MAKALAH

PENERAPAN KONSEP KECERDASAN BUATAN MELALUI PEMROGRAMAN KOMPUTER

MILIK PERPUSTAKAAN IKIP PADANG
13 - 8 - 99
SUMBER / HARGA : H /
KOLEKSI : K1
NO. INVENTARIS : 558 / H / 99 p (1)
KLASIFIKASI : 003.5 Gan p

Oleh

Drs. Ganefri, M.Pd

Jurusan Pendidikan Teknik Elektro

FAKULTAS PENDIDIKAN TEKNOLOGI DAN KEJURUAN

IKIP PADANG 1999

DAFTAR ISI

Halaman

A. Pendahuluan 1

B. Permasalahan. 2

C. Sistem Pakar 4

D. Bahasa Pemrograman Untuk Aplikasi Kecerdasan Buatan . . . . . .

1. Bahasa Konvensional 7

2. Prolog 8

3. Lisp 10

E. Operasi Dasar Sistem Pakar 12

F. Menara Hanoi. 14

G. Simpulan 20

PENERAPAN KONSEP KECERDASAN BUATAN MELALUI PEMROGRAMAN KOMPUTER

==========

Oleh : Drs. Ganefri, M.Pd

A. Pendahuluan

Konsep komputer yang berfikir mengisyairatkan bahwa komputer mengeksekusikan program yang cerdas. Tetapi terdapat perdebatan mengenai apakah ada program yang cerdas, dengan kata lain dapatkah komputer berfikir? Perdebatan ini tidak mudah diatasi sebab menyangkut tentang penafsiran definisi kecerdasan. Pertanyaan yang muncul dari perdebatan ini adalah apa yang membedakan program yang cerdas dengan program yang "tidak cerdas"? Oxford advanced learner's Dictionary mendefenisikan kata kecerdasan (*intelligence*) sebagai "**power of learning, understanding and reasoning**". Dalam kontek ini reasoning berarti berfikir. Sampai sekarang orang sukar menjelaskan bagaimana mereka berfikir, mereka hanya dapat mengatakan apa yang mereka fikir! Kenyataannnya orang tidak memahami bagaimana mereka berfikir (jika dapat, mereka tentu tidak menghabiskan banyak pekerjaan untuk membuat komputer berfikir!).

Anjing yang terlatih dikatakan cerdik jika ia dapat mengambil surat kabar dari halaman rumah dan memberikannya kepada tuannya yang sedang duduk diberanda. Demikian juga anak berusia satu tahun yang melakukan pekerjaan seperti diatas. Tidak terlalu sulit membuat robot yang dikontrol komputer yang mampu melakukan pekerjaaan seperti diatas. Tetapi orang tidak akan mengatakan

bahwa robot tersebut cerdas karna dapat membawa surat kabar! alasannya adalah robot yang membawa surat kabar tersebut merupakan mesin yang menjalankan program yang dibuat oleh pemrogram, dan robot tidak memikirkannya (sebagian orang mengatakan bahwa pemrogram robotlah yang berfikir).

Contoh robot dan surat kabar menimbulkan masalah yang seperti disebutkan tadi yaitu bahwa orang tidak mengetahui bagaimana berfikir. Karena program mengambil surat kabar dari halaman adalah mudah dibuat, lalu timbl kecenderungan untuk mengatakan bahwa program tidak cerdas sebab pekerjaaan tersebut mudah dikerjakan. Karena mereka tidak memahami proses berfiir, mereka menganggap secara salah bahwa alat yang dibuat oleh manusia tidak dapat cerdas sebab manusia betul-betul memahaminya. Pada dasarnya mereka percaya bahwa yang dicipta selalu lebih rendah daripada penciptanya!.

B. Permasalahan

Berfikir selalu dihubungkan dengan konsep kehendak bebas (*free will*), yaitu hanya makhluk dengan kehendak untuk berfikirlah yang dapat berfikir. Descartes menulis pernyataan filosofis "**Cogito, ergo sum**", artinya "aku berfikir, maka aku ada, yang menyatakan eksistensi manusia itu ditentukan oleh kemampuannya berfikir. Pada contoh robot dan surat kabar, tampak bahwa anjing dan anak kecil memilih untuk mengambil surat kabar, ia tidak melakukan yang lainnya. Dapatkah komputer memilih untuk melakukan sesuatu? ini adalah salah satu masalah filosofis di abad dua puluh satu. Beberapa pemrogram meyakini bahwa "mesin adalah mesin"; komputer tidak mempunyai kehendak bebas sebab ia tidak mempunyai akal, ia hanya merupakan rangkaian elektronik. Maka komputer

tidak mungkin memilih melakukan sesuatu. Ini adalah alasan yang menarik. Tetapi, pemrogram yang lain mempunyai pandangan yang berlawanan. Kelomok kedua ini memberikan contoh berikut. Suatu sistem komputer telah diprogram untuk mengontrol berat muatan yang diangkut oleh truk. Jika sebuah truk mengangkut suatu muatan yang melebihi kapasitas yang telah ditentukan maka komputer akan memberikan suatu sinyal tertentu yang ditujukan kepada pengemudi bahwa muatan yang diangkut melebihi batas. Dalam hal ini komputer bertindak dalam menentukan suatu masalah. Apakah komputer dapat berfikir? sebagaimana telah dijelaskan diatas bahwa terdapat dua pandangan yang saling berlawanan mengenai masalah ini. Sampai sekarang perdebatan masih berlanjut. Banyak orang yakin bahwa tidak mungkin menentukan apakah komputer dapat berfikir ataukah tidak. Tetapi, beberapa contoh memperlihatkan secara jelas bahwa seseorang dapat membuat komputer yang bertingkah laku seperti manusia.

Beberapa program memperlihatkan kecerdasannya. Kesulitannya disini adalah berhubungan dengan ketidakcermatan tentang defenisi kecerdasan. Definisi kamus memperlihatkan bahwa istilah kecerdasan menyiratkan kecerdasan manusia. Implikasi ini menyulitkan untuk mengakui adanya kemungkinan bahwa mesin dapat berfikir. Sebaliknya, jika implikasi in dihilangkan, mudah mengatakan bahwa mungkin terdapat program yang cerdas. Jika perbedaann ini dipahami, definisi program yang cerdas menjadi jelas. Program yang cerdas adalah program yang menunjukkan kelakuan yang sama seperti manusia jika dihadapkan dengan problem yang sejenis. Ini bukan berarti program tersebut dapat menyelesaikan semua masalah, seperti juga manusia. Bidang kecerdasan buatan (*Artificial*

Intelligence) terdiri dari banyak cabang yaitu pencarian dan penyelesaian masalah, sistem pakar, pengolahan bahasa alami, pengenalan pola, robotik, proses cara belajar mesin, logika dan ketidaktentuan. Karena keterbatasan ruang, pada makalah ini akan dibahas sistim pakar saja.

C. Sistim Pakar

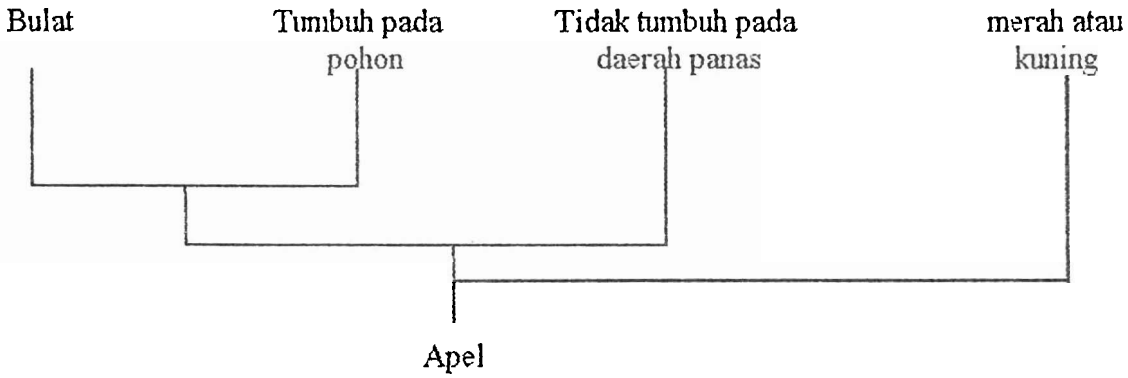
Sistim pakar (*expert system*) adalah program komputer yang dirancang untuk bertingkah laku seperti seorang ahli dalam suatu bidang khusus, misalnya kedokteran, fisika, psikologi dsb. Apa keuntungan sistim pakar? Tidak seperti pakar manusia yang harus tidur, makan, bersantai; sistim pakar dapat digunakan selama 24 jam per hari. Selain itu sistim pakar tidak pernah mati. Pengetahuan pada sistim pakar dapat disimpan sehingga kemungkinan kecil dapat hilang. Keuntungan lain dari sistim pakar adalah ia selalu pada kondisi optimum. Jika pakar manusia lelah, maka semangat kerjanya menurun. Tetapi sistim pakar akan bekerja sebaik-baiknya. Keuntungan terakhir sistim pakar adalah setelah terdapat satu komputer sistim pakar, Anda dapat menciptakan sistim pakar yang lain hanya dengan mengutipkan program dari satu mesin ke mesin lainnya. Manusia memerlukan waktu yang lama untuk menjadi ahli pada bidang tertentu. Ini berarti sulit memperoleh manusia pakar yang baru. Setiap sistim pakar mempunyai dua bagian; basis pengetahuan dan mesin inferensi. Basis pengetahuan adalah pangkalan data (data base) yang menyimpan informasi tertentu mengenai suatu masalah. Contoh basis pengetahuan sedehana adalah sebagai berikut.

Tabel 1. Contoh Basis Pengetahuan Sederhana

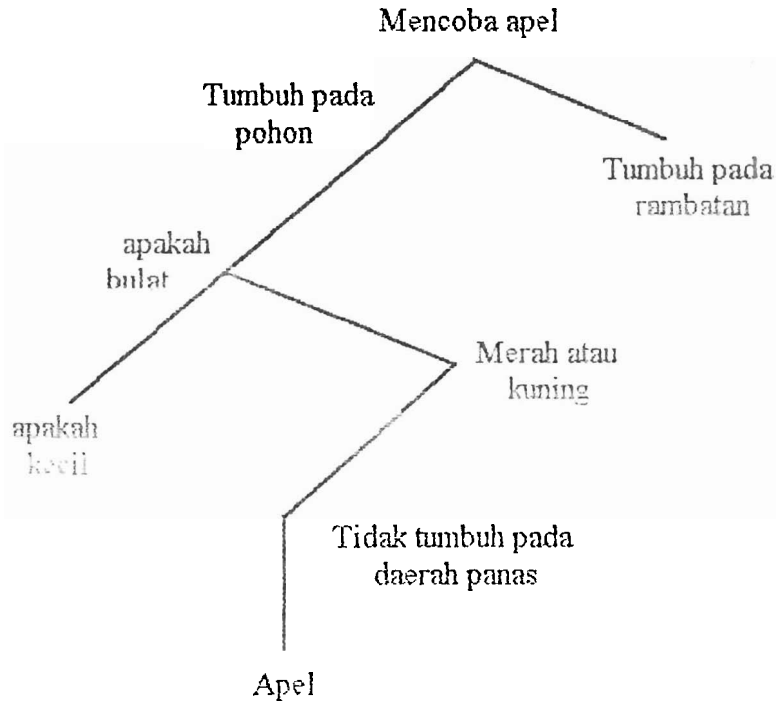
Objek	Atribut
Semangka	tumbuh merambat berukuran bulat besar tidak berduri Kulit berwarna hijau Isi berwarna merah atau kuning
Apel	tumbuh pada pohon bentuk bulat tidak tumbuh pada daerah panas berwarna merah atau hijau

Atribut dapat ditambah serinci mungkin. Semakin lengkap basis pengetahuan suatu objek berarti atributnya semakin rinci. Mesin inferensi adalah bagian sistim pakar yang mencocokkan objek dengan informasi atribut pada basis pengetahuan. Ada dua katagori mesin inferensi, yaitu deterministik dan probabilistik. Untuk memahami perbedaan dua katagori ini, anggaplah bahwa ada dua pakar, satu pakar kimia dan lainnya pakar ilmu sosial. Pakar kimia dapat mengatakan dengan pasti bahwa jika suatu atom mempunyai dua elektron, maka atom itu adalah helium. Tidak ada perselisihan mengenai identitas atom sebab banyaknya elektron menentukan menentukan tipe atom. Tetapi jika Anda bertanya kepada Pakar ilmu sosial tentang bagaimana cara terbaik untuk menaggulangi kenakalan remaja, pakar tersebut akan memberi jawaban yang diklasifikasikan sebagai kemungkinan atau dengan kata lain jawabannya tidak pasti. Pada tulisan ini akan dibahas hanya sistim pakar deterministik sebab logikanya jelas. Ada dua cara untuk membangun mesin inferensi yaitu metoda maju dan metoda mundur. Metoda maju memulai dengan informasi lalu mencoba menemukan objek yang sesuai dengan informasi.

Diagram yang mengilustrasikan metoda metoda maju diperlihatkan di Gb. 1. Mesin inferensi akan memikirkan objek apel ketika diberi atribut yang benar. Metoda ini pada dasarnya membangun pohon dari daun-daun menuju akar. Metoda mundur mulai dengan hipotesis (suatu objek) dan memerlukan informasi untuk menolak maupun menerimanya. Metoda ini diperlihatkan di Gb.2. Ini adalah proses yang berlawanan dengan metoda maju.



Gambar 1. Metoda maju untuk objek apel



Gambar 2. Metoda mundur untuk objek apel

D. Bahasa Pemrograman untuk Aplikasi Kecerdasan Buatan

Pembuatan suatu sistim kecerdasan buatan memerlukan bahasa pemrograman. Bahasa-bahasa apa saja yang dapat dipakai? Ada dua pilihan untuk menggunakan bahasa pemrograman. Pertama, dapat digunakan bahasa konvensional seperti BASIC, C, Pascal maupun Fortran. Kedua, dapat digunakan bahasa-bahasa yang dirancang khusus untuk mengembangkan aplikasi kecerdasan buatan seperti LISP dan PROLOG.

1. Bahasa-bahasa Konvensional

Bahasa-bahasa konvensional yang dipakai untuk mengembangkan aplikasi kecerdasan buatan harus merupakan bahasa serba guna atau dapat juga bahasa aplikasi teknik. Bahasa-bahasa yang memenuhi syarat ini misalnya BASIC, C, Pascal dan Fortran. Sedangkan bahasa yang tidak dapat digunakan untuk mengembangkan kecerdasan buatan adalah bahasa yang ditujukan untuk aplikasi bisnis seperti COBOL.

Semua teknik kecerdasan buatan dapat diimplementasikan dengan menggunakan bahasa konvensional. Salah satu bahasa konvensional yang terbaik untuk membuat aplikasi kecerdasan buatan adalah bahasa C. Bahasa ini merupakan bahasa serba guna yang mempunyai kemampuan tinggi. Bahkan, dapat dikatakan bahwa beberapa rutin khusus dalam program aplikasi kecerdasan buatan yang dibuat dengan C sebenarnya lebih jelas daripada bahasa kecerdasan buatan. Tetapi, bahasa konvensional memiliki satu kekurangan besar, yaitu pemrogram harus menjelaskan secara rinci bagaimana

suatu program harus bekerja dan bagaimana cara memperoleh hasil yang maksimal. Bahasa pemrograman konvensional bersifat prosedural, berbicara dengan kalimat perintah; "Lakukan ini, lalu ini, selanjutnya ini". Dengan kata lain, pemrogram memberitahu komputer bagaimana melakukan sesuatu. Hal tersulit disini adalah pembuatan spesifikasi program yang rinci dan perancangan logika program. Sedangkan bahasa kecerdasan buatan bersifat deklaratif. Dalam bahasa kecerdasan buatan, pemrogram memberi tahu komputer apa yang harus dilakukan, dengan memberikan fakta dan aturan tertentu.

Dengan melihat kenyataan di atas, bahasa pemrograman konvensional tidak seratus persen cocok untuk mengembangkan apikasi kecerdasan buatan. Program-program cerdas tidak hanya menerima kalimat-kalimat perintah, tetapi lebih banyak melihat kenyataan-kenyataan yang ada, lalu menarik kesimpulan sendiri. Hal ini lebih mirip dengan cara berfikir manusia. Berikut ini dibahas dua bahasa yang dibuat khusus utuk kecerdasan buatan, yaitu PROLOG dan LISP.

2. PROLOG

Bahasa PROLOG (Programming in Logic) diciptakan oleh Alain Colmeraurer pada tahun 1972 di Universitas Marsilles, Perancis. Meskipun para pemrogram Amerika berpendapat bahwa PROLOG kurang baik dibandingkan LISP, bahasa ini banyak digunakan di Eropa. Bahkan pada tahun1981, Jepang mengumumkan penggunaan PROLOG sebagai dasar bahasa generasi kelima yang akan digunakan oleh superkomputer yang sedang dikembangkan di Jepang.

Bahasa PROLOG merupakan bahasa deklaratif dan deskriptif. Pemrogram menggambarkan rincian persoalan yang diselesaikan dan PROLOG menggunakan prosedur-prosedur yang tersedia untuk menyelesaikan persoalan tersebut. Prosedur-prosedur PROLOG didasarkan pada teknik logika simbolik yang menggunakan silogisme untuk membuktikan suatu proposisi. Sebagai contoh :

Semua profesor pandai ⟶ Semua A memiliki sifat B

Agus seorang profesor ⟶ C anggota A

Karena itu Agus pandai ⟶ C memiliki sifat B

Struktur dasar program PROLOG terdiri dari **Domains**, **Prdicates**, **Goal** dan **Clauses**. Dalam bagian **Domains**, pemrogram mendefinisikan jenis data untuk nama-nama yang digunakan dalam program. Bagian **Predicates** merupakan tempat pendeklarasian klausa-klausa yang terdapat dalam program. Bagian **Goal** memberi tahu PROLOG tujuan program yang diinginkan. Jika **Goal** tidak dituliskan dalam program, PROLOG akan menanyakannya kepada pemakai. Otak program PROLOG adalah bagian **Clauses**, yaitu kumpulan fakta dan aturan. Fakta mewakili hubungan logis antara objek dan atribut-atributnya, sedangkan aturan membentuk kesimpulan atau fakta baru.

Aturan dalam program PROLOG mempunyai bentuk umum x:=-y atau x if y, yaitu menyatakan x sebagai fakta yang benar jika diketahui y adalah benar.

PROLOG memecahkan persoalan dengan mencocokkan data yang diberikan dengan data yang disimpan dalam databasenya. Contoh berikut

menyatakan aturan bahwa semua profesor pandai, lalu menyatakan fakta bahwa Agus seorang profesor.

Predicates

profesor(simbol)

pandai(simbol)

Clauses

pandai(seseorang) : = profesor(seseorang)

profesor(agus).

PROLOG akan menanyakan **Goal** dari program tersebut. Misalkan ditanyakan,"Apakah Agus pandai?" seperti dibawah ini

Goal:pandai(agus)

PROLOG memeriksa aturan yang diberikan dan menemukan bahwa setiap profesor pandai. Maka PROLOG menjawab:

Yes

Bahasa PROLOG agak sukar dipahami oleh mereka yang biasa menggunakan bahasa-bahasa konvensional yang terstruktur.

3. LISP

Bahasa LISP (List Processing) dikembangkan oleh McCarthy pada pertengahan tahun lima puluhan yang kemudian dikembangkan oleh Massachusetts Institute of Technology (MIT). Bahasa ini merupakan bahasa kecerdasan buatan yang paling populer di Amerika Serikat.

LISP mengenal dua macam struktur data, disebut *atom* dan *list*. Atom adalah data yang tidak dapat dibagi lebih lanjut. Sedangkan list adalah daftar yang disusun dari atom-atom atau list-list lainya. LISP bekerja dengan pengolahan list, dan inilah asal nama bahasa tersebut. List dipisahkan satu sama lain oleh tanda kurung, dan menyebabkan banyak sekali tanda kurung terdapat dalam program.

Fungsi-fungsi yang telah di sediakan oleh LISP disebut *primitive*. Salah satu contoh primitive adalah perintah **setq** yang berguna untuk memberikan nilai kepada suatu atom. Contoh sebagai berikut;

(setq hewan (kucing anjing kuda sapi))

Yang artinya sama dengan penegasan ; = pada pascal.

hewan : = 'kucing anjing kuda sapi';

Semua program dalam bahasa LISP adalah ekspresi, yaitu kesatuan pengolahan atom dan list. Program LISP disusun dari pemanggilan fungsi dituliskan sebagai list dengan atom pertama sebagai nama fungsi dan sisa isi list merupakan data yang akan diolah. Dta ini disebut argumen dan dapat berupa list maupun atom. Sebagai contoh, pernyataan di bawah ini mendifinisikan fungsi rata-rata untuk mencari nilai rata-rata dari tiga variabel a,b dan c;

(defun rata-rata (a b c) (/ (+ a b c) 3))

Karena prosedur dan data mempunyai bentuk yang sama dalam LISP, sesuatu yang berupa prosedur dalam suatu program dapat berupa data dalam

program lainya. Program dapat beroperasi pada data, program lain dan program itu sendiri. Ini dapat merupakan hal yang membingungkan bagi pemula.

E. Operasi Dasar Sistim Pakar

Basis pengetahuan mengandung objek beserta atributnya. Untuk memahami kerja mesin inferensi, digunakan basis pengetahuan sebagai berikut;

Objek	Atribut
1	A, B, C
2	A, B, Y
3	B, X
4	A, B, D

Pada tingkat paling sederhana, mesin inferensi mulai dengan menganggap bahwa objek 1 adalah tujuan dan mencoba mengkonfirmasikan dengan menanyakan apakah tujuan mempunyai atribut objek 1. Jika ya, lalu mesin inferensi melaporkan bahwa objek 1 adalah jawabannya. Jika tidak, lalu mesin inferensi melanjutkan ke objek 2 dan menanyakan atribut objek 2. Proses ini berulang sampai mesin inferensi menemukan objek yang benar ataukah tidak ada objek yang sesuai. Jika objek 4 adala tujuannya, dialog akan berlangsung sebagai berikut.

Pakar : Apakah ia mempunyai (atribut) A?
Pemakai : Ya
Pakar : Apakah ia mempunyai B?
Pemakai : Ya
Pakar : Apakah ia mempunyai C?
Pemakai : Tidak /*menolak objek 1*/

MILIK UPT PERPUSTAKAAN
IKIP PADANG

Pakar : Apakah ia mempunyai A? /*pemborosan*/
Pemakai : Ya
Pakar : Apakah ia mempunyai B? /*pemborosan*/
Pemakai : Ya
Pakar : Apakah ia mempunyai Y?
Pemakai : Tidak /*menolak objek 2*/
Pakar : Apakah ia mempunyai B? /*pemborosan*/
Pemakai : Ya
Pakar : Apakah ia mempunyai X? /*tidak perlu*/
Pemakai : Tidak /*menolak objek 3*/
Pakar : Apakah ia mempunyai A? /*pemborosan*/
Pemakai : Ya
Pakar : Apakah ia mempunyai B? /*pemborosan*/
Pemakai : Ya
Pakar : Apakah ia mempunyai D? /*ditemukan*/
Pemakai : Ya
Pakar : Ia adalah objek 4

Sistim pakar yang bekerja seperti ini tidak hanya membosankan, tetapi juga bekerja sangat lama jika terdapat objek yang banyak. Sistim seperti ini menunjukkan dua kelemahan. Pertama, ia memerlukan atribut yang sama beberapa kali. Kedua, ketika ia menuju objek 3, ia menanyakan pertanyaaan yang tidak perlu. Sistim sebaiknya telah mengetahui dari pertanyaan sebelumnya bahwa objek yang dicari mempunyai atribut A dan tidak perlu ditanyakan lagi. Jadi apa yang diperlukan oleh mesin inferensi adalah seperti dialog berikut ini.

Pakar : Apakah ia mempunyai A?
Pemakai : Ya
Pakar : Apakah ia mempunyai B?
Pemakai : Ya

Pakar : Apakah ia mempunyai C?

Pemakai : Tidak /*menolak objek 1*/

Pakar : Apakah ia mempunyai Y?

Pemakai : Tidak /*menolak obek 2 dan objek 3*/

Pakar : Apakah ia mempunyai D? /*ditemukan*/

Pemakai : Ya

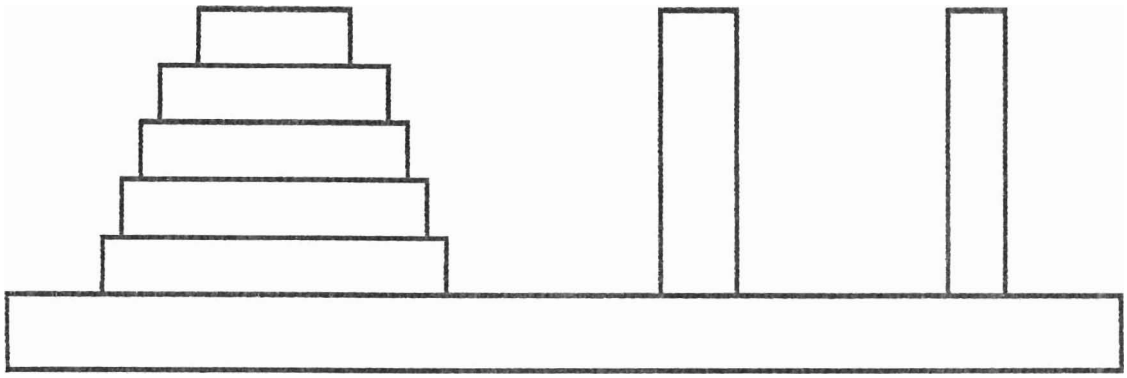
Pakar : Ia adalah objek 4

Tipe mesin inferensi semacam ini harus mempunyai sifat-sifat sebagai berikut :

1. Sistim pakar tidak memerlukan atribut yang sama dua kali,
2. Sistim pakar sebaiknya menolak dengan segera dan melewati objek yang atributnya telah disangkal.

F. Menara Hanoi

Contoh program berikut memperlihatkan bagaimana komputer dapat melakukan pekerjaan yang tampaknya sederhana, tetapi sebenarnya cukup rumit. Problem berkenaan dengan permainan anak-anak yang cukup terkenal yaitu Menara Hanoi. Permainan ini mempunyai tiga as dan beberapa piringan yang berbeda-beda ukurannya. Tiap piringan mempunyai lubang ditengah-tengahnya, sehingga piringan dapat ditumpuk pada as. mula-mula piringan ditumpuk pada as paling kiri dengan urutan sesuai dengan ukurannya, paling besar dibawah dan paling kecil di puncak (lihat Gambar 3)



Gambar 3. Menara hanoi

Tujuan permainan ini adalah memindahkan semua piringan di as kiri ke as kanan, tanpa pernah meletakkan piringan yang lebih besar di atas piringan yang lebih kecil. Hanya satu piringan boleh dipindahkan pada suatu saat dan setiap piringan harus selalu ditempatkan pada salah satu as.

Strateginya adalah menganggap salah satu as sebagai asal dan satu as lainnya sebagai tujuan. As ketiga digunakan sebagai penyimpan sementara. Jika mula-mula terdapat n piringan di tumpuk pada as kiri, problem memindahkan semua n piringan ke as kanan dapat dinyatakan sebagai berikut.

1. Pindahkan n-1 piringan dari as kiri ke as tengah dengan menggunakan as kanan sebagai penyimpanan sementara.
2. Pindahkan piringan sisanya ke as kanan.
3. Pindahkan n-1 piringan di as tengah ke as kanan dengan menggunakan as kiri sebagai penyimpanan sementara.

ILIK UPT PER
IKIP PADA

Program menara Hanoi ini ditulis dengan menggunakan Turbo BASIC. Hampir pada setiap akhir instruksi terdapat penjelasan, sehingga disini tidak diperlukan lagi penjelasan.

```
$STACK 32767
%X = 1
%Y = 0
%Prompt = 24
%PiringanMak = 13 ' menunjukkan banyaknya piringan maksimum
%CursorOff=0
'deklarasi variabel global
DIM BanyakPiringan %(1:%PiringanMak + 1),MenaraAsal%(1:%PiringanMak + 1)
DIM MenaraTujuan %(1:%PiringanMak + 1),Piringan$(1:%PiringanMak + 1)
DIM PosisiPiringan %(%PiringanMak,1),TinggiMenara %(1:3)
BanyakGerakan %=0
LajurBawah %=24
DasarMenara=2
SUB Init'prosedur ini untuk menginisiasi layar dan banyaknya 'piringan yang digunakan
SHARED BanyakPiringan%(),DasarMenara%
LOCAL C%
KEY OFF
COLOR 7,0  'menginisiasi warna layar
CLS
```

003.5
Jan
p.1


```
COLOR 4,0
LOCATE 1,26,%CursorOff
PRINT "MENARA HANOI"
COLOR 6,0
LOCATE %Prompt, %X, %CursorOff
PRINT "Banyak Piringan (1 TO "+STR$(%PiringanMak)+ ") ";
DO 'memperoleh banyak piringan dari pemakai
   LOCATE %Prompt,LEN("Banyak Piringan (1 TO"+STR%(%PiringanMak)+")")
   INPUT BanyakPiringan%(1)
   IF BanyakPiringan%(1)>%PiringanMak THEN BEEP
   LOOP UNTIL BanyakPiringan%(1)< = %PiringanMak
   DasarMenara% = DasarMenara% + BanyakPiringan%(!)
   COLOR 7,0
   LOCATE %Prompt, %X< %CursorOff
   PRINT SPACE$(79)
END SUB 'akhir prosedur inisiasi
SUB TampilanLayarPermainan 'prosedur ini menampilkan pesan pada layar
  SHARED BanyakPiringan %(),DasarMenara%
  LOCATE 1,26,%CursorOff
  COLOR 4,0
  PRINT "MENARA HANOI UNTUK "BanyakPiringan%(1)"PIRINGAN"
  LOCATE DasarMenara%+1, %X, %CursorOff 'posisi cursor
  COLOR 1,0 'Menetapkan warna tampilan
```

```
  PRINT STRING%(80, 176); 'menampilkan balok pada layar
  COLOR 7,0
END SUB 'akhir prosedur TampilanLayarPermainan
SUB BuatGerakan(JumlahGerakan %)
  SHARED Pengulangan %,BanyakPiringan %(),MenaraAsal %()
  SHARED MenaraTujuan %(),TinggiMenara %()
  INCR Pengulangan %
  IF BanyakPiringan %(Pengulangan %) =() THEN
    DECR Pengulangan %
    EXIT SUB
  END IF
  BanyakPiringan %(Pengulangan % + 1) = BanyakPiringan %(Pengulangan%)-1
  MenaraAsal %(Pengulangan % + 1) = MenaraAsal %(Pengulangan %)
  MenaraTujuan %(Pengulangan % + 1) = 6-
    MenaraTujaua %(Pengulangan %)-MenaraTujuan %(Pengulangan %)
  CALL BuatGerakan(JumlahGerakan %)
  INCR JumlahGerakan %
  CALL TampilanGerakan
  BanyakPiringan %(Pengulangan % + 1) = BanyakPiringan(Pengulangan %)-1
  MenaraAsal %(Pengulangan % + 1) = 6-
  MenaraAsal %(Pengulangan %)-MenaraTujuan %(Pengulangan%)
  MenaraTujuan %(Pengulangan % + 1) = MenaraTujuan %(Pengulangan %)
  CALL BuatGerakan(JumlahGerakan%)
```

```
  DECR Pengulangan %
END SUB 'akhir prosedur BuatGerakan
SUB TampilanGerakan
  SHARED Kolom %,Pengulangan %,BanyakPiringan %(),MenaraAsal %()
  SHARED TinggiMenara %(),PosisiPiringan %()
  SHARED MenaraTujuan %(),Piringan$()
  IF MenaraTujuan %(Penguangan %) = 1 THEN
    Kolom % = 1
  ELSEIF MenaraTujuan %(Pengulangan %) = 2 THEN
    Kolom % = 27
  ELSEIF MenaraTujuan %(Pengulangan) = 3 THEN
    Kolom = 54
  END IF
  LOCATE PosisiPiringan %(BanyakPiringan %(Pengulangan %), %Y),
    PosisiPiringan %(BanyakPiringan %(Pengulangan %), %X), %CursorOff
  COLOR 7,0
  PRINT SPACE % (26)
  INCR TinggiMenara %(MenaraTujuan %(Pengulangan %))<Kolom %,
  %CursorOFF
  COLOR BanyakPiringan % (Pengulangan %) MOD 14 + 1,0
  PRINT Piringan$(BanyakPiringan %(Pengulangan %));
  COLOR 7,0
PosisiPiringan % (BanyakPiringan % (Pengulangan %) , % Y ) =    ,
```

```
LOCATE 2,1,%CursorOFF:PRINT "Waktu akhir :"TIME$
LOCATE %Prompt,26,%CursorOff
PRINT "DISELESAIKAN DALAM" BanyakGerakan %"GERAKAN"
END 'akhir permainan
```

G. SIMPULAN

Komputer adalah peralatan teknologi yang mempunyai pengaruh sangat besar terhadap kehidupan manusa pada umumnya dan terhadap dunia pendidikan pada khususnya. Penggunaan komputer untuk pengajaran meliputi: latihan, tutorial, simulasi, permainan, demonstrasi, evaluasi, informasi dan kecerdasan buatan. Hal yang terakhir ini menyangkut konsep berfikir itu sendiri.

Dikaitkan dengan teori perkembangan kognitif dari Piaget yaitu dari sensori-motorik, preoperasional, berfikir kongret sampai berfikir formal, maka kecerdasan buatan dari komputer mirip dengan teori perkembangan Peaget tahap terakhir.

Bidang kecerdasan buatan terdiri dari banyak cabang yaitu pencairan dan penyelesaian masalah, sistim pakar, pengolahan bahasa alami, pengenalan pola, robotik, proses belajar mesin, logika dan ketidak pastian

Sistim pakar adalah program komputer yang dirancang untuk bertingkah laku seperti seorang ahli dalam suatu bidang khusus, misalnya kedokteran, fisika, psikologi. Keuntungan sistim pakar adalah ia dapat digunakan selama dua puluh empat jam perhari, sistim pakar tidak pernah mati dan selalu pada kondisi optimum.

Kecerdasan buatan dapat dirancang dengan bahasa pemrograman konvensional seperti BASIC, C, Pacsal, Fortran maupun menggunakan bahasa

MILIK UPT PERPUSTAK

khusus untuk kecerdasan buatan seperti PROLOG dan LISP. Pada makalah ini diberi contoh membuat simulasi menara Hanoi dengan menggunakan bahasa pemrograman BASIC.

Daftar Pustaka

Gottfried Byron S. 1986. *Programming with Pascal*. Singapore: Mc. Graw-Hill.

Schild, Herbert. 1987. *Artificial Intelligence Using C*. California: Osborne Mc. Graw-Hill.

Borland. 1987. *Turbo BASIC Owner's Handbook*. California: Borland International, Inc.

Rolstan. 1988. *Principles of Artificial Intelligence and Expert Systems Development*. New York: Mc. Graw-Hill.