

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu (Indra, dkk : 2013 : 47).

Ada beberapa definisi sistem antara lain adalah sebagai berikut :

1. Menurut Winda dan Umi, (2016 : 21) : Sistem adalah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan.
2. Menurut Eka, (2014 : 28) : Sistem adalah Prosedur logis dan rasional untuk merancang suatu rangkaian komponen yang berhubungan satu dengan yang lainnya dengan maksud untuk berfungsi sebagai suatu kesatuan dalam usaha mencapai suatu tujuan telah ditentukan.

II.1.2. Klasifikasi Sistem

Menurut Indra, dkk (2013 : 48), sistem dapat diklasifikasikan dari beberapa sudut pandang diantaranya sebagai berikut :

1. Sistem Klasifikasi abstrak dan fisik, dimana sistem fisik jelas tampak secara fisik misalnya sistem komputer, alat elektronik atau sistem komunikasi. Sedangkan sistem abstrak adalah sistem yang merupakan ide-ide, teori yang tidak terlihat secara fisik seperti sistem teknologi dimana, sistem logika hubungan manusia dan Tuhan.

2. Sistem alamiah dan sistem buatan. Sistem alami adalah sistem yang tercipta secara alami oleh alam dan sistem buatan sistem yang sengaja dibuat oleh manusia.
3. Sistem tertentu dan tak tentu. Sistem tentu adalah sistem yang sifatnya dapat diprediksikan, interaksi antara elemen dalam sistem tersebut dapat dijabarkan dengan pasti, hingga *output* dan *input* dari sistem ini dapat diketahui dan diramalkan. Contoh dari sistem ini misalnya Sistem komputer. Sedangkan sistem tak tentu adalah sistem ditentukan atau acak yang memiliki unsur probabilitas.
4. Sistem tertutup dan terbuka. Dimana sistem tertutup adalah sistem yang sifatnya bekerja secara otomatis dan lingkungan luar tidak terlalu mempengaruhi sistem. Sedangkan sistem terbuka adalah sistem yang berhubungan langsung dengan lingkungan luar sistem, *input*, *output* dan proses jalannya sistem tergantung pada lingkungannya. Sistem informasi adalah, suatu sistem dalam organisasi untuk memenuhi kebutuhan akan pengolahan data tertentu, dalam mendukung kegiatan operasi yang memiliki sifat managerial dan kegiatan strategis dan penyediaan *output* berupa laporan bagi pihak-pihak tertentu dari luar maupun dalam organisasi.

II.2. Sistem Pakar

II.2.1. Sistem Pakar(*expert system*)

Sistem pakar (*expert system*) adalah sistem yang berusaha mengadopsi pengetahuan manusia kekomputer agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para pakar. Tujuan mengembangkan sistem pakar sebenarnya bukan untuk menggantikan peran manusia, tetapi untuk mengalihkan pengetahuan manusia kedalam bentuk sistem, sehingga dapat digunakan oleh orang banyak dan tidak terbatas oleh waktu (Yasidah, dkk, 2013 : 33).

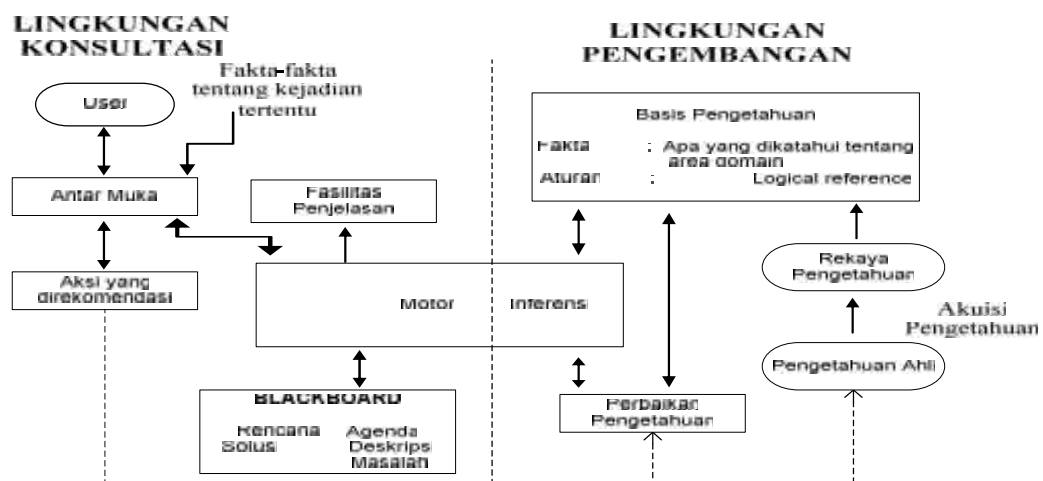
Ada beberapa definisi lain mengenai Sistem Pakar antara lain:

1. Menurut Yasidah Nur istiqomahdan Abdul Fadlil (2013 : 34) : Untuk membangun sistem yang difungsikan untuk menirukan seorang pakar manusia harus bisa melakukan hal-hal yang dapatdikerjakan oleh para pakar.
2. Menurut MuhdIhsan, dkk (2017 : 129) :Sistem pakar dibuat hanya pada *domain* pengetahuan tertentu untuk suatu kepakaran tertentu yang mendekati kemampuan manusia di salah satu bidang saja.
3. Menurut Joko Minardi (2016 : 84) : Sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan sistem pakar digunakan untuk memasukkan pengetahuan pakar ke dalam lingkungan sistem pakar, sedangkan lingkungan konsultasi digunakan oleh pengguna yang bukan pakar untuk memperoleh pengetahuan pakar.

II.2.2. Struktur Sistem Pakar

Ada dua bagian dari sistem pakar, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*).

Lingkungan pengembangan digunakan oleh pembuat sistem pakar untuk membangun komponen-komponennya dan memperkenalkan pengetahuan ke dalam *knowledge base* (basis pengetahuan). Lingkungan konsultasi sehingga pengguna mendapatkan pengetahuan dan nasihat dari sistem pakar layaknya berkonsultasi dengan seorang pakar. Adapun struktur pada sistem pakar dapat dilihat pada gambar II.1. sebagai berikut :



Gambar II.1. Struktur Sistem Pakar

Sumber : (Mhd Ihsan, dkk : 2017 : 129)

Keterangan :

1. Akuisisi pengetahuan

Subsistem ini digunakan untuk memasukkan pengetahuan dari seorang pakar dengan cara merekayasa pengetahuan agar bisa diproses oleh komputer dan menaruhnya kedalam basis pengetahuan dengan format tertentu (dalam bentuk representasi pengetahuan). Sumber-sumber pengetahuan bisa diperbolehkan dari pakar, buku, dokumen, multimedia, basisdata. Laporan riset khusus, dan informasi yang terdapat di *web*.

2. Basis pengetahuan (*Knowledge Base*)

Basis pengetahuan mengandung pengetahuan yang diperlukan untuk memahami, memformulasikan, dan menyelesaikan masalah. Basis pengetahuan terdiri dari dua elemen dasar, yaitu :

- a. Fakta, misalnya situasi, kondisi, untuk permasalahan yang ada.
- b. Aturan (*rule*), untuk mengarahkan penggunaan pengetahuan dalam memecahkan masalah.

3. Mesin Inferensi (*Inference Engine*)

Mesin inferensi adalah sebuah program yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada, memanipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan dalam basis pengetahuan untuk mencapai solusi atau kesimpulan. Dalam prosesnya, mesin inferensi menggunakan strategi pengendalian, yaitu strategi yang berfungsi sebagai panduan arahan dalam melakukan proses penalaran. Ada tiga teknik pengendalian yang

digunakan, yaitu *forward chaining*, *backward chaining*, dan gabungan dari kedua teknik tersebut.

4. Daerah Kerja (*Blackboard*)

Untuk merekam hasil sementara yang akan dijadikan sebagai keputusan dan untuk menjelaskan sebuah masalah yang sedang terjadi, sistem pakar membutuhkan *blackboard* yaitu area pada memori yang berfungsi sebagai basis data. Tiga tipe keputusan yang dapat direkam pada *blackboard*, yaitu:

- a. Rencana : bagaimana menghadapi masalah.
- b. Agenda : aksi-aksi potensial yang sedang menunggu untuk dieksekusi.
- c. Solusi: calon aksi yang akan dibangkitkan.

5. Antarmuka Pengguna (*User Interface*)

Digunakan sebagai media komunikasi antara pengguna dan sistem pakar, komunikasi ini paling bagus bila disajikan dalam bahasa alami (*natural language*) dan dilengkapi dengan grafik, menu, dan formulir elektronik. Pada bagian ini akan terjadi dialog antara sistem pakar dan pengguna.

6. Subsistem Penjelasan (*Explanation Subsystem/Justifire*)

Berfungsi memberi penjelasan kepada pengguna, bagaimana suatu kesimpulan dapat diambil. Kemampuan seperti ini sangat penting bagi pengguna untuk mengetahui proses pemindahan keahlian pakar maupun dalam pemecahan masalah.

7. Sistem Perbaiki Pengetahuan (*Knowledge Refining System*)

Kemampuan memperbaiki pengetahuan (*knowledge refining system*) dari seorang pakar diperlukan untuk menganalisis pengetahuan, belajar dari kesalahan masa lalu. Kemudian memperbaiki pengetahuannya sehingga dapat dipakai pada masa yang akan datang. Kemampuan evaluasi diri seperti itu diperlukan oleh program agar menganalisa alasan-alasan kesuksesan dan kegagalannya dalam mengambil kesimpulan. Dengan cara ini basis pengetahuan yang lebih baik dan penalaran yang lebih efektif akan dihasilkan.

8. Pengguna (*User*)

Pada umumnya pengguna sistem pakar bukanlah seorang pakar (*non-expert*) yang membutuhkan solusi, saran, atau pelatihan (*training*) dan berbagai permasalahan yang ada.

II.3. Penyakit Dismenore Primer

Dismenore primer yaitu dismenore sejak *menarche*, dengan gejala haid nyeri dan tidak ada kelainan dari alat kandungan. Penyebab kejadian dismenore primer yaitu adanya jumlah prostaglandin F2a yang berlebihan pada darah menstruasi yang pada akhirnya merangsang hiperaktifitas uterus, F2a merupakan stimulant kuat terhadap uterus dan vasokonstriktor pada endometrium fase sekresi, banyak teori telah dikemukakan untuk menerangkan penyebab dismenorea primer, tetapi patofisiologinya belum jelas dimengerti. Rupanya beberapa faktor memegang peranan sebagai penyebab dismenore primer.

Penyebab utama dismenore primer adalah adanya prostaglandin F2a (PGF2a) yang dihasilkan di endometrium. PGF2a merupakan hormon yang diperlukan untuk menstimulasi kontraksi uterus selama menstruasi (Dita Trimayasari & Kadar Kuswandi, 2014 :195).

Beberapa hal yang menyebabkan nyeri dapat berkurang dan mereda, yaitu gerakan tertentu, istirahat yang cukup, mengambil nafas dalam penggunaan obat. Selain hal tersebut yang paling berpengaruh dalam meredakan nyeri adalah hal-hal yang dipercaya yang sifatnya psikologis pada penderita dapat membantu mengatasi rasa nyeri (Noor Azizah, dkk, 2015 : 81).

II.4. Dempster Shafer

Dempster Shafer adalah suatu teori matematika untuk pembuktian berdasarkan *belief functions and plausible reasoning* (fungsi kepercayaan dan pemikiran yang masuk akal), yang digunakan untuk mengkombinasikan potongan informasi yang terpisah (bukti) untuk mengkalkulasi kemungkinan dari suatu peristiwa. Teori ini dikembangkan oleh Arthur P. Dempster dan Glenn Shafer. Secara umum teori *Dempster Shafer* ditulis dalam suatu *interval [Belief, Plausibility]*. *Belief (Bel)* adalah ukuran kekuatan *evidence* dalam mendukung suatu himpunan proposisi. Jika bernilai 0 maka mengindikasikan bahwa tidak ada *evidence*, dan jika bernilai 1 menunjukkan adanya kepastian. *Plausibility (Pl)* dinotasikan sebagai : $Pl(s) = 1 - Bel(-s)$ *Plausibility* juga bernilai 0 sampai 1. Jika yakin akan -s, maka dapat dikatakan bahwa $Bel(-s)=1$, dan $Pl(-s)=0$. *Plausibility* akan mengurangi tingkat kepercayaan dari *evidence*. Pada teori *Dempster Shafer* kita mengenal adanya *frame of discernment* yang dinotasikan dengan θ dan *mass*

function yang dinotasikan dengan m . *Frame* ini merupakan semesta pembicaraan dari sekumpulan hipotesis. Tujuannya adalah mengkaitkan ukuran kepercayaan dengan elemen-elemen θ . Tidak semua *evidence* secara langsung mendukung tiap-tiap elemen. Untuk itu perlu adanya fungsi densitas (m). Nilai m tidak hanya mendefinisikan elemen-elemen θ saja, namun juga semua subset-nya. Dan harus menunjukkan bahwa jumlah semua m dalam subset θ sama dengan 1. Andaikan diketahui X adalah subset dari θ , dengan m_1 sebagai fungsi densitasnya, dan Y juga merupakan subset dari θ dengan m_2 sebagai fungsi densitasnya, maka kita dapat membentuk fungsi kombinasi m_1 dan m_2 sebagai m_3 , menggunakan aturan yang lebih dikenal dengan *Dempster's Rule of Combination* (Triara Puspitasari, dkk, 2016 : 3)

$$m_3(Z) = \frac{\sum_{X \cap Y = Z} m_1(X) \cdot m_2(Y)}{1 - \sum_{X \cap Y = \emptyset} m_1(X) \cdot m_2(Y)}$$

Keterangan :

$m_1(X)$ adalah *mass function* dari *evidence* X

$m_2(Y)$ adalah *mass function* dari *evidence* Y

$m_3(Z)$ adalah *mass function* dari *evidence* Z

Contoh kasus :

Pada contoh dibawah ini, akan di cari persentase kemungkinan dari gangguan-gangguan Autisme dengan menggunakan perhitungan pada *table* dibawah ini :

Tabel.II.1. ContohGejalaPilihanPasien

No	Kode	Gejala	Bobot
1	GP16	Ekspresimukakuranghidup	0.7
2	GP18	Menolakuntukdipeluk	0.8

Maka untuk menghitung nilai *Dempster Shafer* (DS) gangguan **Autisme** yang dipilih dengan menggunakan nilai *believe* yang telah ditentukan pada setiap gejala.

$$Pl(\square) = 1 - Bel$$

Dimana nilai bel (*believe*) merupakan nilai bobot yang di *input* oleh pakar, maka untuk mencari nilai dari kedua gejala diatas, terlebih dahulu dicari nilai $\square$, seperti yang dibawah ini.

Gejala1 :Eksperesi muka kurang hidup

(GP16)

Maka : GP16(bel) = 0.7

$$\begin{aligned} GP16(\square) &= 1-0,7 \\ &= 0.3 \end{aligned}$$

Gejala2 :Menolak untuk dipeluk (GP18)

Maka : GP18 (bel) = 0.8

$$\begin{aligned} GP18(\square) &= 1-0,8 \\ &= 0,2 \end{aligned}$$

Maka untuk mencari nilai dari GPn, digunakan rumus :

$$m_3(Z) = \frac{\sum_{x \cap y = Z} m_1(X).m_2(Y)}{1 - \sum_{x \cap y = \emptyset} m_1(X).m_2(Y)}$$

Makan ilai GPn dari gejala diatas adalah :

$$\begin{aligned} GPn &= \frac{0,7*0.8}{1-(0.3*0.2)} \\ &= 0.56 / 1-0.06 \\ &= 0.56 / 0.94 \\ &= 0.59 \end{aligned}$$

II.5. UML

UML adalah bahasa pemodelan untuk *sistem* atau perangkat lunak yang berparadigma berorientasi objek. Pemodelan sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. Salah satu jenis diagram dalam UML adalah *Use Case Diagram* (Lukman arif sanjani, dkk, 2014 : 88).

II.5.1 Jenis-Jenis Diagram UML

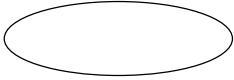
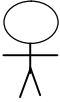
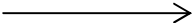
Berikut ini adalah mengenai berbagai diagram UML serta tujuan dan simbol-simbolnya, yaitu :

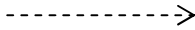
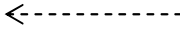
1. Diagram *Use Case* (*Use Case Diagram*)

Use Case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara actor dengan sistem seorang/sebuah actor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu (Andy : 2013 : 19).

Berikut ini adalah simbol-simbol yang ada pada diagram *use case* dapat dilihat pada Tabel II.1:

Tabel II.2. Simbol-simbol *Use Case Diagram*

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 mengidentifikasi siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidentifikasi aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidentifikasi 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 : Gata, 2013 : 4)

2. Diagram Kelas (*Class Diagram*)

Class Diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah obyek dan merupakan inti dari pengembangan dan desain berorientasi obyek. Sebuah *class diagram* digunakan untuk menunjukkan keberadaan dari kelas dan hubungannya di dalam pandangan *logic* dari sebuah sistem. Sebuah kelas tunggal merepresentasikan sebuah sudut pandang dari struktur kelas dari sebuah sistem. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi) (Andy : 2013 : 19).

Berikut ini adalah simbol-simbol yang ada pada diagram *class diagram* dapat dilihat pada Tabel II.2. :

Tabel II.3. 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 : Gata, 2013 : 9)

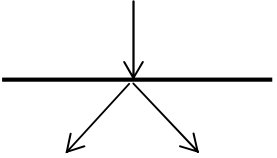
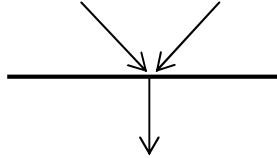
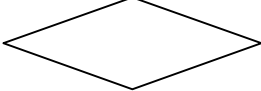
3. Diagram Aktivitas (*Activity Diagram*)

Activity diagram menyediakan gambaran visual dari aliran aktifitas, baik dalam sistem, bisnis, alur kerja, atau proses lainnya. Diagram ini berfokus pada kegiatan yang dilakukan dan siapa (atau apa) yang bertanggung jawab atas kinerja dari kegiatan tersebut. Dalam banyak hal, diagram ini memainkan peran yang mirip dengan diagram alur, Tetapi perbedaan utama antara mereka dan notasi *flowchar* tadalah bahwa mereka mendukung perilaku secara parallel (Andy : 2013 : 19).

Berikut ini adalah simbol-simbol yang ada pada *activity diagram* dapat dilihat pada Tabel II.3. :

Tabel II.4. Simbol-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.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>True</i> , <i>False</i> .
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

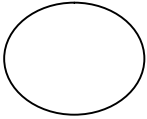
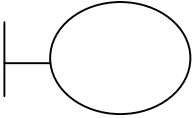
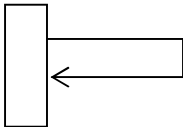
(Sumber : Gata, 2013 : 6)

4. Diagram Rangkaian (*Sequence Diagram*)

Sequence Diagram digunakan untuk melacak eksekusi dari sebuah skenario *use case*. *Sequence diagram* menggambarkan interaksi dengan menandai setiap partisipan dengan garis hidup yang berjalan secara vertical kebawah halaman dan urutan pesan dengan membaca kearah bawah halaman (Andy : 2013 : 19).

Berikut ini adalah simbol-simbol yang ada pada diagram *Sequence Diagram* dapat dilihat pada Tabel II.4.:

Tabel II.5. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>EntityClass</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<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.
	<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.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.

	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Gata, 2013 : 7)

II.6. Microsoft Visual Basic 2010

Microsoft *Visual Basic* merupakan sebuah Bahasa pemrograman yang menawarkan IDE (*Integrated Development Environment*) *visual* untuk membuat program perangkat lunak berbasis sistem operasi Microsoft *Windows* dengan menggunakan model pemrograman. *Visual Basic* merupakan turunan Bahasa pemrograman *BASIC* yang menawarkan pengembangan perangkat lunak berbasis grafik dengan cepat, beberapa bahasa *script* seperti VBA (*Visual Basic for Applications*) dan VBScript (*Visual Basic Scripting Edition*), mirip seperti halnya *Visual Basic*, tetapi cara kerjanya yang berbeda. Para programmer dapat membangun aplikasi dengan menggunakan komponen yang disediakan oleh microsoft *visual basic* (Anggoro, dkk, 2013 : 39).

II.7. SQL Server 2008

SQL (*Structured Query Language*) adalah bahasa *non procedural* untuk mengakses data pada *database* relasional. SQL adalah bahasa *database* yang dipergunakan dalam menyelesaikan permasalahan dalam *database* serta mempunyai kelebihan dalam mengolah data. Standar SQL mula-mula didefinisikan oleh ISO (*International Standards Organization*) dan ANSI (*The American National Standards Institute*) yang dikenal dengan sebutan SQL86 (Eka Iswandy, 2014 : 30).