

BAB II

TINJAUAN PUSTAKA

II.1 Teori Sistem

Menurut Tata Sutabri (2012:3), Suatu Sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisasi, saling berinteraksi, saling bergantung satu sama lain, dan terpadu. Secara garis besar terdapat dua kelompok pendekatan yang dilakukan yaitu:

1. Pendekatan sistem yang lebih menekankan pada elemen-elemen atau kelompoknya, didalam hal ini sistem ini didefinisikan sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.
2. Pendekatan sistem sebagai jaringan kerja dari prosedur, yang lebih menekankan urutan operasi didalam sistem. Prosedur didefinisikan sebagai urutan operasi kerja (tulis-menulis), yang biasanya melibatkan beberapa orang didalam satu atau lebih departement yang diterapkan untuk menjamin penanganan yang seragam dari transaksi bisnis yang terjadi.

Pendekatan sistem yang lebih menekankan pada elemen-elemen atau komponennya mendefinisikan sistem sebagai sekumpulan elemen-elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Dengan demikian didalam suatu sistem, komponen-komponen ini tidak dapat

berdiri sendiri, tetapi sebaliknya, saling berhubungan hingga berbentuk suatu kesatuan hingga tujuan sistem dapat tercapai.

II.1.1 Karakteristik Sistem

Menurut Tata Sutabri (2012:13), sistem mempunyai beberapa karakteristik atau sifat-sifat tertentu antara lain:

1. Komponen sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama untuk membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian dari sistem. Setiap sistem tidak peduli berapapun kecilnya, selalu mengandung komponen-komponen atau subsistem.

2. Batasan sistem (*Boundary*)

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

3. Lingkungan luar sistem (*Environment*)

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung sistem (*Interface*)

Merupakan media penghubung antara suatu subsistem dengan subsistem yang lainnya. Melalui perhubungan ini memungkinkan sumber-sumber daya mengalir dari subsistem yang lainnya.

5. Masukkan sistem (*input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

6. Keluaran sistem (*output*)

Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah sistem (*Process*)

Suatu sistem yang dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran sistem (*Objective*)

Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya, suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

II.1.2 Klasifikasi Sistem

Menurut Tata Sutabri (2012:15), sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya sebagai berikut:

1. Sistem abstrak dan sistem fisik

Sistem abstrak adalah sistem yang berisi gagasan atau konsep. Misalnya, sistem teknologi yang berisi gagasan tentang hubungan manusia dan Tuhan. Sistem fisik merupakan sistem yang secara fisik dapat dilihat. Misalnya sistem komputer, sistem sekolah, sistem akuntansi, dan sistem transportasi.

2. Sistem alamiah dan sistem buatan manusia

Sistem alamiah adalah sistem yang terjadi karena alam (tidak dibuat manusia). Misalnya, sistem tata surya. Sistem buatan manusia adalah sistem yang dibuat oleh manusia. Misalnya, sistem komputer dan sistem mobil.

3. Sistem tertentu dan sistem tak tertentu

Sistem tertentu adalah sistem yang operasinya dapat diprediksi secara tepat. Misalnya, sistem komputer. Sistem tak tertentu adalah sistem yang tidak dapat diramal dengan pasti karena mengandung unsur probabilitas. Misalnya, sistem arisan dan sistem sediaan.

4. Sistem tertutup dan sistem terbuka

Sistem tertutup adalah sistem yang tidak bertukar materi, informasi, atau energi dengan lingkungannya. Misalnya, reaksi kimia dalam tabung terisolasi. Sistem terbuka adalah sistem yang berhubungan dengan lingkungan dan diperbaharui oleh lingkungan.

II.2. Konsep Dasar Informasi

Tata sutabri (2012:22), Informasi adalah data yang telah diklasifikasi atau diolah atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan sistem mengolah data menjadi informasi atau tepatnya mengolah data dari bentuk tak berguna menjadi berguna bagi penerimanya. Teori informasi lebih tepat disebut teori matematika, komunikasi yang juga memberikan beberapa pandangan yang berguna bagi sistem informasi. Sumber sistem informasi adalah data, data merupakan kenyataan yang menggambarkan suatu

kejadian serta merupakan suatu kesatuan yang nyata merupakan suatu bentuk yang masih mentah yang belum menghasilkan suatu informasi.

II.2.1 Kualitas Informasi

Informasi yang berkualitas memiliki 3 kriteria, yaitu:

1. Akurat (*accurate*)

Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan.

Akurat juga berarti informasi harus jelas mencerminkan maksudnya.

2. Tepat waktu (*Timelines*)

Informasi yang datang pada si penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi karena informasi merupakan dalam pengambilan keputusan.

3. Relevan (*relevance*)

Informasi tersebut mempunyai manfaat untuk pemakaian. Informasi yang disampaikan harus mempunyai keterkaitan dengan masalah yang akan dibahas.

II.3. Konsep Dasar Sistem Informasi

Tata Sutabri (2012:38), Sistem Informasi adalah suatu sistem didalam sistem 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 tertentu dengan laporan-laporan yang diperlukan. Defenisi umum sistem informasi adalah “ Sebuah sistem yang terdiri atas rangkaian subsistem informasi terhadap

pengolahan data untuk menghasilkan informasi yang berguna dalam pengambilan keputusan.”

II.3.1 Komponen Sistem Informasi

Tata Sutabri (2010:39), dalam suatu sistem informasi terdapat komponen-komponen yang saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran.

1. Blok Masukkan (*input block*)

Input mewakili data yang masuk kedalam sistem informasi. Input termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

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 basisdata dengan cara sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*outpu Block*)

Merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakaian sistem.

4. Blok teknologi (*technology block*)

Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, dan membantu pengendalian diri dari sistem secara keseluruhan. Teknologi terdiri dari tiga bagian utama, yaitu teknisi (*brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

5. Blok basis data (*database block*)

Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan diprangkat keras komputer menggunakan perangkat lunak untuk memanipulasi.

6. Blok kendali (*control block*)

Pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem seperti, kecurangan-kecurangan, kegagalan sistem itu sendiri, debu dan air, dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung diatasi.

II.4. Sistem Pendukung Keputusan (SPK)

II.4.1 Pengertian Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah (2014:1), Sistem pendukung keputusan (SPK) dibangun untuk mendukung solusi atas suatu masalah atau untuk suatu peluang. Aplikasi Sistem pendukung keputusan (SPK) digunakan untuk pengambilan keputusan. Aplikasi Sistem pendukung keputusan (SPK) menggunakan CBIS (*Computer Based Information System*) yang fleksibel, interaktif dan dapat diadaptasi yang dikembangkan untuk mendukung solusi masalah manajemen spesifik yang tidak terstruktur.

Menurut Banzek dkk, (1980) dalam buku "*Decission Support System and Intelligent System*" (Turban 2005:137) mendefenisikan Sistem pendukung keputusan (SPK) sebagai sistem berbasis komputer yang terdiri dari tiga komponen yang salin berinteraksi, sistem bahasa (mekanisme untuk memberikan komunikasi antara pengguna dan komponen sistem pendukung keputusan lain),

sistem pengetahuan (repository pengetahuan domain masalah yang ada pada Sistem pendukung keputusan (SPK) atau sebagai data atau sebagai prosedur), dan sistem pemrosesan masalah (hubungan antara dua komponen lainnya, terdiri dari satu atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan).

II.4.2 Karakteristik Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah (2014:2), karakteristik sistem pendukung keputusan (SPK) yaitu:

1. Mendukung proses pengambilan keputusan suatu organisasi atau perusahaan
2. Adanya *interfse* manusia/mesin dimana manusia (*user*) tetap memegang kontrol proses pengambilan keputusan
3. Mendukung pengambilan keputusan untuk membahas masalah terstruktur, semi terstruktur serta mendukung beberapa keputusan yang saling berinteraksi.
4. Memiliki kapasitas dialog untuk memperoleh informasi sesuai dengan kebutuhan.
5. Memiliki sub sistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan sistem.

II.4.3 Ciri – ciri Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah (2014:2), kriteria atau ciri - ciri sistem pendukung keputusan (SPK) yaitu:

1. Banyak pilihan / alternatif
2. Ada kendala / surat
3. Mengikuti suatu pola atau model tingkah laku, baik yang terstruktur maupun tidak terstruktur
4. Banyak input / variabel
5. Ada faktor resiko, dibutuhkan kecepatan, ketepatan dan keakuratan

II.5. Display Contest

Display contest merupakan kreatifitas dalam *mendisplay product*. Para peserta *display contest* saling berlomba dalam penyajian, penampilan, dan tata letak barang. Sehingga dibutuhkan kreatifitas dalam menghias dan menyajikan tampilan *display* pada sebuah *product*, yang tentunya didukung oleh material serta property yang difasilitasi oleh produsen selaku penyelenggara *display contest*. Dengan mengandalkan kreatifitas serta ide-ide dalam kreasi yang ditampilkan sesuai dengan tema yang ditentukan oleh penyelenggara dan juga penjualan mencapai hasil yang maksimal diharapkan menjadi pertimbangan para juri *display contest* untuk menentukan siapa yang layak menyandang predikat juara dalam *display contest*. (CV.Citra Hanoch Niagantara).

II.6. Algoritma *K-Nearest Neighbor* (K-NN)

Menurut Nobertus Krisandi (2013;34), Klasifikasi merupakan proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui. Model itu sendiri bisa berupa aturan “jika-maka”, berupa

decision tree, formula matematis atau neural network. Metode – metode klasifikasi antara lain C4.5, *RainForest*, *Naive Bayesian*, *Neural Network*, *Genetic Algorithm*, *fuzzy*, *case-based reasoning*, dan *k-Nearest Neighbor*.

Algoritma K-NN adalah suatu metode yang menggunakan algoritma *supervised*. Perbedaan antara *supervised learning* dengan *unsupervised learning* adalah pada *supervised learning* bertujuan untuk menemukan pola baru dalam data dengan menghubungkan pola data yang sudah ada dengan data yang baru. Sedangkan pada *unsupervised learning*, data belum memiliki pola apapun, dan tujuan *unsupervised learning* untuk menemukan pola dalam sebuah data. Tujuan dari algoritma K-NN adalah untuk mengklasifikasi objek baru berdasarkan atribut dan *training sample*. Dimana hasil dari sampel uji yang diklasifikasikan berdasarkan pada memori. Algoritma K-NN menggunakan klasifikasi ketetanggan sebagai nilai prediksi dari sampel uji yang baru. Jarak yang digunakan adalah jarak *Euclidian Distance*. Jarak *Euclidian* adalah jarak yang paling umum digunakan pada data numerik. *Euclidian Distance* didefinisikan sebagai berikut:

$$d(a_r, b_r) = \sqrt{\sum_{r=1}^n (a_r - b_r)^2}$$

Keterangan:

D : Jarak kedekatan

a : data training

b : data testing

n : jumlah atribut individu antara 1 s.d n

: fungsi *similitary* atribut antara kasus A dan kasus B

r : Atribut individu antara 1 sampai dengan

II.6.1 Langkah K-Nearest Neighbor

Langkah-langkah untuk menghitung metode *K-Nearest Neighbor* antara lain :

1. Menentukan parameter K , K = jumlah tetangga yang terdekat
2. Hitung antara jarak data baru dengan semua data training
3. Urutkan jarak tersebut dan tetapkan tetangga terdekat berdasarkan minimum ke – K
4. Periksa kelas dari tetangga terdekat

II.6.2 Kelebihan K-Nearest Neighbor

Berikut ini adalah kelebihan yang dari metode *K-Nearest Neighbor*, yaitu :

1. Lebih efektif didata *training* yang lebih besar.
2. Dapat menghasilkan data yang lebih akurat.

II.7. Microsoft Visual Studio 2010

Wahana Komputer (2012:2), Microsoft Visual Studio adalah sebuah *Integrated Development Environment* buatan Microsoft Corporation. Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan diatas windows) ataupun *managed code* (dalam bentuk microsoft *Intermediate Language* di atas .NET *Framework*). Selain itu, Visual Studio juga dapat mengembangkan aplikasi *Silverlight*, aplikasi *windows mobile* (yang berjalan diatas .NET *Compact Framework*). Visual Studio mencakup sebuah kode editor yang didukung oleh fitur *intellisense* atau yang disebut dengan *code refactoring*. *Debugger* telah terintegrasi bekerja pada level *source level debugger* dan level *debugger* mesin.

Tool built in mencakup form *desainer* untuk membangun sebuah aplikasi GUI, *web desainer*, *class desainer* dan database *schema desainer*.

Microsoft Visual Studio mendukung bahasa pemrograman yang berbeda. Adapun bahasa pemrograman yang didukung oleh visual studio adalah visual C++, Visual Basic, Visual C#. Visual studio juga dapat mendukung bahasa pemrograman lain seperti M, *python* dan *ruby* yang semuanya itu terdapat pada *pack extra* yang terpisah dari visual studio.

II.7.1 Manfaat Visual Studio 2010

Beberapa kemampuan atau manfaat dari Visual Studio 2010 diantaranya seperti:

1. Untuk membuat program aplikasi berbasis windows
2. Untuk membuat objek-objek pembantu program seperti, misalnya : kontrol *ActiveX*, *file Help*, aplikasi Internet dan sebagainya.
3. Menguji program (*debugging*) dan menghasilkan program berakhiran *EXE* yang bersifat *executable* atau dapat langsung dijalankan

II.7.2 Antar Muka Visual Studio 2010

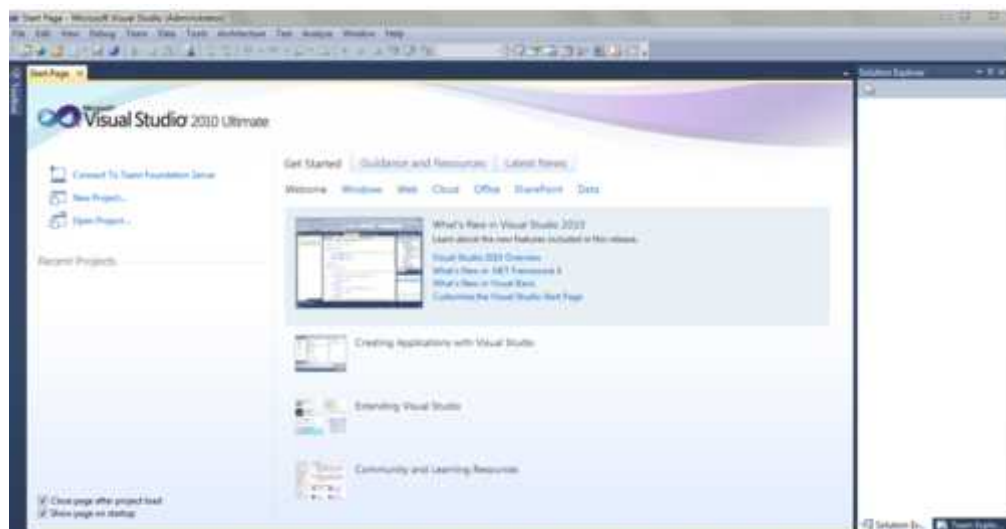
Edy Winarno (2010:2), Saat menjalankan *Visual Studio 2010* pertama kali muncul jendela *chose default environment setting*.disini bisa memilih apakah ingin memilih antar muka di visual studio. Untuk *programmer* visual studio lebih baik memilih *Visual Basic Development center*.

1. Pada saat pertama membuka visual studio akan diminta memilih lingkungan pemrograman, Untuk *programmer* visual studio lebih baik memilih *Visual Basic Development center setting*.



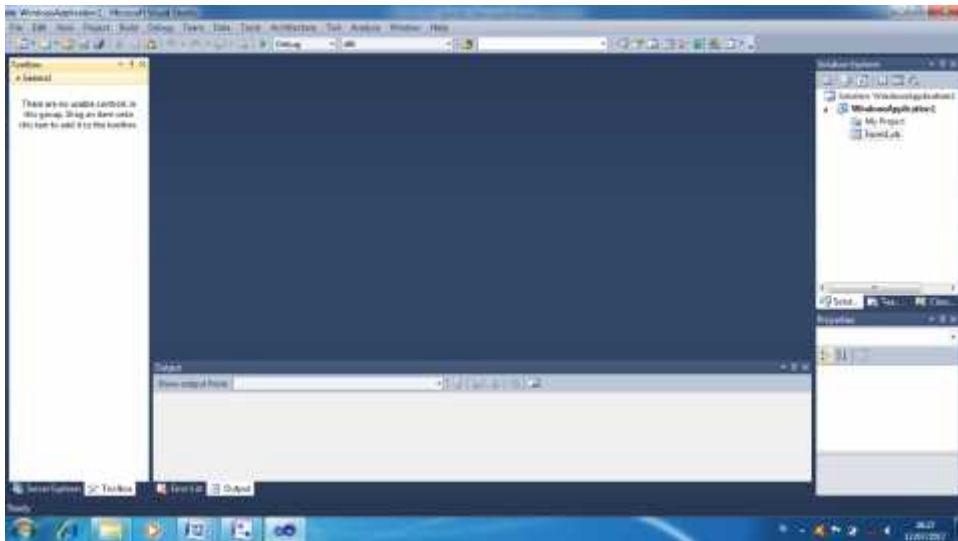
**Gambar II.1. Pemulihan Default Environment Setting
(Sumber : Edy Winarno : 2010 : 2)**

2. Kemudian akan ditampilkan jendela start up visual studio, pilih new project



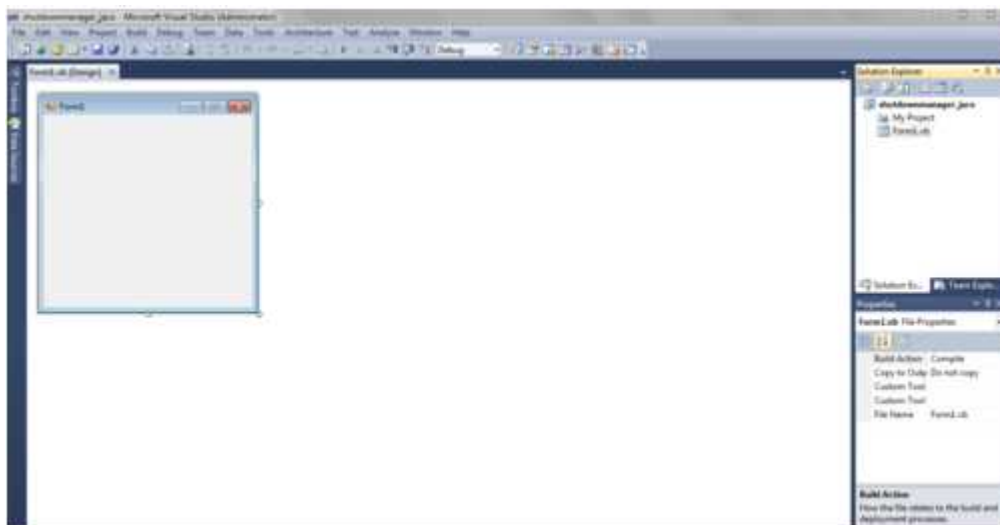
**Gambar II.2. Start Page di Visual Studio 2010
(Sumber : Edy Winarno : 2010 : 3)**

3. Jika *start page* ditutup, terlihat tampilan seperti berikut ini:



**Gambar II.3. Tampilan IDE (*Integrated Development Environment*)
Setelah *Start Page* Tertutup
(Sumber : Edy Winarno : 2010 : 3)**

4. Kemudian akan ditampilkan Form seperti dibawah ini,berikutnya akan di jelaskan bagian-bagian penting .di bagian kiri ada toolbox , dan di kanan ada solution explorer dan properties. solution explorer berisi semua file di myproject, dan bagian properties berguna untuk mengatur nilai-nilai atribut.



Gambar II.4. Tampilan Lengkap IDE
(Sumber : Edy Winarno : 2010 : 3)

II.8. Microsoft *SQL Server 2008 R2*

Wahana Komputer (2008:2), Microsoft *SQL Server 2008 R2* adalah sebuah terobosan baru dari Microsoft *SQL Server 2008 R2* dalam bidang database. Microsoft *SQL Server 2008 R2* adalah DBMS (Database Management System) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam dunia pengolahan data menyusul pendahuluannya seperti IBM dan Oracle. Microsoft *SQL Server 2008 R2* dibuat pada saat kemajuan dalam bidang *hardware* sedemikian besar. Oleh karena itu dapat dipastikan bahwa Microsoft *SQL Server 2008 R2* membawa beberapa terobosan dalam bidang pengolahan data dan penyimpanan data.

Microsoft merilis *SQL Server 2008* dalam beberapa versi yang disesuaikan dengan segment-segment pasar yang dituju. Versi-versi tersebut adalah sebagai berikut. Menurut cara pemrosesan data pada prosesor maka Microsoft mengelompokkan produk ini berdasarkan 2 jenis yaitu :

1. Versi 32-bit (x86), yang biasanya digunakan untuk komputer dengan *single prosesor* (Pentium 4) atau lebih tepatnya prosesor 32 bit dan sistem operasi Windows XP.
2. Versi 64-bit (x64), yang biasanya digunakan untuk komputer dengan lebih dari satu prosesor (Misalnya Core 2 Duo) dan system operasi 64 bit seperti Windows XP 64, Vista, dan Windows 7. Sedangkan secara keseluruhan terdapat versi versi seperti berikut ini:
3. Versi Compact, ini adalah versi “Tipis” dari semua versi yang ada. Versi ini seperti versi desktop pada SQL Server 2000. Versi ini juga digunakan pada *handheld device* seperti Pocket PC, PDA, *Smart Phone*, Tablet PC.
4. Versi Express, ini adalah versi “Ringan” dari semua versi yang ada (tetapi versi ini berbeda dengan versi compact) dan paling cocok untuk latihan para pengembang aplikasi.



Gambar II.5. Tampilan SQL Server
(Sumber : Wahana Komputer : 2008 : 41)

II.9. Normalisasi (*Normalization*)

Normalisasi merupakan suatu proses yang memudahkan desain struktur *table* secara benar sehingga *query* yang tak dapat ditanyakan tidak muncul. Disamping itu, normalisasi cenderung meminimumkan duplikasi data di dalam suatu basis data. Ini memiliki keunggulan dalam mengurangi ruang simpan yang dibutuhkan maupun mempercepat *query*. (Sri Andriani : 2015).

Normalisasi sendiri merupakan cara pendekatan lain dalam membangun desain logik basis data relasional yang tidak secara langsung berkaitan dengan model data, tetapi dengan menerapkan sejumlah aturan dan kriteria standar untuk menghasilkan struktur tabel yang normal. Namun demikian, dalam pelaksanaannya desain logik basis data relasional yang didasari baik oleh prinsip normalisasi maupun yang didasari oleh transformasi secara hati-hati dari model E-R ke bentuk fisik akan menghasilkan hasil yang mirip. (Fathansyah : 2015:41)



II.10. *Unified Modelling Language (UML)*


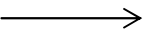
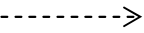
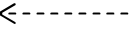
Menurut Windu Gata, Grace (2013:4), *Unified Modeling Language (UML)* adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

a. *Usecase Diagram*

Usecase diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Usecase* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *usecase* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Tabel II.1. Simbol *Use Case*

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu



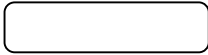
	dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

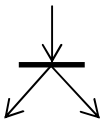
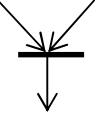
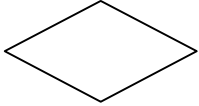
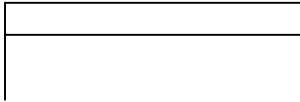
(Sumber : Ade Hendini: 2016)

b. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity Diagram* yaitu:

Tabel II.2. Simbol *Activity Diagram*

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.

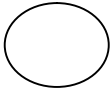
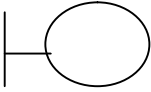
	<p><i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.</p>
	<p><i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.</p>
	<p><i>Decision Points</i>, menggambarkan pilihan untuk pengambilan keputusan, <i>true</i>, <i>false</i>.</p>
	<p><i>Swimlane</i>, pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.</p>

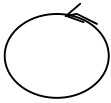

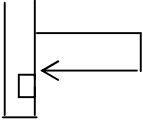


(Sumber : Ade Hendini: 2016)

c. Diagram Urutan (*Sequence Diagram*)

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu:

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<p><i>Entity Class</i>, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.</p>
	<p><i>Boundary Class</i>, berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.</p>

	<p><i>Control class</i>, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.</p>
	<p><i>Message</i>, simbol mengirim pesan antar <i>class</i>.</p>
	<p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>
	<p><i>Activation</i>, <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

(Sumber : Ade Hendini: 2016)

d. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class Diagram* secara khas meliputi : Kelas (*Class*), Relasi *Assosiations*, *Generalitation* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang

disebut dengan *Multiplicity* atau *Cardinality*. (Jurnal Khatulistiwa Informatika, Vol. IV, No. 2 Desember 2016)

Tabel II.4. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Ade Hendini: 2016)