

## BAB II

### LANDASAN TEORI

#### 2.1 Studi Keperustakaan

Adapun beberapa penelitian yang telah dilakukan sebelumnya mengenai penerapan metode *Case Based Reasoning* (CBR) pada sistem pakar diantaranya sebagai berikut.

##### 2.1.1 Berdasarkan Kesamaan Metode

Penelitian Muhammad Shaid, dkk (2013) tentang sistem pakar pertumbuhan balita berbasis web dengan metode *Case Based Reasoning*. Pada penelitian ini sistem dibuat sebagai sarana untuk menganalisa pertumbuhan balita, yang diharapkan akan bisa membantu bidan dan dokter anak dalam pencarian data pertumbuhan balita. Metode *Case Based Reasoning* digunakan dalam aplikasi pertumbuhan balita dengan perhitungan *nearest neighbor*, dimana data kasus baru dibandingkan dengan perhitungannya dengan data kasus lama yang ada di database, dan kemudian dihitung kriteria kemiripannya berdasarkan rumus atau ketentuan yang berlaku. Hasil database dapat menghasilkan data keluaran yang berupa perbandingan antara kasus lama dengan kasus baru untuk penentuan kriteria kemiripan kasus pertumbuhan balita, yang dipengaruhi oleh jenis kelamin, usia, berat badan, dan tinggi badan. Pada penelitian ini memiliki kesamaan pada metode yang digunakan tetapi memiliki perbedaan pada kasus, kasus yang di angkat pada penelitian ini tentang sistem pakar pertumbuhan palita.

Selanjutnya penelitian Faza Akmal, dkk (2014) tentang sistem pakar untuk mendiagnosa penyakit lambung dengan implementasi metode CBR (*Case Based Reasoning*) berbasis web. Pada penelitian ini sistem dijadikan sebagai salah satu solusi alternatif untuk penyampaian informasi kepada masyarakat tentang penyakit lambung yang menyerang manusia mencakup gejala penyakit, penyebab penyakit, serta pencegahan atau penanganannya. Pada penelitian ini memiliki kesamaan pada metode yang digunakan tetapi memiliki perbedaan pada penyakit yang digunakan, penyakit yang digunakan pada penelitian ini adalah penyakit lambung.

### **2.1.2 Berdasarkan Kesamaan Penyakit**

Penelitian Lasmedi Afuan (2008) dengan judul “ Sistem Pakar Berbasis Web Untuk Diagnosa Penyakit Kulit Manusia Menggunakan *Fuzzy Mamdani* (berdasarkan indeks kekuatan dan kelemahan)”. Bidang kesehatan merupakan salah satu bidang yang sudah menggunakan teknologi komputer, salah satunya digunakan untuk mendiagnosa penyakit kulit pada manusia. Mengingat banyaknya keluhan dan gejala yang dirasakan, maka akan sangat sulit untuk menentukan jenis penyakit kulit yang diderita oleh setiap pasien. Untuk mengatasi hal ini, maka diperlukan sebuah sistem pakar yang merupakan salah satu cabang dari kecerdasan buatan (*Artificial Intelligence*) untuk membantu mendiagnosa penyakit kulit berdasarkan gejala atau keluhan yang dirasakan oleh pasien. Sistem pakar dibuat dengan menggunakan penalaran teori *Fuzzy Mamdani*. Sistem ini diharapkan dapat memberikan saran untuk melakukan tindakan atau pengobatan

yang sesuai dengan jenis penyakit kulit yang diderita pasien. Diharapkan dengan sistem ini, orang awam dapat menyelesaikan masalah tertentu tanpa bantuan para ahli dalam bidang tersebut. Sedangkan bagi para ahli, sistem ini dapat digunakan untuk mendokumentasikan pengetahuan yang dimiliki. Setelah diuji dan dianalisis dengan melibatkan perhitungan secara manual, dapat diketahui bahwa secara garis besar hasil yang didapat dari perhitungan sistem sama dengan perhitungan manual. Sehingga secara umum sistem telah dapat memberikan solusi terhadap pengguna dan sistem berjalan sesuai dengan yang diharapkan. Dapat dilihat pada penelitian ini memiliki kesamaan pada penyakit yaitu sama-sama menggunakan penyakit kulit, sedangkan memiliki perbedaan pada metode yang digunakan, dalam penelitian ini menggunakan penalaran metode *fuzzy* mamdani.

Penelitian selanjutnya Wardhani (2011) dengan judul “Perancangan Program Sistem Pakar untuk Mendiagnosa Penyakit Sendi dan Kulit pada Manusia dengan Memanfaatkan Teknologi Dongle”, Tujuan dari perancangan sistem pakar diagnosa sendi dan kulit adalah untuk memecahkan masalah masyarakat di bidang kesehatan, terutama ketika mereka memerlukan konsultasi dengan dokter spesialis. Hal ini diperlukan bahwa sistem dikembangkan untuk mendukung perangkat lunak keamanan, seperti menggunakan sistem dongle untuk memastikan bahwa hanya pengguna yang berwenang dapat menggunakan aplikasi sistem pakar. Sistem ini menerapkan metode *forward chaining*. Selain itu mempekerjakan metode *waterfall* dalam analisis dan desain sistem. Hasil penelitian ini menyediakan sistem untuk mendiagnosa penyakit sendi dan kulit yang dapat digunakan pengguna dengan memasukkan gejala penyakit. Ini

membantu medis praktisi untuk mendiagnosa penyakit sehingga mereka dapat menemukan yang paling terapi yang tepat bagi pasien. Tidak ada cara lain untuk menjalankan aplikasi kecuali menggunakan dongle. Untuk penelitian terakhir ini memiliki kesamaan pada penyakit, namun memiliki perbedaan pada metode yang digunakan, pada penelitian ini menggunakan metode *forward chaining*.

## 2.2 Pengertian Sistem Pakar

Istilah sistem pakar berasal dari istilah *knowledge-based expert system*. Istilah ini muncul karena untuk memecahkan masalah, sistem pakar menggunakan pengetahuan seorang pakar yang dimasukkan ke dalam komputer. Seseorang yang bukan pakar menggunakan sistem pakar untuk meningkatkan kemampuan pemecahan masalah, sedangkan seorang pakar menggunakan sistem pakar untuk *knowledge assistant* (Sutojo, 2011:156).

Sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikan olehnya. Selain manfaat, ada juga beberapa kekurangan yang terdapat pada sistem pakar. Adapun beberapa pengertian sistem pakar menurut para ahli antara lain sebagai berikut :

1. Menurut (A. Farid Aziz, 1994) mendefinisikan bahwa sistem pakar adalah sebuah perangkat lunak komputer yang memiliki basis pengetahuan untuk domain tertentu dan menggunakan penalaran inferensi menyerupai seorang pakar dalam menyelesaikan masalah.

2. Menurut (Suparman, 1991:99) mendefinisikan bahwa sistem pakar adalah suatu program kecerdasan buatan yang menggabungkan dasar pengetahuan (*Knowledge base*) dengan sistem Inferensi.
3. Menurut (Suparman, 1991:13) mendefinisikan bahwa sistem pakar bukan untuk menggantikan kedudukan seorang pakar tetapi hanya memasyarakatkan pengetahuan dan pengalaman pakar-pakar yang berpengalaman (*inferensi*), komputer dapat disejajarkan sebagai alat bantu yang dapat digunakan secara praktis dalam memecahkan masalah dan pengambilan keputusan.

Aplikasi sistem pakar menyentuh beberapa area permasalahan berikut : Interpretasi, Prediksi, Diagnosis, Disain, *Planning*, *Monitoring*, *Debugging*, *Reparasi*, *Instruction*, dan Kontrol. Konsep dasar sistem pakar meliputi enam hal berikut : Kepakaran (*Expertise*), Pakar (*Expert*), Pemindahan Kepakaran (*Transferring Expertise*), Inferensi (*Inferencing*), Aturan-aturan (*Rule*), dan kemampuan menjelaskan (*Explanation Capability*).

### 2.2.1 Karakteristik Sistem Pakar

Seperti halnya sistem-sistem lain, sistem pakar juga mempunyai karakteristik antara lain sebagai berikut :

1. Pengetahuan sistem pakar merupakan suatu konsep, bukan berbentuk numeris. Hal ini dikarenakan komputer melakukan poses pengolahan data secara numeric sedangkan keahlian dari seorang pakar adalah fakta dan aturan-aturan.

2. Informasi dari sistem pakar tidak selalu lengkap, subjektif, tidak konsisten, subjek terus berubah dan tergantung pada kondisi lingkungan sehingga keputusan yang diambil bersifat tidak pasti dan tidak mutlak “ya” atau “tidak” akan tetapi menurut ukuran kebenaran tertentu. Oleh karena itu dibutuhkan kemampuan sistem untuk belajar secara mandiri dalam menyelesaikan masalah-masalah dengan pertimbangan-pertimbangan khusus.
3. Kemungkinan solusi sistem pakar terhadap suatu permasalahan adalah bervariasi dan mempunyai banyak pilihan jawaban yang dapat diterima, semua faktor yang ditelusuri memiliki ruang masalah yang luas dan tidak pasti. Oleh karena itu diperlukan fleksibilitas sistem dalam menangani kemungkinan solusi dari berbagai permasalahan yang ada.
4. Perubahan atau pengembangan pengetahuan dalam sistem pakar dapat terjadi setiap saat bahkan sepanjang waktu sehingga diperlukan kemudahan dalam memodifikasi sistem untuk menampung jumlah pengetahuan yang semakin besar dan semakin bervariasi.
5. Pandangan dan pendapat setiap pakar tidaklah selalu sama, oleh karena itu tidak ada jaminan bahwa solusi sistem pakar merupakan jawaban yang pasti benar. Setiap pakar akan memberikan pertimbangan-pertimbangan berdasarkan faktor subyektif.
6. Keputusan merupakan bagian terpenting dari sistem pakar. Sistem pakar harus memberikan solusi yang akurat berdasarkan masukkan pengetahuan meskipun solusinya sulit sehingga fasilitas informasi sistem harus selalu diperlukan.

### 2.2.2 Manfaat Sistem Pakar

Sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikannya, antara lain :

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.
2. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas, dengan member nasehat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi di lingkungan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.
7. Memiliki kehandalan yakni sistem pakar tidak pernah menjadi bosan dan kelelahan atau sakit.
8. Meningkatkan kapabilitas sistem computer. Integrasi sistem pakar dengan computer lain membuat sistem lebih efektif dan mencakup lebih banyak aplikasi.
9. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti. Berbeda dengan sistem computer konvensional, sistem pakar dapat bekerja dengan informasi yang tidak lengkap. Pengguna dapat merespon dengan: “tidak tahu” dan “tidak yakin” pada satu atau lebih pertanyaan selama konsultasi dan sistem pakar tetap akan memberikan jawabannya.
10. Bisa digunakan sebagai media pelengkap dalam pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih

berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.

11. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

### 2.2.3 Konsep Dasar Sistem Pakar

Adapun Konsep dasar dalam sistem pakar dapat meliputi enam hal antara lain sebagai berikut :

#### 1. Kepakaran (*Expertise*)

Kepakaran merupakan suatu pengetahuan yang diperoleh dari pelatihan, membaca, dan pengalaman. Kepakaran inilah yang memungkinkan para ahli dapat mengambil keputusan lebih cepat dan lebih baik daripada seseorang yang bukan pakar. Kepakaran itu sendiri meliputi pengetahuan tentang.

- a. Fakta-fakta tentang bidang permasalahan tertentu,
- b. Teori-teori tentang bidang permasalahan tertentu,
- c. Aturan-aturan dan prosedur-prosedur menurut bidang permasalahan umumnya,
- d. Aturan heuristic yang harus dikerjakan dalam suatu situasi tertentu,
- e. Strategi global untuk memecahkan permasalahan,
- f. Pengetahuan tentang pengetahuan (*meta knowledge*).

## 2. Pakar (*Expert*)

Pakar adalah seorang yang mempunyai pengetahuan, pengalaman, dan metode khusus, serta mampu menerapkannya untuk memecahkan masalah atau memberi nasehat. Seorang pakar harus mampu menjelaskan dan mempelajari hal-hal yang berkaitan dengan topic permasalahan, jika perlu harus mampu menyusun kembali pengetahuan-pengetahuan yang didapatkan, dan dapat memecahkan aturan-aturan serta menentukan relevansi kepakarannya. Jadi seseorang pakar harus mampu melakukan kegiatan-kegiatan berikut :

- a. Mengenal dan memformulasikan permasalahan,
- b. Memecahkan permasalahan dengan cepat dan tepat,
- c. Menerangkan pemecahannya,
- d. Belajar dari pengalaman,
- e. Merestrukturasikan pengetahuan,
- f. Memecahkan aturan-aturan,
- g. Menentukan relevansi.

## 3. Pemindehan kepakaran (*Transferring Expertise*)

Tujuan dari sistem pakar adalah memindahkan kepakaran dari seseorang pakar ke dalam computer, kemudian ditransfer kepada orang lain yang bukan pakar. Proses ini melibatkan empat kegiatan yaitu :

- a. Akuisisi pengetahuan (dari pakar atau sumber lain),
- b. Representasi pengetahuan (pada komputer),
- c. Inferensi pengetahuan,
- d. Pemindehan pengetahuan ke pengguna.

#### 4. Inferensi (*Inferencing*)

Inferensi adalah sebuah prosedur (program) yang mempunyai kemampuan dalam melakukan penalaran. Inferensi ditampilkan pada suatu komponen yang disebut mesin inferensi yang mencakup prosedur-prosedur mengenai pemecahan masalah. Semua pengetahuan yang dimiliki oleh seorang pakar disimpan pada basis pengetahuan oleh sistem pakar. Tugas mesin inferensi adalah mengambil kesimpulan berdasarkan basis pengetahuan yang dimilikinya.

#### 5. Kemampuan menjelaskan (*Explanation Capability*)

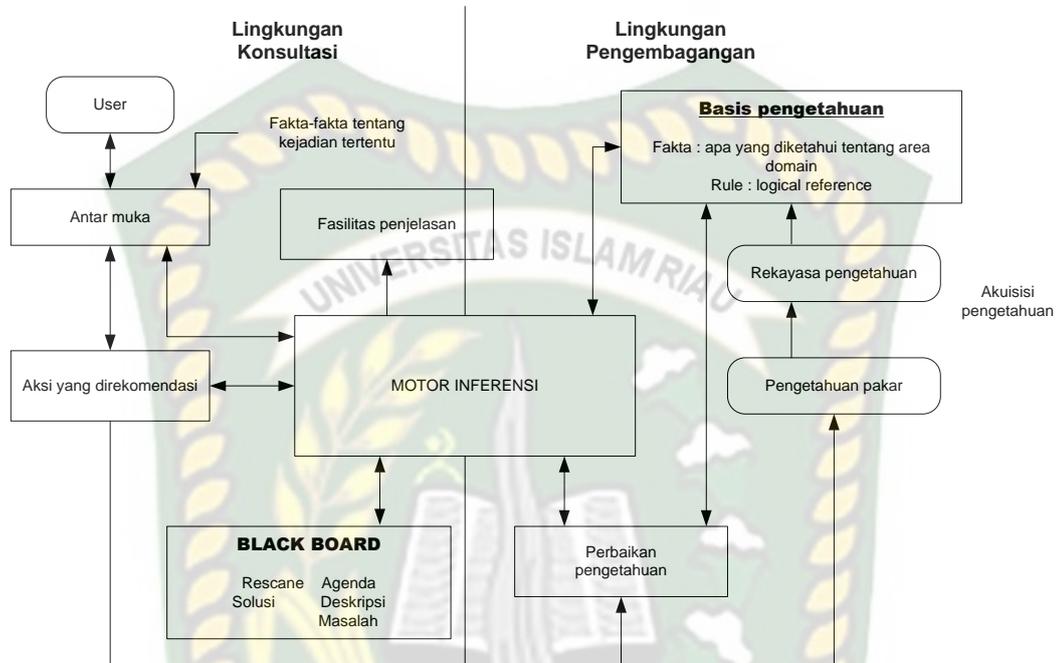
Fasilitas lain dari sistem pakar adalah kemampuan untuk menjelaskan saran atau rekomendasi yang diberikannya. Penjelasan dilakukannya dalam subsistem yang disebut subsistem penjelasan (*explanation*). Bagian dari sistem ini memungkinkan sistem untuk memeriksa penalaran yang dibuatnya sendiri dan menjelaskan operasi-operasinya.

#### 2.2.4 Struktur Sistem Pakar

Ada dua bagian penting dalam sistem pakar, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan digunakan oleh pembuat sistem pakar untuk membangun komponen-komponennya dan memperkenalkan pengetahuan ke dalam knowledge base (basis pengetahuan).

Lingkungan konsultasi digunakan oleh pengguna untuk berkonsultasi sehingga pengguna mendapatkan pengetahuan dan nasihat dari sistem pakar layaknya berkonsultasi dengan seorang pakar (Sutojo, 2011:166).

Komponen-komponen yang penting dalam sebuah sistem pakar ditunjukkan pada gambar 2.1 berikut :



**Gambar 2.1 Komponen-Komponen Yang Penting Dalam Sebuah Sistem Pakar**

Sumber : Sutojo, 2010, Kecerdasan Buatan.

### 2.3 Metode *Case Based Reasoning* (CBR)

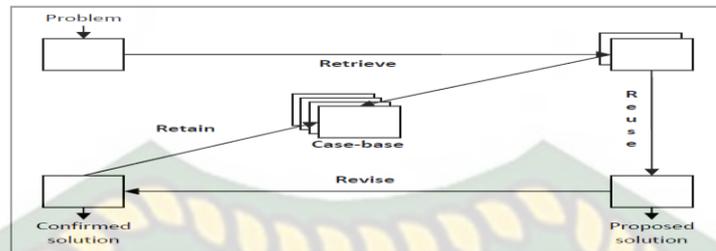
*Case Based Reasoning* (CBR) suatu model penalaran yang menggabungkan pemecahan masalah, pemahaman dan pembelajaran serta memadukan keseluruhannya dengan pemrosesan memori. Tugas tersebut dilakukan dengan memanfaatkan kasus yang pernah dialami oleh sistem, yang mana kasus merupakan pengetahuan dalam konteks tertentu yang mewakili suatu pengalaman yang menjadi dasar pembelajaran untuk mencapai tujuan sistem.

Dalam definisi lain CBR merupakan metode pemecahan masalah/kasus baru dengan melakukan adaptasi terhadap metode yang digunakan untuk memecahkan masalah/kasus lama.

Alur proses CBR dalam memecahkan masalah didefinisikan dalam 4 langkah RE, yaitu :

- a. *REtrieve*, mengambil masalah/kasus yang paling serupa. Pada saat terdapat kasus baru, proses retrieve akan melakukan dua langkah pemrosesan, yaitu pengenalan kasus dan pencarian persamaan kasus pada database
- b. *REuse*, menggunakan kembali masalah/kasus untuk mencoba memecahkan masalah/kasus. Langkah ini dilakukan setelah proses retrieve selesai dilakukan, di dalam proses reuse sistem akan menggunakan informasi kasus sebelumnya yang memiliki kesamaan untuk menyelesaikan kasus yang baru, pada proses reuse akan menyalin, menyeleksi, dan melengkapi informasi yang akan digunakan.
- c. *REvise*, merevisi solusi yang diajukan jika diperlukan. Proses selanjutnya ini dilakukan apabila solusi baru yang dihasilkan memiliki persamaan 80% maka solusi baru dapat di revise untuk mengatasi dan meninjau kembali kesalahan-kesalahan yang dihasilkan dari proses reuse, jika berhasil maka akan dilanjutkan dengan proses retain.
- d. *REtain*, mempertahankan/menyimpan solusi baru sebagai bagian dari masalah/kasus baru. Pada proses terakhir solusi baru yang telah di revise akan disimpan ke dalam database untuk memecahkan masalah/kasus yang akan datang.

Adapun alur proses dari CBR dapat dilihat pada gambar 2.2 berikut :



**Gambar 2.2 Alur Proses Metode CBR**

Sumber : Kecerdasan Buatan (Sutojo: 2010)

Masalah/kasus baru akan dicocokkan dengan kasus-kasus dalam CBR sistem, dan satu atau lebih kasus serupa yang akan diambil. Solusi yang disarankan berdasarkan kasus yang serupa kemudian digunakan dan diujikan terhadap kasus baru. Kecuali jika kasus disarankan sangat cocok dengan kasus baru, maka kemungkinan akan diperlukan revisi terhadap solusi, dan menghasilkan kasus baru untuk dipertahankan/disimpan. Keseluruhan proses CBR tersebut dijalankan tanpa intervensi manusia atau dengan kata lain secara otomatis.

Dalam mencari kasus lama yang memiliki kemiripan dengan kasus baru maka akan digunakan konsep *similarity measure* yang diperkenalkan oleh Tversky 1977 yaitu dengan menggunakan persamaan berikut :

$$SM_{pq} = \frac{\alpha(\text{common})}{\alpha(\text{common}) + \beta(\text{different})}$$

**Gambar 2.3 Rumus *Similarity Measure***

Di mana SM adalah *similarity measure* dengan p adalah kasus baru dan q adalah kasus yang tersimpan di *case base*. *Common* merupakan jumlah atribut yang sama, sedangkan *different* adalah jumlah atribut yang beda. Untuk suatu atribut yang mana ada kasus lama dan kasus baru memiliki nilai yang sama akan

diberi nilai 1 sedangkan bila atribut kasus yang baru dan kasus yang lama tidak sama maka diberikan nilai 0.

- a. Jika atribut ada pada kasus baru dan ada pada kasus lama maka diberi nilai 1 (Satu).
- b. Jika atribut tidak ada pada kasus baru dan tidak ada pada kasus lama maka diberi nilai 1 (Satu).
- c. Jika atribut ada pada kasus baru dan tidak ada pada kasus lama maka diberi nilai 0 (Nol).
- d. Jika atribut tidak ada pada kasus baru dan ada pada kasus lama maka diberi nilai 0 (Nol).

### **1.3.1 Kelebihan Metode *Case Based Reasoning* (CBR)**

Adapun kelebihan dari metode CBR antara lain sebagai berikut :

- a. Memecahkan masalah dengan mudah karena dapat mengambil solusi dengan cepat dan tepat.
- b. Semakin banyak pengalaman yang tersimpan di dalam sistem maka sistem akan menjadi lebih pintar dalam menemukan solusi untuk sebuah kasus.

### **1.3.2 Kekurangan Metode *Case Based Reasoning* (CBR)**

Adapun kekurangan dari metode CBR antara lain sebagai berikut :

- a. Tidak menjamin solusi yang didapat itu menjadi solusi terbaik atau maksimal, karena dalam metode ini sangat bergantung pada kasus yang pernah terjadi,

makadari itu jika solusi dari kasus yang pernah terjadi salah, maka dalam hal ini tahapan *Revise* sangat diperlukan untuk mengurangi tingkat kesalahannya.

- b. Semakin banyak pengalaman yang tersimpan di basis kasus, maka tidak menutup kemungkinan dalam menyelesaikan suatu kasus baru akan memakan waktu yang lama, karena harus mencari dan membandingkan dengan basis kasus yang paling mirip.

#### **2.4 Analisis Kasus Pada Metode *Case Based Reasoning* (CBR)**

Metode *Case Based Reasoning* merupakan metode yang menerapkan 4 tahapan proses, yaitu retrieve, reuse, revise, dan retain. Cara kerja sistem secara umum berpedoman pada basis pengetahuan yang dimiliki oleh sistem yang bersumber dari kasus-kasus yang pernah dialami oleh pasien pengidap penyakit kulit kepala yang kemudian dihitung tingkat kemiripannya dengan kasus yang baru yang dimasukkan pengguna. Berdasarkan tingkat kemiripan kasus inilah sistem akan mengeluarkan diagnosis jenis jenis penyakit kulit kepala beserta solusi pencegahan dan pengobatan penyakit tersebut. Analisa data kasus-kasus lama pada tabel 2.1 berikut:

**Tabel 2.1 Kasus yang sudah ada**

No	Gejala	Kasus Yang Sudah Ada		
		Kasus 1	Kasus 2	Kasus 3
1.	Jenis Kulit Kepala	Kering	Kering	Berminyak
2.	Kulit kepala bersisik	1	0	0
3.	Kulit kepala berkerak	1	0	0
4.	Kulit kepala berkerak berwarna kuning	0	0	0
5.	Kulit kepala meradang	1	0	0
6.	Ruam Pada Kulit Kepala	0	0	1
7.	Kulit Kepala Menebal (Plak)	1	0	0
8.	Kulit Kepala Terkelupas Berguguran	0	1	0
9.	Serbuk Putih Dirambut	0	1	1
10.	Kulit Kepala Terasa Gatal	0	1	1
Penyakit yang terdeteksi		Penyakit A	Penyakit B	Penyakit C
Solusi yang diberikan		Solusi A	Solusi B	Solusi B

Selanjutnya terdapat kasus baru dengan gejala-gejala seperti pada tabel 2.2 berikut:

**Tabel 2.2 Kasus Baru**

No	Gejala	Kasus Baru (X)
1.	Jenis Kulit Kepala	Kering
2.	Kulit kepala bersisik	0
3.	Kulit kepala berkerak	0
4.	Kulit kepala berkerak berwarna kuning	0
5.	Kulit kepala meradang	0
6.	Ruam Pada Kulit Kepala	0
7.	Kulit Kepala Menebal (Plak)	0
8.	Kulit Kepala Terkelupas Berguguran	1
9.	Serbuk Putih Dirambut	1
10.	Kulit Kepala Terasa Gatal	0

#### 2.4.1 Proses *Retrieve*

Proses Retrieve merupakan proses pencarian kemiripan kasus baru dengan kasus yang lama. Pencarian kemiripan antara kasus baru dengan kasus lama dilakukan dengan cara mencocokkan gejala yang diinputkan oleh pengguna dengan gejala yang ada pada basis pengetahuan. Pada awal proses diagnosa pengguna

akan menginputkan gejala-gejala penyakit kulit kepala yang muncul dan selanjutnya akan diproses secara keseluruhan. Kemudian sistem akan melakukan proses perhitungan nilai kemiripan (*similarity*) dengan ketentuan seperti dibawah ini.

- a. Jika gejala ada pada kasus baru dan ada pada kasus yang sudah ada maka bernilai 1 (Satu).
- b. Jika gejala tidak ada pada kasus baru dan tidak ada pada kasus yang sudah ada maka bernilai 1 (Satu).
- c. Jika gejala ada pada kasus baru dan tidak ada pada kasus yang sudah ada maka bernilai 0 (Nol).
- d. Jika gejala tidak ada pada kasus baru dan ada pada kasus yang sudah ada maka bernilai 0 (Nol).

Perhitungan dilakukan menggunakan rumus pada gambar 2.3.

**Tabel 2.3 Kemiripan Kasus Baru Dengan Kasus Yang Sudah Ada**

Kasus Baru (x)	Kemiripan Dengan Kasus Yang Sudah Ada		
	Kasus 1	Kasus 2	Kasus 3
1	1	1	0
0	0	1	1
0	0	1	1
0	1	1	1
0	0	1	1
0	1	1	0
0	0	1	1
1	0	1	0
1	0	1	1
0	0	0	0
<i>Common</i>	3	9	6
<i>Diferent</i>	7	1	4
Nilai Kemiripan	0.3	0.9	0.6

Maka didapatkan nilai kemiripan yang paling besar 0.9 pada kasus ke 2, jadi kesimpulannya kasus baru (x) tersebut mirip dengan kasus 3 dengan jenis penyakit B dengan salah solusi B.

## **2.5 Penyakit Kulit Kepala Pada Manusia**

Penyakit kulit kepala adalah penyakit yang umum dan sering dialami oleh banyak orang, aktifitas di luar ruangan dan perawatan rambut yang kurang bersih kadang bisa memicu terjadinya beberapa gangguan kulit, misalnya saja ketombe, kulit kepala yang bersisik dan kulit kepala yang kering. Hal ini sering diistilahkan dengan “Patologi Rambut”. Dengan demikian patologi rambut itu adalah suatu ilmu yang mempelajari dan mengetahui keadaan kulit kepala dan rambut yang tidak sehat

### **2.5.1. Jenis-Jenis Penyakit Kulit Kepala Pada Manusia**

Jenis-jenis penyakit kulit kepala pada manusia antara lain :

#### **1. Bisul (*Furunkulosis*)**

Kelainan ini merupakan peradangan terbanyak yang di sebabkan oleh bakteri stafilokokus. Bisul ini sering dimulai dari wujud sebagai peradangan folicle rambut kemudian menjalar ke jaringan sekitarnya. Bakteri ini menyerang pada bagian tengah jaringan yang kemudian bakteri tersebut akan mati. Jaringan kulit disekitarnya akan menjadi lebih padat, yang selanjutnya membentuk dinding. Bisul yang membatasi jaringan sentral yang mati yaitu

mata bisul. Folicle rambut dan rambut akan menjadi hancur, sehingga rambut menjadi rontok.



**Gambar 2.4 Bisul (*Furunkulosis*) Pada Kulit Kepala**

## 2. Bisul Batu (*Korbunkulosis*)

Biasanya diawali dengan timbulnya peradangan, terutama pada folicle rambut yang berdekatan sehingga tumbuhlah bisul yang besar dengan mata lebih dari satu. Kelainan ini juga disebabkan oleh bakteri. Kelainan ini sering ditemukan pada penderita diabetes melitus/kencing manis. Apabila kelainan ini mengalami penyembuhan, maka akan meninggalkan bekas jaringan parut dan sering berakibat kebotakan yang permanen.



**Gambar 2.5 Bisul Batu (*Karbunkulosis*)**

### 3. *Dermatitis Papularis Capilliti*

Kelainan ini merupakan peradangan pada folicle rambut, disertai dengan penanahan (supurasi) kulit dibagian belakang kepala. Bisa dilihat secara gambaran klinis berupa pita melintang dibelakang kepala. Pita itu terdiri atas folicle yang meradang dengan bagian tepi-tepi yang kasar dan keras, karena pembentukan jaringan parut berlebihan (*keloid*). Keadaan ini sangat kronis dan disebabkan bakteri *stafilokokus*.



**Gambar 2.6 *Dermatitis Papularis Capilliti***

### 4. Kutu Kepala (*Pediculosis Capitis*)

Merupakan infeksi pada kulit kepala karena gangguan parasit jenis serangga yaitu kutu kepala. Pada dasarnya kutu itu sendiri tidak mempengaruhi pertumbuhan rambut, namun menimbulkan rasa gatal yang hebat pada kulit kepala karena gigitannya. Karena rasa gatal ini, penderita menggaruk-garuk yang akan menyebabkan infeksi sekunder. Pada kasus-kasus yang berat dapat dijumpai adanya abses atau borok yang banyak dijumpai di daerah belakang kepala . rambut didaerah ini kering dan kusam , bahkan dapat bergumpal-gumpal karena nanah yang mengering sehingga berbau busuk.

Telur-telur kutu (nits) melekat pada batang rambut umumnya pada daerah di belakang telinga.



**Gambar 2.7 Kutu Kepala (*Pediculus Capitis*)**

#### 5. Kudis (*Scabies*)

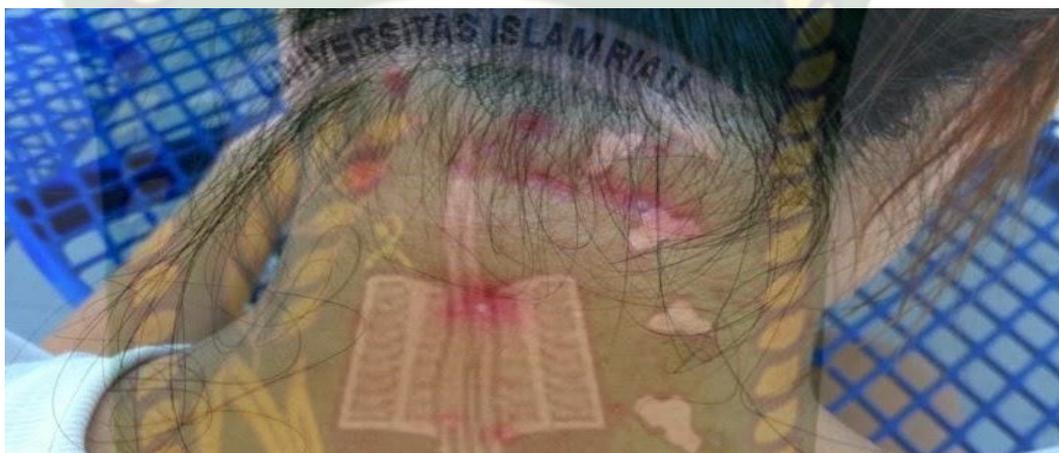
Penyakit kudis adalah penyakit kulit yang disebabkan tungau yang menyerang kulit. Tungau kecil yang disebut *sarcoptes scabiei* yang mendirikan sarang di lapisan luar kulit manusia, inang tungau bertelur didalam kulit, tungau ini menyebabkan gatal tanpa henti dan bentol kemerahan.



**Gambar 2.8 Kudis (*Scabies*)**

## 6. Peradangan Folikel Rambut (*Folikulitis*)

*Folikulitis* merupakan infeksi bakteri di folikel rambut dan menyebabkan pembentukan pustula lubang folikel rambut. Infeksi ini bisa *superfisial* (*impetigo folikular* atau *impetigo bockhart*) atau dalam (*sikosis barbae*), infeksi *folikulitis* dapat dilihat di gambar 2.9.



**Gambar 2.9 Peradangan Folikel Rambut (*Folikulitis*)**

## 7. Psoriasis Kulit Kepala

Yakni salah satu jenis penyakit kulit kepala seperti ketombe, bersisik atau berkerak. Penyebab psoriasis belum diketahui dengan pasti. Namun banyak yang menduga psoriasis disebabkan autoimun. Dimana sistem imun tersebut berbalik menyerang sel kulit sehat. Faktor pemicu psoriasis diantaranya karena konsumsi alkohol, terbakar sinar UV, tergigit serangga, infeksi, dan pengaruh obat-obatan tertentu. Gejala psoriasis kulit kepala dapat ditandai dengan adanya ruam, bersisik tebal, terasa gatal-gatal, sebagian kulit terlihat mengelupas / berguguran seperti ketombe.



**Gambar 2.10 Psoriasis Kulit Kepala**

#### 8. *Dermatitis Seboroik*

Adalah pelepasan sel-sel kulit kepala yang sudah mati secara berlebihan.

Berdasarkan wujudnya dermatitis seboroik terbagi menjadi 2 yaitu :

a. *dermatitis seboroik* kering, dapat dilihat dengan tanda-tanda yaitu adanya sisik berwarna putih hingga kuning kehitam-hitaman, mengkilat serta kering pada kulit kepala. Akibatnya terjadi sensasi gatal luar biasa pada kulit kepala, rambut rontok karena terganggu pertumbuhannya. Penyakit ini kebanyakan disebabkan oleh jamur. Jamur itu jatuh dari atas kepala atau pindahan dari sisir rambut dan berkembangbiak. Apabila daya tahan tubuh menurun, perkembangan jamur ini akan sangat mudah dan cepat. Jamur akan mengganggu fisiologi kulit dan akan menyebabkan proses pembentukan stratum corneum yang lebih cepat sehingga terdapat sisik-sisik ketombe yang bertumpuk. Seandainya kebersihan kulit kepala dan rambut kurang baik ini merupakan faktor yang sangat memudahkan untuk berkembangbiaknya jamur tersebut.

b. *dermatitis seboroik* basah sangat banyak terjadi pada orang yang memiliki jenis kulit kepala dan rambut berminyak dan kurang memperhatikan kebersihan sehingga kadang-kadang dermatitis seboroik basah ini agak berbau dibandingkan dengan dermatitis seboroik kering.



**Gambar 2.11 Sindap (*Dermatitis Seboroik*)**

#### 9. *Tinea Capitis*

*Tinea capitis* yakni salah satu penyakit yang sering menyerang kulit kepala. Jamur penyebab *tinea capitis* yakni *Trichophyton/Microsporum*. Umumnya *tinea capitis* menyerang pada pusar kulit kepala. Penyakit ini dapat menular pada bagian lain sehingga meninggalkan bekas. Gejala *tinea capitis* ditandai dengan adanya rambut rontok, kulit meradang, berkerak atau bersisik, terasa sakit dan terlihat seperti ketombe yang dapat menyebabkan botak permanen.



**Gambar 2.12 Tinea Capitis**

## 2.6 Akuisisi Pengetahuan

Akuisisi pengetahuan adalah akumulasi, transfer dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan kedalam program komputer. Dalam tahap ini *knowledge engineer* berusaha menyerap pengetahuan untuk selanjutnya di transfer ke dalam basis pengetahuan. Pengetahuan diperoleh dari pakar dilengkapi dengan buku, basis data, laporan penelitian dan pengalaman pemakai.

Wawancara adalah metode akuisisi yang paling banyak digunakan. Metode ini melibatkan pembicaraan dengan pakar secara langsung dalam suatu wawancara. Analisis protocol dalam metode akuisisi ini, pakar diminta untuk melakukan suatu pekerjaan dan mengungkapkan proses pemikirannya, dengan menggunakan kata-kata pekerjaan tersebut direkam, dituliskan, dan dianalisis.

## 2.7 Database

*Database* adalah sekumpulan data yang terdiri dari koleksi berbagai file yang berisi informasi, yang disimpan dengan cara tertentu sehingga redundansi atau kondisi yang berlebihan yang tidak perlu dapat dihindarkan. Begitu pula data yang disimpan tersebut tidak tergantung pada aplikasinya dan mampu melayani dari beberapa aplikasi yang berbeda. Komputer berhasil membantu kita untuk menyimpan, mengelola, dan memanfaatkan data ini secara efektif dengan suatu system yang disebut dengan system database. Sistem *database* ialah kombinasi perangkat lunak dan perangkat keras computer yang dipakai untuk melaksanakan pekerjaan-pekerjaan tertentu atas sejumlah besar data.

## 2.8 Data Flow Diagram (DFD)

*Data Flow Diagram* (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama *Bubble chart*, *Bubble diagram*, model proses, diagram alur kerja, atau model fungsi.

DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem.

DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

a. Tujuan DFD

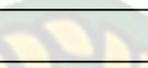
1. Memberikan indikasi bagaimana ada informasi pada satu data bergerak melalui sistem.
2. Menggambarkan fungsi-fungsi yang mentransformasi aliran data.

b. Manfaat DFD

1. *Data Flow Diagram* (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi.
2. DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem.
3. DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

Komponen Data *Flow Diagram* :

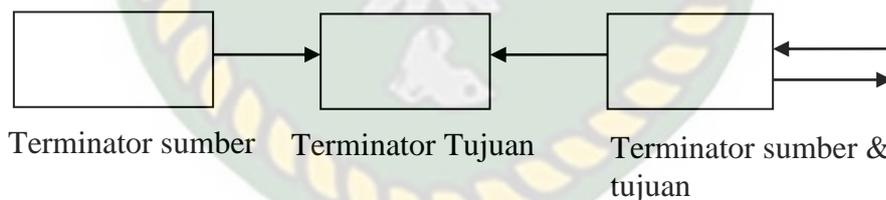
**Tabel 2.4 Komponen DFD Menurut Yordan dan DeMarco**

Terminator	Proses	Data Store	Alur Data
			

### 2.8.1 Komponen Terminator / Entitas Luar

Terminator mewakili entitas eksternal yang berkomunikasi dengan sistem yang sedang dikembangkan. Biasanya terminator dikenal dengan nama entitas luar (*external entity*). Terdapat dua jenis terminator :

1. Terminator Sumber (*source*) : merupakan terminator yang menjadi sumber.
2. Terminator Tujuan (*sink*) : merupakan terminator yang menjadi tujuan data informasi sistem.

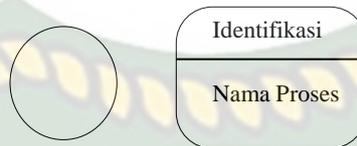


**Gambar 2.13 Komponen Terminator/Entitas Luar**

### 2.8.2 Komponen Proses

Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin, atau komputer dan hasil suatu arus data yang masuk ke dalam proses untuk dilakukan arus data yang akan keluar dari proses. Suatu proses dapat ditunjukkan

dengan simbol lingkaran atau dengan simbol empat persegi panjang tegak dengan sudut-sudutnya tumpul.



Notasi Proses di DFD

### Gambar 2.14 Komponen Proses

Ada beberapa hal yang perlu diperhatikan tentang proses :

1. Proses harus memiliki *input* dan *output*.
2. Proses dapat dihubungkan dengan komponen terminator, data *store* atau proses melalui alur data.

Sistem/bagian/divisi/departemen yang sedang dianalisis oleh profesional sistem digambarkan dengan komponen proses.

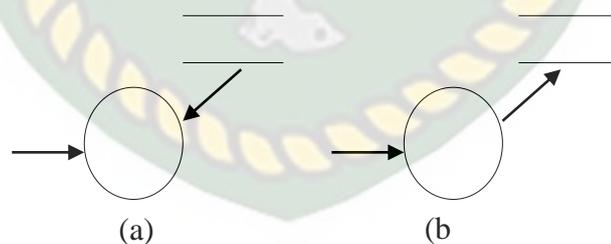
#### 2.8.3 Komponen Proses Data Store

Komponen ini digunakan untuk membuat model sekumpulan paket data dan diberi nama dengan kata benda jamak, misalnya *Mahasiswa*. Data *store* ini biasanya berkaitan dengan penyimpanan-penyimpanan, seperti file atau *database* yang berkaitan dengan penyimpanan secara komputerisasi, misalnya file disket, file *hardisk*, file pita magnetik. Data store juga berkaitan dengan penyimpanan secara manual seperti buku alamat, file folder, dan agenda. Suatu data *store* dihubungkan dengan alur data hanya pada komponen proses, tidak dengan

komponen DFD lainnya. Alur data yang menghubungkan data *store* dengan suatu proses mempunyai pengertian sebagai berikut :

1. Alur data dari data *store* yang berarti sebagai pembacaan atau pengaksesan satu paket tunggal data, lebih dari satu paket data, sebagian dari satu paket tunggal data, atau sebagian dari lebih dari satu paket data untuk suatu proses (*lihat gambar (a)*).
2. Alur data ke data *store* yang berarti sebagai peng-*update*an data, seperti menambah satu paket data baru atau lebih, menghapus satu paket atau lebih, atau mengubah/memodifikasi satu paket data atau lebih (*lihat gambar (b)*).

Pada pengertian pertama jelaslah bahwa data *store* tidak berubah, jika suatu paket data/informasi berpindah dari data *store* ke suatu proses. Sebaliknya pada pengertian kedua data *store* berubah sebagai hasil alur yang memasuki data *store*. Dengan kata lain, proses alur data bertanggung jawab terhadap perubahan yang terjadi pada data *store*.



**Gambar 2.15 Alur Data Ke data Store**

#### 2.8.4 Komponen Data Flow / Alur Data

Suatu data *flow*/alur data digambarkan dengan anak panah, yang menunjukkan arah menuju ke dan keluar dari suatu proses. Alur data ini digunakan untuk menerangkan perpindahan data atau paket data/informasi dari

satu bagian sistem ke bagian lainnya. Selain menunjukkan arah, alur data pada model yang dibuat oleh profesional sistem dapat merepresentasikan *bit*, karakter, pesan, formulir, bilangan real, dan macam-macam informasi yang berkaitan dengan komputer. Alur data juga dapat merepresentasikan data/informasi yang tidak berkaitan dengan komputer. Alur data perlu diberi nama sesuai dengan data/informasi yang dimaksud, biasanya pemberian nama pada alur data dilakukan dengan menggunakan kata benda.

## 2.9 *Entity Relationship Diagram (ERD)*

Menurut Octaviani (2010) ERD (Entity Relationship Diagram) adalah sebuah diagram yang secara konseptual memetakan hubungan antar penyimpanan pada diagram DFD di atas. ERD ini digunakan untuk melakukan permodelan terhadap struktur data dan hubungannya. Penggunaan ERD ini dilakukan untuk mengurangi tingkat kerumitan penyusunan sebuah database yang baik. Entity dapat berarti sebuah obyek yang dapat dibedakan dengan obyek lainnya.

Ada dua macam atribut yang dikenakan dalam entity yaitu atribut yang berperan sebagai kunci primer dan atribut deskriptif. Hal ini seperti setiap entity memiliki himpunan yang diperlukan sebuah primary key untuk membedakan anggota-anggota dalam himpunan tersebut. Atribut dapat memiliki sifat-sifat sebagai berikut :

1. Atomic, atomic adalah sifat dari atribut yang menggambarkan bahwa atribut tersebut berisi nilai yang spesifik dan tidak dapat dipecah lagi.

2. Multivalued, sifat ini menandakan atribut ini bisa memiliki lebih dari satu nilai untuk tiap entity tertentu.
3. Composite, atribut yang bersifat komposit adalah atribut yang nilainya adalah gabungan dari beberapa atribut yang bersifat atomik.

Menurut Octaviani (2010) ada beberapa derajat relasi yang dapat terjadi yaitu:

- a. Hubungan Satu – ke – Satu (one to one) Menggambarkan bahwa antara 1 anggota entity A hanya dapat berhubungan dengan 1 anggota entity B. Biasanya derajat relasi ini digambarkan dengan simbol 1-1.
- b. Hubungan satu – ke – banyak (one to many) Menggambarkan bahwa 1 anggota entity A dapat memiliki hubungan dengan lebih dari 1 anggota entity B. Biasanya derajat relasi ini digambarkan dengan simbol 1-N.
- c. Hubungan Banyak – ke – banyak (many to many) Menggambarkan bahwa lebih dari satu anggota A dapat memiliki hubungan dengan lebih dari satu anggota entity B. Simbol yang digunakan adalah N-N.

Simbol dan Notasi Entity Relational Diagram (ER Diagram) terlihat pada tabel berikut :

**Tabel 2.5 Simbol dan Notasi ERD**

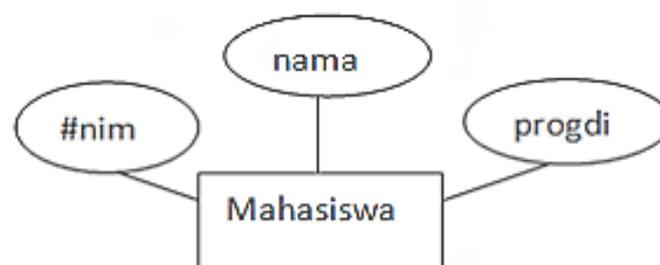
Notasi	Keterangan
	Entitas adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai
	Relasi menunjukkan adanya hubungan diantara sejumlah entitas yang berbeda
	Atribut berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai key diberi garis bawah)
	Garis, sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut

Entitas dapat merupakan :

1. Sebuah elemen lingkungan dari perusahaan , seperti customer atau supplier
2. Suatu sumber daya , seperti suatu piutang dagang, suatu produk, atau suatu penjual
3. Suatu arus informasi , seperti suatu penjualan pemesanan atau suatu faktur

Di dalam sebuah entitas terdapat beberapa atribut. Atribut merupakan gambaran karakteristik dari sebuah entitas atau himpunan entitas.

Contoh : atribut untuk himpunan entitas mahasiswa adalah **nim (PK)**, **nama**, **program studi**.

**Gambar 2.16 Contoh Atribut**

Penggunaan *key* merupakan cara untuk membedakan suatu entitas didalam himpunan entitas dengan entitas lain. *Key* dipilih karena unik, untuk setiap entitas sehingga bisa di bedakan dari entitas yang lain. Kita bisa mendefinisikan *key* sebagai satu atau gabungan dari beberapa atribut yang dapat membedakan semua *row* dalam relasi secara unik.

Ada 3 macam *key* :

1. **Super Key**, *Superkey* yaitu satu atau lebih atribut (kumpulan atribut) yang dapat membedakan setiap baris data dalam sebuah relasi secara unik. Contoh super *key* yaitu =
  1. Nim, nama, alamat, kota
  2. Nim, nama, alamat
  3. Nim, nama
  4. Nim
2. **Candidat Key**, kumpulan atribut minimal yang dapat membedakan setiap baris data dalam sebuah relasi secara unik. Contoh : Nim
3. **Primary Key**, Primary key merupakan salah satu dari candidate key yang terpilih. Alasan pemilihan primary key :
  - a. Lebih sering di jadikan acuan
  - b. Lebih ringkas
  - c. Jaminan keunikan key lebih baik

Contoh dari primary key adalah Nim.

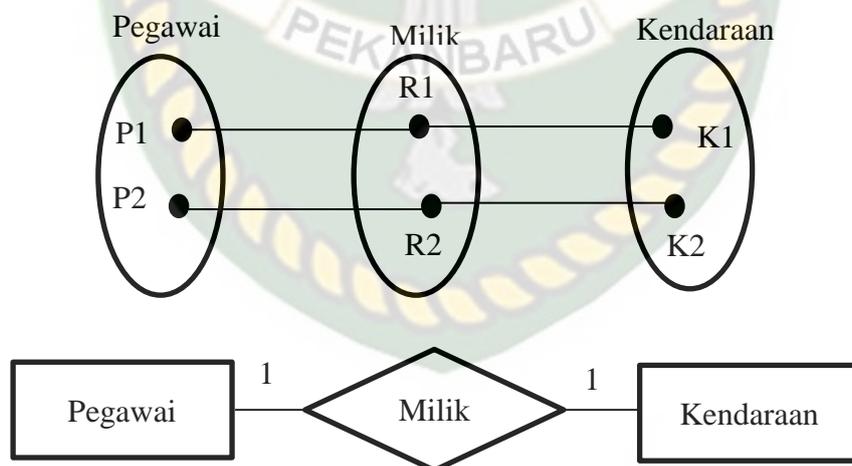
Jika sebuah primary key terhubung ke table/entity lain, maka keberadaan primary key pada entity tersebut di sebut sebagai foreign key ( kunci tamu ).

Misal : Primary Key Kode Dosen dari entity Dosen digunakan juga pada field entity KRS, maka keberadaan field Kode Dosen pada entity KRS disebut sebagai foreign key.

Dalam ERD, hubungan (relasi) dapat terdiri dari sejumlah entitas yang disebut dengan derajat relasi. Derajat relasi maksimum disebut dengan kardinalitas sedangkan derajat minimum disebut dengan modalitas. Jadi kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas lain. Kardinalitas relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dapat berupa :

### 1. Satu ke satu (one to one/ 1-1)

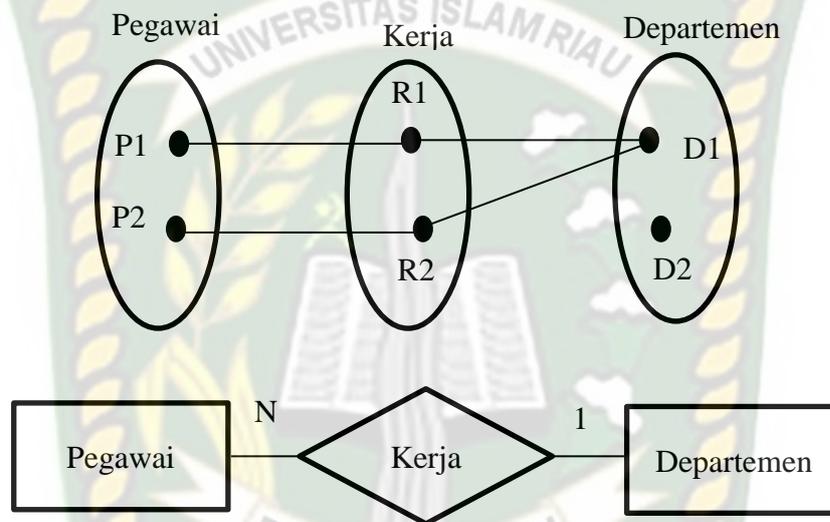
Setiap entitas pada himpunan entitas A dapat berelasi dengan paling banyak satu entitas pada himpunan entitas B, demikian juga sebaliknya.



Gambar 2.17 Relasi Satu Ke Satu

## 2. Satu ke banyak (*one to many/ 1- N*) / N-1

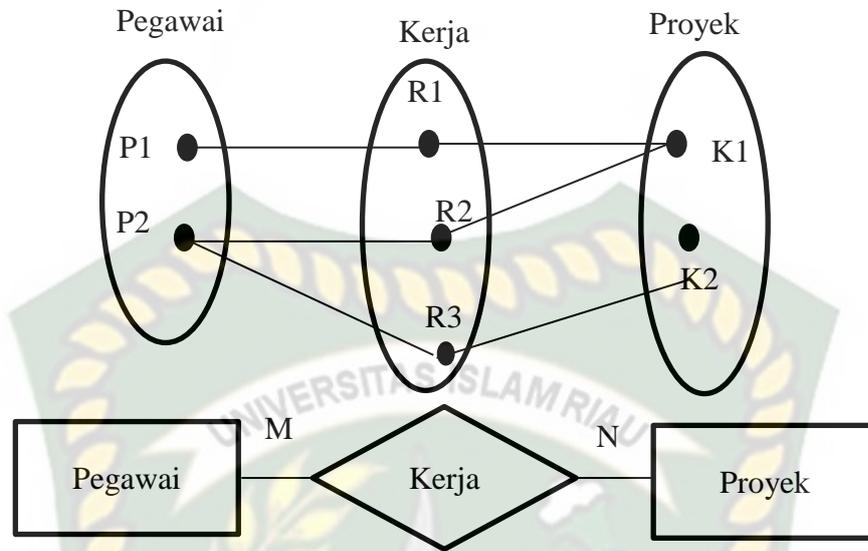
Setiap entitas pada himpunan entitas A dapat berelasi dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya. Atau Setiap entitas pada himpunan entitas A hanya dapat berelasi dengan satu entitas pada himpunan entitas B, tetapi tidak sebaliknya.



Gambar 2.18 Relasi Satu ke Banyak

## 3. Banyak ke banyak (*many to many/ N -N*)

Setiap entitas pada himpunan entitas A dapat berelasi dengan banyak entitas pada himpunan entitas B, demikian juga sebaliknya.



**Gambar 2.18 Relasi Banyak ke Banyak**

### 2.10 Flow Chart

*Flow chart* memiliki 2 model atau jenis yaitu Sistem *Flow chart* dan Program *Flow chart*. *Flow chart* adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flow chart* menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian.

*Flow chart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut. *Flow chart* program dihasilkan dari Flowchart Sistem. Flowchart program merupakan keterangan yang lebih rinci tentang bagaimana setiap langkah program atau prosedur sesungguhnya dilaksanakan.

*Flow chart* ini menunjukkan setiap langkah program atau prosedur dalam urutan yang tepat saat terjadi. Programmer menggunakan *flow chart* program untuk menggambarkan urutan instruksi dari program komputer. Analisis Sistem menggunakan *flow chart* program untuk menggambarkan urutan tugas-tugas pekerjaan dalam suatu prosedur atau operasi (*eWolf Community*).

a. Pedoman-Pedoman Dalam Membuat *Flow chart*

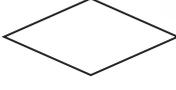
Bila seorang analis dan programmer akan membuat flowchart, ada beberapa petunjuk yang harus diperhatikan, seperti :

1. *Flow chart* digambarkan dari halaman *atas* ke *bawah* dan dari *kiri* ke *kanan*.
2. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat dimengerti oleh pembacanya.
3. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.
4. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja.
5. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
6. Lingkup dan *range* dari aktifitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu digambarkan pada *flow chart* yang sama. Simbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.
7. Gunakan simbol-simbol flowchart yang standar.

## b. Simbol-Simbol Flowchart

Simbol-simbol flowchart yang biasanya dipakai adalah simbol-simbol flowchart standar yang dikeluarkan oleh ANSI dan ISO.

**Table 2.6 Simbol-simbol *Flow chart***

<b>SIMBOL</b>	<b>NAMA</b>	<b>FUNGSI</b>
	<b>TERMINATOR</b>	Permulaan/akhir program
	<b>GARIS ALIR (FLOW LINE)</b>	Arah aliran program
	<b>PREPARATION</b>	Proses inisialisasi/pemberian harga awal
	<b>PROSES</b>	Proses perhitungan/proses pengolahan data
	<b>user/OUTPUT DATA</b>	Proses <i>input/output</i> data, parameter, informasi
	<b>PREDEFINED PROCESS (SUB PROGRAM)</b>	Permulaan sub program/proses menjalankan sub program
	<b>DECISION</b>	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	<b>ON PAGE CONNECTOR</b>	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	<b>OFF PAGE CONNECTOR</b>	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

Sumber : Erhans, 1996, *Membuat Program Dengan dBASE III + Melalui Logika Flow chart.*