



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

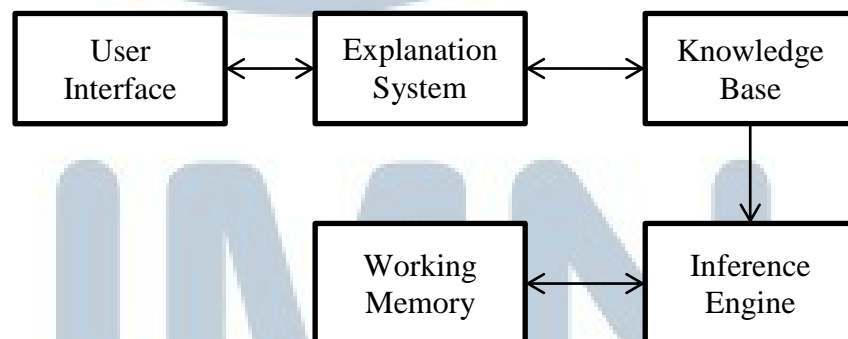
## BAB II

### LANDASAN TEORI

#### 2.1 Sistem Pakar

Sistem pakar merupakan sebuah program komputer yang dirancang untuk memodelkan kemampuan menyelesaikan masalah seperti layaknya seorang ahli. Seorang ahli adalah seorang individu yang memiliki kemampuan pemahaman yang superior dari suatu masalah, misalnya seorang dokter, penasehat keuangan, pakar mesin mobil, dan sebagainya (Giarratano & Riley: 1989).

Menurut Anjaneyulu (Anjaneyulu: 1998), sistem pakar memiliki lima komponen yang dapat dilihat pada Gambar 2.1 berikut.



Gambar 2.1 Komponen dalam Sistem Pakar

- *User Interface*

Merupakan bagian dari sistem pakar yang bertugas untuk berinteraksi dengan pengguna dengan cara menerima *input* dari pengguna dan menampilkan *output* kepada pengguna

- *Explanation System*

Merupakan bagian dari sistem pakar yang bertugas untuk menjelaskan atau menginterpretasikan pengetahuan dan memberikannya kepada pengguna melalui *user interface*.

- *Knowledge Base*

Merupakan bagian dari sistem pakar yang berisi domain pengetahuan yang dibutuhkan untuk memahami, merumuskan dan menyelesaikan masalah. *Knowledge base* memiliki dua elemen dasar, yaitu fakta yang menggambarkan situasi masalah yang terkait dan heuristik khusus atau *rules*, yang langsung menggunakan pengetahuan untuk menyelesaikan masalah khusus.

- *Inference Engine*

Merupakan bagian dari sistem pakar yang bertugas memproses dan mencocokkan fakta-fakta yang ada pada *working memory* dengan domain pengetahuan yang terdapat pada *knowledge base*, untuk menarik kesimpulan dari masalah yang dihadapi. *Inference engine* ini memodelkan proses berpikir pada manusia.

- *Working Memory*

Merupakan bagian dari sistem pakar yang berisi fakta-fakta masalah yang ditemukan. *Working memory* berisi fakta-fakta tentang suatu masalah yang ditemukan dalam proses konsultasi.

Sebuah sistem pakar dibuat untuk melakukan pekerjaan layaknya seorang pakar dalam domain yang lebih spesifik. Dengan demikian, karakteristik yang paling penting dari sistem pakar yaitu kinerja yang baik.

Terdapat beberapa alasan mengapa sistem pakar sangat disarankan untuk menyelesaikan masalah (Giarratano & Riley: 1989). Alasan-alasan tersebut adalah sebagai berikut:

- Dapat menyediakan kepakaran setiap waktu dan diberbagai lokasi
- Secara otomatis mengerjakan tugas-tugas rutin yang membutuhkan seorang pakar.
- Sebagai pesiapan apabila seorang pakar akan pensiun atau pergi
- Lebih murah karena memakai seorang pakar membutuhkan biaya yang mahal
- Kepakaran dapat dipergunakan pada lingkungan yang tidak bersahabat sekalipun (*hostile environment*)

Proses mengembangkan sebuah sistem pakar juga memiliki keuntungan tidak langsung karena pengetahuan dari pakar harus disalurkan secara eksplisit ke dalam komputer. Karena pengetahuan tersebut disalurkan secara eksplisit, tidak secara implisit tersimpan dalam otak sang pakar, pengetahuan tersebut dapat diperiksa kebenaran, konsistensi, dan kelengkapannya sehingga meningkatkan kualitas dari pengetahuan.

Sebuah sistem pakar biasanya didesain dengan memiliki karakteristik sebagai berikut (Giarratano & Riley: 1989).

- Kinerja tinggi

Sistem harus mampu memberikan respon dengan level kompetensi yang sebanding atau lebih baik dari pakar. Dengan kata lain, kualitas saran atau jawaban dari sistem pakar harus sangat tinggi.

- Waktu respon yang cepat

Sistem harus mampu memberikan jawaban dalam waktu yang sebanding atau lebih baik dari seorang pakar. Sebuah sistem pakar yang memakan waktu seminggu untuk memberikan jawaban dibandingkan dengan seorang pakar yang mampu memberikan jawaban dalam waktu satu jam tidak akan terlalu berguna.

- Dapat dipercaya

Sistem harus dapat dipercaya dan tidak menimbulkan masalah seperti *crash*.

- Dapat dimengerti

Sistem harus dapat menjelaskan langkah-langkah yang dilakukan untuk mencapai jawaban kepada pengguna.

Sudah banyak pemanfaatan sistem pakar dalam kehidupan. Contohnya adalah sistem pakar pertama di dunia, dengan nama DENDRAL yang digunakan untuk membantu mengidentifikasi struktur molekular senyawa yang tidak dikenal. DENDRAL dikembangkan oleh Joshua Lederberg, Ed Feigenbaum, Bruce Buchanan, dan timnya yang berasal dari Stanford University. DENDRAL akan menganalisis sampel yang masuk, kemudian akan membandingkannya dengan “ahli kimia”

yang pengetahuannya sudah disimpan dalam *knowledge base*. Proses analisis, penyeleksian, dan pengujian akan menghasilkan struktur molekular yang dicari (Lederberg: 1987).

Pengetahuan dari sistem pakar dapat direpresentasikan dengan beberapa cara – aturan dan objek. Salah satu metode yang paling umum digunakan adalah dengan menggunakan aturan IF... THEN, misalnya:

IF lampu berwarna merah THEN berhenti

Jika pada kenyataannya terjadi bahwa lampu lalu lintas berwarna merah, maka aturan tersebut “lampu berwarna merah” terpenuhi dan akan dilakukan aksi “berhenti”.

## 2.2 Inferensi

Inferensi merupakan mekanisme berpikir dan pola penalaran yang digunakan sistem untuk menarik kesimpulan (Krishnamoorthy & Rajeev: 1996). Inferensi dilakukan dengan menganalisa masalah tertentu dan selanjutnya mencari jawaban atau kesimpulan yang terbaik. Pada umumnya, terdapat dua metode inferensi, yaitu *Forward Chaining* dan *Backward Chaining*.

- *Forward Chaining*

*Forward Chaining* merupakan metode inferensi yang melakukan penalaran dari suatu masalah pada solusinya. Jika klausa premis sesuai dengan situasi (bernilai TRUE), maka proses akan menyatakan konklusi. *Forward chaining* bersifat *data-driven* karena inferensi dimulai dengan informasi yang tersedia dan baru konklusi

diperoleh. Jika suatu aplikasi menghasilkan *tree* yang lebar dan tidak dalam, maka gunakan *forward chaining*. Contoh pengaplikasian dari *forward chaining* dapat dilihat sebagai berikut. Terdapat 10 buah peraturan yang tersimpan dalam *knowledge base*.

R1 : if A and B then C

R2 : if C then D

R3 : if A and E then F

R4 : if A then G

R5 : if F and G then D

R6 : if G and E then H

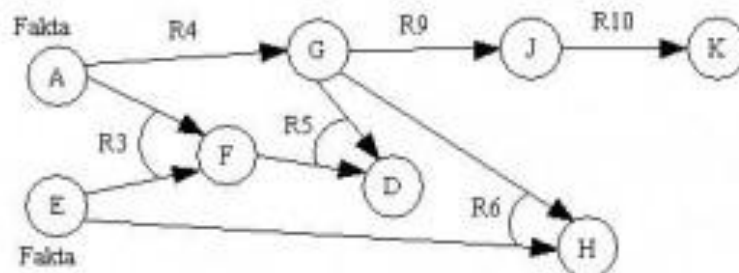
R7 : if C and H then I

R8 : if I and A then J

R9 : if G then J

R10 : if J then K

Fakta awal yang diberikan hanya A dan E, ingin membuktikan apakah K bernilai benar. Proses penalaran *forward chaining* terlihat pada gambar berikut.

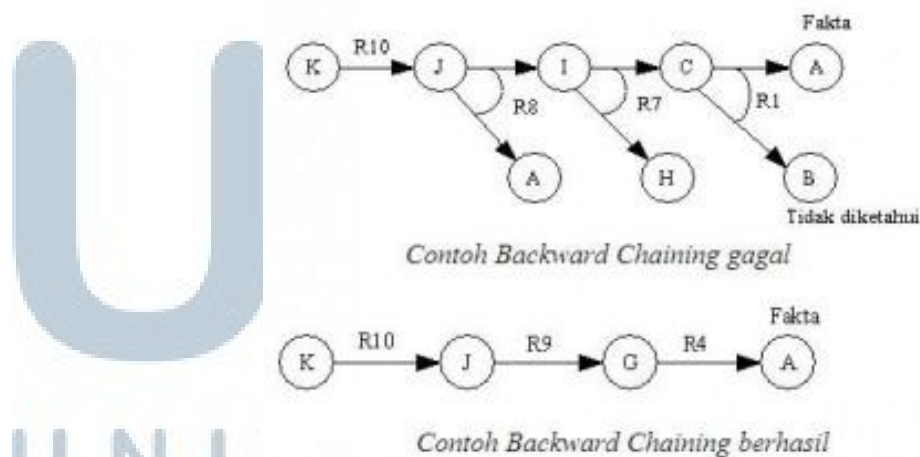


Gambar 2.2 Penalaran *forward chaining*

- *Backward Chaining*

Metode *Backward Chaining* menggunakan pendekatan *goal-driven* (dimulai dari harapan apa yang akan terjadi atau hipotesis, kemudian mencari bukti yang mendukung atau berlawanan dengan hipotesis). Metode ini sering memerlukan perumusan dan pengujian hipotesis sementara. Metode ini umumnya dipakai bila suatu aplikasi menghasilkan *tree* yang sempit namun cukup dalam.

Contoh pengaplikasian dari metode *backward chaining* ini mengikuti contoh *forward chaining*, terdapat 10 aturan yang sama pada basis pengetahuan dan fakta awal yang diberikan hanya A dan E. ingin membuktikan apakah K bernilai benar. Proses penalaran *backward chaining* terlihat pada gambar berikut.



Gambar 2.3 Penalaran *backward chaining*



### 2.3 Pola Makan

Menurut Lutfi Utami, salah satu cara hidup sehat adalah mengatur pola makan yang disesuaikan dengan kebutuhan tubuh. Dengan mengatur pola makan sesuai dengan kebutuhan tubuh, diharapkan energi yang diserap oleh tubuh manusia akan dikeluarkan tanpa ada kelebihan.

Energi dalam konteks nutrisi disebut kalori. Kalori merupakan hasil pembakaran karbohidrat, lemak, protein, serat, dan gula yang terjadi pada pernapasan. Pengaturan pola makan yang disesuaikan dengan kebutuhan tubuh harus memperhatikan kalori yang boleh dikonsumsi oleh seseorang. Beberapa hal yang memengaruhi kebutuhan kalori seseorang adalah usia, jenis kelamin, berat badan, tinggi badan, dan tingkat aktivitas.

Untuk mengetahui kebutuhan kalori seseorang, dapat dilakukan penghitungan berdasarkan *Basal Metabolic Rate (BMR)*. *Basal Metabolic Rate* merupakan jumlah energi yang dihabiskan manusia pada saat istirahat (<http://www.shapefit.com>).

Langkah pertama yang harus dilakukan adalah menghitung BMR. Rumus untuk menghitung BMR dapat dilihat pada Rumus 2.1.

$$\text{BMR pria} = 66 + (12,3 \times \text{BB}) + (5 \times \text{TB}) - (6,8 \times \text{U})$$

$$\text{BMR wanita} = 655 + (9,6 \times \text{BB}) + (1,8 \times \text{TB}) - (4,7 \times \text{U}) \dots\dots\dots(2.1)$$

di mana:

BB = Berat Badan (dalam kg)

TB = Tinggi Badan (dalam cm)

U = Usia (dalam tahun)

Langkah kedua adalah menentukan tingkat aktivitas. Aktivitas seseorang memiliki efek yang cukup besar dalam pengelolaan energi manusia. Jumlah energi yang digunakan untuk aktivitas akan berbeda-beda sesuai dengan intensitas dan jenis aktivitasnya (<http://www.topendsports.com>).

Salah satu cara untuk mengukur intensitas dari aktivitas adalah dengan Metabolic Equivalent Task (MET). Pemakaian energi oleh aktivitas didapat dari pengukuran jumlah oksigen yang dikonsumsi selama aktivitas untuk mendapatkan rata-rata oksigen yang digunakan per satuan waktu (<http://www.uihealthcare.com>). Nilai ini kemudian dibandingkan dengan jumlah oksigen yang dikonsumsi ketika istirahat. Tingkat aktivitas yang diukur berdasarkan MET dapat dibagi menjadi tiga tingkat, yaitu ringan, sedang, dan berat. Tingkat aktivitas beserta contoh aktivitasnya yang dapat dilihat pada Tabel 2.1.

Tabel 2.1 Tingkat Aktivitas beserta Nilai MET dan kegiatannya

Tingkat Aktivitas	MET/jam	Aktivitas
Ringan	1,3	Berdiri
	1,5	Membaca, berbicara
	1,8	Duduk, belajar, mencatat
	2	Berjalan pelan (kurang dari 1.5km per jam), bermain musik, menjaga toko, memasak, pekerjaan rumah tangga
Sedang	3	Menjalankan kendaraan berat (traktor, bus, truk), naik-turun tangga, mengangkut barang ringan, pekerjaan kantoran
	4	Bersepeda santai (kurang dari 2 km/jam), mengecat, pekerjaan tukang
	4,5	Berenang
	5	Berjalan cepat, bersepeda (2-4 km/jam)
Berat	6	Tenis lapangan, berkebun (mencangkul,

		menggali), mendaki gunung
7		Jogging
8		Kegiatan aerobik (Push-up, Sit-up, dll)
10		Berlari (8-10 km per jam)
13,5		Berlari (10-12 km per jam)

Setelah mendapatkan BMR, kita dapat menerapkan persamaan Harris-Benedict untuk mendapatkan kebutuhan kalori harian yang didapat dari mengalikan BMR dengan tingkat aktivitas dari seseorang.

$$\text{Kebutuhan Kalori} = \text{BMR} \times \text{Tingkat Aktivitas} \dots\dots\dots (2.2)$$

Persamaan Harris-Benedict ini merupakan metode yang dipakai untuk memperkirakan kebutuhan kalori untuk mempertahankan berat badan sekarang. Persamaan ini mengasumsikan komposisi tubuh normal, dengan berat badan seimbang, sehingga persamaan ini mungkin kurang akurat untuk orang yang sangat gemuk atau orang yang sangat kurus (Roza & Shizgal: 1984).

Setiap tingkat aktivitas memiliki konstanta tersendiri dalam perhitungan kebutuhan kalori harian. Tabel konstanta tingkat aktivitas dapat dilihat pada Tabel 2.2.

Tabel 2.2 Konstanta Tingkat Aktivitas

<b>Aktivitas</b>	<b>Pria</b>	<b>Wanita</b>
Ringan	1,3	1,3
Sedang	1,65	1,55
Berat	1,76	1,7

Kemudian, harus dilakukan pengecekan proporsi tubuh dengan *Body Mass Index* (BMI) dengan rumus berikut.

$$BMI = \frac{\text{Berat Badan}}{(\text{Tinggi Badan})^2} \dots\dots\dots(2.3)$$

Berdasarkan BMI, dapat diketahui proporsi tubuh yang dikategorikan menjadi

- Underweight : BMI kurang dari 18,5
- Normal : BMI antara 18,5 dan 24,9
- Overweight : BMI lebih dari 24,9

Kemudian, dari kategori proporsionalitas tubuh, perlu dilakukan penyesuaian kebutuhan kalori dengan aturan sebagai berikut.

- Underweight : Kekurangan berat badan, sehingga kalori yang harus dikonsumsi akan ditambahkan.

$$\text{Kalori} = \text{Kalori} + 100$$

- Normal : Berat badan ideal, tidak perlu penyesuaian kebutuhan kalori

- Overweight : Kelebihan berat badan sehingga kalori yang harus dikonsumsi akan dikurangi

$$\text{Kalori} = \text{Kalori} - 500$$

Contoh perhitungannya, apabila seorang pria berusia 20 tahun dengan tinggi badan 170 cm dan berat badan 75 kg ingin mengetahui kebutuhan kalori hariannya. Aktivitas yang dilakukan pria tersebut sehari-

hari adalah pekerjaan kantoran. Hal pertama yang harus dilakukan adalah menghitung BMR.

$$BMR = 66 + (13,7 \times BB) + (5 \times TB) - (6,8 \times U)$$

$$BMR = 66 + (13,7 \times 75) + (5 \times 170) - (6,8 \times 20)$$

$$BMR = 66 + 1027,5 + 850 - 136$$

$$BMR = 1807,5$$

Kemudian, yang harus dilakukan adalah menentukan tingkat aktivitas. Aktivitas pria tersebut adalah pekerjaan kantoran yang dapat dikategorikan aktivitas sedang. Dengan demikian, konstanta tingkat aktivitasnya adalah 1,65.

Setelah itu, perlu dilakukan pengecekan proporsi tubuh dengan BMI.

$$BMI = \frac{\text{Berat Badan}}{\text{Tinggi Badan}^2}$$

$$BMI = \frac{75}{1,70^2}$$

$$BMI = 25,9 = \text{Overweight}$$

Setelah itu, dengan mengalikan konstanta tingkat aktivitas dan BMR, didapatlah kebutuhan kalori harian dari pria tersebut.

$$\text{Kebutuhan Kalori} = BMR \times \text{Aktivitas}$$

$$\text{Kebutuhan Kalori} = 1807,5 \times 1,65 \text{ kal}$$

$$\text{Kebutuhan Kalori} = 2982,375 \text{ kal}$$

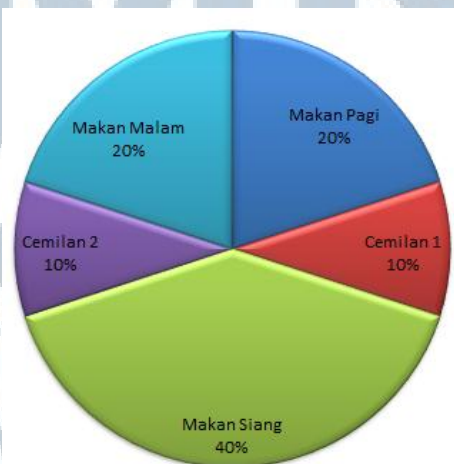
Berdasarkan perhitungan BMI, pria tersebut mengalami overweight, sehingga kebutuhan kalorinya harus dikurangi sebesar 700 kalori.

$$\text{Kebutuhan Kalori} = 2982,375 - 700 \text{ kal}$$

$$\text{Kebutuhan Kalori} = 2282,375 \text{ kal}$$

Menurut Lutfi Utami, toleransi kebutuhan kalori seseorang adalah 90% dari perhitungan kalori. Dengan demikian, kebutuhan kalori dari pria tersebut memiliki rentang yang berkisar antara 2000 kal s.d. 2282,375 kal.

Lutfi Utami juga menyebutkan, pola makan yang ideal adalah dengan makan lima kali sehari, dengan komposisi makan pagi sebesar 20% dari kebutuhan kalori per hari, makan siang sebesar 40% dari kebutuhan kalori per hari, makan malam sebesar 20% dari kebutuhan kalori per hari, dan dua kali cemilan dengan komposisi sekitar 10% dari kebutuhan kalori per hari. Cemilan ini dapat diselipkan di antara makan pagi dan makan siang, serta di antara makan siang dan makan malam. Ilustrasi komposisi makan dalam sehari dapat dilihat pada Gambar 2.2.



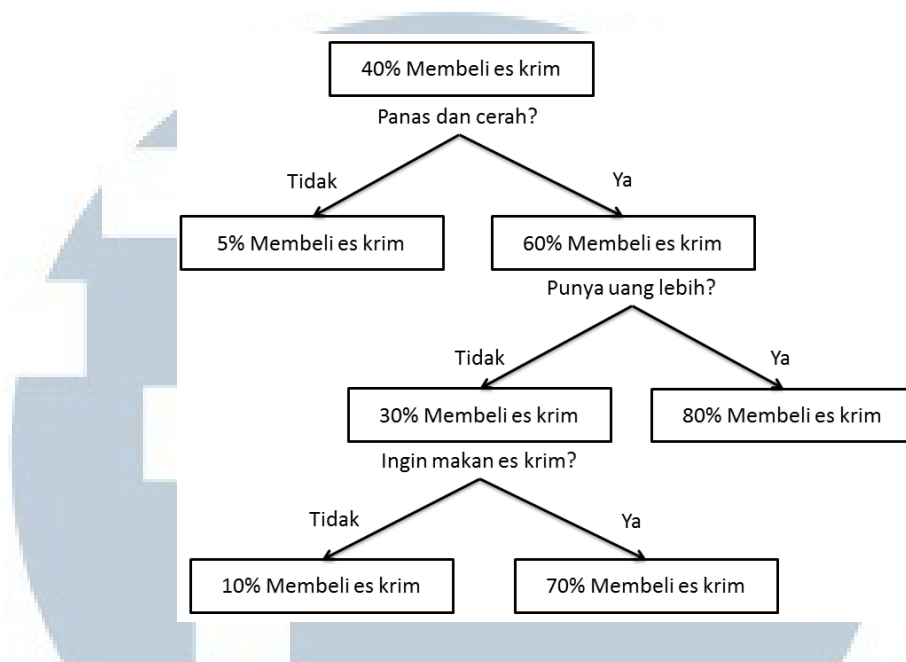
Gambar 2.4 Komposisi Makan dalam Sehari

## 2.4 Decision Tree

Menurut Neville, *decision tree* adalah metode yang berguna untuk pengambilan keputusan. *Decision tree* dimulai dari sebuah *root node* (titik awal) yang dipakai untuk mengambil tindakan. Dari *root node ini*, dibuat aturan yang memecah data menjadi beberapa bagian, yang kemudian akan dipecah lagi berdasarkan aturan lain sehingga membentuk pohon yang menghasilkan keputusan (Neville: 1999).

Contoh pemodelan *decision tree* oleh Neville dimulai dari pengumpulan data yang dilakukan melalui pengamatan terhadap orang di sebuah taman di sekitar kios es krim. Pemilik kios ingin tahu apa yang memengaruhi orang untuk membeli es krim. Dari hasil pengamatan, diketahui 40% orang membeli es krim. Hal ini direpresentasikan sebagai node akar pohon di bagian atas diagram. Aturan pertama membagi data sesuai dengan cuaca. Ketika cuaca tidak cerah dan panas, hanya 5% orang yang membeli es krim. Hal ini diwakili dalam daun pada cabang kiri. Pada hari-hari cerah dan panas, 60% orang membeli es krim. Pohon itu mewakili populasi sebagai node internal yang dibagi lagi menjadi dua cabang.

U M N  
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 2.5 Contoh Decision Tree

Data dalam *decision tree* dinyatakan dalam bentuk tabel dengan atribut dan *record*. Atribut menyatakan suatu parameter yang dibuat sebagai kriteria dalam pembentukan *tree*. Misalnya, untuk menentukan main tenis, kriteria yang diperhatikan adalah cuaca, angin dan temperatur.

Salah satu atribut merupakan atribut yang menyatakan data solusi per-item data yang disebut dengan target atribut. Atribut memiliki nilai yang dinamakan dengan *instance*. Misalnya, atribut cuaca memiliki instance berupa cerah, berawan, dan hujan. Contoh tabel atribut dapat dilihat pada tabel berikut.

Tabel 2.3 Contoh Atribut

Sample	Atribut			Target Atribut
Hari	Cuaca	Angin	Temperatur	Main?
1	Cerah	Kencang	Panas	Tidak
2	Cerah	Pelan	Panas	Ya
3	Berawan	Kencang	Sedang	Tidak
4	Hujan	Kencang	Dingin	Tidak
5	Cerah	Lambat	Dingin	Ya



*Entropy (S)* adalah jumlah bit yang diperkirakan dibutuhkan untuk dapat mengekstrak suatu kelas (+ atau -) dari sejumlah data acak pada ruang sample  $S$ . Entropy juga selanjutnya dipergunakan untuk menentukan atribut mana dari data yang akan digunakan sebagai *node root*. Entropy didefinisikan dengan

$$Entropy (S) = -p_+ \log_2 p_+ - p_- \log_2 p_- \dots\dots\dots(2.4)$$

di mana :

$S$  adalah data sample yang digunakan untuk training

$P_+$  adalah jumlah mendukung data sample

$P_-$  adalah jumlah yang tidak mendukung data sample

Berikut adalah contoh pembentukan *decision tree* untuk penentuan apakah seseorang penderita hipertensi. Data diambil dengan 8 sampel, dengan pemikiran bahwa yang mempengaruhi seseorang menderita hipertensi atau tidak adalah usia, berat badan, dan jenis kelamin. Usia mempunyai atribut muda dan tua. Berat badan memiliki atribut *underweight*, *average*, dan *overweight*. Jenis kelamin memiliki atribut pria dan wanita. Data dari 8 sampel adalah sebagai berikut.

Tabel 2.4 Contoh Sampel Penentuan Penderita Hipertensi

<b>Nama</b>	<b>Usia</b>	<b>Berat</b>	<b>Kelamin</b>	<b>Hipertensi</b>
Ali	Muda	Overweight	Pria	Ya
Dustin	Muda	Underweight	Pria	Tidak
Ayu	Muda	Average	Wanita	Tidak
Budi	Tua	Overweight	Pria	Tidak
Henri	Tua	Overweight	Pria	Ya
Dinal	Muda	Underweight	Pria	Tidak
Rina	Tua	Overweight	Wanita	Ya
Jarwo	Tua	Average	Pria	Tidak

Langkah penyelesaian yang harus dilakukan:

1. Menentukan node yg menjadi root

Untuk menentukan *node root*, digunakan nilai entropy dari setiap kriteria dengan data sampel yang ditentukan. *Node* yang dipilih adalah yang memiliki kriteria dengan *node* yang paling kecil.

- Hitung entropy untuk *node* usia

Tabel 2.5 Kriteria Usia

Usia	Hipertensi	Jumlah
Muda	Ya	1
Muda	Tidak	3
Tua	Ya	2
Tua	Tidak	2

Usia: muda

$$q_1 = -\frac{1}{4} \log_2 \frac{1}{4} - \frac{3}{4} \log_2 \frac{3}{4} = 0,81$$

Usia: tua

$$q_2 = -\frac{2}{4} \log_2 \frac{2}{4} - \frac{2}{4} \log_2 \frac{2}{4} = 1$$

Entropy untuk Usia

$$E = \frac{4}{8} q_1 + \frac{4}{8} q_2 = 0,91$$

- Hitung entropy untuk *node* jenis kelamin

Tabel 2.6 Kriteria Jenis Kelamin

Kelamin	Hipertensi	Jumlah
Pria	Ya	2
Pria	Tidak	4
Wanita	Ya	1
Wanita	Tidak	1

Kelamin: Pria

$$q_1 = -\frac{2}{6} \log_2 \frac{2}{6} - \frac{4}{6} \log_2 \frac{4}{6} = 0,5$$

Kelamin: Wanita

$$q_2 = -\frac{1}{4} \log_2 \frac{1}{4} - \frac{1}{4} \log_2 \frac{1}{4} = 1$$

Entropy untuk Kelamin

$$E = \frac{6}{8} q_1 + \frac{2}{8} q_2 = 0,94$$

- Hitung entropy untuk *node* berat badan

Tabel 2.7 Kriteria Berat Badan

Berat	Hipertensi	Jumlah
Overweight	Ya	3
Overweight	Tidak	1
Average	Ya	0
Average	Tidak	2
Underweight	Ya	0
Underweight	Tidak	2

Berat: Overweight

$$q_1 = -\frac{3}{8} \log_2 \frac{3}{8} - \frac{1}{8} \log_2 \frac{1}{8} = 0,415$$

Berat: Average

$$q_2 = -\frac{0}{8} \log_2 \frac{0}{8} - \frac{2}{8} \log_2 \frac{2}{8} = 0,5$$

Berat: Underweight

$$q_3 = -\frac{0}{8} \log_2 \frac{0}{8} - \frac{2}{8} \log_2 \frac{2}{8} = 0,5$$

Entropy untuk Berat

$$E = \frac{4}{8}q_1 + \frac{2}{8}q_2 + \frac{2}{8}q_1 = 0,41$$

Karena atribut berat badan memiliki entropy paling kecil, maka berat badan dipilih sebagai *root node*.

## 2. Membentuk *Tree*

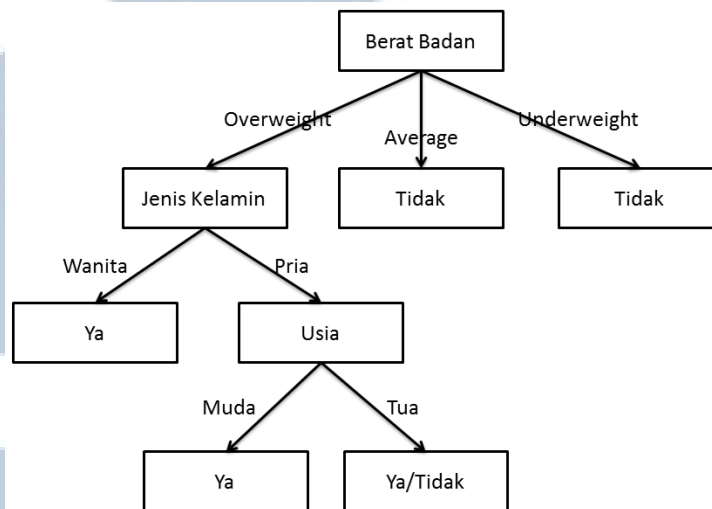
*Leaf node* berikutnya dipilih pada bagian yang memiliki nilai + dan -, pada contoh ini, hanya pada bagian berat = overweight yang memiliki nilai + dan -. Untuk penyusunan *leaf node*, juga dilakukan penghitungan entropy.

Tabel 2.8 Kriteria untuk Pemilihan Leaf Node

Usia	Hipertensi	Jumlah	Kelamin	Hipertensi	Jumlah
Muda	Ya	1	Pria	Ya	2
	Tidak	0		Tidak	1
Tua	Ya	2	Wanita	Ya	1
	Tidak	1		Tidak	0
Entropy = 0,69			Entropy = 0,69		

*Leaf node* usia dan jenis kelamin memiliki Entropy yang sama, sehingga tidak ada cara lain selain *node* dipilih secara acak. Dengan demikian, *decision tree* dapat disusun sebagai berikut.

U M N  
 U N I V E R S I T A S  
 M U L T I M E D I A  
 N U S A N T A R A



Gambar 2.6 Decision Tree untuk Hipertensi

Pada sampel, usia: tua, ternyata terdapat satu data yang menyatakan ya dan satu data yang menyatakan tidak, keadaan ini perlu dicermati. Pilihan hanya dapat ditentukan dengan bantuan dari seseorang yang ahli dalam bidang ini.

### 3. Menyusun *rules*

Dari *decision tree* yang telah dibentuk, dapat disusun *rules* sebagai berikut.

Rule 1 : IF berat = average OR berat = underweight

THEN hipertensi = tidak

Rule 2 : IF berat = overweight AND kelamin = wanita

THEN hipertensi = ya

Rule 3 : IF berat = overweight AND kelamin = pria AND usia = muda

THEN hipertensi = ya

Rule 4 : IF berat = overweight AND kelamin = pria AND usia = tua

THEN hipertensi = tidak

## 2.5 Teknologi Web

Beberapa tahun belakangan ini, *internet*, terutama *world wide web* (atau disingkat *web*), berkembang sangat cepat sebagai media bertukar informasi untuk berbagai aplikasi seperti *web banking*, *online trading*, *e-commerce*, *e-government*, *digital library*, dan sebagainya. Alasan mendasar dari perkembangan web yang sangat pesat ini adalah karena pendistribusian informasi yang berlangsung dengan cepat, mudah, fleksibel, dan murah. Banyak perusahaan memiliki kebutuhan untuk memenuhi kepuasan pelanggannya atau atau kepastian atas kebutuhan dari pelanggan untuk meminimalisir terjadinya kesalahan. Selain itu, faktor biaya juga memegang peran penting perusahaan menggunakan teknologi *web* untuk otomatisasi pekerjaan.

Salah satu keuntungan sistem pakar yang sangat terasa manfaatnya adalah kemampuan untuk menyediakan layanan pakar tanpa mengharuskan kehadiran pakar saat itu juga. Salah satu contohnya adalah sistem pakar diagnosa. Misalnya, sebuah perusahaan yang bergerak di bidang perawatan *elevator*, dapat menyediakan sistem pakar yang mampu menentukan apakah masalah yang ditemukan pada *elevator* memerlukan teknisi untuk memperbaiki *elevator* tersebut atau pelanggannya dapat menangani masalah tersebut sendiri. Hal ini dapat menghemat waktu dan biaya di kedua belah pihak (<http://ptcpartners.com>).

Apabila sistem pakar diterapkan secara *online*, akan memberikan keuntungan yang jauh lebih besar lagi. Dengan penerapan sistem pakar secara *online*, akan didapat layanan pakar di mana saja, kapan saja selama terhubung dengan *internet*.

Menurut Grove (Grove: 2000), terdapat beberapa alasan yang mendukung dikembangkannya sistem pakar berbasis *web*, yaitu:

- *Internet* diakses oleh banyak orang

Penggunaan *internet* terus bertambah banyak, dengan perkiraan pengguna *internet* sebesar 100 juta di Amerika Serikat dan 200 juta di seluruh dunia (NUA Internet Surveys, 2000). Penggunaan *internet* menjadi hal umum dalam transaksi bisnis. Kehadiran teknologi nirkabel juga membuat akses *internet* menjadi portabel. Kehadiran *internet* dapat menjadi lahan berkembangnya sistem pakar berbasis *web* yang dapat diakses di mana saja di belahan dunia.

- *Web browser* dapat menyediakan antarmuka multimedia

*Web browser* yang kompatibel dengan HTML tersedia di semua perangkat komputer. Format yang disediakan oleh HTML menjadi antarmuka standar yang dapat dimanfaatkan oleh pengembang *web*. Kehadiran standar antarmuka ini tidak hanya meringankan pekerjaan pengembang web, tetapi juga mengurangi waktu untuk melakukan *training* pada *user*. Komponen multimedia (gambar, tulisan, suara, dan lainnya) diintegrasikan dalam antarmuka berbasis *web*.

- Ketersediaan komponen pengembangan sistem pakar yang kompatibel dengan *internet*

Terdapat berbagai komponen pengembangan sistem pakar yang memfasilitasi sistem berbasis *web*, termasuk *server*, antarmuka HTML, dan komabilitas dengan protokol *internet*, contohnya:

- *Java Expert System Shell* (Jess) merupakan komponen pengembang sistem pakar berorientasi objek dengan komponen Java (<http://herzberg.ca.sandia.gov/jess>)
- XpertRule KBS merupakan komponen pengembang sistem pakar berbasis aturan yang memiliki antarmuka berbasis *web* dengan memanfaatkan teknologi *Microsoft Active Server Page* (<http://www.attar.com>)
- Acquire, merupakan komponen pengembang sistem pakar yang juga mendukung sistem pakar berbasis aturan dan sistem pakar berbasis pola. Pengembangan antarmuka berbasis internet didukung dengan bahasa Java dan kontrol *ActiveX* (<http://www.aiinc.ca>)
- Aplikasi berbasis *internet* dapat diakses di mana saja

*Internet* menyediakan lingkungan pengembangan aplikasi yang tidak tergantung *platform* dan tersedia di mana saja. Untuk sistem pakar berbasis *web*, hal ini memiliki arti bahwa tidak diperlukan adanya instalasi perangkat lunak untuk sistem pakar sebelum dapat digunakan. Aplikasi tersedia kapan saja dan di mana saja. Selain itu,



aplikasi dapat dimodifikasi tanpa ada efek langsung terhadap pengguna.

- Munculnya protokol yang mendukung sistem pakar  
Protokol seperti CORBA (*Common Object Request Broker Architecture*), DCOM (*Distributed Component Object Model*), dan Jini menyediakan mekanisme standar untuk pertukaran informasi dan layanan antar aplikasi yang dapat dimanfaatkan dalam membangun sistem pakar.

Terdapat beberapa contoh sistem pakar berbasis web, yaitu sebagai berikut.

- *The Expert System for Thermodynamics (TEST)*, dikembangkan di San Diego State University (Bhattacharjee: 2000), yang membantu menjawab masalah termodinamika. TEST didesain berdasarkan *decision tree* berbasis HTML yang memanfaatkan *Java applet* untuk melakukan kalkulasi.
- *The Reptile Identification Helper* (Grove & Hulse: 1999) merupakan sistem pakar berbasis web untuk mengidentifikasi spesimen amfibi dan reptil yang ditemukan di Pennsylvania. *The Reptile Identification Helper* didesain sebagai aplikasi *client-server*, dengan memanfaatkan *Java Expert System Shell* sebagai *inference engine* dan HTML beserta Javascript sebagai antarmuka (<http://www.nsm.iup.edu/pha/rih>)
- *The Coating Alternatives Guide* dan *Solvent Alternatives Guide*, dikembangkan oleh Pollution Prevention Program di Research Triangle

Institute dengan tujuan untuk mengurangi polusi industri. Kedua sistem tersebut merupakan sistem pakar berbasis aturan yang dikembangkan dengan ColdFusion.

## 2.6 IPAQ Questionnaire

IPAQ Questionnaire merupakan kuesioner yang disediakan oleh IPAQ (International Physical Activity Questionnaire), sebuah lembaga yang menyediakan instrumen yang dapat dipergunakan untuk mendapatkan perkiraan yang sebanding dengan aktivitas fisik (<https://sites.google.com/site/theipaq/>). Terdapat dua versi dari kuesioner, yaitu versi pendek yang cocok untuk survei dan versi panjang yang menyediakan informasi yang lebih detail untuk keperluan penelitian atau evaluasi. IPAQ Questionnaire ini tersedia secara publik, dapat diakses oleh siapa saja, dan tidak diperlukan izin untuk menggunakannya.

## 2.7 Interaksi Makanan dan Makanan

Menurut Lutfi Utami, interaksi makanan dan makanan merupakan efek samping yang tidak diinginkan yang terjadi di dalam tubuh manusia karena mengonsumsi dua jenis atau lebih makanan tertentu secara bersamaan. Pada umumnya, interaksi seperti ini dianggap tidak berbahaya, namun dalam jangka waktu panjang dikhawatirkan dapat menjadi efek samping yang serius. Beberapa contoh interaksi makanan dan makanan yaitu:

- Susu kedelai - telur

Susu kedelai bisa mengurangi aktivitas dari enzim protease, yang digunakan oleh tubuh untuk membantu memetabolisme protein, sementara telur banyak protein.

- Buah - *seafood*

Buah-buahan mengandung tannin yang jika digabungkan dengan protein bisa menghasilkan zat yang tidak larut dan tidak bisa dicerna. Sebaiknya mengonsumsi buah-buahan tersebut 4 jam setelah makan makanan laut.

- Susu - coklat

Susu kaya akan protein dan kalsium sementara coklat kaya akan asam oksalat. Jika dikonsumsi bersama-sama, kalsium dari susu dan asam oksalat dari coklat bisa bergabung dan membentuk kalsium oksalat yang tidak dapat larut dalam tubuh. Akibatnya bukan hanya tidak bisa dicerna oleh tubuh tapi bisa menyebabkan diare.

UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA