

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Pakar

II.1.1. Sistem

Sistem Merupakan Kumpulan elemen yang saling berkaitan yang bertanggung jawab memproses masukan (*input*) sehingga menghasilkan (*output*). Suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan merupakan kegiatan strategi dari suatu organisasi, serta menyediakan laporan-laporan yang diperlukan oleh pihak luar. (Kusrini; 2007: 13)

II.1.2. Pakar

Pakar adalah seorang individu yang memiliki pengetahuan khusus, pemahaman, pengalaman, metode-metode yang digunakan untuk memecahkan persoalan dalam bidang tertentu.

Seorang pakar memiliki kemampuan kepakaran, yaitu :

1. Dapat mengenali dan merumuskan suatu masalah.
2. Menyelesaikan masalah dengan cepat dan tepat.
3. Menjelaskan solusi dari suatu masalah.
4. Restrukturisasi pengetahuan.
5. Belajar dari pengalaman

Selain itu, pakar juga memiliki kemampuan untuk mengaplikasikan pengetahuannya memberikan saran serta pemecahan masalah pada domain tertentu. Ini merupakan pekerjaan pakar, memberikan pengetahuan tentang bagaimana seorang melaksanakan tugas menyelesaikan masalah. (Rika Rosnelly ; 2012: 10)

II.1.3. Sistem Pakar

Sistem pakar adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar. Pakar yang dimaksud di sini adalah orang yang mempunyai keahlian khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam. Terdapat 2 bagian komponen utama dalam sistem pakar yaitu basis pengetahuan dan mesin inferensi. Basis pengetahuan yang digunakan yaitu dengan *rule-based reasoning* (penalaran berbasis aturan). Pada penalaran berbasis aturan, pengetahuan direpresentasikan dengan menggunakan aturan berbentuk *IF-THEN*. Mesin inferensi yang digunakan adalah forward chaining yaitu pencocokan fakta atau pernyataan dimulai dari bagian sebelah kiri. Dengan kata lain, penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis. (Rika Rosnelly ; 2012)

Keuntungan Sistem Pakar adalah sebagai berikut :

- a. Menjadikan pengetahuan dan nasehat mudah didapat.
- b. Meningkatkan *output* dan produktivitas.
- c. Menyimpan kemampuan dan keahlian pakar

- d. Meningkatkan penyelesaian masalah, menerusi paduan pakar.
- e. Meningkatkan reliabilitas.
- f. Memberikan *respons* (jawaban) yang cepat.
- g. Merupakan penduan yang *intelligence* (cerdas).
- h. Dapat bekerja dengan informasi yang lengkap dan mengandung ketidakpastian.
- i. *Intelligence database* (basis data cerdas), bahwa sistem pakar dapat digunakan untuk mengakses basis data dengan cara cerdas.

Kelemahan Sistem Pakar sebagai berikut :

- a. Masalah dalam mendapatkan pengetahuan dimana pengetahuan tidak selalu bisa didapatkan dengan mudah, kadangkala pakar dari masalah yang kita buat tidak ada, dan walaupun ada kadang-kadang pendekatan yang dimiliki pakar berbeda-beda.
- b. Untuk membuat suatu sistem pakar yang benar-benar berkualitas tinggi sangatlah sulit dan memerlukan biaya yang sangat besar untuk pengembangan dan pemeliharaannya.
- c. Boleh jadi sistem tak dapat membuat keputusan.
- d. Sistem pakar tidaklah 100% menguntungkan, walaupun seorang tetap tidak sempurna atau tidak selalu benar. Oleh karena itu perlu diuji ulang secara teliti sebelum digunakan. Dalam hal ini peran manusia tetap merupakan faktor dominan.

II.2. Metode Inferensi

Inferensi merupakan proses untuk menghasilkan informasi dari fakta yang diketahui atau diasumsikan. Inferensi adalah konklusi logis (*logical conclusion*) atau implikasi berdasarkan informasi yang tersedia. Dalam sistem pakar proses inferensi dilakukan dalam suatu modul yang disebut *Inference Engine* (Mesin Inferensi).

Ketika representasi pengetahuan (RP) pada bagian knowledge base telah lengkap, atau paling tidak telah berada pada level yang cukup akurat, maka RP tersebut telah siap digunakan. *Inference engine* merupakan modul yang berisi program tentang bagaimana mengendalikan proses *reasoning*. (Kusrini ; 2008: 8)

II.2.1. Runut Maju (*Forward Chaining*)

Forward chaining berarti menggunakan himpunan aturan kondisi-aksi. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan di jalankan, kemudian aturan tersebut dijalankan. Mungkin proses menambahkan data ke memori kerja. Proses diulang sampai ditemukan suatu hasil.

Metode inferensi runut maju (*Forward chaining*) cocok digunakan untuk menangani masalah pengendalian (*Controlling*) dan peramalan (*Prognosis*). Untuk memudahkan pemahaman mengenai metode ini, akan diberikan ilustrasi khusus pembuatan sistem pakar sebagai berikut :

Ingin diperoleh konklusi dari daftar konklusi yang ada berdasarkan premis-premis dalam aturan dan fakta yang diberikan oleh *user*. Berikut ini adalah daftar aturannya:

Aturan 9:

Jika Premis 1
Dan Premis 2
Dan Premis 3
Maka Konklusi 1

Aturan 10:

Jika Premis 1
Dan Premis 3
Dan Premis 4
Maka Konklusi 2

Aturan 11:

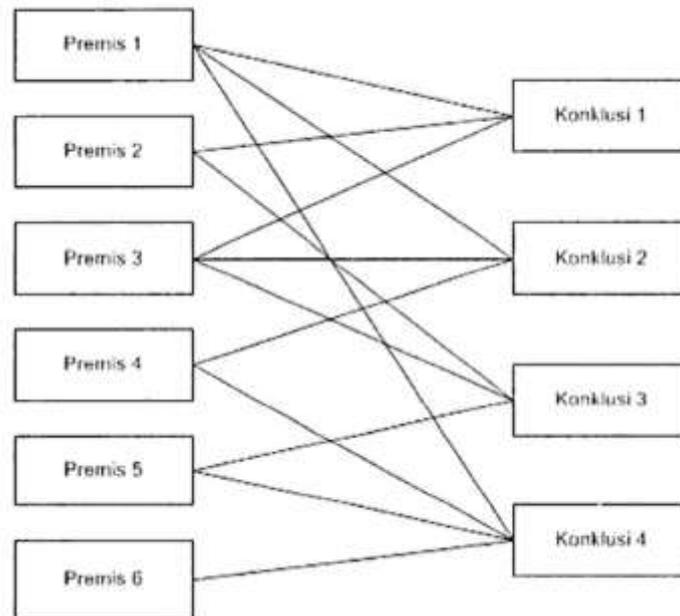
Jika Premis 2
Dan Premis 3
Dan Premis 5
Maka Konklusi 3

Aturan 12:

Jika Premis 1
Dan Premis 4
Dan Premis 5
Dan Premis 6

Maka Konklusi 4

Jika aturan ini kita gambarkan sebagai sebuah *graph* yang memetakan antara premis-premis dan konklusi-konklusi akan tampak seperti Gambar II.1.



Gambar II.1. Graph Pengetahuan

Sumber : Kusrini, Aplikasi Sistem Pakar ; 2008 : 10

Penelusuran Maju pada kasus ini adalah untuk mengetahui apakah suatu fakta yang dialami oleh pengguna itu termasuk konklusi 1, konklusi 2, konklusi 3, atau konklusi 4 atau bahkan bukan salah satu dari konklusi tersebut, yang artinya sistem belum mampu mengambil kesimpulan karena keterbatasan aturan.

Dalam penalaran ini, user diminta memasukkan premis-premis yang dialami. Untuk memudahkan pengguna, sistem dapat memunculkan daftar premis yang mungkin sehingga user dapat memberikan umpan balik premis mana yang

dialami dengan memilih satu atau beberapa dari daftar premis yang tersedia.

Berarti daftar premisnya adalah :

Premis 1, Premis2, Premis3, Premis4, Premis5, dan Premis6

Berdasarkan premis-premis yang dipilih, maka sistem akan mencari aturan yang sesuai sehingga akan diperoleh konklusinya. Seandainya user memilih premis1, premis2, dan premis3 maka aturan yang terpilih adalah aturan 1 dengan konklusinya adalah konklusi1. Seandainya user memilih premis 1 dan premis 6, maka sistem akan mengarah pada aturan 4 dengan konklusinya adalah konklusi 4, tetapi karena aturan tersebut premisnya adalah premis 1, premis 4, premis 5 dan premis 6, maka premis-premis yang dipilih oleh *user* tidak cukup untuk mengambil kesimpulan konklusi 4 sebagai konklusi terpilih. (Kusrini ; 2008 : 10)

II.3. Zakat

Zakat menurut bahasa, berarti *nama'* yang artinya kesuburan, *thaharah* berarti kesucian, *barakah* berarti keberkatan dan berarti juga *tazkiyah tathhir* yang artinya mensucikan. Syara' memakai kata tersebut untuk kesua arti ini. Pertama, dengan zakat diharapkan akan mendatangkan kesuburan pahala. Karenanya dinamakanlah “ harta yang dikeluarkan itu” dengan zakat. Kedua, zakat merupakan suatu kenyataan jiwa yang suci dari kikir dan dosa.

Imam Nawawi mengatakan bahwa, “zakat mengandung makna kesuburan.” Kata yang dipakai untuk dua arti, subur dan suci. Zakat digunakan untuk sedekah wajib, sedekah sunnah, nafkah, kemaafan dan kebenaran.

Menurut garis besarnya, zakat terbagi menjadi dua. Pertama, *Zakat Mal* (harta) : emas, perak, binatang, tumbuh-tumbuhan (buah-buahan dan biji-bijian) dan barang perniagaan. Kedua, *Zakat Nafs*, zakat jiwa yang juga disebut juga “*Zakatul Fithrah.*” (zakat yang diberikan berkenan dengan selesainya mengerjakan *shiyam* (puasa) yang di fardhukan).(M.Hasbi ash-Shiddieqy; 2012 :3)

II.3.1. Zakat Mal

Zakat Mal, Atau zakat harta benda, telah difardhukan Allah sejak permulaan Islam sebelum Nabi SAW berhijrah ke Madinah. Tidak heran urusan ini amat cepat diperhatikan Islam, karena urusan tolong-menolong adalah urusan yang sangat diperlukan dalam pergaulan hidup dan dibutuhkan oleh seluruh lapisan masyarakat.

Pada awalnya, zakat di fardhukan tanpa ditentukan kadarnya dan tanpa pula diterangkan dengan jelas harta-harta yang dikenakan zakatnya. Syara' hanya menyuruh mengeluarkan zakat. Banyak sedikitnya terserah kepada kemauan dan kebaikan para pemberi zakat sendiri. Hal tersebut berjalan hingga tahun kedua Hijriyah. Mereka yang menerima pada masa itu adalah dua golongan saja, yaitu : fakir dan miskin.

Pada tahun kedua Hijriyah, bersamaan dengan tahun 623 Masehi, barulah Syara' menentukan harta-harta yang dizakatkan, serta kadar masing-masing. (M.Hasbi ash-Shiddieqy; 2012 :8)

Allah berfirman dalam Al Qur'an :

... وَالَّذِينَ يَكْتُمُونَ الذَّهَبَ وَالْفِضَّةَ وَلَا
 يُنْفِقُونَهَا فِي سَبِيلِ اللَّهِ فَبَشِّرْهُمْ بِعَذَابٍ أَلِيمٍ
 ﴿٩٤﴾ يَوْمَ نُحْمَىٰ عَلَيْهَا فِي نَارِ جَهَنَّمَ فَتُكْوَىٰ
 بِهَا جِبَاهُهُمْ وَجُنُوبُهُمْ وَظُهُورُهُمْ هَذَا مَا
 كُنْتُمْ تَكْتُمُونَ

...Dan orang-orang yang menyimpan emas dan perak dan tidak menafkahkanya pada jalan Allâh, maka beritahukanlah kepada mereka (bahwa mereka akan mendapat) siksa yang pedih. Pada hari dipanaskan emas dan perak itu dalam neraka Jahannam, lalu dahi, lambung dan punggung mereka dibakar dengannya, (lalu dikatakan) kepada mereka: “Inilah harta bendamu yang kamu simpan untuk dirimu sendiri, maka rasakanlah sekarang (akibat dari) apa yang kamu simpan itu”. (Qs. at-Taubah/9:34-35)

Tafsiran Hadist dari Ibnu Katsir rahimahullâh, dari Rasûlullâh Shallallâhu 'Alaihi Wasallam, beliau berkata:

“Dinyatakan bahwa setiap orang yang mencintai sesuatu dan lebih mendahulukannya dibanding ketaatan kepada Allâh, niscaya ia akan disiksa dengannya. Dan dikarenakan orang-orang yang disebut pada ayat ini lebih suka untuk menimbun harta kekayaannya daripada mentaati keridhaan Allâh, maka mereka akan disiksa dengan harta kekayaannya. Sebagaimana halnya Abu Lahab, dengan dibantu oleh istrinya, ia tak henti-hentinya memusuhi Rasûlullâh Shallallâhu 'Alaihi Wasallam, maka kelak pada hari kiamat, istrinya akan berbalik ikut serta menyiksa dirinya. Di leher istri Abu Lahab akan terikat tali dari

sabut, dengannya ia mengumpulkan kayu-kayu bakar di neraka, lalu ia menimpakannya kepada Abu Lahab. Dengan cara ini, siksa Abu Lahab semakin terasa pedih, karena dilakukan oleh orang yang semasa hidupnya di dunia paling ia cintai. Demikianlah halnya para penimbun harta kekayaan. Harta kekayaan yang sangat ia cintai, kelak pada hari kiamat menjadi hal yang paling menyedihkannya. Di neraka Jahannam, harta kekayaannya itu akan dipanaskan, lalu digunakan untuk membakar dahi, perut, dan punggung mereka”.

Tafsiran Hadist dari Ibnu Hajar al-Asqalâni *rahimahullâh*, dari *Rasûlullâh Shallallâhu 'Alaihi Wasallam*, beliau berkata:

“Dan hikmah dikembalikannya seluruh harta yang pernah ia miliki, padahal hak Allâh (zakat) yang wajib dikeluarkan hanyalah sebagiannya saja, ialah karena zakat yang harus dikeluarkan menyatu dengan seluruh harta dan tidak dapat dibedakan. Dan karena harta yang tidak dikeluarkan zakatnya adalah harta yang tidak suci”.

II.3.2. Akuisisi Pengetahuan

Kegiatan ini merupakan proses untuk mendapatkan pengetahuan mengenai ketentuan zakat pertanian dan peternakan, berikut table yang dihasilkan :

Tabel II.1. Tabel Akuisisi Pengetahuan

Jenis	Nisab	Kadar
Padi	5 wasaq atau 750 kg	5% untuk pengairan dengan irigasi 10% untuk pengairan tadah hujan
Gabah	1350 gr	5% untuk pengairan dengan irigasi 10% untuk pengairan tadah hujan
Tanaman lain/ buah-buahan/ sayuran dll.	5 wasaq atau 750 kg	5% untuk pengairan dengan irigasi 10% untuk pengairan tadah hujan

Sapi /Kerbau	30-39 ekor	1 ekor anak sapi/kerbau berumur 1 s/d 2 tahun
	40-59 ekor	1 ekor anak sapi/kerbau berumur 2 s/d 3 tahun
	60-69 ekor	2 ekor anak sapi/kerbau berumur 1 s/d 2 tahun
	70-79 ekor	1 ekor anak sapi/kerbau berumur 1 s/d 2 tahun dan 1 ekor anak sapi/kerbau berumur 1 s/d 2 tahun
	80-89 ekor	2 ekor anak sapi/kerbau berumur 1 s/d 2 tahun
	90-99 ekor	3 ekor anak sapi/kerbau berumur 1 s/d 2 tahun
	100-109 ekor	1 ekor anak sapi/kerbau berumur 1 s/d 2 tahun dan 1 ekor anak sapi/kerbau berumur 1 s/d 2 tahun
	110-119 ekor	2 ekor anak sapi/kerbau berumur 2 s/d 3 tahun dan 1 ekor anak sapi/kerbau berumur 1 s/d 2 tahun
	120-129 ekor	3 ekor anak sapi/kerbau berumur 2 s/d 3 tahun dan 4 ekor anak sapi/kerbau berumur 1 s/d 2 tahun
Kambing/Domba	40-120 ekor	1 ekor kambing
	121-200 ekor	2 ekor kambing
	201-299 ekor	3 ekor kambing
	300-399 ekor	4 ekor kambing
Ternak Unggas	85 gr emas	2,5 %

Sumber : Anggi Endah Purnama; 2013 : 5

II.4. Sekilas Sejarah PHP

PHP Hypertext Preprocessor atau sering disebut PHP merupakan bahasa pemrograman berbasis *server-side* yang dapat melakukan *parsing script* php menjadi *script* web sehingga dari sisi *client* menghasilkan suatu tampilan yang menarik. PHP merupakan pengembangan dari FI atau *Form Interface* yang dibuat oleh Rasmus Lerdof pada tahun 1995.

Berbeda dengan HTML, kode PHP tidak diberikan secara langsung oleh server ketika ada permintaan atau *request* dari sisi *client* namun dengan cara pemrosesan dari sisi server. Kode PHP disisipkan pada kode HTML. Perbedaan

dari kode (*script*) HTML dan PHP yaitu setiap kode PHP ditulis selalu diberi tag pembuka yaitu `<?php` dan pada akhir kode PHP diberi tag penutup yaitu `?>`.

PHP sangat berperan besar ketika Anda ingin membuat website keren yang dinamis karena dapat melakukan banyak hal, seperti membaca file, menulis file, menampilkan gambar, animasi atau video, dan yang paling cocok adalah dapat melakukan koneksi terhadap *database*. (YM Kusuma Ardhana; 2012: 88).

II.4.1 Embedded Script

Kode (*script*) PHP yang sering disebut dengan istilah *embedded script* yaitu *script* PHP yang disisipkan diantara *script* HTML. Jadi dapat dikatakan *script* PHP hanya ditulis atau disisipkan ketika dibutuhkan saja, seperti menampilkan data dari *database*, meng-*upload* file, *delete* data, *edit* data dan lain sebagainya. (YM Kusuma Ardhana; 2012: 88).

Contoh program PHP adalah sebagai berikut :

```
<html>
  <head>
    <tittle> ... Embedded Script ... </tittle>
  </head>
  <body>
    <?php
      Echo "Hello, Selamat Datang di Dunia PHP";
    ?>
  </body>
</html>
```

II.5. Dreamweaver

Dreamweaver adalah sebuah *software* web *design* yang menawarkan cara mendesain website dengan dua langkah sekaligus dalam satu waktu, yaitu

mendesain dan memprogram. *Dreamweaver* memiliki satu jendela mini disebut *HTML source*, tempat kode-kode HTML ditulis. Setiap kali kita mendesain web, seperti menulis kata-kata, meletakkan gambar, membuat tabel dan proses lainnya, tag-tag HTML akan tertulis secara langsung mengiringi proses pengaturan *website*. Artinya kita memiliki kesempatan untuk mendesain *website* hanya dengan menulis tag-tag dan teks lain di jendela *HTML Source* dan hasilnya dapat dilihat langsung dilayar.

Jendela tersebut hanya merupakan salah satu kelebihan *dreamweaver*. Selain itu ia mampu mengenali tag-tag lain dari luar HTML. Seperti *Cold Fusion* dan *ASP*, serta mendukung script-script dinamik HTML dan *CSS style*. Jika saatnya tiba anda akan mengenal hampir sebagian besar fasilitas yang ada di *dreamweaver* ini. *Dreanweaver* memberikan pilihan *workspace* baru yang terintegrasi , menyediakan lingkungan pengembangan yang lebih lengkap dan window dokumen yang menyebut pemakai.

II.6. MySQL

MySQL adalah *Software RDBMS* (atau server *database*) yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user (multi-user)*, dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*).

Saat ini, *MySQL* banyak digunakan di berbagai kalangan untuk melakukan penyimpanan dan pengolahan data, mulai dari kalangan akademis sampai ke industri, baik industri kecil, menengah maupun besar.

Lisensi MySQL terbagi menjadi dua. Anda dapat menggunakan MySQL sebagai produk *open source* di bawah GNU *General Public License* (gratis) atau dapat membeli lisensi dari versi komersialnya. MySQL versi komersial tentu memiliki nilai lebih atau kemampuan-kemampuan yang tidak disertakan pada versi gratis. Pada kenyataannya, untuk keperluan industri menengah kebawah, versi gratis masih dapat digunakan dengan baik. (Budi Raharjo; 2012: 22).

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

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.

Sejarah UML sendiri terbagi dalam dua fase; sebelum dan sesudah munculnya UML. Dalam fase sebelum, UML sebenarnya sudah mulai diperkenalkan sejak tahun 1990an namun notasi yang dikembangkan oleh para ahli analisis dan desain berbeda-beda, sehingga dapat dikatakan belum memiliki standarisasi.

Fase kedua; dilandasi dengan pemikiran untuk mempersatukan metode tersebut dan dimotori 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 Rasional Software

Corporation dan Ivar Jacobson dengan metode OOSE (*Object-Oriented Software Engineering*) yang bekerja pada perusahaan Objectory Rasional.

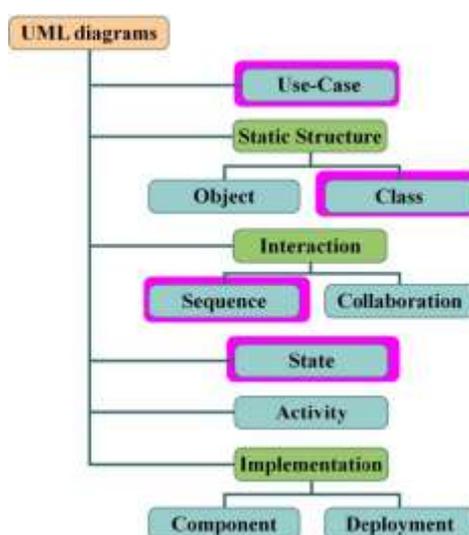
Sebagai pencetus metode-metode tersebut mereka bertiga berinisiatif untuk menciptakan bahasa pemodelan terpadu sehingga pada tahun 1996 mereka berhasil merilis UML versi 0.9 dan 0.91 melalui *Request for Proposal* (RFP) yang dikeluarkan oleh OMG.

Kemudian pada Januari 1997 IBM, *ObjecTime*, *Platinum Technology*, *Ptech*, *Taskon*, *Reich Technologies* dan *Softeam* 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 (<http://www.uml.org/>).

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 pembuat program.

Secara filosofi UML diilhami 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. (Haviluddin; 2011: 01).

Berikut gambar dari diagram UML



Gambar II.2. Diagram UML

Sumber : Havaluddin, Memahami Penggunaan UML (Unified Modelling Language); 2011 : 02

II.7.1. Tujuan Pemanfaatan UML

Tujuan dari penggunaan diagram seperti diungkapkan oleh Schuller J. (2004), *“The purpose of the diagrams is to present multiple views of a system; this set of multiple views is called a model”*.

Berikut tujuan utama dalam desain UML adalah (Havaluddin; 2011: 02) :

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan visual yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.
2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.
3. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.
4. Memberikan dasar formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek (OO).
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.
7. Memiliki integrasi praktik terbaik.

II.7.2. Komponen-komponen UML

Sejauh ini para pakar merasa lebih mudah dalam menganalisa dan mendesain atau memodelkan suatu sistem karena UML memiliki seperangkat aturan dan notasi dalam bentuk grafis yang cukup spesifik.

Komponen atau notasi UML diturunkan dari 3 (tiga) notasi yang telah ada sebelumnya yaitu Grady Booch, OOD (*Object-Oriented Design*), Jim Rumbaugh, OMT (*Object Modelling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

Pada UML versi 2 terdiri atas tiga kategori dan memiliki 13 jenis diagram yaitu (Haviluddin; 2011: 03) :

1. Struktur Diagram

Menggambaran elemen dari spesifikasi dimulai dengan kelas, obyek, dan hubungan mereka, dan beralih ke dokumen arsitektur logis dari suatu sistem.

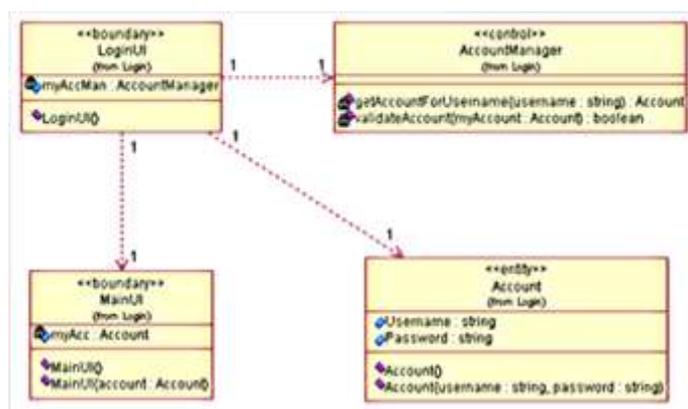
Struktur diagram dalam UML terdiri atas :

1) *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 menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.

Class memiliki tiga area pokok :

1. Nama (dan *stereotype*)
2. Atribut
3. Metoda



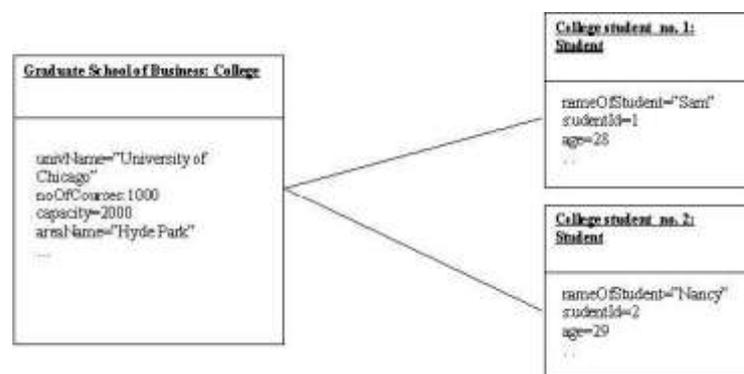
Gambar II.3. Notasi class diagram

Sumber : Haviluddin, Memahami Penggunaan UML (Unified Modelling Language); 2011 : 03

2) *Object diagram*

Object diagram menggambarkan kejelasan kelas dan warisan dan kadang-kadang diambil ketika merencanakan kelas, atau untuk membantu pemangku kepentingan non-program yang mungkin menemukan diagram kelas terlalu abstrak.

Berikut notasi *object diagram*.

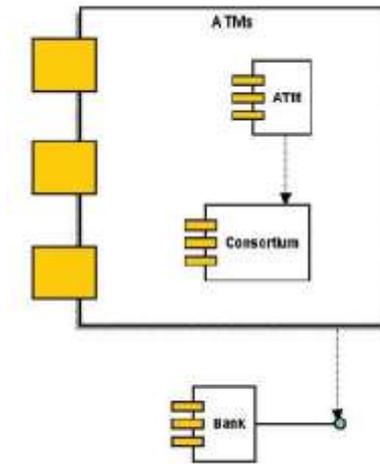


Gambar II.4. Notasi *object diagram*

Sumber : Havaluddin, Memahami Penggunaan UML (Unified Modelling Language); 2011 : 03

3) *Component diagram*

Component diagram menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini dilaksanakan.

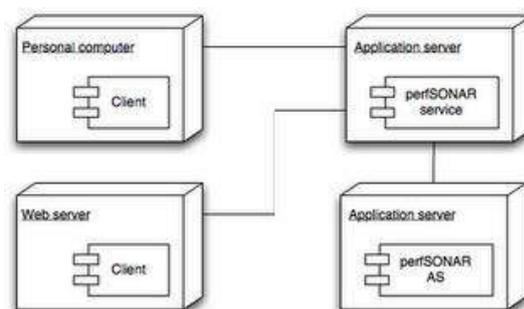


Gambar II.5. Notasi *component diagram*

Sumber : Haviluddin, Memahami Penggunaan UML (Unified Modelling Language); 2011 : 03

4) *Deployment diagram (Collaboration diagram in version 1.x)*

Deployment diagram memberikan gambaran dari arsitektur fisik perangkat lunak, perangkat keras, dan artefak dari sistem. *Deployment diagram* dapat dianggap sebagai ujung spektrum dari kasus penggunaan, menggambarkan bentuk fisik dari sistem yang bertentangan dengan gambar konseptual dari pengguna dan perangkat berinteraksi dengan sistem.

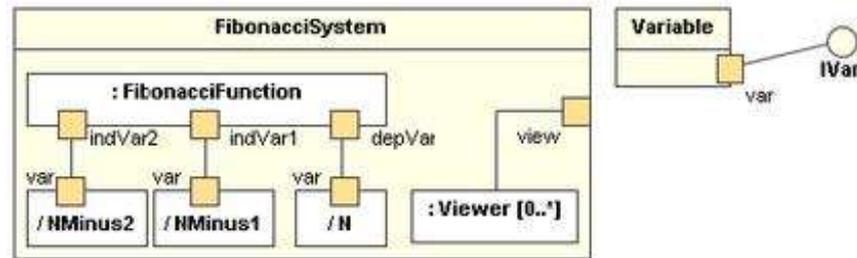


Gambar II.6. Notasi *deployment diagram*

Sumber : Haviluddin, Memahami Penggunaan UML (Unified Modelling Language); 2011 : 04

5) *Composite structure diagram*

Sebuah diagram struktur komposit mirip dengan diagram kelas, tetapi menggambarkan bagian individu, bukan seluruh kelas. Kita dapat menambahkan konektor untuk menghubungkan dua atau lebih bagian dalam atau ketergantungan hubungan asosiasi.

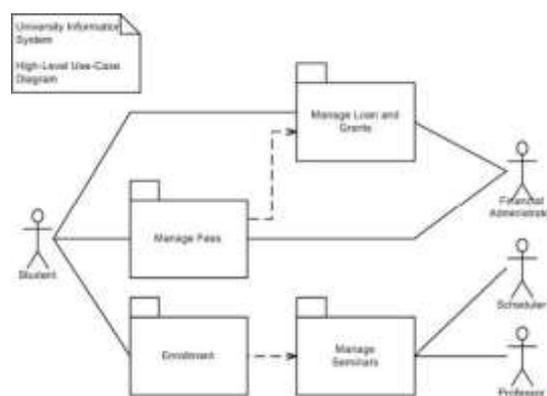


Gambar II.7. Notasi composite diagram

Sumber : Havaluddin, Memahami Penggunaan UML (Unified Modelling Language); 2011 : 04

6) *Package diagram*

Paket diagram biasanya digunakan untuk menggambarkan tingkat organisasi yang tinggi dari suatu proyek *software*. Atau dengan kata lain untuk menghasilkan diagram ketergantungan paket untuk setiap paket dalam Pohon Model.



Gambar II.8. Notasi Package diagram

Sumber : Havaluddin, Memahami Penggunaan UML (Unified Modelling Language); 2011 : 04

2. Behavior Diagram

Menggambarkan ciri-ciri behavior/metode/ fungsi dari sebuah sistem atau business process. Behavior diagram dalam UML terdiri atas :

1) *Use case diagram*

Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai *elips horizontal* dalam suatu diagram UML *use case*.

Use Case memiliki dua istilah :

1. *System use case*; interaksi dengan sistem.
2. *Business use case*; interaksi bisnis dengan konsumen atau kejadian nyata



Gambar II.9. Notasi *Use case diagram*

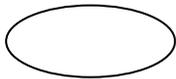
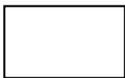
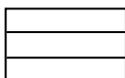
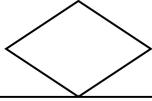
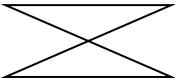
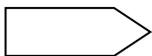
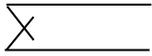
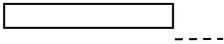
Sumber : Haviluddin, Memahami Penggunaan UML (Unified Modelling Language); 2011 : 04

2) *Activity diagram*

Menggambarkan aktifitas-aktifitas, objek, state, transisi state dan event. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas.

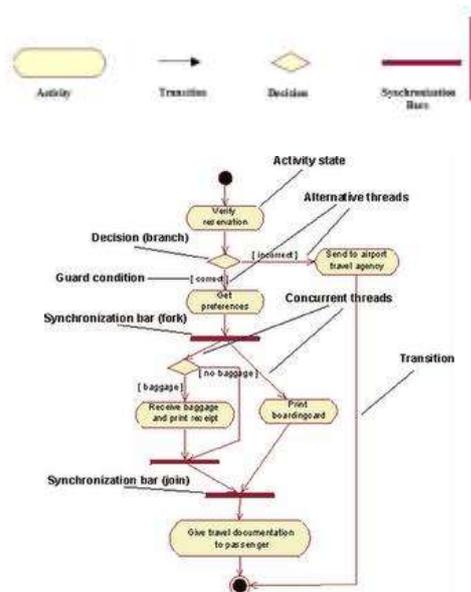
Tabel II.2. Simbol-simbol dalam UML

Simbol	Keterangan
--------	------------

	Actor; mewakili peran orang
	Use Case; Interaksi antara system dan actor
	Sistem
	Class
	Message
	Titik awal
	Titik akhir
	Activity
	Pilihan untuk pengambilan keputusan
	Fork; digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	Rake; menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan
	Aliran akhir (Flow Final)
	Activation (Mewakili sebuah eksekusi operasi dari participant)
	Lifeline (Setiap participant terhubung dengan garis titik-titik)



Sumber : Windu Gata ; 2013 : 04

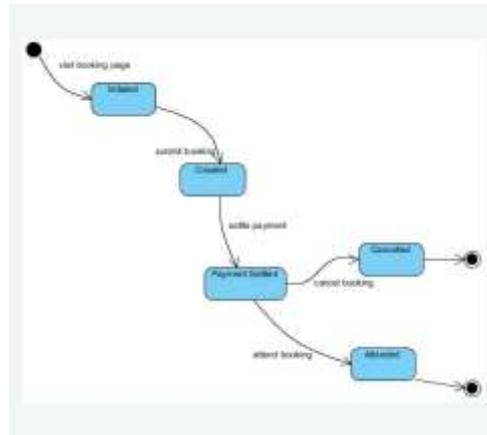


Gambar II.10. Notasi Activity diagram

Sumber : Havaluddin, Memahami Penggunaan UML (Unified Modelling Language); 2011 : 04

3) *State Machine diagram (State chart diagram in version 1.x)*

Menggambarakan *state*, transisi *state* dan *event*.



Gambar II.11. Notasi State Machine diagram

Sumber : Havaluddin, Memahami Penggunaan UML (Unified Modelling Language); 2011 : 05

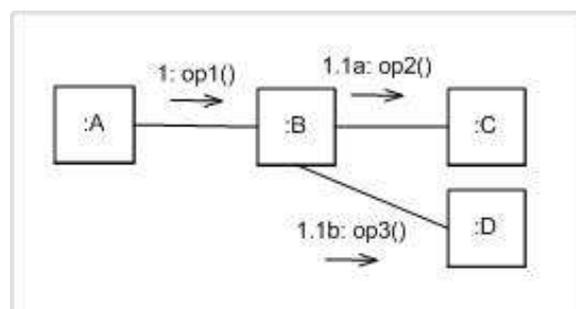
3. Interaction diagram

Bagian dari behavior diagram yang menggambarkan interaksi objek.

Interaction diagram dalam UML terdiri atas :

1) *Communication diagram*

Serupa dengan sequence diagram, tetapi diagram komunikasi juga digunakan untuk memodelkan perilaku dinamis dari use case. Bila dibandingkan dengan Sequence diagram, diagram komunikasi lebih terfokus pada menampilkan kolaborasi benda daripada urutan waktu.

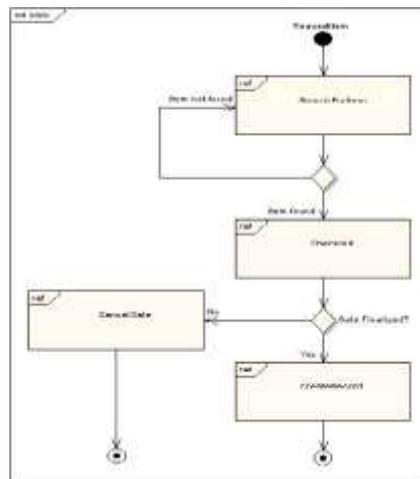


Gambar II.12. Notasi *communication diagram*

Sumber : Havaluddin, *Memahami Penggunaan UML (Unified Modelling Language)*; 2011 : 05

2) *Interaction Overview diagram*

Interaksi overview diagram berfokus pada gambaran aliran kendali interaksi dimana node adalah interaksi atau kejadian interaksi.

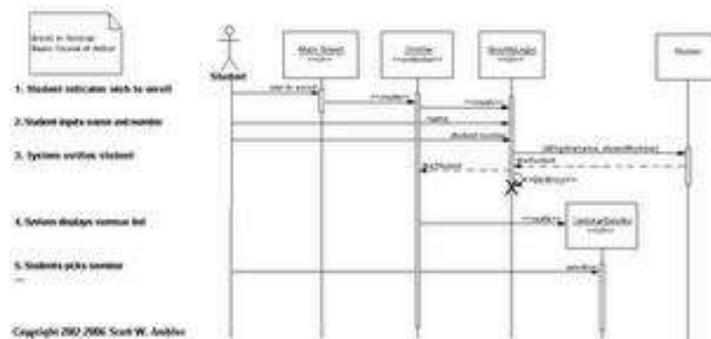


Gambar II.13. Notasi *overview diagram*

Sumber : Havaluddin, *Memahami Penggunaan UML (Unified Modelling Language)*; 2011 : 05

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*.

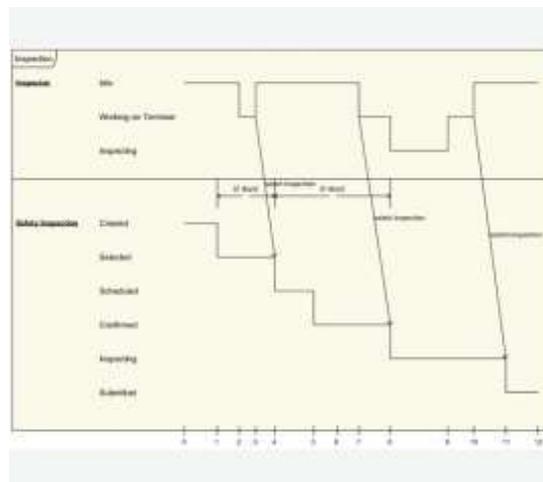


Gambar II.14. Notasi Sequence diagram

Sumber : Havaluddin, Memahami Penggunaan UML (Unified Modelling Language); 2011 : 05

4) Timing diagram

Timing diagram di UML didasarkan pada diagram waktu hardware awalnya dikembangkan oleh para insinyur listrik.



Gambar II.15. Notasi Sequence diagram

Sumber : Havaluddin, Memahami Penggunaan UML (Unified Modelling Language); 2011 : 05