

BAB II

LANDASAN TEORI

2.1 Defenisi Sistem Pakar

Sistem pakar adalah suatu program komputer cerdas yang menggunakan pengetahuan dan prosedur inferensi untuk menyelesaikan masalah yang cukup sulit sehingga membutuhkan seorang ahli untuk menyelesaikannya (Widiastuti wenny, 2012). Sistem informasi ini juga mengadopsi pengetahuan dari manusia ke komputer, agar komputer dapat menyelesaikan masalah layaknya seorang pakar (kusumadewi, 2003).

Suatu sistem pakar merupakan paket perangkat lunak atau pakar program komputer yang ditujukan sebagai penyedia nasihat dan sarana bantu dalam memecahkan masalah di bidang-bidang spesialisasi tertentu seperti sains, rekayasa, matematika, kedokteran, pendidikan, dan sebagainya (Arhami, 2005).

Pada dasarnya sistem pakar diterapkan untuk mendukung aktivitas pemecahan masalah. Beberapa aktivitas pemecahan masalah yang dimaksud seperti (Lestari, 2012).

a) Interpretasi

Membuat kesimpulan atau deskripsi dari sekumpulan data mentah. Pengambilan keputusan dari hasil observasi, termasuk pengenalan ucapan, analisis citra, interpretasi sinyal, dll.

b) Prediksi

Memproyeksikan akibat-akibat yang dimungkinkan dari situasi-situasi tertentu. Contoh: prediksi demografi, prediksi ekonomi, dll.

c) Diagnosis

Menentukan sebab malfungsi dalam situasi kompleks yang didasarkan pada gejala-gejala yang teramati diagnosis medis, elektronis, mekanis, dll.

d) Perancangan (desain)

Menentukan konfigurasi komponen-komponen sistem yang cocok dengan tujuan-tujuan kinerja tertentu yang memenuhi kendala-kendala tertentu.

Contoh: perancangan layout sirkuit, bangunan.

e) Perencanaan

Merencanakan serangkaian tindakan yang akan dapat mencapai sejumlah tujuan dengan kondisi awal tertentu. Contoh: perencanaan keuangan, militer, dll.

f) Monitoring

Membandingkan hasil pengamatan dengan kondisi yang diharapkan.

Contoh: *computer aided monitoring system*.

g) *Debugging*

Menentukan dan menginterpretasikan cara-cara untuk mengatasi malfungsi.

h) Instruksi

Mendeteksi dan mengoreksi defisiensi dalam pemahaman domain subjek.

Contoh: melakukan instruksi untuk diagnosis dan *debugging*.

i) Kontrol

Mengatur tingkah laku suatu environment yang kompleks. Contoh:

melakukan kontrol terhadap interpretasi, prediksi, perbaikan dan monitoring kelakuan sistem.

2.1.1 Manfaat Sistem Pakar

Manfaat sistem pakar yaitu :

1. Memungkinkan orang awam bisa mengerjakan pekerjaan para ahli.
2. Bisa melakukan proses secara berulang secara otomatis.
3. Menyimpan pengetahuan dan keahlian para pakar.
4. Mampu dan melestarikan keahlian para pakar (terutama yang termasuk keahlian langka).
5. Mampu beroperasi dalam lingkungan yang berbahaya.
6. Memiliki kemampuan untuk bekerja dengan informasi yang tidak lengkap dan mengandung ketidakpastian.
7. Tidak memerlukan biaya saat tidak digunakan, sedangkan pada pakar manusia memerlukan biaya sehari-hari.
8. Dapat digandakan (diperbanyak) sesuai kebutuhan dengan waktu yang minimal dan sedikit biaya.
9. Dapat memecahkan masalah lebih cepat daripada kemampuan manusia dengan catatan menggunakan data yang sama.
10. Menghemat waktu dalam pengambilan keputusan
11. Meningkatkan kualitas dan produktivitas
12. Memiliki reabilitas

13. Meningkatkan kapabilitas sistem komputer

14. Sebagai media pelengkap dalam pelatihan (Kholida halum subiyanto, 2013)

2.1.2 Kelemahan Sistem Pakar

Kekurangan sistem pakar sebagai berikut :

1. Pengembangan sistem pakar sangat sulit, lebih sulit daripada membuat software konvensional.
2. Sistem pakar sangat mahal.
3. Pada awal perkembangannya, hampir semua sistem pakar masih harus diimplementasikan dalam komputer besar atau komputer mini.
4. Sistem pakar tidak 100% handal.
5. Boleh jadi sistem tak dapat membuat keputusan (Arhami, 2005)

2.1.3 Ciri-Ciri Sistem Pakar

- a) Terbatas pada domain keahlian tertentu.
- b) Dapat memberikan penalaran untuk data yang tidak pasti.
- c) Dapat mengemukakan rangkaian alasan yang diberikannya dengan cara yang dapat dipahami.
- d) Berdasarkan pada kaidah atau *rule* tertentu.
- e) Dirancang untuk dapat dikembangkan secara bertahap.
- f) Pengetahuan dan mekanisme inferensi jelas terpisah.
- g) Keluarannya bersifat anjuran.
- h) Sistem dapat mengaktifkan kaidah secara searah yang sesuai yang dituntun oleh dialog dengan pemakai. (Widiastuti Weny, 2012)

2.1.4 Tujuan Sistem Pakar

Tujuan dari sebuah sistem pakar adalah untuk mentransfer kepakaran yang dimiliki seorang pakar ke dalam komputer, dan kemudian kepada orang lain (*nonexpert*) (Arhami,2005). Aktivitas yang dilakukan untuk memindahkan kepakaran adalah :

1. *Knowledge acquisition* (dari pakar atau sumber lainnya)

Akuisi pengetahuan (*knowledge acquisition*) yaitu kegiatan mencari dan mengumpulkan pengetahuan dari para ahli atau sumber keahlian yang lain.

2. *Knowledge representation* (ke dalam komputer)

Representasi pengetahuan adalah kegiatan menyimpan dan mengatur penyimpanan pengetahuan yang diperoleh dari komputer. Pengetahuan berupa fakta dan aturan disimpan dalam komputer sebagai sebuah komponen yang disebut basis pengetahuan.

3. *Knowledge Inferencing* (Inferensi Pengetahuan)

pengetahuan adalah kegiatan melakukan inferensi berdasarkan pengetahuan yang telah disimpan didalam komputer.

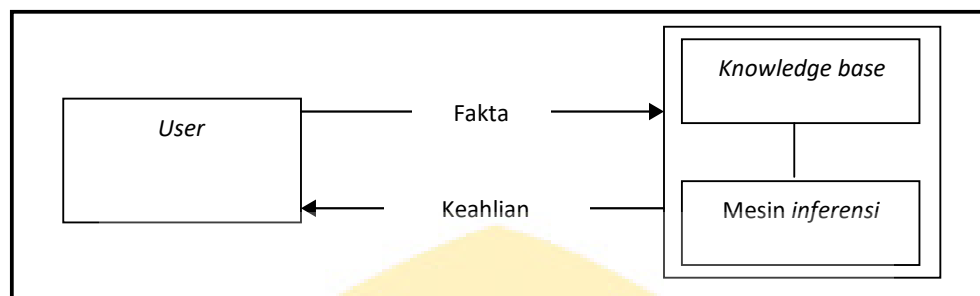
4. *Knowledge Transferring* (Pemindahan Pengetahuan)

Pemindahan pengetahuan adalah kegiatan pemindahan pengetahuan dari komputer ke pemakai yang tidak ahli.

2.1.5 Konsep Dasar Sistem Pakar

Tiap-tiap orang mempunyai keahlian masing-masing yang mungkin satu orang dengan yang lainnya mempunyai keahlian berbeda, tergantung pengetahuannya masing-masing. Komputer dapat diprogram untuk berbuat seperti ahli dalam bidang

tertentu. Komputer yang demikian dapat dijadikan konsultan atau tenaga ahli di bidang tertentu yang dapat menjawab pertanyaan dan nasihat-nasihat yang dibutuhkan. Sistem yang demikian disebut dengan sistem pakar.



(Sumber: Muhammad Arhami 2005, "Konsep Dasar Sistem Pakar")

Gambar 2.1 Konsep dasar fungsi sistem pakar

Pengguna (*user*) menyampaikan fakta atau informasi untuk sistem pakar dan kemudian menerima saran dari pakar atau jawaban ahlinya. Bagian dalam sistem pakar terdiri dari dua komponen utama, yaitu *knowledge base* yang berisi pengetahuan dasar dari pakar atau ahlinya dan mesin *inferensi* yang menggambarkan kesimpulan. Kesimpulan tersebut merupakan respon dari sistem pakar atas permintaan pengguna.

Suatu pengetahuan (*knowledge*) bersifat khusus untuk Sistem pakar menyerupai kepakaran manusia yang secara umum dirancang untuk menjadi pakar (Arhami 2005). Sebagai contoh, sistem pakar kedokteran yang dirancang untuk mendiagnosis infeksi penyakit akan mempunyai suatu uraian pengetahuan (*knowledge*) tentang gejala-gejala penyakit yang disebabkan oleh infeksi penyakit (Arhami 2005). Dalam hasil penelitian ini domain pengetahuan (*knowledge*-nya) adalah bidang kedokteran yang terdiri dari *knowledge* tentang penyakit, gejala, dan cara pengobatan.

Menurut Turban [Arhami] terdapat tiga orang yang terlibat dalam lingkungan sistem pakar, yaitu:

1. Pakar

Orang yang memiliki pengetahuan khusus, pendapat, pengalaman, dan metode, serta kemampuan untuk mengaplikasikan keahliannya tersebut untuk menyelesaikan masalah.

2. Perekeyasa sistem

Orang yang membantu pakar dalam menyusun area permasalahan dengan menginterpretasikan dan mengintegrasikan jawaban pakar atas pertanyaan yang diajukan, menggambarkan analogi, dan menerangkan kesulitan-kesulitan konseptual.

3. Pengguna

Sistem pakar memiliki beberapa pengguna yaitu pengguna bukan pakar, pelajar, pembangun sistem pakar yang ingin meningkatkan dan menambah basis pengetahuan, dan pakar.

2.1.6 Karakteristik Sistem Pakar

Karakteristik atau ciri-ciri dalam sistem pakar adalah sebagai berikut :

1. Basis pengetahuan mudah diperbaharui.
2. Kemampuan mempelajari fakta atau kejadian baru dari pengalamannya sendiri.
3. Dapat digunakan dalam berbagai jenis komputer.
4. Mudah dimodifikasi, yaitu dengan menambah atau menghapus suatu kemampuan dari basis pengetahuannya.

5. Heuristik dalam menggunakan pengetahuan (yang sering kali tidak sempurna) untuk mendapatkan penyelesaiannya. (Arhami, 2005).

2.1.7 Struktur Sistem Pakar

Menurut Nita Merlina dan Rahmat Hidayat (2012) bahwa sistem pakar terdiri atas dua bagian pokok, yaitu lingkungan pengembangan (development environment) dan lingkungan konsultasi (consultation environment).

- a. Lingkungan pengembangan digunakan sebagai pembangunan sistem pakar, baik dari segi pembangunan komponen maupun basis pengetahuan.
- b. Lingkungan konsultasi digunakan oleh seorang yang bukan ahli untuk berkonsultasi.

2.2 Konsep Pemodelan Sistem

Pemodelan adalah suatu representasi atau formalisasi dalam bahasa tertentu yang disepakati dari suatu sistem nyata. Sistem adalah sistem yang sedang berlangsung dalam kehidupan, sistem yang dijadikan titik perhatian dan dipermasalahkan. Pemodelan sistem adalah proses membangun atau membentuk sebuah model dari suatu sistem nyata dalam bahasa formal tertentu.

2.2.1 Unsur Dari Sistem

Unsur dari sistem adalah sebagai berikut :

- a. Adanya elemen-elemen
- b. Adanya interaksi atau hubungan antar elemen
- c. Adanya suatu yang mengikat elemen-elemen tersebut menjadi satu kesatuan.

- d. Terdapat tujuan bersama sebagai hasil akhir

2.2.2 Jenis Pemodelan Sistem

Jenis pemodelan sistem adalah sebagai berikut :

- a. Model matematis : mewakili sebuah sistem secara simbol matematis (berbentuk rumus dan nilai (besaran), atribut sebagai variabel, aktivitas sebagai fungsi matematik.
- b. Model informasi : mewakili sebuah sistem dalam bentuk grafik atau tabel (multi mensional).

2.3 Metode Case Based Reasoning

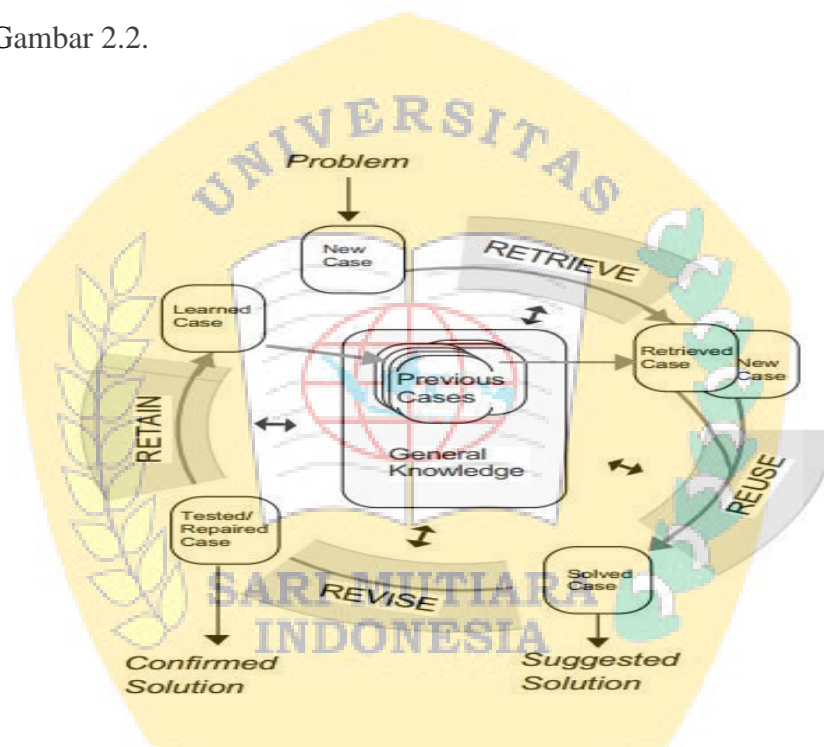
Secara umum, *Case-Based Reasoning* (CBR) merupakan suatu konsep penalaran dalam pemecahan masalah melalui catatan penanganan kasus yang pernah dilakukan oleh seorang ahli. CBR merupakan sebuah cara untuk menyelesaikan suatu kasus baru dengan cara mengadaptasi atau mengingat solusi yang terdapat pada kasus sebelumnya (kasus lama) yang mirip dengan kasus baru tersebut. Kasus lama tersebut disimpan dalam tempat yang disebut dengan basis kasus.

Secara singkat, tahap-tahap penyelesaian masalah berbasis CBR adalah sebagai berikut:

- a) Pengambilan kembali kasus-kasus yang sesuai dari memori (hal ini membutuhkan pemberian indeks terhadap kasus-kasus dengan menyesuaikan fitur-fiturnya).
- b) Pemilihan sekelompok kasus-kasus yang terbaik.

- c) Pemilihan atau menentukan penyelesaian.
- d) Evaluasi terhadap penyelesaian (hal ini dimaksudkan untuk meyakinkan agar tidak mengulang penyelesaian yang salah)
- e) Penyimpanan penyelesaian kasus terbaru dalam penyimpan kasus/memori.

Berdasarkan tahap-tahap tersebut, timbullah siklus yang terjadi dalam CBR. Aamodt dan Plaza (1994) menggambarkan siklus tersebut seperti pada Gambar 2.2.



Gambar 2.2 Siklus CBR (Aamodt dan Plaza, 1994)

2.3.1 Algoritma *k-Nearest Neighbor* (*k*-NN)

Algoritma *nearest neighbor* melibatkan teknik *similarity* (kemiripan) (Aamodt dan Plaza, 1994). Teknik *similarity* akan menghitung nilai atau tingkat kemiripan antara kasus baru yang dimasukkan dengan kasus lama yang terdapat dalam basis kasus. Setelah didapatkan nilai kemiripan pada setiap kasus lama,

kemudian dicari nilai terdekat dengan kasus baru yang dimasukkan (nilai terdekat $k = 1$). Nilai terdekat tersebut yang digunakan sebagai identitas tujuan.

Teknik *similarity* yang digunakan untuk menghitung tingkat kemiripan antara kasus baru yang dimasukkan dengan kasus lama yang terdapat dalam basis kasus seperti pada persamaan.

$$\text{Similarityn (T,S)} = \frac{\sum_{i=1}^n (T_i.S_i) \times W_i}{W_i}$$

Dimana :

T : kasus baru, S : kasus yang ada dalam penyimpanan, n : jumlah atribut dalam setiap kasus,

i : atribut individu antara I s.d.n,f merupakan fungsi similarity untuk fitur I dalam kasus T dan kasus T dan kasus S, dan W menunjukkan bobot yang diberikan pada atribut ke-i.

2.4 Pengertian Sepeda Motor

Sepeda motor adalah kendaraan beroda dua yang digerakkan oleh sebuah mesin. Letak kedua roda sebaris lurus dan pada kecepatan tinggi sepeda motor tetap stabil disebabkan oleh gaya giroskopik (Michaux, 1868). Sedangkan pada kecepatan rendah, kestabilan atau keseimbangan sepeda motor bergantung kepada pengaturan setang oleh pengendara. Penggunaan sepeda motor di Indonesia sangat populer karena harganya yang relatif murah, terjangkau untuk sebagian besar kalangan dan penggunaan bahan bakarnya serta biaya operasionalnya cukup hemat.

2.5 Basis Data (*Data base*)

2.5.1 Pengertian Basis Data

Basis data adalah dua atau lebih simpanan data dengan elemen-elemen data penghubung, yang dapat diakses lebih dari satu cara. Basis data dinyatakan dengan tehnik-tehnik formal dan manajemen basis data (Abdillah, 2012). Menurut Connolly dan Begg (2010), basis data adalah sebuah kumpulan data yang secara logis terkait dan dirancang untuk memenuhi suatu kebutuhan informasi dari sebuah organisasi.

Dari penjabaran defenisi basis data diatas maka kesimpulan bahwa basis data adalah kumpulan data yang memiliki elemen- elemen saling terhubung dalam memenuhi kebutuhan organisasi.

2.5.2 Tujuan Basis Data

1. Mengatur data sehingga diperoleh kemudahan, ketepatan dan kecepatan dalam penggunaan kembali data tersebut.
2. Tidak adanya redundansi dan menjaga konsistensi data.
3. Pengaturan dalam pemilahan data sesuai dengan fungsi dan jenisnya.

2.5.3 Manfaat Penggunaan Basis Data

1. Kecepatan dan Kemudahan (*Speed*)

Dengan menggunakan basis data pengambilan informasi dapat dilakukan dengan cepat dan mudah.

2. Kebersamaan Pemakaian (*Sharability*)

Sebuah basis data dapat digunakan oleh banyak user dan banyak aplikasi.

3. Pemusatan Kontrol Data

cukup dengan satu basis data untuk banyak keperluan, pengontrolan terhadap data juga cukup dilakukan di satu tempat saja.

4. Efisiensi Ruang Penyimpanan (*Space*)

Dengan pemakain bersama, kita tidak perlu menyediakan tempat penyimpanan di berbagai tempat, tetapi cukup satu saja sehingga ini akan menghemat ruang penyimpanan data yang dimiliki oleh sebuah organisasi.

5. Keakuratan (*Accuracy*)

Penerapan secara ketat aturan tipe data, domain data, keunikan data, hubungan antara data, dan lain-lain, dapat menekan keakuratan dalam pemasukan/penyimpanan data.

6. Ketersediaan (*Availability*)

Dengan basis data kita dapat mem-backup data, memilah-milah data mana yang masih diperlukan dan data mana yang perlu kita simpan ke tempat lain.

7. Keamanan (*Security*)

Kebanyakan DBMS dilengkapi dengan fasilitas manajemen pengguna diberikan hak akses yang berbeda-beda sesuai dengan pengguna dan posisinya. Basis data bisa diberikan passwordnya untuk membatasi orang yang mengaksesnya.

8. Kemudahan dalam Pembuatan Program Aplikasi Baru

Pengguna basis data merupakan bagian dari perkembangan teknologi. Dengan adanya basis data pembuatan aplikasi bisa memanfaatkan kemampuan dari DBMS, sehingga pembuatan aplikasi tidak perlu mengurus penyimpanan data, tetapi cukup mengatur interface untuk pengguna.

2.5.2 Bagian Bagian Basis Data

Beberapa hal yang termaksud unsur-unsur dari basis data adalah sebagai berikut

1. *Entititas*

Entititas adalah orang, tempat, kejadian atau konsep yang informasinya direkam. Pada bidang kesehatan *Entity* adalah Pasien, Dokter, Kamar

2. *Field*

Setiap *entity* mempunyai atribut atau sebutan untuk mewakili suatu *entity*. Seorang siswa dapat dilihat dari atributnya misalnya, NIM, Nama_siswa, Alamat.

3. *Record*

Record adalah kumpulan isi elemen data (*atribut*) yang saling berhubungan menginformasikan tentang suatu entity secara lengkap. Contoh Kumpulan atribut NIP, Nama, dan alamat berisikan “01001245566”, Sanusi, Jl. Hati suci No 2 Kupang.

4. *Data Value*

Merupakan data aktual atau informasi yang disimpan di tiap data elemen.

Isi atribut disebut nilai data.

5. Kunci Elemen Data (*Key Data Element*)

Tanda pengenal yang secara unik mengidentifikasi entitas dari suatu kumpulan entitas. Contoh Entitas Mahasiswa yang mempunyai atribut-atribut npm, nama, alamat, tanggal lahir menggunakan Kunci Elemen Data npm.

2.5.5 Relasi Tabel Basis Data

Beberapa bentuk relasi tabel pada basis data sebagai berikut :

1. *One to one relationship*

Hubungan antara file pertama dengan file kedua adalah satu berbanding satu. Hubungan tersebut dapat digambarkan dengan tanda kotak untuk menunjukkan table dan relasi antara keduanya diwakilkan dengan tanda panah tunggal.



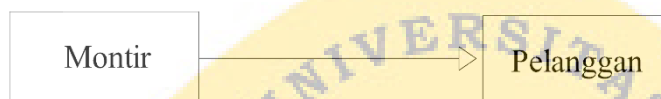
2. *One to many relationship*

Hubungan antara file pertama dengan file kedua adalah satu berbanding banyak atau dapat pula dibalik banyak lawan satu. Hubungan tersebut dapat digambarkan dengan tanda kotak untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah ganda pada salah satu hubungan.



3. *Many to many relationship*

Hubungan antara file pertama dengan file kedua adalah banyak berbanding banyak. Hubungan tersebut dapat digambarkan dengan tanda kotak untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah ganda pada kedua hubungan tersebut.



2.6 Pengertian MySQL

Menurut Nugroho (2013), “MySQL adalah software atau program Database Server”. Sedangkan SQL adalah bahasa pemrogramannya, bahasa permintaan (query) dalam database server termasuk dalam MySQL itu sendiri. SQL juga dipakai dalam software database server lain, seperti SQL Server, Oracle, PostgreSQL dan lainnya.

Menurut Buana (2014), “MySQL Merupakan database server yang paling sering digunakan dalam pemograman PHP. MySQL digunakan untuk menyimpan data dalam database dan memanipulasi data-data yang diperlukan. Manipulasi data tersebut berupa menambah, mengubah, dan menghapus data yang berada dalam database”.

2.6.1 Keunggulan *MySQL*

Berikut ini adalah keunggulan *MySQL* :

1. Bersifat *open source*
2. Menggunakan bahasa *SQL (Structure Query Language)*
3. *Super performance* dan *reliable*
4. Sangat mudah dipelajari.
5. Memiliki dukungan support (group) pengguna *MySQL*.
6. Mampu lintas *platform dan Multiuser*

2.7 *Unified Modeling Language (UML)*

Menurut Widodo (2011), “UML adalah bahasa pemodelan standar yang memiliki sintak dan semantik”. Menurut Henderi (2010), “UML adalah sebuah bahasa pemodelan yang telah menjadi standar dalam industri *software* untuk visualisasi, merancang, dan mendokumentasikan sistem perangkat lunak”.

Menurut Nugroho (2009), “UML (*Unified Modeling Language*) adalah Metodologi kolaborasi antara metoda-metoda Booch, OMT (*Object Modeling Technique*), serta OOSE (*Object Oriented Software Engineering*) dan beberapa metoda lainnya, merupakan metodologi yang paling sering digunakan saat ini untuk analisa dan perancangan sistem dengan penggunaan bahasa “pemrograman berorientasi objek” (OOB)”.





Berdasarkan pendapat yang dikemukakan di atas dapat ditarik kesimpulan bahwa UML adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk divisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari


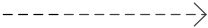
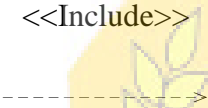
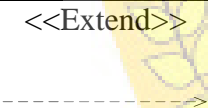
sebuah sistem pengembangan perangkat lunak berbasis OOP (*Object Oriented Programming*).

2.7.1 Use Case Diagram

Use case diagram adalah rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. Berikut pengertian Use case diagram menurut Satzinger, “Use Case Diagram merupakan rangkaian tindakan yang dilakukan oleh sistem, aktor mewakili user atau sistem lain yang berinteraksi dengan sistem yang dimodelkan”. (Satzinger, 2011). Dapat dilihat pada tabel 2.1

Tabel 2.1 Simbol Use Case Diagram

Symbol	Penjelasan
	<p><i>Actor</i></p> <p>Menspesifikasikan seperangkat peranan yang user sistem dapat diperankan ketika berinteraksi dengan use case.</p>
	<p><i>Association</i></p> <p>Menggambarkan interaksi antara actor dan use case</p>
	<p><i>Generalization</i></p> <p>Relasi antar use case, dimana salah satunya dalam bentuk yang lebih umum dari yang lain.</p>
	<p><i>Use Case</i></p>

	Sebuah deskripsi dari seperangkat aksi-aksi berurutan yang ditampilkan pada sebuah sistem.
	<i>System</i> Tempat seluruh aktivitas-aktivitas sistem yang sedang berjalan.
	<i>Dependancy</i> Untuk menggambarkan ketergantungan sebuah <i>use case</i> dengan <i>use case</i> lainnya.
	<i>Include</i> Menggambarkan bahwa keseluruhan dari sebuah <i>use case</i> merupakan fungsionalitas <i>use case</i> lainnya.
	<i>Extend</i> Menggambarkan hubungan antara <i>use case</i> dimana sebuah <i>use case</i> merupakan fungsionalitas <i>use case</i> lainnya apabila kondisi tertentu dipenuhi.

2.7.2 Activity Diagram

Diagram aktivitas adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. Diagram aktivitas mempunyai peran seperti halnya flowchart, akan tetapi perbedaannya dengan flowchart adalah

diagram aktivitas bisa mendukung perilaku paralel sedangkan flowchart tidak bisa.

(Satzinger, 2011). Dapat dilihat pada tabel 2.2

Tabel 2.2 Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

2.7.3 Class Diagram

Diagram kelas atau *Class diagram* sangat membantu dalam visualisasi struktur kelas dari suatu sistem. Hal ini disebabkan karena *class* adalah deskripsi kelompok objek-objek dengan properti, operasi dan relasi yang sama. Disamping itu diagram

kelas bisa memberikan pandangan global atas sebuah sistem. Hal tersebut tercermin dari *class-class* yang ada dan relasinya satu dengan lainnya. Itulah sebabnya diagram kelas menjadi diagram yang paling populer di UML. (Satzinger, 2011)

1. *Assosiasi* adalah *class-class* yang berhubungan satu sama lain secara konseptual. yaitu menghubungkan dua kelas menjadi satu asosiasi.
2. *Attribut* adalah properti dari sebuah kelas. *Attribut* ini menjelaskan batas nilai yang mungkin ada pada objek dari kelas. Sebuah kelas mungkin mempunyai nol atau lebih *attribut*.

Operasi adalah sesuatu yang bisa dilakukan oleh sebuah *class* atau yang anda (atau *class* yang lain) dapat lakukan untuk sebuah *class*. Sama halnya dengan *attribut*, kita bisa juga memberikan tambahan informasi untuk operasi dengan menambahkan parameter yang akan dilakukan oleh operasi dengan tanda kurung.

2.8 Microsoft Visual Studio 2010

Visual basic 2010 adalah merupakan salah satu bagian dari produk pemrograman terbaru yang dikeluarkan oleh *Microsoft Visual Studio 2010*. Sebagai produk lingkungan pengembangan terintegrasi atau *IDE* andalan yang dikeluarkan oleh *Microsoft*, visual studio 2010 menambahkan perbaikan-perbaikan fitur fitur baru yang lebih lengkap visual studio pendahuluannya, yaitu *Microsoft visual studio 2008*. (Wahana Komputer; 2010).

Sedangkan menurut aswan (2012) *Visual Basic 2010* adalah salah satu bagian dari *microsoft visual studio 2010*. Sebuah alat yang digunakan oleh pengembang *windows* dari berbagai level untuk mengembangkan dan membangun aplikasi yang

bergerak diatas system. *NET Framework*, dengan menggunakan bahasa *BASIC*, *Visual Basic* menyediakan cara cepat dan mudah untuk membuat aplikasi. Setiap generasi baru dari perangkat lunak bahasa pemograman datang karena adanya keterbatasan dari generasi sebelumnya. Teknologi *device, hardware, network* dan internet baru yang muncul menyebabkan bahasa pemograman yang ada tidak lagi menjadi alat yang ideal untuk mengembangkan perangkat lunak yang dapat bekerja dengan teknologi baru tersebut.

2.8.1 *Visual Basic .NET*

Menurut Alexander F.K. Sibero, *visual basic .NET* adalah bahasa pemrograman yang dikembangkan oleh perusahaan Microsoft. *Visual basic .NET* merupakan pengembangan dari versi sebelumnya, yaitu *visual basic 6.0*. Perbedaan mendasar antara *visual basic .NET* dengan versi-versi sebelumnya adalah kemampuan OOP (*Object Oriented Programming*) yang telah ditanamkan pada *visual basic .NET*. (Sibero, 2010). Bahasa program tersebut tersusun mirip dengan bahasa Inggris yang biasa sederhana yang berfungsi sebagai pembelajaran bagi konsep dasar pemrograman komputer, Sejak saat itu, banyak versi *BASIC* yang dikembangkan untuk digunakan pada berbagai platform komputer.