

BAB II

KAJIAN PUSTAKA DAN DASAR TEORI

2.1 Kajian Pustaka

Terdapat beberapa penelitian dalam pendiagnosaan awal penyakit asma, diantaranya adalah penelitian dengan judul "Aplikasi Diagnosa Penyakit Asma Menggunakan *Bayesian Network* Berbasis Web". Penelitian tersebut menggunakan 4 atribut pengelompokan jenis penyakit asma yaitu asma intermiten (AI), asma persisten ringan (APR), asma persisten sedang (APS), asma persisten berat (APB) menurut buku Panduan Pelayanan Medik Perhimpunan Dokter Spesialis Penyakit Dalam Indonesia dengan menggunakan metode Naives Bayes. Penelitian tersebut menghasilkan akurasi cukup baik yaitu sebesar 83,33% karena jenis dari analisa dokter tidak merata, dokter menganalisa dengan 2 atau 3 jenis penyakit sementara sistem menghasilkan 4 jenis penyakit (YUN-2012).

Pada penelitian ini akan digunakan algoritma ID3 dan sistem *Case Based Reasoning* (CBR) yang dianggap cocok untuk diterapkan pada kasus pendiagnosaan awal penyakit asma. Karena sistem *Case Based Reasoning* (CBR) mampu menyimpan pengetahuan baru pada ID3 sehingga mampu meningkatkan kualitas solusi yang diperoleh. Pada penelitian sebelumnya dengan judul "Analisis dan Implementasi Sistem Pendiagnosis Penyakit Tuberculosis Menggunakan Metode *Case Based Reasoning* (CBR)" yang dilakukan oleh Bimmo menggunakan metode *naive bayes* dengan hasil akurasi yang cukup baik yaitu rata – rata lebih dari 85% dari jumlah data sebesar 120 data record menggunakan 34 atribut sebagai parameternya (BIM-14).

Pada penelitian ini algoritma ID3 akan digunakan untuk menghasilkan pohon keputusan yang berisi aturan – aturan (*rule*) untuk mendiagnosa jenis penyakit asma. Selanjutnya aturan yang ada akan disimpan dalam *knowledge base* untuk selanjutnya digunakan ketika ada kasus baru yang mirip dengan kasus lama, namun jika kasus baru tidak ada kecocokan maka solusi dari permasalahan tersebut akan menjadi pengetahuan baru yang akan disimpan di *case base* yang

selanjutnya akan digunakan untuk menyelesaikan kasus baru yang lain di masa mendatang. Penelitian ini menggunakan 100 data sampel dengan 6 atribut yaitu batuk, dahak, darah, sesak napas, nyeri dada, dan demam. Masing – masing atribut memiliki dua nilai untuk masing – masing atribut yaitu ya dan tidak.

Pada penelitian sebelumnya dengan judul “Implementasi Sistem Pengambilan Keputusan Untuk Menentukan Tingkat Penyakit Periodentitis Menggunakan Pohon Keputusan”, metode yang digunakan adalah metode ID3 untuk membentuk pohon keputusan. Penelitian tersebut menggunakan 11 atribut yang terdiri dari poket, hiperplasi, resesi, perdarahan, peradangan, kalkulus, plak, gigi, kegoyangan, vitalitas, migrasi dan malposisi. Jenis tingkatan penyakit periodentitis dibedakan menjadi beberapa tingkatan yaitu 0, 1, 2, 3 dan 4. Pengujian yang dilakukan menghasilkan akurasi yang cukup baik dengan akurasi tertinggi sebesar 96% dan akurasi terendah sebesar 90,62% (HAR-11).

2.2 DasarTeori

2.2.1 Penyakit Asma

Asma adalah suatu kondisi paru-paru kronis yang ditandai dengan sulit bernapas. Saluran pernapasan penderita asma sangat sensitif dan memberi respons yang sangat berlebihan jika mengalami rangsangan atau gangguan. Saluran pernapasan tersebut bereaksi dengan cara menyempit dan menghalangi udara yang masuk. Penyempitan atau hambatan ini bisa mengakibatkan salah satu atau gabungan dari berbagai gejala mulai dari batuk, sesak, napas pendek, tersengal-sengal, hingga napas yang berbunyi “ngik-ngik”.

Dalam penelitian ini terdapat dua jenis penyakit asma yang dijadikan acuan yaitu:

2.2.1.1 Asma Bronkial

Asma bronkial adalah suatu penyakit dengan ciri - ciri meningkatnya respon trakea dan bronkus terhadap berbagai rangsangan dengan manifestasi adanya penyempitan jalan nafas yang luas dan derajatnya dapat berubah-ubah baik secara spontan maupun hasil dari pengobatan (TAN-03).

Kelainan dasar penyempitan saluran pernapasan yang berakibat timbulnya sesak napas adalah gabungan dari keadaan berikut :

- Kejang/berkerutnya otot polos dari saluran pernapasan
- Sembab/pembengkakan selaput lendir
- Proses peradangan
- Pembentukan dan timbunan lendir yang berlebihan dalam rongga saluran pernapasan.

2.2.1.2 Bronkitis Akut

Bronkitis akut merupakan penyakit saluran napas yang sering ditemukan di masyarakat. Penyakit ini menjadi masalah kesehatan karena sifatnya yang kronis dan persisten serta progresif. Infeksi saluran nafas merupakan masalah klinis yang sering dijumpai pada penderita bronkitis akut yang dapat memperberat penyakitnya. Eksaserbasi infeksi akut akan mempercepat kerusakan yang telah terjadi, disamping itu kuman yang menyebabkan eksaserbasi juga berpengaruh terhadap mortalitas dan morbiditas penyakit ini. Semakin sering terjadi eksaserbasi, maka mortalitas dan morbiditas penyakit ini akan semakin meningkat (SOE-04).

Tanda dan gejala bronkitis akut yang mungkin timbul:

- Batuk disertai dahak (lendir)
- Demam
- Sakit kepala ringan
- Panas dingin atau menggigil
- Sesak nafas (PED-13)

Berdasarkan data rekam medis pasien yang didapatkan maka parameter yang digunakan dalam penelitian ini, yaitu:

- a. Batuk
- b. Dahak
- c. Darah
- d. Sesak napas
- e. Nyeri dada
- f. Demam

2.2.2 Klasifikasi

Klasifikasi adalah pengelompokan data ke dalam kelas-kelas yang sudah ditentukan sebelumnya. Klasifikasi dapat juga didefinisikan sebagai proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data. Dengan tujuan untuk memperkirakan kelas dari suatu objek yang labelnya tidak diketahui (PRM-03). Sedangkan menurut Moertini, klasifikasi dapat dikatakan sebagai fungsi pembelajaran yang memetakan sebuah unsur (item) data ke dalam salah satu dari beberapa kelas yang sudah didefinisikan sebelumnya (MOE-02).

Sasaran permasalahan-permasalahan klasifikasi ialah untuk mengidentifikasi karakteristik yang menunjukkan pada kelompok mana setiap kasus ditempatkan. Hal ini dapat digunakan untuk memahami data yang sudah ada, atau untuk memprediksi data yang baru. Proses klasifikasi pada *decision tree* biasanya dibagi menjadi dua fase yaitu *Learning* dan *Test*. Fase *learning* yaitu sebagian data yang sudah diketahui kelasnya digunakan untuk membentuk aturan *tree*. Sedangkan pada fase *test* aturan yang telah dibentuk digunakan untuk menganalisa data yang baru dan menguji tingkat akurasi datanya. Bila tingkat akurasinya terpenuhi, maka model tersebut dapat digunakan untuk memprediksi data yang baru.

2.2.3 Pohon (Tree)

Graf terhubung yang tidak mengandung sirkuit disebut pohon (*tree*). Dalam kehidupan sehari-hari, orang telah lama menggunakan *tree* untuk menggambarkan hirarki. Misalnya, *tree* silsilah keluarga, struktur organisasi, organisasi pertandingan, dan lain-lain. Para ahli bahasa menggunakan *tree* untuk menguraikan kalimat, yang disebut pohon parsing (*parse tree*).

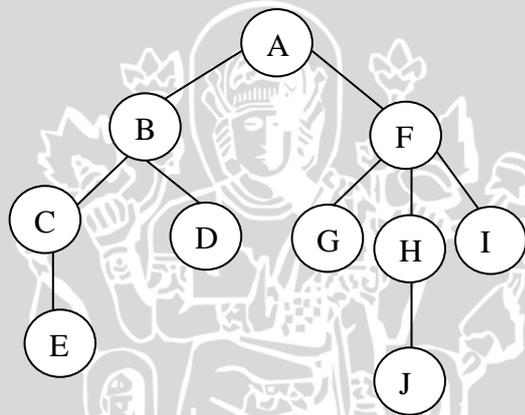
Tree sudah lama digunakan sejak tahun 1857, ketika matematikawan asal Inggris Arthur Cayley menggunakan pohon untuk menghitung jumlah senyawa kimia.

2.2.3.1 Pengertian pohon (*tree*)

Pohon atau *tree* adalah salah satu bentuk *graph* terhubung yang tidak mengandung sirkuit (MUN-06). Karena merupakan *graph* terhubung. Maka pada *tree* selalu terdapat *path* atau jalur yang menghubungkan setiap dua simpul dalam *tree*. Kali ini sampai pada pembahasan suatu bentuk *tree*, yang dilengkapi dengan apa yang disebut akar atau *root*. Seperti yang terdapat pada gambar 2.4, *tree* semacam itu disebut pohon berakar.

2.2.3.2 Istilah-istilah dasar pada pohon (*tree*)

Pada Gambar 2.1 akan dijelaskan beberapa istilah dasar yang mendukung terbentuknya suatu *tree*. Istilah tersebut adalah :



Gambar 2.1.Pohon Sembarang.

Sumber : (MUN-06)

1. Simpul (*node*)

Yang dimaksud dengan *node* atau simpul pada sebuah *tree* yaitu sejumlah elemen-elemen dari *tree* yang akan terisi oleh data atau informasi beserta pointer percabangannya.

Tree pada gambar 2.1 terlihat bahwa yang merupakan *node* adalah a, b, c, d, e, f, g, h, I, dan j. terlihat ada 10 *node* yang berisi data atau informasi, pada *node* disini terdapat 2 istilah yaitu simpul luar (*external node*) dan simpul dalam (*internal node*).

External node (daun) yaitu simpul yang tidak memiliki anak atau tidak mempunyai keturunan di bawahnya. Yang merupakan daun pada gambar 2.1 adalah simpul d, e, g, I, dan j.

Internal node adalah semua yang bukan merupakan akar dan simpul luar. Atau dengan kata lain yaitu *node* yang bukan daun. Pada gambar 2.1 yang menunjukkan *node internal* yaitu pada *node* b, c, f, dan h.

2. Akar (*root*)

Pada suatu *tree* keberadaan *root* sangatlah penting. Awal dari *tree* dengan suatu *node* itulah yang disebut dengan *root*. Atau dengan kata lain *root* adalah *node* paling atas dari suatu struktur *tree*. Pada gambar 2.1 *root* dinyatakan dengan *node* A. Bila suatu *node* mempunyai cabang di bawahnya baik itu satu, dua atau lebih maka *node* tersebut dikatakan mempunyai anak (*child*). Jadi *node* A disebut sebagai orangtua (*parents*) dari cabang pointer yang di bawahnya satu tingkat.

Pada gambar 2.1 dapat dilihat bahwa simpul *node* A adalah *parents* dari *node* b dan *node* f, sedangkan *node* b dan *node* f adalah saudara (*sibling*). Dan *node* b merupakan *parents* dari *node* c dan d dan seterusnya.

3. Tingkat (*level*)

Level dari suatu *tree* juga disebut dengan tingkat. Suatu *node* ditentukan dengan pertama kali menentukan *root* sebagai tingkat 1, jika suatu *node* dinyatakan sebagai tingkat n maka simpul-simpul yang merupakan anaknya dikatakan berada dalam tingkat n + 1.

Dengan melihat gambar 2.1 tersebut level 1 terletak pada *node* a, yang kemudian diikuti dengan level 2 dibawahnya yang terdiri dari *node* b dan *node* f. level 3 terletak pada *node* c, d, g, h, dan I. Sedangkan level 4 yang merupakan level terakhir dari gambar terletak pada *node* e dan *node* j. Sehingga pada gambar 2.1 terdapat 4 level yang disetiap levelnya memiliki *node* yang merupakan *child*.

4. Derajat (*degree*)

Derajat pada suatu *tree* yaitu banyaknya *pointer* percabangan yang menunjukkan turunan dari *node* tersebut.

Dari gambar 2.1 dapat dikatakan bahwa :

- Node* a dan b berderajat 2
- Node* f berderajat 3

- c) *Node* c dan h berderajat 1
- d) *Node* d, g, h, dan I berderajat 0

5. Tinggi / kedalaman (*height / depth*)

Tinggi atau kedalaman yang biasa disebut juga dengan *height* atau *depth* dari suatu *tree* yaitu level maksimum dari *node* yang ada pada *tree* tersebut dikurangi dengan a, karena *root* pada *tree* tidak termasuk pada hitungan tinggi atau kedalaman. Dengan melihat gambar 2.1 maka *tree* tersebut dapat dikatakan memiliki kedalaman 3 yang berakar pada *node* a.

6. Ancestor

Ancestor adalah semua *node* yang terletak satu jalur dengan *node* yang dicari, dari akar sampai pada *node* yang ditinjau. Contoh pada gambar 2.1 ancestor terdapat pada *node* a, b, c, dan e.

7. Hutan (*forest*)

Hutan (*forest*) adalah kumpulan sejumlah *tree* yang tidak saling berhubungan karena jika pada gambar 2.1 sebagai contoh *node* a dihapus maka anak-anak dibawahnya akan membentuk suatu *tree* yang baru. Dimana antar *tree* batu tidak akan saling ada hubungan. Dan dari situlah terbentuk hutan (Munir : 2006).

2.2.4 *Decision Tree*

Decision tree learning adalah salah satu metode pembelajaran yang sangat populer dan banyak digunakan secara praktis. Metode ini merupakan metode yang memudahkan pengguna untuk menemukan fungsi-fungsi pendekatan yang bernilai diskrit dan tahan terhadap data-data yang terdapat kesalahan (*noisy data*) serta mampu mempelajari ekspresi-ekspresi *disjunctive* (SUY-07).

Metode *Decision Tree* termasuk kedalam *Supervised Learning* yaitu setiap masukan memiliki keluaran yang nantinya akan dijadikan sebagai acuan pembelajaran (*learning*). Maka dalam proses pembelajarannya membutuhkan kumpulan data yang banyak dengan kemungkinan variasi yang berbeda-beda pada tiap kelas yang telah didefinisikannya. Hal ini disebabkan metode *Decision Tree* menganalisa dari data yang telah ada. Apabila data tersebut secara sistematis

kekurangan data pembelajarannya maka hasil klasifikasi akan menjadi kurang optimal.

Decision Tree merupakan sebuah struktur pohon, dimana setiap *node* pohon merepresentasikan atribut yang telah diuji, setiap cabang merupakan suatu pembagian hasil uji, dan *node* daun (*leaf*) merepresentasikan kelompok kelas tertentu. Level *node* teratas dari sebuah *Decision Tree* adalah *node* akar (*root*) yang biasanya berupa atribut yang paling memiliki pengaruh terbesar pada suatu kelas tertentu. Pada proses mengklasifikasi data yang tidak diketahui, nilai atribut akan diuji dengan cara melacak jalur dari *node* akar (*root*) sampai *node* akhir (*daun*) dan kemudian akan diprediksi kelas yang memiliki suatu data baru tertentu.

2.2.5 Algoritma *Iterative Dichotomiser 3* (ID3)

Salah satu algoritma induksi pohon keputusan adalah ID3 (*Iterative Dichotomiser 3*) yang menjadi algoritma *decision tree* yang paling dasar. Algoritma ini melakukan pencarian secara menyeluruh pada semua kemungkinan pohon keputusan. Algoritma ID3 dapat diimplementasikan menggunakan fungsi rekursif yaitu fungsi yang memanggil dirinya sendiri (SUY-07).

ID3 dikembangkan oleh J. Ross Quinlan dengan konsep membangun *decision tree* secara *top-down* dengan mengecek atribut mana yang harus diletakkan pada *root* dengan cara menghitung nilai *entropy* dan *information gain* untuk mengukur efektivitas suatu atribut dalam mengklasifikasikan kumpulan sampel data.

Algoritma pada metode ini berbasis pada Occam's razor yaitu lebih memilih pohon keputusan yang lebih kecil (teori sederhana) dibanding yang lebih besar. Tetapi tidak dapat selalu menghasilkan pohon keputusan yang paling kecil, sehingga Occam's Razor bersifat heuristic. Occam's Razor diformalisasi menggunakan konsep dari entropy informasi. Berikut algoritma dari ID3 :

Input : sampel training, label training, atribut

- Membuat simpul akar untuk pohon yang dibuat
- Jika semua sampel positif, berhenti dengan suatu pohon dengan satu simpul

akar beri label (+)

- Jika semua sampel negative, berhenti dengan suatu pohon dengan satu simpul akar beri label (-)
- Jika atribut kosong, berhenti dengan suatu pohon dengan satu simpul akar, dengan label sesuai nilai yang terbanyak yang ada pada label training
- Untuk yang lain, Mulai
 - $A \leftarrow$ Atribut yang mengklasifikasikan sampel dengan hasil terbaik (berdasarkan gain ratio)
 - Atribut keputusan untuk simpul akar $\leftarrow A$
 - Untuk setiap nilai, V_i yang mungkin untuk A ,
 - Tambahkan cabang dibawah akar yang berhubungan dengan $A = V_i$
 - Tentukan sampel SV_i sebagai subset dari sampel yang mempunyai nilai V_i untuk atribut A
 - Jika sampel SV_i kosong,
 - ✚ Dibawah cabang tambahkan simpul daun dengan label = nilai yang terbanyak yang ada pada label training
 - ✚ Yang lain, tambah cabang baru dibawah cabang yang sekarang (sampel training, label training, atribut-[A])
- Berhenti

Adapun sampel data yang digunakan oleh ID3 memiliki beberapa syarat, yaitu :

1. Deskripsi atribut-nilai. Atribut yang sama harus mendeskripsikan tiap contoh dan memiliki jumlah nilai yang sudah ditentukan.
2. Kelas yang sudah didefinisikan sebelumnya. Suatu atribut contoh harus sudah didefinisikan, karena mereka tidak dipelajari oleh ID3.
3. Kelas-kelas yang diskrit. Kelas harus digambarkan dengan jelas.
4. Jumlah contoh yang cukup. Karena pembangkitan induktif digunakan, maka dibutuhkan test case yang cukup untuk membedakan pola yang valid dari peluang suatu kejadian.

Berikut ini merupakan beberapa keuntungan dan kerugian dari penggunaan algoritma ID3 :

a. Keuntungan :

- Dapat membuat aturan prediksi yang mudah dimengerti.
- Membangun pohon keputusan dengan cepat.
- Membangun pohon keputusan yang pendek.
- Hanya membutuhkan beberapa tes atribut hingga semua data diklasifikasikan.

b. Kerugian :

- Jika contoh yang diteliti terlalu kecil/ sederhana mungkin membuat data *over-classified*.
- Hanya satu atribut yang dapat dites dalam satu waktu untuk membuat keputusan.
- Mengelompokkan data yang berkelanjutan mungkin terhitung mahal, karena *tree* yang harus dibuat juga banyak.

2.2.6 Entropy Dan Information Gain

Untuk menghitung *information gain* terlebih dahulu kita harus memahami suatu ukuran lain yang disebut *entropy*. Didalam bidang teori informasi, *entropy* sering digunakan sebagai suatu parameter untuk mengukur heterogenitas (keberagaman) dari suatu kumpulan sampel data. Jika kumpulan sampel data semakin heterogen, maka hasil *entropy*nya semakin besar. Secara matematis *entropy* dirumuskan pada persamaan (2-1):

$$. Entropy(S) = P_a \log_2 P_a - P_b \log_2 P_b \quad (2-1)$$

Keterangan :

S = ruang (data) yang digunakan sampel untuk training

P_a = jumlah yang bersolusi positif (mendukung) pada data sampel untuk kriteria tertentu

P_b = jumlah yang bersolusi negative (tidak mendukung) pada data sampel untuk kriteria tertentu

Setelah mendapat nilai *entropy* untuk suatu kumpulan data, maka kita dapat mengukur efektivitas suatu atribut dalam mengklasifikasi data. Ukuran efektivitas ini disebut sebagai *information gain*. Secara matematis, *information gain* dari suatu atribut A, dituliskan pada persamaan (2-2) :

$$Gain (S, A) = Entropy(S) - \sum \frac{|S_v|}{|S|} Entropy(S_v) \quad (2-2)$$

Keterangan :

Gain (S, A) : nilai gain pada seluruh sampel data dari atribut A

A : Atribut

v : menyatakan suatu nilai yang mungkin untuk atribut A

Values (A) : himpunan nilai-nilai yang mungkin untuk atribut A

|S_v| : Jumlah sampel untuk nilai v

|S| : Jumlah seluruh sampel data

Entropy (S_v) : entropy untuk sampel-sampel yang memiliki nilai v

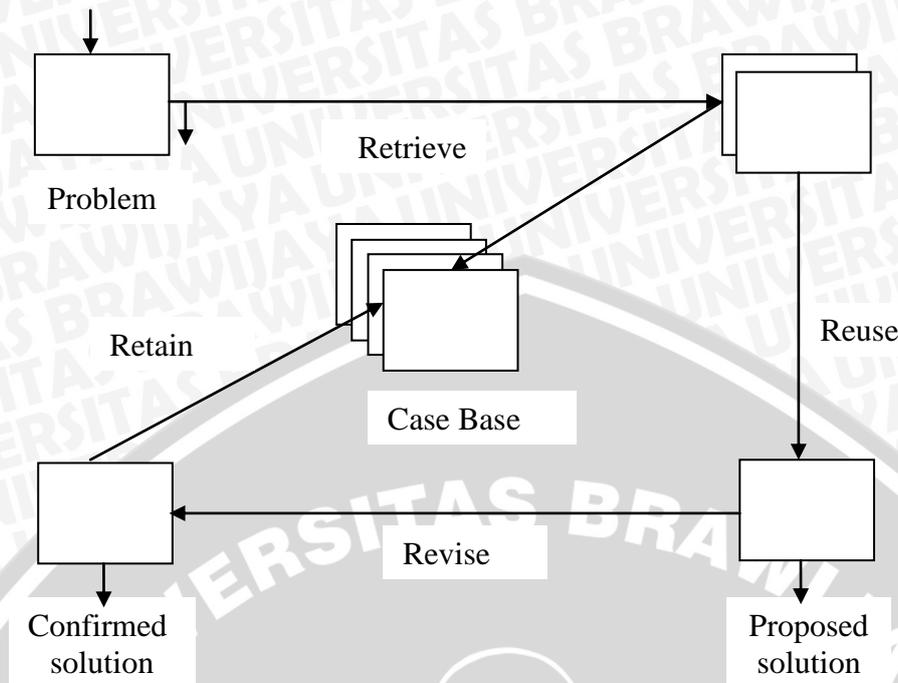
Entropy (S) : entropy pada sampel data

(Suyanto, 2007:139)

2.2.7 Case Based Reasoning

Case Based Reasoning (CBR) merupakan sebuah metode yang merupakan cabang ilmu dari *Artificial Intelligence* (AI) yang digunakan untuk memberikan solusi dari permasalahan dengan menggunakan pengetahuan kasus lama untuk menyelesaikan kasus baru yang memiliki kemiripan dengan kasus sebelumnya (SAL-10).

Perbedaan CBR dari pendekatan AI lainnya yaitu CBR merupakan suatu pendekatan ke arah incremental yaitu pembelajaran yang terus menerus dengan cara melakukan adaptasi pada kasus baru yang dimasukkan ke dalam *database* penyimpanan kasus (*case base*) sehingga secara tidak langsung pengetahuan CBR akan bertambah.



Gambar 2.2 Alur Proses CBR.

Sumber : (LUT-10)

Secara umum metode *Case Based Reasoning* (CBR) terdiri dari 4 tahapan, yaitu (BEG-09) :

a. Retrieve

Dalam proses retrieve dimulai dari penggambaran masalah dengan mengidentifikasi masalah yang baru dengan melihat kemiripan dari kasus lama dan kasus baru dan diakhiri jika ditemukan kecocokan terhadap kasus lama yang tingkat kecocokannya paling tinggi.

b. Reuse

Pada proses reuse, sistem akan menggunakan kembali informasi dari kasus lama untuk menyelesaikan kasus yang baru.

c. Revise

Pada proses ini sistem akan mengevaluasi kembali informasi yang didapatkan dari kasus lama untuk mengatasi kasus baru sehingga dapat memberikan solusi masalah baru.

d. Retain

Pada proses retain ini solusi baru yang dihasilkan akan disimpan ke dalam knowledge base untuk selanjutnya digunakan menyelesaikan kasus baru yang memiliki kemiripan di masa mendatang. Pada gambar 2.2 dapat dilihat alur proses dari CBR.

2.2.8 Pengertian Dasar Sistem

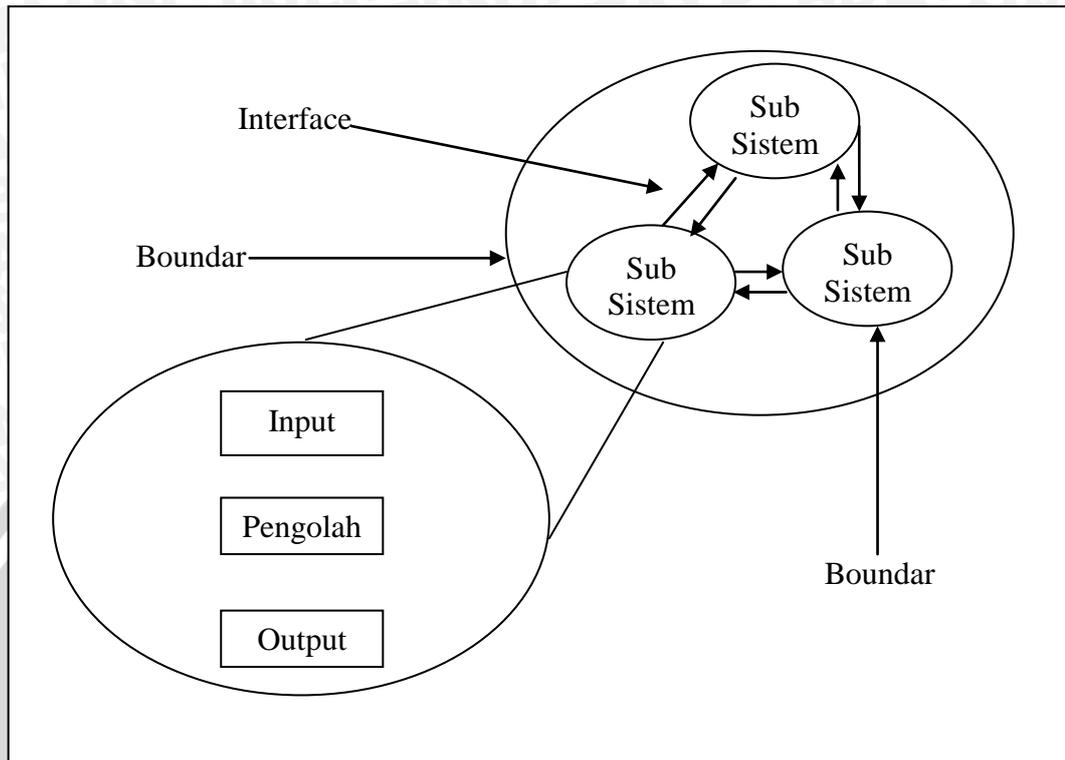
Sistem berasal dari bahasa Latin (*systēma*) dan bahasa Yunani (*sustēma*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi.

Secara sederhana suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel-variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu. Teori sistem secara umum pertama kali diuraikan oleh *Kenneth Boulding*, terutama menekankan pentingnya perhatian terhadap setiap bagian yang membentuk sebuah sistem.

Teori sistem mengatakan bahwa setiap unsur pembentuk organisasi adalah penting dan harus mendapat perhatian yang utuh. Unsur-unsur yang mewakili suatu sistem secara umum adalah masukan (*input*), pengolahan (*prosesing*), dan keluaran (*output*) (WIJ-08). Di samping itu suatu sistem dapat pula dikembangkan dengan menyertakan media penyimpanan. Sistem dapat terbuka dan tertutup.

2.2.9 Karakteristik Sistem

Sebuah sistem terdiri atas bagian-bagian yang saling berkaitan dan bervariasi bersama-sama untuk mencapai beberapa sasaran dan maksud. Sebuah sistem bukanlah seperangkat unsur yang tersusun secara teratur, tetapi terdiri atas unsur yang dapat dikenal yang saling melengkapi karena suatu maksud, tujuan dan sasaran.



Gambar 2.3 Karakteristik Sistem.

Sumber : (JOG-05)

Pada Gambar 2.3 ditunjukkan bahwa sistem mempunyai karakteristik atau sifat-sifat tertentu yaitu :

a. **Komponen Sistem (*Components*)**

Suatu sistem terdiri atas sejumlah komponen yang saling berinteraksi, yang bekerjasama membentuk suatu kesatuan.

b. **Batas Sistem (*Boundary*)**

Batas sistem merupakan daerah yang membatasi antara sistem yang satu dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem memungkinkan suatu sistem dipandang sebagai suatu kesatuan yang tidak dapat dipisah-pisahkan.

c. **Lingkungan Luar Sistem (*Environment*)**

Lingkungan luar dari sistem adalah apapun di luar ruang lingkup sistem yang mempengaruhi operasi sistem. Lingkungan luar dapat bersifat menguntungkan dan dapat pula bersifat merugikan sistem tersebut.

Lingkungan luar yang menguntungkan merupakan energi bagi sistem tersebut, yang dengan demikian lingkungan luar tersebut harus selalu dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus dikendalikan, jika tidak maka akan mengganggu kelangsungan hidup sistem tersebut.

d. Penghubung Sistem (*Interface*)

Interface merupakan media penghubung antara suatu subsistem dengan subsistem lainnya. Penghubung ini memungkinkan sumber daya mengalir dari suatu subsistem ke subsistem lainnya. Keluaran (*Output*) dari suatu subsistem akan menjadi masukan (*Input*) untuk subsistem lainnya dengan melalui penghubung. Dengan demikian terjadi suatu integrasi sistem yang membentuk satu kesatuan.

e. Masukan Sistem (*Input*)

Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Sebagai contoh, di dalam suatu unit sistem komputer, “program” adalah *maintenance input* yang digunakan untuk mengoperasikan komputer sementara “data” adalah signal input yang akan diolah menjadi informasi.

f. Keluaran Sistem (*Output*)

Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan untuk subsistem lain.

g. Pengolah Sistem (*Proses*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

h. Sasaran Sistem (*Objective*)

Suatu sistem mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.