

Representasi Pengetahuan: Kuliah Mata Kuliah AI 15 – 22, catatan, slide

www.myreaders.info/, RC Chakraborty, email rcchak@gmail.com , 01 Juni 2010

www.myreaders.info/html/artificial_intelligence.html



Representasi Pengetahuan Masalah, Logika Predikat, Aturan

Kecerdasan Buatan

Representasi pengetahuan dalam AI, topik: perkembangan pengetahuan, model, kategori, peta tipologi, dan hubungan; Pemetaan antara fakta dan representasi, representasi maju dan mundur, KR persyaratan sistem; skema KR – relasional, dapat diwariskan, inferensial, deklaratif dan prosedural; Isu dalam KR - atribut, hubungan, granularitas. Representasi pengetahuan menggunakan logika : Proposisional logika - pernyataan, variabel, simbol, kata sambung, nilai kebenaran, kontingensi, tautologi, kontradiksi, anteseden, konsekuensi, argumen; Logika predikat – predikat, ekspresi logika, kuantifier, rumus; Mewakili hubungan “IsA” dan “Instance”; Fungsi dan predikat yang dapat dihitung. Representasi pengetahuan menggunakan aturan : aturan deklaratif, prosedural, dan meta; Logika karakteristik pemrograman - pernyataan, bahasa, sintaksis, data objek, klausa, predikat, kalimat, subjek, kueri; Pemrograman paradigma – model komputasi, imperatif, fungsional, dan model logika; penalaran, dan konflik.

Representasi Pengetahuan

Masalah, Logika Predikat, Aturan

Kecerdasan Buatan

Topik

(Kuliah 15, 16, 17, 18, 19, 20, 21, 22)

8 jam)

Slide

Tanggal 26-03-2016

1.Representasi Pengetahuan

Pendahuluan – Perkembangan Pengetahuan, model KR, kategori : tipologi peta, jenis, hubungan, kerangka kerja, pemetaan, maju & mundur representasi, persyaratan sistem KR; skema KR - relasional, dapat diwariskan, inferensial, deklaratif, prosedural; masalah KR - atribut, hubungan, granularitas.

2.KR Menggunakan Logika Predikat

27-47

Logika sebagai bahasa; Representasi logika: Logika proposisional, pernyataan, variabel, simbol, konektif, nilai kebenaran, kontingensi, tautologi, kontradiksi, anteseden, konsekuensi, argumen; logika predikat – predikat, ekspresi logika, kuantifier, rumus; Mewakili "IsA" dan hubungan "Instansi"; Fungsi dan predikat yang dapat dihitung; Resolusi.

3.KR Menggunakan Aturan

48-78

Jenis-jenis Aturan - deklaratif, prosedural, meta aturan; Ayat prosedural pengetahuan deklaratif & bahasa; Pemrograman logika – karakteristik, pernyataan, bahasa, sintaksis & terminologi, Komponen data - sederhana & objek data terstruktur, Komponen Program - klausa, predikat, kalimat, subjek, pertanyaan; Paradigma pemrograman – model komputasi, model imperatif, model fungsional, model logika; Penalaran - Rantai maju dan mundur, resolusi konflik; Kontrol pengetahuan.

4.Referensi

79

Representasi Pengetahuan

Masalah, Logika Predikat, Aturan

Bagaimana kita merepresentasikan apa yang kita ketahui?

- **Pengetahuan** adalah istilah umum.

Jawaban atas pertanyaan, "*bagaimana merepresentasikan pengetahuan*", membutuhkan analisis untuk membedakan antara pengetahuan "*Bagaimana*" dan pengetahuan "*itu*".

■ mengetahui "*Bagaimana* untuk melakukan sesuatu".

misalnya "cara mengendarai mobil" adalah Pengetahuan prosedural.

■ mengetahui "*itu* sesuatu itu benar atau salah".

misalnya "itu adalah batas kecepatan untuk mobil di jalan tol" adalah Deklaratif pengetahuan.

- **Pengetahuan dan Representasi** adalah dua entitas yang berbeda. Mereka memainkan peran peran sentral namun dapat dibedakan dalam sistem cerdas.

■ Pengetahuan adalah deskripsi dunia.

Ini menentukan suatu *kompetensi sistem* dengan apa yang diketahuinya.

■ Representasi adalah cara pengetahuan dikodekan.

Ini mendefinisikan sebuah *kinerja sistem* dalam melakukan sesuatu.

- Berbagai jenis pengetahuan memerlukan jenis representasi yang berbeda. Representasi Pengetahuan *model/mekanisme* sering kali didasarkan pada:

◇ **Logika**

◇ **Aturan**

◇ **Bingkai**

◇ **Jaringan Semantik**

- Berbagai jenis pengetahuan memerlukan jenis pengetahuan yang berbeda. *pemikiran*.

1. Pendahuluan

Pengetahuan adalah istilah umum.

Pengetahuan adalah sebuah perkembangan yang dimulai dengan *data* yang kegunaannya terbatas.

Dengan mengorganisasikan atau menganalisis data, kita memahami apa arti data tersebut, dan ini menjadi *informasi*.

Interpretasi atau evaluasi hasil informasi *pengetahuan*.

Pemahaman terhadap prinsip-prinsip yang terkandung dalam pengetahuan adalah *kebijaksanaan*.

• Perkembangan Pengetahuan



Gambar 1 Perkembangan Pengetahuan

■ **Data** dipandang sebagai kumpulan: Contoh : Sedang hujan.

fakta yang tidak berhubungan.

■ **Informasi** muncul ketika: Contoh :

Suhu turun 15

hubungan antar fakta adalah

derajat dan kemudian mulai turun hujan.

ditetapkan dan dipahami;

Memberikan jawaban ke "WHO",

"apa", "dimana", dan "kapan".

■ **Pengetahuan** muncul Kapan : Contoh : Jika kelembabannya sangat tinggi dan suhunya turun secara signifikan, maka atmosfer tidak mungkin menahannya

hubungan di antara pola

diidentifikasi dan dipahami;

kelembaban, sehingga turunlah hujan.

Memberikan jawaban sebagai *"Bagaimana"*.

■ **Kebijaksanaan** adalah puncak dari: Contoh : Meliputi pemahaman

pemahaman, mengungkap

dari semua interaksi yang terjadi

prinsip hubungan yang

antara hujan, penguapan, udara

mendeskripsikan pola.

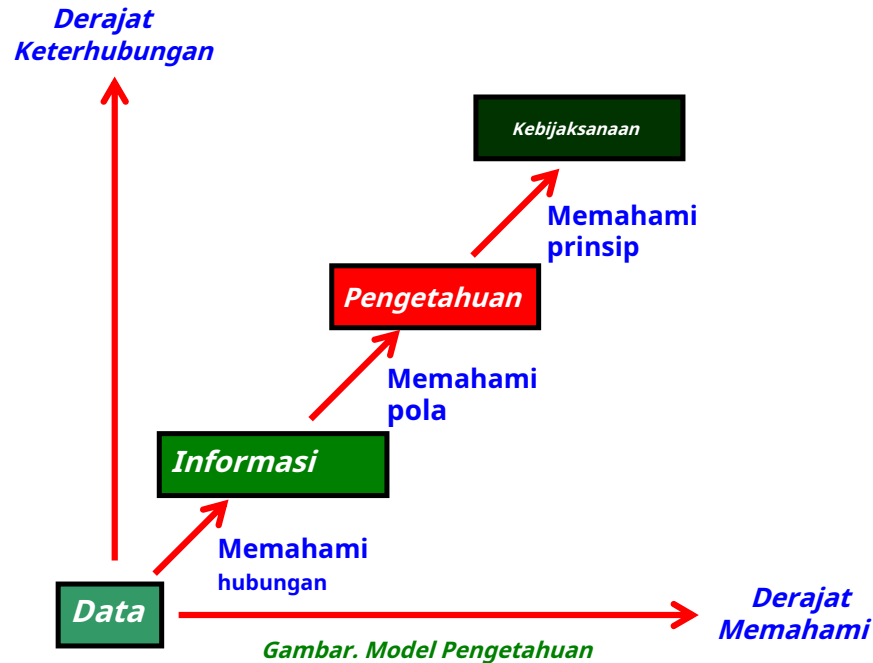
arus, gradien suhu dan

perubahan.

Memberikan jawaban sebagai *"Mengapa"*.

Model Pengetahuan((Bellinger, 1980)

Model pengetahuan menyatakan bahwa ketika tingkat *"keterhubungan"* Dan *"memahami"* meningkat, kita maju dari *data* melalui *informasi* Dan *pengetahuan* ke *kebijaksanaan*.



Gambar. Model Pengetahuan

Model tersebut mewakili *transisi* Dan *memahami*.

itu *transisi* berasal dari *data*, ke *informasi*, ke *pengetahuan*, dan akhirnya

ke *kebijaksanaan* Bahasa Indonesia:

itu *memahami* mendukung transisi dari satu tahap ke tahap berikutnya panggung.

Perbedaan antara *data*, *informasi*, *pengetahuan*, Dan *kebijaksanaan* tidak sangat diskrit. Mereka lebih seperti nuansa abu-abu, daripada hitam dan putih (Shedroff, 2001).

"data" Dan *"informasi"* berurusan dengan *masa lalu*; mereka didasarkan pada pertemuan *Fakta* dan menambahkan konteks.

"pengetahuan" berurusan dengan *hadiah* yang memungkinkan kita untuk tampil.

"kebijaksanaan" berhubungan dengan *masa depan*, memperoleh visi tentang apa yang akan terjadi, lebih tepatnya daripada apa yang ada atau yang telah ada.

Kategori Pengetahuan

Pengetahuan dikategorikan menjadi dua jenis utama: *Diam-diam* Dan *Eksplisit*.

ketentuan "**Diam-diam**" "sesuai dengan ""*tidak resmi*" atau "*implisit*" jenis pengetahuan,

ketentuan "**Eksplisit**" "sesuai dengan "*resmi*" jenis pengetahuan.

Pengetahuan diam-diam

◇ Ada dalam diri manusia;
itu diwujudkan.

◇ Sulit untuk diutarakan secara formal.

◇ Sulit untuk berkomunikasi atau membagikan.

◇ Sulit untuk dicuri atau disalin.

◇ Digambar dari pengalaman,
tindakan, wawasan subjektif.

Pengetahuan eksplisit

◇ Ada di luar manusia;
itu tertanam.

◇ Dapat diartikulasikan secara formal.

◇ Dapat dibagikan, disalin, diproses
dan disimpan.

◇ Mudah dicuri atau disalin

◇ Diambil dari artefak jenis tertentu seperti
prinsip, prosedur, proses,
konsep.

(Slide berikutnya menjelaskan lebih lanjut tentang pengetahuan tacit dan eksplisit).

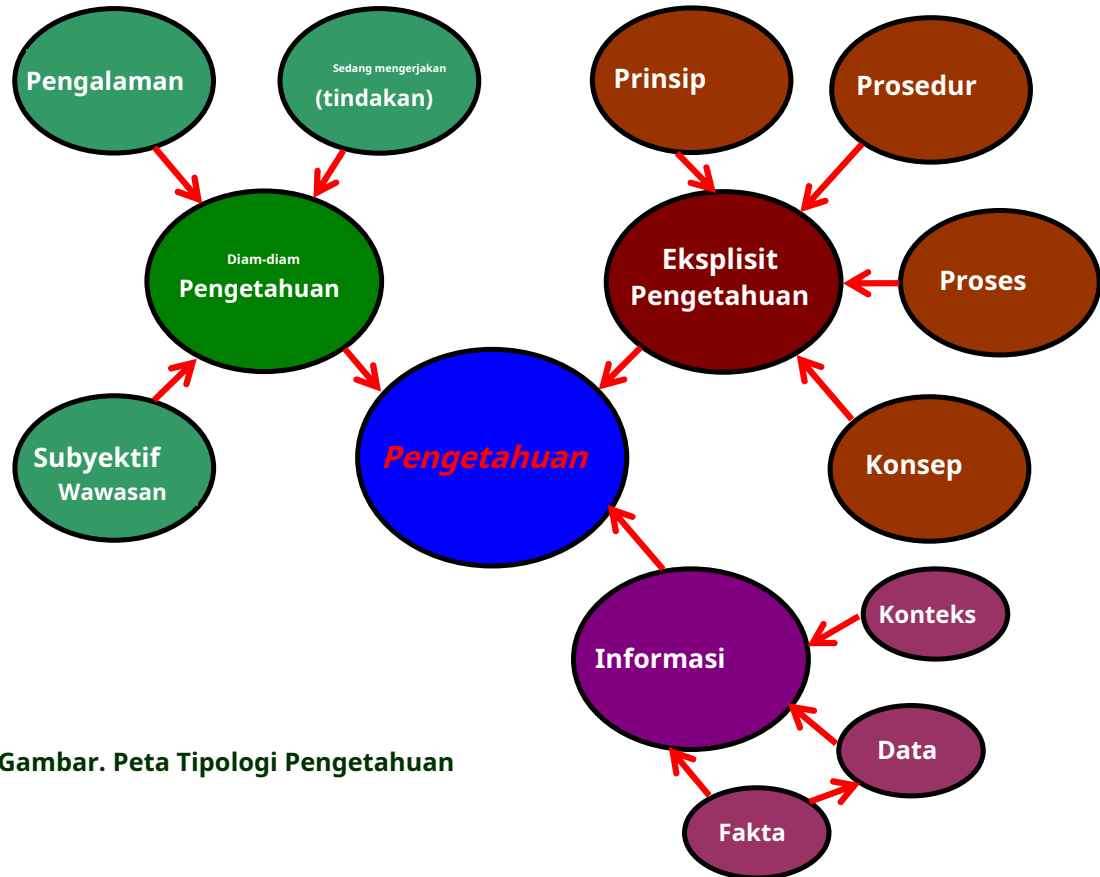
■ Peta Tipologi Pengetahuan

Peta ini menunjukkan dua jenis pengetahuan - *Diam-diam* Dan *Eksplisit* pengetahuan.

Pengetahuan diam-diam berasal dari "pengalaman", "tindakan", "subyektif", "wawasan"

Pengetahuan eksplisit berasal dari "prinsip", "prosedur", "proses",

"konsep", melalui konten yang ditranskripsi atau artefak dari beberapa jenis.



Gambar. Peta Tipologi Pengetahuan

- ◇ **Fakta** :adalah data atau contoh yang bersifat spesifik dan unik.
- ◇ **Konsep** :adalah kelas item, kata, atau ide yang diketahui oleh suatu kelompok nama umum dan memiliki fitur umum.
- ◇ **Proses**:adalah aliran kejadian atau aktivitas yang menggambarkan bagaimana sesuatu bekerja, bukan bagaimana melakukan sesuatu.
- ◇ **Prosedur**:adalah serangkaian tindakan dan keputusan langkah demi langkah yang menghasilkan tercapainya suatu tugas.
- ◇ **Prinsip** :adalah pedoman, aturan, dan parameter yang mengatur; prinsip memungkinkan untuk membuat prediksi dan menarik implikasi;

Artefak ini digunakan dalam proses penciptaan pengetahuan untuk menciptakan dua Jenis-jenis pengetahuan:**deklaratif** Dan **prosedural** dijelaskan di bawah ini.

Jenis Pengetahuan

Psikolog kognitif memilah pengetahuan menjadi **Deklaratif** dan **Prosedural** kategori dan beberapa peneliti menambahkan **Strategis** sebagai kategori ketiga.

#Tentang *pengetahuan prosedural* Bahasa Indonesia: ada beberapa perbedaan pandangan.

- Pertama, ia dekat dengan pengetahuan Tacit, ia terwujud dalam tindakan-tindakan tertentu. sesuatu yang belum dapat diungkapkan dengan kata-kata; misalnya, kita membaca wajah dan suasana hati.
- Yang lain, ini dekat dengan pengetahuan deklaratif; perbedaannya adalah bahwa suatu tugas atau metode dijelaskan, bukan fakta atau benda.

#Semua *pengetahuan deklaratif* adalah pengetahuan eksplisit; itu adalah pengetahuan yang dapat dan telah diartikulasikan.

Itu *pengetahuan strategis* dianggap sebagai bagian dari deklaratif pengetahuan.

Pengetahuan prosedural

◇ Pengetahuan tentang "*bagaimana caranya*

sesuatu"; misalnya, untuk menentukan apakah

Peter atau Robert lebih tua, temukan dulu

usia mereka.

◇ Berfokus pada tugas yang harus diselesaikan

dilakukan untuk mencapai suatu tujuan tertentu

tujuan atau sasaran.

◇ Contoh: prosedur, aturan,

strategi, agenda, model.

Pengetahuan deklaratif

◇ Pengetahuan tentang "*itu*

sesuatu itu benar atau salah". misalnya,

Sebuah mobil memiliki empat ban; Petrus adalah

lebih tua dari Robert;

◇ Merujuk ke representasi dari

objek dan peristiwa; pengetahuan

tentang fakta dan hubungan;

◇ Contoh: konsep, objek,

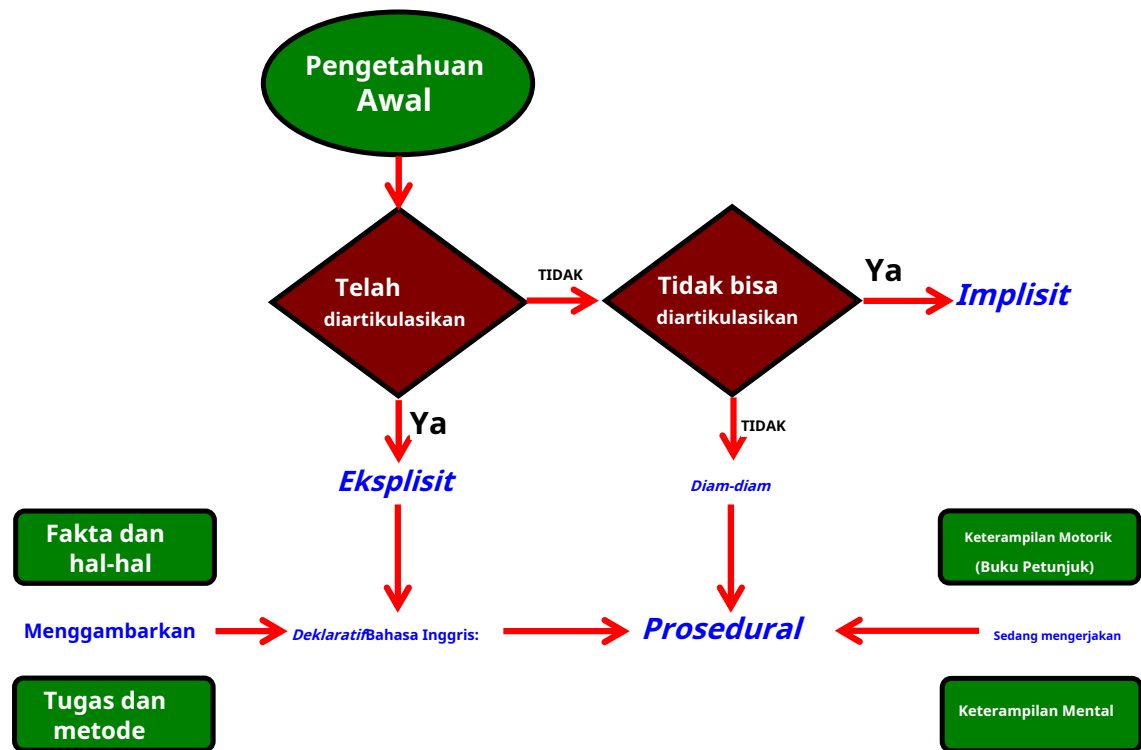
fakta, proposisi, pernyataan,

jaringan semantik, logika dan

model deskriptif.

Hubungan antar Jenis Pengetahuan

Hubungan antara *eksplisit* Bahasa Indonesia: *implisit* Bahasa Indonesia: *diam-diam* Bahasa Indonesia: *deklaratif* Dan *prosedural* pengetahuan diilustrasikan di bawah ini.



Ara. Hubungan antar jenis pengetahuan

Gambar tersebut menunjukkan :

Deklaratif pengetahuan terikat pada *"menggambarkan"* Dan

Prosedural pengetahuan terikat pada *"sedang mengerjakan."*

Panah vertikal yang menghubungkan *eksplisit* dengan *deklaratif* Dan *diam-diam* dengan *prosedural*, menunjukkan adanya hubungan yang kuat di antara mereka.

Panah horizontal yang menghubungkan *deklaratif* Dan *prosedural* menunjukkan bahwa kita sering berkembang *pengetahuan prosedural* sebagai hasil dari memulai dengan *deklaratif* pengetahuan. Yaitu, kita sering kali "mengetahui tentang" sebelum kita "mengetahui caranya".

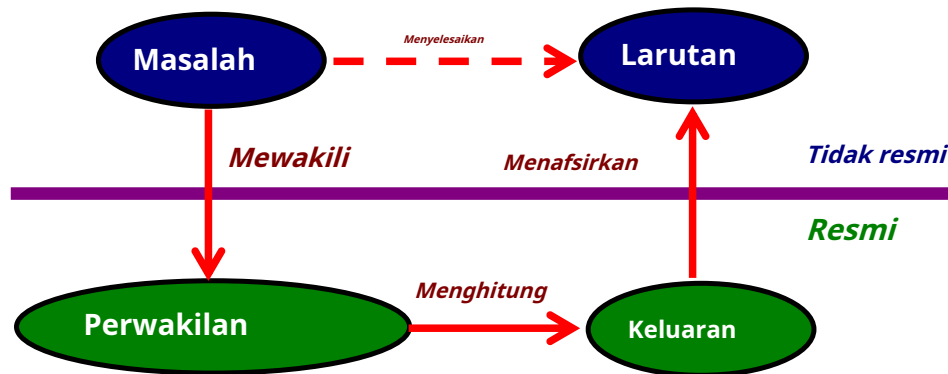
Oleh karena itu, kita dapat melihat:

- semua pengetahuan prosedural sebagai pengetahuan diam-diam, dan
- semua pengetahuan deklaratif sebagai pengetahuan eksplisit.

Kerangka Representasi Pengetahuan ((Bahasa Indonesia: Kolam Renang 1998)

Komputer memerlukan deskripsi masalah yang terdefinisi dengan baik untuk diproses dan memberikan solusi yang dapat diterima dan terdefinisi dengan baik.

Untuk mengumpulkan fragmen pengetahuan, pertama-tama kita perlu merumuskan deskripsi. dalam bahasa lisan kita dan kemudian menyajikannya dalam bahasa formal sehingga komputer dapat mengerti. Komputer kemudian dapat menggunakan algoritma untuk menghitung jawaban. Proses ini diilustrasikan di bawah ini.



Gambar. Kerangka Representasi Pengetahuan

Langkah-langkahnya adalah

- Formalisme informal dari masalah terjadi terlebih dahulu.
- Kemudian direpresentasikan secara formal dan komputer menghasilkan output.
- Output ini kemudian dapat direpresentasikan dalam solusi yang dijelaskan secara informal yang dipahami pengguna atau diperiksa konsistensinya.

Catatan : Pemecahan masalah memerlukan

- representasi pengetahuan formal, Dan
- konversi pengetahuan informal menjadi pengetahuan formal, yaitu
- konversi pengetahuan implisit menjadi pengetahuan eksplisit.

Pengetahuan dan Representasi

Pemecahan masalah memerlukan sejumlah besar pengetahuan dan beberapa mekanisme untuk memanipulasi pengetahuan itu.

Pengetahuan dan Representasi adalah entitas yang berbeda, memainkan peran peran yang sentral namun dapat dibedakan dalam sistem cerdas.

- **Pengetahuan** adalah deskripsi dunia;
itu menentukan *kompetensi sistem* berdasarkan apa yang diketahuinya.
- **Perwakilan** adalah cara pengetahuan dikodekan;
ini mendefinisikan *kinerja sistem* dalam melakukan sesuatu.

Dengan kata sederhana, kita:

- perlu tahu tentang *hal-hal yang ingin kami wakili* Bahasa Indonesia: Dan
- membutuhkan beberapa cara yang *hal-hal yang bisa kita manipulasi*.

◇ mengetahui hal-hal untuk mewakili	‡ Objek	- fakta tentang objek dalam domain.
	‡ Acara	- tindakan yang terjadi di domain.
	#Pertunjukan	- pengetahuan tentang cara melakukan sesuatu
	‡ Metafora pengetahuan	- pengetahuan tentang apa yang kita ketahui
◇ butuh berarti memanipulasi	#Memerlukan beberapa formalisme	- untuk apa yang kami wakili;

Dengan demikian, representasi pengetahuan dapat dianggap pada dua tingkat:

- (A) *tingkat pengetahu* di mana fakta-fakta dijelaskan, dan
- (B) *tingkat simbol* di mana representasi objek, yang didefinisikan dalam istilah simbol, dapat dimanipulasi dalam program.

Catatan: Representasi yang baik memungkinkan akses cepat dan akurat ke pengetahuan dan pemahaman tentang konten.

Pemetaan antara Fakta dan Representasi

Pengetahuan adalah kumpulan "*fakta*" dari beberapa domain.

Kita membutuhkan representasi dari "*fakta*" yang dapat dimanipulasi oleh suatu program.

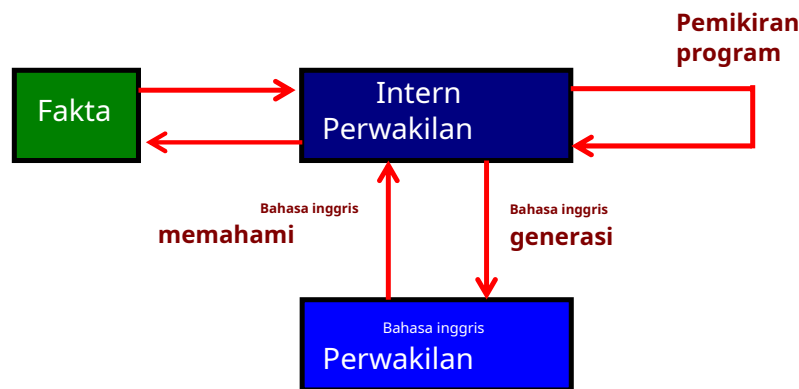
Bahasa Inggris Normal tidak cukup, terlalu sulit saat ini untuk program komputer.

menarik kesimpulan dalam bahasa alami.

Oleh karena itu, beberapa representasi simbolis diperlukan.

Oleh karena itu, kita harus bisa memetakan "*fakta menjadi simbol*" Dan "*simbol untuk fakta*" menggunakan *pemetaan representasi maju dan mundur*.

Contoh : Pertimbangkan sebuah kalimat bahasa Inggris



Fakta

◇ Spot adalah seekor anjing

◇ anjing (Spot)

◇ $\forall x : \text{anjing}(x) \rightarrow \text{ekor kuda}(x)$

Representasi

Afaktadiwakili dalam *kalimat bahasa inggris*

Menggunakan *fungsi pemetaan maju* itu di atas *fakta* direpresentasikan dalam *logika*

A *representasi logis* dari *fakta* itu

"semua anjing punya ekor"

Sekarang menggunakan

deduktif

mekanisme

kita bisa menghasilkan yang baru

representasi objek :

◇ hastail (bintik)

Representasi objek baru

◇ Spot memiliki ekor

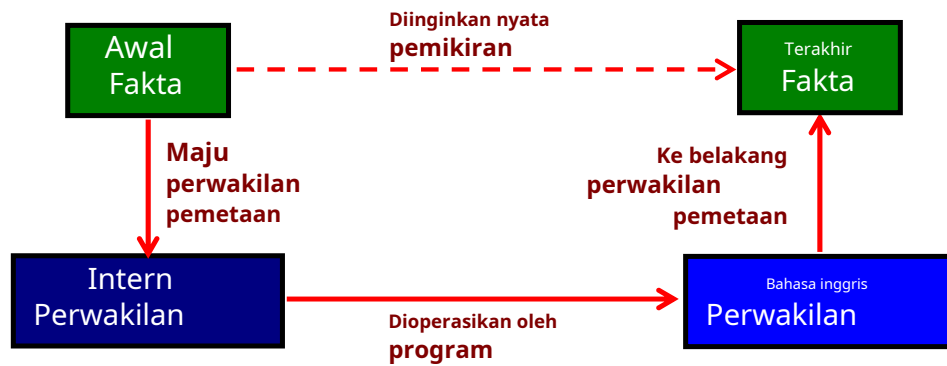
Menggunakan *fungsi pemetaan mundur* ke

[itu adalah pengetahuan baru]

menghasilkan kalimat bahasa Inggris

■ Representasi Maju dan Mundur

Representasi maju dan mundur diuraikan di bawah ini:



‡ Garis putus-putus di atas menunjukkan *penalaran abstrak* proses yang program ini dimaksudkan untuk memodelkan.

‡ Garis padat di bagian bawah menunjukkan *penalaran konkret* proses yang dilakukan program tersebut.

Persyaratan Sistem KR

Representasi pengetahuan yang baik memungkinkan akses yang cepat dan akurat ke pengetahuan dan pemahaman tentang konten.

Suatu sistem representasi pengetahuan harus memiliki sifat-sifat berikut.

- ◇ Representasional Kecukupan Itu kemampuan untuk mewakili segala macam ilmu yang dibutuhkan dalam domain tersebut.
- ◇ Kecukupan Inferensial Itu kemampuan untuk memanipulasi representasional struktur untuk mendapatkan struktur baru yang sesuai untuk pengetahuan baru yang disimpulkan dari pengetahuan lama.
- ◇ Efisiensi Inferensial Itu kemampuan untuk menggabungkan informasi tambahan ke dalam struktur pengetahuan yang dapat digunakan untuk memfokuskan perhatian pada mekanisme inferensi ke arah yang paling menjanjikan.
- ◇ Akuisisi Efisiensi Itu kemampuan untuk memperoleh pengetahuan baru menggunakan metode otomatis jika memungkinkan daripada ketergantungan pada campur tangan manusia.

Catatan: Sampai saat ini tidak ada sistem tunggal yang dapat mengoptimalkan semua properti di atas.

2 Skema Representasi Pengetahuan

Ada 5 jenis representasi Pengetahuan kami:

Relasional, Inheritable, Inferensial, dan Deklaratif/Prosedural.

◇ Pengetahuan Relasional:

- menyediakan kerangka kerja untuk membandingkan dua objek berdasarkan padanannya atribut.
- setiap contoh dimana dua objek berbeda dibandingkan adalah jenis pengetahuan relasional.

◇ Pengetahuan yang Dapat Diwariskan

- diperoleh dari objek terkait.
- ini menetapkan struktur di mana objek baru dibuat yang mungkin mewarisi semua atau sebagian atribut dari objek yang ada.

◇ Pengetahuan Inferensial

- disimpulkan dari objek melalui hubungan antar objek.
- misalnya, sebuah kata saja merupakan sintaksis yang sederhana, tetapi dengan bantuan kata lain kata-kata dalam frasa pembaca dapat menyimpulkan lebih banyak dari sebuah kata; ini Inferensi dalam linguistik disebut semantik.

◇ Pengetahuan Deklaratif

- pernyataan di mana pengetahuan ditentukan, tetapi penggunaannya bahwa pengetahuan itu harus ditaruh, tidak diberikan.
- misalnya hukum, nama orang; ini adalah fakta yang dapat berdiri sendiri, tidak tergantung pada pengetahuan lain;

Pengetahuan Prosedural

- representasi di mana informasi kontrol, untuk menggunakan pengetahuan, tertanam dalam pengetahuan itu sendiri.
- misalnya program komputer, petunjuk, dan resep; ini menunjukkan penggunaan atau implementasi khusus;

Skema KR ini dirinci dalam beberapa slide berikutnya

Pengetahuan Relasional:

Pengetahuan ini mengaitkan unsur-unsur satu domain dengan domain lain.

- Pengetahuan relasional terdiri dari objek yang terdiri dari atribut dan nilai terkait yang sesuai.
- Hasil dari jenis pengetahuan ini adalah pemetaan elemen-elemen di antara domain yang berbeda.

Tabel di bawah menunjukkan cara sederhana untuk menyimpan fakta.

- Fakta-fakta mengenai sekumpulan objek diletakkan secara sistematis dalam kolom-kolom.
- Representasi ini memberikan sedikit kesempatan untuk inferensi.

Tabel - Pengetahuan Relasional Sederhana

Pemain	Tinggi	Berat	Kelelawar - Lemparan
Harun	6-0	180	Benar - Benar
Mei	5-10	170	Benar - Benar
Rut	6-2	215	Kiri - Kiri
Williams	6-3	205	Kiri - Kanan

‡Mengingat fakta-fakta yang ada, tidak mungkin untuk menjawab pertanyaan sederhana seperti:

"Siapa pemain yang paling berat?"

tetapi jika prosedur untuk menemukan pemain terberat disediakan, maka ini

Fakta akan memungkinkan prosedur itu untuk menghitung jawaban.

‡Kita dapat menanyakan hal-hal seperti siapa yang "memukul – kiri" dan "melempar – kanan".

Pengetahuan yang Dapat Diwariskan:

Di sini elemen pengetahuan mewarisi atribut dari induknya.

Pengetahuan tersebut diwujudkan dalam hierarki desain yang ditemukan di domain fungsional, fisik dan proses. Dalam hierarki, elemen-elemen mewarisi atribut dari orang tuanya, tetapi dalam banyak kasus tidak semua atribut elemen induk harus ditentukan pada elemen anak.

Itu *warisan* adalah bentuk inferensi yang kuat, namun tidak memadai.

KR perlu dilengkapi dengan mekanisme inferensi.

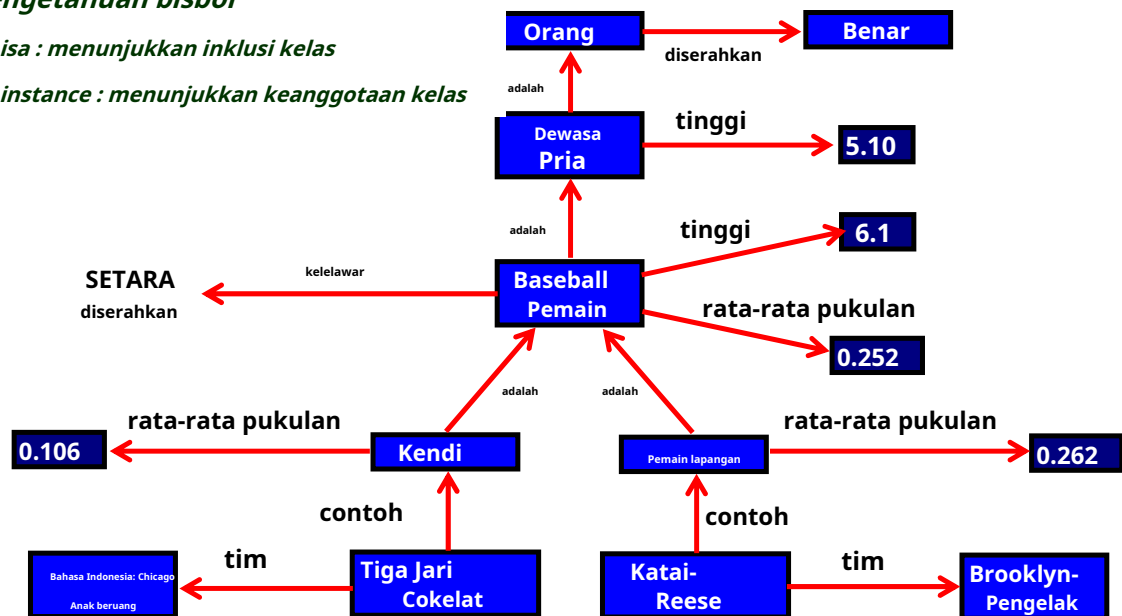
KR dalam struktur hierarki, ditunjukkan di bawah ini, disebut *"jaringan semantik"* atau koleksi *"bingkai"* atau *"struktur slot dan pengisi"*. Strukturnya menunjukkan pewarisan properti dan cara penyisipan pengetahuan tambahan.

Pewarisan properti: Objek atau elemen tertentu atribut dan nilai dari kelas yang lebih umum. diorganisasikan dalam hierarki umum.

kelas mewarisi
Kelas-kelasnya adalah

Pengetahuan bisbol

- *isa* : menunjukkan inklusi kelas
- *instance* : menunjukkan keanggotaan kelas



Gambar. Representasi pengetahuan yang dapat diwariskan (KR)

‡Panah yang diarahkan mewakili *atribut* (*isa*, *contoh*, *tim*) berasal

pada objek yang sedang dideskripsikan dan berakhir pada objek atau nilainya.

‡Node kotak mewakili *benda* dan *nilai-nilai* dari atribut.

[Dilanjutkan di slide berikutnya]

[dari slide sebelumnya – contoh]

◇ Melihat node sebagai bingkai

Contoh : Pemain bisbol

adalah : Pria dewasa
 Bates : Setara tangan
 Tinggi : 6.1
 Rata-rata pukulan : 0.252

◇ Algoritma :Warisan Properti

Ambil nilai **Bahasa Indonesia: V** untuk suatu atribut **A** dari sebuah objek contoh **HAI**.

Langkah-langkah yang harus diikuti:

1. Temukan objek **HAI** dalam basis pengetahuan.
2. Jika ada nilai untuk atribut tersebut **A** lalu laporkan nilai tersebut.
3. Jika tidak ada nilai untuk atribut instance; Jika tidak, maka gagal.
4. Atau, pindah ke node yang sesuai dengan nilai tersebut dan cari nilai untuk atribut **A**; Jika ditemukan, laporkan.
5. Jika tidak, lakukan sampai tidak ada lagi nilai untuk "**adalah**" "atribut atau sampai jawaban ditemukan:
 - (a) Dapatkan nilai dari "**adalah**" "atribut dan pindah ke node tersebut.
 - (b) Lihat apakah ada nilai untuk atribut tersebut **A**; Jika ya, laporkan.

Algoritma ini sederhana. Ini menggambarkan mekanisme dasar pewarisan. Tidak disebutkan apa yang harus dilakukan jika ada lebih dari satu nilai dari contoh atau "**adalah**" "atribut.

Hal ini dapat diterapkan pada contoh basis pengetahuan yang diilustrasikan, di slide sebelumnya, untuk mendapatkan jawaban atas pertanyaan berikut:

- tim (Pee-Wee-Reese) = Brooklyn–Dodger
- rata-rata pukulan (Three-Finger-Brown) = 0,106
- tinggi (Pee-Wee-Reese) = 6,1
- kelelawar (Three Finger Brown) = benar

[Untuk penjelasannya - lihat buku tentang AI oleh Elaine Rich & Kevin Knight, halaman 112]

Pengetahuan Inferensial:

Pengetahuan ini menghasilkan informasi baru dari informasi yang diberikan.

Informasi baru ini tidak memerlukan pengumpulan data lebih lanjut dari sumbernya, tetapi memerlukan analisis informasi yang diberikan untuk menghasilkan informasi baru pengetahuan.

Contoh :

- mengingat sekumpulan hubungan dan nilai, seseorang dapat menyimpulkan nilai lain atau hubungan.
- logika predikat (deduksi matematika) digunakan untuk menyimpulkan dari suatu himpunan atribut.
- inferensi melalui logika predikat menggunakan serangkaian operasi logis untuk menghubungkan data individual.
- Simbol yang digunakan untuk operasi logika adalah:

" \rightarrow " (implikasi), " \neg " (tidak), " \vee " (atau), " \wedge " (dan),
 " \forall " (untuk semua), " \exists " (ada).

Contoh pernyataan logika predikat:

1. "Rasa takjub" adalah nama seekor anjing : **anjing (heran)**
2. Semua anjing termasuk dalam kelas hewan : **$\forall x : \text{anjing}(x) \rightarrow \text{Hewan}(x)$**
3. Semua hewan hidup di darat atau di air: **$\forall x : \text{hewan}(x) \rightarrow \text{hidup}(x, \text{daratan}) \vee \text{hidup}(x, \text{perairan})$**

Dari ketiga pernyataan ini kita dapat menyimpulkan bahwa:

"Rasa takjub hidup di darat atau di air".

Catatan: Jika informasi lebih lanjut tersedia mengenai objek-objek ini dan isinya, hubungan, maka lebih banyak pengetahuan yang dapat disimpulkan.

Pengetahuan Deklaratif/Prosedural

Perbedaan antara pengetahuan Deklaratif/Prosedural tidak terlalu jelas.

Pengetahuan deklaratif:

Di sini, pengetahuan didasarkan pada **fakta deklaratif** tentang *aksioma* dan *domain*.

- aksioma dianggap benar kecuali ditemukan contoh yang bertentangan membatalkannya.
- Domain mewakili dunia fisik dan fungsionalitas yang dipersepsikan.
- aksioma dan domain dengan demikian hanya ada dan berfungsi sebagai deklaratif pernyataan yang dapat berdiri sendiri.

Pengetahuan prosedural:

Di sini, pengetahuan adalah **proses pemetaan** antara domain yang menentukan "**apa yang harus dilakukan ketika**" dan representasinya adalah "**cara membuatnya**" lebih tepatnya dibandingkan "*apa itu*" Pengetahuan prosedural:

- mungkin memiliki efisiensi inferensial, tetapi tidak memiliki kecukupan inferensial dan efisiensi akuisisi.
- direpresentasikan sebagai program kecil yang tahu cara melakukan hal-hal tertentu, bagaimana cara melanjutkan.

Contoh: Parser dalam bahasa alami memiliki pengetahuan bahwa kata benda

Frasa tersebut dapat mengandung artikel, kata sifat, dan kata benda. Oleh karena itu, frasa tersebut disebut rutin yang mengetahui cara memproses artikel, kata sifat, dan kata benda.

3 Isu dalam Representasi Pengetahuan

Tujuan mendasar dari Representasi Pengetahuan adalah untuk memfasilitasi menyimpulkan (simpulan) dari pengetahuan.

Masalah yang muncul saat menggunakan teknik KR banyak sekali. Beberapa di antaranya dijelaskan di bawah ini.

◇Atribut Penting:

Atribut apa pun dari objek sangat mendasar sehingga terjadi di hampir setiap domain masalah?

◇Hubungan antar atribut:

Hubungan penting apa saja yang ada di antara atribut objek?

◇Memilih Granularitas :

Pada tingkat detail apa pengetahuan harus direpresentasikan?

◇Seperangkat objek :

Bagaimana himpunan objek direpresentasikan?

◇Menemukan struktur yang tepat:

Mengingat banyaknya pengetahuan yang tersimpan, bagaimana bagian-bagian yang relevan dapat diakses?

Catatan: Masalah ini dijelaskan secara singkat, merujuk pada contoh sebelumnya, Gambar.

KR yang dapat diwariskan. Untuk detailnya, pembaca dapat merujuk pada buku tentang AI karya Elaine Rich & Kevin

Ksatria- halaman 115 – 126.

Atribut Penting: (Contoh Ref. - Gambar. KR yang Dapat Diwariskan)

Ada atribut-atribut yang memiliki arti penting secara umum.

Ada dua atribut "contoh" dan "adalah", yang bersifat umum pentingnya. Atribut-atribut ini penting karena mendukung warisan properti.

- **Hubungan antar Atribut:** Bahasa Indonesia: (Contoh Ref. - Gambar. KR yang Dapat Diwariskan)

Atribut untuk mendeskripsikan objek adalah entitas yang diwakilinya.

Hubungan antara atribut suatu objek, terlepas dari spesifiknya pengetahuan yang mereka kodekan, mungkin memiliki sifat-sifat seperti:

Invers, eksistensi dalam hierarki isa, teknik penalaran tentang nilai dan atribut bernilai tunggal.

◇ Kebalikan :

Ini tentang pemeriksaan konsistensi Bahasa Indonesia: sementara nilai ditambahkan ke satu atribut.

Entitas-entitas tersebut saling terkait satu sama lain dengan berbagai cara. Gambar menunjukkan atribut (*isa, instance, dan tim*), masing-masing dengan panah yang diarahkan, Berasal dari objek yang sedang dijelaskan dan berakhir di objek atau nilainya.

Ada dua cara untuk mewujudkan hal ini:

• pertama, mewakili dua hubungan dalam *representasi tunggal*; misalnya, a representasi logis, *tim(Pee-Wee-Reese, Brooklyn-Dodgers)* Bahasa Indonesia: yang dapat diartikan sebagai pernyataan tentang Pee-Wee-Reese atau Brooklyn-Dodger.

• kedua, gunakan atribut yang fokus pada entitas tunggal tetapi menggunakannya dalam pasangan, satu kebalikan dari yang lain; misalnya, satu, *tim = Brooklyn-Pemain Dodger* Bahasa Indonesia: dan yang lainnya, *tim = Pee-Wee-Reese, . . .*

Pendekatan kedua ini diikuti dalam jaringan semantik dan berbasis bingkai sistem, disertai dengan alat akuisisi pengetahuan yang menjamin konsistensi slot terbalik dengan memeriksa, setiap kali nilai ditambahkan ke satu atribut, maka nilai yang sesuai ditambahkan ke kebalikannya.

◇ Keberadaan dalam hierarki "isa" :

Ini tentang *generalisasi-spesialisasi* Bahasa Indonesia: seperti, kelas objek dan subset khusus dari kelas-kelas tersebut. Ada atribut dan spesialisasi atribut.

Contoh: atribut "*tinggi*" adalah spesialisasi atribut umum

"*ukuran fisik*" yang pada gilirannya merupakan spesialisasi dari "*atribut fisik*".

Hubungan generalisasi-spesialisasi untuk atribut ini adalah penting karena mendukung pewarisan.

◇ Teknik penalaran tentang nilai :

Ini tentang *penalaran nilai atribut* tidak diberikan secara eksplisit.

Beberapa jenis informasi digunakan dalam penalaran, seperti,

tinggi : harus dalam satuan panjang,

usia : : umur seseorang tidak boleh lebih dari umur orang tua orang tersebut.

Nilai-nilai tersebut sering ditentukan saat basis pengetahuan dibuat.

◇ Atribut bernilai tunggal :

Ini tentang sebuah *atribut spesifik* itu dijamin untuk mengambil yang unik nilai.

Contoh: Seorang pemain baseball pada suatu waktu hanya dapat memiliki satu *tinggi* dan menjadi anggota hanya satu tim. Sistem KR mengambil pendekatan yang berbeda untuk menyediakan dukungan bagi atribut bernilai tunggal.

Memilih Granularitas

Tingkat pengetahuan apa yang harus direpresentasikan dan apa itu primitif?

- Haruskah jumlahnya sedikit atau haruskah jumlahnya banyak?
primitif tingkat rendah atau fakta tingkat tinggi.
- Fakta tingkat tinggi mungkin tidak memadai untuk inferensi sementara fakta tingkat rendah primitif mungkin memerlukan banyak penyimpanan.

Contoh Granularitas:

- Misalkan kita tertarik untuk mengikuti fakta-fakta berikut
John melihat Sue.
- Hal ini dapat direpresentasikan sebagai
Terlihat (agen (John), objek (Sue))
- Representasi seperti itu akan memudahkan untuk menjawab pertanyaan-pertanyaan seperti:
Siapa yang melihat Sue?
- Misalkan kita ingin tahu
Apakah John melihat Sue?
- Jika hanya ada satu fakta, kita tidak dapat menemukan jawabannya.
- Kita dapat menambahkan fakta lain, seperti
Berbintik (x, y) → gergaji (x, y)
- Sekarang, kita dapat menyimpulkan jawaban pertanyaan itu.

Seperangkat Objek

Sifat-sifat tertentu dari objek yang benar sebagai anggota suatu himpunan tetapi tidak sebagai individu;

Contoh: Perhatikan pernyataan yang dibuat dalam kalimat berikut

"ada lagi *domba* dibandingkan *rakyat* di Australia", dan

"*Bahasa inggris*" Penuturnya dapat ditemukan di seluruh dunia."

Untuk menjelaskan fakta-fakta ini, satu-satunya cara adalah dengan melampirkan pernyataan pada himpunan tersebut.

mewakili *orang*, *domba*, Dan *Bahasa inggris*.

Alasan untuk merepresentasikan kumpulan objek adalah:

Jika suatu sifat berlaku untuk semua atau sebagian besar elemen suatu himpunan, maka akan lebih efisien jika kita mengasosiasikannya sekali dengan himpunan daripada mengaitkannya secara eksplisit dengan setiap elemen himpunan.

Hal ini dilakukan dengan berbagai cara :

- dalam representasi logis melalui penggunaan *pengukur universal*, Dan
- dalam struktur hirarkis dimana simpul mewakili himpunan, *warisan menyebarkan* tetapkan level pernyataan ke individu.

Contoh: menegaskan *besar (gajah)* Bahasa Indonesia:

Ingatlah untuk membuat perbedaan yang jelas antara,

- apakah kita menegaskan beberapa properti dari himpunan itu sendiri, cara, *kumpulan gajah itu besar* Bahasa Indonesia: atau
- menegaskan beberapa properti yang berlaku untuk elemen-elemen individual dari himpunan tersebut, cara, *apapun yang berupa gajah itu besar*.

Ada tiga cara di mana himpunan dapat direpresentasikan:

- (a) Nama, seperti pada contoh – Ref Fig. KR yang dapat diwariskan, simpul - Baseball- Pemain dan predikat sebagai Bola dan Pemukul dalam representasi logis.
- (b) Definisi ekstensional adalah mencantumkan angka-angka, dan
- (c) Dalam definisi tensional adalah untuk memberikan aturan, yang mengembalikan benar atau salah Tergantung pada apakah objek tersebut ada di dalam set atau tidak.

[Pembaca dapat merujuk buku tentang AI oleh Elaine Rich & Kevin Knight- halaman 122 - 123]

Menemukan Struktur yang Tepat

Akses ke struktur yang tepat untuk menggambarkan situasi tertentu.

Hal ini memerlukan pemilihan struktur awal dan kemudian merevisi pilihan tersebut.

Dalam melakukan hal ini, perlu dipecahkan masalah-masalah berikut ini:

- cara melakukan pemilihan awal struktur yang paling tepat.
- cara mengisi rincian yang sesuai dari situasi saat ini.
- bagaimana menemukan struktur yang lebih baik jika struktur yang dipilih pada awalnya ternyata tidak sesuai.
- apa yang harus dilakukan jika tidak ada satu pun struktur yang tersedia yang sesuai.
- kapan harus membuat dan mengingat struktur baru.

Tidak ada metode yang baik dan umum untuk menyelesaikan semua masalah ini.

Beberapa teknik representasi pengetahuan menyelesaikan sebagian masalah tersebut.

[Pembaca dapat merujuk buku tentang AI oleh Elaine Rich & Kevin Knight- halaman 124 - 126]

2. KR menggunakan Logika Predikat

Pada bagian sebelumnya banyak diilustrasikan tentang pengetahuan dan KR masalah terkait. Bagian ini mengilustrasikan:

Bagaimana pengetahuan dapat direpresentasikan sebagai "struktur simbol" yang menjadi ciri khas sedikit pengetahuan tentang objek, konsep, fakta, aturan, strategi;

Contoh :	"merah"	mewakili	warna merahBahasa Indonesia:
	"mobil1 " "	mewakili	mobil sayaBahasa Indonesia:
	"merah(mobil1)"	mewakili	fakta bahwa mobilku berwarna merah.

Asumsi tentang KR:

- *Perilaku Cerdas* dapat dicapai melalui manipulasi struktur simbol.
- *Bahasa KR* dirancang untuk memfasilitasi operasi atas struktur simbol, memiliki sintaksis dan semantik yang tepat;
Sintaksis memberitahu ekspresi mana yang sah?,
misalnya, merah1(mobil1), merah1 mobil1, mobil1(merah1), merah1(mobil1 & mobil2) ?;
Semantik memberitahu apa arti suatu ekspresi?
misalnya, properti "merah tua" berlaku untuk mobil saya.
- *Membuat Inferensi*, menarik kesimpulan baru dari fakta yang ada.

Dan

Untuk memenuhi asumsi tentang KR, kita memerlukan notasi formal yang memungkinkan inferensi dan pemecahan masalah otomatis. Salah satu pilihan yang populer adalah penggunaan **logika**.

Logika

Logika berkaitan dengan kebenaran pernyataan tentang dunia.

Umumnya setiap pernyataan adalah *BENAR* atau *PALSU*.

Logika meliputi: *Sintaksis Bahasa Indonesia: Semantik Dan Prosedur Inferensi*.

◇ Sintaksis :

Menentukan *simbol* dalam bahasa tentang bagaimana mereka dapat digabungkan untuk membentuk kalimat. Fakta-fakta tentang dunia direpresentasikan sebagai kalimat dalam logika.

◇ Semantik :

Menentukan cara menetapkan nilai kebenaran pada kalimat berdasarkan *arti* di dunia. Menentukan fakta apa yang dirujuk oleh suatu kalimat.

Fakta adalah sebuah klaim tentang dunia, dan bisa jadi *BENAR* atau *PALSU*.

◇ Prosedur Inferensi :

Menentukan *metode* untuk menghitung kalimat baru dari yang sudah ada kalimat.

Catatan

Fakta: adalah klaim tentang dunia yang *BENAR* atau *PALSU*.

Perwakilan: adalah sebuah ungkapan (kalimat), yang merupakan singkatan dari *benda* dan *hubungan*.

Kalimat : dapat dikodekan dalam program komputer.

Logika sebagai Bahasa KR

Logika adalah bahasa untuk penalaran, kumpulan aturan yang digunakan saat melakukan sesuatu. penalaran logis. Logika dipelajari sebagai bahasa KR dalam kecerdasan buatan.

◇ **Logika** adalah suatu sistem formal dimana rumus atau kalimat mempunyai kebenaran atau nilai salah.

◇ **Masalah perancangan bahasa KR** adalah sebuah tradeoff antara apa yang

(A) *Ekspresif* cukup untuk mewakili objek dan hubungan penting dalam suatu domain masalah.

(B) *Efisien* cukup dalam penalaran dan menjawab pertanyaan tentang informasi implisit dalam jumlah waktu yang wajar.

◇ **Logika ada beberapa jenis :** *Logika proposisional, logika predikat, Logika temporal, logika modal, logika deskriptif*;

Mereka mewakili berbagai hal dan memungkinkan penarikan kesimpulan yang lebih atau kurang efisien.

◇ **Logika proposisional dan logika predikat merupakan dasar bagi semua logika.**
Logika Proposisional adalah studi tentang pernyataan dan konektivitasnya.
Logika Predikat adalah studi tentang individu dan sifat-sifat mereka.

1 Representasi Logika

Logika dapat digunakan untuk merepresentasikan fakta-fakta sederhana.

Itu *Fakta* adalah klaim tentang dunia yang *BENAR* atau *PALSU*.

Untuk membangun representasi berbasis Logika:

- ◇ Pengguna mendefinisikan satu set primitif *simbol* dan terkait *semantik*.
- ◇ Logika mendefinisikan cara menyatukan simbol-simbol sehingga pengguna dapat mendefinisikannya legal *kalimat* dalam bahasa yang mewakili *BENAR* fakta.
- ◇ Logika mendefinisikan cara menyimpulkan *kalimat baru* dari yang sudah ada.
- ◇ Kalimat - baik *BENAR* atau *PALSU* tapi tidak keduanya disebut *proposisi*.
- ◇ Kalimat deklaratif menyatakan suatu *penyataan* dengan proposisi sebagai konten; contoh:

deklaratif "*salju itu berwarna putih*" mengungkapkan bahwa *salju itu berwarna putih* Bahasa Indonesia:

lebih jauh, "*salju itu berwarna putih*" mengungkapkan bahwa *salju itu berwarna putih* adalah *BENAR*.

Pada bagian ini, pertama **Logika Proposisional (PL)** dijelaskan secara singkat dan kemudian itu **Logika predikat** diilustrasikan secara rinci.

Logika Proposisional (PL)

Proposisi adalah suatu pernyataan, yang dalam bahasa Inggris merupakan suatu pernyataan deklaratif. kalimat. Setiap proposisi adalah BENAR atau SALAH.

Contoh: (a) Langit berwarna biru., (b) Salju itu dingin. , (c) $12 * 12 = 144$

- ‡ Proposisi adalah "kalimat", benar atau salah tetapi tidak keduanya.
- ‡ Kalimat adalah unit terkecil dalam logika proposisional.
- ‡ Jika proposisi itu benar, maka nilai kebenarannya adalah "benar".
- ‡ Jika proposisi itu salah, maka nilai kebenarannya adalah "salah".

Contoh :

Kalimat	Nilai kebenaran	Usulan (Y/T)
"Rumput itu hijau"	"BENAR"	Ya
" $2 + 5 = 5$ "	"PALSU"	Ya
"Tutup pintunya"	-	TIDAK
"Apakah diluar panas?"	-	TIDAK
" $x > 2$ " dimana x adalah variabel	-	TIDAK
" $x = x$ "	-	(karena x tidak terdefinisi) TIDAK

(tidak tahu apa itu "x" dan "="; " $3 = 3$ "
atau "udara sama dengan udara" atau
"Air sama dengan air"
(tidak memiliki arti)

- Logika proposisional merupakan dasar bagi semua logika.
- Logika proposisional juga disebut kalkulus proposisional, Kalkulus kalimat, dan Logika proposisional. kalkulus, atau aljabar Boolean.
- Logika proposisional menjelaskan cara menggabungkan dan/atau memodifikasi keseluruhan proposisi, pernyataan atau kalimat untuk membentuk sesuatu yang lebih rumit proposisi, pernyataan atau kalimat, serta logika hubungan dan properti yang diperoleh dari metode menggabungkan atau mengubah pernyataan.

■ Pernyataan, Variabel dan Simbol

Istilah-istilah ini dan beberapa istilah terkait lainnya, seperti, *penghubung*, *nilai kebenaran* **Bahasa Indonesia**: *kontingensi*, *tautologi*, *kontradiksi*, *anteseden*, *konsekuensi*, *argumen* dijelaskan di bawah ini.

◇ Pernyataan

Sederhana pernyataan (kalimat), BENAR atau SALAH, yang tidak mengandung pernyataan lain sebagai bagiannya, adalah proposisi dasar; huruf kecil, **p, q, dan r** **Bahasa Indonesia**: adalah simbol untuk pernyataan sederhana.

Besar, majemuk atau kompleks pernyataan tersebut dibangun dari dasar proposisi dengan menggabungkannya dengan kata hubung.

◇ Konektif atau Operator

Itu **kata sambung bergabung** pernyataan sederhana menjadi senyawa, dan menggabungkan senyawa menjadi senyawa yang lebih besar.

Tabel di bawah ini menunjukkan, **konjungsi dasar** dan simbol-simbolnya :

- tercantum dalam urutan prioritas operasi yang menurun;
- operasi dengan prioritas lebih tinggi diselesaikan terlebih dahulu.

Contoh rumus : (((a ∧ ¬b) ∨ c) → D) ↔ ¬ (sebuah ∨ c))

Konektif dan Simbol dalam urutan prioritas operasi yang menurun

Koneksi	Simbol				Baca sebagai
pernyataan	P				"p itu benar"
penyangkalan	¬P	~	!	BUKAN	"p salah"
konjungsi	P ∧ Q	Bahasa Indonesia: &	&	DAN	"p dan q keduanya benar"
pemisahan	P ∨ q	Bahasa Indonesia: atau		ATAU	"baik p benar, atau q benar, atau keduanya"
implikasi	P → Q	⊃	⇒	jika ..maka	"jika p benar, maka q benar" " p menyiratkan q "
persamaan derajatnya	↔	≡	↔	jika dan hanya jika	"p dan q keduanya benar atau keduanya salah"

Catatan: Proposisi dan konjungsi merupakan unsur dasar logika proposisional.

◆ Nilai Kebenaran

Nilai kebenaran suatu pernyataan adalah itu *KEBENARAN* atau *KEPALSUAN* Bahasa Indonesia:

Contoh :

P adalah *BENAR* atau *PALSI* Bahasa Indonesia:

~hal adalah *BENAR* atau *PALSI* Bahasa Indonesia:

p v q adalah *BENAR* atau *PALSI* Bahasa Indonesia: dan sebagainya.

menggunakan "**T**" atau "**1**" berarti *BENAR*.

menggunakan "**F**" atau "**angka 0**" berarti *PALSI*

Tabel kebenaran mendefinisikan dasar kata penghubung:

P	Q	~ hal	~q	$P \wedge Q$	$p v q$	$P \rightarrow Q$	$P \leftrightarrow Q$	$Q \rightarrow P$
T	T	F	F	T	T	T	T	T
T	F	F	T	F	T	F	F	T
F	T	T	F	F	T	T	F	F
F	F	T	T	F	F	T	T	T

[Slide berikutnya menunjukkan nilai kebenaran dari sekelompok proposisi, disebut tautologi, kontradiksi, kontingensi, anteseden, konsekuensi. Mereka bentuk argumen dimana satu proposisi mengklaim mengikuti proposisi lain secara logis dalil]

◇ Tautologi

Suatu proposisi yang selalu benar disebut "tautologi".

misalnya, $(P \text{ lawan } \neg P)$ selalu benar terlepas dari nilai kebenaran proposisi tersebut P .

◇ Kontradiksi

Suatu proposisi yang selalu salah disebut "kontradiksi".

misalnya, $(P \wedge \neg P)$ selalu salah terlepas dari nilai kebenaran proposisi tersebut P .

◇ Kontingensi

Sebuah proposisi disebut sebagai "kontingensi", jika proposisi tersebut juga tidak *Aulangan yg tdk berguna* juga tidak *kontradiksi*.

misalnya, $(P \text{ melawan } Q)$ adalah suatu kemungkinan.

◇ Anteseden, Konsekuen

Keduanya adalah bagian dari pernyataan kondisional.

Dalam pernyataan kondisional, $P \rightarrow Q$ Bahasa Indonesia: itu

pernyataan pertama atau "jika - klausa" (Di Sini P) disebut *mendahului* Bahasa Indonesia:

pernyataan ke 2 atau "lalu - klausa" (Di Sini Q) disebut *akibat*.

◇ Argumen

Argumen adalah suatu demonstrasi atau bukti suatu pernyataan.

Contoh : "**Burung itu adalah burung gagak; karena itu, warnanya hitam.**"

Argumen apa pun dapat diungkapkan sebagai pernyataan majemuk.

Dalam logika, argumen adalah sekumpulan satu atau lebih pernyataan deklaratif yang bermakna.

kalimat (atau "proposisi") yang dikenal sebagai *tempat* bersama dengan kalimat deklaratif bermakna lainnya (atau "proposisi") yang dikenal sebagai *kesimpulan*.

Premis adalah suatu proposisi yang memberikan alasan, dasar, atau bukti untuk suatu hal. menerima beberapa proposisi lain, yang disebut kesimpulan.

Kesimpulan adalah sebuah proposisi yang konon dibangun atas dasar dasar proposisi lainnya.

Ambil semua premis, gabungkan semuanya, dan buat konjungsi tersebut menjadi anteseden dari suatu kondisi dan buatlah kesimpulannya konsekuensi. Pernyataan implikasi ini disebut dengan konsekuensi. *kondisional dari argumen.*

Catatan: Setiap argumen memiliki kondisi yang sesuai, dan setiap pernyataan implikasi memiliki argumen yang sesuai. Karena kondisional yang sesuai dari suatu argumen adalah suatu pernyataan, oleh karena itu entah tautologi, atau kontradiksi, atau kontingensi.

‡ Sebuah argumen adalah **sah**

"**jika dan hanya jika**" kondisional yang sesuai adalah *ulangan yg tdk berguna*.

‡ Dua pernyataan adalah **konsisten**

"**jika dan hanya jika**" Konjungsi keduanya bukanlah suatu kontradiksi.

‡ Dua pernyataan adalah **setara secara logis**

"**jika dan hanya jika**" kolom tabel kebenarannya identik;

"**jika dan hanya jika**" pernyataan kesetaraan mereka menggunakan " \equiv " adalah suatu tautologi.

Catatan: Tabel kebenaran cukup memadai untuk menguji *validitas, tautologi, kontradiksi, kontingensi, konsistensi, dan kesetaraan*.

Logika Predikat

Logika proposisional tidak cukup kuat untuk semua jenis pernyataan;

Contoh : Pernyataan " $x > 1$ ", Di mana x adalah variabel, bukan proposisi karena itu tidak benar atau salah kecuali nilai x didefinisikan.

Untuk $x > 1$ menjadi sebuah proposisi,

- salah satu kami mengganti nomor tertentu untuk x Bahasa Indonesia:
- atau x adalah menjadi sesuatu seperti

"Ada bilangan x yang memenuhi $x > 1$ " Bahasa Indonesia:
- atau **"Untuk setiap angka x , berlaku $x > 1$ ".**

Perhatikan contoh :

"Semua manusia itu fana.

Socrates adalah seorang pria.

Kalau begitu Socrates itu fana",

Hal ini tidak dapat diungkapkan dalam logika proposisional sebagai sesuatu yang terbatas dan logis. argumen yang valid (rumus).

Kita membutuhkan bahasa : yang memungkinkan kita untuk mendeskripsikan properti (*predikat*) dari objek, atau hubungan antar objek yang direpresentasikan oleh variabel.

Logika predikat memenuhi persyaratan suatu bahasa.

- *Logika predikat* cukup kuat untuk berekspresi dan bernalar.
- *Logika predikat* dibangun atas ide-ide *logika proposisional*.

■ Predikat :

Setiap "kalimat" yang lengkap mengandung dua bagian: "subjek" dan "predikat".
"predikat".

Itu *subjek* adalah tentang apa (atau siapa) kalimat tersebut.

Itu *predikat* menceritakan sesuatu tentang subjek;

Contoh :

A *kalimat* "Judy {berlari}".

Subjek adalah Judy Dan predikat adalah *berjalan*.

Predikat, selalu mencakup kata kerja, menceritakan sesuatu tentang subjek.

**Predikat adalah pola frasa kata kerja yang menjelaskan properti suatu kata kerja.
objek, atau hubungan antar objek yang direpresentasikan oleh variabel.**

Contoh:

"Mobil yang dikendarai Tom *berwarna biru*" ;

"Langit *berwarna biru*" ;

"Sampul buku ini *berwarna biru*"

Predikat adalah "*berwarna biru*" Bahasa Indonesia: menjelaskan properti.

Predikat diberi nama; Biarkan '**B**' adalah nama untuk predikat "*berwarna_biru*".

Kalimat tersebut direpresentasikan sebagai "**B(x)**" Bahasa Indonesia: dibaca sebagai "*x itu biru*" Bahasa Indonesia:

Simbol "**X**" mewakili Objek yang sembarangan.

■ Ekspresi Logika Predikat :

Operator proposisional menggabungkan predikat, seperti

Jika ($p(\dots) \wedge (\neg Q(\dots) \vee R(\dots))$)

Operator logika :

Contoh Disjungsi (ATAU) dan konjungsi (DAN).

Perhatikan ekspresi dengan simbol logika masing-masing

Bahasa Indonesia: Dan \wedge

$x < y$ Bahasa Indonesia: (kamu $< z$ & Bahasa Indonesia: $z < x$)

yang

BENAR Bahasa Indonesia: (BENAR \wedge BENAR) Bahasa Indonesia:

Dengan menerapkan tabel kebenaran, ditemukan

BENAR

Tugas untuk < adalah 3, 2, 1

untuk x, y , dan z

Dan

Kemudian

Nilainya bisa PALSU atau BENAR

$3 < 2$ Bahasa Indonesia: ($2 < 1 \wedge 1 < 3$)

Dia

PALSU

■ Logika Predikat Kuantifier

Seperti dikatakan di atas, $x > 1$ bukan proposisi dan mengapa?

Juga dikatakan, bahwa untuk $x > 1$ untuk menjadi sebuah proposisi, apa yang dibutuhkan?

Secara umum, predikat dengan variabel (disebut rumus atom) yang dapat dijadikan sebuah proposisi dengan menerapkan salah satu dari dua hal berikut operasi untuk masing-masing variabelnya :

1. Tetapkan nilai pada variabel; misalnya, $x > 1$, jika 3 ditugaskan ke x menjadi $3 > 1$ Bahasa Indonesia: dan kemudian menjadi pernyataan yang benar, maka dari itu dalil.
2. Mengukur variabel dengan menggunakan quantifier pada rumus predikat logika (disebut rumus yang terbentuk dengan baik), seperti $x > 1$ atau Contoh soal $P(x)$, oleh menggunakan Quantifier pada variabel.

Terapkan Kuantifier pada Variabel

‡ Variabel x

- * $x > 5$ bukan sebuah proposisi, kebenarannya tergantung di atas nilai variabel x
- * untuk menalar pernyataan tersebut, x perlu dideklarasikan

‡ Pernyataan $x : \text{sebuah}$

- * $x : \text{sebuah}$ mendeklarasikan variabel x
- * $x : \text{sebuah}$ dibaca sebagai " x adalah elemen dari himpunan a "

‡ Pernyataan P adalah pernyataan tentang x

- * $Qx : \text{sebuah} \cdot P$ adalah kuantifikasi pernyataan

 - adalah kuantifikasi pernyataan
 - penyataan
 - deklarasi
 - variabel x sebagai elemen dari himpunan a
 - pembilang

- * Kuantifier ada dua jenis:

universal kuantifier, dilambangkan dengan simbol \forall Dan
eksistensial kuantifier, dilambangkan dengan simbol \exists

Catatan: Beberapa slide berikutnya menceritakan lebih lanjut tentang kedua Kuantifier ini.

■ Semesta Wacana

Semesta wacana, disebut juga domain wacana atau semesta.

Ini menunjukkan :

- *Akumpulan entitas* yang dibahas oleh pengukur kuantitas.
- *entitas* dapat berupa himpunan bilangan riil, himpunan bilangan bulat, himpunan semua mobil di tempat parkir, sekumpulan semua siswa di dalam kelas, dll.
- *semesta* Oleh karena itu, merupakan domain variabel (individu).
- *proposisi* dalam logika predikat ada pernyataan pada objek suatu semesta.

Alam semesta sering kali tersirat dalam praktik, namun seharusnya hal ini jelas dari konteksnya.

Contoh:

- Tentang "bilangan asli" **untuk Semua x, y ($x < y$ atau $x = y$ atau $x > y$)** Bahasa Indonesia: tidak perlu lebih tepat lagi dan mengatakan **untuk Semua x, y di \mathbb{N}** Bahasa Indonesia: Karena **N** tersirat, menjadi semesta wacana.
- Tentang sifat yang berlaku untuk bilangan asli tetapi tidak berlaku untuk bilangan riil angka, perlu untuk menentukan berapa nilai yang diperbolehkan **X**
Dan **kamu** adalah.

■ Terapkan Kuantifier Universal \forall "Untuk Semua"

Kuantifikasi Universal memungkinkan kita membuat pernyataan tentang koleksi objek.

‡ Kuantifikasi universal: $\forall x : \text{sebuah} \bullet p$

- * baca "untuk semua x di dalam A Bahasa Indonesia: P memegang "
- * A adalah alam semesta wacana
- * x adalah anggota domain wacana.
- * P adalah pernyataan tentang x

‡ Secara proposisional Bentuknya dituliskan sebagai : $\forall x$ Contoh soal $P(x)$

- * baca "untuk semua x contoh soal $P(x)$ memegang "
- "untuk setiap x contoh soal $P(x)$ memegang " atau
- "untuk setiap x contoh soal $P(x)$ memegang "
- * Di mana $P(x)$ adalah predikat,
- $\forall x$ berarti semua benda x di alam semesta
- contoh soal $P(x)$ berlaku untuk setiap objek x di alam semesta

‡ Contoh : Bahasa Inggris ke Proposisional membentuk

- * "Semua mobil punya roda"
- $\forall x : \text{mobil} \bullet x \text{ memiliki roda}$
- * Contoh soal $P(x)$
- Di mana $P(x)$ adalah predikatnya memberitahu : $x \text{ memiliki roda}$
- x adalah variabel untuk objek x 'mobil' yang mengisi jagat wacana

■ Terapkan Kuantifier Eksistensial \exists "Ada"

Kuantifikasi Eksistensial memungkinkan kita untuk menyatakan bahwa suatu objek memang ada tanpa menyebutkannya.

‡ Kuantifikasi eksistensial: $\exists x : \text{sebuah} \cdot p$

* membaca "ada sebuah x seperti yang p memegang "

* A adalah alam semesta wacana

* x adalah anggota domain wacana.

* P adalah pernyataan tentang x

‡ Secara proposisional Bentuknya dituliskan sebagai : \exists Contoh soal $P(x)$

* membaca "ada sebuah x seperti yang Contoh soal $P(x)$ " atau

"ada setidaknya satu x seperti yang Contoh soal $P(x)$ "

* Di mana Contoh soal $P(x)$ adalah predikat

$\exists x$ berarti setidaknya satu objek x di alam semesta

Contoh soal $P(x)$ benar untuk setidaknya satu objek x di alam semesta

‡ Contoh : Bahasa Inggris ke Proposisional membentuk

* "Seseorang mencintaimu"

$\exists x : \text{Seseorang} \cdot x \text{ mencintaimu}$

* Contoh soal $P(x)$

Di mana Contoh soal $P(x)$ predikatnya memberitahu : ' x mencintaimu

x adalah variabel untuk objek 'seseorang' itu mendiami jagat wacana

■ Rumus

Dalam logika matematika, rumus adalah jenis objek abstrak.

Token suatu rumus adalah sebuah simbol atau rangkaian simbol yang dapat diartikan sebagai unit yang bermakna dalam bahasa formal.

‡ Ketentuan

Didefinisikan secara rekursif sebagai variabel, atau konstanta, atau fungsi seperti

$f(t_1, \dots, t_N)$, Di mana f adalah simbol fungsi n-ary, dan t_1, \dots, t_N

adalah *ketentuan* Penerapan predikat pada istilah menghasilkan *rumus atom*.

‡ Rumus atom

Rumus atom (atau hanya atom) adalah rumus yang tidak memiliki kedalaman struktur proposisional, yaitu suatu rumus yang tidak mengandung logika kata hubung atau rumus yang tidak memiliki subrumus yang ketat.

- **Atom** dengan demikian merupakan rumus logika yang paling sederhana dan terbentuk dengan baik.
- **Rumus Senyawa** dibentuk dengan menggabungkan atom rumus yang menggunakan konjungsi logika.
- **Formula yang terbentuk dengan baik** ("wff") adalah sebuah simbol atau rangkaian simbol (a rumus) yang dihasilkan oleh tata bahasa formal dari suatu bahasa formal.

Rumus atom adalah salah satu bentuk :

- $t_1 = \text{untuk Bahasa Indonesia}$: Di mana t_1 dan t_2 adalah istilah, atau
- $R(t_1, \dots, t_N)$, Di mana R adalah simbol relasi n-ary, dan t_1, \dots, t_N adalah istilah.
- $\neg \text{sebuah}$ adalah sebuah rumus ketika A adalah suatu rumus.
- $(A \wedge B)$ dan $((\text{biasanya}))$ adalah rumus ketika A dan B adalah rumus

‡ Rumus senyawa : contoh

$$(((A \wedge B) \wedge C) \vee ((\neg \text{sebuah} \wedge B) \wedge C)) \vee ((A \wedge \neg b) \wedge C)$$

2 Mewakili Hubungan “IsA” dan “Instance”

Pernyataan logika, mengandung *subjek* Bahasa Indonesia: *predikat*, Dan *obyek*, adalah dijelaskan. Disebutkan pula dua atribut penting "*contoh*" Dan "*adalah*"; di dalam struktur hierarkis (Ref. Gambar. KR yang Dapat Diwariskan).

Atribut "**Adalah**" Dan "**Contoh**" mendukung pewarisan properti dan bermain peran penting dalam representasi pengetahuan.

Cara kedua atribut ini "**contoh**" Dan "**adalah**", diungkapkan secara logis ditunjukkan pada contoh berikut ini :

■ Contoh : Sebuah kalimat sederhana seperti "Joe adalah seorang musisi"

◇ Di Sini "**adalah sebuah**" (ditelepon **Adalah**) adalah cara untuk mengungkapkan sesuatu secara logis disebut *hubungan kelas-instance* antara subjek diwakili oleh istilah "**Joe**" Dan "**pemusik**".

◇ "**Joe**" adalah sebuah *contoh dari kelas benda* ditelepon "**pemusik**".

"Joe" memainkan peran *contoh* Bahasa Indonesia:

"**pemusik**" memainkan peran *kelas* dalam kalimat itu.

◇ Catatan: Dalam kalimat seperti itu, meskipun bagi manusia tidak ada kebingungan, tetapi untuk komputer setiap hubungan harus didefinisikan secara eksplisit.

Hal ini ditetapkan sebagai:

	[Joe]	Adalah	[Pemusik]
yaitu,	[Contoh]	Adalah	[Kelas]

3 Fungsi Komputasi dan Predikat

Tujuannya adalah untuk mendefinisikan kelas fungsi yang dapat dihitung dalam hal F.

Hal ini diungkapkan sebagai **Bahasa Indonesia: C { F }** dijelaskan di bawah ini menggunakan dua contoh:

(1) "evaluasi faktorial n" dan (2) "ekspresi untuk fungsi segitiga".

■ **Contoh 1** :Eksprei kondisional untuk mendefinisikan **faktorial** yaitu **N!**

◇Eksprei

"jika p_1 lalu e_1 kalau tidak jika p_2 lalu e_2 jika tidak p_N lalu e_N ".

yaitu. (hal 1 – Bahasa Inggris: e1 Bahasa Indonesia: hal.2 – Bahasa Inggris: e2 hal → id)

Di Sini **P_1 tahun P_2 , P_N** adalah ekspresi proposisional yang mengambil nilai-nilai **T** atau **F** untuk benar dan salah.

◇Nilai dari ($P_1 \rightarrow$ Bahasa Inggris: 1 tahun $P_2 \rightarrow$ Bahasa Inggris: 2, $P_N \rightarrow$ Bahasa Inggris: N) adalah nilai dari

Bahasa Inggris: sesuai dengan yang pertama **P** yang memiliki nilai **T**.

◇ Ekspresi yang mendefinisikan **N!**, **$n = 5$** , secara rekursif adalah:

tidak! = bilangan bulat $n (n-1)!$ untuk $n \geq 1$

5! = 1x2x3x4x5 = 120

0! = 1

Definisi di atas menggabungkan sebuah contoh itu :

jika hasil perkalian tidak ada angka yaitu **0! = 1** Bahasa Indonesia:

maka hanya, hubungan rekursif (**Tentukan $n+1$ dan $xn+2!$ bekerja untuk $n=0$ = tidak ada**

◇ Penggunaan ekspresi kondisional di atas untuk mendefinisikan fungsi **N!**

secara rekursif adalah **$n! = (n = 0) \rightarrow 1$ tahun $n - n \neq 0 = 0 \rightarrow$ Tentukan $n-1!$**

◇ **Contoh:** Mengevaluasi **2!** sesuai dengan definisi di atas.

2! = (2 = 0 → 1 tahun 2 ≠ 0 → 2. (2 - 1)!)

= 2 x 1!

= 2x (1 = 0) → 1 tahun 1 ≠ 0 → 1. (1 - 1)!)

= 2 lembar 1 lembar 0!

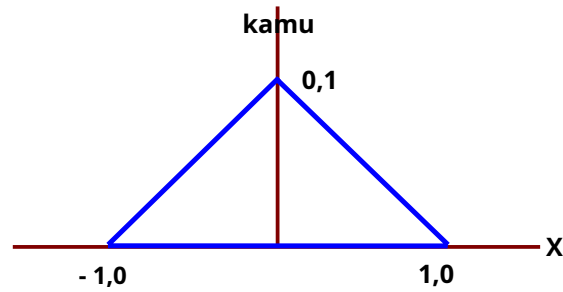
= 2 lembar 1 lembar (0 = 0 → 1, 0 ≠ 0 → 0. (0 - 1)!)

= 2 lembar 1 lembar 1

= 2

■ Contoh 2: Ekspresi kondisional untuk fungsi segitiga

◇ Grafik fungsi segitiga yang terkenal ditunjukkan di bawah ini



Ara. Fungsi Segitiga

ekspresi kondisional untuk fungsi segitiga adalah

$$-x = (x < 0) \rightarrow -x, x - \text{panjang} \geq \text{angka } 0 \rightarrow X$$

◇ Fungsi segitiga pada grafik di atas direpresentasikan oleh ekspresi kondisional

$$\text{tiga}(x) = (x) \leq -1 \rightarrow 0, 10 \text{ detik} \leq \text{angka } 0 \rightarrow -x, x \leq 1 \rightarrow x_1, x_2 > 1 \rightarrow \text{Bahasa Indonesia: } 0)$$

Resolusi adalah suatu prosedur yang digunakan untuk membuktikan argumen-argumen yang yang dapat diungkapkan dalam logika predikat adalah benar.

Resolusi adalah suatu prosedur yang menghasilkan bukti dengan sanggahan atau kontradiksi.

Resolusi mengarah pada penolakan teknik pembuktian teorema untuk kalimat dalam logika proposisional dan logika tingkat pertama.

- Resolusi adalah aturan inferensi.
- Resolusi adalah pembuktian teorema terkomputerisasi.
- Resolusi sejauh ini hanya didefinisikan untuk Logika Proposisional. Strateginya adalah bahwa teknik Resolusi Logika Proposisional diadopsi dalam Logika Predikat.

Pada slide sebelumnya, representasi Pengetahuan menggunakan logika predikat telah diilustrasikan. Pendekatan populer lainnya terhadap Pengetahuan representasi disebut *Aturan produksi* Bahasa Indonesia: *jaringan semantik* Dan *bingkai*.

Aturan produksi, kadang-kadang disebut **JIKA-MAKA** aturan yang paling populer KR.

Aturan produksi adalah bentuk KR yang sederhana tetapi kuat.

Aturan produksi memberikan fleksibilitas dalam menggabungkan deklaratif dan representasi prosedural untuk menggunakannya dalam bentuk terpadu.

Contoh aturan produksi :

- JIKA kondisi MAKA tindakan
- Premis JIKA MAKA kesimpulan
- JIKA proposisi p_1 dan proposisi p_2 benar MAKA proposisi p_3 adalah benar

Keuntungan aturan produksi :

- mereka modular,
- Setiap aturan mendefinisikan bagian pengetahuan yang kecil dan independen.
- aturan baru dapat ditambahkan dan aturan lama dihapus
- Aturan biasanya bersifat independen dari aturan lainnya.

Aturan produksi sebagai mekanisme representasi pengetahuan digunakan dalam desain banyak "**Sistem berbasis aturan**" juga disebut "**Sistem produksi**".

Jenis Aturan

Tiga jenis aturan paling banyak digunakan dalam sistem produksi berbasis Aturan.

■ Aturan Deklaratif Pengetahuan:

Aturan-aturan ini menyatakan semua fakta dan hubungan tentang suatu masalah.

Contoh :

JIKA tingkat inflasi menurun
KEMUDIAN harga emas turun.

Aturan-aturan ini adalah bagian dari basis pengetahuan.

■ Aturan Prosedur Inferensi

Aturan-aturan ini memberikan saran tentang cara menyelesaikan suatu masalah, sementara fakta-fakta tertentu diketahui.

Contoh :

JIKA data yang dibutuhkan tidak ada di sistem
KEMUDIAN memintanya kepada pengguna.

Aturan-aturan ini adalah bagian dari mesin inferensi.

■ Aturan meta

Ini adalah aturan untuk membuat aturan. Meta-aturan beralasan tentang aturan mana harus dipertimbangkan untuk dipecah.

Contoh :

JIKA aturan yang tidak menyebutkan tujuan saat ini dalam premisnya,
DAN ada aturan yang menyebutkan tujuan saat ini dalam premisnya,
MAKA aturan pertama harus digunakan, bukan aturan kedua.

- Meta-aturan mengarahkan penalaran daripada benar-benar melakukan pemikiran.
- Meta-aturan menentukan aturan mana yang harus dipertimbangkan dan di mana perintah yang harus dijalankan.

Pengetahuan Prosedural versus Pengetahuan Deklaratif

Kedua jenis pengetahuan ini didefinisikan dalam slide sebelumnya.

■ Pengetahuan Prosedural: **mengetahui 'bagaimana melakukan'**

Termasuk: *aturan, strategi, agenda, prosedur, model*.

Ini menjelaskan *apa yang harus dilakukan* untuk mencapai kesimpulan tertentu.

Contoh

Aturan: Untuk menentukan apakah Peter atau Robert lebih tua, pertama-tama cari usia mereka.

Ini adalah pengetahuan tentang '*bagaimana caranya*' sesuatu. Itu terwujud dalam melakukan sesuatu, misalnya keterampilan manual atau mental tidak dapat direduksi menjadi kata-kata. Hal ini dipegang oleh individu dengan cara yang tidak memungkinkannya untuk dikomunikasikan secara langsung kepada individu lain.

Menerima deskripsi langkah-langkah tugas atau prosedur. Tampak mirip dengan pengetahuan deklaratif, kecuali tugas atau metodenya yang dideskripsikan, bukan fakta atau benda.

■ Pengetahuan Deklaratif: **mengetahui 'apa', mengetahui 'itu'**

Termasuk: *konsep, objek, fakta, proposisi, pernyataan, model*.

Ini adalah pengetahuan tentang *Fakta Dan hubungan, itu*

- dapat diungkapkan dalam pernyataan yang sederhana dan jelas,
- dapat ditambahkan dan dimodifikasi tanpa kesulitan.

Contoh: Sebuah mobil memiliki empat ban; Peter lebih tua dari Robert.

Pengetahuan deklaratif dan pengetahuan eksplisit adalah pengetahuan yang diartikulasikan dan dapat diperlakukan sebagai sinonim untuk sebagian besar tujuan praktis.

Pengetahuan deklaratif direpresentasikan dalam format yang dapat dimanipulasi, diurai, dan dianalisis terlepas dari isinya.

■ Perbandingan :

Perbandingan antara Pengetahuan Prosedural dan Deklaratif:

Pengetahuan Prosedural

- Sulit untuk di-debug
- Kotak hitam
- Samar
- Berorientasi pada proses
- Ekstensi dapat mempengaruhi stabilitas
- Eksekusi cepat dan langsung
- Tipe data sederhana dapat digunakan
- Representasi dalam bentuk serangkaian aturan yang diorganisasikan ke dalam rutin dan subrutin.

Pengetahuan Deklaratif

- Mudah untuk divalidasi
- Kotak putih
- Eksplisit
- Berorientasi pada data
- Ekstensi mudah
- Lambat (memerlukan interpretasi)
- Mungkin memerlukan tipe data tingkat tinggi
- Representasi dalam bentuk sistem produksi, seluruh rangkaian aturan untuk melaksanakan tugas.

■ Perbandingan :

Perbandingan antara Bahasa Prosedural dan Bahasa Deklaratif :

Bahasa Prosedural	Bahasa Deklaratif
<ul style="list-style-type: none"> • Basic, C++, Cobol, dll. • Sebagian besar pekerjaan dilakukan oleh penerjemah Engine adalah bahasa • Untuk satu tugas banyak baris kode • Programmer harus memiliki keterampilan dalam menerjemahkan tujuan ke dalam baris dari kode prosedural • Membutuhkan manajemen minimum mengelola data aktual • Programmer mengerti dan memiliki setiap langkah kode • Data yang terekspos ke programmer eksekusi kode • Lebih rentan terhadap kegagalan karena struktur data • Secara tradisional lebih cepat, tapi itu berubah • Kode prosedur terkait erat dengan front end • Kode terintegrasi erat dengan struktur penyimpanan data • Programmer bekerja dengan pointer • Pengetahuan tentang trik pengkodean berlaku hanya untuk satu bahasa 	<ul style="list-style-type: none"> • Bahasa Indonesia: SQL • Sebagian besar pekerjaan yang dilakukan oleh Data dalam DBMS • Untuk satu tugas satu pernyataan SQL • Programmer harus memiliki keterampilan menyatakan tujuan dengan jelas sebagai pernyataan SQL • Mengandalkan DBMS yang mendukung SQL untuk menyimpan data dan menjalankan pernyataan SQL. • Programmer tidak memiliki akses interaksi ke dengan eksekusi pernyataan SQL • Programmer menerima data di akhir selama sebagai satu set lengkap • Lebih tahan terhadap perubahan pada struktur data • Awalnya lebih lambat, tetapi sekarang pengaturan catatan kecepatan • Pernyataan SQL yang sama akan berfungsi di dengan paling depan berakhir Kode terkait erat dengan bagian depan. • Kode secara longgar terkait dengan struktur data; DBMS menangani masalah struktural • Programmer tidak peduli dengan atau kursor posisi • Pengetahuan tentang trik SQL berlaku untuk bahasa apa pun yang menggunakan SQL

2 Pemrograman Logika

Pemrograman logika menawarkan formalisme untuk menentukan perhitungan dalam istilah hubungan logis antara entitas.

- **program logika** adalah kumpulan pernyataan logika
- **programmer** menggambarkan semua hubungan logis yang relevan antara berbagai entitas.
- **komputasi** menentukan apakah suatu kesimpulan tertentu mengikuti atau tidak dari pernyataan logis tersebut.

• Karakteristik Program Logika

Program logika dicirikan oleh serangkaian relasi dan inferensi.

- **program** terdiri dari serangkaian aksioma dan pernyataan tujuan.
- **Aturan Inferensi** menentukan apakah aksioma cukup untuk memastikan kebenaran pernyataan tujuan.
- **eksekusi** dari program logika sesuai dengan konstruksi bukti dari pernyataan tujuan dari aksioma.
- **programmer** menentukan hubungan logis dasar, tidak menentukan cara penerapan aturan inferensi.

Dengan demikian **Logika + Kontrol = Algoritma**

• Contoh Pernyataan Logika

- **Pernyataan**
Kakek-nenek adalah orang tua dari orang tua.
- Pernyataan yang diungkapkan dalam istilah logika yang lebih erat kaitannya sebagai
Seseorang adalah kakek-nenek jika dia memiliki anak dan anak itu adalah orang tua.
- Pernyataan yang diungkapkan dalam logika orde pertama sebagai
(untuk semua) x : kakek-nenek (x, y) :- induk (x, z), induk (z, y)
dibaca sebagai x adalah kakek dari y
jika x adalah orang tua dari z dan z adalah orang tua dari y

Bahasa Pemrograman Logika

Bahasa pemrograman meliputi:

- itu *sintaksis*
- itu *semantik* dari program dan
- itu *model komputasional*.

Ada banyak cara untuk mengorganisasikan perhitungan. Cara yang paling umum paradigma adalah **prosedural**. Program ini menentukan perhitungan dengan mengatakan "*Bagaimana*" itu harus dilakukan. *FORTRAN*, *C*, dan *Bahasa berorientasi objek* termasuk dalam pendekatan umum ini.

Paradigma lain adalah **deklaratif**. Program ini menentukan perhitungan dengan memberikan sifat-sifat jawaban yang benar. *Prolog* dan *bahasa data logika kolesterol LDL* adalah contoh bahasa deklaratif, menekankan logika properti suatu komputasi.

Prolog dan LDL disebut bahasa pemrograman logika.

PROLOG (PROgramming LOGic) merupakan salah satu pemrograman Logika yang paling populer. Bahasa Inggris meningkat dalam ranah Kecerdasan Buatan (AI). Itu menjadi populer di kalangan peneliti AI, yang tahu lebih banyak tentang "apa" dan "bagaimana" perilaku cerdas tercapai.

Sintaksis dan Terminologi(relevan dengan program Prolog)

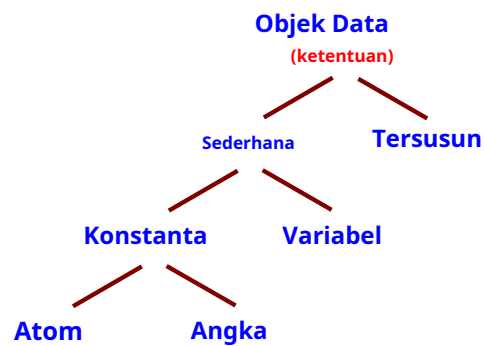
Dalam bahasa apa pun, pembentukan komponen (ekspresi, pernyataan, dll.), dipandu oleh aturan sintaksis.

Komponennya dibagi menjadi dua bagian:

(A) komponen data dan (B) komponen program.

(A) Komponen data:

Komponen data adalah kumpulan objek data yang mengikuti hierarki.



Objek data segala jenis juga disebut

Aketentuan. Suatu istilah adalah suatu konstanta, suatu variabel

atau istilah majemuk.

Objek data sederhana

adalah

bukan

dapat terurai; misalnya *atom*, *angka*, *konstanta*, *variabel*.

Sintaksis membedakan objek data,

Oleh karena itu, tidak perlu mendeklarasikannya.

Objek data terstruktur terbuat dari beberapa komponen.

Semua komponen data ini dijelaskan di slide berikutnya.

(A)Objek Data:

Objek data apapun disebut *ketentuan*.

◇ **Istilah:** Contoh

‡ **Konstanta:**

Menunjukkan elemen seperti *bilangan bulat, floating point, atom*.

‡ **Variabel:**

Menunjukkan suatu elemen tunggal namun tidak ditentukan; *simbol* untuk variabel dimulai dengan huruf besar atau garis bawah.

‡ **Istilah majemuk:**

Meliputi sebuah *fungsi* dan urutan satu atau lebih senyawa istilah yang disebut *argumen*.

► **Fungsi:** ditandai dengan nama dan jumlahnya

argumen; nama adalah sebuah *atom*, dan jumlah argumennya adalah

keberagaman.

$$f/n = f(t_1, t_2, \dots, t_N)$$

Di mana *f* adalah nama dari *fungsi* dan *n* adalah *keberagaman*

t₁ t₂ ... t_N adalah *argumen*

f/n menunjukkan *fungsi f* dari *keberagaman n*

Funktor dengan nama sama tetapi aritas berbeda adalah berbeda.

‡ **Ground dan non-ground:**

Ketentuan adalah tanah jika tidak mengandung variabel (hanya konstanta tanda); jika tidak, maka itu bukan landasan.

Sasaran adalah atom atau istilah majemuk, dan umumnya tidak tanah.

(B)Objek Data Sederhana:Atom, Angka, Variabel

◇ Atom

#huruf kecil, mungkin diikuti oleh huruf lain dari salah satu huruf besar, angka, dan karakter garis bawah.

misalnya **A** **lebih besar dari** **dua_B_atau_tidak_2_b**

‡ serangkaian karakter khusus seperti: + - * / \ = < > : ~ # \$ &

misalnya **<>** **# # &&** **::=**

#*Arangkaian*karakter apa pun yang diapit dalam tanda kutip tunggal.

misalnya **Bahasa Inggris: 'ABC'** **'1234'** **'sebuah**

#berikut ini juga *atom* **!** **Bahasa Indonesia**

◇ Angka

‡ Aplikasi yang melibatkan perhitungan numerik berat jarang ditulis dalam Prolog.

‡ *bilangan bulat* perwakilan: misalnya **0** **- 16** **33** **+ 100**

‡ *angka riil* ditulis dalam notasi standar atau ilmiah,

misalnya **0.5** **- 3.1416** **6.23e+23** **11.0e-3** **- 2.6e-2**

◇ Variabel

‡dimulai dengan huruf kapital, mungkin diikuti oleh huruf-huruf lainnya baik huruf besar, angka, dan karakter garis bawah.

misalnya **X25** **Daftar** **Kata benda_Frasa**

(C)Objek Data Terstruktur:Struktur Umum ,

◇ Struktur Umum

#istilah terstruktur dibentuk secara sintaksis oleh sebuah funktor dan daftar argumen.

#functor adalah atom.

#daftar argumen muncul di antara tanda kurung.

#argumen dipisahkan dengan koma.

#setiap argumen adalah istilah (yaitu, objek data Prolog apa pun).

#jumlah argumen dari suatu istilah terstruktur disebut *aritas*.

#misalnya **lebih besar dari(9, 6)** **f(a, b, c), h(d))** **ditambah(2, 3, 5)**

Catatan : Struktur dalam Prolog adalah mekanisme untuk menggabungkan istilah bersama-sama, seperti bilangan bulat 2, 3, 5 digabungkan dengan funktor *plus*.

◇ Struktur Khusus

#Dalam Prolog, kumpulan istilah yang terurut disebut *daftar*.

#Daftar adalah istilah terstruktur dan Prolog menawarkan cara yang nyaman notasi untuk mewakilinya:

- * Daftar kosong dilambangkan dengan atom **[]**.
- * Daftar yang tidak kosong membawa elemen di antara tanda kurung siku, memisahkan elemen dengan koma.

misalnya **[bach, lebah]** **[apel, jeruk, anggur]**

(B)Komponen Program

Program Prolog adalah kumpulan predikat atau aturan.

Predikat membentuk hubungan antara objek.

(A)Klausa, Predikat, Kalimat, Subjek

#**Ayat** adalah kumpulan kata-kata yang berhubungan secara tata bahasa.

#**Predikat** terdiri dari satu klausa atau lebih.

#Klausa adalah blok bangunan **kalimat** Bahasa Indonesia:

Setiap kalimat mengandung satu atau lebih klausa.

#Kalimat Lengkap memiliki dua bagian: **subjek** Dan **predikat**.

Ha subjek adalah apa (atau siapa) yang dibahas dalam kalimat tersebut.

Ha predikat menceritakan sesuatu tentang subjek.

#Contoh 1 : **"sapi makan rumput"**.

Ini adalah sebuah klausul, karena mengandung

itu **subjek** **"sapi"** Dan

itu **predikat** **"makan rumput."**

#Contoh 2: **"sapi sedang makan rumput terlihat dari jalan raya"**

Ini adalah klausul yang lengkap.

itu **subjek** **"sapi makan rumput"** Dan

itu **predikat** **"terlihat dari jalan raya"** membuat lengkap
pikiran.

(b) Predikat dan Klausa

Secara sintaksis, predikat tersusun dari satu klausa atau lebih.

‡Bentuk umum klausa adalah

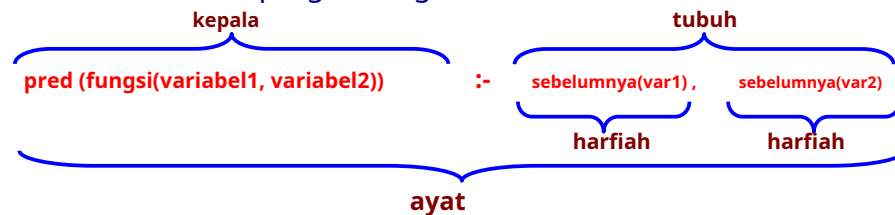
<sisi-kiri> :- <sisi-kanan>.

di mana LHS adalah satu tujuan yang disebut "*sasaran*" dan RHS terdiri dari satu atau lebih tujuan, dipisahkan dengan koma, ditelepon "*sub-tujuan*" dari gawang di sisi kiri.

Simbol "**:-**" diucapkan sebagai "itulah masalahnya"

atau "sedemikian rupa sehingga"

‡Struktur klausa dalam program logika



Secara **harfiah** mewakili pilihan yang mungkin dalam tipe primitif bahasa tertentu. Beberapa pilihan jenis literal adalah sering kali berupa bilangan bulat, floating point, Boolean, dan string karakter.

‡Contoh :

```

kakek_nenek (X, Z)      :-   induk(X, Y), induk(Y, Z).
orang tua (X, Y)      :-   ibu(X, Y).
orang tua (X, Y)      :-   ayah(X, Y).
  
```

Baca seolah-olah x adalah ibu dari y maka x adalah orang tua dari y

[Dilanjutkan di slide berikutnya]

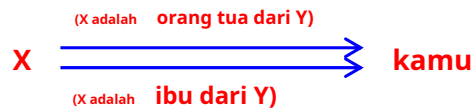
[Lanjutan dari slide sebelumnya]

#Interpretasi:

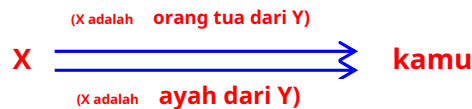
- * Suatu klausul menentukan kebenaran bersyarat dari tujuan di sebelah kiri; tujuan di LHS diasumsikan benar jika sub-tujuan di RHS semuanya benar. Suatu predikat adalah benar jika setidaknya salah satu klausanya benar.
- * Sebuah **individu "X"** adalah **kakek nenek dari "Z"** jika sebuah **induk** dari itu sama **"X"** adalah **"kamu"** Dan **"kamu"** adalah **induk** dari itu **"Z"**.



- * Sebuah **individu "X"** adalah **orang tua dari "Y"** jika **"Y"** adalah **ibu dari "X"**



- * Sebuah **individu "X"** adalah **orang tua dari "Y"** jika **"Y"** adalah **ayah dari "X"**.



(c) Klausa Unit -kasus khusus

Berbeda dengan contoh kebenaran bersyarat sebelumnya, kita sering menjumpai hubungan tanpa syarat yang berlaku.

‡ Dalam Prolog, klausa yang benar tanpa syarat disebut

klausa unit atau *fakta*.

‡ Contoh : Hubungan tanpa syarat mengatakan

'X' adalah ayah dari 'Y' adalah benar tanpa syarat.

Hubungan ini sebagai suatu *Klausa Prolog* adalah

ayah(X, Y) :- BENAR.

Diartikan sebagai hubungan ayah antara **X** dan **kamu** selalu

benar; atau hanya dinyatakan sebagai **X** adalah ayah dari **kamu**.

‡ Tujuan benar sudah tertanam dalam Prolog dan selalu berlaku.

‡ Prolog menawarkan sintaksis yang lebih sederhana untuk mengekspresikan klausa unit atau fakta

ayah(X, Y)

yaitu Bahasa Inggris **":-BENAR"** bagian tersebut dihilangkan begitu saja.

(D)Pertanyaan

Dalam Prolog, kueri adalah pernyataan yang disebut direktif.
Kasus khusus dari direktif disebut kueri.

‡Secara sintaksis, direktif adalah klausa dengan sisi kiri yang kosong.

Contoh :? -**kakek nenek(Q, Z)**

Kueri ini Q diartikan sebagai :**Siapa kakek nenek dari Z?**

Dengan mengeluarkan pertanyaan Q Prolog mencoba untuk menetapkan validitas spesifik hubungan.

Jawaban dari slide sebelumnya adalah(**X adalah kakek nenek dari Z**)

‡Hasil dari eksekusi query adalah sukses atau gagal

Kesuksesan, berarti tujuan yang ditentukan dalam kueri berlaku sesuai dengan fakta dan aturan program.

Kegagalan, berarti tujuan yang ditentukan dalam kueri tidak berlaku sesuai dengan fakta dan aturan program.

Paradigma Pemrograman :Model Perhitungan

Deskripsi lengkap tentang bahasa pemrograman meliputi:

model komputasi, sintaksis, semantik, dan pertimbangan pragmatis
yang membentuk bahasa.

Model Perhitungan:

Model komputasi adalah kumpulan nilai dan operasi, sementara komputasi adalah penerapan serangkaian operasi pada suatu nilai untuk menghasilkan nilai lain.

Ada tiga model komputasi dasar:

(a) Imperatif, (b) Fungsional, dan (c) Logika.

Selain itu, ada dua paradigma pemrograman:

(a) konkuren (b) pemrograman berorientasi objek.

Meskipun keduanya bukan model komputasi, namun mereka memiliki peringkat pentingnya dengan model komputasional.

(a) Model Imperatif

Model komputasi Imperatif, terdiri dari sebuah negara dan sebuah operasi penugasan yang digunakan untuk mengubah keadaan.

Program terdiri dari serangkaian perintah.

Perhitungan adalah perubahan dalam keadaan.

Contoh : Fungsi Linear

Fungsi linier $kamu = 2x + 3$ dapat ditulis sebagai

$$Y := 2 * X + 3$$

Implementasinya memerlukan penentuan nilai X di negara bagian dan kemudian menciptakan keadaan baru yang berbeda dari keadaan lama.

$$\text{Negara Baru: } X = 3, \quad Ynya = 9,$$

Model imperatif paling dekat dengan model perangkat keras yang digunakan program dieksekusi, yang menjadikannya model paling efisien dalam hal waktu eksekusi.

(b) Model fungsional

Model komputasi fungsional terdiri dari sekumpulan nilai, fungsi, dan pengoperasian fungsi. Fungsi dapat diberi nama dan disusun dengan fungsi lainnya. Ia dapat mengambil fungsi lainnya sebagai argumen dan mengembalikan hasil.

Program terdiri dari definisi fungsi.

Perhitungan adalah penerapan fungsi pada nilai.

‡ Contoh 1 : Fungsi Linear

Fungsi linier $y = 2x + 3$ dapat didefinisikan sebagai :

$$\text{Persamaan (x)} = 2 \cdot x + 3$$

‡ Contoh 2: Tentukan nilai Keliling.

Tetapkan nilai pada Radius, yang menentukan nilai pada Keliling.

$$\text{Keliling} = 2 \times \pi \times \text{jari-jari}, \quad \text{dimana } \pi = 3,14$$

Generalisasikan Keliling dengan variabel "radius"

yaitu

$$\text{Keliling (jari-jari)} = 2 \times \pi \times \text{jari-jari}, \text{ dimana } \pi = 3,14$$

Model fungsional dikembangkan selama bertahun-tahun. Notasi dan metode membentuk dasar yang mendasari metodologi penyelesaian masalah.

(c) Model Logika

Model logika komputasi didasarkan pada hubungan dan logika kesimpulan.

Program terdiri dari definisi relasi.

Perhitungan adalah inferensi (adalah suatu bukti).

‡Contoh 1:Fungsi linier

Fungsi linier $kamu = 2x + 3$ dapat direpresentasikan sebagai:

$f(x,y)$ adalah bilangan bulat positif yang menyatakan jika Y adalah $2*X + 3$.

Di sini fungsi mewakili hubungan antara X dan $kamu$.

‡Contoh 2: Tentukan nilai Keliling.

Perhitungan keliling dapat direpresentasikan sebagai:

Lingkaran (R, C) jika $Pi = 3,14$ dan $C = 2*Pi*R$.

Di sini fungsi tersebut direpresentasikan sebagai hubungan antara jari-jari R dan keliling C .

‡Contoh 3: Tentukan mortalitas Socrates dan Penelope.

Program ini dimaksudkan untuk menentukan kematian Socrates dan Penelope.

Faktanya Socrates dan Penelope adalah manusia.

Aturannya adalah bahwa semua manusia itu fana, itu benar.

untuk semua X , jika X adalah manusia, maka X adalah makhluk fana.

Untuk menentukan kematian Socrates atau Penelope, buatlah asumsi bahwa tidak ada manusia, yaitu $\neg fana(Y)$

[model logika dilanjutkan di slide berikutnya]

[model logika dilanjutkan pada slide sebelumnya]

‡ Bentuk ekuivalen dari fakta dan aturan yang telah dijelaskan sebelumnya adalah

manusia (Socrates)

fana (X) jika manusia (X)

‡ Untuk menentukan mortalitas Socrates dan Penelope, kami membuat asumsi bahwa tidak ada manusia yaitu **\neg fana (Y)**

‡ Perhitungan (bukti) bahwa Socrates adalah makhluk fana

1. (a)	manusia(Socrates)	Fakta
2.	fana(X) jika manusia(X)	Aturan
3	\negmanusia(Y)	anggapan
4.(a)	X = Y	
4.(b)	\negmanusia(Y)	dari 2 & 3 dengan penyatuan dan modus tollens
5.	Y = Sokrates	dari 1 dan 4 dengan penyatuan
6.	Kontradiksi	5, 4b, dan 1

‡ Penjelasan :

- * Baris pertama adalah pernyataan "Socrates adalah seorang manusia."
- * Baris ke 2 adalah sebuah frasa "semua manusia itu fana" ke dalam yang setara "untuk semua X, jika X adalah manusia, maka X adalah makhluk fana".
- * Baris ke-3 ditambahkan pada set untuk menentukan kematian Socrates.
- * Baris ke 4 merupakan deduksi dari baris 2 dan 3. Hal ini dibenarkan oleh aturan inferensi modus tollens yang menyatakan bahwa jika kesimpulan dari suatu aturannya diketahui salah, maka hipotesisnya pun demikian.
- * Variabel X dan Y disatukan karena memiliki nilai yang sama.
- * Dengan penyatuan, Baris 5, 4b, dan 1 menghasilkan kontradiksi dan mengidentifikasi Socrates sebagai manusia biasa.
- * Perlu dicatat bahwa resolusi adalah aturan inferensi yang mencari kontradiksi dan hal ini difasilitasi oleh penyatuan yang menentukan apakah ada substitusi yang membuat dua istilah menjadi sama.

Model logika memformalkan proses penalaran. Hal ini terkait dengan model relasional. basis data dan sistem pakar.

3 Penalaran Maju versus Penalaran Mundur

Arsitektur sistem berbasis aturan terdiri dari *seperangkat aturan* Bahasa Indonesia: *sekumpulan fakta*, Dan sebuah *mesin inferensi* Kebutuhannya adalah menemukan fakta-fakta baru apa yang dapat disimpulkan.

Mengingat adanya seperangkat aturan, pada dasarnya ada dua cara untuk menghasilkan pengetahuan: satu, **rantai maju** dan yang lainnya, **rantai mundur**.

■ **Rantai maju:** juga disebut data driven.

Dimulai dengan fakta, dan melihat aturan apa yang berlaku.

■ **Rantai mundur:** juga disebut berorientasi pada tujuan.

Dimulai dengan sesuatu untuk diketahui, dan mencari aturan yang akan membantu dalam menjawabnya.

■ Contoh 1

Aturan	R1 :	JIKA panas DAN berasap	KEMUDIAN	api
Aturan	R2 :	JIKA bunyi alarm	KEMUDIAN	berasap
Aturan	R3 :	JIKA api	KEMUDIAN	nyalakan alat penyiram
Fakta	Bahasa Indonesia: F1	bunyi alarm	[Diberikan]	
Fakta	F2 :	panas	[Diberikan]	

■ Contoh 2

Aturan	R1 :	JIKA panas DAN berasap	LALU TAMBAHKAN	api
Aturan	R2 :	JIKA bunyi alarm	LALU TAMBAHKAN	berasap
Aturan	R3 :	JIKA api	LALU TAMBAHKAN	switch_on_sprinklers
Fakta	Bahasa Indonesia: F1	bunyi alarm	[Diberikan]	
Fakta	F2 :	panas	[Diberikan]	

■ Contoh 3: Rantai Maju yang umum

Aturan R1:	JIKA panas DAN berasap LALU TAMBAHKAN api	
Aturan R2 :	JIKA bunyi alarm	LALU TAMBAHKAN berasap
Aturan R3 :	Jika api	LALU TAMBAHKAN switch_on_sprinklers
Fakta Bahasa Indonesia: F1	bunyi alarm	[Diberikan]
Fakta F2 :	panas	[Diberikan]
Fakta F4 :	berasap	[dari F1 oleh R2]
Fakta F2 :	api	[dari F2, F4 oleh R1]
Fakta F6 :	nyalakan alat penyiram	[dari F2 oleh R3]

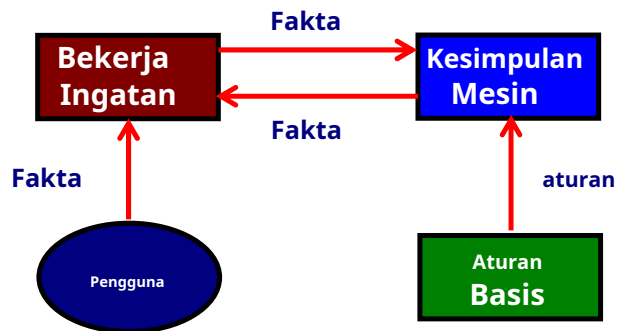
■ Contoh 4: Rantai Balik yang Khas

Aturan R1 :	JIKA panas DAN berasap	KEMUDIAN tembak
Aturan R2 :	JIKA alarm berbunyi bip	KEMUDIAN berasap
Aturan R3 :	Jika _api	KEMUDIAN nyalakan alat penyiram
Fakta Bahasa Indonesia: F1	panas	[Diberikan]
Fakta F2 :	bunyi alarm	[Diberikan]
Sasaran :	Haruskah saya menyalakan alat penyiram?	

Rantai Maju

Sistem rantai maju, properti, algoritma, Dan konflik strategi resolusi diilustrasikan.

■ Sistem rantai maju



- ‡ fakta disimpan dalam memori kerja
- ‡ aturan kondisi-tindakan mewakili tindakan yang akan diambil ketika ditentukan fakta terjadi dalam memori kerja.
- ‡ Biasanya, tindakan melibatkan penambahan atau penghapusan fakta dari pekerjaan ingatan.

■ Properti Forward Chaining

- ‡ semua aturan yang dapat menembakan memang menembak.
- ‡ dapat menjadi tidak efisien - menyebabkan aturan palsu yang berlaku, masalah yang tidak terfokus memecahkan
- ‡ serangkaian aturan yang dapat dijalankan yang dikenal sebagai rangkaian konflik.
- ‡ Keputusan tentang aturan mana yang akan diterapkan adalah penyelesaian konflik.

■ Algoritma rantai maju

- SAYA

Mengulang

- ‡ Kumpulkan aturan yang kondisinya cocok dengan fakta di WM.
- ‡ Lakukan tindakan yang ditunjukkan oleh aturan.
- (tambahkan fakta ke WM atau hapus fakta dari WM)

Sampai masalah terpecahkan atau tidak ada kondisi yang cocok

Terapkan pada Contoh 2 diperpanjang(menambahkan 2 aturan lagi dan 1 fakta)

Aturan	R1 :	JIKA panas DAN berasap	LALU TAMBAHKAN	api
Aturan	R2 :	JIKA bunyi alarm	LALU TAMBAHKAN	berasap
Aturan	R3 :	Jika api	LALU TAMBAHKAN	switch_on_sprinklers
Aturan	R4 :	JIKA kering	LALU TAMBAHKAN	switch_on_humidifier
Aturan	R5 :	JIKA alat penyiram_aktif	LALU HAPUS	kering
Fakta	Bahasa Indonesia: F1	bunyi alarm	[Diberikan]	
Fakta	F2 :	panas	[Diberikan]	
Fakta	F2 :	Kering	[Diberikan]	

Sekarang, **dua aturan bisa menyala (R2 dan R4)**

Aturan	R4	Pelembab ADD aktif	[dari F2]
Aturan R2		TAMBAHKAN berasap	[dari F1]
[diikuti oleh urutan [tindakan]		TAMBAHKAN api	[dari F2 oleh R1]
		TAMBAHKAN switch_on_sprinklers	[oleh R3]
		HAPUS kering, pelembabnya mati	[oleh R5]
		yaitu konflik!	

■ Algoritma rantai maju

- II(diterapkan pada contoh 2 di atas)

Mengulang

- ‡ Kumpulkan aturan yang kondisinya cocok dengan fakta di WM.
- ‡ Jika lebih dari satu aturan cocok seperti yang dinyatakan di atas maka
 - ◇ Gunakan *strategi resolusi konflik* untuk menghilangkan semua kecuali satu
- ‡ Lakukan tindakan yang ditunjukkan oleh aturan
- (tambahkan fakta ke WM atau hapus fakta dari WM)

Sampai masalah terpecahkan atau tidak ada kondisi yang cocok

■ Strategi Resolusi Konflik

Konflik diatur adalah sekumpulan aturan yang syarat-syaratnya telah terpenuhi oleh elemen memori kerja.

Resolusi konflik biasanya memilih satu aturan untuk dijalankan.

Mekanisme resolusi konflik yang populer adalah:

Refraktori, Keterkinian, Spesifisitas.

◇ Tahan panas

- ‡ aturan tidak boleh mengizinkan penembakan lebih dari satu kali pada waktu yang sama data.
- ‡ membuang aturan yang dieksekusi dari set konflik.
- ‡ mencegah terjadinya perulangan yang tidak diinginkan.

◇ Kebaruan

- ‡ peringkat instansiasi dalam hal keterkinian elemen dalam premis aturan.
- ‡ Aturan yang menggunakan data terkini lebih disukai.
- # Elemen memori kerja diberi tanda waktu yang menunjukkan pada jam berapa siklus setiap fakta ditambahkan ke memori kerja.

◇ Kekhususan

- ‡ Aturan yang memiliki lebih banyak kondisi dan
Oleh karena itu lebih sulit untuk memuaskan, lebih disukai daripada yang lebih umum aturan dengan lebih sedikit ketentuan.
- ‡ Aturan yang lebih spesifik lebih baik karena mengambil lebih banyak waktu data ke dalam akun.

■ Alternatif untuk Resolusi Konflik –Gunakan Meta Knowledge

Daripada strategi resolusi konflik, terkadang kita ingin menggunakan pengetahuan dalam memutuskan aturan mana yang akan diterapkan. **Aturan meta** alasan tentang aturan mana yang harus dipertimbangkan untuk pemecatan. Mereka mengarahkan penalaran daripada daripada benar-benar melakukan penalaran.

‡ Meta-pengetahuan : pengetahuan tentang pengetahuan untuk memandu pencarian.

‡ Contoh meta-pengetahuan

JIKA set konflik berisi aturan apa pun (c, a) seperti yang
a = "hewan adalah mamalia"
KEMUDIAN api (c, a)

‡ Contoh ini mengatakan meta-pengetahuan mengkodekan pengetahuan tentang bagaimana untuk memandu pencarian solusi.

‡ Meta-pengetahuan, dikodekan secara eksplisit dalam bentuk aturan dengan "objek tingkat" pengetahuan.

Rantai Mundur

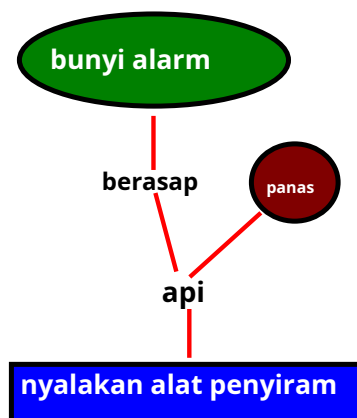
Sistem rantai mundur dan algoritmanya diilustrasikan.

■ Sistem rantai mundur

- ‡ Rantai mundur berarti penalaran dari tujuan kembali ke fakta.
Idenya adalah untuk fokus pada pencarian.
- ‡ Aturan dan fakta diproses menggunakan interpreter backward chaining.
- ‡ Memeriksa hipotesis, misalnya ""Haruskah saya menyalakan alat penyiram?"

■ Algoritma rantai mundur

- ‡ Buktikan tujuan **G**
- Jika **G** ada pada fakta awal, itu terbukti.
- Jika tidak, temukan aturan yang dapat digunakan untuk menyimpulkan **G**, Dan
cobalah untuk membuktikan setiap kondisi aturan tersebut.



Pengkodean aturan

Aturan	R1 :	JIKA panas DAN berasap	KEMUDIAN tembak
Aturan	R2 :	JIKA alarm berbunyi bip	KEMUDIAN berasap
Aturan	R3 :	Jika kebakaran	KEMUDIAN nyalakan alat penyiram
Fakta	F1 :	Bahasa Indonesia: F panas	[Diberikan]
Fakta	F2 :	bunyi alarm	[Diberikan]
Sasaran :		Haruskah saya menyalakan alat penyiram?	

Rantai Maju vs Rantai Mundur

#Tergantung pada masalah dan pada properti rangkaian aturan.

#Rantai mundur kemungkinan lebih baik jika ada hipotesis yang jelas.

Contoh : Masalah diagnostik atau masalah klasifikasi, Masalah medis sistem pakar

#Rantai maju mungkin lebih baik jika hipotesisnya kurang jelas dan ingin melihat apa yang dapat disimpulkan dari situasi saat ini;

Contoh: Sistem sintesis - desain / konfigurasi.

3.4 Pengetahuan Kontrol

Algoritma terdiri dari: komponen logika, yang menentukan pengetahuan untuk digunakan dalam memecahkan masalah, dan *komponen kontrol*, yang menentukan strategi pemecahan masalah yang dengannya pengetahuan tersebut digunakan.

Dengan demikian **Algoritma = Logika + Kontrol**.

Komponen logika menentukan makna algoritma sedangkan komponen kontrol hanya mempengaruhi efisiensinya.

Suatu algoritma dapat diformulasikan dengan cara yang berbeda, menghasilkan hasil yang sama. perilaku. Satu formulasi, mungkin memiliki pernyataan yang jelas dalam komponen logika tetapi menggunakan strategi pemecahan masalah yang canggih dalam pengendalian komponen. Rumusan lainnya, mungkin memiliki logika yang rumit komponen tetapi menggunakan strategi pemecahan masalah yang sederhana.

Efisiensi suatu algoritma seringkali dapat ditingkatkan dengan meningkatkan komponen kontrol tanpa mengubah logika algoritma dan oleh karena itu tanpa mengubah arti dari algoritma tersebut.

Tren dalam database mengarah pada pemisahan logika dan kontrol.

Bahasa pemrograman saat ini tidak membedakan keduanya.

programmer menentukan logika dan kontrol dalam satu bahasa.

mekanisme eksekusi hanya menjalankan pemecahan masalah yang paling mendasar kemampuan.

Program komputer akan lebih sering benar, lebih mudah diperbaiki, dan lebih mudah beradaptasi dengan masalah baru saat bahasa pemrograman logika dan kontrol terpisah, dan ketika mekanisme eksekusi menyediakan lebih banyak fasilitas pemecahan masalah yang kuat seperti yang disediakan oleh kecerdasan sistem pembuktian teorema.

4. Referensi : Buku Teks

1. *"Kecerdasan Buatan", oleh Elaine Rich dan Kevin Knight, (2006), McGraw Hill companies Inc., Bab 4-6, halaman 105-192.*
 2. *""Kecerdasan Buatan: Pendekatan Modern"" oleh Stuart Russell dan Peter Norvig, (2002), Prentice Hall, Bab 7-10, halaman 194-369.*
 3. *"Kecerdasan Komputasional: Pendekatan Logika", oleh David Poole, Alan Mackworth, dan Randy Goebel, (1998), Oxford University Press, Bab 5-7, halaman 169-278.*
 4. *"Kecerdasan Buatan: Struktur dan Strategi untuk Pemecahan Masalah Kompleks", oleh George F. Luger, (2002), Addison-Wesley, Bab 2, 7, halaman 35-77, 227-273.*
 5. *"AI: Sebuah Sintesis Baru", oleh Nils J. Nilsson, (1998), Morgan Kaufmann Inc., Bab 13, Halaman 217-229.*
 6. *"Kecerdasan Buatan: Teori dan Praktik", oleh Thomas Dean, (1994), Addison-Wesley, Bab 3, 6, Halaman 71-130, 255-296.*
- Nomor telepon 7. *Dokumen terkait dari sumber terbuka, terutama internet. Daftar lengkap sedang dipersiapkan untuk disertakan di kemudian hari.*