**4**

**KNOWLEDGE REPRESENTATION**

JUMLAH PERTEMUAN : 1 PERTEMUAN

TUJUAN INSTRUKSIONAL KHUSUS :

1. Memahami mengapa representasi pengetahuan diperlukan dalam Artificial Intelligence
2. Memahami cara pengetahuan dipresentasikan dengan baik.
3. Memahami bahwa dengan representasi pengetahuan yang tepat, sebuah masalah dapat diselesaikan.

**MATERI :**

* 1. **PENGERTIAN**

Pengetahuan dari suatu domain merupakan informasi. Informasi dari pengetahuan harus diorganisasikan dalam sebuah *knowledge base* atau basis pengetahuan agar mudah digunakan. Basis pengetahuan dapat ditata dalam sebuah database, dan dapat diorganisasikan menjadi beberapa konfigurasi yang berbeda agar mudah dalam mengambil kesimpulan yang cepat dan penalaran pengetahuan.

Karakeristik umum representasi pengetahuan (Turban,2001) ada dua, yaitu :

1. Representasi pengetahuan dapat diprogramkan dengan beragam bahasa komputer yang ada dan dapat disimpan dalam memori.
2. Representasi pengetahuan dirancang sedemikian rupa sehingga fakta dan pengetahuan lainnya dapat dimanfaatkan untuk penalaran, yaitu basis data yang berisi struktur data yang dapat dimanipulasi dengan suatu sistem inferensi yang menggunakan pencarian dan teknik pencocokan pola pada basis pengetahuan untuk menggambarkan kesimpulan atau bentuk lainnya sebagai suatu fungsi kecerdasan.

Klasifikasi susunan atau pola representasi menurut Mylopoulus dan Levesque, menjadi empat kategori, yaitu :

1. Representasi Logika, menggunakan logika formal untuk merepresentasikan basis pengetahuan. Representasi yang menggunakan logika yang paling banyak digunakan adalah representasi yang menggunakan kalkulus predikat orde pertama.
2. Representasi Prosedural, representasi ini menggambarkan pengetahuan sebagai sekumpulan instruksi untuk memecahkan masalah.
3. Representasi network, representasi ini menangkap pengetahuan sebagai sebuah graph dimana node-nodenya menggambarkan objek atau konsep dalam problema yang dihadapi, sedangkan lengkung-lengkungnya (arc) menggambarkan hubungan atau asosiasi antara mereka. Contoh : jaringan semantic dan graph konseptual.
4. Representasi terstruktur, bahasa-bahasa representasi terstruktur memperluas jaringan dengan membuat setiap nodenya menjadi sebuah struktur data kompleks yang berisi tempat-tempat bernama dengan nilai tertentu. Nilai ini dapat berupa data numerik, atau simbol sederhana, pointer ke frame lain, atau bahkan merupakan prosedur untuk mengerjakan tugas tertentu. Contoh : skrip (script), frame, dan objek.
   1. **LOGIC KNOWLEDGE REPRESENTATION**

Logika adalah salah satu bahasa representasi yang paling tua dari studi kecerdasan buatan. Logika juga merupakan landasan bagi bermacam sistem yang menggunakan logika sebagai alat dalam sistem.

Logika adalah sistem formal dalam bentuk formula atau kalimat yang memiliki nilai benar atau salah. Suatu kalimat logika terdiri dari :

* Sintaks, adalah simbol khusus dalam bentuk bahasa dan dikombinasikan dalam bentuk kalimat. Fakta yang didapat direpresentasikan sebagai kalimat logika.
* Semantik, adalah fakta yang menjadi semesta kalimat, dan bagaimana menandai nilai kebenaran untuk suatu kalimat berdasarkan makna. Fakta adalah penegasan tentang semesta dan kemungkinan bernilai benar atau salah.
* Prosedur pengambilan keputusan, yaitu metode mekanik untuk perhitungan (penurunan) baru (kebenaran) kalimat dari kalimat yang ada.

**Proses Logika**

**OUTPUT**

Inferensi

Atau

konklusi

**INPUT**

Premises

Atau

Fakta

Gambar 4.1 Proses Logika

Gambar 4.1 menggambarkan proses bagaimana informasi diberikan, pernyataan dibuat, atau observasi dicatat. Input disebut premis, yang digunakan oleh proses logika untuk menghasilkan output. Proses ini dapat diketahui kebenaran fakta yang dapat digunakan untuk menurunkan fakta baru yang juga memiliki kebenaran.

Untuk dapat dimanipulasi oleh komputer, logika dibentuk menjadi logika simbolik. Logika simbolik adalah kaidah dan prosedur yang memungkinkan menghasilkan gambaran inferensi dari berbagai premis dengan menggunakan berbagai teknik logika. Ada dua macam bentuk dasar logika, yaitu logika proposisi dan logika predikat.

1. **Logika proposisi**

Logika proposisi adalah pernyataan yang menyatakan nilai benar atau salah. Logika proposisi terdiri dari premis atau konklusi, masing-masing dinyatakan dengan simbol huruf besar alphabet.

Contoh logika proposisi sederhana :

**Pernyataan A** : Karyawan libur setiap hari Minggu

**Pernyataan B** : Hari ini hari Rabu

**Konklusi C** : Karyawan tidak libur hari ini

Contoh diatas contoh sederhana. Pada umumnya proposisi yang sering ditemukan adalah proposisi yang lebih kompleks. Untuk premis yang lebih kompleks digunakan operator penghubung (konjungsi, disjungsi, negasi, implikasi dan biimplikasi).

Nilai benar atau salah sebuah pernyataan dapat dibuat menjadi tabel yang berisi semua kemungkinan. Lebih lanjut lagi dapat dibuat ekuivalensi antara nilai kebenaran beberapa pernyataan.

Contoh tabel kebenaran untuk logika konektif

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **P** | **Q** | **P Q** | **P Q** | **P Q** | **P Q** |
| T | T | T | T | T | T |
| T | F | F | T | F | F |
| F | T | F | T | T | T |
| F | F | F | F | T | T |

1. **Logika predikat**

Logika predikat adalah cara untuk menyelesaikan argumen yang tidak dapat diselesaikan dengan logika proposisi.

Contoh 1.

1. Semua mahasiswa pasti pandai
2. Badu seorang mahasiswa
3. Dengan demikian, Badu pasti pandai

Jika argumen diatas akan dibuktikan validitasnya dengan logika proposisi, dengan mengikuti prosedur logika proposisi dengan menentukan variabel-variabel proposisi.

A = Semua mahasiswa pasti pandai

B = Badu seorang mahasiswa

C = Badu pasti pandai

Selanjutnya akan menjadi :

A premis 1

B premis 2

kesimpulan

Jadi dalam bentuk ekspresi logika akan menjadi

Dengan bentuk ekspresi logika berikut tidak ada hukum logika proposisional yang dapat digunakan untuk membuktikan validitas argumen tersebut. Karena tidak ada yang dapat menghubungkan ketiga proposisi tersebut.

Jadi suatu argumen, memang ada yang tidak dapat ditangani oleh logika proposisional. Oleh karena itu, logika proposisional dikembangkan menjadi logika predikat (*predicate logic*) atau kalkulus predikat atau lengkapnya *first order predicate logic* (fopl).

Contoh 2.

Badu dan Dewi berpacaran

Dalam logika proposisional akan dipecah menjadi dua pernyataan, “Badu berpacaran” dan “Dewi berpacaran “ . Hasilnya tidak diketahui dengan siapa Badu atau Dewi berpacaran.

Dengan logika predikat, kata “berpacaran” adalah predikat, sedangkan Badu dan Dewi adalah entitas yang dihubungkan oleh predikat, disebut *term*. Sebagai pelengkap *term* dan predikat, biasanya menggunakan kuantor (*quantifiers)*, sedangkan prosesnya disebut pengkuantoran (*quantification*). Kuantor mengindikasikan berapa banyak perulangan pada pernyataan tertentu yang bernilai benar. Kuantor terdiri dari :

1. Kuantor universal (*universal quantifiers*) yang mengindikasikan suatu pernyataan selalu benar.
2. Kuantor eksistensial (*existential quantifiers*) yang mengindikasikan bahwa suatu pernyataan kadang-kadang bernilai benar atau mungkin juga salah.

**KOMPONEN-KOMPONEN SINTAK**

1. TERM. Term atau subjek sama seperti kata benda atau kata ganti atau bisa juga disebut objek. Contoh : Badu, manusia, angka, x, dll.
2. PREDIKAT. Properti dari subjek atau sesuatu yang menjelaskan subjek. Contoh : >5, berpacaran, dll.
3. KUANTOR. Kuantor untuk mengindikasikan jika suatu pernyataan bernilai benar, kadang benar atau tidak pernah benar. Contoh : semua, ada, beberapa, sebagian, dll.

Pemberian nilai benar dan salah tergantung pernyataan tersebut dan *universe of discourse* yang membentuknya. Kemudian dilakukan proses pengkuantoran jika dijumpai kata yang menunjukkan jumlah tertentu.

**UNIVERSE OF DISCOURSE**

*Universe of discourse* atau domain digunakan untuk menghilangkan ambiguitas atau penafsiran antara satu dengan lainnya.

**Definisi** : *Universe of discourse* atau domain adalah kumpulan dari semua orang, ide, simbol, struktur data, dan lain-lain yang mempengaruhi argumen logis dengan batasan tertentu. Elemen atau bagian dari *universe of discourse* disebut individual-individual.

Contoh 3.

1. Nani adalah ibu dari Bowo
2. Nani adalah ibu dari Ratna
3. Dua orang siapa saja yang ibunya sama adalah saudara sekandung
4. Dengan demikian, Bowo dan Ratna saudara sekandung

Pada argumen diatas, *universe* *of* *discourse* dapat menunjukan pada orang-orang yang bertempat tinggal sama, satu RT, satu RW, desa kelurahan kabupaten, negara ataupun benua.

Jika argumen berisi angka-angka, domainnya berupa sekumpulan bilangan integer, sekumpulan bilangan asli, sekumpulan bilangan real, dll. Kebenaran suatu pernyataan tergantung dari domain yang dipilih. Pernyataan “ada suatu bilangan pecahan” akan benar pada domain dari bilangan asli, tetapi salah pada bilangan integer.

Elemen dari domain disebut individual-individual. Individual dapat berupa angka, orang, struktur data, dll. Pada umumnya *universe of discourse* harus berisi minimal satu individual. Jadi kumpulan bilangan asli kurang dari 0 tidak memiliki suatu nilai universal karena tidak ada bilangan asli negatif.

Untuk menunjukkan individual digunakan konstanta individual. Jika *universe of discourse* berisi orang-orang, konstanta individual adalah nama-nama orang tersebut. Pada kasus bilangan asli, konstanta individual 0,1,2,3,… konstanta harus unik tidak boleh ada yang sama.

**PREDIKAT**

Predikat menghasilkan penjelasan tentang individual atau subjek dari pernyataan tersebut (term).

Bentuk umum : **PREDIKAT** (individu[obyek]1, individu[obyek]2)

Misalnya proporsisi: Mobil berada dalam garasi

Dinyatakan menjadi:

* Di dalam (mobil, garasi)
  + Di dalam = produk (keterangan)
* Mobil = Argumen (obyek)
  + Garasi = Argumen (obyek)

Contoh lain:

* Proposisi : Rojali suka Juleha
* Kalkulus Predikat : SUKA (Rojali, Juleha)
* Proposisi : Pintu Terbuka
* Kalkulus Predikat : BUKA (pintu)
* Proposisi : Sensor cahaya aktif
* Kalkulus Predikat : AKTIF (sensor cahaya)

Pengetahuan diekspresikan dalam kalkulus predikat yang dapat dimanipulasi agar dapat diinferensi/dinalar. Pangkalan pengetahuan dibentuk dengan menggunakan variabel sebagai simbol-simbol untuk merancang obyek

Misalnya:

x = Rojali

y = Juleha

Maka proporsisinya menjadi Suka(x,y)

Predikat kalkulus membolehkan penggunaan simbol untuk mewakili fungsi-fungsi

Misalnya:

ayah(Juleha) = Jojon

ibu(Rojali) = Dorce

Fungsi dapat digunakan bersamaan dengan predikat. Misalnya predikat berikut menjelaskan bahwa Jojon dan Dorce adalah berteman.

teman(ayah(Juleha),ibu(Rojali))=teman(Jojon,Dorce)

Predikat yang diikuti daftar term disebut rumus atomik. Rumus atomik ini dapat digabung dengan rumus atomik lain dengan perangkai. Misalnya :

Ibu(Ratna,Nani)ibu(Nani, Ratna)

Hasil rumus atomik adalah benar atau salah. Jika predikat memiliki 2 argumen, pemberian nilai dapat diberika dengan baris menunjuk argumen pertama, dan kolom menunjuk argumen kedua. Misalnya pemberian nilai pada predikat “lebih besar dari “ adalah benar jika argumen pertama lebih besar dari argumen kedua. Jadi “lebihbesar(4,3) bernilai T sedangkan “lebihbesar(3,4) bernilai F.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 |
| 1 | F | F | F | F |
| 2 | **T** | F | F | F |
| 3 | **T** | **T** | F | F |
| 4 | **T** | **T** | **T** | F |

* 1. ***SCRIPT***

*Script* merupakan skema representasi pengetahuan yang menggambarkan urutan kejadian. *Script* merepresentasikan situasi, pengetahuan stereotip atau pengetahuan yang didasarkan pada karakteristik yang sudah dikenal dan merupakan pengalaman-pengalaman. Penggambaran urutan peristiwa pada *script* menggunakan serangkaian slot yang berisi informasi tentang orang, objek dan tindakan-tindakan yang terjadi dalam suatu peristiwa.

Elemen-elemen script yang tipikal :

1. Kondisi masukan menggambarkan situasi yang harus dipenuhi sebelum terjadi atau berlaku suatu peristiwa yang ada dalam *script*.
2. Prop mengacu kepada objek yang digunakan dalam urutan peristiwa yang terjadi.
3. Role mengacu kepada orang-orang yang terlibat dalam *script*. Hasilnya adalah kondisi yang ada sesudah peristiwa dalam *script* berlangsung.
4. Track mengacu kepada variasi yang mungkin terjadi dalam *script* tertentu.
5. Scene menggambarkan urutan peristiwa aktual yang terjadi
6. Hasil merupakan kondisi setelah urutan peristiwa dalam *script* terjadi.

Contoh : *script* berangkat ke restoran

|  |  |
| --- | --- |
| **SCRIPT restoran** | |
| Jalur (Track) : | Restoran swalayan (fast food) |
| Peran (Roles) : | Tamu pelayan |
| Pendukung (Prop) : | Counter, baki, makanan, uang, serbet, garam, merica, dll |
| Kondisi masukan : | tamu lapar – tamu punya uang |
| **Adegan (Scene) 1 :** | **Masuk** |
|  | * + Tamu masuk restoran   + Tamu antri   + Tamu baca menu dari daftar menu dan mengambil keputusan tentang apa yang akan dipesan |
| **Adegan 2 :** | **Pesanan** |
|  | * + Tamu memberikan pesanan pada pelayan   + Pelayan mengambil pesanan dan meletakkan makanan di atas baki   + Tamu membayar |
| **Adegan 3 :** | **Makan** |
|  | * Tamu mengambil serbet, sedotan, garam dll * Tamu membawa baki makanan ke meja kosong * Tamu makan dengan cepat |
| **Adegan 4 :** | **Pulang** |
|  | * Tamu merapikan barang-barangnya * Tamu meninggalkan restoran * Tamu naik mobil dan pulang |
| **Hasil :** | * Tamu merasa kenyang * Uang tamu jadi habis * Tamu senang * Tamu kecewa * Tamu sakit perut |

*Script* tersebut menggambarkan situasi tamu yang sedang makan. Adegan - adegan opsional menggambarkan situasi seorang pelayan yang sedang membersihkan meja makan dan bukan menggambarkan orang yang sedang makan.

*Script* dapat digunakan untuk meramalkan yang akan terjadi dalam situasi tertentu. Penggunaan *script* dapat dilakukan dengan menyimpan pengetahuan dalam komputer dalam bentuk simbol, biasanya dengan bahasa LISP atau bahasa pemrograman lain. Saat ada pertanyaan mengenai orang dan kondisi, kemudian terjadi proses pencarian jawaban dengan melacak dan mencocokan pola untuk menguji *script* yang sudah diberikan. Contoh : Apa yang pertama kali dilakukan tamu direstoran swalayan? Yang pertama dilakukan adalah parkir kendaraan. Kejadian-kejadian lain di restoran dapat diramalkan karena sebagian besar kejadian berdasarkan pengalaman-pengalaman sebelumnya.

Kelebihan *script* :

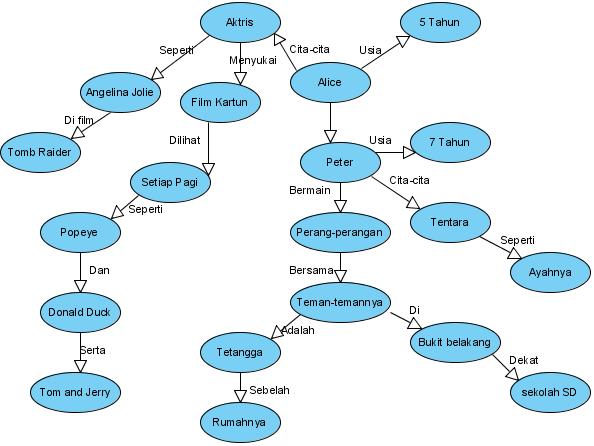
1. *Script* menyediakan beberapa cara yang sangat alami untuk merepresentasikan suatu informasi yang lazim, dengan masalah yang bersumber dari sistem AI.
2. *Script* juga menyediakan struktur hierarki untuk merepresentasikan informasi melalui inklusi subscript dengan script.
3. Struktur representasi pengetahuan lainnya seperti aturan kaidah boleh jadi digunakan dalam cara alami dengan script formal.

Kemampuan *script* :

1. *Script* dapat memprediksi kejadian dan menjawab pertanyaan tentang informasi yang tidak terinci di dalam baris cerita.
2. *Script* menyediakan suatu kerangka kerja untuk mengintegrasikan sekumpulan observasi ke dalam interpretasi yang jelas.
3. *Script* menydiakan skema untuk kejadian yang tidak biasa
   1. **JARINGAN SEMANTIK**

Semua jaringan semantik dibangun dari dua komponen dasar, yaitu :

1. Node-node (simpul) yang merepresentasikan objek, konsep atau situasi. Node ditunjukkan atau diindikasikan oleh kotak atau lingkaran.
2. Arc (busur) merepresentasikan hubungan antara node-node. Arc ditunjukkan oleh panah yang menghubungkan node, dan menunjukkan link atau edge.



Gambar 4.2 Contoh Jaringan Semantik

Gambar 4.2 terdiri dari sejumlah node yang merepresentasikan objek dan menggambarkan informasi tentang objek tersebut. Node atau objek dapat berupa benda, orang, konsep, kejadian, atau tindakan. Atribut suatu objek harus merepresentasikan ukuran, warna, kelas, umur, keaslian, atau karakteristik lainya.

Node dihubungkan oleh link atau arc yang menunjukkan hubungan antara objek-objek yang beragam dan menggambarkan faktor-faktor. Secara umum ada 3 tipe untuk menerangkan link:

1. IS-A (ISA) : adalah
2. KIND OF (AKO) : jenis dari
3. HAS-A : mempunyai

Berikut ini contoh jaringan semantik sederhana. Jaringan sederhana ini dapat diperluas dengan menambah node dan menghubungkan dengan node yang bersesuaikan.

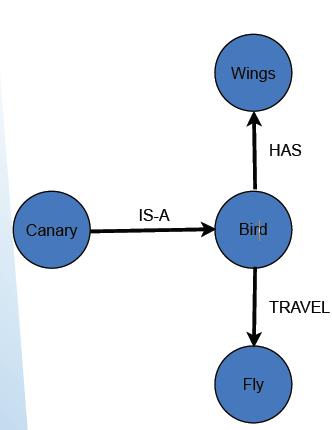
Umumnya penambahan dapat dilakukan dalam 3 cara :

* + Objek yang sama

Penambahan node “Penguin merupakan objek yang sama dengan “Canary” yang merupakan hubungan “IS-A” dari node “Bird”

* + Objek yang lebih khusus

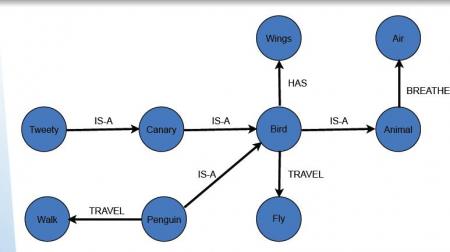
Penambahan node “Tweety” yang merupakan objek khusus dari node “Canary”. Penambahan tersebut memberikan informasi bahwa “Tweety” adalah termasuk “Bird

****

Gambar 4.3 jaringan semantik sederhana

* + Objek yang lebih umum

Objek yang lebih umum dapat dihubungkan dengan arc “IS-A”. contoh : penambahan node “Animal” pada node “Bird” yang menginformasikan bahwa “Bird” termasuk dalam objek “Animal”.



Gambar 4.4 Jaringan semantik yang lebih luas

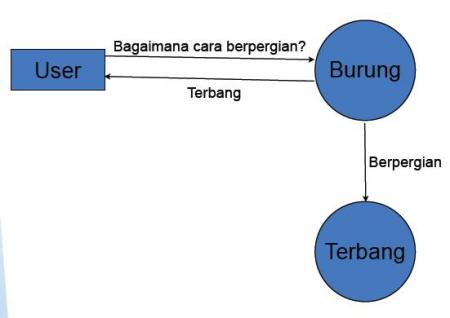
**Operasi pada jaringan semantik**

Salah satu cara mengoperasikan jaringan semantik adalah bertanya pada node.

Kasus 1:

Jika kita bertanya pada “Burung”, “Bagaimana cara anda berpergian?”

Jawabannya “Terbang”



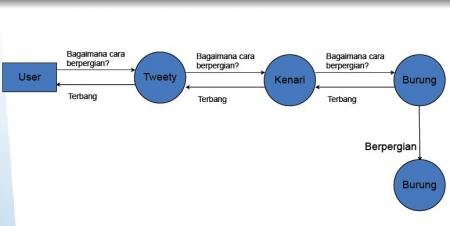
Gambar 4.5 Proses Jaringan Semantik

kasus 2 :

jika bertanya pada “Tweety”, “Bagaimana kamu terbang?”

jawabanya : “Terbang”

pada kasus 2, node tidak menemukan jawaban pada lokal arc maka ia akan mencari pada link dengan hubungan “IS-A” yang dimiliki node tersebut.



Gambar 4.6 jaringan semantik solusi

Kasus 3 :

* Penanganan Pengecualian (Exception handling)

Jika kita bertanya pada “Penguin”, “Bagaimana cara anda berpergian?”

Jawaban normal adalah “Terbang”

* Tidak sesuai dengan fakta

Perlu ditambahkan arc khusus pada node “Penguin” untuk meng-override informasi

**Object-Atributte-Value (OAV)**

Ada 3 hal yang sering digunakan untuk membangun jaringan semantik adalah Objek, Atribut, Value (OAV),

* + Objek dapat berupa bentuk fisik atau konsep
  + Atribut adalah karakteristik atau sifat dari objek tersebut
  + Value (nilai) besaran/nilai/takaran spesifik dari atribut tersebut pada situasi tertentu
  + Dapat berupa numerik, string atau boolean
  + Sebuah objek bisa memiliki beberapa attribute disebut OAV Multi-attribute

Representasi OAV dapat berupa tabel. Bentuk ini cocok untuk daftar pengetahuan. Dalam bentuk tabel diterjemahkan ke dalam kode komputer dengan aturan induksi. Contoh triple OAV dapat dilihat pada tabel berikut :

|  |  |  |
| --- | --- | --- |
| **Objek** | **Atribut** | **Nilai** |
| Apel | Warna | Merah |
| Apel | Tipe | Machintosh |
| Apel | Jumlah | 100 |
| Anggur | Warna | Merah |
| Anggur | Tipe | Seedles |
| Anggur | Jumlah | 500 |

Tujuan penggunaan tripel OAV adalah merepresentasikan fakta dan pola, guna menyesuaikan fakta dalam aturan antecedent.

* 1. **FRAME**

Frame dibuat untuk tujuan memberikan struktur yang berisi informasi umum yang kemudian menjadi suatu keadaan yang khusus. Menurut Minsky (1975), frame dapat dipandang sebagai struktur data statistik yang digunakan untuk merepresentasikan situasi yang telah dipahami dan stereotip. Struktur berupa frame seolah-olah mengatur pengetahuan kita tentang dunia sekitar.

Frame berupa kumpulan slot atribut untuk mendeskripsikan pengetahuan. Pengetahuan yang termuat dalam slot dapat berupa kejadian, lokasi, situasi atau elemen-elemen lain. Frame digunakan untuk representasi pengetahuan deklaratif.

Proses penalaran yang dilakukan oleh frame adalah menginformasikan sebagai harapan (ekspektasi). Jumlah ekspektasi ini mengisi slot dan memeriksa apakah ia sesuai dengan situasi yang berlaku atau tidak. Slot menggambarkan atribut seperti nama pabrik, model, asal-usul pabrik, jenis mobil, jumlah pintu, mesin dan karakteristik lainnya.

Minsky menggambarkan frame sebagai suatu jaringan dari node dan relasi. Level teratas frame menunjukan atribut yang selalu bernilai benar tentang situasi. Level paling bawah mempunyai terminal atau slot yang harus diisi dengan contoh atau data yang spesifik. Terminal dapat mengkhususkan kondisi dengan subframe terkecil. Kumpulan frame dapat dihubungkan membentuk sistem frame. Slot dapat memiliki nilai default untuk slot dalam situasi yang khas. Nilai ini dapat dimodifikasi dengan proses pembatalan.

Contoh frame monitoring pasien dalam ruang operasi (RO) :

Frame Monitoring RO

Spesialisasi dari : monitoring

Tipe : monitoring elektromagnetis, monitoring gula darah,

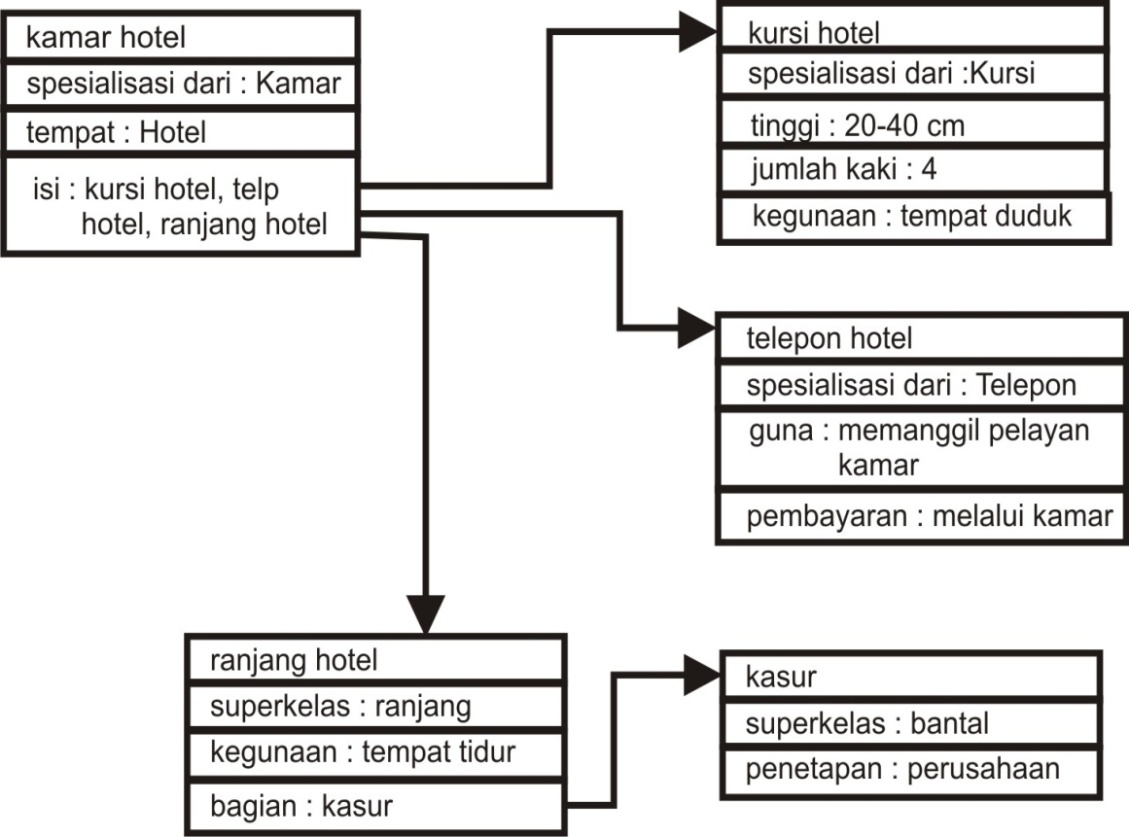
Karakteristik fisik : berat badan, tinggi, umur

Operasi/pembedahan : perut, dada, plastik

Frame individual dapat dipandang sebagai sebuah struktur data yang dalam beberapa hal mirip dengan record tentang informasi yang relevan dengan entitas-entitas stereotip. Slot-slot dalam frame tersebut berisi informasi sebagai berikut :

1. Informasi identitas frame
2. Hubungan frame dengan frame yang lain
3. Penggambaran persyaratan yang dibutuhkan frame
4. Informasi prosedural untuk menggunakan struktur yang digambarkan
5. Informasi default frame
6. Informasi baru

Contoh lain adalah kamar hotel dan komponennya



Gambar 4.5 Deskripsi frame untuk kamar hotel

Secara umum frame memiliki 2 elemen dasar yaitu slot dan facet yang merupakan subslot. Slot merupakan kumpulan atribut yang menjelaskan objek yang dipresentasikan oleh frame dan subslot menjelaskan pengetahuan atau prosedur dari atribut pada slot.

Subslot dapat berbentuk :

* Value : menjelaskan nilai dari suatu atribut
* Default : nilai yang digunakan jika suatu slot kosong atau tidak dideskripsikan pada frame instansiasi.
* Range : menandakan jenis informasi yang dapat muncul pada slot tersebut (cth : 0 s.d 100)
* If added : berisi informasi prosedural yang berupa suatu tindakan yang akan dikerjakan jika nilai dari slot diisi (berubah)
* If needed : facet ini digunakan pada kasus dimana tidak ada value pada slot. Suatu prosedur akan dikerjakan untuk menghasilkan suatu value
* Other : slot bisa berisi frame, rule, jaringan semantik ataupun tipe lain informasi

Contoh frame dalam bentuk kartu mahasiswa.

|  |  |
| --- | --- |
| Nama Mahasiswa |  |
|  |  |
| Alamat |  |
|  |  |
| Mata kuliah | Semester |
|  |  |
|  |  |
|  |  |

Dengan form di atas dapat dibentuk frame dengan struktur desain ditunjukan pada gambar berikut :

|  |  |  |
| --- | --- | --- |
| Nama Frame | Obyek1 | Obyek1 IS-A |
|  |  | Obyek2 |
| Kelas | Obyek 2 |  |
|  |  |  |
| Properti |  |  |
| Properti 1 | Nilai 1 |  |
| Properti 2 | Nilai 2 |  |
| Properti 3 | Nilai 3 |  |

Gambar diatas memiliki tambahan field yang diberi nama Kelas yang akan diisi dengan sebuah nilai (Obyek2) yang merupakan nama frame lain yang berhubungan dengan objek1. Obyek1 IS-A obyek2.

Frame kelas merepresentasikan karakteristik umum dari suatu objek dimana dalam setiap frame tersebut dapat difinisikan properti umum yang biasanya dimiliki oleh semua objek dalam kelas tersebut.

Properti ada dua jenis, yaitu statis dan dinamis. Properti statis merupakan fitur objek yang tidak dapat berubah, sedangkan properti dinamis merupakan fitur yang dapat berubah selama sistem berjalan.

* 1. **SISTEM PRODUKSI**

Konsep sistem produksi atau kaidah produksi telah diperkenalkan oleh Post tahun 1943. Kemudian ditampilkan kembali dalam proses bahasa alami dalam kaidah penulisan Chomsky tahun 1957. Aturan produksi yang diusulkan untuk memodelkan penyelesaian permasalahan tingkah laku manusia oleh Newel dan Simon tahun 1972. Kaidah produksi menjadi acuan yang sering digunakan dalam sistem inferensi, sistem berbasis kaidah dan dalam kasus penyelesaian tingkah laku manusia, produksi sederhana.

Peran sistem produksi serta kekurangan dan kelebihannya :

1. Kekuatan skema representasi pengetahuan. Sistem produksi tidak hanya direpresentasikan dalam pengetahuan tapi juga kegiatan. Arsitektur sistem produksi pada kenyataannya ekivalen dengan bentuk umum mesi Turing dan diidentifikasikan sebagai “Mesin Newell”
2. Jembatan penghubung penelitian AI dengan sistem pakar. Sistem produksi memberikan suatu bahasa yang merupakan representasi dari pengetahuan pakar yang sangat alami.
3. Model heuristik untuk perilaku manusia. Studi mengenai perilaku manusia merupakan protokol asli yang membawa Newell merumuskan konsep produksi. Paradigma mengenai perilaku manusia menyajikan fungsi heuristik yang bernilai dari pemicu penelitian labih lanjut.

**Definisi Sistem Produksi**

Pengetahuan dalam sistem produksi dapat direpresentasikan oleh himpunan kaidah dalam bentuk

IF [kondisi] THEN [aksi]

Dengan sebuah kontrol sistem dan basis data. Untuk sintaks IF\_THEN, kaidah produksi digambarkan sebagai pasangan-pasangan berikut kondisi-aksi, anteseden-konsequent, pola-aksi, situasi-responsi.

Contoh :

IF [mengendarai 60 km per jam melewati batas AND melihat melalui kaca spion ada lampu merah yang berkedip-kedip] THEN [pinggirkan AND berhenti]

IF [Sakit kepala AND sakit tenggorokan AND hidung tersumbat] THEN [ambil aspirin AND istirahat]

Berikut ini bentuk yang mengklasifikasikan suatu objek :

|  |
| --- |
| IF [kategori adalah sebuah form OR sebuah pewarnaan OR sebuah tekstur]  THEN [objek mempunyai sebuah permukaan] |
| IF [kategori adalah sebuah permukaan OR jelek OR kualitas tactile]  THEN [objek mempunya kualitas luar] |
| IF [kategori adalah sebuah ukuran OR kualitas luar OR massa OR suatu zat]  THEN [objek mempunyai kualitas fisik] |

Secara umum terdapat pemisahan secara rapi struktur program dari data. Ada tiga elemen dasar dari semua sistem produksi, yaitu :

1. Database Global

Database global merupakan struktur data utama dari sistem produksi. Database mulai dari sebuah daftar sederhana atau matrks kecil sampai kompleks, relasi dan struktur indeks. Database merupakan struktur dasar kaidah produksi dapat beroperasi. Database juga merupakan struktur dinamis dan berubah-ubah secara kontinu sebagai hasil operasi kaidah produksi.

1. Kaidah produksi

Kidah produksi mempunyai bagian kondisi IF yang disebut bagian kanan dan aksi THEN yang disebut bagian kiri. Jika sisi kiri kadang dinamakan kondisi atau premis yang dipenuhi database maka kaidah-kaidah dapat diterapkan dan subjek menjadi pemicu sistem kontrol.

1. Sistem kontrol

Sistem kontrol merupakan program penerjemah yang esensial untuk mengontrol urutan dimana kaidah-kaidah produksi dipicu dan menyelesaikan konflik jika lebih dari satu kaidah yang diaplikasikan. Sistem kontrol yang secara berulang mengaplikasikan kaidah untuk database hingga sebuah gambaran tujuan dihasilkan. Kemudian sistem akan mendeteksi kejadian seperti tujuan dan record kaidah yang telah diaplikasikan untuk mencapainya sebagai referensi sebelumnya.

**Kaidah Produksi, Pengetahuan, dan Kaidah Inferensi**

Untuk merumuskan beberapa domain pengetahuan secara akurat, diperlukan banyak kaidah produksi. Aturan ini menyediakan rincian objek, karakteristik, dan tindakan-tindakan yang harus diambil. Untuk dapat meliputi suatu objek, diperlukan rincian aturan yang biasanya saling berkaitan dan saling mengacu. Aturan ini membentuk pangkalan pengetahuan yang kemudian menjadi bagian sistem produksi.

Contoh :

Bercak pada daun is sangat banyak

Ujung daun kering is banyak

Variabel linguistik “bercak pada daun” memiliki nilai linguistik “sangat banyak”, sedangkan pada contoh kedua, variabel linguistik “ujung daun kering” memiliki nilai linguistik “banyak”.

Aturan produksi bisa memiliki lebih dari satu proposisi. Proposisi-proposisi tersebut dihubungkan dengan menggunakan operator logika AND atau OR.

Contoh :

Bercak pada daun is sangat banyak

AND ujung daun kering is banyak

Aturan produksi dapat digunakan untuk menyatakan hubungan antara pernyataan lingustik yang masing-masing merupakan bagian premis dan konklusi.

Contoh :

IF bercak pada daun is sangat banyak

AND Ujung daun kering is banyak

THEN penyakit bercak ungu is puso

Menurut Turban (2001) dapat dibuat dalam beberapa bentuk, yaitu :

1. IF Premis THEN Konklusi, misal : IF pendapatan tinggi THEN kemungkinan besar pajak tinggi.
2. Konklusi IF Premis, misal : Kemungkinan besar pajak tinggi IF pendapatan tinggi.
3. Inklusi dari ELSE, misal : IF pendapatan tinggi OR deduksi anda luar biasa THEN kemungkinan besar pajak tinggi OR ELSE kemungkinan besar pajak rendah.
4. Kaidah yang lebih kompleks, misal : IF tingat kredit tinggi AND gaji lebih dari 10 juta OR aset lebih dari 25 juta AND histori pelunasan lancar THEN peminjaman disetujui hingga 5 juta dan daftar pinjaman termasuk dalam kategori B.

Kaidah umum dalam AI adalah : pengetahuan dan inferensi.

Kaidah pengetahuan atau kaidah deklaratif menyatakan semua fakta dan hubungannya tentang suatu masalah.

Kaidah inferensi atau kaidah prosedural pada sisi lain merupakan nasihat atau saran tentang bagaimana menyelesaikan suatu masalah yang diberikan dengan fakta tertentu.

Contoh : kaidah pengetahuan seorang pebisnis bidang penjualan dan pembelian emas.

|  |  |
| --- | --- |
| Kaidah 1 | IF konflik internasional dimulai THEN harga emas naik |
| Kaidah 2 | IF tingkat inflasi mengalami kemunduran THEN harga emas turun |
| Kaidah 3 | IF konflik internasional berlangsung selama lebih dari 7 hari AND IF konflik ini terjadi di Timur Tengah THEN beli emas |

Kaidah inferensi (prosedural) seperti berikut :

|  |  |
| --- | --- |
| Kaidah 1 | IF data dibutuhkan tidak dalam sistem THEN permintaannya dari pengguna |
| Kaidah 2 | IF lebih dari satu kaidah digunakan THEN nonaktifkan kaidah sembarang yang tidak menambah data baru |

Kaidah inferensi berisi kaidah tentang kaidah. Tipe kaidah ini dinamakan kaidah metarule. Mereka menyinggung kaidah lain (atau kaidah yang sama).

Representasi kaidah dapat diaplikasikan bila dibutuhkan untuk merekomendasikan suatu bagian aksi berdasarkan kejadaian yang dapat diobservasi. Berikuti ini beberpa keuntungan menggunakan kaidah, yaitu :

1. Kaidah mudah dimengerti. Mereka mudah disampaikan karena merupakan bentuk alami dari pengetahuan
2. Inferensi dan penjelasan mudah diperoleh atau diturunkan
3. Modifikasi dan perawatan relatif lebih mudah
4. Ketidakpastian lebih mudak dikombinasikan dengan kaidah
5. Setiap kaidah sering tidak saling terkait

Keterbatasan kaidah, yaitu :

1. Pengetahuan yang kompleks membutuhkan ribuan kaidah, yang mungkin agak sukar membuatnya menggunakan sistem maupun perawatannya.
2. Pembangun menyukai kaidah, oleh karena itu mereka mencoba semua kekuatan pengetahuan. Bebrapa program mempunyai kesulitan dalam mengevaluasi sistem berbasis kaidah dan membuat inferensi.
3. Sistem dengan banyak kaidah mempunyai keterbatasan dalam pencarian dalam kontrol program. Beberapa program mempunyai kesulitan dalam mengevaluasi sistem berbasis kaidah dan membuat inferensi.

LATIHAN

1. Jelaskan representasi pengetahuan dan berikan contoh-contohnya
2. Apakah keuntungan utama dari penggabungan kaidah dan frame
3. Buat representasi pengetahuan menggunakan script untuk :
   1. Memesan makanan di restoran swalayan
   2. Membuat program penjumlahan dua bilangan
   3. Berobat ke dokter gigi
4. Cari jurnal ilmiah dengan topik kecerdasan buatan. Buat representasi pengetahuan yang digunakan dalam jurnal tersebut.