal Level*)

Merupakan level yang paling rendah karena menggambarkan bagaimana data disimpan pada kondisi yang sebenarnya pada server.

2. Level Konseptual (*Conseptual Level*)

Merupakan level yang menggambarkan data yang disimpan pada database dan dijelaskan secara keseluruhan antara data tersebut.

3. Level Pandangan (*View Level*)

Merupakan level yang menggambarkan sebagian dari seluruh database sesuai dengan kebutuhan user (SmitDev Community ; 2010 :3-4).

II.8.2. *SQL Server Configuration Manager*

Ada 3 tingkatan dalam *Sql Server Configuration Manager* adalah sebagai berikut :

1. *Restart Service*

Kita bisa melakukan restart pada setiap *SQL Server Service* tersebut jika diperlukan.

2. *Start Mode*

Setiap *SQL Service* telah dikonfigurasi secara *default* menjalankan *SQL Server Service* saat *server restart* atau baru dinyalakan.

3. *Network Configuration*

Pada *SQL Server Network Configuration* terdapat beberapa protocol yang bisa digunakan untuk melakukan antar muka dengan *SQL Server 2008* yaitu :

a. *Shared Memory*

Dengan *shared memory*, koneksi database bisa ditempatkan pada server lokal tanpa menggunakan network.

b. *Named Pipes*

Dengan *Named Pipes*, koneksi bisa dilakukan dengan *high speed LAN*.

c. *TCP/IP*

Dengan *TCP/IP*, koneksi *database* bisa dilakukan dengan koneksi *network* yang menggunakan *TCP/IP* (SmitDev Community; 2010 : 16-18).