### 2.6.3 Cacar Monyet

Istilah cacar monyet memang relatif tidak sepopuler istilah cacar lainnya yang telah disebutkan di atas. Penyakit ini nama ilmiahnya adalah *impetigo bulosa*, atau ada pula yang menamakan impetigo vesikulo-bulosa. Berbeda dengan jenis cacar lainnya yang disebabkan karena infeksi virus, cacar monyet ini disebabkan oleh bakteri *Staphylococcus aureus*. Secara klinis, penderita tidak mengalami demam ataupun gejala umum seperti pada cacar air ataupun *herpes zoster*. Gejala yang didapatkan adalah adanya gelembung yang munculnya terutama di ketiak, dada, dan punggung. Gelembung yang muncul ini cepat pecah dan jumlahnya tidak begitu banyak, namun kerap kali disertai pula oleh miliaria (biang keringat). Penyakit ini memang tidaklah seberat penyakit lainnya, karena terbatas pada lapisan kulit saja. Namun tentunya tidak berarti tidak perlu diobati. Apalagi karena yang menjadi penyebab adalah bakteri, yang untuk memusnahkannya diperlukan obat antibiotika yang dioleskan pada tempat yang terkena. Ada yang mengatakan, penyakit ini disebut cacar monyet, sebab kelainan yang tampak di kulit memang bagi orang awam sedikit banyak mirip dengan penyakit cacar ( Sumber : Sin[13] ).

di dalam tubuhnya terdapat virus ini, namun kebanyakan orang memang tidak mengalami penyakit *Herpes Zoster*. Hal ini disebabkan daya tahan tubuh yang baik yang dapat menekan virus ini berkembang. Sebaliknya, pada orang yang daya tahannya sedang menurun, tak jarang penyakit ini tiba-tiba muncul menyerang. Gejala yang terjadi pada penyakit ini awalnya hampir sama dengan cacar air, yaitu terjadi demam dan badan terasa pegal-pegal. Selanjutnya sedikit berbeda dengan penyakit cacar air, walaupun virus penyebabnya sama. Pada *Herpes Zoster*, gelembung muncul dalam suatu kelompok yang menyerupai garis lebar dengan dasar kulit kemerahan, yang muncul dari bagian belakang tubuh dan menjalar ke arah depan pada salah satu sisi tubuh. Mungkin karena gambaran kelainan yang seperti gambar ular ini, maka ada yang menemakannya cacar ular. Sebenarnya gelembung ini bisa muncul di bagian tubuh mana saja, termasuk wajah, namun yang paling sering adalah dari punggung ke bagian dada. Ada mitos yang mengatakan, bila deretan gelembung muncul dari kedua sisi tubuh, dan kedua ujungnya bertemu, maka akan fatal akibatnya. Mitos ini tidaklah tepat, namun ada unsur benarnya juga. Yang jelas, deretan gelembung memang umumnya muncul hanya di salah satusisisaja .Bila sampai muncul di kedua sisi, berarti infeksi yang terjadi sangat berat, dan daya tahan tubuh penderita dalam keadaan sangat lemah dan buruk. Tentunya kondisi fisik yang demikian ini memang memiliki risiko yang bisa berakibat fatal. Walaupun jarang, kasus seperti ini dapat dijumpai pada penderita yang mendapat terapi imunosupresan (penekanan sistem kekebalan tubuh) dosis tinggi dalam jangka panjang atau pada penderita HIV / AIDS ( Sumber Sin :[13]) .

bernama virus *varisela-zoster*. Serupa dengan cacar, gejala yang muncul sama-sama ada demam. Akan tetapi perbedaan terdapat pada gelembung yang muncul kecil-kecil dan tidak serentak, yang dimulai dari bagian tubuh penderita lalu menjalar keanggota tubuh lainnya. Secara umum, penyakit cacar air ini jauh lebih ringan dan tidak seberbahaya penyakit cacar. Vaksinasi penyakit ini sesungguhnya sudah ada cukup lama, namun hingga kini belum banyak dilakukan di Indonesia. Vaksinasi cacar air sampai hari ini belum menjadi bagian dari program imunisasi dasar yang diwajibkan, mengingat biayanya yang masih mahal sehingga tidak semua orang mampu menjangkaunya. Ikatan Dokter Anak Indonesia (IDAI) saat ini menjadwalkan pemberian imunisasi cacar air ini bagi anak usia 10 tahun ke atas, bila memang belum terkena cacar air. Namun jika dikehendaki dan memang mampu, vaksin sudah boleh diberikan setelah usia 1 tahun, dan diulang 10 tahun kemudian untuk melindunginya saat dewasa.

### 2.6.2 Cacar Ular

Walaupun namanya cacar ular, penyakit ini tidak disebabkan oleh ular. Cacar ular adalah nama awam untuk penyakit *Herpes Zoster*. Penyakit ini merupakan bentuk reaktivasi penyakit cacar air (varisela) yang pernah diderita seseorang sebelumnya. Perlu diketahui, bila seseorang terkena infeksi virus *varisela-zoster* untuk pertama kali, maka akan timbul penyakit cacar air. Setelah sembuh, virus tersebut tidaklah musnah seluruhnya dari tubuh penderita, melainkan berdiam di dalam tubuh penderita, tepatnya di ganglion saraf tepi penderitanya. Virus yang berdiam dalam tubuh penderita ini dapat sewaktu-waktu muncul kembali dan menyebabkan penyakit yang dinamai *Herpes Zoster*. Walau

## 2.6 Penyakit Cacar

Penyakit cacar dalam bahasa medis disebut *variola,* sedangkan dalam bahasa Inggris disebut *small pox*. Penyakit yang disebabkan oleh virus *poks (pox virus)* ini sudah ada sejak berabad-abad yang lalu dan sangat mudah menular. Gejala yang terjadi bagi yang terinfeksi adalah demam, dan muncul gelembung-gelembung berisi nanah secara serentak di kulit daerah wajah, tangan, kaki, dan akhirnya seluruh tubuh. Penyakit ini kerap berakibat fatal, terutama bila mengenai bayi atau lanjut usia. Bagi yang bisa sembuh pun, akan memberikan bekas dikulit berupa bopeng-bopeng. Untungnya, penyakit ini sudah tidak ada lagi dunia. Di abad 19, seluruh dunia berupaya memberantas penyakit ini dengan imunisasi. Di wilayah Indonesia, imunisasi cacar telah dilakukan sejak tahun 1856, oleh Pemerintahan Hindia - Belanda. Setelah proses yang panjang dan penuh kerja keras, akhirnya penyakit ini tidak ditemukan lagi di Indonesia sejak tahun 1974 dan selanjutnya WHO menetapkan Indonesia bebas dari cacar. Berkat kerjasama seluruh dunia, akhirnya tahun 1980 pun dunia dinyatakan sudah bebas dari penyakit cacar. Seiring dengan musnahnya penyakit cacar ini, akhirnya sejak saat itu tidak lagi diperlukan vaksinasi cacar. Berita dan pembahasannya pun makin lama makin menghilang. Tak heran, akhirnya ada juga sebagian orang awam yang tidak pernah mendengar dan asing dengan penyakit cacar ini ( Sumber : sin[13] ).

### 2.6.1 Cacar Air

Cacar air, walaupun namanya mirip dengan cacar, merupakan penyakit yang berbeda. Cacar air, dalam bahasa medisnya disebut *'varisela',* dan dalam bahasa Inggris dinamai *chicken pox*. Penyakit ini disebabkan oleh virus yang

g.*Component diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time, link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*.

h. *Deployment/physical diagram* menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras apa),bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisikal.

## 2.5 Perangkat Lunak Pendukung

### Borland Delphi 7.0

Bahasa pemrograman yang biasa digunakan untuk membangun sistem pakar adalah PROLOG (*PROgramming LOGic*) dan LISP (*LISt Programming*). Namun, dalam penelitian ini penulis menggunakan Borland Delphi 7.0 *Enterprise* sebagai perangkat lunak pendukung, karena dengan menggunakan perangkat lunak ini kita dapat membangun aplikasi dengan cepat dan mudah. Selain itu, Delphi juga menggunakan bahasa pascal yang telah dikenal di berbagai kalangan *programmer*

yang dapat dikerjakan dengan metode-metode tersebut. UML berfokus pada suatu bahasa pemodelan standar bukan pada proses standar.

### 2.4.2 Komponen-komponen UML

Menurut Nug[10] UML memiliki bagian-bagian yang dapat mempermudah pembuatan sistem diantaranya :

a. *Use case diagram, Use case diagram* menggambarkan interaksi antara *actor* dengan proses atau sistem yang dibuat. *Use case diagram* mempunyai beberapa bagian penting seperti: *Actor, Use Case, Undirectional Association , Generalization*

b. Class *diagram*, *class diagram* menggambarkan interaksi antar *class* serta atribut-atribut yang melekat pada *class* tersebut.

c. *Sequence diagram*, *Sequence diagram* merupakan interaksi antara objek-objek dalam suatu sistem dan terjadi komunikasi yang berupa pesan (*message*) serta parameter waktu.

d. *Collaboration Diagram*, *Collaboration Diagram* hampir sama dengan *sequence diagram* tetapi berbeda pada objek yang di titik tekankan, *collaboration* lebih menekankan pada pemunculan objek itu sendiri sedangkan *sequence diagram* lebih pada penyampaian *message* dengan parameter waktu.

e. *Activity diagram*, Activity *diagram* menggambarkan proses-proses yang terjadi mulai aktivitas dimulai sampai aktivitas berhenti.

f. *Statechart Diagram***,** *Statechart Diagram* merupakan diagram yang menggambarkan perubahan *state* dari *state* yang satu ke *state* lainnya.

objek. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya.Model Data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

## 2.4 Pengantar UML

Dalam merancang tugas akhir ini penulis menggunakan UML sebagai alat bantu (*tools*) dalam perancangan dan pendokumentasian sistem yang akan dibuat.

### 2.4.1 Pengertian UML

**Unified Modelling Language** (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

UML Menggabungkan konsep berarah objek seperti Booch, OMT, OOSE, dsb sehingga menjadi bahasa pemodelan tunggal. UML menekankan pada apa

berlawanan dengan bahasa fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas-pertama.

**f.** *Inheritas* adalah mengatur polimorfisme dan enskapsulasi dengan mengijinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada - objek-objek ini dapat membagi (dan memperluas) perilaku mereka tanpa haru mengimplementasi ulang perilaku tersebut (bahasa berbasis-objek tidak selalu memiliki inheritas.)

Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut(terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh anggap kita memiliki sebuah departemen yang memiliki manager, sekretaris, petugas administrasi data dan lainnya. Misal manager tersebut ingin memperoleh data dari bag administrasi maka manager tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bag administrasi untuk mengambilnya. Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas adminiistrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek -objek yang ada karena setiap objek memiliki job descriptionnya sendiri.

### 2.3.2 Metode Beorientasi Objek

Metode Data Berorientasi Objek atau *object-oriented programming (OOP)* merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-

**c.** *Abstract* adalah kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.

**d.** *Enkapsulasi* adalah memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi ijin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

**e**. *Polimorfism*e adalah melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim. Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tungal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini

jika pengujian berjalan baik, pemakai dapat menggunakan sistem pakar pada langkah 7

Pemeliharaan sistem pakar. Sistem pakar harus dipelihara seperti subsistem CBIS yang lain, dan hal ini dapat dilakukan pada langkah 8.

## 2.3 Pendekatan Beorientasi Objek

### 2.3.1 Karakteristik Objek

Menurut Nug[9] Karakteristik objek antara lain :

**a.** ***Class*** adalah kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh 'class of dog' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi object. Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

**b**. *Object* adalah membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer; objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.

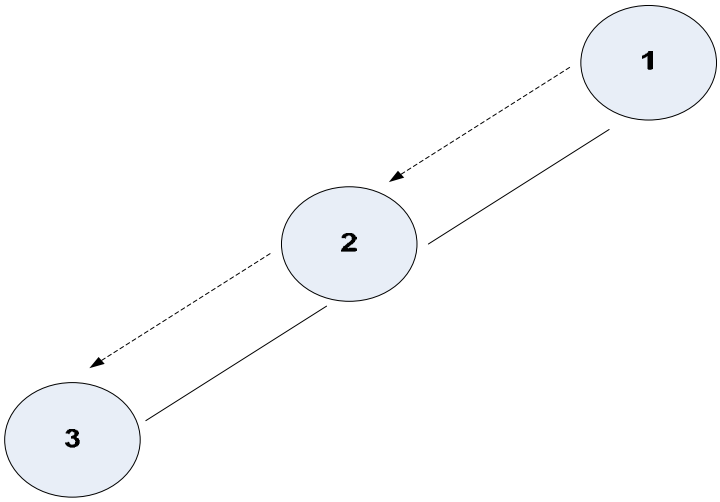
Gambar  tersebut menjelaskan bagaimana analisis sistem bekerja sama dengan pakar untuk memulai proses pengembangan dan mengembangkan prototipe sistem pakar, dan kemudian memperoleh partisipasi pemakai.

Memulai proses pengembangan tugas analisa sistem dalam langkah 1 adalah mempelajari *problem domain* dan menentukan apakah pekerjaan pengembangan harus dilakukan. Jika ya analis menjadi cukup tahu tentang problem domain untuk melanjutkan proyek.

Mengembangkan prototipe sistem pakar. Dalam langkah 2 sampai 5 analis bekerjasama dengan pakar. Dalam beberapa kasus manajer yang mengidentifikasi kebutuhan sistem bertindak sebagai pakar, atau pakar dapat merupakan manajer lain dalam perusahaan atau seseorang dari luar seperti konsultan, atau satu atau beberapa manajer perusahaan lain. Bahkan mungkin perlu meminta bantuan beberapa pakar untuk mendapatkan pengetahuan yang dibutuhkan. Analis dan pakar mengikuti pendekatan sistem dengan pertama – tama mendefinisikan masalah pada langkah 2. pada langkah 3 mereka menemukan perangkat aturan dan menyatukan aturan-aturan tersebut ke dalam prototipe.

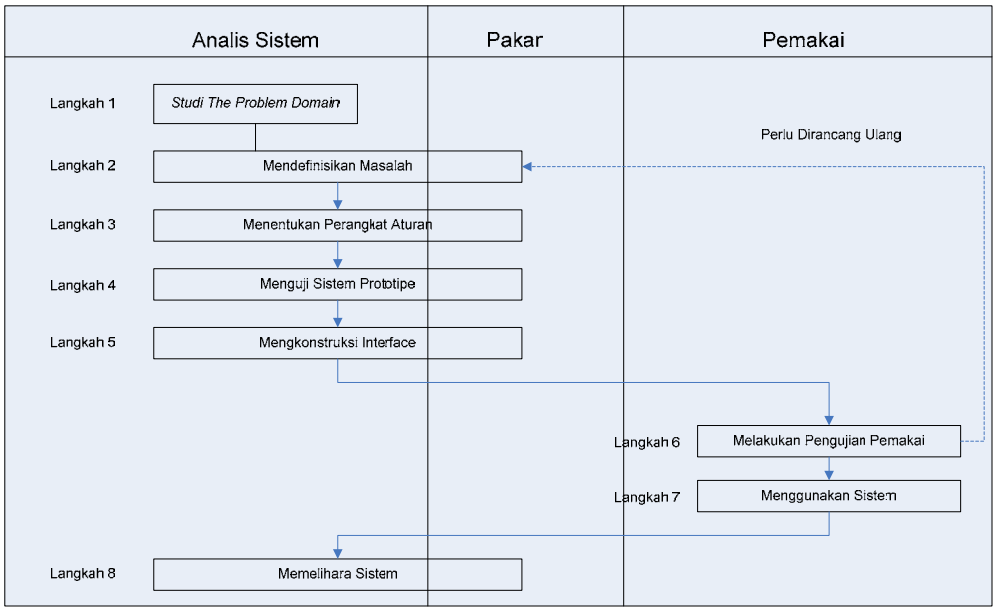
Analis dan pakar menguji prototipe pada langkah 4, dan jika pengujian tidak berjalan baik prototipe diperbaiki, dimulai lagi dengan langkah 2. jika pengujian berjalan seperti yang diharapkan, analis dan pakar menambahkan *user interface* pada langkah 5

Partisipasi pemakai. Selanjutnya pemakai perlu melakukan pengujian pada prototipe. Sistem analis dan pakar memberikan pelatihan yang diperlukan bagi pemakai dan pada langkah 6 pengujian dilakukan. Jika pemakai tidak puas dengan hasilnya, analis dan pakar melakukan pendekatan baru dimulai dengan langkah 2




**Gambar 2.4 Pelacakan pertama mendalam (*depth first search*).**
**( Sumber : Kus [7] )**

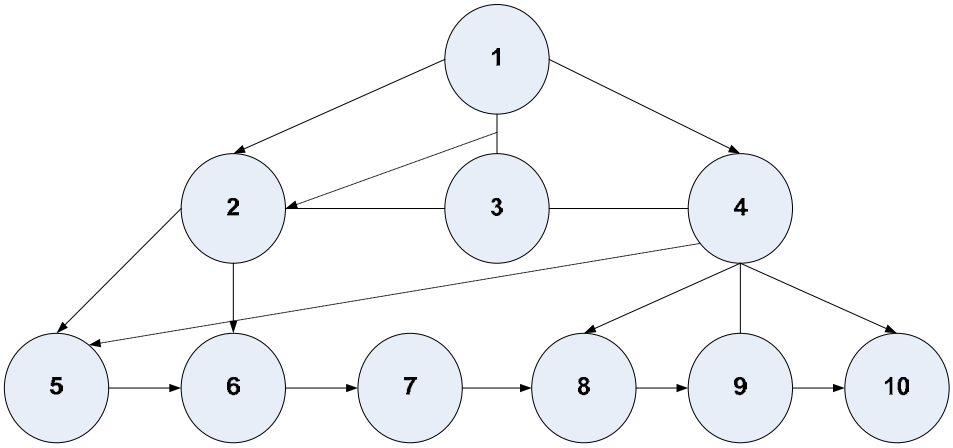
### 2.2.6 Tahap – Tahap Perancangan Sistem Pakar




**Gambar 2.5 Tahap – tahap perancangan sistem pakar**
**( Sumber : Jr [5] )**

a. Pelacakan melebar pertama (*breadth first search*)

Pelacakan (*breadth first search*) menguji semua noda dalam pohon pelacakan. Noda yang ada pada setiap tingkat seluruhnya diuji sebelum pindah ke tingkat berikutnya. Pelacakan *breadth first search* sederhana digambarkan dalam gambar




**Gambar 2.3 Pelacakan melebar pertama (*breadth first search*)**
**( Sumber : Kus [7] )**

Pelacakan ini berangkat dari noda awal dan pertama menurun, kemudian menelusur dari kiri ke kanan

b. Pelacakan pertama mendalam (*depth first search*)

Proses pencarian akan dilakukan pada semua level anaknya sebelum dilakukan pencarian ke node – node yang selevel, pencarian dimulai dari node akar ke level yang lebih tinggi. Proses ini diulangi terus sehingga diketemukan solusi. Pelacakan depth first search digambarkan sebagai berikut

### 2.2.4 Cara Representasi

Cara representasi dalam sistem pakar Menurut (Sumber : Tur[15] ) terbagi dalam tiga teknik, yaitu:

**a. *Production Rule***

*Production Rule* adalah model ide dasar dari sistem yang memopresentasikan pengetahuan dengan bentuk pasangan kondisi aksi (Jika-Maka).

**b. *Semantic Network / Decision Tree* (Pohon keputusan)**

*Semantic Network / Decision Tree* adalah gambaran grafis dari pengetahuan yang terdiri node atau symbol dan hubungan atau link yang memperlihatkan hubungan hirarkis antar objek.

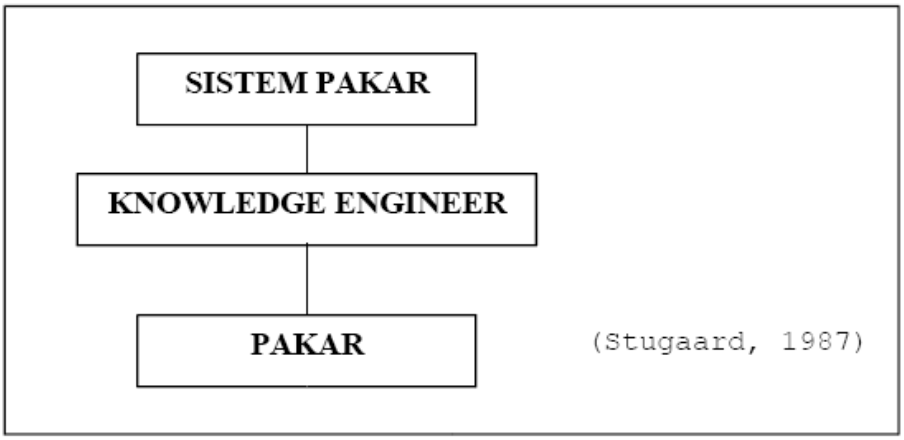
**c. *Frame***

*Frame* adalah struktur data yang berisi semua pengetahuan tentang objek tertentu.

### 2.2.5 Metoda Pencarian dan Pelacakan

Metoda pencarian yang digunakan dalam sistem pakar ini menggunakan *forward chainning* dan teknik pelacakan melebar pertama (*breadth first search*). Namun selain dua teknik tersebut terdapat pula teknik pelacakan pertama mendalam (*depth first search*). Namun dalam pembuatan sistem pakar ini teknik yang digunakan adalah pelacakan melebar pertama (*breadth first search*), dan pelacakan buta (*blind search*).

yaitu seorang spesialis sistem yang menterjemahkan pengetahuan yang dimiliki seorang pakar menjadi pengetahuan yang akan tersimpan dalam basis pengetahuan pada sebuah sistem pakar (gambar 2).




**Gambar 2.2 Knowledge Engineer**
**( Sumber : Stu[14] )**

**b. Modul Konsultasi**

Sistem pakar pada modul konsultasi apabila sistem memberikan konsultasi berupa jawaban atas permasalahan yang diajukan oleh pemakai pada modul ini pemakai yang awam berinteraksi dengan sistem dengan memasukkan data dan jawaban-jawaban pertanyaan sistem. Data yang dimasukkan oleh pemakai ditempatkan dalam database sistem dan kemudian diakses oleh pembangkit inference untuk mendapatkan kesimpulan.

**c. Modul Penjelasan**

Modul Penjelasan adalah menjelaskan proses pengambilan keputusan yang dilakukan oleh sistem

### 2.2.2 Kategori Sistem Pakar

Berikut Merupakan Tabel kategori dari sistem pakar

**Tabel 2.1 Kategori Sistem Pakar**

| Kategori | Masalah yang diselesaikan. |
|---|---|
| Diagnosis | Menduga kegagalan sistem dari observasi Contoh: Mendiagnosa kerusakan mesin mobil. |
| Interprestasi | Menduga gambaran situasi dari observasi. |
| Prediksi | Menduga akibat yang mungkin terjadi dari situasi tertentu. Contoh: Prediksi Cuaca besok berdasarkan data-data sebelumnya. |
| Perencanaan | Mengembangkan rencana-rencana untuk mencapai tujuan tertentu. |
| Monitoring | Membandingkan observasi terhadap rencana hasil. |
| Debugging | Memberikan obat bagi kegagalan fungsi. |
| Instruksi | Mendiagnosa, mununjukkan unjuk kerja. |
| Kontrol | Mengimplementasikan, memprediksi, mengontrol kegiatan yang membutuhkan presisi yang tinggi. |

### 2.2.3 Sistem Kerja Pakar

Menurut Stu [14] sistem kerja pakar terbagi dalam tiga modul yaitu:

**a. Modul Penerimaan Pengetahuan**

Untuk mendapatkan pengetahuan sistem pakar dilakukan proses penerimaan pengetahuan. Proses ini dilakukan melalui interaksi dengan pakar penerimaan pengetahuan dilakukan dengan bantuan *Knowledge Engineer (KE)*,

Dua teknik Inference, yaitu:

**a. *Backward Chaining* (Pelacakan kebelakang)**

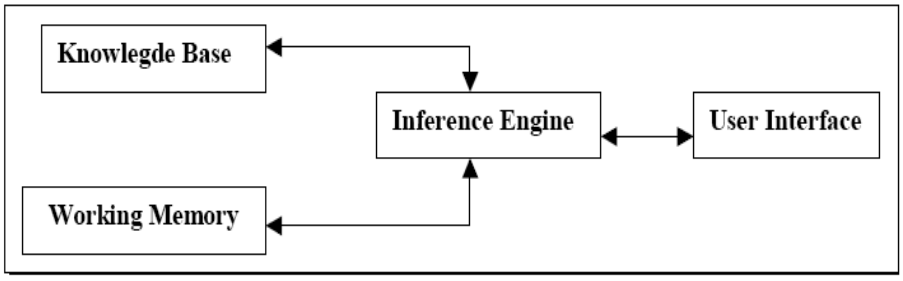
Melalui penalaranya dari sekumpulan hipotesis menuju fakta-fakta yang mendukung tersebut, jadi proses pelacakan berjalan mundur dimulai dengan menentukan kesimpulan yang akan dicari baru kemudian fakta-fakta pembangun kesimpulan atau *a Goal Driven*.

**b.*Forward Chaining* (Pelacakan ke depan)**

*Forward Chaining* merupakan kebalikan dari *Backward Chaining* yaitu mulai dari kumpulan data menuju kesimpulan. Suatu kasus kesimpulannya dibangun berdasarkan fakta-fakta yang telah diketahui atau data driven.

4. *User Interface* (Antarmuka Pemakai)

Antarmuka pemakai adalah bagian penghubung antara program sistem pakar dengan pemakai. Pada bagian memungkinkan pengguna untuk memasukkan instruksi dan informasi ke dalam sistem pakar serta menerima penjelasan dan kesimpulan (gambar 1).




**Gambar 2.1 User Interface**

( Sumber : Tur[15] )

**Menurut Tur[16] Komponen sistem pakar terbagi menjadi empat bagian, yaitu:**

1. *Knowledge Base* (Basis Pengetahuan)

Knowledge Base merupakan inti dari program sistem pakar karena basis pengetahuan itu merupakan presentasi pengetahuan atau knowledge representation. Basis pengetahuan adalah sebuah basis data yang menyimpan aturan-aturan tentang suatu domain knowledge/pengetahuan tertentu. Basis pengetahuan ini terdiri dari kumpulan objek beserta aturan dan atributnya (sifat atau cirinya). Contoh : If hewan merupakan sayap dan bertelur then hewan jenis burung.

2. *Working Memory* (Basis Data atau Memori Kerja)

Working memory adalah bagian yang mengandung semua fakta-fakta baik fakta awal pada saat sistem beroperasi maupun fakta-fakta pada saat pengambilan kesimpulan sedang dilaksanakan selama sistem pakar beroperasi basis data berada di dalam memori kerja.

3. *Inference Engine* (Mesin Inferensia)

Inference Engine adalah bagian yang menyediakan mekanisme fungsi berfikir dan pola-pola penalaran sistem yang digunakan oleh seorang pakar.

1. Mekanisme ini akan menganalisa masalah tertentu dan selanjutnya akan mencari jawabanatau kesimpulan yang terbaik.
2. Mesin ini akan dimulai pelacakannya dengan mencocokan kaidah-kaidah dalam basis pengetahuan dengan fakta-fakta yang ada dalam basis data.

disebut sebagai sistem pakar pertama. Hans Moravec mengembangkan kendaraan terkendali komputer pertama untuk mengatasi jalan berintang yang kusut secara mandiri.

Pada tahun 1980-an, jaringan syaraf digunakan secara meluas dengan algoritma perambatan balik, pertama kali diterangkan oleh Paul John Werbos pada 1974. Tahun 1990-an ditandai perolehan besar dalam berbagai bidang AI dan demonstrasi berbagai macam aplikasi. Lebih khusus *Deep Blue*, sebuah komputer permainan catur, mengalahkan Garry Kasparov dalam sebuah pertandingan 6 game yang terkenal pada tahun 1997. DARPA menyatakan bahwa biaya yang disimpan melalui penerapan metode AI untuk unit penjadwalan dalam Perang Teluk pertama telah mengganti seluruh investasi dalam penelitian AI sejak tahun 1950 pada pemerintah AS.

## 2.2 Konsep Sistem pakar

### 2.2.1 Ciri-ciri sistem pakar

Ciri-ciri sistem pakar menurut Tur[15] adalah sebagai berikut:

1. Terbatas pada domain keahlian tertentu.
2. Dapat memberikan penalaran untuk data data yang tidak pasti.
3. Dapat mengemukan rangkaian alasan-alasan yang diberikannya dengan cara yang dapat dipahami.
4. Berdasarkan pada kaidah/Rule tertentu.
5. Dirancang untuk dapat dikembangkan secara bertahap.
6. Keluaranya bersifat anjuran.

Charles Babbage dan Ada Lovelace bekerja pada mesin penghitung mekanis yang dapat diprogram.

Bertrand Russell dan Alfred North Whitehead menerbitkan *Principia Mathematica*, yang merombak logika formal. Warren McCulloch dan Walter Pitts menerbitkan "Kalkulus Logis Gagasan yang tetap ada dalam Aktivitas " pada 1943 yang meletakkan pondasi untuk jaringan syaraf.

Tahun 1950-an adalah periode usaha aktif dalam AI. Program AI pertama yang bekerja ditulis pada 1951 untuk menjalankan mesin Ferranti Mark I di University of Manchester (UK): sebuah program permainan naskah yang ditulis oleh Christopher Strachey dan program permainan catur yang ditulis oleh Dietrich Prinz. John McCarthy membuat istilah "kecerdasan buatan" pada konferensi pertama yang disediakan untuk pokok persoalan ini, pada 1956. Dia juga menemukan bahasa pemrograman Lisp. Alan Turing memperkenalkan "Turing test" sebagai sebuah cara untuk mengoperasionalkan test perilaku cerdas. Joseph Weizenbaum membangun ELIZA, sebuah chatterbot yang menerapkan psikoterapi Rogerian.

Selama tahun 1960-an dan 1970-an, Joel Moses mendemonstrasikan kekuatan pertimbangan simbolis untuk mengintegrasikan masalah di dalam program *Macsyma*, program berbasis pengetahuan yang sukses pertama kali dalam bidang matematika. Marvin Minsky dan Seymour Papert menerbitkan *Perceptrons*, yang mendemostrasikan batas jaringan syaraf sederhana dan Alain Colmerauer mengembangkan bahasa komputer *Prolog*. Ted Shortliffe mendemonstrasikan kekuatan sistem berbasis aturan untuk representasi pengetahuan dan inferensi dalam diagnosa dan terapi medis yang kadangkala

simbolis, AI yang tak teratur dan perhitungan lunak. Metoda-metoda pokok AI menurut wik [16] meliputi:

1) Jaringan Syaraf: sistem dengan kemampuan pengenalan pola yang sangat kuat

2) Sistem Fuzzy: teknik-teknik untuk pertimbangan di bawah ketidakpastian, telah digunakan secara meluas dalam industri modern dan sistem kendali produk konsumen.

3) Komputasi Evolusioner: menerapkan konsep-konsep yang terinspirasi secara biologis seperti populasi, mutasi dan "survival of the fittest" untuk menghasilkan pemecahan masalah yang lebih baik.

Metoda-metoda ini terutama dibagi menjadi algoritma evolusioner (misalnya algoritma genetik) dan kecerdasan berkelompok (misalnya algoritma semut). Dengan sistem cerdas hibrid, percobaan-percobaan dibuat untuk menggabungkan kedua kelompok ini. Aturan inferensi pakar dapat dibangkitkan melalui jaringan syaraf atau aturan produksi dari pembelajaran statistik seperti dalam ACT-R. Sebuah pendekatan baru yang menjanjikan disebutkan bahwa penguatan kecerdasan mencoba untuk mencapai kecerdasan buatan dalam proses pengembangan evolusioner sebagai efek samping dari penguatan kecerdasan manusia melalui teknologi.

### 2.1.2 Sejarah AI (*Artificial Intellegence*)

Pada awal abad 17, René Descartes mengemukakan bahwa tubuh hewan bukanlah apa-apa melainkan hanya mesin-mesin yang rumit. Blaise Pascal menciptakan mesin penghitung digital mekanis pertama pada 1642. Pada 19,

# BAB II

# TINJAUAN PUSTAKA

## 2.1 Konsep Kecerdasan Buatan

### 2.1.1 Dasar Pemikiran Kecerdasan Buatan

Secara garis besar, AI terbagi ke dalam dua faham pemikiran yaitu AI Konvensional dan Kecerdasan Komputasional (CI, *Computational Intelligence*). AI konvensional kebanyakan melibatkan metoda-metoda yang sekarang diklasifiksikan sebagai pembelajaran mesin, yang ditandai dengan formalisme dan analisis statistik. Dikenal juga sebagai AI simbolis, AI logis, AI murni dan AI cara lama (GOFAI, *Good Old Fashioned Artificial Intelligence*). Metoda-metodanya menurut wik[16] meliputi:

1) Sistem pakar: menerapkan kapabilitas pertimbangan untuk mencapai kesimpulan. Sebuah sistem pakar dapat memproses sejumlah besar informasi yang diketahui dan menyediakan kesimpulan-kesimpulan berdasarkan pada informasi-informasi tersebut.
2) Petimbangan berdasar kasus
3) Jaringan Bayesian
4) AI berdasarkan tingkah laku: metoda modular pada pembentukan sistem AI secara manual

Kecerdasan komputasional melibatkan pengembangan atau pembelajaran iteratif (misalnya penalaan parameter seperti dalam sistem koneksionis. Pembelajaran ini berdasarkan pada data empiris dan diasosiasikan dengan AI non-