



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

TINJAUAN PUSTAKA

2.1 Kalori

Kalori adalah satuan unit yang digunakan untuk mengukur nilai tenaga atau energi. Kandungan kalori di dalam suatu makanan bergantung kepada karbohidrat, protein, dan lemak yang ada dalam makanan itu sendiri. Makanan yang mengandung karbohidrat, protein, dan lemak menghasilkan kalori bagi tubuh. Lemak menghasilkan kalori yang terbanyak yaitu sebesar 9 kalori/gram, sedangkan karbohidrat dan protein mengandung 4 kalori tiap gramnya (Graha, 2010).

Makanan yang mengandung banyak lemak adalah makanan yang mengandung tinggi kalori, sebaliknya makanan yang mengandung air yang tinggi seperti buah-buahan dan sayuran memiliki kadar kalori yang rendah. Apabila kalori dalam tubuh melebihi dari yang dibutuhkan, maka kalori yang tidak digunakan itu akan disimpan tubuh dalam dua bentuk yaitu (Graha, 2010):

- Sebagai glikogen yang disimpan di dalam hati dan otot.
- Sebagai lemak yang disimpan di bawah jaringan kulit

Kalori yang masuk ke tubuh berasal dari makanan yang dikonsumsi, karenanya perlu diatur pola makan agar kalori yang masuk dapat optimal di pergunakan tubuh dan tidak tertumpuk di badan yang akhirnya mengakibatkan kegemukan. Untuk itu

perlu diperhatikan asupan yang dikonsumsi dan bagaimana mengonsumsi makanan yang masuk ke dalam tubuh itu.

Berikut ini adalah daftar kandungan kalori pada beberapa makanan yang sering di konsumsi sehari-hari :

Tabel 2.1 Daftar kalori makanan (Graha, 2010)

No	Makanan	Porsi	Berat(g)	Kalori/porsi (kkal)
1	Abon	1 sendok makan sedang	9	25
2	Ampela ayam	1 potong sedang	10	10
3	Apple pie	1 potong sedang	70	166
4	Ayam	1 potong sedang	50	88
5	Babat	1 potong sedang	30	34
6	Bacang	1 potong sedang	70	72
7	Bakso	1 porsi sedang	250	190
8	Bakwan	1 potong sedang	40	109
9	Batagor	1 bungkus sedang	152	231
10	Bawal	1 ekor sedang	132	102
11	Bayam rebus	1 mangkuk sedang	223	51
12	Bayam, tumis+oncom	1 mangkuk sedang	240	245
13	Bayam, tumis bersantan	1 mangkuk sedang	100	178
14	Bebek	1 potong sedang	45	88
15	Beras giling masak (nasi)	$\frac{3}{4}$ gelas sedang	100	178
16	Bihun goreng	1 porsi sedang	200	308
17	Bika ambon	1 buah sedang	70	158
18	Biskuat krim coklat	1 bungkus kecil	16,5	81
19	Biskuat krim	1 bungkus kecil	16,5	81
20	Biskuat merah	1 bungkus kecil	5,17	24
21	Biskuat susu krim	1 bungkus kecil	16,5	81
22	Biskuit	1 keping	10	46
23	Biskuit marrie roma	1 keping	10	46
24	Blackforest	1 potong sedang	50	182
25	Brownies	1 potong sedang	45	210
26	Bubur sumsum	1 gelas sedang	200	70
27	Buntil	1 buah sedang	100	106
28	Buras	1 sendok makan sedang	70	88

No	Makanan	Porsi	Berat(g)	Kalori/posisi (kkal)
29	Cakue/roti goreng	1 potong sedang	40	158
30	Cendol/dawet	1 gelas sedang	150	45
31	Cincau	1 gelas sedang	150	2
32	Combro	1 buah sedang	50	105
33	Dadar gulung	1 porsi sedang	60	173
34	Daging sapi	1 potong sedang	50	104
35	Daun katuk, rebus	1 gelas sedang	100	53
36	Dendeng daging sapi	1 potong sedang	16	68
37	Energen	1 bungkus	30	133
38	Dodol	1 potong sedang	17	67
39	Es krim	1 cup sedang	60	94
40	Es Mambo	1 bungkus sedang	25	38
41	Es sirup	1 gelas sedang	125	56
42	Gado-gado	1 porsi sedang	150	203
43	Getuk lindri	1 porsi sedang	35	60
44	Goreng oncom	1 buah sedang	30	109
45	Gudeg	1 porsi sedang	100	53
46	Hati ayam	1 potong sedang	50	58
47	Hati sapi	1 potong sedang	50	68

2.2 Backward Chaining

Suatu perkalian inferensi yang menggabungkan suatu permasalahan dengan solusinya disebut dengan rantai (*chain*). Suatu rantai yang dilintasi dari suatu hipotesa kembali ke fakta yang mendukung hipotesa tersebut adalah *backward chaining*. Cara lain menggambarkan *backward chaining* adalah dalam hal tujuan yang dapat dipenuhi dengan pemenuhan sub tujuannya (Kusrini, 2006). Menurut Giarattano dan Riley dalam runut balik, penalaran dimulai dengan tujuan merunut balik ke jalur yang akan mengarahkan ke tujuan tersebut (Rilley, 1996)

Runut balik disebut juga sebagai *goal-driven reasoning*, merupakan cara yang efisien untuk memecahkan masalah yang dimodelkan sebagai masalah pemilihan terstruktur. Tujuan dari inferensi ini adalah mengambil pilihan terbaik dari banyak

kemungkinan (Kusrini, 2006). Menurut Schnupp metode inferensi ini cocok digunakan untuk memecahkan masalah diagnosis.

Dalam metode ini, pendekatan yang dimotori tujuan (*goal-driven*). Dalam pendekatan ini pelacakan dimulai dari tujuan, selanjutnya dicari aturan yang memiliki tujuan tersebut untuk kesimpulannya. Selanjutnya proses pelacakan menggunakan premis untuk aturan tersebut sebagai tujuan baru dan mencari aturan lain dengan tujuan baru sebagai kesimpulannya. Proses berlanjut sampai semua kemungkinan ditemukan (Kusrini, 2008).

Langkah - langkah perancangan sistem dengan menggunakan metode *backward chaining* (Kusrini, 2006) :

1. Dengan membuat *dependency diagram*, *dependency diagram* menunjukkan faktor-faktor kritis, pertanyaan yang diinputkan, *rules*, *values*, dan rekomendasi yang dihasilkan.
2. Pereduksian *rule*, *rule* ini dibuat berdasarkan *dependency diagram* yang sudah terbentuk. Struktur dan *syntax* penulisan *rule* adalah sebagai berikut:
 - *RULE label* : label berisi nama *rule* tersebut.
 - *IF* : sebagai penanda awal kondisi pada sebuah *rule*.
 - *THEN* : sebagai penanda awal kesimpulan dari sebuah *rule*.

Operator yang dapat digunakan pada IF-THEN rule adalah:

- *AND* : semua kondisi yang dihubungkan dengan operator ini harus bernilai benar, agar kondisi keseluruhan *rule* tersebut bernilai benar.

- OR : bila semua kondisi yang dihubungkan oleh operator ini harus bernilai salah, maka kondisi keseluruhan *rule* tersebut bernilai salah. Bila ada sama kondisi atau lebih yang bernilai benar maka keseluruhan *rule* bernilai benar.

Salah satu metode inferensi yang berhubungan dekat dengan *backward chaining* adalah *forward chaining*. *Forward chaining* adalah suatu strategi pengambilan keputusan yang dimulai dari bagian premis (fakta) menuju konklusi atau kesimpulan akhir (Kusrini, 2006). Contoh: (Safia Dhany, 2009)

IF Sulit bernafas

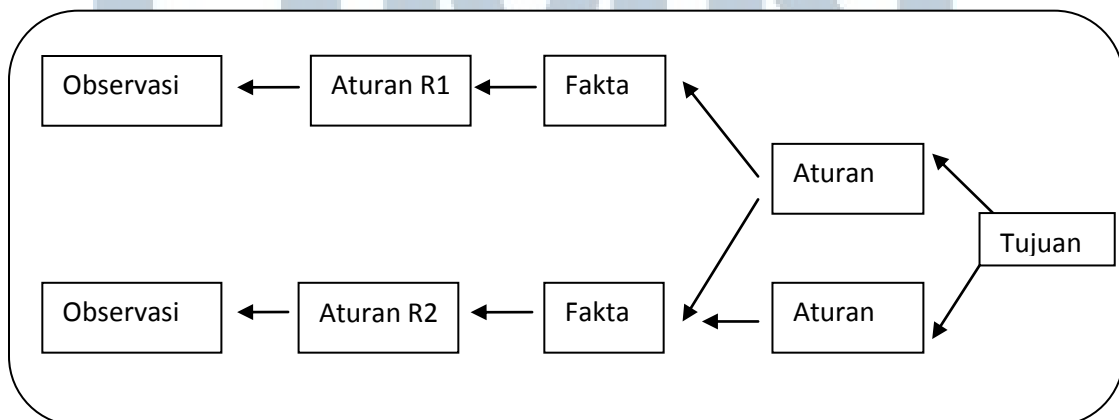
AND Pilek

AND Batuk kering

AND Tarikan nafas berbunyi kasar dan penghembusan nafas berbunyi mengi

THEN Laringitis

Secara sederhana dapat dijelaskan bahwa untuk kaidah di atas, agar sistem dapat mencapai konklusi, harus diinput terlebih dahulu fakta sulit bernafas, pilek, tarikan nafas berbunyi kasar dan penghembusan nafas berbunyi mengi, baru sistem dapat mengeluarkan konklusi bahwa penyakit yang diderita adalah laringitis.



Gambar 2.1 Proses *Backward Chaining* (Sumber : Kusrini.2006)

2.3 Food Combining

Food Combining adalah salah satu metode diet yang saat ini sedang populer di kalangan masyarakat. Dimana diet dilakukan dengan cara mengombinasikan menu-menu makanan sehat yang dapat memenuhi kebutuhan kalori yang dibutuhkan. Metode ini bertujuan mengatur asupan makanan yang sesuai dengan mekanisme alamiah tubuh (sistem pencernaan) agar tidak terjadi penumpukan kelebihan zat makanan karena proses metabolisme tubuh lebih efisien (Gunawan, 2006).

Bahan makanan adalah segala sesuatu yang berasal dari sumber hayati dan air, baik yang diolah maupun tidak diolah, yang diperuntukkan sebagai makanan dan minuman bagi konsumsi manusia termasuk bahan tambahan pangan, bahan baku pangan, dan bahan lain yang digunakan dalam proses penyiapan, pengolahan dari atau pembuatan makanan dan minuman (Gunawan, 2006).

Energi yang berada di dalam tubuh kita dipergunakan untuk :

1. Melakukan pekerjaan eksternal,
2. Melakukan pekerjaan internal dan untuk mereka yang masih tumbuh,
3. Keperluan pertumbuhan, yaitu untuk sintesis senyawa-senyawa baru.

Setiap makanan tidak sama banyaknya dalam menghasilkan energi, padahal manusia harus mendapatkan sejumlah makanan tertentu yang setiap harinya untuk menghasilkan energi, terutama untuk mempertahankan proses kerja tubuh dan menjalankan kegiatan-kegiatan fisik. Cara-cara menentukan kebutuhan kalori menggunakan teori RBW (*Relative Body Weight*) atau sering disebut dengan teori berat badan relatif (Hartono, 2006)

$$RBW = (BB \text{ (Kg)} / (TB \text{ (Cm)} - 100)) \times 100 \%$$

... rumus 2.1

BB = Berat Badan

TB = Tinggi Badan

Dengan ketentuan :

- a. Kurus jika $RBW < 90\%$
- b. Normal jika $RBW = 90 - 100 \%$
- c. Gemuk jika $RBW > 110 \%$ atau kurang 120%
- d. Obesitas Ringan $RBW 120 - 130 \%$
- e. Obesitas Sedang $RBW 130 - 140 \%$
- f. Obesitas Berat $RBW > 140 \%$

Kebutuhan kalorinya adalah :

- a. Orang Kurus $BB \times 40 - 60$ Kalori
- b. Orang Normal $BB \times 30$ Kalori
- c. Orang Gemuk $BB \times 20$ Kalori
- d. Orang Obesitas $BB \times (10 - 15)$ Kalori

Berikut ini adalah contoh beberapa kandungan kalori dalam bahan makanan per 100 gram.

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

Tabel 2.2 Kandungan Kalori Bahan Makanan per 100 gram
(Untoro, 2008)

Nama	Kalori	Nama	Kalori
Beras giling	360	Jagung	307
Kentang	83	Tahu	68
Tempe	149	Ikan segar	113
Ayam	302	Udang segar	91
Daging kambing	154	Bayam	36
Daging sapi	207	Daun melinjo	99
Telur ayam	162	Telur bebek	189
Kacang panjang	44	Kangkung	29
Kol	24	Daun pepaya	79
Tomat	20	Wortel	40
Wortel	40	Gula pasir	364

Tabel 2.3 Kandungan Kalori dalam Menu Makanan
(Untoro, 2008)

Nama	Kalori	Nama	Kalori
Nasi goreng ayam	572	Kwetiauw goreng	385
Nasi rames	680	Mi kuah pangsit	410
Nasi soto ayam	596	Mi kering bakso sapi	390

Nama	Kalori	Nama	Kalori
Bubur ketan item dengan santan	530	Bubur kacang ijo dengan santan	435
Nasi rawon	680	Gado-gado lontong	488
Tumis taoge dengan tahu goreng	210	Tumis caisim saus tiram	255
Lontong sayur	420	Bakwan	280
Udang goreng tepung	145	Puyunghai isi kapri, wortel dan udang	255
Telur balado	70	Kacang cah sapi	350

2.4 Android

Android adalah sistem operasi yang berbasis linux untuk telepon seluler seperti *smart phone* dan komputer tablet. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc membeli Android Inc, yang merupakan pendatang baru yang membuat piranti lunak untuk ponsel. Kemudian untuk mengembangkan android dibentuklah *Open Handset Alliance* yang merupakan konsosium dari 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi, termasuk Google, HTC (*High Tech Computer*), Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia (developer.android.com).

Berikut adalah beberapa tipe dari sistem operasi android (developer.android.com):

- Versi 1.1

- Versi 1.5 (Cupcake)
- Versi 1.6 (Donut)
- Versi 2.0/2.1 (Eclair)
- Versi 2.2 (Froyo)
- Versi 2.3 (Gingerbread)
- Versi 3.0 (Honeycomb)
- Versi 4.0 (Ice Cream Sandwich)

Fitur-fitur yang dimiliki oleh Android diantaranya (developer.android.com).

1. *Application framework* yang memungkinkan penggunaan kembali dan penggantian komponen-komponen yang tersedia.
2. *Dalvik virtual machine* yang dioptimalkan untuk perangkat *mobile*.
3. *Integrated browser* berbasis *open source Webkit engine*.
4. *Optimized graphics* yang diperkuat dengan *custom 2D graphics library* dan *3D graphics* berbasis OpenGL ES 1.0.
5. *SQLite* untuk penyimpanan data.
6. *Media support* untuk format audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
7. *GSM Telephony*
8. *Bluetooth, EDGE, #3G, and WiFi*

Secara garis besar arsitektur Android dapat dijelaskan dan digambarkan

sebagai berikut :



Gambar 2.2 Arsitektur Android (developer.android.com)

1. Linux Kernel

Sistem operasi Android dikembangkan dari GNU / kernel Linux, diciptakan oleh Linus Torvald ketika ia masih menjadi mahasiswa di Universitas Helsinki pada tahun 1991. Android dapat dimasukkan ke berbagai perangkat, hal ini karena GNU/Linux mendukung kernel yang memiliki lapisan abstraksi perangkat keras. Selain itu, dengan GNU/kernel Linux juga mendukung manajemen memori, manajemen proses, jaringan dan jenis *service* lainnya.

2. Libraries

Layer di atas kernel adalah Pustaka Asli (*Native Libraries*). Pustaka asli android semua ditulis dalam C atau C + +, dikompilasi untuk arsitektur perangkat keras

khusus yang digunakan oleh *smartphone*, dan diinstal oleh vendor *smartphone* juga. Beberapa pustaka asli yang paling penting adalah sebagai berikut : Surface Manager, 2D dan 3D graphics, media *codecs*, *SQL database*, dan *Browser engine*.

3. *Android Runtime*

Lapisan di atas kernel adalah Android runtime, termasuk mesin virtual Dalvik (Dalvik VM) dan pustaka inti Java. Dalvik VM adalah implementasi Java dari Google, dioptimalkan untuk perangkat *mobile*. Semua kode yang Anda tulis untuk Android akan ditulis di Java dan dijalankan dalam *Virtual Machine*.

4. *Application Framework*

Lapisan di atas *Native Libraries* dan *runtime* adalah lapisan *Application Framework*. Lapisan ini menyediakan blok tingkat tinggi, bagian paling penting dari framework adalah *Activity Manager*, *Content Provider*, *Resource Manager*, *Location Manager*, dan *Notification Manager*.

5. *Application dan Widget*

Lapisan teratas dalam diagram arsitektur Android adalah *Applications* and *Widgets*. *Applications* adalah program yang dapat mengambil alih seluruh layar dan berinteraksi dengan pengguna. Di sisi lain. *Widget* (kadang-kadang disebut *gadget*), hanya beroperasi di sebuah persegi panjang kecil dari aplikasi layar *Home*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA