

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Informasi

Sistem informasi secara sederhana dapat diartikan sebagai kumpulan dari beberapa komponen yang saling berinteraksi untuk mencapai hasil dari satu tujuan. Pengertian sederhana ini sesuai dengan pendapat O'Brien (2006,p5)" Sistem Informasi dapat merupakan kombinasi teratur apapun dari orang-orang, *hardware*, *software*, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah dan menyebarkan informasi dalam sebuah organisasi". Menurut Whitten (2004,p12) "*information system is an arrangement of people, data, process and information technology that interact to collect, process, store, and provide as output the information needed to support an organization*". Defenisi tersebut dapat dijelaskan sistem informasi adalah susunan dari orang, data, pemrosesan dan teknologi informasi yang dibutuhkan untuk mendukung sebuah organisasi (Henny Hendarti; 2011:1)

II.1.1. Sistem Pakar

Dalam ilmu komputer, banya ahli yang berkonsentrasi pada pengembangan kecerdasan buatan atau *Artificial Inteligence* (AI). AI adalah suatu studi khusus di mana tujuannya adalah membuat komputer berpikir dan bertindak seperti manusia. Banyak implementasi AI dalam bidang komputer, misalnya *Decision Support System* (Sistem Pendukung Keputusan), *Robotic*, *Natural Language* (Bahasa Alami), *Neural Network* (Jaringan Saraf), dan lain-lain.

Contoh bidang lain pengembangan kecerdasan buatan adalah sistem pakar yang menggabungkan pengetahuan dan penelusuran data untuk memecahkan masalah yang secara normal memerlukan keahlian manusia. Tujuan pengembangan sistem pakar sebenarnya bukan untuk menggantikan peran manusia, tetapi untuk mensubstitusikan pengetahuan manusia ke dalam bentuk sistem, sehingga dapat digunakan oleh orang banyak (Andi; 2009: 3).

Ada banyak manfaat yang dapat diperoleh dengan mengembangkan system pakar, antara lain :

1. Masyarakat awam non-pakar dapat memanfaatkan keahlian di dalam bidang tertentu tanpa kehadiran langsung seorang pakar.
2. meningkatkan produktivitas kerja, yaitu bertambah efisiensi pekerjaan tertentu serta hasil solusi kerja.
3. penghematan waktu dalam meyelesaikan masalah yang kompleks.
4. Memberikan penyederhanaan solusi untuk kasus-kasus yang kompleks dan berulang-ulang.
5. pengetahuan dari seorang pakar dapat didokumentasikan tanpa ada batas waktu.
6. Memungkinkan penggabungan berbagai bidang pengetahuan dari berbagai pakar untuk dikombinasikan.

Berikut ini merupakan perbandingan antara kemampuan pakar manusia dan sistem komputer yang menjadi pertimbangan pengembangan sistem pakar.

Tabel II.1 Perbandingan Sistem Pakar Dengan Cara Pikir Manusia

Pakar Manusia	Sistem Pakar
Terbatas waktu karena manusia membutuhkan istirahat.	Tidak terbatas karena dapat digunakan kapanpun juga.
Tempat akses bersifat lokal pada suatu tempat saja dimana pakar berada.	Dapat digunakan diberbagai tempat.
Pengetahuan bersifat variabel dan dapat berubah-ubah tergantung situasi.	Pengetahuan bersifat konsisten.
Kecepatan untuk menentukan solusi sifatnya bervariasi.	Kecepatan untuk memberikan solusi konsisten dan lebih cepat daripada manusia.
Biaya yang harus dibayar untuk konsultasi biasanya sangat mahal.	Biaya yang dikeluarkan lebih murah.

Sumber : Andi (2009 : 4)

Selain dari beberapa manfaat yang diperoleh, ada juga kelemahan pengembangan sistem pakar, yaitu :

1. Daya kerja dan produktivitas manusia menjadi berkurang karena semuanya dilakukan secara otomatis oleh sistem.
2. Pengembangan perangkat lunak sistem pakar lebih sulit dibandingkan dengan perangkat lunak konvensional. Hal ini dapat dilihat dari tabel perbandingan berikut ini :

Tabel II.1 Perbandingan Perangkat Lunak Sistem Pakar dan Konvensional

Perangkat Lunak Konvensional	Perangkat Lunak Sistem Pakar
Fokus pada solusi.	Fokus pada permasalahan.
Pengembangan dapat dilakukan secara individu.	Pengembangan dilakukan oleh tim kerja.
Pengembangan secara sekuensial.	Pengembangan secara interaktif.

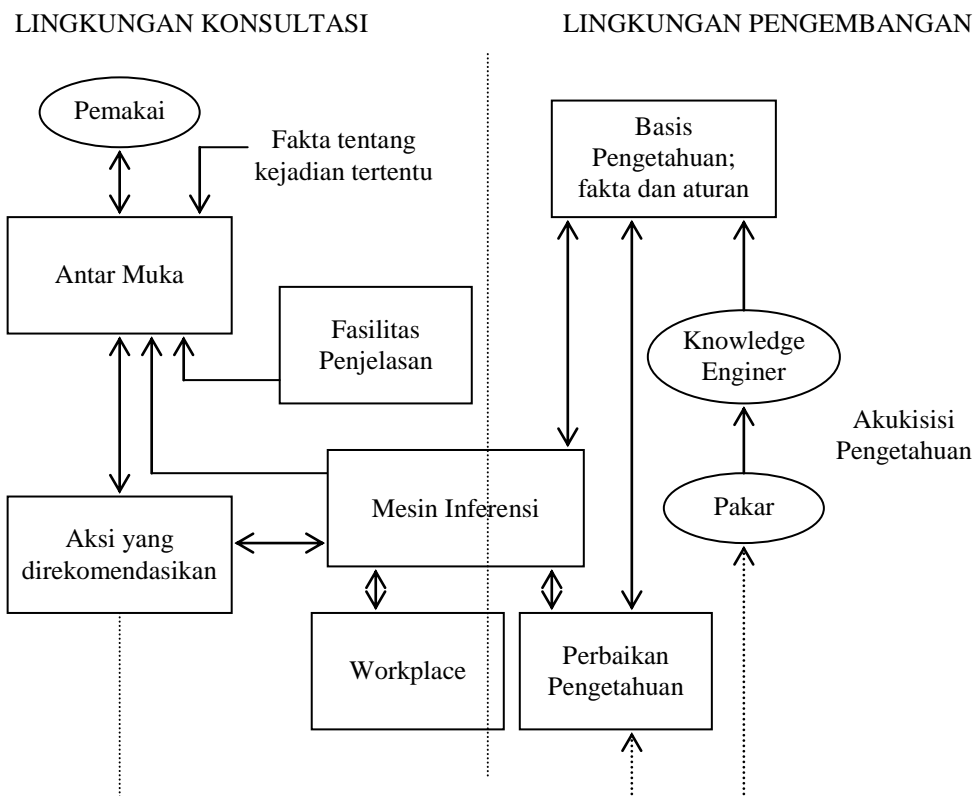
Sumber : Andi (2009 : 4)

II.1.2. Metode Forward Chaining

Suatu perkalian inferensi yang menghubungkan suatu permasalahan dengan solusinya disebut dengan rantai (*chain*). Suatu rantai yang dicari atau dilewati/dilintasi dari suatu permasalahan untuk memperoleh solusinya disebut dengan *forward chaining*. Cara lain menggambarkan *forward chaining* ini adalah dengan penalaran dari fakta menuju konklusi yang terdapat dari fakta (Arhami; 2005 : 111).

II.1.3. Struktur Sistem Pakar

Sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*) (Turban, 1995). Lingkungan pengembangan sistem pakar digunakan untuk memasukkan pengetahuan pakar ke dalam lingkungan sistem pakar, sedangkan lingkungan konsultasi digunakan oleh pengguna yang bukan pakar guna memperoleh pengetahuan pakar. Komponen-komponen sistem pakar dalam kedua bagian tersebut dapat dilihat dalam gambar II.1. berikut ini :



Gambar II.1 Arsitektur sistem pakar (Sumber: Turban 1995)

Komponen-komponen yang terdapat dalam sistem pakar adalah seperti yang terdapat pada gambar II.1, yaitu *User Interface* (antar muka pengguna), basis pengetahuan, akuisisi pengetahuan, mesin inferensi, *workplace*, fasilitas penjelasan, perbaikan pengetahuan (Arhami; 2005: 14).

II.2. Nokia WD2

Salah satu *vendor handphone* terbesar saat ini adalah NOKIA, dimana pada *handphone* Nokia itu sendiri memiliki beberapa golongan seperti salah satunya adalah WD2 yang akan menjadi pembahasan dari penulis. Seperti *handphone* Nokia dengan type 6600 yang juga merupakan salah satu type *handphone* dari golongan Nokia WD2. Spesifikasi hardware dan software secara umum dapat dijelaskan seperti dibawah ini :

1. Ic Power (CCONT) : Mendistribusikan atau membagikan arus.
2. Ic Audio (COBBA) : Ic ini mempunyai fungsi sebagai pengolah sinyal suara yang masuk dari IC RF kemudian diteruskan kepada *speaker*.
3. IC Cpu : CPU merupakan serangkaian komponen yang terintegrasi dan akan berfungsi sesuai dengan tugasnya masing-masing.
4. IC Flash : Komponen ini sebagai media penyimpanan data pada ponsel yang permanen dalam kata dapat diubah dan ditambah dengan data-data yang berada pada computer.
5. EEPROM (*Electrically Erase Progammable Read Only Memori*) : Komponen ini berfungsi sebagai tempat penyimpanan data pada ponsel yang dirancang tidak tergantung pada adanya arus listrik dari ponsel tersebut.
6. MCU (*Master Control Unit*) : MCU adalah data yang ada didalam ponsel yang terletak berada pada IC Audio, data ini bersifat permanen dari pabrik.
7. PPM (*Progress Program Maneger*) : Data ini sama dengan MCU, karena berupa program yang dikirim ke dalam ponsel.
8. IC RAM : Komponen ini pada dasarnya juga merupakan penyimpanan data, tetapi sifatnya hanya sementara (*non volatile*), karena komponen ini cara kerjanya tergantung pada arus listrik yang disimpan dalam konponan ini.
9. IC Charging : Komponen ini akan bekerja secara otomatis pada saat pengisian, yang berfungsi untuk mengisi tegangan baterai yang dikendalikan oleh CPU Melalui IC pengontrol.
10. IC UI : Fungsinya sebagai pengontrol data yang diperintahkan oleh IC CPUY pada Vibrator, Buzzer, Led dan bersifat sebagai saklar otomatis dalam ponsel.

11. IC PA : Fungsinya sebagai penguat akhir sinyal yang akan dipancarkan melalui komponen switch antena yang terdapat pada ponsel.
12. IC RF (Hagar) :Berfungsi sebagai pengontrol sinyal RX(masuk) dan TX(keluar), agar setiap bagian dapat bekerja dengan baik.
13. IC VCO (*Voltage Control Oscillator*) : Fungsinya sebagai isolator atau pembangkit frekuensi yang akan dikirim melalui bagian TX (pemancaran) dan frekuensi yang masuk (penerimaan) agar sama dengan yang dipancarkan.
14. RX/TX : Sebagai jalan untuk masuk keluarnya sinyal.
15. LCD (*Liquid Cristal Display*) : Komponen ponsel yang satu ini berfungsi sebagai alat yang akan menampilkan semua aktivitas dari ponsel, sebagai media komunikasi baca tulis pada ponsel.
16. Keypad : Komponen ini berfungsi sebagai peralatan input yang memberikan perintah data CPU ponsel untuk diproses.
17. Infrared : Infrared adalah suatu alat atau piranti untuk mentransfer data atau program.
18. Bluetooth : Alat ini digunakan untuk mentransfer data atau berfungsi sebagai mediator antara suatu perangkat dengan perangkat lainnya.
19. Baterai : Fungsinya sebagai sumber arus listrik yang diperlukan untuk memberikan arus listrik pada ponsel (Tim KompuLab : 2008: 15-20).

II.3. Unified Modeling Language (UML)

UML (*Unified Modeling Language*) adalah suatu alat Bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek (Munawar ; 2005 : 17). Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang

memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

Meskipun UML sudah banyak menyediakan diagram yang bisa membantu mendefinisikan suatu aplikasi, tidak berarti bahwa semua diagram tersebut akan bisa menjawab persoalan yang ada. Adapun tipe diagram UML yang ada seperti pada Tabel II.1.

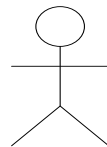
Tabel II.1 Tipe Diagram UML

Diagram	Tujuan	Keterangan
Activity	Prilaku prosedural dan paralel	Sudah ada di UML 1
Class	Class, fitur dan relasinya	Sudah ada di UML 1
Communication	Interaksi diantara objek. Lebih menekankan kepada link	Di UML 1 disebut collaboration
Component	Struktur dan koneksi dari komponen	Sudah ada di UML 1
Composite Structure	Dekomposisi sebuah class saat runtime	Baru untuk UML 2
Deployment	Penyebaran/instalasi ke klien	Sudah ada di UML 1
Interaction Overview	Gabungan dari activity dan sequence diagram	Baru untuk UML 1
Object	Contoh konfigurasi instance	Tidak resmi ada di UML 1
Package	Struktur hierarki saat kompilasi	Tidak resmi ada di UML 1
Sequence	Interaksi antara objek. Lebih menekankan pada urutan.	Sudah ada di UML 1
State Machine	Bagaimana event mengubah sebuah objek	Sudah ada di UML 1
Timing	Interaksi antar objek. Lebih menekankan pada waktu	Sudah ada di UML 1
Use Case	Bagaimana user berinteraksi dengan sebuah sistem	Sudah ada di UML 1

Sumber : Munawar (2009 : 23)

II.3.1. Actor

Actor adalah *abstraction* dari orang dan *system* yang lain yang mengaktifkan fungsi dari target *system*. Orang atau *system* bisa muncul dalam beberapa peran. Perlu dicatat bahwa *actor* berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. Berikut notasi *actor* dalam UML:



Gambar II.2 : Notasi Actor pada UML

Sumber : Munawar (2009 : 64)

II.3.2. Class

Class, dalam notasi UML digambarkan dengan kotak. Nama *class* menggunakan huruf besar diawal kalimatnya dan diletakkan diatas kotak. Bila *class* mempunyai nama yang terdiri dari 2 suku kata atau lebih, maka semua suku kata digabungkan tanpa spasi dengan huruf awal tiap suku kata menggunakan huruf besar. Berikut notasi *class* dalam UML:



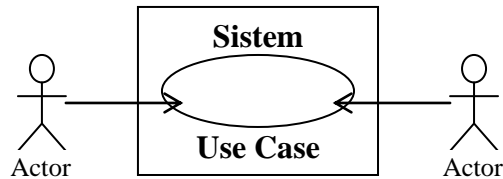
Gambar II.3 : Notasi Class di UML

Sumber : Munawar (2009 : 35)

II.3.3. Use Case

Use Case adalah alat bantu terbaik guna menstimulasi pengguna potensial untuk mengatakan tentang suatu *system* dari sudut pandangnya. Tidak selalu mudah bagi pengguna untuk menyatakan bagaimana mereka bermaksud

menggunakan sebuah *system*. Karena *system* pengembangan tradisional sering ceroboh dalam melakukan analisis, akibatnya pengguna seringkali susah menjawabnya tatkala dimintai masukan tentang sesuatu. Notasi *use case* dapat dilihat pada gambar II.3 :



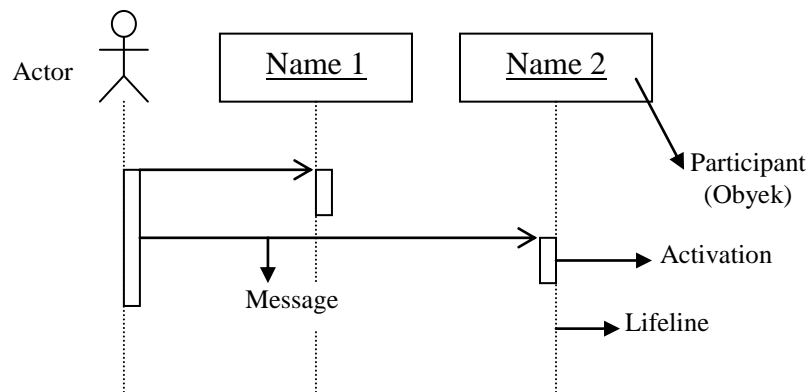
Gambar II.4 : Notasi Use Case pada UML

Sumber : Munawar (2009 : 64)

II.3.4. Sequence Diagram

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini dalam *use case*.

Komponen utama *sequence diagram* terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. *Message* dengan tanda panah dan waktu yang ditunjukkan dengan progress vertical. Berikut Contoh *sequence diagram* :



Gambar II.5 : Simbol-simbol yang ada pada Sequence Diagram







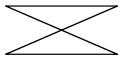



Sumber : Munawar (2009 : 89)

II.3.5. Activity Diagram

Activity diagram adalah teknik untuk mendiskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity Diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa.

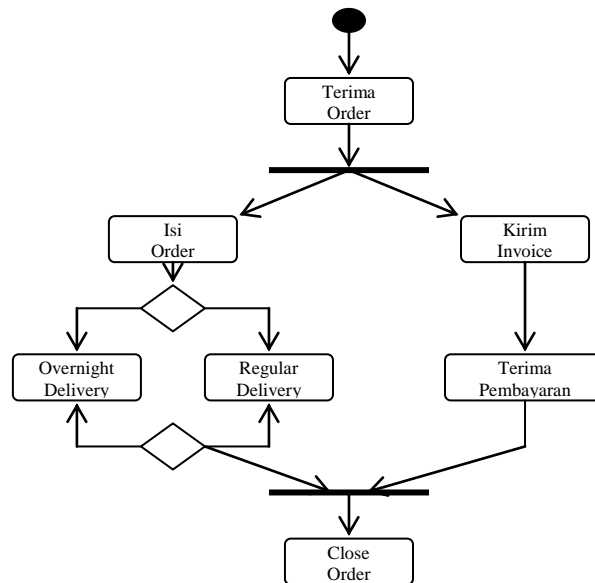
Berikut adalah simbol-simbol yang sering digunakan pada saat pembuatan *activity diagram*.

Tabel II.2 Simbol-simbol yang sering dipakai pada *Activity Diagram*

Simbol	Keterangan
	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)

Sumber : Munawar (2009 : 109)

Adapun contoh dari *Activity Diagram* dapat di lihat pada Gambar II.6.



Gambar II.6 : Contoh *Activity Diagram* Sederhana

Sumber : Munawar (2009 : 111)

II.4. Pengertian Database

Database merupakan komponen terpenting dalam pembangunan SI, karena menjadi tempat untuk menampung dan mengorganisasikan seluruh data yang ada dalam sistem, sehingga dapat dieksplorasi untuk menyusun-menyusun informasi-informasi dalam berbagai bentuk. *Database* merupakan himpunan kelompok data yang saling berkaitan. Data tersebut diorganisasikan sedemikian rupa agar tidak terjadi duplikasi yang tidak perlu, sehingga dapat diolah atau dieksplorasi secara cepat dan mudah untuk menghasilkan informasi (Budi Sutedjo Dharma Oetomo; 2006: 99).

II.4.1. Hierarki Data Dalam Database

Data dalam sebuah *database* disusun berdasarkan sistem hierarki yang unik, yaitu:

1. **Database**, merupakan kumpulan file yang saling terkait satu sama lain, misalnya file data induk karyawan, file jabatan file penggajian dan lain sebagainya. Kumpulan file yang tidak saling terkait satu sama lain tidak dapat disebut *database*, misalnya file data induk karyawan, file tamu undangan perkawinan, file barang retail pasar swalayan.
2. **File**, yaitu kumpulan dari *record* yang saling terkait dan memiliki format *field* yang sama dan sejenis.
3. **Record**, yaitu kumpulan *field* yang menggambarkan suatu unit data individu tertentu.
4. **Field**, yaitu atribut dari *record* yang menunjukkan suatu item dari data, seperti nama, alamat, dan lain sebagainya.
5. **Byte**, yaitu atribut dari *field* yang berupa huruf yang membentuk nilai dari sebuah *field*. Huruf tersebut dapat berupa numerik maupun abjad atau karakter khusus
6. **Bit**, yaitu bagian terkecil dari data secara keseluruhan, yaitu berupa karakter ASCII nol atau satu yang merupakan komponen pembentuk *byte* (Budi Sutedjo Dharma Oetomo, S.kom., MM : 2006: 102).

II.4.2. MySQL (*phpMyAdmin*)

PhpMyAdmin adalah aplikasi berbasis *web* yang dibuat dari pemrograman PHP dan *Java Script*. *PhpMyAdmin* juga dapat disebut sebagai *tools* yang berguna untuk mengakses *database MySQL Server* dalam bentuk tampilan *web*. dengan adanya *phpMyAdmin*, semua pekerjaan menjadi mudah, karena tanpa

harus mengerti perintah-perintah dasar SQL namun sudah dapat memanajemen *database* dan data yang ada didalamnya (Bunafit Nugroho: 2009: 13).

II.5. Bahasa Pemrograman Java.

Java merupakan pemrograman berorientasi objek dan bebas *platform*, dikembangkan oleh SUN Micro System dengan sejumlah keunggulan yang memungkinkan Java dijadikan sebagai bahasa pengembangan *enterprise* (Rijalul Fikri, dkk; 2005: 15).

Java lahir karena ketidakpuasan seorang insinyur di SUN Micro System bernama James Gosling. Ia tidak puas dengan kompiler C++ (yang ia gunakan untuk membuat *software* yang di-embed pada peralatan elektronik) karena dinilai terlalu banyak menghasilkan *bug*, berbiaya besar, sangat bergantung terhadap *platform*. Gosling merasa perlu membuat kompiler baru sebagai solusi terhadap sejumlah kelemahan pada C++ tersebut.

Kompiler baru tersebut diberi nama dengan Oak. Kompiler ini mirip dengan C++ tetapi dengan sejumlah pengurangan fitur yang dianggap kurang menguntungkan dalam pengembangan, seperti *multiple inheritance*, konversi tipe secara otomatis, penggunaan pointer dan manajemen memori.

Pada tahun 1994, Oak diubah namanya menjadi Java. Pada era ini, java divisikan sebagai bahasa yang memiliki dukungan baik terhadap *web* (Rijalul Fikri, dkk; 2005: 19)