

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu.

Pendekatan sistem memberikan banyak manfaat dalam memahami lingkungan kita. Pendekatan sistem berusaha menjelaskan sesuatu yang dipandang dari sudut pandang sistem serta berusaha menemukan struktur unsur yang membentuk sistem tersebut. Setiap sistem merupakan bagian dari sistem lain yang lebih besar dan terdiri dari berbagai sistem yang lebih kecil yang disebut subsistem.

Dari uraian diatas dapat diambil kesimpulan bahwa suatu sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu.

Menurut James A. Hall (2009:6) Sistem adalah kelompok dari dua atau lebih komponen atau subsitem yang saling berhubungan yang berfungsi dengan tujuan yang sama.

Menurut Asbon Hendra (2012:157) Sistem merupakan kumpulan unsure atau elemen-elemen yang saling berkaitan/berinteraksi dan saling memengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu.

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:

a. Komponen (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan.

b. Batas Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

c. Lingkungan Luar Sistem (*Environment*)

Environment merupakan segala sesuatu di luar batas sistem yang memengaruhi operasi dari suatu sistem

d. Penghubung Sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya untuk membentuk satu kesatuan sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem yang lainnya.

e. Masukan Sistem (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa Masukan Perawatan (*Maintenance Input*) adalah energi yang dimasukan

supaya sistem tersebut dapat beroperasi. Masukan sinyal (*Signal Input*) adalah energi yang diproses untuk didapatkan keluaran.

f. Keluaran Sistem (*Output*)

Merupakan hasil dari energi yang diolah oleh sistem, meliputi *output* yang berguna.

g. Pengolah Sistem (*Process*)

Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan.

h. Tujuan Sistem (*Goal*)

Setiap *sistem* pasti mempunyai tujuan ataupun sasaran yang memengaruhi *input* yang dibutuhkan dan *output* yang dihasilkan. Dengan kata lain, suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenai sasaran atau tujuannya.

II.1.2. Klasifikasi Sistem

Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi yang ada di dalam sistem tersebut. Oleh karena itu, sistem dapat diklasifikasikan dari beberapa sudut pandangan, diantaranya:

a. Sistem Abstrak (*Abstract System*)

Sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik.

b. Sistem Fisik (*Physical System*)

Merupakan sistem yang ada secara fisik sehingga setiap makhluk dapat melihatnya.

c. Sistem Alamiah (*Natural System*)

Sistem yang terjadi melalui proses alam, dalam artian tidak dibuat oleh manusia.

d. Sistem Buatan Manusia (*Human Made System*)

Sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut *machine system*.

e. Sistem Tertentu (*Deterministic System*)

Beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan.

f. Sistem Tak Tentu (*Probabilistic System*)

Sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

g. Sistem Tertutup (*Closed System*)

Sistem yang tidak berhubungan dan tidak terpengaruh dengan sistem luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya.

h. Sistem Terbuka (*Open System*)

Sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya.

II.2. Informasi

Menurut Tata Sutabri (2012:29) Informasi adalah sebuah istilah yang tepat dalam pemakaian umum. Informasi dapat mengenai data mentah, data tersusun, kapasitas sebuah saluran komunikasi, dan lain sebagainya. Informasi ibarat darah

yang mengalir di dalam tubuh suatu organisasi sehingga informasi ini sangat penting di dalam organisasi.

Menurut Asbon Hendra (2012:167) Informasi merupakan data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. Jadi, ada suatu proses transformasi data menjadi suatu informasi = input – proses – *output*.

II.3. Sistem Informasi

Suatu sistem informasi dapat di definisikan secara teknis sebagai satuan komponen yang saling berhubungan yang mengumpulkan atau mendapatkan kembali, memproses, menyimpan, dan mendistribusikan informasi untuk mendukung pengambilan keputusan dan kendali dalam suatu organisasi. Sistem informasi berisi informasi tentang orang-orang tertentu, tempat-tempat, dan hal-hal di dalam organisasi atau di lingkungan sekitarnya.

Menurut James A. Hall (2009:9) Informasi adalah serangkaian prosedur formal di mana data dikumpul, diproses, menjadi informasi dan didistribusi ke para pengguna. Tata Sutabri (2012:46) menyatakan, sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.

II.4. Pengertian Penjualan

Penjualan adalah suatu sistem keseluruhan dari kegiatan usaha yang ditujukan untuk merencanakan, menentukan harga, mempromosikan dan mendistribusikan barang, jasa, ide kepada pasar sasaran agar dapat mencapai tujuan organisasi (*Indonesia Jurnal on Networking and Security – Volume 3 No 1 Januari 2014;24:Sugyanto*).

II.5. Visual Basic Net

A. M. Hirin (2011:2) menyatakan, *Visual Basic* adalah salah satu bahasa pemrograman berbasis desktop yang dikeluarkan oleh perusahaan perangkat lunak komputer terbesar yaitu *Microsoft*. *Visual Basic* merupakan salah satu bahasa pemrograman paling laris dan paling sukses di dunia. Dimana tercatat sampai pada tahun 2005 *Visual Basic* merupakan bahasa pemrograman yang paling banyak dipakai oleh para programmer bahkan diyakini sampai saat ini. Menjadi pilihan berbagai kalangan tentunya *Visual Basic* memiliki berbagai hal yang patut dijadikan alasan, selain bahasa pemrograman yang sangat mudah dipelajari oleh berbagai kalangan baik awam maupun ahli, *Visual Basic* yang didukung penuh oleh produsennya (*Microsoft*) selalu dikembangkan dan disesuaikan dengan kebutuhan zaman seperti penyesuaian model pemrograman modern yang berbasis OOP (*Object Oriented Programming*).

II.5.1 Sejarah Visual Basic Net

Setelah *Visual Basic* versi 6.0, *Microsoft* melakukan perubahan besar pada bahasa pemrograman *Visual Basic* versi selanjutnya. Dimana ditambahkan suatu

pustaka-pustaka yang terangkai menjadi suatu kesatuan disebut .net (dotnet) *framework*. Selain itu ditambah pula permodelan pemrograman berorientasi objek yang disebut *Object Oriented Programming* atau sering disingkat dengan OOP.

Tabel II.1 Versi Visual Basic

Versi	Tahun Rilis	.Net Framework
Visual Basic 1.0	Mei, 1991	-
Visual Basic 2.0	November, 1992	-
Visual Basic 3.0	Juni, 1993	-
Visual Basic 4.0	Agustus, 1995	-
Visual Basic 5.0	Februari, 1995	-
Visual Basic 6.0	Oktober, 1998	-
Visual Basic 7.0 (vb.net 2002)	Februari, 2002	Framework 1.0
Visual Basic 7.1 (vb.net 2003)	April, 2003	Framework 1.5
Visual Basic 8.0 (vb.net 2005)	Oktober, 2005	Framework 2.0
Visual Basic 9.0 (vb.net 2008)	November, 2007	Framework 3.5
Visual Basic 10 (vb.net 2010)	April, 2010	Framework 4.0

Sumber: A. M. Hirin (2011:3)

Pada versi *Visual Basic* 2002 dan seterusnya, semua program *Visual Basic* berjalan dengan *framework* sesuai versi masing-masing. Hal ini tentunya memiliki berbagai dampak baik dampak positif maupun negatif. Keuntungan dengan adanya *framework* ini pembuat program dari *Visual Basic* terkesan lebih mudah dan singkat karena dalam *framework* telah terbungkus berbagai komponen dan

class yang siap pakai, sehingga kita tidak perlu menulis kode yang terlalu panjang untuk melakukan berbagai fungsi tertentu.

Dimana hal ini sangat berbeda dengan versi *Visual Basic* sebelumnya yang tidak disertai dengan *class* yang siap saji melalui *framework*. Kekurangannya tentu saja program yang dibuat akan memakan lebih banyak tempat (memori) baik memori hardisk maupun memori RAM karena *framework* juga harus didistribusi dengan terpasang pada komputer target. Akan tetapi dengan melihat spesifikasi komputer-komputer saat ini hal tersebut sudah tidak perlu dipermasalahkan lagi.

II.6. SQL Server

Menurut Agus Saputra (2013:11) Microsoft SQL Server merupakan salah satu produk RDBMS (*Relational database management system*). Merupakan salah satu produk andalan yang dibuat oleh Microsoft yang berfungsi sebagai relational database. Entah mengapa nama Microsoft SQL Server tersebut, lebih mengena dengan sebutan SQL Server. Mungkin memang karna penyebutannya sudah tak asing lagi di telinga kita. Penulis pun kadang-kadang suka menyebutnya dengan SQL Server.

Microsoft SQL Server sering mendukung SQL sebagai bahasa pemroses query. Seperti yang kita ketahui, SQL merupakan bahasa standar internasional untuk proses query database dan SQL ini sudah banyak sekali digunakan pada hampir semua aplikasi, baik itu e-commerce, pendidikan, organisasi, pemerintah, atau bahkan personal sekalipun. Sehingga dengan demikian, SQL Server patut kita pertimbangkan sebagai database program kita.

II.6.1. Sejarah SQL Server

Menurut Agus Saputra (2013: 12) pada tahun 1988, Microsoft mengeluarkan versi pertama dari SQL Server. Pada saat itu, SQL Server masih didesain untuk platform OS/2 dan dikembangkan bersama antara Microsoft dengan Sybase.

Pada awal tahun 1990-an, Microsoft mulai untuk membuat versi baru dari SQL Server untuk platform NT-nya. Selama proses perkembangan tersebut, Microsoft memutuskan bahwa Microsoft SQL Server harus bisa terintegrasi kuat dengan operasi NT-nya. Akhirnya pada tahun 1993, Windows NT 3.1 dan SQL Server 4.2. Untuk NT dirilis oleh Microsoft. Target Microsoft untuk melakukan kombinasi antara performa database server yang tinggi serta kemudahan cara penggunaan dan administrasi tercapai melalui SQL Server tersebut.

Microsoft terus berhasil memasarkan SQL Server dan menjadi database yang cukup populer saat itu. Pada tahun 1994, Microsoft dan Sybase secara resmi mengakhiri kerja samanya. Kemudian di tahun 1995, Microsoft kembali merilis versi 6.0 dari SQL Server. Versi ini merupakan versi yang paling penting karena sebagian besar merupakan hasil tulis ulang dan juga merupakan hasil desain ulang dari teknologi mesin sebelumnya. Versi 6.0 ini menawarkan peningkatan pada performa administrasi yang tersentralisasi.

Pada tahun 1996, Microsoft merilis SQL Server 6.5 dan pada tahun 1998 Microsoft merilis kembali SQL Server versi 7.0 (teknologi database-nya kembali ditulis ulang agar lebih optimal). Akhirnya pada tahun 2000, Microsoft mengeluarkan SQL Server 2000 yang merupakan versi yang paling banyak

digunakan. Versi SQL Server 2000 ini berbasis *framework* yang ada pada versi 7.0 sebelumnya. Sedangkan versi terbaru dan yang terakhir saat ini adalah SQL Server 2008, dengan penambahan berbagai fitur-fitur yang dapat memudahkan pengguna membuat suatu database maupun table-tabel yang akan dirancang.

II.7. UML (*Unified Modelling Language*)

Unified Modelling Language (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industry untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak (*Jurnal Teknologi Informasi DINAMIK Volume XIV, No. 1, Januari 2009;23-29:Prastuti Sulistyorini*).

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek. (*Jurnal Informatika Mulawarman Vol 6 No. 1 Februari 2011;1:Haviluddin*).

Sejarah UML terbagi dalam dua fase, sebelum dan sesudah munculnya UML. Dalam fase sebelum, UML sebenarnya sudah mulai diperkenalkan oleh para ahli analisis dan desain berbeda-beda sehingga dapat dikatakan belum memiliki standarisasi. Fase kedua, dilandasi dengan pemikiran untuk mempersatukan metode tersebut dan dikembangkan oleh *Object Management Group* (OMG) maka pengembangan UML dimulai pada akhir tahun 1994 ketika Grady Booch dengan metode OOD (*Object-Oriented Design*), Jim Rumbaugh

dengan metode OMT (*Object Modelling Technique*) mereka ini bekerja pada Rational Software dan Ivar Jacobson dengan metode OOSE (*Object-Oriented Software Engineering*) yang bekerja pada perusahaan Objectory Rational.

Sebagai pencetus metode-metode tersebut mereka bertiga berinisiatif untuk menciptakan bahasa pemodelan terpadu sehingga pada tahun 1996 mereka berhasil merilis UML versi 0.9 dan 0.91 melalui *Request for Proposal* (RFP) yang dikeluarkan oleh OMG. Kemudian pada Januari 1997 IBM, ObjectTime, Platinum Technology, Ptech, Taskon, Reich Technologies dan Softteam juga menanggapi *Request for Proposal* (RFP) yang dikeluarkan oleh OMG tersebut dan menyatakan kesediaan untuk bergabung.

Perusahaan-perusahaan ini menyumbangkan ide-ide mereka, dan bersama para mitra menghasilkan UML revisi 1.1. Fokus dari UML versi rilis 1.1 ini adalah untuk meningkatkan kejelasan UML Semantik versi rilis 1.0. Hingga saat ini UML versi terbaru adalah versi 2.0. Saat ini sebagian besar para perancang sistem informasi dalam menggambarkan informasi dengan memanfaatkan UML diagram dengan tujuan utama untuk membantu tim proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuatan program.

Secara filosofi UML dilengkapi oleh konsep yang telah ada yaitu konsep permodelan *Object Oriented* karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik.

II.7.1. *Use Case*

Use case adalah deskripsi fungsi dari sebuah sistem perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut skenario. *Use case* merupakan awal yang sangat baik untuk setiap fase pengembangan berbasis objek, design, testing, dan dokumentasi yang menggambarkan kebutuhan sistem dari sudut pandang di luar sistem. Perlu diingat bahwa *use case* hanya menetapkan apa yang seharusnya dikerjakan oleh sistem, yaitu kebutuhan fungsional sistem dan tidak untuk menentukan kebutuhan nonfungsional, misalnya: sasaran kinerja, bahasa pemrograman dan lain sebagainya. (*Jurnal Informatika Mulawarman Vol 6 No. 1 Februari 2011;6:Haviluddin*).

II.7.2. *Class Diagram*

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class diagram* membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class diagram* berperan dalam struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. (*Jurnal Informatika Mulawarman Vol 6 No. 1 Februari 2011;3:Haviluddin*).

Class diagram membantu dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak. *Class diagram*

memperlihatkan hubungan antara kelas dan penjelasan detail tiap-tiap kelas di dalam model desain (dalam *logical view*) dari suatu sistem. Selama proses analisis, *class diagram* memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Selama proses analisis, *class diagram* memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama tahap desain, *class diagram* berperan dalam membentuk arsitektur sistem yang dibuat. (*Jurnal Teknologi Informasi DINAMIK Volume XIV, No. 1, Januari 2009;26-29:Prastuti Sulistyorini*).

II.7.3. Sequence Diagram

Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *Sequence Diagram* adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram*. (*Jurnal Informatika Mulawarman Vol 6 No. 1 Februari 2011;5:Haviluddin*).

Sequence diagram menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus berasosiasi dengan *use case Sequence diagram* memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu di dalam *use case*. *Sequence diagram* sebaiknya digunakan di awal tahap desain atau analisis karena kesederhanaannya dan mudah untuk dimengerti. (*Jurnal Teknologi Informasi DINAMIK Volume XIV, No. 1, Januari 2009;27-29:Prastuti Sulistyorini*).

II.7.4. Activity Diagram

Activity Diagram memodelkan alur kerja (*workflow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses. Diagram ini sangat mirip dengan sebuah flowchart karena dapat dimodelkan sebuah alur kerja dari satu aktivitas ke aktivitas lainnya atau dari satu aktivitas ke dalam keadaan sesaat (*state*). Seringkali bermanfaat bila dibuat sebuah *activity* terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan. *Activity diagram* juga sangat berguna ketika ingin menggambarkan perilaku parallel atau menjelaskan bagaimana perilaku dalam berbagai *use case* berinteraksi. Dapat digunakan *statechart diagram* untuk memodelkan perilaku dinamis satu kelas atau objek. *Statechart diagram* memperlihatkan urutan keadaan sesaat (*state*) yang dilalui sebuah objek, kejadian yang menyebabkan sebuah transisi dari satu *state* atau aktivitas ke *state* aktivitas lainnya, dan aksi yang menyebabkan perubahan satu *state* atau aktivitas. Diagram aktivitas paling cocok digunakan untuk memodelkan urutan aktivitas dalam suatu proses. (*Jurnal Teknologi Informasi DINAMIK Volume XIV, No. 1, Januari 2009;28-29:Prastuti Sulistyorini*).