

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Menurut Anastasia Diana dan Lilis Setiawati dalam bukunya Sistem Informasi Akuntansi (2011 ; 3) sistem Merupakan serangkaian bagian yang saling tergantung dan bekerja sama untuk mencapai tujuan tertentu. Suatu sistem pasti tersusun dari sub-sub sistem yang lebih kecil yang juga saling tergantung tergantung dan bekerja sama mencapai tujuan.

Menurut Hanif Al Fatta dalam bukunya Analisis dan Perancangan Sistem Informasi (2007 ; 1) sistem merupakan suatu kumpulan atau himpunan dari unsur-atau variabel-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung satu sama lain.

Menurut George H. Bodner dan William S. Hoopwood dalam bukunya Sistem Informasi Akuntansi (2006 ; 3) sistem merupakan sekumpulan sumber daya yang saling terkait untuk mencapai suatu tujuan,

II.1.1. Informasi

Menurut Budi Sutedjo Dharma Oetomo dalam bukunya Perencanaan dan Perancangan Sistem Informasi (2006 ; 11) informasi merupakan bentuk yang mudah dipahami dan merupakan pengetahuan yang relevan dan dibutuhkan untuk menambah wawasan bagi pemakainya untuk mencapai tujuan.

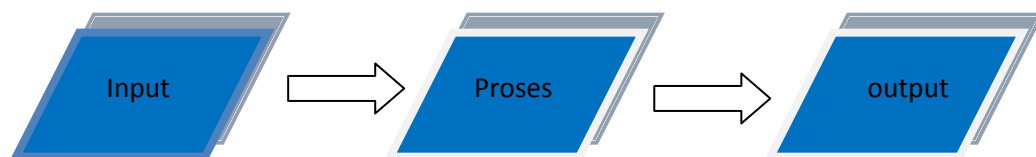
Menurut George H. Bodner dan William S. Hoopwood dalam bukunya Sistem Informasi Akuntansi (2006 ; 3) informasi merupakan suatu data yang diorganisasi yang dapat mendukung keputusan ketepatan pengambilan keputusan.

Menurut Hanif Al Fatta dalam bukunya Analisis dan Perancangan Sistem Informasi (2007 ; 1). Mcleod (1995) mengatakan bahwa informasi merupakan data yang telah diproses dan memiliki arti.

II.1.2. Sistem Informasi

Menurut Anastasia Diana dan Lilis Setiawati dalam bukunya Sistem Informasi Akuntansi (2011 ; 4) Sistem Informasi yang kadang kala disebut pemrosesan data, merupakan sistem buatan manusia yang yang terdiri dari sekumpulan komponen - baik manual ataupun berbasis komputer yang terintegrasi untuk menyimpulkan, menyimpan, dan mengelola data serta menyediakan informasi kepada pihak- pihak yang berkepentingan sebagai pemakai informasi tersebut.

Input dalam sistem informasi adalah data yang yang relevan untuk menghasilkan informasi yang diinginkan. Proses adalah langkah-langkah yang perlu dilakukan untuk mengolah data menjadi informasi, Sedangkan outputnya adalah berupa formasi yang merupakan hasil pemrosesan data.



Gambar II.1. Komponen Sistem Informasi

Sumber : Sistem Informasi Akuntansi

(Anastasia Diana dan Lilis Setiawati ; 2011 : 4)

Sedangkan menurut Budi Sutedjo Dharma Oetomo dalam bukunya Perencanaan dan Perancangan Sistem Informasi (2006 ; 11) Sistem Informasi

dapat didefinisikan sebagai kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi.

Menurut Hanif Al Fatta dalam bukunya Analisis dan Perancangan Sistem Informasi (2007 ; 1) sistem informasi sistem data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang.

Masih menurut Hanif Al Fatta ada beberapa sistem yang yang penting untuk diingat sebagai konsep pengembangan sistem, yaitu :

1. Dekomposisi

Untuk menganalisa dan memahami secara menyeluruh sebuah sistem yang besar, biasanya dibutuhkan waktu yang cukup lama. Untuk mempermudah pekerjaan ini digunakan konsep dekomposisi. Dekomposisi adalah pembagian sistem-sistem kedalam komponen-komponen yang lebih lebih kecil (subsistem). Dekomposisi memiliki beberapa keuntungan diantaranya :

- a. Analis menjadi lebih mudah mengatur dan menganalisa setiap subsistem secara lebih rinci.
- b. Pada pengembangan sistem, sistem bisa didekomposisi menjadi beberapa modul. Pengembangan beberapa modul. Pengembangan beberapa modul bisa dilakukan secara paralel dengan syarat tidak ada ketergantungan antar modul yang dibangun.

2. Modularitas

Konsep modularitas berhubungan dengan dekomposisi. Pada saat melakukan dekomposisi, diharapkan sistem yang besar terbagi menjadi subsistem-subsistem yang relatif sama ukurannya. Dengan modul-modul ini maka beban kerja mengembangkan sistem bisa didistribusikan secara merata pada sumber daya yang ada. Pengembangan sistem yang lebih sederhana karena hanya terfokus pada satu modul terlebih dahulu, baru dilakukan integrasi antar modul.

3. Coupling

Dari modul-modul yang kita peroleh, kadang-kadang ditemukan beberapa modul yang memiliki ketergantungan dengan modul yang lain. Pada kasus seperti ini, modul-modul yang saling tergantung harus dipasangkan (*decouple*). Dengan cara ini bisa diketahui modul yang bisa bekerja secara independen dan modul-modul yang harus diselesaikan terlebih dahulu sebelum modul yang lain bisa bekerja.

4. Kohesi

Dari kelompok *Coupling* antar modul, kita bisa dapatkan kelompok-kelompok modul dengan karakteristik yang hampir sama. Di sini muncul konsep kohesi dimana kelompok modul itu harus dianalisis bersama-sama dengan kelompok modul yang saling berkohesi.

II.1.3. Komponen Sistem Informasi

Stair (1992) menjelaskan bahwa sistem informasi berbasis komputer (CBIS)

Dalam suatu organisasi terdiri atas-atas komponen-komponen berikut :

- a. Perangkat Keras, yaitu perangkat keras komponen untuk melengkapi kegiatan Memasukkan data, memproses data, dan keluaran data.
- b. Perangkat lunak, yaitu program dan instruksi yang diberikan ke komputer.
- c. Database, yaitu kumpulan data dan informasi yang diorganisasikan sedemikian rupa sehingga mudah diakses pengguna sistem informasi.
- d. Telekomunikasi, yaitu komunikasi yang menghubungkan antara pengguna sistem dengan sistem komputer secara bersama-sama ke dalam suatu jaringan kerja yang efektif.
- e. Manusia, yaitu personel dari sistem informasi, meliputi manajer, analis, programmer, dan operator, serta bertanggung jawab terhadap perawatan sistem.

Prosedur yaitu tata cara yang meliputi strategi, kebijakan, metode, dan peraturan-peraturan dalam menggunakan sistem informasi berbasis komputer.

Hanif Al Fatta (2007 ; 9-10).

II.2. Akuntansi

Akuntansi merupakan proses mengidentifikasi, mengukur, mencatat dan mengkomunikasikan peristiwa-peristiwa ekonomi dari suatu organisasi (bisnis maupun nonbisnis) kepada pihak-pihak yang berkepentingan dengan informasi bisnis tersebut (pengguna informasi). Pada dasarnya focus utama dari akuntansi adalah transaksi bisnis. (Anastasia Diana dan Lilis Setiawati ; 2011: 14).

Akuntansi menurut Johar Arifin dan Busono Adi Wicaksono dalam bukunya Komputer Akuntansi dengan Microsoft Excel (2006 ; 10) memiliki dua definisi yakni dari sudut pemakai, akuntansi merupakan suatu disiplin yang menyediakan

informasi yang diperlukan untuk melaksanakan suatu kegiatan secara efisien dan mengevaluasi kegiatan suatu organisasi. Sedangkan dari sudut proses kegiatan, merupakan suatu proses pencatatan, penggolongan, peringkasan, pelaporan, dan penganalisaan data keuangan suatu organisasi.

Akuntansi adalah suatu sistem informasi yang mengidentifikasi, mencatat, dan mengkomunikasikan peristiwa-peristiwa ekonomi dari suatu organisasi kepada para pengguna yang berkepentingan (Jerry J. Weygent, Donald E. Keiso, Paul D. Kimmel ; 2007: 4)

II.2.1. Sistem Informasi Akuntansi

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

Menurut George H. Bodner dan William S. Hoopwood dalam bukunya Sistem Informasi Akuntansi (2006 ; 3) Sistem Informasi Akuntansi merupakan kumpulan sumber daya manusia dan peralatan, yang dirancang untuk mengubah data keuangan dan data lainnya kedalam informasi. Informasi tersebut itu dikomunikasikan kepada pembuat keputusan. Sistem Informasi Akuntansi melakukan hal tersebut entah dengan sistem manual atau melalui sistem terkomputerisasi. Juga sistem berbasis komputer yang yang dirancang untuk mentransformasi data akuntansi menjadi informasi. Namun, bisa juga memiliki cakupan arti yang lebih luas, yaitu mencakup juga siklus pemrosesan transaksi , penggunaan teknologi informasi dan pengembangan sistem informasi. (George H. Bodner dan William S. Hoopwood ; 2006: 8).

II.2.2. Tujuan Sistem Informasi Akuntansi

Lingkup Sistem Informasi Akuntansi dapat dijelaskan dari manfaat yang didapat dari informasi akuntansi. Manfaat dan tujuan Sistem Informasi Akuntansi tersebut adalah sebagai berikut :

1. Mengamankan Harta/Kekayaan perusahaan. Harta/ kekayaan disini meliputi kas perusahaan, persediaan barang dagangan, termasuk asset tetap perusahaan. Tidak ada pemilik yang senang jika uang perusahaan dicuri orang(entah itu karyawan maupun orang asing). Contoh, seorang memiliki usaha jasa persewaan komik. Pemilik menempatkan seorang kasir di tempat persewaan tersebut. Setiap malam, pemilik akan mengambil kas hasil persewaan. Tentunya, pemilik tidak suka jika tersebut tidak meyetorkan seluruh kas yang diterima. Kesempatan untuk mencuri uang perusahaan seperti ini dapat diminimalkan, jika pemilik persewaan komik tersebut membangun sistem yang baik. Bagaimana caranya, akan dibahas lebih lanjut diulasan-ulasan berikutnya.
2. Menghasilkan beragam informasi untuk pengambilan keputusan. Misal, pengelola toko swalayan memerlukan informasi mengenai barang apa diminati oleh konsumen. Membeli barang dagangan yang kurang laku berarti kas akan terjebak pada persediaan (yang sulit laku tersebut) dan berarti kehilangan kesempatan untuk membeli barang yang laku. Hal ini sangat penting, karena toko swalayan pada dasarnya tidak dapat mengambil margin laba yang tinggi (karena ketatnya persaingan antar toko swalayan). Jadi, toko swalayan akan lebih mengandalkan pada perputaran persediaan yang cepat. Oleh karena itu

informasi mengenai persediaan yang laris menjadi kunci sukses sebuah swalayan. Informasi semacam ini dapat diakses dengan mudah jika toko swalayan tersebut membangun informasi dengan baik.

3. Menghasilkan informasi untuk pihak eksternal. Setiap pengelola usaha memiliki kewajiban untuk membayar pajak. Besarnya pajak yang dibayar tergantung pada omset penjualan (jika pemilih menggunakan norma dalam perhitungan pajaknya) atau tergantung pada laba rugi usaha (jika pengelola memilih untuk tidak menggunakan norma dalam perhitungan pajaknya). Tanpa sistem yang baik, bisa jadi pengelola kesulitan untuk menentukan besarnya omset dan besarnya laba rugi usaha. Selain untuk kepentingan perpajakan, adakalanya pengelola usaha usah juga terlibat dalam kegiatan utang piutang dengan bank atau koperasi simpan pinjam. Bank membutuhkan informasi omset dan laba rugi usaha untuk memutuskan besarnya utang yang akan diberikan.
4. Menghasilkan informasi untuk penilaian kinerja karyawan atau divisi. Sistem informasi dapat juga dimanfaatkan untuk penilaian kinerja karyawan atau divisi. Sebagai contoh, pengelola toko swalayan dapat memanfaatkan data penjualan untuk menilai kinerja pasar.

II.3. Normalisasi Data

Normalisasi data adalah proses dimana table-table pada database diletakkan dalam hal kesalingtergantungan diantara field-field pada suatu tabel. Misalnya, jika pada tabel tersebut, maka tabel tersebut harus dipecah menjadi banyak tabel. Banyaknya

tabel pecahannya tergantung pada seberapa ketergantungannya. Tiap tabel hanya boleh memiliki sebuah field kunci yang menjadi ketergantungan dari field lainnya dalam tabel tersebut. Pada proses normalisasi data, aturan yang dijadikan acuan adalah metode ketergantungan fungsional. Teorinya adalah bahwa setiap kolom dalam sebuah tabel data_siswa ada field nomor induk dan field nama siswa serta field tanggal lahir. Maka ketergantungan fungsionalnya dapat dinyatakan sebagai berikut : nmr_induk---tgl lahir. Artinya nm_siswa memiliki ketergantungan fungsional terhadap nmr_induk. Field nm_siswa isinya ditentukan oleh field nmr_induk. Demikian juga dengan field tgl_lahir isinya juga ditentukan oleh induk oleh nmr_induk. Maksud dari semua itu adalah agar field kunci yang Field nmr_induk dapat dijadikan patokan untuk mengisi nm_siswa dan field lainnya. (Wahana Komputer : 2010 ; 32).

II.3.1. Bentuk-bentuk Normalisasi

Ada beberapa bentuk normalisasi , yaitu :

- **Normal Form Pertama (1st Normal Form)**, pada tahapan ini tabel di-dekomposisi dari tabel bentuk tidak normal yang kemudian dipisahkan menjadi tabel-tabel kecil yang memiliki kriteria tidak memiliki tidak memiliki atribut yang bernilai ganda dan komposit. Semua atribut yang harus bersifat atomik.
- **Normal Form Kedua (2st Normal Form)**, pada tahapan ini tabel dianggap memenuhi normal kedua jika pada tabel tersebut tersebut semua atribut yang bukan kunci primer tabel tersebut bergantung penuh terhadap kunci

primer tabel tersebut. Dalam hal ini semua tabel diatas telah memenuhi kunci bentu normal form yang kedua. Adapun kunci primer masing-masing tabel adalah sebagai berikut. :

- Untuk tabel anggota kunci primernya **kode_agt**.
 - Untuk tabel buku kunci primernya **kd_bk**.
 - Untuk tabel transaksi kunci primernya **no_trans**.
- **Normal Form Ketiga (3rd Normal Form)**, setiap atribut pada tabel selain kunci primer atau kunci utama harus bergantung penuh pada kunci utama. Bentuk normal ketiga biasanya digunakan bila ada masih tabel yang belum efisien. Biasanya penggunaan bentuk normal(normalisasi) hanya sampai pada bentuk ketiga, dan tabel yang dihasilkan telah memiliki kualitas untuk membentuk sebuah database yang dapat diandalkan. Semua tabel diatas juga telah memenuhi bentuk normal tahap ketiga.(Wahana Komputer ; 2010 : 33-35).

II.4. UML (*Unified Modeling Language*)

Menurut Prabowo Pudjo Widodo dan Herlawati dalam bukunya Menggunakan UML (2011: 6) UML singkatan dari Unified Modelling Language yang berarti bahasa pemodelan standar dia mengatakan UML sebagai bahasa karena memiliki sintaks dan semantik. Ketika kita membuat model menggunakan konsep UML ada aturan-aturan yang harus diikuti. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk

yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

Meskipun UML sudah banyak menyediakan diagram yang bisa membantu mendefinisikan suatu aplikasi, tidak berarti bahwa semua diagram tersebut akan bisa menjawab persoalan yang ada

UML diaplikasikan untuk maksud tertentu, biasanya antara lain :

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

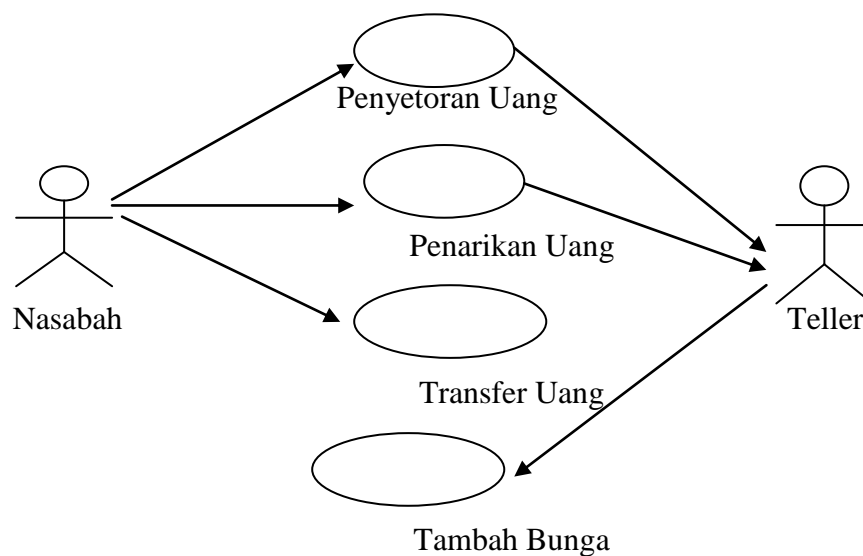
II.4.1. Diagram-Diagram UML

Diagram merupakan bahasa model untuk menggambarkan, membangun, dan mendokumentasikan sistem yang akan mereka rancang (Prabowo Pudjo Widodo dan Herlawati ; 2011: 7).

Menurut Prabowo Pudjo Widodo dan Herlawati dalam bukunya Menggunakan UML (2011 ; 10) ada beberapa jenis diagram itu, antara lain :

1. Diagram kelas. Bersifat statis. Diagram ini memperlihatkan himpuna kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek.

2. Diagram paket (*Package Diagram*). Bersifat statis, diagram ini memperlihatkan kumpulan kelas-kelas. Merupakan bagian dari diagram komponen.
3. Diagram *Use Case*. Bersifat statis. Diagram ini memperlihatkan himpunan use case dan actor-aktor. (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan dan diharapkan pengguna.



Gambar II.2 : Diagram Use Case

Sumber : "Menggunakan UML
(Prabowo Pudjo Widodo dan Herlawati ; 2011 :19)"

4. Diagram Interaksi atau *sequence* (urutan). Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan pada suatu waktu tertentu.
5. Diagram komunikasi (*Communication Diagram*). Bersifat dinamis. Diagram sebagai pengganti. Diagram kolaborasi *UML* 1.4 yang menekankan organisasi struktural dari objek yang menerima serta mengirim pesan.

6. Diagram Statechart (*StateChart Diagram*). Bersifat dinamis. Diagram status memperlihatkan keadaan pada sistem, memuat status (*state*), transisi, kejadian serta aktivitas. Diagram ini penting untuk memperlihatkan sifat dinamis dari antarmuka (*interface*), kelas, kolaborasi, dan terutama penting untuk pemodelan sistem-sistem yang reaktif.
7. Diagram Aktivitas (*Activity Diagram*). Bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan member tekanan pada aliran kendali antar objek.
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 dengan diagram kelas dimana komponen secara tipikal dipetakan kedalam satu atau lebih kelas-kelas, antarmuka-antarmuka serta kolaborasi-kolaborasi.
9. Diagram deployment (*Deployment diagram*). Bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*). Menurut simpul-simpul beserta komponen-komponen yang ada didalamnya. 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 kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*).

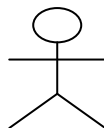
Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai dengan kebutuhan. Pada UML dimungkinkan kita menggunakan diagram-diagram lainnya (misalnya *Data Flow Diagram*, *Entity Relationship Diagram*) dan sebagainya.

Ada 3 komponen pembentuk Diagram use case, yaitu :

1. Aktor (*Actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
2. Use Case, aktivitas/sarana yang disiapkan oleh bisnis/sistem
3. Hubungan (*Link*), aktor mana saja yang terlibat use case ini.

1. Actor

Sebuah actor mencirikan suatu bagian *outside user* atau susunan yang berkaitan dengan user yang berinteraksi dengan sistem [Rumbaugh, Booch, dan Jacobson 1999]. Dalam model use case, aktor merupakan satu-satunya kesatuan eksternal yang berinteraksi dengan sistem. Berikut notasi actor dalam *UML*:

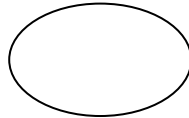


Gambar II.3 : Notasi Actor pada UML
Sumber : "Menggunakan UML
 (Prabowo Pudjo Widodo dan Herlawati ; 2011 : 19)"

2. Use Case

Menurut (pilone, 2005 : bab 7.1) *case* menggambarkan fungsi tertentu dalam suatu sistem berupa komponen, kejadian atau kelas. Sedangkan (Whitten, 2004: 258) mengartikan *use case* sebagai urutan langkah-langkah yang secara tindakan

saling terkait(skenario), baik teromatisasi maupun secara manual, untuk tujuan melengkapi satu bisnis tunggal. Use case digambarkan dalam bentuk ellips/ oval.



Gambar II.4 : Notasi Class pada UML
Sumber : "Menggunakan UM"L
(Prabowo Pudjo Widodo dan Herlawati ; 2011 : 22)"

3. Kelas (Class)

Menurut (Whitten, 2004 : 410) mengartikan kelas sebagai suatu set objek yang memiliki atribut dan perilaku yang sama. Kelas kadang-kadang disebut kelas objek (*Object class*) Secara alami, objek yang berupa buku analisis desain dan pemrograman tersruktur kita kelompokkan dalam satu kelas, yaitu kelas buku. Kedua objek memiliki atribut dan perilaku yang serupa. Contohnya, kedua objek mungkin memiliki atribut yang serupa seperti nomor ISBN, judul, tanggal penerbitan, edisi dan sebagainya. Demikian juga, kedua objek memiliki perilaku yang sama misalnya membuka dan menutup.

Secara teknis, (Pender, 2003: bab 5) mengartikan sebuah kelas sebagai suatu definisi sumber daya yang termasuk didalamnya informasi-informasi yang menggambarkan fitur suatu entitas dan bagaimana penggunaannya. Sedangkan objek adalah suatu entitas yang bersifat unik yang mengikuti aturan-aturan yang sudah didefinisikan dalam kelasnya.

4. Aktivitas dan aksi

Aktivitas merupakan kumpulan aksi-aksi. Aksi-aksi melaksanakan langkah sekali saja dan tidak boleh dipecah menjadi beberapa langkah lagi. Contoh aksi yaitu:

- ❖ Fungsi matematika.
- ❖ Pemanggilan perilaku.
- ❖ Pemrosesan data.

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

Tujuan fundamental UML 2.0 adalah memungkinkan user tidak hanya menggambarkan hubungan struktural melainkan juga proses dan aliran kejadian. Diagram interaksi menggabungkan bersama diagram-diagram lainnya dan memperlihatkan komunikasi antar diagram-diagram tersebut. UML 2.0 memiliki perkembangan yang lebih baik dari versi sebelumnya dalam menggambarkan aliran kontrol yang kompleks, oleh karena itu muncul beberapa terminologi yang baru yang harus diketahui. Diagram Interaksi dimaksudkan untuk mengembangkan komunikasi antar objek, bukan memanipulasi data saat berkomunikasi (Pitone, 2005: bab 10).

(Douglas, 2004: bab 3.5) menyebutkan ada tiga diagram [rimer UML dalam memodelkan skenario interaksi, yaitu : diagram urutan (*sequence diagram*), diagram waktu (*timing diagram*) dan diagram komunikasi(*communication diagram*). Namun demikian (Pilone, 2005: bab10) menyatakan bahwa diagram yang paling banyak dipakai adalah diagram urutan.

II.4.2. Evolusi UML

(Chonoles, 2003: bab 1) menjelaskan bahwa sebelum ada UML, para pengembang bahasa pemrograman berorientasi objek sulit berkomunikasi satu sama lain. Ada kira-kira 50 jenis notasi dan grafik yang menggambarkan bahasa pemrograman berorientasi objek pada saat itu.

Para pengguna notasi yang berlainan ini saling berebut pengaruh agar notasi yang digunakan menjadi notasi standar. Walaupun dijumpai beberapa notasi yang cukup jelas dan sangat cocok untuk menyelesaikan problem pembuatan perangkat lunak *Objek Oriented Program(OOP)*, tetapi belum diakui oleh pengembang sistem yang lain.

Pada bulan oktober 1994. Jim Rumbaugh, penemu notasi *Object Modelling Technique (OMT)* dan Grady Booch, penemu *Booch Method* (Metode Booch) bersama-sama menyatukan notasi mereka. Ditahun yang sama Ivar Jacobsen (penemu *Objectory Method*) ikut bergabung, Ketiga orang ini sering disebut "*Three Amigos*", bersama –sama membangun notasi standar *OOP* untuk software Rational IBM. Jim Rumbaugh banyak memeberi masukan dalam hal pembuatan notasi dan analisa *UML*. Grady Booch merancang secara detail kapabilitas *UML*.

Sedang Ivar Jacobsen berusaha membuat *UML* cocok dengan model bisnis dan mencoba mengembangkan *use case* diagram lebih lanjut.

Ketiga pengembang *UML* tersebut mengalami kesulitan akibat kompleksnya permasalahan OOP yang ada, namun mereka dibantu oleh Object Management Group(OMG). OMG adalah gabungan dari kurang lebih 800 perusahaan pengembang perangkat lunak berorientasi objek. Setelah perdebatan sengit yang cukup lama, konsensus tentang notasi berhasil dicapai dengan sukses pada bulan november 1997 berkat bantuan OMG.

Sejak tahun 1997, divisi *Revision Task Force (RTF)* milik *OMG* beberapa kali merevisi *UML*. Revisi dimaksudkan untuk memperkuat konsistensi notasi, meningkatkan kekompakan antara user dan pengembang perangkat lunak. Akan tetapi *UML* terpaksa mengikuti perkembangan software-software berbasis objek yang ada (misalnya Java) dari sisi pendekatan komponen (*Component – Based Development*) dan kemampuan *tools software-software* tersebut. Setelah dilakukan secara sistematis, akhirnya dihasilkan *UML 2.0* pada tahun 2003.

Adapun beberapa konsep original objek yang selain diatas, adalah sebagai berikut :

- ❖ Abstraksi (*Abstraction*). Abstraksi adalah gambaran dari sesuatu yang real. Walaupun dalam objek real memiliki jumlah informasi yang tidak terbatas, abstraksi hanya berupa informasi yang penting saja yang mewakili objek real tersebut.

- ❖ Enkapsulasi (*Encapsulation*). Enkapsulasi menggambarkan cara mengorganisir informasi sehingga digunakan secara efisien dalam aplikasi perangkat lunak.
- ❖ Asosiasi (*Association*) dan *Link*. Asosiasi mendefinisikan tipe hubungan. *Link* merupakan abstraksi dari hubungan khusus yang diatur dalam kelas.
- ❖ Agregasi (*Aggregation*). Agregasi adalah tipe dari asosiasi yang menyatakan bahwa salah satu memegang kontrol terhadap objek yang lain. Kelas agregat mendefinisikan aturan-aturan bagaimana kelas yang diagregasi berperilaku. Agregasi objek mendistribusikan sifat-sifatnya ke anggota objeknya.
- ❖ Komposisi (*Composition*). Komposisi merupakan tipe suatu agregasi yang menyatakan bahwa objek anggota hanya anggota dari satu agregasi.
- ❖ Generalisasi, spesialisasi dari pewarisan (*inheritance*). Generalisasi adalah kelas yang memiliki fitur yang diwariskan ke kelas-kelas lain. Spesialisasi adalah kelas yang memiliki fitur yang unik terhadap subset kelasnya. Pewarisan merupakan suatu prinsip yang memungkinkan suatu kelas spesialisasi mengakses fitur yang ada pada kelas umumnya.
- ❖ *Polimorfisme, polimorfisme* mengandung makna bahwa suatu operasi yang sama mungkin dijalankan dengan cara/metode (*method*) yang berbeda. Oleh karena itu dalam pemrograman berorientasi objek, dibedakan antara metode dan operasi
- ❖ Kohesi (*Cohesion*). Merupakan ukuran seberapa baik bagian dari suatu objek mendukung kebutuhan tunggal suatu objek.

- ❖ Koping (*Coupling*). Merupakan ukuran ketergantungan antar objek

II.5. Database

Database atau basis data merupakan sekumpulan data yang memiliki hubungan secara logika dan diatur dengan susunan tertentu serta disimpan dalam media penyimpanan komputer. (Wahana Komputer : 2010 : 24).

Data itu sendiri merupakan representasi dari semua fakta yang ada pada dunia nyata. Database sering digunakan untuk melakukan proses terhadap data- data yang lain untuk menghasilkan sebuah informasi tertentu.

Pada era kemajuan teknologi sekarang ini, nilai informasi sangatlah penting bagi kemajuan perusahaan. Oleh karena itu, penggunaan dan penguasaan database sangat penting. Dalam database ada sebutan untuk satuan data yaitu :

- Karakter, ini adalah satuan data terkecil. Data terdiri atas susunan karakter yang pada akhirnya mewakili data yang memiliki arti dari sebuah fakta.
- Field, adalah kumpulan dari karakter yang mewakili fakta tertentu misalnya seperti nama siswa, tanggal lahir dan lain-lain. Dalam dunia perancangan database, field juga disebut atribut. Bila dipandang dari sudut pemrograman berorientasi objek maka sebuah field akan memiliki dua properti utama yaitu properti nama dan properti type. Properti name atau nama adalah properti yang dari field yang mewakili data sejenis yang disimpannya. Sedangkan properti type adalah properti yang mengatur tipe dari data yang akan ditampungnya. Misalnya nama fieldnya adalah nama

siswa maka tipe datanya adalah char, bila nama fieldnya adalah tanggal lahir maka tipe datanya adalah date. Field dilihat seperti kolom.

- Record, adalah kumpulan dari field. Pada record anda akan dapat menemukan banyak sekali informasi penting dengan cara mengkombinasikan field-field yang in.
- Tabel, adalah sekumpulan dari record-record yang memiliki kesamaan entity dalam dunia nyata. Kumpulan dari tabel adalah database, wujud fisik sebuah database dalam komputer adalah sebuah file yang didalamnya terdapat berbagai tingkatan data yang telah diebutkan diatas.
- File, adalah bentuk dari penyimpanan data. File database berisi semua data yang telah disusun dan diorganisasikan sedemikian rupa sehingga memudahkan pemberian informasi.

II.5.1 Sistem Database

Sebuah sistem database adalah sebuah kumpulan dari komponen-komponen database- database yang meliputi :

- Database
- Database Server
- Komponen Client Software
- Aplikasi Database

Aplikasi database adalah sebuah software yang khusus didesain dan digunakan oleh user atau pihak lainnya seperti penyedia jasa pemrograman atau konsultan. Misalnya aplikasi POS, SIA, Sistem Perpustakaan dan lain-lain.

Sedangkan client software adalah salah satu komponen yang termasuk dalam sistem database yang memungkinkan software aplikasi database mengakses data secara remote pada sebuah server database adalah menangani manajemen data. Tiap client software berkomunikasi dengan server untuk menyalurkan permintaan data lewat SQL dan server memprosesnya dalam urutan tertentu dan mengirimkan data tersebut kembali ke client software. Untuk definisi database telah dijelaskan sebelumnya. Kewajiban utama dari sistem database adalah menyediakan antarmuka (interface) kepada user untuk membuat database, dan mengolahnya (mencari, menghapus, mengedit).

II.5.2. Sistem Database Relasional

SQL Server 2008 termasuk salah satu mesin database relasional. Ide tentang sistem database relasional pertama kali dicetuskan oleh seorang yang bernama E.F.Codd lewat sebuah artikel yang berjudul "A Relational Model of Data for Large Shared Data Banks. Sistem Database relational berdasarkan pada metode matematika. Sistem ini menggantikan metode lama yang berdasarkan network dan hierarki. Metode relasional ini bekerja berdasarkan keterkaitan antartabel.

II.5.3. Query Language

Seperti sudah disinggung diatas bahwa untuk mendapatkan informasi tertentu dari sebuah server database user harus mengirimkan permintaan lewat bahasa SQL yang dipahami oleh server database. Demikian juga dengan SQL Server 2008 yang memiliki bahasa permintaan (SQL) sendiri yang disebut Transact-SQL.

Bahasa SQL berawal dari sebuah proyek dengan kode sistem R yang diprakarsai oleh IBM pada tahun 1980. Proyek ini membuktikan bahwa mungkin saja membuat sebuah sistem relasional database berdasarkan pendekatan yang diperkenalkan oleh E.F.Codd.

Dengan bahasa permintaan ini user dapat melakukan permintaan terhadap lebih dari satu/ sekumpulan record pada database. Bukan itu saja user dimungkinkan untuk melakukan permintaan data pada sekumpulan record-record baik yang ada pada satu tabel atau bahkan dengan tabel lainnya dan itu dapat dilakukan dengan hanya menggunakan sebuah statement SQL.

SQL dapat dibedakan oleh dua jenis subbahasa, yaitu :

- Data Definition Language (DDL), bahasa ini digunakan untuk “membangun”struktur database. Contoh dari bahasa ini adalah CREATE, DROP, ALTER. Bahasa ini Dikenakan pada database, kolom (field), dan index.
- Data manipulation language (DML), jenis SQL ini berfungsi unruk melakukan manaipulasi terhadap data yang ada seperti record, field. Contoh perintahnya adalah DELETE, UPDATE, INSERT, dan yang paling terkenal adalah SELECT.

II.5.4. Analisa Sistem

Analisa sistem adalah sebuah proses penelaahan sebuah sistem informasi dan membaginya kedalam komponen-komponen penyusunnya sehingga diketahui permasalahan-permasalahan serta kebutuhan-kebutuhan yang akan timbul,

sehingga dapat dilaporkan secara lengkap serta diusulkan perbaikan-perbaikan pada sistem tersebut.

Tahap-tahap analisa sistem

Pada analisa sistem dikenal beberapa tahap, yaitu:

1. Identifikasi masalah yang ada pada sistem informasi tersebut
2. Memahami cara kerja sistem
3. Melakukan sistem
4. Melaporkan hasil analisa sistem.

II.6. SQL Server 2008 Express

Merupakan DBMS (Database Management System) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengolahan data agar dapat menyusul pendahulunya seperti IBM dan Oracle (Wahana Komputer : 2010 ; 3).

SQL Server 2008 dibuat saat kemajuan dalam bidang hardware sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa SQL Server 2008 membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data,

II.6.1. Versi- Versi dari SQL Server 2008 Express

Microsoft meriis SQL Server 2008 dalam beberapa versi yang disesuaikan demgan segemen-segmen pasar yang dituju, Versi-versi tersebut adalah menurut cara pemrosesan data pada prosesor maka Microsoft mengelompokkan produk ini berdasarkan 2 jenis yaitu:

- Versi 32-bit(x86), yang biasanya digunakan untuk komputer dengan single processor (Pentium 4) atau lebih tepatnya processor 32 bit dan sistem operasi Windows XP.
- Versi 64 bit(x64), yang biasanya digunakan untuk komputer dengan lebih dari satu processor (misalnya: core2 duo) dan sistem operasi 64 bit seperti Windows XP 64, Vista, dan Windows 7.

II.7. Microsoft Visual Studio 2010

Visual Basic 2010 merupakan salah satu bagian dari produk pemrograman terbaru yang dikeluarkan oleh Microsoft, yaitu Microsoft Visual Studio 2010, Visual Studio 2010 merupakan produk pemrograman andalan dari Microsoft Corporation, dimana didalamnya berisi beberapa jenis IDE pemrograman seperti Visual Basic, Visual C++, Visual Web Developer, Visual C#, DAN Visual F3.

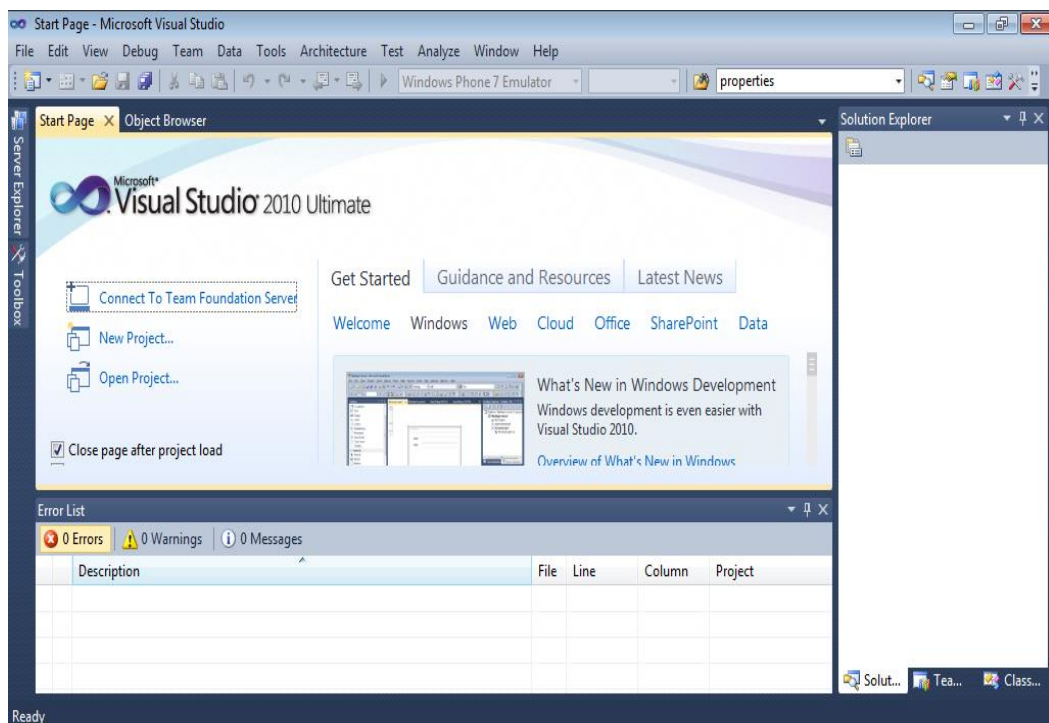
Semua IDE pemrograman tersebut sudah mendukung penuh implementasi .Net Framework terbaru, yaitu .Net Framework 4.0 yang merupakan pengembangan dari .Net 3.5. Adapun database standar yang digunakan adalah Microsoft SQL Server 2008 R2.

Visual Basic 2010 merupakan versi perbaikan dan pengembangan dari versi pendahulunya, yaitu visual basic 2008. Beberapa pengembangan yang terdapat didalamnya antara lain dukungan terhadap library terbaru dari Microsoft, yaitu Net. Framework 4.0, dukungan terhadap pengembangan aplikasi menggunakan Microsoft SilverLight dukungan terhadap aplikasi berbasis Cloud Computing,

serta perluasan dukungan terhadap database-database, baik standalone maupun database server.

Bahasa Visual Basic 2010 sendiri awalnya berasal dari bahasa pemrograman yang sangat populer di kalangan programmer komputer, yaitu bahasa Basic, yang oleh Microsoft diadaptasi dalam program Microsoft Quick Basic. Seiring dengan berkembangnya teknologi komputasi dan desain, Microsoft mengeluarkan produk yang dinamakan Microsoft Visual Studio dengan Visual Basic didalamnya. Saat ini versi Microsoft Visual Studio yang beredar adalah versi 10 yang populer dengan nama Microsoft Visual Studio 2010, yang didalamnya termasuk Microsoft Visual Basic 2010. (Wahana Komputer ; 2011 : 2-3).

Adapun tampilan awal dari Microsoft Visual Studio 2010 adalah sebagai berikut :



Gambar II.5. Microsoft Visual Studio 2010

Sumber (Wahana Komputer : 2011 ; 3)

