



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

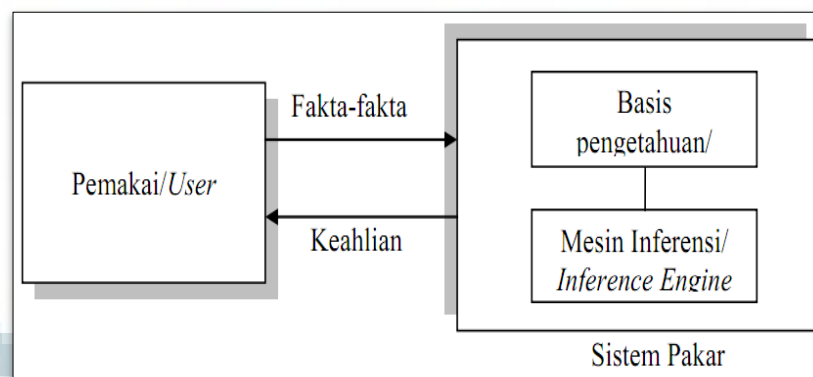
2.1 Sistem Pakar

Sistem pakar (*expert system*) adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut (Kusrini, 2006). Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para ahli. Sistem pakar berfungsi sebagai asisten yang pandai dari seorang pakar. Sistem pakar juga dapat memberikan penjelasan terhadap langkah yang diambil dan memberikan alasan atas saran atau kesimpulan yang ditemukan. Beberapa definisi para ahli mengenai sistem pakar (Kusumadewi, 2002)

1. Menurut Martin dan Oxman : Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah, yang biasanya hanya dapat diselesaikan oleh seorang pakar dalam bidang tertentu.
2. Menurut Ignizio : Sistem pakar merupakan bidang yang dicirikan oleh sistem berbasis pengetahuan (*Knowledge Base System*), memungkinkan adanya komponen untuk berpikir dan mengambil kesimpulan dari sekumpulan kaidah.
3. Menurut Giarratano and Riley : Sistem pakar adalah salah satu cabang kecerdasan buatan yang menggunakan pengetahuan-pengetahuan khusus yang dimiliki oleh seorang ahli untuk menyelesaikan masalah tertentu.

Pengetahuan dalam sistem pakar bisa berupa keahlian atau pengetahuan yang biasanya tersedia di buku-buku, majalah-majalah, dan orang yang berpengetahuan. Dalam hal ini, pengetahuan dianggap sebagai tingkatan yang lebih rendah daripada keahlian langka kebanyakan. Kata-kata *expert system*, *knowledge-based system*, dan *knowledge-based expert system* sering dianggap sama. Banyak orang yang menggunakan *expert system* karena lebih singkat, walaupun mungkin tidak ada keahlian dalam *expert system* tersebut, melainkan pengetahuan.

Dalam penyusunannya, sistem pakar mengkombinasikan kaidah-kaidah penarikan kesimpulan (*inference rules*) dengan basis pengetahuan (*knowledge-base*) tertentu yang diberikan oleh satu atau lebih pakar dalam bidang tertentu. Kombinasi dari kedua hal tersebut disimpan dalam komputer, yang selanjutnya digunakan dalam proses pengambilan keputusan untuk penyelesaian masalah tertentu. Untuk lebih jelasnya, dapat melihat fungsi dasar dari sistem pakar pada Gambar 2.1.



Gambar 2.1 Fungsi dasar dari suatu sistem pakar
Sumber : Giarratano and Riley, 2005

Pada Gambar 2.1, *user* memasukkan fakta-fakta atau informasi lainnya dalam sistem pakar dan menerima saran dari pakar atau keahlian sebagai tanggapannya. *Knowledge-base* berisikan pengetahuan yang bekerja dengan *inference engine* dalam membuat kesimpulan-kesimpulan. Kesimpulan-kesimpulan ini merupakan tanggapan terhadap kebutuhan-kebutuhan *user* terhadap keahlian tertentu.

Secara garis besar banyak kelebihan yang dimiliki oleh sistem pakar, antara lain (Giarratano and Riley, 2005) :

1. Meningkatkan ketersediaan (*increased availability*), dimana keahlian dapat disediakan dalam berbagai perangkat keras komputer yang sesuai.
2. Mengurangi biaya (*reduced cost*), dimana biaya untuk menyediakan keahlian pada setiap *user* jauh lebih rendah.
3. Mengurangi bahaya (*reduced danger*), dimana sistem pakar dapat dipergunakan dalam lingkungan yang mungkin saja berbahaya bagi manusia.
4. Permanen (*permanence*), bahwa pengetahuan sistem pakar permanen tidak seperti manusia yang bisa pensiun, berhenti, atau meninggal sehingga keahliannya tidak dapat dipergunakan lagi.
5. Keahlian ganda (*multiple expertise*), dimana pengetahuan dari berbagai pakar dapat bekerja secara simultan dan bersambung dalam suatu masalah.
6. Meningkatkan kepercayaan (*increased reliability*), dimana *user* mendapatkan pendapat pembandingan terhadap pendapat pakar lainnya (manusia bukan komputer) dari sistem pakar yang digunakan sehingga meningkatkan kenyamanan dalam mengambil suatu keputusan.

7. Penjelasan (*explanation*), sistem pakar dapat menjelaskan secara detail alasan yang mendukung pengambilan keputusan.
8. Tanggapan yang cepat (*fast response*), dimana sistem pakar dapat memberikan tanggapan yang lebih cepat dibandingkan dengan seorang manusia.
9. Tetap, tidak emosional, dan tanggapan yang lengkap setiap waktu, tidak seperti manusia yang dapat berubah kapan saja (tergantung pada emosinya).
10. Pengajar yang cerdas (*intelligent tutor*), dimana *user* dapat menjalankan contoh program dan menjelaskan alasan dari sistem.
11. Basis data yang cerdas (*intelligent database*), dimana sistem pakar dapat dipergunakan untuk mengakses basis data dengan cara yang cerdas.

Konsep dasar sistem pakar mencakup beberapa persoalan mendasar, antara lain siapa yang disebut pakar, apa yang dimaksud dengan keahlian, bagaimana keahlian dapat ditransfer, dan bagaimana sistem bekerja. Konsep dasar sistem pakar terdiri atas (Turban, 2004) :

1. Kepakaran (*Expertise*)

Kepakaran merupakan penguasaan pengetahuan di bidang tertentu yang diperoleh dari serangkaian pelatihan, membaca, atau pengalaman.

2. Pakar (*Expert*)

Seorang pakar adalah orang yang memiliki pengetahuan, penilaian, pengalaman, metode khusus, serta kemampuan untuk menerapkan bakat ini dalam memberi nasihat dan memecahkan masalah.

3. Pengalihan Kepakaran

Tujuan utama sistem pakar adalah mengalihkan kepakaran seorang pakar ke dalam komputer yang akan digunakan oleh pihak lain yang bukan pakar, untuk menemukan solusi atas permasalahan.

4. Penalaran (*Inference*)

Salah satu fitur yang dimiliki oleh sistem pakar adalah kemampuan untuk menalar. Jika kepakaran sudah tersimpan sebagai basis pengetahuan dan tersedia program yang mampu mengakses basis data, maka komputer harus dapat diprogram untuk membuat inferensi.

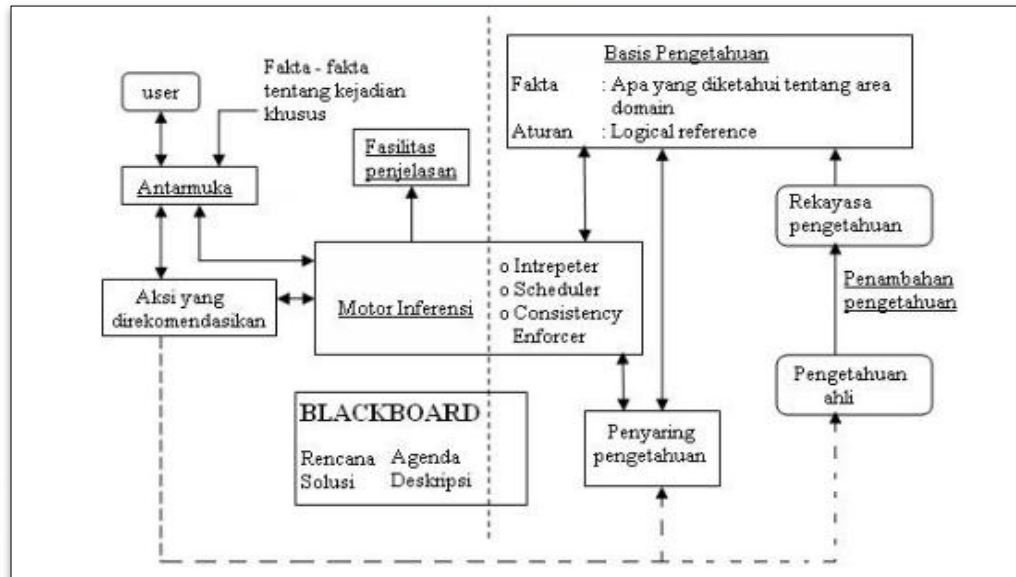
5. Aturan (*Rules*)

Sebagian besar sistem pakar adalah sistem berbasis aturan. Aturan biasanya berbentuk *IF – THEN*. Aturan digunakan sebagai prosedur untuk memecahkan permasalahan.

6. Kemampuan Penjelasan (*Explanation Capability*)

Kemampuan menjelaskan merupakan komponen tambahan dari sistem pakar yang berfungsi untuk memberikan penjelasan kepada pengguna mengapa suatu pertanyaan ditanyakan oleh sistem pakar, bagaimana kesimpulan dapat diperoleh, kenapa solusi tertentu ditolak, dan apa rencananya untuk mencapai solusi.

Sistem pakar disusun oleh 2 bagian utama, yaitu lingkungan pengembangan dan lingkungan konsultasi. Lingkungan pengembangan digunakan sebagai pembangun sistem pakar baik dari segi pembangun, maupun basis pengetahuan. Lingkungan konsultasi digunakan oleh seseorang yang bukan pakar untuk memperoleh pengetahuan pakar.



Gambar 2.2 Struktur Sistem Pakar
Sumber : Kusumadewi, 2003

Sistem pakar yang baik harus memiliki fasilitas informasi yang handal, mudah dimodifikasi, dapat digunakan dalam berbagai jenis komputer, dan memiliki kemampuan untuk belajar beradaptasi.

2.2 Makanan Sehat

Makanan sehat adalah dengan meramu berbagai jenis makanan yang seimbang sehingga terpenuhi seluruh kebutuhan gizi bagi tubuh dan mampu dirasakan secara fisik dan mental (Prasetyono, 2009). Tujuan memakan makanan sehat bagi tubuh manusia adalah untuk menjaga agar tubuh tetap sehat, tumbuh, dan berkembang secara baik.

Makanan yang sehat harus terdiri dari makanan utama dan makanan penunjang. Makanan sehat tersebut juga dikenal dengan istilah 4 (empat) dan 5 (lima) sempurna, tetapi kepopulerannya sudah mulai memudar karena berbagai alasan. Makanan sehat menurut para ahli gizi mengandung 4 macam makanan,

yaitu makanan pokok, lauk pauk, sayur, dan buah. Zat-zat gizi yang terkandung dalam makanan sehat adalah (Almatsier, 2011) :

1. Protein

Protein dalam tubuh merupakan asam amino esensial yang diperlukan sebagai zat pembangun, yaitu untuk (Mangunkusumo, 2008) :

- Pertumbuhan dan pembentukan protein dalam serum, hemoglobin, enzim, hormon, dan antibodi.
- Memelihara keseimbangan asam basa cairan tubuh.
- Menggantikan sel-sel yang rusak.
- Sumber energi.

2. Mineral

Mineral merupakan bagian dari tubuh dan memegang peranan penting dalam pemeliharaan fungsi tubuh, baik pada tingkat sel, jaringan, organ, maupun fungsi tubuh secara keseluruhan. Dalam ketersediaannya, tidak semua bahan makanan mengandung mineral yang bisa digunakan untuk keperluan tubuh, dan dimanfaatkan oleh tubuh. Fungsi umum mineral adalah memelihara keseimbangan asam basa tubuh dan mengkatalisasi reaksi dalam pemecahan karbohidrat, lemak, dan protein.

3. Vitamin

Vitamin adalah zat organik yang kompleks yang dibutuhkan dalam jumlah sangat kecil dan umumnya tidak dapat dibentuk oleh tubuh. Oleh karena itu harus didatangkan dari makanan. Vitamin termasuk kelompok zat pengatur pertumbuhan dan pemeliharaan kehidupan. Tiap vitamin mempunyai tugas

spesifik di dalam tubuh. Vitamin dapat rusak karena penyimpanan dan pengolahan. (Sunita, 2007).

4. Lemak

Lemak merupakan zat padat energi, dimana kandungannya dua kali lipat dari karbohidrat dan protein. Makanan yang mengandung lemak adalah daging berlemak, margarin, minyak goreng, jeroan, keju, dan lain-lain. Lemak mempunyai fungsi membentuk jaringan tubuh dan sebagai pelarut vitamin A,D,E, K.

Makanan sehat dan bergizi seimbang bukan berarti makanan yang mahal dan enak. Ada beberapa makanan yang dianggap tidak sehat oleh *World Health Organization* (WHO), yaitu gorengan, makanan kalengan, makanan asinan, makanan daging yang diproses (ham, sosis), makanan dari daging yang berlemak dan jeroan, olahan keju, mie instan, makanan yang dibakar atau dipanggang, sajian manis beku, dan makanan kering.

2.3 Kalori

Kalori merupakan satuan yang digunakan untuk menyatakan jumlah energi (Boyle, 2010). Pada umumnya, kalori digunakan untuk menunjukkan jumlah energi yang terkandung dalam makanan. Kalori dapat diperoleh dari asupan gizi yang mengandung nutrisi, seperti karbohidrat, lemak, protein, dan alkohol.

Jumlah kalori dalam makanan diperlukan untuk memperhitungkan keseimbangan energi. Apabila jumlah kalori yang dikonsumsi lebih kecil dari kalori yang digunakan, berat badan akan berkurang karena cadangan energi dari

lemak akan digunakan. Sebaliknya, apabila jumlah kalori yang masuk lebih besar dari kalori yang digunakan, berat badan akan meningkat. Kelebihan energi akan disimpan sebagai lemak. Adapun penumpukan lemak yang berlebihan dapat meningkatkan resiko terjadinya hipertensi, obesitas, penyakit jantung, *stroke*, dan diabetes. Karena itu, asupan kalori perlu dikontrol untuk menjaga berat badan dan mencegah terjadinya penyakit metabolik.

Tubuh membutuhkan energi (yang disebut kalori) dalam melakukan aktivitas sehari-hari. Pembagian kalori yang seimbang mengacu pada Piramida Makanan. Gambar di bawah ini menunjukkan pola konsumsi umum dengan Piramida Makanan yang mengacu pada Departemen Kesehatan RI.



Gambar 2.3 Pola Konsumsi Umum Piramida Makanan
Sumber : BPOM, 2007

Kebutuhan kalori tiap individu berbeda-beda. Untuk menghitung kebutuhan kalori (kalori yang dibutuhkan untuk kegiatan sehari-hari) dapat ditentukan dengan cara menghitung *Basal Metabolic Rate* (BMR). Sebelum

menghitung kebutuhan BMR, harus dihitung dahulu berat badan ideal. Untuk menentukan berat badan ideal seseorang, ada 2 cara sederhana, yaitu :

- Cara 1 : menggunakan rumus Broca

$$BBI = (TB - 100) - (0.1 \times (TB - 100)) \dots \dots \dots \text{Rumus 2.1}$$

Keterangan : TB = Tinggi Badan (cm)

- Cara 2 : menggunakan *Body Mass Index* (BMI)

$$BMI = \frac{BB}{TB \times TB} \dots \dots \dots \text{Rumus 2.2}$$

Keterangan : BB = Berat Badan (kg) TB = Tinggi Badan (m)

Tabel 2.1 Tabel Klasifikasi Internasional berdasarkan BMI
Sumber : WHO, 2013

Underweight	< 18.50	< 18.50
Severe thinness	< 16.00	< 16.00
Moderate thinness	16.00 - 16.99	16.00 - 16.99
Mild thinness	17.00 - 18.49	17.00 - 18.49
Normal range	18.50 - 24.99	18.50 - 22.99
		23.00 - 24.99
Overweight	≥ 25.00	≥ 25.00
Pre-obese	25.00 - 29.99	25.00 - 27.49
		27.50 - 29.99
Obese	≥ 30.00	≥ 30.00
Obese class I	30.00 - 34.99	30.00 - 32.49
		32.50 - 34.99
Obese class II	35.00 - 39.99	35.00 - 37.49
		37.50 - 39.99
Obese class III	≥ 40.00	≥ 40.00

Setelah menghitung berat badan ideal, hitunglah kalori basal sebelum menghitung BMR yang dibutuhkan dengan cara :

- Pria = BBI x 30 kkal.....Rumus 2.3
- Wanita = BBI x 25 kkal..... Rumus2.4

Lalu tambahkan dengan faktor aktivitas dan stress pada kebutuhan kalori basal :

- Tambahkan 10 – 20 % pada aktivitas ringan
- Tambahkan 20 – 30 % pada aktivitas sedang
- Tambahkan 30 – 40 % pada aktivitas berat

Tabel 2.2 Tabel Klasifikasi Aktivitas Harian
Sumber : Prof.Dr. Ir. C. Hanny Wijaya

Ringan (tambahkan 10–20%)	Sedang (tambahkan 20-30%)	Berat (tambahkan 30–40%)
Menyetir (10%)	Kerja rumah (20%)	Aerobik (40%)
Kerja kantoran (10%)	Bersepeda (30%)	Bersepeda mendaki (40%)
Mengajar (20%)	Bowling (30%)	Dansa (40%)
Berjalan (20%)	Berjalan cepat (30%)	Jogging (40%)
Memancing (20%)	Berkebun (30%)	Panjat tebing (50%)
Membaca (20%)		Atlit (50%)

Setelah itu, kurangi perhitungan kalori basal pada kondisi kelebihan berat badan (BB) dan disesuaikan dengan usia.

Tabel 2.3 Tabel Penyesuaian Perhitungan Kalori
Sumber : Prof.Dr. Ir. C. Hanny Wijaya

Kondisi dan Umur	Koreksi
40 – 59 tahun	-5%
60 – 69 tahun	-10%
> 70 tahun	-20%
Kelebihan Berat Badan	- 20 sampai -30%
Kekurangan Berat Badan	+ 20 sampai +30%

Setelah melakukan langkah-langkah diatas, maka total kebutuhan kalori harian dapat dihitung dengan rumus 2.5.

$$\text{Total Kebutuhan Kalori Harian} = \text{KKB} + \text{KFA} + \text{KFU} \dots\dots\dots \text{Rumus 2.5}$$

Keterangan : KKB = Kebutuhan Kalori Basal

KFA = Koreksi Faktor Aktivitas

KFU = Koreksi Faktor Umur

2.4 Golongan Darah

Golongan darah adalah pengklasifikasian darah dari suatu individu berdasarkan ada atau tidaknya zat antigen warisan pada permukaan membran sel darah merah. Golongan darah manusia dibagi menjadi beberapa macam. Hal ini dapat dilihat dari *aglutinogen* (antigen) dan *aglutinin* (antibodi) yang terkandung dalam darah seseorang.

Penggolongan darah ini pertama kali ditemukan oleh Dr. Lendsteiner dan Donath. Penemuan Karl Landsteiner diawali dari penelitiannya, yaitu ketika *eritrosit* seseorang dicampur dengan serum darah orang lain, maka terjadi penggumpalan (*aglutinasi*). Tetapi pada orang lain, campuran itu tidak menyebabkan penggumpalan darah. *Aglutinogen* yang terdapat pada *eritrosit* orang tertentu dapat bereaksi dengan zat *aglutinin* yang terdapat pada serum darah. *Aglutinogen* dibedakan menjadi dua yaitu:

1. *Aglutinogen A* : memiliki enzim *glukosil transferase* yang mengandung *glutiasetil glukosamin* pada rangka *glikoprotein*.

2. *Aglutinogen B* : memiliki enzim *galaktose* pada rangka *glikoprotein*. *Aglutinin* dibedakan menjadi *aglutinin α* dan *β* .

Darah seseorang dapat mengandung *aglutinogen A* saja, *aglutinogen B* saja, *aglutinogen A* dan *B*, dan tidak mengandung *aglutinogen* sama sekali. Adanya *aglutinogen* dan *aglutinin* inilah yang menjadi dasar penggolongan darah manusia berdasarkan sistem ABO. Menurut sistem ABO, golongan darah manusia dibedakan menjadi empat, yaitu sebagai berikut.

Tabel 2.4 Tabel Golongan Darah Manusia
Sumber : Purnomo, 2009

Golongan Darah	Keterangan
A	Apabila di dalam sel darah seseorang mengandung <i>aglutinogen A</i> dan serumnya mengandung <i>aglutinin β</i> sehingga dapat dirumuskan (A, β).
B	Apabila di dalam sel darah seseorang terdapat <i>aglutinogen B</i> , sedangkan dalam serumnya terdapat <i>aglutinin α</i> sehingga dirumuskan (B, α)
AB	Apabila di dalam sel darah seseorang terdapat <i>aglutinogen A</i> dan <i>B</i> , sedangkan di dalam serumnya tidak mengandung <i>aglutinin</i> , sehingga dapat dirumuskan (AB,-)
O	Apabila di dalam sel darah seseorang tidak terdapat <i>aglutinogen</i> , sedangkan dalam serumnya mengandung <i>aglutinin α</i> dan <i>β</i> sehingga dapat dirumuskan (-, α, β).

Pada penelitiannya, Leindsteiner juga menemukan *aglutinogen* yang terdapat pada darah kera, *Maccacus rhesus*, sehingga diberi nama *aglutinogen rhesus*. Dari fakta ini, kemudian golongan darah dibedakan menjadi dua, yaitu sebagai berikut.

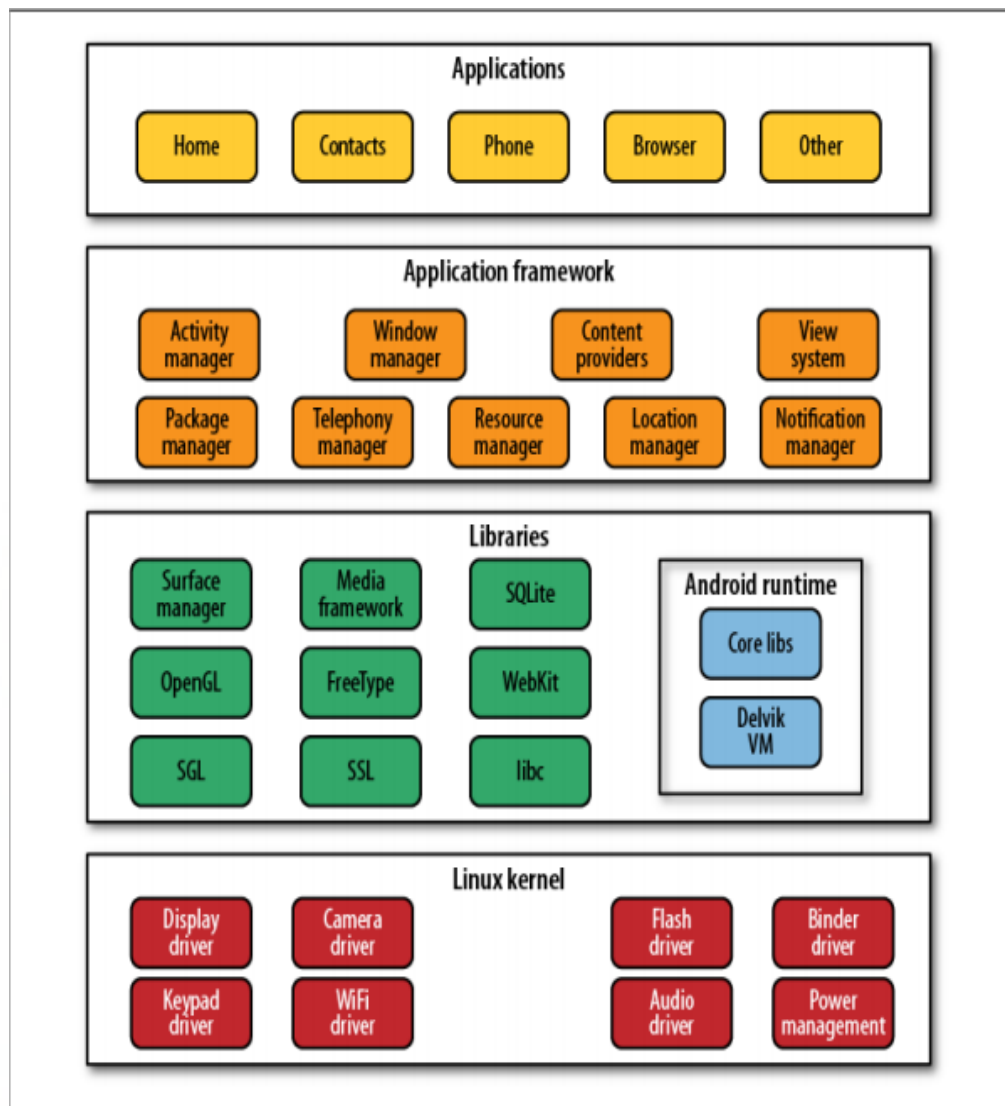
1. Golongan darah Rh+, jika di dalam sel darah seseorang terdapat *aglutinogen rhesus*.
2. Golongan darah Rh-, jika di dalam sel darah seseorang tidak terdapat *aglutinogen rhesus*.

Sistem *rhesus* ini dalam tranfusi darah harus diperhatikan. Apabila golongan darah Rh+, maka tidak boleh digunakan sebagai donor untuk golongan darah Rh-, karena bisa terjadi *aglutinasi* (penggumpalan).

2.5 Android

Android merupakan *subset* perangkat lunak untuk perangkat mobile yang meliputi sistem operasi, *middleware* dan aplikasi inti yang dikeluarkan oleh Google, sedangkan Android SDK (*Software Development Kit*) menyediakan *tools* dan API (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi pada *platform* Android dengan menggunakan bahasa pemrograman Java.

Android pertama kali dikembangkan oleh perusahaan bernama Android Inc., dan pada tahun 2005 dibeli oleh raksasa internet Google. Android bersifat *open source*. Semua susunannya, mulai dari *low-level* Linux modul sampai dengan *native libraries*, dan *application framework* sampai *complete application*, semuanya benar-benar terbuka (Gargenta, 2011).



Gambar 2.4 Android Platform Architecture
Sumber : Gargenta, 2011

Android memiliki komunitas developer yang besar untuk pengembangan aplikasi. Hal ini membuat fungsi Android menjadi lebih luas dan beragam. *Android Market* (sekarang *Google Play*) merupakan tempat mengunduh (*download*) aplikasi Android baik gratis ataupun berbayar yang dikelola oleh Google. Android sudah memiliki banyak versi. Untuk setiap versi android diberi kode nama berdasarkan nama hidangan makanan.

Tabel 2.5 Android Versions
 Sumber : Android Developer, 2013

Version	Code Name	API
1.6	Donut	4
2.1	Éclair	7
2.2	Froyo	8
2.3 – 2.3.2	Gingerbread	9
2.3.3 – 2.3.7	Gingerbread	10
3.2	Honeycomb	13
4.0.3 – 4.0.4	Ice Cream Sandwich	15
4.1	Jelly Bean	16
4.2	Jelly Bean	17

2.6 Fuzzy Logic

Logika *Fuzzy* (*Fuzzy Logic*) adalah suatu cara yang tepat untuk memetakan suatu ruang *input* ke dalam ruang *output* (Kusumadewi, 2010). Sistem *Fuzzy* pertama kali diperkenalkan oleh Prof. L. A. Zadeh dari Barkelay pada tahun 1965, dimana Zadeh memperkenalkan teori yang memiliki objek-objek dari himpunan *fuzzy* yang memiliki batasan yang tidak presisi dan keanggotaannya dalam himpunan *fuzzy*, dan bukan dalam bentuk logika benar (*true*) atau salah (*false*), tapi dinyatakan dalam derajat (*degree*). Konsep seperti ini disebut dengan *Fuzziness* dan teorinya dinamakan *Fuzzy Set Theory*.

Sistem Fuzzy merupakan penduga numerik yang terstruktur dan dinamis. Untuk sistem yang sangat rumit, penggunaan *Fuzzy Logic* adalah salah satu pemecahannya. Sistem tradisional dirancang untuk mengontrol keluaran tunggal yang berasal dari beberapa masukan yang tidak saling berhubungan. Karena

ketidaktergantungan ini, penambahan masukan yang baru akan merumitkan proses kontrol dan membutuhkan proses perhitungan kembali dari semua fungsi. Kebalikannya, penambahan masukan baru pada sistem *fuzzy*, yaitu sistem yang bekerja berdasarkan prinsip-prinsip logika *fuzzy*, hanya membutuhkan penambahan fungsi keanggotaan yang baru dan aturan-aturan yang berhubungan dengannya.

Secara umum, sistem *fuzzy* sangat cocok untuk penalaran pendekatan, terutama untuk sistem yang menangani masalah-masalah yang sulit didefinisikan dengan menggunakan model matematis. Misalkan, nilai masukan dan parameter sebuah sistem bersifat kurang akurat atau kurang jelas sehingga sulit mendefinisikan model matematikanya.

Sistem *fuzzy* mempunyai beberapa keuntungan bila dibandingkan dengan sistem tradisional, misalkan pada jumlah aturan yang digunakan. Keuntungan lainnya adalah sistem *fuzzy* mempunyai kemampuan penalaran yang mirip dengan kemampuan penalaran manusia. Hal ini disebabkan karena sistem *fuzzy* mempunyai kemampuan untuk memberikan respon berdasarkan informasi yang bersifat kualitatif, tidak akurat, dan ambigu. Ada beberapa alasan penggunaan Logika *Fuzzy* (Kusumadewi, 2010) :

1. Logika *Fuzzy* sangat fleksibel.
2. Logika *Fuzzy* memiliki toleransi.
3. Konsep logika *Fuzzy* mudah dimengerti. Konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana dan mudah dimengerti.

4. Logika *Fuzzy* mampu memodelkan fungsi-fungsi nonlinear yang sangat kompleks.
5. Logika *Fuzzy* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. Logika *Fuzzy* dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
7. Logika *Fuzzy* didasarkan pada bahasa alami.

Ada beberapa metode untuk merepresentasikan hasil logika *fuzzy* yaitu metode Tsukamoto, Sugeno dan Mamdani. Pada metode Tsukamoto, setiap konsekuen direpresentasikan dengan himpunan *fuzzy* dengan fungsi keanggotaan monoton. *Output* hasil inferensi masing-masing aturan adalah z , berupa himpunan biasa (*crisp*) yang ditetapkan berdasarkan predikatnya. Hasil akhir diperoleh dengan menggunakan rata-rata terbobotnya. (Kusumadewi, 2002).

Metode Sugeno mirip dengan metode Mamdani, hanya *output* tidak berupa himpunan *fuzzy*, melainkan berupa konstanta atau persamaan linear. Ada dua model metode Sugeno yaitu model *fuzzy* sugeno orde nol dan model *fuzzy* sugeno orde satu. Bentuk umum model *fuzzy* sugeno orde nol adalah :

$$IF (x1 \text{ is } A1) \text{ o } (x2 \text{ is } A2) \text{ o } \dots \text{ o } (xn \text{ is } An) THEN z = k$$

Bentuk umum model *fuzzy* Sugeno orde satu adalah :

$$IF (x1 \text{ is } A1) \text{ o } (x2 \text{ is } A2) \text{ o } \dots \text{ o } (xn \text{ is } An) THEN z = p1.x1 + \dots + pn.xn + q$$

Pada metode Mamdani, aplikasi fungsi implikasi menggunakan *MIN*, sedang komposisi aturan menggunakan metode *MAX*. Metode Mamdani dikenal juga dengan metode *MAX-MIN*. Inferensi *output* yang dihasilkan berupa bilangan

fuzzy maka harus ditentukan suatu nilai crisp tertentu sebagai *output*. Proses ini dikenal dengan *defuzzifikasi*. Ada beberapa tahapan untuk mendapatkan *output* yaitu:

1. Pembentukan himpunan *fuzzy*

Pada metode Mamdani, baik variabel *input* maupun variabel *output* dibagi menjadi satu atau lebih himpunan *fuzzy*.

2. Aplikasi fungsi implikasi

Pada metode Mamdani, fungsi implikasi yang digunakan adalah *Min*.

3. Komposisi Aturan

Tidak seperti penalaran monoton, apabila sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari kumpulan dan korelasi antar aturan. Ada 3 metode yang digunakan dalam melakukan inferensi sistem *fuzzy* yaitu : *Max*, *Additive*, dan Probabilistik *OR*.

a. Metode *Max (Maximum)*

Pada metode ini solusi himpunan *fuzzy* diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerah *fuzzy* dan mengaplikasikan ke *output* dengan menggunakan operator *OR (union)*. Jika semua proposisi telah dievaluasi, maka *output* akan berisi suatu himpunan *fuzzy* yang merefleksikan kontribusi dari tiap-tiap proposisi. Secara umum dapat dituliskan :

$$\mu_{sf}[xi] \leftarrow \max (\mu_{sf}[xi] , \mu_{kf}[xi]) \dots \dots \dots \text{Rumus 2.6}$$

dengan :

$\mu_{sf}[xi]$ = nilai keanggotaan solusi *fuzzy* sampai aturan ke-i

$\mu_{kf}[x_i]$ =nilai keanggotaan konsekuen *fuzzy* aturan ke-i

b. Metode *Additive (Sum)*

Solusi himpunan *fuzzy* diperoleh dengan cara melakukan *bounded-sum* terhadap semua *output* daerah *fuzzy*. Secara umum dituliskan :

$$\mu_{sf}[x_i] \leftarrow \max (1, \mu_{sf}[x_i] + \mu_{kf}[x_i]) \dots \dots \dots \text{Rumus 2.7}$$

dengan :

$\mu_{sf}[x_i]$ =nilai keanggotaan solusi *fuzzy* sampai aturan ke-i

$\mu_{kf}[x_i]$ =nilai keanggotaan konsekuen *fuzzy* aturan ke-i

c. Metode Probabilistik *OR*

Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara melakukan *product* terhadap semua *output* daerah *fuzzy*. Secara umum dituliskan :

$$\mu_{sf}[x_i] \leftarrow \max (\mu_{sf}[x_i] + \mu_{kf}[x_i]) - (\mu_{sf}[x_i] * \mu_{kf}[x_i]) \dots \dots \dots \text{Rumus 2.8}$$

dengan :

$\mu_{sf}[x_i]$ =nilai keanggotaan solusi *fuzzy* sampai aturan ke-i

$\mu_{kf}[x_i]$ =nilai keanggotaan konsekuen *fuzzy* aturan ke-i

4. Penegasan (*Defuzzifikasi*)

Input dari proses *defuzzifikasi* adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada domain himpunan *fuzzy* tersebut sehingga jika diberikan suatu himpunan *fuzzy* dalam *range* tertentu, maka harus dapat diambil suatu nilai *crisp* tertentu sebagai *output*. Ada beberapa metoda yang dipakai dalam *defuzzifikasi*:

a. Metode *Centroid*

Pada metode ini, penetapan nilai *crisp* dengan cara mengambil titik pusat daerah *fuzzy*.

b. Metode *Bisektor*

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai pada domain *fuzzy* yang memiliki nilai keanggotaan seperti dari jumlah total nilai keanggotaan pada daerah *fuzzy*.

c. Metode *Means of Maximum* (MOM)

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum.

d. Metode *Largest of Maximum* (LOM)

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai terbesar dari domain yang memiliki nilai keanggotaan maksimum.

e. Metode *Smallest of Maksimum* (SOM)

Solusi *crisp* diperoleh dengan cara mengambil nilai terkecil dari domain yang memiliki nilai keanggotaan maksimum.

2.6.1 Himpunan Fuzzy

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu *item* x dalam suatu himpunan A , yang sering ditulis dengan $\mu_A[x]$, memiliki 2 (dua) kemungkinan (Kusumadewi, 2010) yaitu:

1. Satu (1), yang berarti bahwa suatu *item* menjadi anggota dalam suatu himpunan, atau
2. Nol (0), yang berarti bahwa suatu *item* tidak menjadi anggota dalam suatu himpunan.

Terkadang kemiripan antara keanggotaan *fuzzy* dengan probabilitas menimbulkan kerancuan. Keduanya memiliki nilai pada interval $[0,1]$, tetapi interpretasi nilainya sangat berbeda antara kedua kasus tersebut. Keanggotaan *fuzzy* memberikan suatu ukuran terhadap pendapat atau keputusan, sedangkan probabilitas mengindikasikan proporsi terhadap keseringan suatu hasil bernilai benar dalam jangka panjang. Misalnya, jika nilai keanggotaan bernilai suatu himpunan *fuzzy* USIA adalah 0.9, maka tidak perlu dipermasalahkan berapa seringnya nilai itu diulang secara individual untuk mengharapkan suatu hasil yang hampir pasti muda. Di lain pihak, nilai probabilitas 0.9 usia berarti 10% dari himpunan tersebut diharapkan tidak muda. Himpunan fuzzy memiliki 2 atribut, yaitu:

1. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti: MUDA, PAROBAYA, TUA.
2. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variable seperti: 40, 25, 50, dan seterusnya.

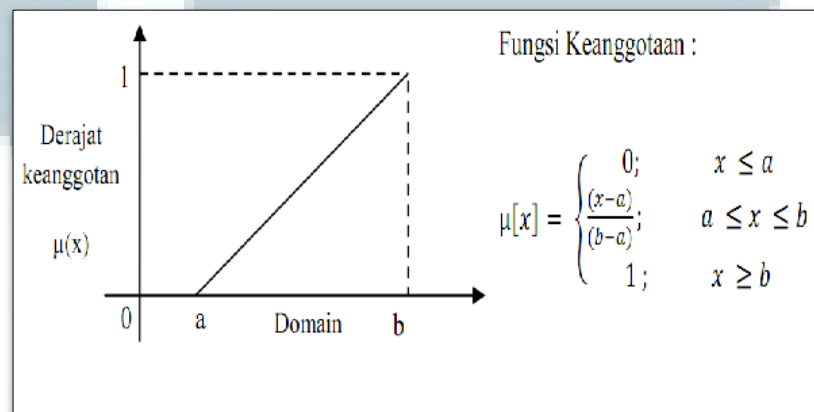
2.6.2 Fungsi Keanggotaan

Fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data kedalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah melalui pendekatan fungsi. Apabila U menyatakan himpunan universal dan A adalah himpunan fungsi

fuzzy dalam U , maka A dapat dinyatakan sebagai pasangan terurut. Ada beberapa fungsi yang bisa digunakan.

1. Representasi Linear

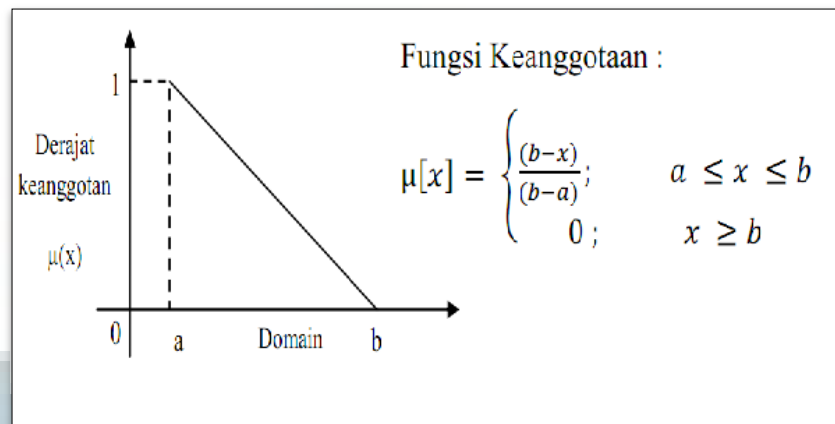
Pada representasi linear, pemetaan *input* ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas. Ada 2 keadaan himpunan *fuzzy* yang linear. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol (0) bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi (Kusumadewi, 2010). Seperti terlihat pada gambar 2.5.



Gambar 2.5 Representasi linear naik

Sumber : Kusumadewi, 2010

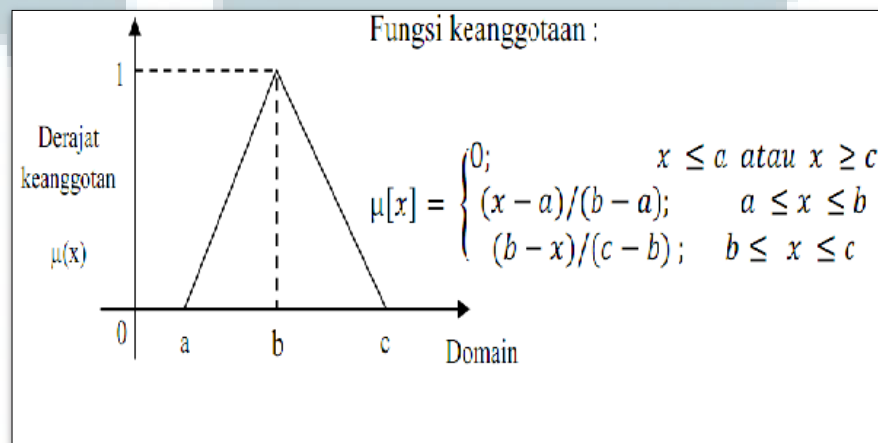
Kedua, merupakan kebalikan dari yang pertama. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah. Seperti terlihat pada gambar 2.6.



Gambar 2.6 Representasi linear turun
Sumber : Kusumadewi, 2010

2. Representasi Kurva Segitiga

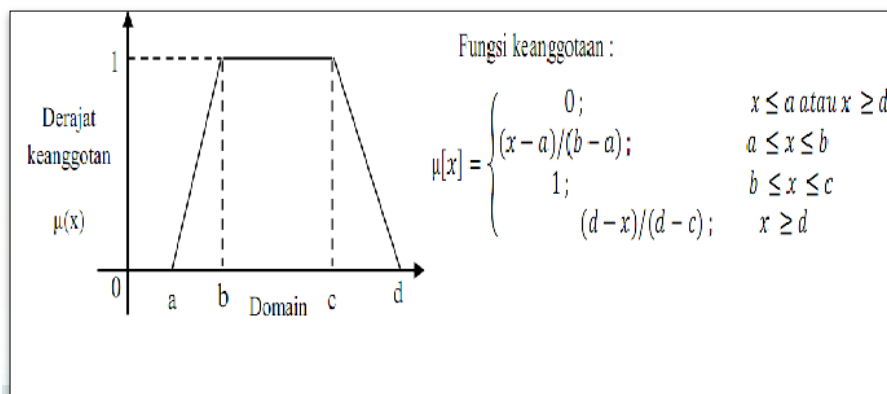
Kurva segitiga pada dasarnya merupakan gabungan antara 2 (dua) garis (linear). Seperti terlihat pada gambar 2.7.



Gambar 2.7 Representasi kurva segitiga
Sumber : Kusumadewi, 2010

3. Representasi kurva trapezium

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada titik yang memiliki nilai keanggotaan 1. Seperti terlihat pada Gambar 2.8.



Gambar 2.8 Representasi kurva trapesium
Sumber : Kusumadewi, 2010

2.6.3 Rule IF – THEN

Rule adalah sebuah struktur *knowledge* yang menghubungkan beberapa informasi yang sudah diketahui ke informasi lain sehingga dapat disimpulkan. Sebuah *rule* adalah sebuah bentuk *knowledge* yang *procedural*. Dengan demikian, yang dimaksud dengan sistem pakar berbasis *rule* adalah sebuah program komputer untuk memproses masalah dari informasi spesifik yang terdapat dalam memori aktif dengan sebuah set dari *rule* dalam *knowledge base*, dengan menggunakan *inference engine* untuk menghasilkan informasi baru.

Struktur *rule* secara logika menghubungkan satu atau lebih premis yang terletak dalam bagian IF dengan satu atau lebih konsekuen yang terletak dalam bagian THEN. Secara umum, sebuah *rule* dapat memiliki premis jamak dihubungkan dengan pernyataan AND (konjungsi) pernyataan OR (disjungsi) atau kombinasi dari keduanya.

Dalam sistem pakar berbasis *rule* domain *knowledge* ditampung dalam sebuah set dari *rules* dan dimasukkan dalam basis sistem pengetahuan. Sistem menggunakan aturan ini dengan informasi selama berada dalam memori aktif

untuk memecahkan masalah. Sistem pakar berbasis *rule* mempunyai arsitektur yang dapat dijelaskan sebagai berikut.

1. *User interface*

Digunakan sebagai media oleh *user* untuk melihat dan berinteraksi dengan sistem.

2. *Developer interface*

Media yang digunakan untuk mengembangkan sistem oleh *engineer*.

3. Fasilitas penjelasan

Sub sistem yang berfungsi untuk menyediakan penjelasan dalam sistem reasoning.

4. Program eksternal

Program seperti *database*, *spreadsheet*, yang bekerja dalam mendukung keseluruhan sistem.

UMMN