**BAB II**

**DASAR TEORI**

* 1. **Pengertian Sistem Pakar**

Secara umum, sistem pakar (*expert system*) adalah bagian dari ilmu kecerdasan buatan yang secara spesifik berusaha mengadopsi kepakaran seseorang di bidang tertentu ke dalam suatu sistem atau program komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru cara kerja para ahli. Dengan sistem pakar ini, orang awam pun dapat menyelesaikan masalah yang cukup rumit yang sebenernya hanya dapat diselesaikan dengan bantuan para ahli. Bagi para ahli, sistem pakar ini juga akan membantu aktivitasnya sebagai asisten yang sangat berpengalaman

Terdapat beberapa definisi tentang sistem pakar, antara lain :

1. Sistem pakar merupakan salah satu bidang teknik dari kecerdasan buatan yang dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja para pakar atau ahli (Endang Lestari dan Emilya Ully Artha, 2017)
2. Sistem pakar (expert system) adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli (Agus Ramdhani Nugraha dan Andri Sukmaindrayana, 2014)
3. Sistem pakar adalah sistem komputer berbasis pengetahuan yang terpadu di dalam suatu sistem informasi dasar yang ada, sehingga memiliki kemampuan untuk memecahkan berbagai masalah layaknya seorang pakar (Nia Kurniati et al., 2017)
   1. **Beberapa contoh sistem pakar**

Sistem pakar merupakan bagian dalam perkemabangan teknik kecerdasan buatan. Selama beberapa decade yang lalu hingga saat ini telah banyak hasil penelitan dalam bidang sistem pakar yang menghasilkan perangkat lunak. Perangkat lunak ini telah banyak memberikan manfaat bagi pemakainya, beberapa diantaranya adalah :

1. MYCIN

MYCIN dirancang dengan memnggunakan bahasa pemrograman LISP oleh Edward Shortlife dari Standford University pada tahun 1976. Progam MYCIN menyimpan + 500 basis pengetahuan dan basis aturan untuk mendiagnosa penyakit manusia.

1. DENDRAL

DENDRAL dirancang oleh Buchanan dan Feigenbaum pada tahun 1978. Bahasa pemrograman ini dirancang untuk badan antariksa Amerika Serikat, yaitu NASA, dan digunakan untuk penelitian kimia di planet Mars.

1. XCON & XSEL

XCON & XSEL dikembangkan oleh Digital Equipment Corporation (DEC) bekerjasama dengan peneliti dari Carnegie Mellon University (CMU). XCON & XSEL dirancang untuk membantu konfirgurasi sistem komputer besar.

1. SOPHIE

SOPHIE dirancang untuk menganalisa sirkit elektronik.

1. Prospector

Prospector digunakan di dalam geologi untuk membantu mencari dan menemukan deposit.

1. FOLIO

FOLIO dirancang untuk membantu memberikan keputusan bagi seorang manajer dalam hal stok broker dan investasi.

1. DELTA

DELTA dirancang untuk pemeliharaan lokomotif diesel.

* 1. **Ciri dan Karakteristik Sistem Pakar**

Ada berbagai ciri dan karakterisitk yang membedakan sistem pakar dengan sistem yang lain. Ciri dan karakteristik ini adalah pedoman utama dalam pengembangan sistem pakar. Ciri dan karakteristik tersebut antara lain :

1. Pengetahuan sistem pakar merupakan suatu konsep, bukan bentuk numeris. Ini dikarenakan komputer melakukan proses pengolahan data secara numerik sedangkan keahlian dari seorang pakar adalah fakta dan aturan-aturan.
2. Informasi dalam sistem pakar tidak selalu lengkap, subyektif, tidak konsisten, subyek terus berubah dan tergantung pada kondisi lingkungan sehingga keputusan yang diambil bersifat tidak pasti dan tidak mutlak “ya” atau “tidak” akan tetapi menurut kebenaran tertentu.
3. Kemungkinan solusi sistem pakar terhadap suatu permasalahan adalah bervariasi dan mempunyai banyak pilihan jawaban yang dapat diterima, semua factor yang ditelusuri memiliki ruang masalah yang luas dan tidak pasti.
4. Perubahan atau pengembangan pengetahuan dalam sistem pakar dapat terjadi setiap saat bahkan sepanjang waktu sehingga diperlukan kemudahan dalam modifikasi sistem untuk menampung jumlah pengentahuna yang semakin membesar 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 factor subjektif.
6. Keputusan merupakan bagian terpenting dari sistem pakar. Sistem pakar harus memberikan solusi yang akurat berdasakan masukan pengetahuan meskipun solusinya sulit sehingga fasilitas informasi sistem harus selalu diperlukan.
   1. **Kelebihan dan Kelemahan Sistem Pakar**

Menurut Azis (1994) dan Suparman (1991), progam sistem pakar memiliki beberapa kelebihan antara lain:

1. Memungkinkan orang awam bisa bekerja layaknya seorang pakar.
2. Meningkatkan hasil dan produktivitas kerja sebagai akibat dari meningkatnya kualitas kerja dan efisiensi kerja.
3. Mampu mengambil dan melestarikan keahlian para pakar (terutama yang termasuk keahlian langka).
4. Menghemat waktu kerja.
5. Mampu beroperasi dalam lingkungan yang berbahaya.
6. Meningkatkan kapabilitas sistem komputer
7. Menghemat waktu dalam pengambilan keputusan.
8. Pengetahuan dalam sistem pakar dapat dikumpulkan dari beberapa manusia pakar, sehingga pengetahuan yang tersedia lebih lengkap.
9. Dapat bekerja dengan informasi yang kurang lengkap dan mengandung ketidakpastian.

Selain kelebihan-kelebihan di atas, program sistem pakar seperti halnya sistem lainnya, juga memiliki kelemahan-kelemahan. Kelemahan dari sistem pakar menurut Suparman (1991) antara lain :

1. Untuk membuat suatu sistem pakar yang benar-benar berkualitas tinggi sangatlah sulit dan memerlukan biaya yang sangat besar untuk pengembangan dan pemeliharaannya.
2. Masalah dalam mendapatkan pengetahuan di mana pengetahuan tidak selalu bisa didapatkan dengan mudah, karena kadangkala pakar dari masalah yang kita buat tidak ada, dan kalaupun ada terkadang pendekatan yang dimiliki para pakar berbeda-beda.
3. Sistem pakar tidak dapat menjawab dengan baik bila berkaitan maupun bersinggungan dengan hal-hal lain atau dominan dari yang berbeda.
   1. **Konsep Dasar Sistem Pakar**

Menurut Efraim Turban, konsep dasar sistem pakar mengandung: keahlian, ahli, pengalihan keahlian, inferensi, aturan dan kemampuan menjelaskan. (Agus Ramdhani Nugraha dan Andri Sukmaindrayana, 2014)

* + - 1. Keahlian

Keahlian adalah suatu kelebihan penguasaan pengetahuan di bidang tertentu yang diperoleh dari pelatihan, membaca atau pengalaman. Dari penguasaan pengetahuan tersebut memungkinkan para ahli untuk dapat mengambil keputusan lebih cepat dan lebih baik daripada seseorang yang bukan ahli.

* + - 1. Ahli

Ahli adalah sesesorang yang mampu menjelaskan suatu tanggapan, mempelajari hal-hal baru seputar topic permasalahan (domain), menyusun kembali pengetahuan jika dipandang perlu, memecah aturan-aturan jika dibutuhkan, dan menentukan relevan tidaknya keahlian mereka.

* + - 1. Pengalihan Keahlian

Dalam menyelesaikan masalah Sistem Pakar menggunakan penarikan kesimpulan dari fakta dan aturan dari domain sistem yang ada pada sistem. Fakta dan aturan ini dibuat oleh seorang ahli yang kemudian ditransfer ke komputer, dan ditampilkan lagi ke user atau pemakai. Proses pentrafseran keahlian ini membutuhkan beberapa aktivitas yaitu : Pertama, tambahan pengetahuan dari para ahli atau sumber lain. Kedua, representasi pengetahuan. Ketiga, inferensi pengetahuan yaitu kemampuan untuk melakukan penalaran. Keempat, pentransferan pengetahuan dari sistem ke user atau pemakai.

1. *Inferensi*

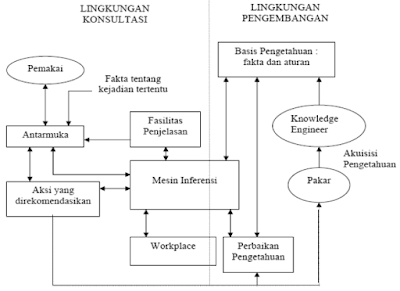
*Inferensi* adalah kemampuan yang dimiliki oleh sisitem untuk menarik kesimpulan atau menyelesaikan masalah. Kemampuan penalaran ini akan ditempatkan dalam mesin *inferensi* dengan memasukkan aturan. Aturan ini berisi pengetahuan yang menggambarkan situasi masalah yang berbentuk prosedur pemecahan masalah.

1. Aturan dan Kemampuan menjelaskan

Sistem pakar dirancang tidak hanya untuk menarik kesimpulan atau menyelesaikan masalah saja. Akan tetapi juga kemampuan merekomendasikan operasi yang dilakukan. Kemampuan untuk merekomendasikan inilah yang membedakan sistem pakar dengan sistem konvensional. Sebagian sistem pakar dibuat dalam bentuk *rule-based system*, yang mana pengetahuan disimpan dalam bentuk aturan-aturan.

* 1. **Struktur Sistem Pakar**

Sistem pakar terdiri dari 2 bagian pokok, yaitu: lingkungan pengembangan (depelopment environment) dan lingkungan konsultasi (consultation environment), (Agus Ramdhani Nugraha dan Andri Sukmaindrayana, 2014). Lingkungan pengembangan sistem pakar digunakan untuk memastikan pengetahuan pakar ke dalam lingkungan sistem pakar, sedangkan lingkungan konsultasi digunakan oleh pengguna bukan pakar guna memperoleh pengetahuan pakar. Komponen-komponen sistem pakar dalam kedua bagian tersebut dapat dilihat dalam gambar 2.1



Gambar 2.1 Struktur Sistem Pakar

Komponen-komponen yang terdapat pada sistem pakar adalah *antarmuka*, *basis pengetahuan*, *akuisisi pengetahuan*, *mesin inferensi*, *workplace*, *fasilitas penjelasan*, dan *perbaikan pengetahuan*.

* + - 1. Antarmuka

Antarmuka merupakan mekanisme yang digunakan oleh pemakai dan sistem pakar untuk berkomunikasi. Antarmuka menerima informasi dari pengguna dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu antarmuka menerima informasi dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pemakai. Menurut McLeod (1995), pada bagian ini terjadi dialog antara program dan pemakai, yang memungkinkan sistem pakar menerima instruksi dan informasi (*input*) dari pemakai, juga memberikan informasi (*output*) kepada pemakai.

* + - 1. Basis Pengetahuan

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi dan penyelesaian masalah, dimana basis pengetahuan ini adalah representasi pengetahuan dari seorang pakar. Metode representasi pengetahuan yang biasa dipergunakan yaitu metode kalkulus predikat (*predicates calculus*), bingkai (*frame*), jaringan semantik (*semantic network*), dan metode *Object-Attribute-Value-Triplets*.

Komponen sistem pakar ini disusun atas dua elemen dasar, yaitu fakta dan aturan. Fakta merupakan informasi tentang obyek dalam area permasalahan tertentu, sedangkan aturan merupakan informasi tentang cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui.

* + - 1. Akuisisi Pengetahuan

Akuisis pengetahuan adalah akumulasi, transfer dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan ke dalam program komputer. Dalam tahap ini knowledge engineer berusaha menyerap pengetahuan untuk selanjutnya ditransfer ke dalam basis pengetahuan.

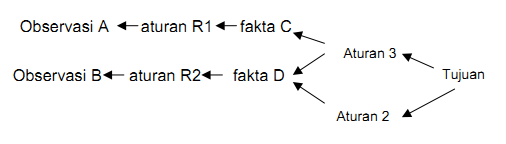
Pengetahuan diperoleh dari pakar, dilengkapi dengan buku, basis data, laporan penelitian dan pengalaman pemakai.

* + - 1. Mesin Inferensi

Mesin inferensi mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah. Mesin inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam workplace dan untuk memformulasi kesimpulan (Fardhian Dwi Saputra, Hindayati Mustafidah dan Suwarno, 2016).

Untuk mengontrol inferensi dalam sistem pakar berbasis aturan digunakan dua model pendekatan, yaitu pelacakan ke belakang (backward chaining) dan pelacakan ke depan (forward chaining).

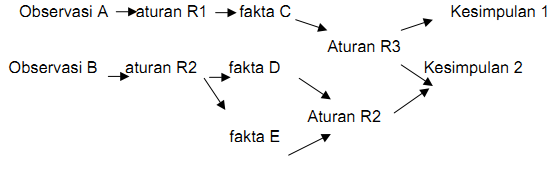
Pelacakan ke belakang adalah pendekatan yang dimotori dengan tujuan (goal-driven), dapat dilihat pada gambar 2.2.



Gambar 2.2 Proses Backward Chaining

Dalam pendekatan ini pelacakan dimulai dari tujuan, selanjutnya dicari aturan yang memiliki tujuan tersebut untuk kesimpulannya.

Sedangkan pelacakan ke depan (forward chaining) adalah pendekatan yang dimotori data (data-driven), dapat dilihat pada gambar 2.3.

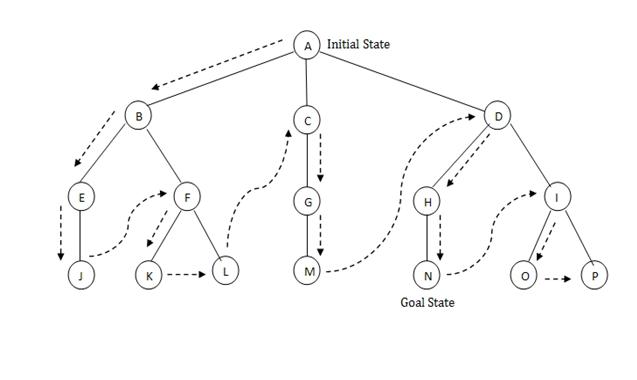


Gambar 2.3 Proses Forward Chaining

Dalam pendekatan ini pelacakan dimulai dari informasi masukkan, dan selanjutnya mencoba untuk mengambil kesimpulan.

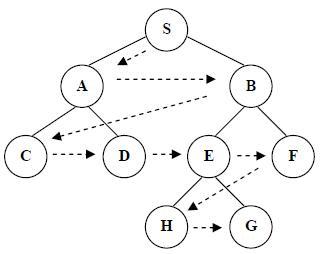
Kedua metode inferensi tersebut dipengaruhi oleh tiga macam teknik penelusuran, yaitu Depth First Search, Breadth First Search dan Best First Search.

Depth First Search adalah teknik penelusuran kaidah secara mendalam dari simpul akar bergerak menurun ke tingkat dalam yang berurutan. Contoh dari penelusuran dengan menggunakan teknik Depth First Search dapat dilihat pada gambar 2.4.



Gambar 2.4 Teknik Penelusuran Depth First Search

Breadth First Search adalah teknik penelusuran pada semua node dalam satu level atau tingkatan diuji terlebih dahulu sebelum pindah ke tingkat selanjutnya. Contoh dari penelusuran dengan menggunakan teknik Breadth First Search dapat dilihat pada gambar 2.5



Gambar 2.5 Teknik Penelusuran Breadth First Search

Best First Search adalah teknik penelusuran yang bekerja berdasarkan kombinasi kedua metode tersebut.

* + - 1. Workplace

Workplace merupakan area dari sekumpulan memori kerja (working memory). Workplace digunakan untuk merekam hasil-hasil dan kesimpulan yang dicapai.

1. Fasilitas Penjelasan

Fasilitas penjelasan adalah komponen tambahan yang akan meningkatkan kemampuan sistem pakar, digunakan untuk melacak respon dan memberikan penjelasan tentang kelakuan sistem pakar secara interaktif melalui pertanyaan.

1. Perbaikan Pengetahuan

Pakar memiliki kemampuan untuk menganalisis dan meningkatkan kinerjanya serta kemampuan untuk belajar dari kinerjanya. Kemampuan tersebut adalah penting dalam pembelajaran terkomputerisasi, sehingga program akan mampu menganalisis penyebab kesuksesan dan kegagalan yang dialaminya.

* 1. **Representasi Pengetahuan**

Representasi pengetahuan merupakan kombinasi sistem berdasarkan dua elemen, yaitu struktur data dan penafsiran prosedur untuk digunakan pengetahuan dalam menyimpan struktur data. Sedangkan pengetahuan itu sendiri dapat diklasifikasikan ke dalam tiga kategori, yaitu :

1. Pengetahuan Prosedural

Pengetahuan prosedural adalah pengetahuan untuk mengetahui bagaimana melakukan sesuatu. Sebagian besar algoritma pemrograman adalah bentuk pengetahuan prosedural, karena mengandung informasi bagaimana menjalankan suatu pekerjaan tertentu.

1. Pengetahuan Deklaratif

Pengetahuan deklaratif adalah pengetahuan untuk mengetahui sesuatu itu benar atau salah, yang strukturnya tersusun atas fakta dan kaidah.

1. Pengetahuan Heuristik

Pengetahuan heuristik adalah pengetahuan yang berbentuk hierarki. Biasanya pengetahuan heuristik ini digambarkan dalam bentuk diagram pohon pengetahuan.

Pengetahuan tersebut perlu dikembangkan karena akan membantu keberhasilan dalam pengembangan sistem pakar. Pengetahuan yang dikembangkan pada dasarnya masih menggunakan bahasa alami dan harus diubah ke dalam bahasa yang dimengerti oleh komputer. Pentransferan pengetahuan dasar ke dalam bahasa yang dapat dimengerti oleh komputer disebut representasi pengetahuan.

Representasi pengetahuan berfungsi sebagai pedoman utama di dalam pengembagan sistem pakar untuk mengambil keputusan dan sebagai pengukur kecerdasan sistem, karena dengan representasi pengetahuan yang lengkap dan mudah dipahami akan membuat sistem lebih pintar dan cerdas.

Dalam membuat representasi pengetahuan terdapat 4 (empat) cara yang dapat digunakan, yaitu :

1. Kalkulus predikat (*predicates calculus*)

Kalukus predikat merupakan cara representasi pengetahuan secara deklaratif. Dalam representasi pengetahuan dengan kalkulus predikat menggunakan simbol yang terdiri dari konstanta, predikat, variabel dan fungsi. Konstanta digunakan untuk merepresentasikan nama spesifikasi obyek. Predikat digunakan untuk menerangkan objek yang telah direpresentasikan. Variable digunakan untuk merepresentasikan kelas umum dari data atau property. Sedangkan fungsi digunakan untuk merepresentasikan pemetaan himpunan entitas ke himpunan lain.

Dalam kalkulus predikat juga menyediakan logika proposisi yang berisi operator logika, seperti AND, OR, NOT, Negasi dan Implikasi. Logika proposisi ini mempunyai 2 (dua) nilai, yaitu Tru yang dinotasikan dengan “T”, dan false yang dinotasikan dengan “F’. Contoh dari logika proposisi dapat dilihat di tabel 2.1.

Tabel 2.1 Table nilai kebenaran

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Pernyataan** | | **AND** | **OR** | **NEGASI** | | **IMPLIKASI** |
| **P** | **Q** | **P ^ Q** | **P v Q** | **P** | **Q** | **P 🡲 Q** |
| **T** | **T** | **T** | **T** | **F** | **F** | **T** |
| **T** | **F** | **F** | **T** | **F** | **T** | **F** |
| **F** | **T** | **F** | **T** | **T** | **F** | **T** |
| **F** | **F** | **F** | **F** | **T** | **T** | **T** |

Selain logika proposisi dalam kalkulus predikat juga terdapat metode kaidah fungsi produksi. Kaidah fungsi produksi tersebut dinotasikan dengan menggunakan aturan *jika – maka*. Penulisan kaidah fungsi produksi ini adalah seperti sebagai berikut ini:

*Jika* pernyataan 1

dan pernyataan 2

dan pernyataan 3

.

.

.

dan pernyataan n

*maka* pernyataan

Dimana bagian *jika* disebut dengan premis dan bagian maka disebut dengan kesimpulan. Jika bagian premis bernilai benar maka bagian kesimpulan juga akan bernilai benar, dan jika bagian premis bernilai salah maka bagian kesimpulan juga akan bernilai salah.

Pernyataan dalam kaidah fungsi produksi ini berupa klausa yang terdiri dari subjek, kata kerja, dan obyek yang menyatakan suatu fakta. Contoh representasi pengetahuan dengan menggunakan kaidah fungsi produksi adalah :

*Jika* kebutuhan drone rakitan untuk kompetisi balap

dan batasan 4 sel baterai

*Maka* drone untuk balap untuk 4 sel baterai

Dari contoh diatas menggambarkan tentang spesifikasi drone sebagai kesimpulan. Kesimpulan akan bernilai benar jika bagian premis yaitu kebutuhan drone rakitan untuk kompetisi balap dan batasan 4 sel baterai bernilai benar.

1. Bingkai (frame)

Bingkai merupakan bentuk representasi pengetahuan yang berupa kumpulan-kumpulan slot-slot yang merupakan atribut untuk mendeskripsikan pengetahuan. Pengetahuan yang termuat dalam slot dapat berubah kejadian, lokasi, situasi ataupun elemen-elemen lain. Dengan menggunakan bingkai, maka akan mempermudah untuk membuat inferensi tentang objek, peristiwa atau situasi baru, karena bingkai menyediakan pangkalan pengetahuan yang ditarik dari berbagai pengalaman.

Contoh representasi pengetahuan dengan menggunakan bingkai dapat dilihat pada gambar 2.6.

Spesifikasi dari : Frame Drone Balap

Kebutuhan : Kompetisi Balap

Ukuran MTM : 230mm

Ukuran Props : 5 inch

Ukuran motor : 2207

Posisi Baterai : Bawah

Ukuran Kamera FPV : Kamera Mini

Gambar 2.6 Contoh Bingkai

1. Jaringan semantic (semantic network)

Jaringan semantik adalah suatu metode representasi pengetahuan dengan menggunakan graf untuk menyatakan node dan hubungan node, dimana node-node mewakili objek-objek dan garis penghubung mewakili hubungan antara objek-objek tersebut (Durkin, 1994).

Struktur jaringan semantik digambarkan secara graf dengan nodes (simpul-simpul) dan arcs (busur-busur sebagai penghubung antar simpul). Nodes seringkali berhubungan dengan obyek, dan arcs berkaitan dengan link yang digunakan untuk menggambarkan dengan jelas hubungan relasi yang ada. Node-node umumnya digunakan secara fisik untuk menunjukkan objek-objek, konsep atau situasi

Hubungan antar node merupakan dasar yang penting dalam jaringan semantik, karena menyajikan struktur perorganisasian pengetahuan. Tanpa hubungan, pengetahuan hanyalah merupakan suatu kumpulan dari fakta yang tidak berelasi dan bermakna. Dengan hubungan, pengetahuna merupakan perpaduan struktur dengan pengetahuan lainnya sehingga dapat diambil kesimpulan.

Contoh representasi pengetahuan dengan menggunakan jaringan semantik terlihat pada gambar 2.7.

Frame

Balap

Ukuran

Motor

Ukuran

Kamera FPV

Ukuran

Prop

Drone Balap

Motor

Kamera FPV

Propeler

Gambar 2.7 Jaringan semantik drone

1. *Object-Attribute-Value Triplets*

*Object-Attribute-Value Triplets* (OAV) adalah sebuah representasi pengetahuan yang dibagi dalam tiga bagian. Bagian tersebut adalah obyek, attribute, dan value. Bentuk umum dari OAV dapat dilihat pada gambar 2.8

Ukuran Diagonal

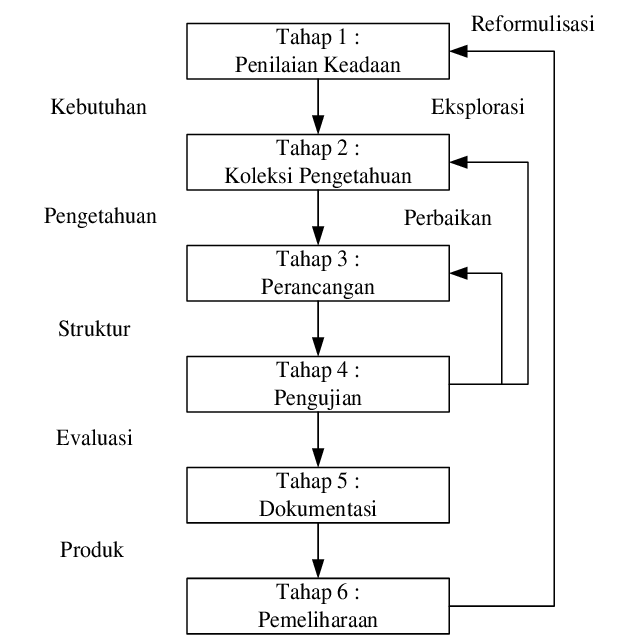
Drone

230mm

Gambar 2.8 Bentuk umum OAV

* 1. **Tahap Pengembanagan Sistem Pakar**

Seperti umumnya, pengembangan perangkat lunak, pada pengembangan sistem pakar ini juga diperlukan beberapa tahapan seperti yang terlihat pada gambar 2.9



Gambar 2.9 Tahap-tahap pengembangan sistem pakar

Secara garis besar pengembangan sistem pakar adalah sebagai berikut :

1. Mengidentifikasi masalah dan kebutuhan. Mengkaji situasi dan memutuskan dengan pasti tentang masalah yang akan dikomputerisasi dan apakah dengan sistem pakar bisa lebih membantu atau tidak.
2. Menentukan masalah yang cocok. Ada beberapa syarat yang harus dipenuhi agar sistem pakar dapat bekerja dengan baik, yaitu:
3. Domain masalah tidak terlalu luas.
4. Kompleksitasnya menengah, artinya jika masalah terlalu mudah (dapat diselesaikan dalam beberapa detik saja) atau masalah yang sangat kompleks seperti prakiraan inflasi tidak perlu menggunakan sistem pakar.
5. Tersedianya ahli.
6. Menghasilkan solusi mental bukan fisik, artinya sistem pakar hanya memberikan anjuran tidak bisa melakukan aktivitas fisik seperti membantu atau merasakan.
7. Persoalan terstruktur dengan baik dan tidak membutuhkan pemikiran rasional (*common sense*). Pemikiran rasional biasanya sulit ditangkap dan disajikan. Bidang-bidang teknis yang tinggi lebih mudah untuk diimplementasikan.
8. Mempertimbangkan alternative. Dalam hal ini ada 2 alternatif yaitu menggunakan sistem pakar atau komputer tradisional.
9. Menghitung pengembalian investasi. Termasuk diantaranya: biaya pembuatan sistem pakar, biaya pemeliharaan, dan biaya training.
10. Memilih alat pengembangan. Bisa digunakan *software* pembuat sistem pakar (seperti: SHELL) atau dirancang dengan bahasa pemrograman sendiri (misalnya: dengan menggunakan PROLOG).
11. Rekayasa Pengetahuan. Perlu dilakukan penyempurnaan terhadap aturan-aturan yang sesuai.
12. Merancang sistem. Bagian ini termasuk pembuatan prototype, serta menterjemahkan pengetahuan menjadi aturan-aturan.
13. Melengkapi pengembangan. Termasuk pengembangan prototype apabila sistem yang telah ada sudah sesuai dengan keinginan.
14. Menguji dan mencari kesalahan sistem.
15. Memelihara sistem. Dalam hal ini harus dilakukan: memperbaharui pengetahuan, mengganti pengetahuan yang sudah ketinggalan, dan memperbaiki sistem agar bisa lebih baik lagi dalam menyelesaikan masalah.