

Selamat Datang

Semoga Tuhan memberi berkah pada kelas ini.

Perencanaan pembelajaran

- TUJUAN

Agar mahasiswa memahami Sistem Pakar

Agar mahasiswa dapat memahami aplikasi dan penerapan dari sistem pakar

MATERI POKOK

Pertemuan ke-	Pokok Bahasan
1	Pengenalan Artificial Intelligent
2	Pengenalan Sistem Pakar (Expert System)
3	Model Sistem Pakar
4	Representasi Pengetahuan (Bagian 1)
5	Representasi Pengetahuan (Bagian 2)
6	Representasi Pengetahuan (Bagian 3) – Logika dan Pengetahuan
7	Review Materi / Kuis (Soal-Soal Essay)
8	UJIAN TENGAH SEMESTER
9	Metode Inferensi: Graph, Trees, Lattice
10	Metode Inferensi: Logika Deduktif dan Silogisme
11	Metode Inferensi: Argumen & logika Proporsional
12	Metode Inferensi: Rangkaian Forward & Backward
13	Ketidakpastian & Paradigma Soft Computing
14	Pengenalan CLIPS
15	Review Materi / Kuis (Soal-Soal Essay)
16	UJIAN AKHIR SEMESTER

Sumber Referensi :

- H.S, Suryadi. Seri Diktat Kuliah: Pengantar Sistem Pakar. Penerbit Gunadarma. Jakarta. 1994.
- Arhami, Muhammad. Konsep Dasar Sistem Pakar. Andi. Yogyakarta. 2005.
- Kusriani. Sistem Pakar, Teori dan Aplikasi. Penerbit Andi. Yogyakarta. 2006.
- Fausett, Laurene V. (1993). Fundamentals of Neural Network: Architectures, Algorithms and Applications. New Jersey: Prentice Hall.

CATATAN :

1. Mahasiswa wajib mempunyai buku referensi tersebut
2. Mahasiswa wajib memiliki slide dan LTM (Lembar Tugas Mahasiswa)
3. Mahasiswa bisa mencari jurnal yang berhubungan dengan Sistem pakar

RENCANA PEMBELAJARAN

- Pertemuan 1 s.d 14 disampaikan dengan Metode Ceramah, Metode Diskusi dan Latihan Soal.
- Sistem Penilaian
 - 10 % Kehadiran
 - 30 % Nilai UTS
 - 40 % Nilai UAS
 - 20 % Tugas (LTM, Kuis)

Deskripsi Singkat Tugas

Mahasiswa mengerjakan soal-soal Lembar Tugas Mahasiswa

PENGENALAN

ARTIFICIAL INTELLIGENCE



Pertemuan ke Satu

Definisi *Artificial Intelligence* (AI) :

Pendekatan Teknik (*An Engineering Approach*)

Merupakan proses di mana peralatan mekanik dapat melaksanakan kegiatan-kegiatan yang berdasarkan pada pemikiran atau kecerdasan buatan.

Pendekatan Ilmiah (*A Scientific Approach*)

Merupakan ilmu yang mempelajari bagaimana sebuah komputer dapat melakukan kegiatan yang dapat dilakukan oleh manusia.

Sejarah AI

- 1955, Pengembangan dalam bidang permainan atau game Catur oleh Shannon
- 1956, Istilah AI mulai dipopulerkan oleh John McCarthy sebagai suatu tema ilmiah di bidang komputer yang diadakan di Dartmouth College.
- 1956, Komputer berbasis AI pertama kali dikembangkan dengan nama Logic Theorist yang melakukan penalaran terbatas untuk teorema kalkulus. Perkembangan ini mendorong para peneliti untuk mengembangkan program lain yang disebut sebagai General Problem Solver (GPS).
- 1963, Checkers-playing program diperkenalkan oleh Samuel, program ini dapat digunakan untuk memainkan game dan punya kemampuan untuk menyimpan pengalaman untuk digunakan pada permainan berikutnya.

- 1972, Newell dan Simon memperkenalkan Teori Logika secara konseptual yang kemudian berkembang pesat dan menjadi acuan pengembangan sistem berbasis AI lainnya.
- 1978, Buchanan dan Feigenbaum mengembangkan bahasa pemrograman DENDRAL, bahasa pemrograman ini dibuat untuk Badan Antariksa AS (NASA) dan digunakan untuk penelitian kimia di planet Mars.

- Pada perkembangan selanjutnya studi pada AI difokuskan pada pemecahan masalah sehari-hari atau memberi pertimbangan yang masuk akal (***commonsense reasoning***) terhadap permasalahan yang dihadapi manusia.
- Hal ini mencakup pertimbangan mengenai suatu objek dan hubungannya dengan objek yang lain (mis : suatu objek mungkin berada dalam objek yang lain) dan pertimbangan tindakan apa yang dapat dilakukan terhadap objek dan konsekuensinya (mis : bila kita melepaskan objek dari genggaman objek tsb akan jatuh dan mungkin pecah).

Beberapa bidang Tugas AI

Bidang Umum

- Perception : Vision, Speech
- Natural Language : Understanding, Generation, Translation
- Commonsense Reasoning
- Robot control

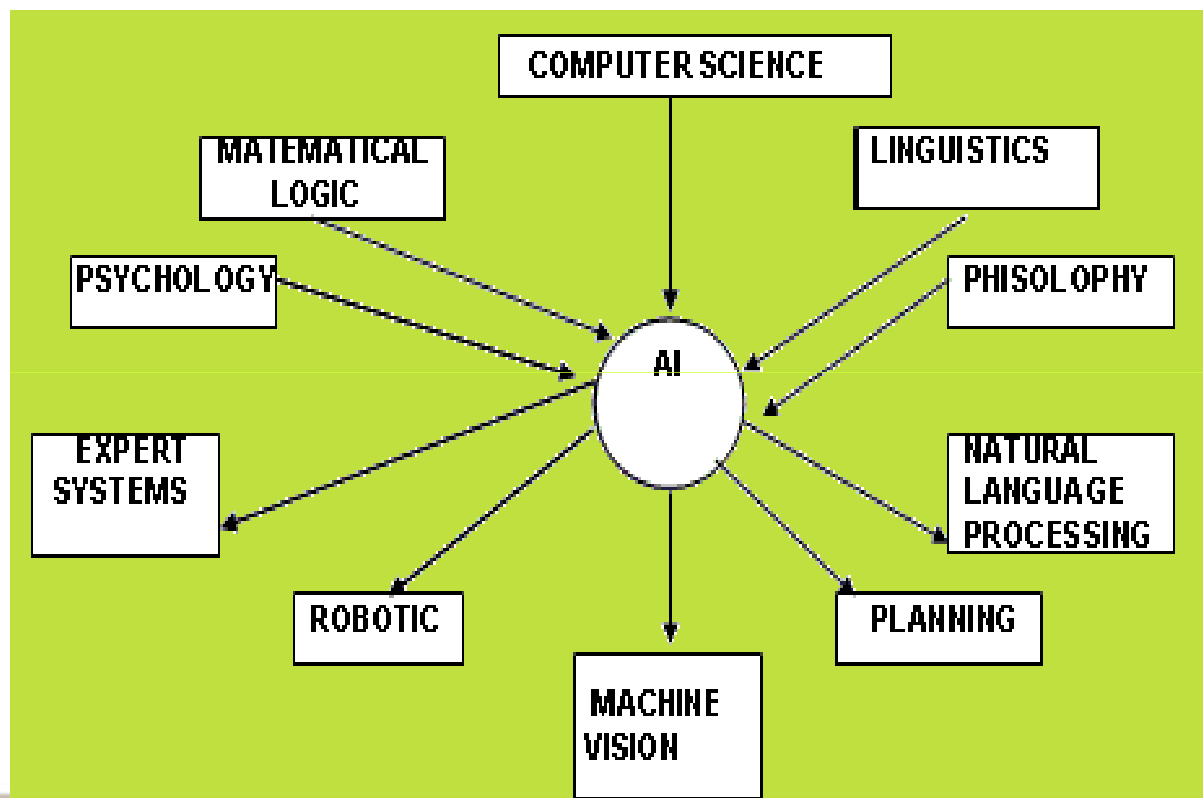
Bidang Formal

- Games : Chess, Bakgammon, Checkers, Go
- Mathematics : Geometry, Logic, Integral Calculus

Bidang Ahli

- Engineering : Design, Fault finding, Manufacturing planning
- Scientific analysis
- Medical diagnosis
- Financial analysis

Bagian-Bagian AI



Teknik-Teknik AI

Beberapa teknik AI yang penting diantaranya

1. **Search** (Pencarian)

menyediakan cara penyelesaian masalah untuk kasus dimana bila tidak ada lagi pendekatan langsung yang dapat digunakan maka pindahkan kerangka kerja kpd teknik langsung yang mungkin untuk dilekatkan.

2. **Use of Knowledge** (Penggunaan Pengetahuan)

menyediakan cara penyelesaian masalah yang lebih kompleks dengan mengeksploitasi struktur dari objek yang terkait dengan masalah tsb.

3. **Abstraction**

menyediakan cara untuk memilah/memisahkan keterangan dan variasi yang penting dari sekian banyak yang tidak penting dimana akan mempercepat penyelesaian masalah.

Pertemuan 2

Pengenalan Expert System (Sistem Pakar)

Definisi Sistem Pakar

Sistem Pakar adalah suatu sistem yang menggabungkan pengetahuan dan penelusuran data untuk memecahkan masalah yang secara normal memerlukan keahlian seorang pakar.

Yang diperlukan untuk membangun sistem pakar adalah sejumlah pengetahuan dan suatu mekanisme untuk mengakses pengetahuan itu secara efisien (mekanisme inferensi) untuk memecahkan masalah.

Kelebihan Sistem Pakar

1. **Increased Availability**

Pengetahuan seorang pakar yang sudah diadaptasi ke bentuk software dapat diperbanyak dan disebarluaskan dalam jumlah yang tidak terbatas

2. **Reduced cost**

biaya memperbanyak software lebih murah dibanding menghadirkan atau melatih seorang pakar

3. **Reduced danger**

dapat digunakan pada keadaan dan atau lokasi yang membahayakan manusia

4. **Permanence**

software sistem pakar dapat digunakan kapan saja tanpa ada batas waktu

5. Multiple expertise

sistem pakar dapat dibuat dan digunakan secara simultan dengan penggabungan beberapa pengetahuan dari banyak pakar

6. Increased reliability

Sifat komputer yang pasti dan selalu benar selama masukan dan algoritma yang diberikan benar shg keluaran sistem pakar dapat menjadi penunjang keputusan yang dapat diterima

7. Explanation

Sistem pakar dibuat dengan mekanisme inferensi yang menghasilkan suatu keluaran disertai penjelasan yang masuk akal, shg sistem pakar dapat juga digunakan untuk menjelaskan suatu teori atau keadaan tertentu

8. Fast Response

Kecepatan proses hardware komputer dapat terus ditingkatkan tanpa batas, shg kecepatan penyelesaian masalah dengan sistem pakar ikut meningkat

9. Steady, unemotional, complete

komputer bekerja dalam keadaan yang relatif tetap setiap saat, keadaan yang sulit sekali dicapai oleh manusia

10. Intelligent tutor

sistem pakar dapat dijadikan sumber belajar yang baik bagi siswa dengan menjalankan kasus-kasus

11. Intelligent database

komputer tidak memiliki sifat lupa, sistem pakar akan memberikan informasi yang sesuai selama tersimpan dalam database nya

Kelemahan Pengembangan Sistem Pakar

Beberapa kelemahan Sistem pakar diantaranya :

- Daya kerja dan produktivitas manusia menjadi berkurang karena semuanya dilakukan secara otomatis oleh sistem
- Pengembangan perangkat lunak sistem pakar lebih sulit dibandingkan perangkat lunak konvensional.

Perbandingan Perangkat Lunak Konvensional dengan Perangkat Lunak Sistem Pakar

Perangkat Lunak Konvensional	Perangkat Lunak Sistem Pakar
Fokus Pada Solusi	Fokus Pada Permasalahan
Pengembangan dapat dilakukan secara individu	Pengembangan dilakukan oleh tim kerja
Pengembangan secara sekuensial	Pengembangan secara iteratif

Karakteristik Sistem Pakar

1. **High Performance**

Sistem pakar harus dapat bekerja pada level kompetensi yang sama atau lebih baik dari seorang pakar

2. **Adequate response time**

Sistem pakar harus dapat bekerja dan menghasilkan solusi atau kesimpulan lebih cepat dari pakar manusia

3. **Good reliability**

Sistem pakar harus dapat dipercaya dimana solusi yang dihasilkan tidak cenderung atau mengacu kepada satu hal atau keadaan tertentu saja

4. **Understandable**

Sistem pakar harus dapat menjelaskan setiap solusi yang dihasilkan sebagaimana juga seorang pakar dapat menjelaskan setiap kesimpulan yang dibuat, bukan black box system

5. **Flexibility**

Karena pengetahuan terus berkembang dan berubah, maka sebuah sistem pakar juga harus memiliki mekanisme untuk menambah, merubah dan memperbaharui pengetahuan yang ada di dalamnya.

6. **Hipotesis**

Sistem pakar dapat melakukan serangkaian hipotesa yang dapat dibandingkan dan tidak bertentangan dengan hipotesa dari seorang pakar dalam masalah yang nyata

Pembagian Sistem Pakar berdasar kelas

1. **Konfigurasi**

Merakit komponen sistem dengan cara yang benar

2. **Diagnosa**

Menarik kesimpulan terhadap masalah yang dihadapi berdasarkan bukti-bukti yang diobservasi

3. **Instruksi**

Metode pengajaran yang cerdas sehingga siswa sbg user dapat melakukan tanya jawab dengan sistemmmseolah berhadapan dengan pengajar atau seorang pakar

4. **Interpretasi**

Menjelaskan data-data yang diobservasi

5. **Monitoring**

Membandingkan data yang diamati dengan data acuan yang diharapkan untuk memberi penilaian terhadap prestasi, kinerja atau masalah tertentu

6. **Perencanaan**

Merancang tindakan atau menentukan langkah yang harus diambil untuk mendapatkan hasil yang dikehendaki

7. **Prognosis**

Membuat perkiraan hasil dari data yang ada atau keadaan yang sudah atau sedang terjadi

8. **Perbaikan**

Menjelaskan tindakan yang berlaku atau yang valid untuk suatu masalah atau keadaan tertentu

9. **Kontrol**

mengatur proses yang berasal dari penggabungan kelas sistem pakar yang lain

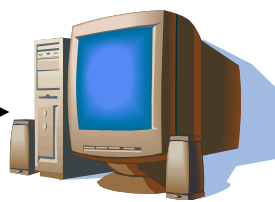
Pertemuan 3 Model Expert System

Unsur Penting Pengembangan Sistem Pakar

Ada tiga unsur penting dalam pengembangan
Sistem Pakar, yaitu :



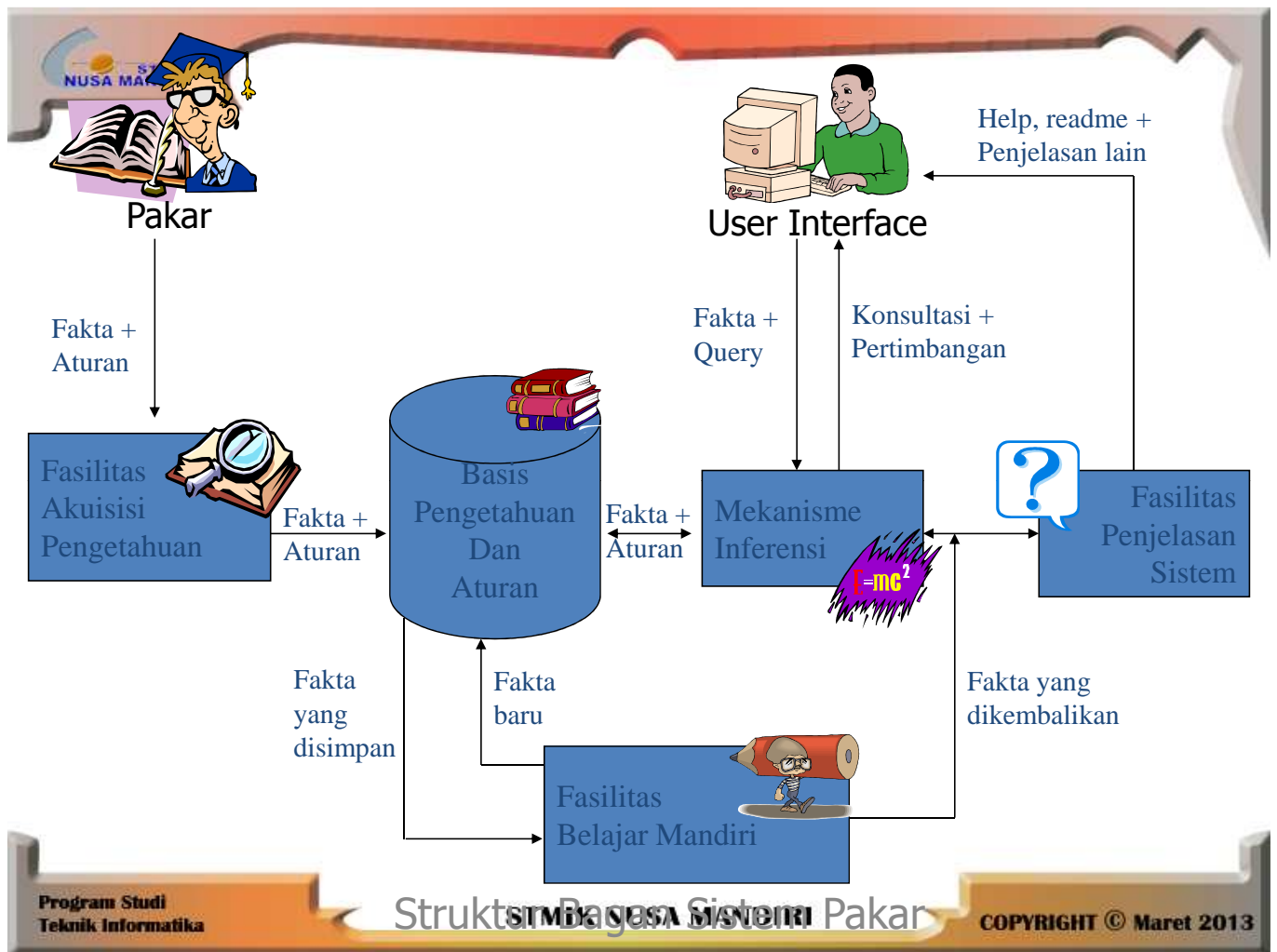
Pakar



Sistem



Pemakai



Komponen Sistem Pakar

Komponen Sistem Pakar terdiri dari :

1. Fasilitas Akuisisi pengetahuan
2. Basis Pengetahuan dan Basis Aturan
3. Mekanisme Inferensi
4. Fasilitas Belajar Mandiri
5. Fasilitas Penjelasan Sistem
6. Antarmuka Pemakai



1. Fasilitas Akuisisi Pengetahuan

Merupakan suatu proses untuk mengumpulkan data-data pengetahuan tentang suatu masalah dari pakar.

Bahan pengetahuan dapat diperoleh dengan berbagai cara, seperti dari buku, jurnal ilmiah, pakar di bidangnya, laporan, literatur dsb. Sumber pengetahuan tsb dijadikan dokumentasi untuk dipelajari, diolah dan diorganisir secara terstruktur menjadi basis pengetahuan.

2. Basis Pengetahuan dan Basis Aturan



Ada beberapa cara merepresentasikan data menjadi basis pengetahuan, seperti dalam bentuk atribut, aturan-aturan, jaringan semantik, frame dan logika. Semua bentuk representasi data tsb bertujuan untuk menyederhanakan data sehingga mudah dimengerti dan mengefektifkan proses pengembangan program.

Dalam pemrograman visual umumnya disediakan sarana untuk mengembangkan tabel-tabel penyimpanan data yang terangkum dalam sebuah database.



3. Mekanisme Inferensi

Adalah bagian sistem pakar yang melakukan penalaran dengan menggunakan isi daftar aturan berdasarkan urutan dan pola tertentu. Selama proses konsultasi antara sistem dan pemakai, mekanisme inferensi menguji aturan satu persatu sampai kondisi aturan itu benar.

Secara umum ada dua teknik utama yang digunakan dalam mekanisme inferensi untuk pengujian aturan, yaitu penalaran maju (*forward reasoning*) dan penalaran mundur (*reverse reasoning*)



4. Fasilitas Belajar Mandiri

Fasilitas ini memungkinkan sistem untuk mengembangkan dirinya sendiri dengan memilah atau mengelompokkan kembali fakta yang sudah ada, memasukkan fakta-fakta baru kedalam basis pengetahuan yang merupakan hasil penurunan (iterasi) dari fakta-fakta sebelumnya dan dapat mengembalikan fakta ke pada mekanisme inferensi sehingga dapat dimintakan fakta lainnya dari pemakai melalui antarmuka pemakai

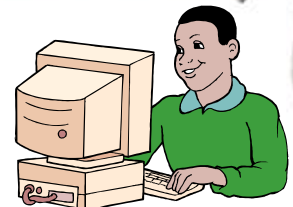


5. Fasilitas Penjelasan Sistem

Merupakan bagian komponen sistem pakar yang memberikan penjelasan tentang bagaimana program dijalankan, apa yang harus dijelaskan kepada pemakai tentang suatu masalah, memberikan rekomendasi kepada pemakai, mengakomodasi kesalahan pemakai dan menjelaskan bagaimana suatu masalah terjadi.

Dalam sistem pakar, fasilitas penjelasan sistem sebaiknya diintegrasikan ke dalam tabel basis pengetahuan dan basis aturan karena hal ini lebih memudahkan perancangan sistem

6. Antarmuka Pemakai



Komponen ini memberikan fasilitas komunikasi antara pemakai dan sistem, memberikan berbagai fasilitas informasi dan berbagai keterangan yang bertujuan untuk membantu mengarahkan alur penelusuran masalah sampai ditemukan solusi.

Syarat utama membangun antarmuka pemakai adalah kemudahan dalam menjalankan sistem, tampilan yang interaktif, komunikatif dan mudah bagi pemakai

REPRESENTASI PENGETAHUAN

(Bagian 1)

Pertemuan keempat

Arti dari Pengetahuan

Pengetahuan merupakan salah satu kata dimana banyak orang mengetahui maknanya, tetapi sulit untuk mendefinisikannya.

Kata pengetahuan memiliki banyak arti, dan kata-kata lain seperti data, fakta dan informasi sering digunakan sebagai sinonim dari pengetahuan.

Epistemology adalah suatu studi tentang pengetahuan, studi ini dihubungkan dengan alam, struktur dan keaslian pengetahuan. Epistemology mengekspresikan pengetahuan dalam tiga bentuk, yaitu secara philosophy, priori dan posteriori

Philosophy digunakan untuk mengekspresikan kebenaran umum (general truth) yang terjadi di alam

Pengetahuan "**a priori**" datang sebelumnya dan bebas dari arti, secara universal benar dan tidak menimbulkan kontradiksi.

Contoh : "semua kejadian pasti ada sebabnya"
"jumlah sudut dalam segitiga 180 derajat"

Pengetahuan "**a posteriori**" adalah pengetahuan yang diperoleh dari arti. Kebenaran atau kesalahan pengetahuan posteriori dapat bervariasi, suatu pernyataan yang benar pada suatu saat, dapat disangkal dan menjadi salah pada saat yang lain.

Klasifikasi lebih lanjut

Pengetahuan diklasifikasikan lebih lanjut kedalam *procedural knowledge*, *declarative knowledge* dan *tacit knowledge*.

Pengetahuan prosedural sering disebut sebagai pengetahuan tentang bagaimana melakukan sesuatu.

Pengetahuan deklaratif mengacu pada pengetahuan bahwa sesuatu itu benar atau salah, sehingga biasanya menghasilkan anjuran atau larangan.

Pengetahuan tacit kadang disebut sebagai "unconscious knowledge" karena tidak dapat diekspresikan dengan bahasa. Contoh : bagaimana cara mengangkat tangan, berjalan, mengendarai sepeda, dsb.

Analogi Wirth

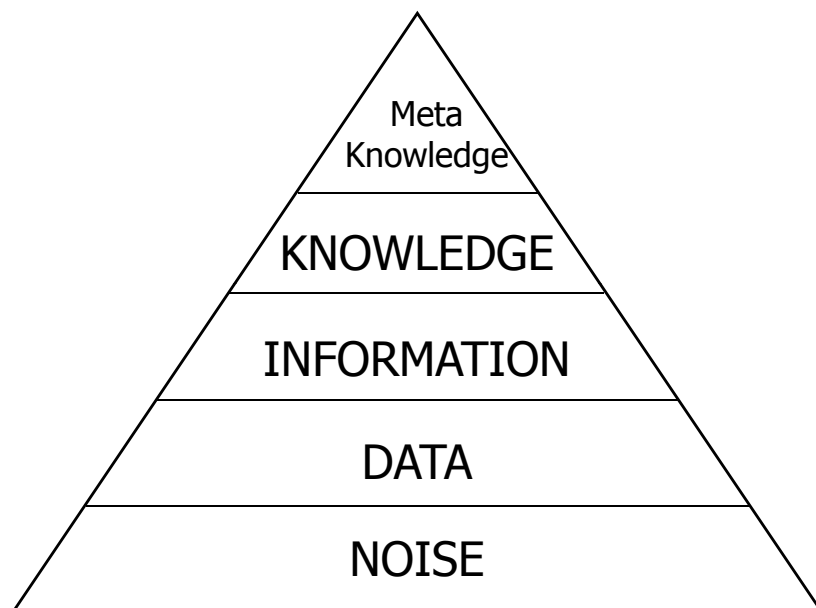
Pengetahuan merupakan hal penting dalam sistem pakar, seperti yang dianalogikan Nicklaus Wirth dalam ekspresi :

Algoritma + Struktur Data = Program

Dan

Pengetahuan + Inferensi = Sistem Pakar

Hirarki Pengetahuan



Bentuk Representasi Pengetahuan

1. Produksi
2. Jaringan Semantik
3. Schemata
4. Frame
5. Logika



Produksi

Produksi atau sering juga disebut himpunan produksi, baris produksi atau baris, merupakan bentuk representasi pengetahuan yang menggunakan Backus-Nour Form (BNF) sebagai metalanguage untuk menentukan sintaks bahasa. Metalanguage adalah suatu bahasa yang digunakan untuk menjelaskan bahasa.

Suatu string atau deretan kata dapat dinyatakan sebagai kata atau kalimat yang valid jika dapat diturunkan atau diderivasi mulai dari simbol start sampai terbentuk kalimat dengan menggunakan baris produksi yang ditentukan.

Grammar merupakan set / rangkaian baris produksi lengkap yang menentukan suatu bahasa secara tidak ambigius.

Parse tree atau Derivation tree merupakan bentuk representasi grafis dari kalimat yang diuraikan kedalam simbol terminal dari seluruh simbol nonterminal yang digunakan untuk mendapatkan kalimat.

Compiler akan membuat parse tree pada saat mencoba menentukan apakah suatu pernyataan dalam program sesuai atau tidak dengan sintaks yang ditentukan.

Cara alternatif penggunaan produksi adalah untuk membuat kalimat dengan mengganti seluruh simbol terminal dengan simbol nonterminal sampai mencapai start (metode backward)



Jaringan Semantik

Merupakan teknik representasi AI klasik yang digunakan untuk informasi proporsional, sehingga jaringan semantik sering disebut juga sebagai **jaringan proporsional**.

Proporsi merupakan kalimat, baik benar maupun salah.

Proporsi merupakan bentuk dari pengetahuan deklaratif karena proporsi menyatakan fakta. Proporsi selalu benar atau salah dan disebut sebagai atomic karena nilai kebenarannya tidak dapat dibagi lagi.

Jaringan semantik pertama kali dikembangkan untuk AI sebagai cara untuk menunjukkan memory manusia dan pemahaman bahasa. Jaringan semantik digunakan untuk menganalisa arti kata dalam kalimat, diterapkan juga pada banyak problem, termasuk representasi pengetahuan.

Struktur jaringan semantik digambarkan secara grafis dalam bentuk nodes dan arcs yang menghubungkannya. Nodes sering juga disebut sebagai objek dan arcs sering juga disebut sebagai links atau edges. Link digunakan untuk mengekspresikan suatu relasi, sedangkan node pada umumnya digunakan untuk menunjukkan objek fisik, konsep atau situasi.

Relasi didalam jaringan semantik sangatlah penting karena relasi tersebut menyediakan struktur pokok untuk pengorganisasian pengetahuan. Tanpa suatu relasi, maka pengetahuan hanya akan merupakan koleksi sederhana dari fakta yang tidak saling berhubungan. Dengan relasi, pengetahuan merupakan struktur kohesif tentang hubungan pengetahuan lain yang dapat disimpulkan.

Tipe Relasi/Link dlm Jaringan Semantik

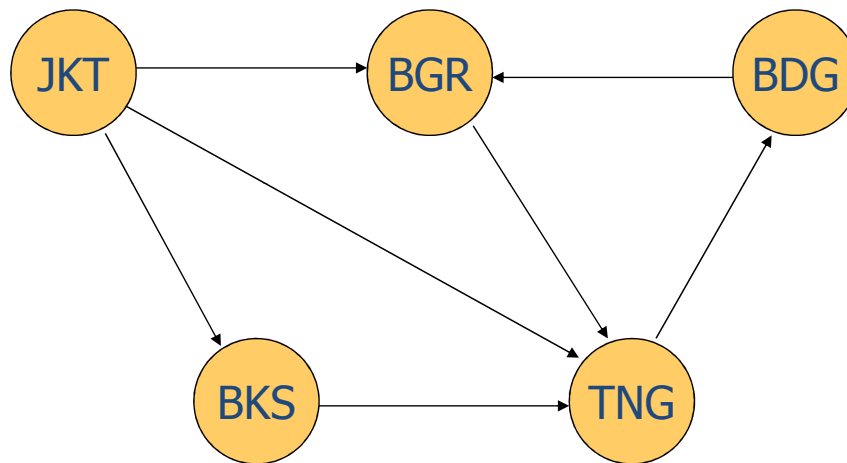
Dua tipe relasi atau link yang sering digunakan pada jaringan semantik adalah is-a (IS-A) dan a-kind-of (AKO).

Link IS-A biasa digunakan untuk menyatakan jarak antar node atau untuk menyatakan suatu objek merupakan anggota dari suatu kelompok objek atau kelas objek tertentu.

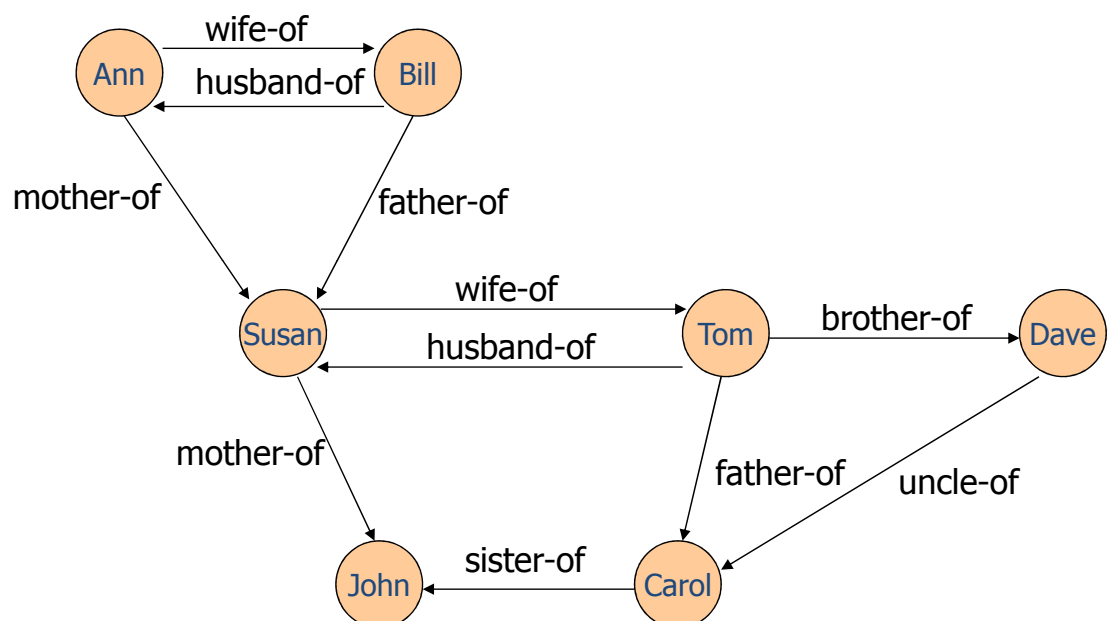
Link AKO digunakan untuk merelasikan satu jenis abjek ke jenis objek lainnya. AKO juga akan menghubungkan jenis individual ke jenis induk dari jenis dimana individual merupakan anak dari jenis tersebut.

Objek didalam jenis/kelas memiliki satu atau lebih atribut secara umum. Setiap atribut memiliki nilai, gabungan atribut dan nilai disebut **properti**.

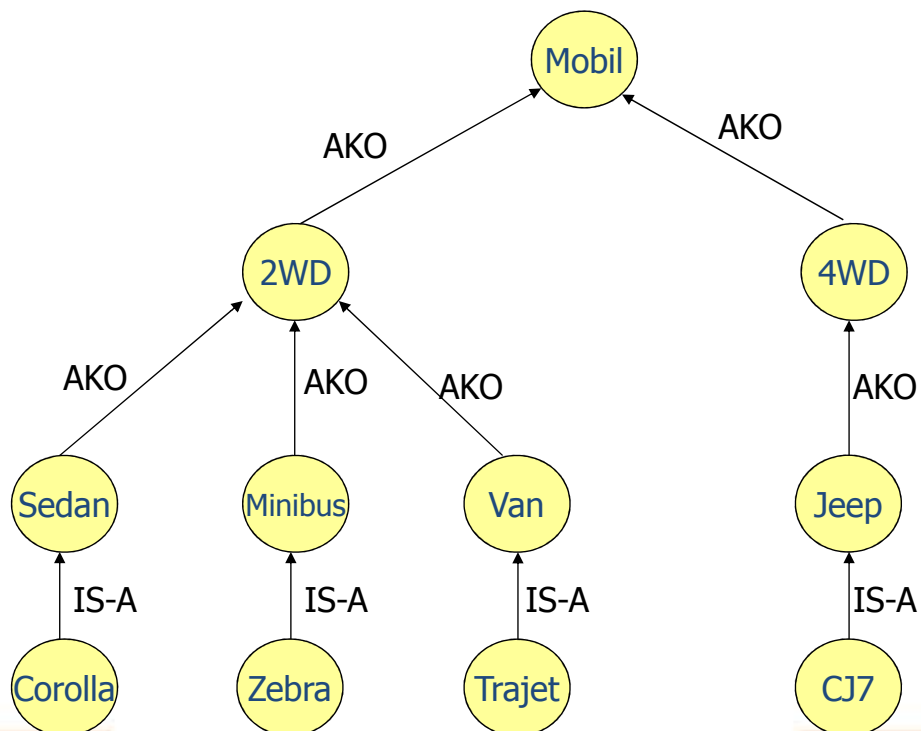
Contoh Jaringan Umum



Contoh Jaringan Semantik



Contoh Jaringan Semantik dengan IS-A & AKO



OAV dlm Jaringan Semantik

Object-attribute-value triple (OAV) atau triplet dapat digunakan untuk memberi karakter semua pengetahuan dalam jaringan semantik dan digunakan dalam sistem pakar MYCIN untuk diagnosa penyakit infeksi.

Representasi triple OAV sangat sesuai untuk pembuatan daftar pengetahuan dalam bentuk tabel dan menterjemah kan tabel ke dalm code komputer dengan induksi baris.

Objeck	Attribute	Value
Apel	warna	Merah
Apel	kuantitas	100
Anggur	warna	Ungu
Anggur	tipe	Tanpa biji

Keterbatasan Jaringan Semantik

1. Kesulitan untuk membuat standar nama link, sehingga berakibat sulit memahami suatu desain jaringan semantik, untuk apa dibuat dan bagaimana dibuat
2. Jaringan semantik, aslinya diusulkan sebagai memori gabungan manusia dimana satu node punya link ke node lainnya, namun ada sekitar 10^4 neuron dan 10^5 link dalam pikiran manusia, kalau semua dipetakan dengan jaringan semantik akan diperlukan waktu yang sangat lama untuk mendapat jawaban, terutama untuk pertanyaan negatif (yang sebenarnya tidak perlu dijawab)
3. Jaringan semantik secara logika tidak memadai karena tidak menentukan pengetahuan dengan cara yang dapat dilakukan oleh logika.
4. Jaringan semantik secara heuristik tidak memadai karena tidak ada cara untuk memasukkan informasi heuristik dalam jaringan untuk mengefisienkan kerja jaringan.

REPRESENTASI PENGETAHUAN (bagian 2)

Pertemuan ke lima

Schemata

Dalam AI, bentuk skema (schema, schemas, shematas) digunakan untuk menjelaskan struktur pengetahuan yang lebih kompleks dibandingkan dengan jaringan semantik.

Bentuk skema berasal dari psikologi dimana menunjukkan organisasi pengetahuan yang kontinyu atau merespond suatu stimuli. Yaitu seperti menciptakan pelajaran hubungan tidak formal antara sebab dan akibat atau cara mengatasinya, penyebab akan diulangi jika menyenangkan atau dihindari jika menyakitkan.

Seseorang tidak perlu memikirkan pengetahuan untuk mengetahui bagaimana atau mengapa melakukan sesuatu yang memang sulit untuk dijelaskan.

Concept Schema

Concept schema merupakan tipe lain dari skema yang dengan skema jenis ini kita dapat menunjukkan konsep.

Jika tiap orang ditanya tentang konsep sesuatu, jawabannya bisa berbeda-beda, karena tiap orang memiliki stereotypes dalam konsep pikirannya masing-masing.

Skema konseptual adalah abstraksi dimana obyek khusus diklasifikasikan dengan properti umum.

Dengan memfokuskan pada property umum dari obyek, maka akan lebih mudah memberi alasan tentangnya tanpa menjadi membingungkan dengan detail yang tidak relevan.

Perbedaan Schema & Semantik

Secara umum, schema memiliki struktur internal di dalam node nya, sedangkan jaringan semantik tidak.

Tabel jaringan semantik seluruhnya adalah pengetahuan tentang node. Jaringan semantik seperti struktur data dalam pengetahuan komputer dimana kunci penelitian juga merupakan data yang disimpan dalam node.

Skema adalah seperti struktur data dimana node berisi record, setiap record mungkin berisi data tunggal, record lain atau pointer ke node yang lainnya.



Frames

Diajukan sebagai metode untuk melihat pemahaman bahasa natural dan bidang lain, frame menyediakan struktur yang cocok untuk menunjukkan objek yang tipikal pada situasi tertentu yang diberikan, seperti stereotype.

Jika jaringan semantik pada dasarnya adalah representasi dua dimensi dari pengetahuan, frame menambahkan dimensi ketiga dengan memungkinkan node untuk mempunyai struktur. Struktur ini dapat berupa nilai sederhana atau frame yang lainnya.

Karakteristik pokok dari frame adalah bahwa frame tersebut menunjukkan pengetahuan yang dihubungkan dengan subyek yang sempit yang mempunyai banyak pengetahuan default.

Frame merupakan kebalikan dari jaringan semantik yang pada umumnya digunakan untuk representasi pengetahuan yang luas

Frame dapat dianalogikan sebagai struktur record pada bahasa tingkat tinggi seperti Pascal.

Berhubungan dengan bidang dan nilai record, dalam frame terdapat slot dan filler yang menentukan stereotype dari suatu objek. Dalam bentuk OAV, slot berhubungan dengan atribut dan filler berhubungan dengan nilai.

Kelebihan frame adalah dalam sistem frame berlaku sifat hirarki dan pewarisan. Dengan menggunakan frame dalam slot dan filler serta pewarisan, maka kita dapat membuat suatu representasi pengetahuan yang kuat.

Khususnya expert system yang berdasarkan pada frame sangat berguna untuk menunjukkan pengetahuan tidak formal karena informasinya disusun berdasarkan pada sebab dan akibat

Contoh Frame untuk obyek Mobil

Slot	Filler
Pabrik	Toyota Astra
Model	Corolla Altis
Tahun	2002
Transmisi	Automatic
Mesin	Bensin
Roda	4
Warna	Silver

Procedural attachments pada Frame

Filler mungkin berupa nilai seperti properti dalam nama slot, atau rentang nilai dalam type slot. Slot mungkin berisi prosedur yang dihadapkan pada slot, atau disebut dengan procedural attachments, yang biasanya terdiri dari tiga type, yaitu :

1. If-needed, merupakan prosedur yang dibuat jika nilai filler yang diperlukan tidak ada atau nilai default tidak sesuai
2. If-added, merupakan prosedur yang dibuat jika akan menambahkan nilai ke slot
3. If-removal, akan digunakan jika suatu nilai dipindahkan atau dikeluarkan dari slot.

Contoh Generic Frame untuk Property

Slot	Filler
Nama	Properti
Spesialisasi dari	Jenis Obyek
Tipe	(rumah, mobil, kapal) Bila ditambah Prosedur : Add_Property
Pemilik	Default : Pribadi Bila diperlukan Prosedur : Find_Pemilik
Lokasi	(rumah, kantor, berpindah)
Status	(bagus, jelek, hilang)
Dalam Garansi	(ya, tidak)

Contoh Frame Mobil – Subframe Generic Properti

Slot	Filler
Nama	Mobil
Spesialisasi dari	Jenis properti
Tipe	(sedan, van, minibus)
Pabrik	(Toyota, Honda, Mitsubhisi)
Lokasi	Berpindah
Roda	4
Transmisi	(manual, automatic)
Mesin	(bensin, disel, gas, tenaga surya)

Contoh Frame mobil

Slot	Filler
Nama	Mobil Heru
Spesialisasi dari	Adalah sedan
Pabrik	Toyota
Pemilik	Heru Sutimbul
Transmisi	Manual
Mesin	Bensin
Status	Bagus
Dalam Garansi	Ya

Kalsifikasi frame berdasarkan aplikasinya

1. Situational frame, berisi pengetahuan tentang bagaimana terjadinya situasi yang diinginkan.
2. Action frame, berisi slot yang menentukan aksi yang akan dilakukan dalam situasi yang diberikan.
3. Casual knowledge frame, gabungan antara situational dan action frame yang digunakan untuk menjelaskan hubungan sebab dan akibat.

Kelemahan Frame

- Pengetahuan dalam frame tidak dapat dipertimbangkan menjadi definisi dari suatu obyek.
- Slot pada frame dapat dikurangi, ditambah atau bahkan dirubah yang juga berarti merubah atribut dari suatu frame obyek, jika ini terjadi pada suatu frame generic yang memiliki sub frame maka akan menyebabkan sifat pewarisan atribut menjadi tidak valid atau dapat dibatalkan karena propertinya tidak lagi bersifat umum (primitif).

Representasi Pengetahuan

(Bagian 3)

Logika dan Himpunan

Pertemuan ke enam

Syllogisme

Adalah logika formal pertama yang dikembangkan oleh filsuf Yunani, **Aristotle** pada abad ke-4 SM.

Syllogisme mempunyai dua *premises* dan satu *conclusion*.

Premise adalah proporsi atau pernyataan yang selalu bernilai benar atau salah karena berdasarkan fakta.

Conclusion adalah kesimpulan yang diturunkan dari dua pernyataan sebelumnya.

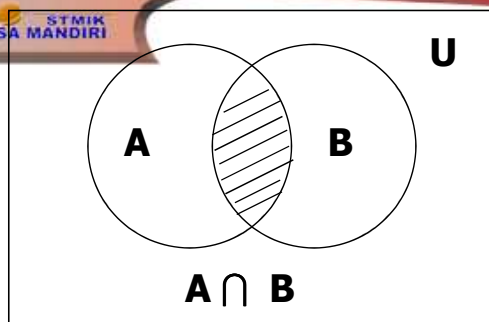
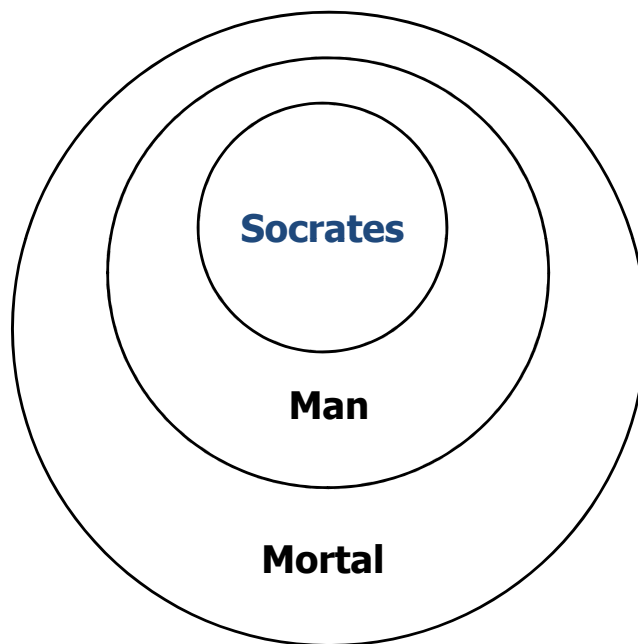
Contoh :

premise : All man are mortal

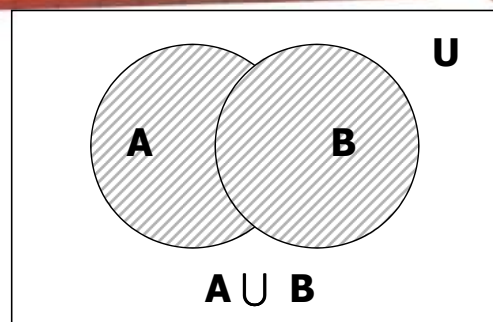
premise : Socrates is a man

Conclusion : Socrates is mortal

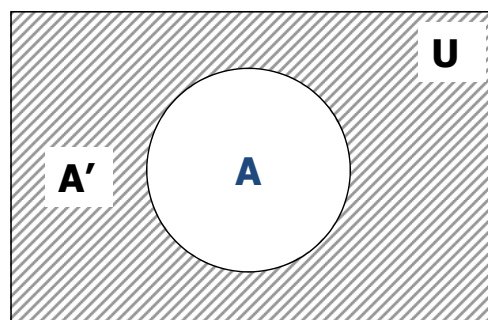
Diagram Venn



Irisan / Intersection



Gabungan / Union



Selain / Complement

Symbolic Logic

Diperkenalkan pertama kali oleh GW Leibnitz abad ke-17 dan disempurnakan oleh ahli matematika Inggris, George Boole yang menerbitkan bukunya tentang symbolic logic pada tahun 1897.

Konsep baru yang diperkenalkan Boole adalah memodifikasi pandangan Aristotle tentang subyek yang harus memiliki keberadaan (existensial import).

Dalam ***pandangan modern*** Boole dapat menyebutkan subyek yang tidak ada atau jenis yang tidak ada elemennya (himpunan kosong) sebagai premises.

Co/: All mermaids swim well

Aksioma

Kontribusi Boole yang lain adalah serangkaian aksioma, yang berisi symbol untuk menunjukkan obyek dan jenis, dan operasi aljabar untuk memanipulasi symbol.

Aksioma merupakan definisi fundamental dari system logika seperti matematika dan logika itu sendiri. Dengan hanya menggunakan aksioma dapat menghasilkan teori.

Teori adalah pernyataan yang dapat dibuktikan dengan menunjukkan bagaimana teori tersebut diperoleh, yaitu dengan menggunakan aksioma.

Logika Proporsional

Kadang disebut sebagai proportional calculus, merupakan logika simbol untuk memanipulasi proporsi, khususnya yang berhubungan dengan manipulasi variabel logika yang mewakili atau menunjukkan suatu proporsi.

Bentuk lain yang digunakan untuk logika proporsional adalah statement calculus atau sentential calculus, dimana statement/sentence atau kalimat pada umumnya dapat diklasifikasikan menjadi 4 type, yaitu :

1. Imperatif / perintah
2. Interogatif / pertanyaan
3. Kalimat seru
4. Deklaratif / pernyataan

Logika proporsional dihubungkan dengan kalimat-kalimat deklaratif yang dapat diklasifikasi sebagai pernyataan benar atau salah. Suatu kalimat deklaratif yang memiliki nilai benar atau salah yang pasti atau dapat ditentukan disebut dengan **statement/pernyataan** atau **proposition/proposisi**. Suatu pernyataan disebut juga sebagai **closed sentence** (kalimat tertutup) karena nilai kebenarannya tidak perlu dipertanyakan lagi.

Contoh : Bujursangkar memiliki empat sisi yang sama (pasti)
Harimau berkaki empat (benar / salah)

Berikut contoh yg tidak termasuk proposisi :

Durian enak sekali (kebenarannya relatif)

Orang itu tinggi (kalimat terbuka)

Compound statement

Adalah pernyataan yang dibuat dengan cara menggabungkan atau menghubungkan beberapa pernyataan tunggal menggunakan konektor logika, spt :

Konektor	Arti
\wedge	AND; konjungsi
\vee	OR; disjungsi
\sim	NOT; negasi
\rightarrow	If . . . Then; kondisional
\leftrightarrow	If and only if; bikondisional

Tabel kebenaran logika compound statement

p	q	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$
T	T	T	T	T	T
T	F	F	T	F	F
F	T	F	T	T	F
F	F	F	F	T	T

Tautology, adalah pernyataan gabungan yang selalu bernilai benar, dimana pernyataan individualnya benar atau salah, mis
: $p \vee \sim p$

Contradiction, adalah pernyataan gabungan yang selalu bernilai salah, dimana pernyataan individualnya benar atau salah, mis
: $p \wedge \sim p$

Contoh pernyataan kondisional

$p \rightarrow q$ dapat diterjemahkan ke bahasa natural menjadi :

p menyatakan q

jika p, maka q

p, hanya jika q

p cukup untuk q

q jika p

q dengan syarat p

p : anda berusia 18 tahun atau lebih

q : anda berhak memilih

Kondisional $p \rightarrow q$ dapat berarti :

Anda berusia 18 tahun atau lebih menyatakan anda berhak memilih

Jika anda berusia 18 tahun atau lebih maka anda berhak memilih

Anda berusia 18 tahun atau lebih, hanya jika anda berhak memilih

Anda berusia 18 tahun atau lebih adalah cukup (memenuhi syarat)
bagi anda untuk memilih

Anda berhak memilih jika anda berusia 18 tahun atau lebih

Anda berhak memilih dengan syarat anda berusia 18 tahun atau
lebih

Logika Predikat order pertama

Problem utama logika proporsional adalah tidak memiliki batasan dan hanya dapat dihubungkan dengan kalimat yang lengkap, yaitu tidak dapat menguji struktur internal suatu pernyataan. Logika proporsional tidak dapat menguji validitas sylogisme spt :

All humans are mortal

All man are humans

Therefor, all women are mortal

Untuk menganalisa kasus lebih luas, dikembangkan logika predikat, yang dihubungkan dengan penggunaan kata khusus yang disebut quantifiers, spt : *all*, *some* dan *no* yang secara eksplisit memberi kuantitas kata lain dan membuat suatu kalimat lebih nyata. Seluruh quantifier dihubungkan dengan how many shg penya cakupan lebih luas dari logika proportional.

Quantifier Universal (All : V)

Digunakan untuk kalimat yang diberi kuantitas memiliki nilai kebenaran yang sama untuk semua pengganti / elemen dalam domain yang sama.

Mis : $(\forall x) (x + x = 2x)$

jika pernyataan $x + x = 2x$ diganti dengan p, jadinya

$(\forall x) (p)$

Jika p adalah kalimat "All triangels are polygon", ditulis

$(\forall x) (\text{if } x \text{ is triangle} \rightarrow x \text{ is polygon})$

dipersingkat dengan predikat function menjadi :

$(\forall x) (\text{triangel } (x) \rightarrow \text{polygon } (x))$

Kesamaan logika :

$(\forall x) P(x) \equiv P(x_1) \wedge P(x_2) \wedge P(x_3) \dots P(x_n)$

Quantifier Eksistensi (some : 3)

Menjelaskan suatu pernyataan yang benar untuk minimal satu anggota domain.

Quantifier eksistensi dapat dibaca atau ditulis dalam bahasa natural menjadi :

there exist, at least one, for some, there is one, some

Untuk menyatakan "*some elephants has three-legged*" :

$$(\exists x) (\text{Elephant}(x) \wedge \text{three-legged}(x))$$

Kesamaan logika

$$(\exists x) P(x) \equiv P(x_1) \vee P(x_2) \vee P(x_3) \dots P(x_n)$$

Quantifier dan Himpunan

Ekspresi Himpunan	Kesamaan Logika Quantifier
$A = B$	$\forall x (x \in A \leftrightarrow x \in B)$
$A \subset B$	$\forall x (x \in A \rightarrow x \in B)$
$A \cap B$	$\forall x (x \in A \wedge x \in B)$
$A \cup B$	$\forall x (x \in A \vee x \in B)$
A'	$\forall x (x \in \mathcal{U} \sim(x \in A))$
\mathcal{U} (Universe)	T (true)
\emptyset (himpunan kosong)	F (false)

Metode Inferensi

GRAPH, TREES, LATTICES

Pertemuan ke sembilan

Pengertian Umum

Graph

Adalah suatu bentuk geometri yang menghubungkan titik-titik (node) dengan garis/tanda panah (arch).

Tree

Bentuk Graph berarah, terbuka dimana selalu memiliki satu node sebagai root/akar atau start, node-node yang memiliki cabang dan node-node yang tidak memiliki cabang yang disebut leave/daun atau end. Root secara otomatis akan menjadi parent bagi node dibawahnya dan begitu juga untuk setiap node yang bukan daun akan menjadi parent untuk node dibawahnya.

Lattice

Bentuk Graph berarah, bisa terbuka bisa tertutup, keunikan lattice dibanding Tree adalah bahwa Lattice dapat memiliki lebih dari satu node sebagai start, dan adanya kemungkinan dimana sebuah node memiliki lebih dari satu parent. Keunikan yang lain bahwa pada Lattice dapat terjadi hanya ada satu node sebagai leave/daun.

Struktur Keputusan

Tree dan Lattice sangat berguna untuk mengklasifikasi obyek karena hirarki alamiahnya dari induk atas anak, contoh : familiy tree.

Aplikasi Tree dan Lattice yang lain adalah dalam membuat gambaran geometris proses pengambilan keputusan, yang dikenal dengan decision tree dan decision lattice, atau secara umum disebut sebagai decision structure / struktur keputusan.

Struktur keputusan adalah skema representasi pengetahuan dan metode pemberian alasan tentang pengetahuan itu.

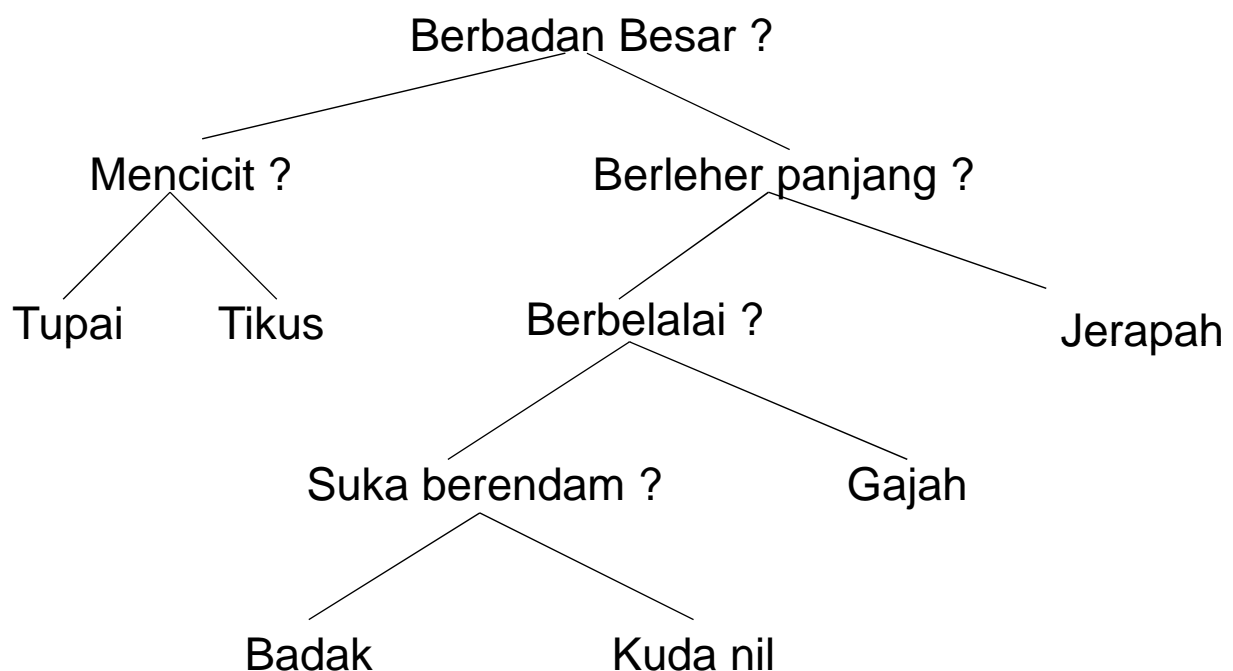
Binary Tree

Jika suatu keputusan adalah binary (memiliki tepat 2 kemungkinan jawaban), maka tree keputusan binary akan mudah dibuat dan sangat efisien.

Dimulai dengan pertanyaan sebagai root, pilihan jawaban (ya atau tidak) sebagai arch/link, setiap pertanyaan berikutnya sebagai simpul bukan daun dan setiap kesimpulan atau jawaban akhir sebagai simpul bukan daun.

Dengan binary tree kita akan mendapatkan maksimum 2^n kesimpulan untuk sebanyak n pertanyaan.

Contoh Tree untuk Menebak Binatang



Struktur dan Baris Produksi

Struktur keputusan dapat secara mekanis diterjemahkan ke dalam baris produksi, hal ini dapat dilakukan dengan mudah dengan mengamati sstruktur dan pembuatan baris IF . . . THEN pada setiap node nya.

Contoh dari tree diatas diterjemahkan kedalam baris sbb:

If Question = "Berbadan besar ?" and Response = "Tidak"
then question := "Apakah mencicit?"

If Question = "Berbadan besar ?" and response ="Ya"
then question :="Apakah berleher panjang ?"

dst

Ruang Stata

Stata adalah koleksi karakteristik yang dapat digunakan untuk menentukan suatu keadaan atau menyatakan suatu obyek tertentu.

Ruang stata merupakan rangkaian pernyataan yang menunjukkan transisi antara pernyataan yang menyatakan bagaimana suatu keadaan dapat terjadi atau bagaimana suatu obyek dapat dicapai.

Untuk menggambarkan ruang stata pada umumnya akan didapatkan bentuk Lattice, atau sering juga disebut sebagai finite state machine (mesin stata hingga) karena menyatakan jumlah keadaan terbatas yang dapat dicapai dengan aturan produksi/mesin yang juga terbatas, atau menggunakan istilah yang lebih umum, diagram keadaan

Contoh Ruang Stata

Mesin Minuman Ringan

Diketahui harga minuman ringan 1200, mesin minuman hanya dapat menerima koin dengan pecahan 500 atau 200. Dapat digambarkan diagram keadaan dengan satu node sebagai simbol start dan satu node sebagai simbol end/penerima atau sukses yang digambar dengan lingkaran bergaris ganda dan node lain untuk keadaan yang bukan start atau sukses dengan lingkaran bergaris tunggal. Untuk arch/panah diberi keterangan dengan pecahan koin yang mungkin (untuk memudahkan gunakan simbol L untuk pecahan 500 dan D untuk pecahan 200).

* Harap digambarkan diagramnya

Contoh lain Ruang Stata

Untuk problem Kera dan Pisang

Diketahui seekor kera berada dalam ruangan, dimana kera berusaha untuk mengambil pisang yang digantung dilangit-langit ruangan. Diruangan terdapat sebuah kursi dan sebuah tangga, pisang hanya akan dicapai oleh kera jika kera menggunakan tanga.

Penyelesaian:

Kasus ini dimulai dalam keadaan yang bervariasi sehingga kita tidak menggambarkan simbol start untuk kasus seperti ini, hanya simbol end atau suksesnya saja yang dibuat selain node yang lain.

*harap digambarkan diagram keadaanya

Struktur AND-OR

Sutau type Tree dan Latice yang berguna untuk menunjukkan masalah dengan rangkaian backward adalah strktur AND-OR.

Cara penggambaran metode backward dengan Struktur AND-OR adalah dengan menempatkan tujuan keadaan atau obyek yang akan dicapai sebagai root/akar dan kemudian menempatkan kemungkinan-kemungkinan untuk mencapai tujuan dibawahnya.

Untuk kondisi OR adalah sebuah node memiliki lebih dari satu cabang kebawah yang dapat dipilih (arch terpisah)

Untuk kondisi AND adalah sebuah node memiliki lebih dari saru cabang yang harus terpenuhi semuanya (arch digabungkan dengan garis lengkung)

Contoh Struktur AND-OR

Tujuan : Menuju tempat kerja

Ketempat kerja dapat dicapai dengan jalan kaki, naik bus atau naik motor. Jalan kaki dapat dilakukan jika kaki cukup sehat.

Untuk naik bus bisa menunggu di halte atau jalan ke terminal.

Untuk naik motor harus dipastikan motor dalam kondisi baik dan ada bensinnya, untuk memastikan kondisi motor baik harus diperiksa sendiri atau orang lain yang memeriksa, untuk mendapatkan bensin membawa motor ke pom bensin atau membeli bensin tanpa motor, yang berarti diperlukan jerigen untuk membawa bensin.

* Buat pohon AND-OR untuk kasus diatas

Metode Inferensi

Logika Deduktif & Sylogisme

Pertemuan ke sepuluh

Umum

Salah satu dari banyak metode yang paling sering digunakan untuk menggambarkan inferensi adalah deduktive logic (logika deduktif), yang digunakan sejak awal untuk menentukan validitas dari *Argumen*.

Satu type argumen logika adalah Sylogisme, yang memiliki dua premises dan satu conclusion.

Dalam argumen, premises digunakan sebagai bukti untuk mendukung conclusion (kesimpulan). Premises disebut juga dengan *antecedent* dan kesimpulan disebut *consequent*.

Karakteristik pokok dari logika deduktif adalah bahwa kesimpulan benar harus mengikuti dari premises yang benar.

Penulisan Argumen

Argumen harus ditulis dalam bentuk yang lebih singkat seperti :

Anyone who can program is intelligent

John can program

∴ Jhon is intelligent

Dimana simbol ∴ digunakan untuk menyatakan berarti/jadi (therefor) dan garis lurus digunakan untuk memisahkan conclusion dari premises

Catagorical Syllogism

Pada umumnya, sylogisme merupakan argumen deduktif yang valid yang mempunyai dua premises dan satu conclusion. Sylogisme klasik merupakan type yang disebut sebagai catagorical syllogisme, dimana premises dan conclusion ditentukan sebagai pernyataan catagorical dari empat bentuk berikut :

<i>Bentuk</i>	<i>Skema</i>	<i>Pengertian</i>
A	All S is P	Universal affirmative
E	No S is P	Universal negative
I	Some S is P	Particular affirmative
O	Some S is not P	Particular negative

Dalam logika, skema kata digunakan untuk menunjukkan bentuk esensial dari argumen. Skema juga menunjukkan bentuk logika dari seluruh sylogisme seperti dalam bentuk berikut :

All M is P

All S is M

∴ All S is P

Subject pada conclusion (S) disebut **minor term**, dan predikat pada conclusion (P) disebut **major term**.

Premis yang berisi minor term disebut **minor premise** yang berisi major term disebut **major premise**

Contoh :

Major Premise : All M is P

Minor Premise : All S is M

Conclusion : All S is P

Adalah sylogisme yang disebut sebagai standard form dengan premise major dan minor yang ditentukan. Subject merupakan sesuatu yang digambarkan/ dijelaskan, sedangkan predicate menggambarkan beberapa property / sifat dari subject.

Contoh lain :

- All microcomputers are computers
- All microcomputers with 512 megabytes
are computers with a lot of memory

Middle Term

Adalah bentuk ketiga dalam sylogisme yang pada umumnya terdapat dikedua premises. Middle term memiliki kedudukan yang penting dalam sylogisme karena dalam sylogisme, kesimpulan tidak dapat di turunkan dari satu premis saja, sehingga middle term pada umumnya akan berfungsi sebagai penghubung dari dua premis

Mood Sylogisme

Mood dari sylogisme ditentukan dengan tiga huruf yang memberikan bentuk major premise, minor premise dan kesimpulan secara respektif, berdasarkan tabel kategori pernyataan.

Contoh :

All M is P

All S is M

••• All S is P

disebut memiliki mood AAA.

* Beri contoh mood lain

Type Sylogisme

Type dari sylogisme ditentukan berdasarkan mood dan figure dari penyusunan minor term (S), major term (P) dan middle term (M), dimana akan ada empat kemungkinan figure susunan S,P dan M, yaitu :

	<i>Figure 1</i>	<i>Figure 2</i>	<i>Figure 3</i>	<i>Figure 4</i>
Major premise	M P	P M	M P	P M
Minor Premise	S M	S M	M S	M S

Contoh :

Bentuk :

All M is P

All S is M

∴ All S is P

Disebut bertype AAA-1

Bentuk :

All M is P

No S is M

∴ No S is P

Disebut bertype AEE-1

dst

Validitas Argumen

Untuk membuktikan validitas argumen sylogisme, digunakan decision procedure, yaitu suatu metode mekanik umum atau algoritma yang dapat menentukan validitas secara otomatis.

Decision procedure untuk sylogisme dapat dikerjakan dengan menggunakan diagram Venn dengan tiga lingkaran yang saling beririsan dimana tiap lingkaran mewakili minor term (S), major term (P) dan middle term (M).

Algoritma penggambaran digram venn

- Beri arsiran untuk daerah yang tidak digunakan / diabaikan
- Pernyataan bentuk universal (A dan E) selalu dikerjakan lebih dulu dari pernyataan particular (O dan I)
- Jika kedua premise universal atau kedua premise particular maka kerjakan major premise dulu
- Untuk daerah yang diwakili dengan some beri tanda *
- Jika ada dua daerah berdampingan yang harus diberi tanda *, maka tanda * diletakkan pada garis yang membatasi kedua daerah tsb
- Untuk daerah yang sudah diarsir tidak dapat lagi diberi tanda *

Contoh

Bentuk AEE-1

All M is P

No S is M

∴ No S is P

Bukan sylogisme valid, dengan contoh :

All microcomputers are computers

No mainframe is a microcomputer

∴ No mainframe is a computer

* Buktikan dengan digram Venn

Contoh

Bentuk EAE-1

No M is P

All S is M

∴ No S is P

Sylogisme valid, dengan contoh :

No microcomputers is mainframe

All laptops is microcomputers

∴ No laptops mainframe

* Buktikan dengan digram Venn

Contoh

Bentuk IAI-4

Some P are M

All M are S

∴ Some S are P

Sylogisme valid, dengan contoh :

Some computers are laptops

All laptops are transportable

∴ Some transportable are computers

* Buktikan dengan digram Venn

Metode Inferensi

Argumen & Logika Proporsional

Pertemuan ke sebelas

Argumen Proporsional

Adalah argumen yang berisi proporsi, sehingga dapat diekspresikan menjadi bentuk formal dalam logika proporsional.

Contoh :

If there is power, the computer will work

There is power

∴ The computer will work

Dapat diekspresikan dengan huruf (logika proporsional) :

p = There is power

q = The computer will work

Sehingga menjadi :

$p \rightarrow q$

p

∴ q

Skema inferensi untuk proporsional diatas disebut dengan berbagai istilah : *Direct Reasoning*, ***modus ponens***, *law of detachment* atau *assuming the antecedent*

Notasi modus ponens

Notasi lain untuk skema modus ponens :

$$p \rightarrow q, p; \therefore q$$

dimana koma digunakan untuk memisahkan premise dan titik koma untuk memisahkan conclusion dari premise.

Bentuk umumnya :

$$P_1, P_2, \dots, P_n; \therefore C$$

dimana P = premise dan C = Conclusion

Kesamaan Logika Pada modus ponens

Perhatikan lagi bentuk :

$$p \rightarrow q, p; \therefore q$$

Dalam logika tanda koma (,) setara dengan notasi \wedge (AND) dan tanda titik koma (;) setara dengan notasi \rightarrow (then), sehingga skema diatas dapat ditulis menjadi :

$$(p \rightarrow q) \wedge p \rightarrow q$$

* Tanda kurung digunakan karena notasi \rightarrow (then) memiliki hirarki lebih rendah dari notasi \wedge (AND)

Validitas modus ponens

Karena modus ponens adalah bentuk khusus dari sylogisme, maka suatu modus ponens dikatakan valid jika untuk kedua premise bernilai benar (T) maka nilai conclusion juga benar (T).

Tabel kebenaran untuk modus ponens diatas :

p	q	$p \rightarrow q$	$(p \rightarrow q) \wedge p$	$(p \rightarrow q) \wedge p \rightarrow q$
T	T	T	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

Argumen Palsu

Perhatikan modus ponens berikut :

If there are no bugs, then program compiles

There are no bugs

∴ The Program compiles

Bandingkan dengan modus ponens berikut :

If there are no bugs, then the program compiles

There program compiles

∴ There are no bugs

Salah satunya palsu, dapat dibuktikan dengan tabel kebenaran logika

Aturan Inferensi untuk logika yang benar

1. Hukum data skema : $p \rightarrow q$
(modus ponendo ponens) p

 $\therefore q$

2. Hukum kontra positif : $p \rightarrow q$

 $\therefore \sim q \sim p$

3. Hukum modus tollens : $p \rightarrow q$
(modus tollendo tollens) $\sim q$

 $\therefore \sim p$

4. Aturan Rangkaian : $p \ q$
(hukum sylogisme) $q \ r$

 $\therefore p \ r$

5. Hukum inferensi disjuntif : $p \vee q$ $p \vee q$
(modus tollendo ponens) $\sim p$ $\sim q$

 $\therefore q$ $\therefore p$

6. Hukum negasi ganda : $\sim(\sim p)$

 $\therefore p$

7. Hukum penyederhanaan : $p \wedge q$ $p \vee q$

 $\therefore p$ $\therefore q$

8. Hukum Konjungsi :

$$\begin{array}{c} p \\ q \\ \hline \therefore p \wedge q \end{array} \quad \begin{array}{c} p \\ \hline \therefore p \vee q \end{array}$$

9. Hukum De Morgan(1)

$$\begin{array}{c} \sim(p \vee q) \\ \hline \therefore \sim p \vee \sim q \end{array} \quad \begin{array}{c} \sim(p \wedge q) \\ \hline \therefore \sim p \wedge \sim q \end{array}$$

10. Hukum De Morgan (2) :

$$\begin{array}{c} \sim(p \wedge q) \\ p \\ \hline \therefore \sim q \end{array} \quad \begin{array}{c} \sim(p \vee q) \\ q \\ \hline \therefore \sim p \end{array}$$

Kondisional dan variant-nya

Kondisional	$p \rightarrow q$
Konversi	$q \rightarrow p$
Inversi	$\sim p \rightarrow \sim q$
Kontra positif	$\sim q \rightarrow \sim p$

Inferensi pada argumen dgn lebih dari 2 premises

Perhatikan contoh berikut :

Harga chip naik hanya jika nilai Yen naik.

Nilai Yen naik hanya jika nilai dollar turun dan
jika nilai dollar turun maka nilai yen naik.

Ketika harga chip naik,
nilai dollar menurun.

Proporsi dapat ditentukan sbb :

C = harga chip naik

Y = nilai Yen naik

D = nilai dollar turun

Sehingga argumen diatas dapat ditulis :

$C \rightarrow Y$

$(Y \rightarrow D) \wedge (D \rightarrow Y)$

C

$\therefore D$

Disederhanakan menjadi :

$C \rightarrow Y$

$Y = D$

C

$\therefore D$

Disubstitusi menjadi :

$C \rightarrow D$

D

$\therefore D$

Memenuhi hukum modus ponens yang valid

Batasan Logika Proporsional

Logika proporsional tidak dapat menginferensi atau membuktikan kebenaran sylogisme klasik, seperti :

All men are mortal

Socrates is a man

therefore, Socrates is mortal

Sebelum dirubah menjadi sylogisme proporsional seperti :

If Socrates is a man, then Socrates is mortal

Socrates is a man

therefor, Socrates is mortal

Yang merupakan bentuk modus ponens

Metode Inferensi

Rangkaian Forward & Backward

Umum

Rangkaian

merupakan salah satu metode inferensi yang berusaha menghubungkan masalah dengan solusinya.

Rangkaian Forward

rangkaian yang menghubungkan keadaan/fakta yang dihadapi sebagai alasan kepada suatu kesimpulan.

Rangkaian Backward

rangkaian yang menghubungkan suatu keadaan/fakta yang dihadapi sebagai suatu hipotesa kepada fakta-fakta lain yang dapat mendukung hipotesa tersebut

Contoh kasus Forward

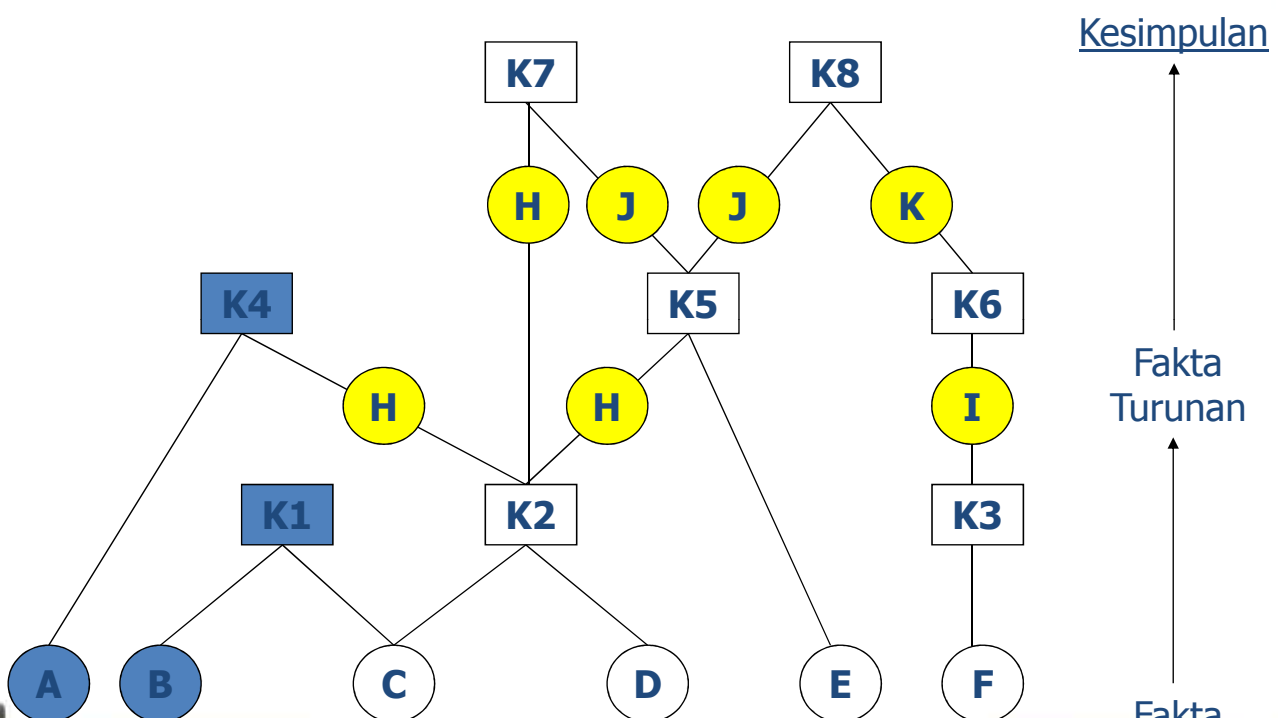
Pada suatu keadaan kita sedang mengendarai mobil, kemudian diikuti mobil polisi dengan serine dan lampu menyala. Dengan rangkaian forward kita dapat berkesimpulan polisi dibelakang kita sedang berusaha mengejar atau menghentikan mobil lain. Bila kemudian polisi sudah tepat disamping mobil kita dan memberi isyarat dengan tangan (keadaan berubah/fakta baru) kita bisa berkesimpulan bahwa polisi meminta kita untuk berhenti. Setiap keadaan berubah atau ada fakta baru kita selalu memiliki alasan baru untuk membuat memilih kesimpulan yang baru atau tetap pada kesimpulan sebelumnya.

Contoh kasus Backward

Untuk keadaan yang sama bila kita menggunakan rangkaian backward kita menjadikan keadaan sebagai hipotesa yang dianggap benar dan berfikir apa sebabnya (mengapa polisi ingin kita berhenti ?) lalu kita mengingat apa yang terjadi sebelumnya (mencari fakta pendukung), apakah kita salah jalur ?, atau melebihi batas kecepatan ?, atau lampu rem tidak menyala ? atau ada yang salah dengan plat nomer ? .

Jika salah satu atau beberapa pertanyaan dijawab dengan ya berarti kita mendapatkan hipotesa baru lagi yang membutuhkan fakta pendukung baru lagi.

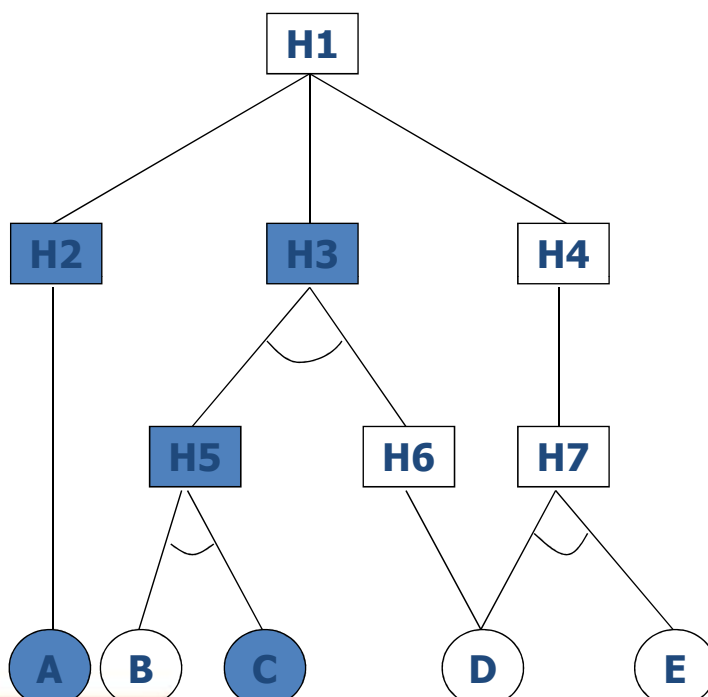
Rangkaian Forward sebagai Bottom-up Reasoning



Karakteristik Forward

- Kesimpulan yang benar hanya dapat diturunkan dari fakta-fakta yang benar
- Suatu fakta dapat membangkitkan / mendukung lebih dari satu kesimpulan
- Semakin banyak fakta yang didapat atau mungkin terjadi akan mengakibatkan semakin banyak pula kesimpulan yang dapat dibuat
- Kesimpulan akan menjadi lebih valid jika didukung oleh lebih banyak fakta
- Melakukan pencarian dengan metode breadth-first, yang berusaha mengumpulkan semua fakta yang mungkin pada setiap level

Rangkaian Backward sebagai Top-Down Reasoning



Hipotesa Awal

Hipotesa Lanjutan

Bukti (Fakta)

Karakteristik Backward

- Bukti adalah hal terpenting dalam backward
- Salah satu cara terpenting mendapatkan bukti adalah dengan mengajukan pertanyaan
- Pertanyaan yang benar (sesuai dengan hipotesa awal) akan mengarahkan kepada bukti yang diinginkan
- Pertanyaan juga berfungsi untuk mempersempit ruang hipotesa
- Melakukan pencarian dengan metode depth-first yang berusaha mendapatkan bukti lanjutan untuk mendukung bukti sebelumnya

Perbandingan Forward & Backward

Forward	Backward
Perencanaan, Pemantauan, kontrol saat ini kemasa depan	Diagnosa keadaan sekarang ke masa lalu
Antecedent terhadap sebab, Bergerak dengan perubahan data	Akibat terhadap antecedent, Bergerak dgn perubahan tujuan
Bergerak maju untuk menemukan kesimpulan yang mengikuti fakta	Bergerak mundur untuk mendapatkan fakta yang mendukung hipotesa
Pencarian melebar (breadth-first)	Pencarian mendalam (depth-first)
Sebab menentukan pencarian	Akibat menentukan pencarian

Beberapa Metode Lain Untuk Inferensi

Analogi

Ide pokok dari pemberian alasan dengan analogi adalah mencoba dan menghubungkan situasi lama (yang pernah terjadi) sebagai penuntun ke situasi yang baru.

Analogi cukup baik diterapkan dalam kehidupan sehari-hari karena begitu banyaknya situasi baru yang kita dapati dalam keseharian kita.

Analogi tidak memiliki formula pembuktian, sehingga analogi dapat disebut sebagai pemberian alasan secara heuristic yang kadang dapat digunakan sebagai hipotesa awal sebelum melakukan rangkaian backward, atau mempersempit ruang pencarian pada rangkaian forward

Jika seorang pasien pergi ke dokter, dokter akan menanyakan gejala apa yang kita rasakan / sakit apa? (bukan apa penyebab kita sakit). Jika gejala yang kita alami sama dengan orang lain yang diketahui menderita sakit "X", maka dokter mungkin menyimpulkan dengan analogi bahwa kita juga menderita sakit "X".

Diagnosa ini mungkin salah karena pada dasarnya tiap orang berbeda (unik) atau gejala yang sama bisa berasal dari penyakit yang berbeda, sehingga dokter tidak menyimpulkan tapi menjadikan diagnosa tadi sebagai hipotesa awal. Dengan demikian sudah menghindari pengambilan kesimpulan yang salah namun sudah mempersempit kemungkinan dan menghemat waktu dan biaya dalam melakukan serangkaian test fisik.

Generate and Test

Merupakan salah satu metode inferensi dalam AI klasik yang berusaha mendapatkan solusi dengan mengenerate atau membuat yang mirip solusi (diduga) melakukan test kemudian berhenti bila itu ternyata solusinya atau membuat lagi yang lain dan melakukan test lagi sampai ditemukan solusi.

Metode ini pertama kali digunakan dalam sistem pakar DENDRAL (1965) untuk melengkapi penentu struktur molekul organik, dengan membuat semua struktur organik yang potensial dan mengetestnya satu persatu.

Program lain yang menggunakan metode ini adalah AM (Artificial Mathematician) untuk menyimpulkan konsep matematika yang baru.

Plan-Generate-Test

Variasi dari Generate-and-test dimana digunakan program perencanaan untuk mengurangi kemungkinan sehingga lebih sedikit yang harus dibuat dan dites.

Contoh sistem pakar yang menggunakan metode ini antara lain adalah MYCIN untuk diagnosa matematika dan juga perencanaan perawatan/pengobatan pasien.

Perencana MYCIN pertama kali membuat daftar prioritas dari obat terapi untuk pasien yang sensitif terhadap obat. Generator kemudian akan mengambil daftar prioritas dari perencana dan membuat subdaftar dari satu atau dua macam obat, sub daftar obat ini kemudian di test terhadap pasien untuk melihat efek kesembuhan, infeksi atau alergi pasien

Ketidakpastian

Ketidakpastian adalah sebutan untuk menyatakan kekurangan informasi yang memadai untuk mengambil suatu keputusan.

Ketidakpastian menjadi masalah karena menghalangi kita untuk membuat keputusan yang paling baik dan bahkan akan mengakibatkan kita membuat keputusan yang jelek. Dalam dunia kesehatan misalnya, ketidakpastian akan mengakibatkan perawatan yang tidak baik dan terapi yang salah. Dalam dunia bisnis, ketidakpastian dapat berarti hilangnya peluang untuk mendapat keuntungan atau bahkan mengakibatkan kerugian yang besar secara finansial.

Contoh sistem pakar klasik yang sukses dalam pengambilan keputusan dibawah ketidakpastian adalah PROSPECTOR yang digunakan untuk membantu eksplorasi bahan tambang (mineral), dan MYCIN dalam diagnosa kesehatan

Dalam PROSPECTOR, kesimpulan datang ketika semua bukti yang diperlukan untuk mengambil kesimpulan tidak diketahui dengan pasti.

Meskipun mungkin untuk mencapai kesimpulan yang lebih baik dan dapat dipercaya dengan melakukan test, tapi akan memakan waktu dan biaya dalam melakukan test.

Dalam perawatan kesehatan juga akan memakan waktu dan biaya untuk melakukan test, dan juga selama test berlangsung, keadaan pasien bisa bertambah parah atau bahkan meninggal. Jadi lebih efektif kita ambil kesimpulan dengan keyakinan 95% daripada melakukan test dulu untuk mendapatkan keyakinan 98%.

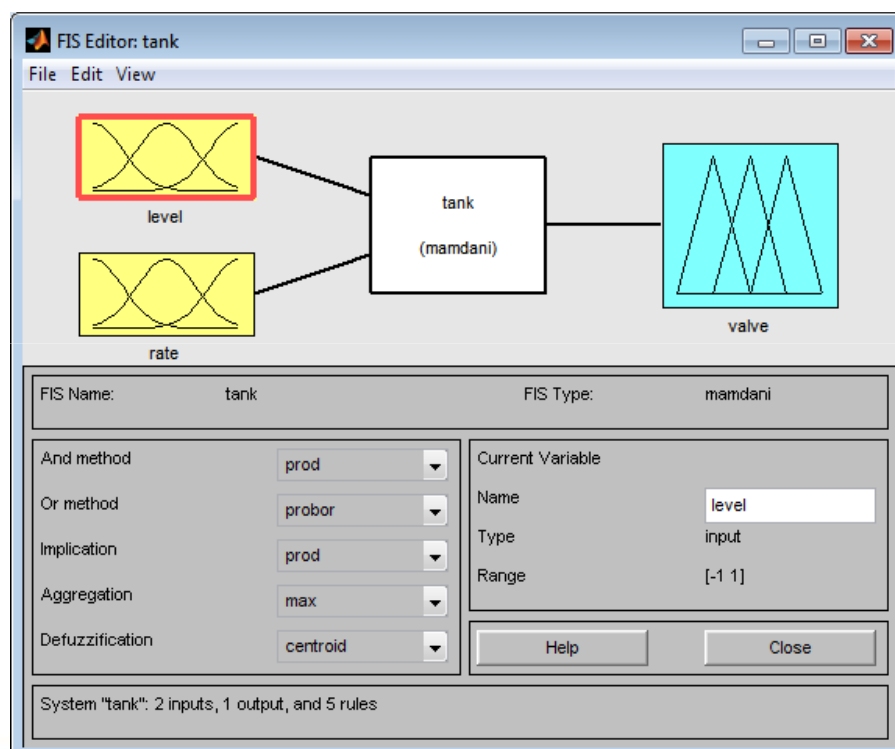
Pengertian Soft Computing

- komputasi yang melibatkan data-data dengan ketidakpastian, ketidakakuratan maupun kebenaran yang parsial
- Teknik-teknik yang digunakan antara lain:
 - Fuzzy Inference Systems
 - ANFIS
 - Jaringan Syaraf Tiruan
 - Algoritma Genetik

Fuzzy Inference System (FIS)

- Sistem inferensi yang mendasarkan logikanya dengan logika yang menyerupai bahasa sehari-hari.
- Dalam keseharian, terkadang kita menyukai informasi misalnya suhu 25 derajat selsius dinyatakan dengan suhu “Sedang”.
- Komponen FIS adalah Fungsi keanggotaan, Rule dan Defuzzifikasi.

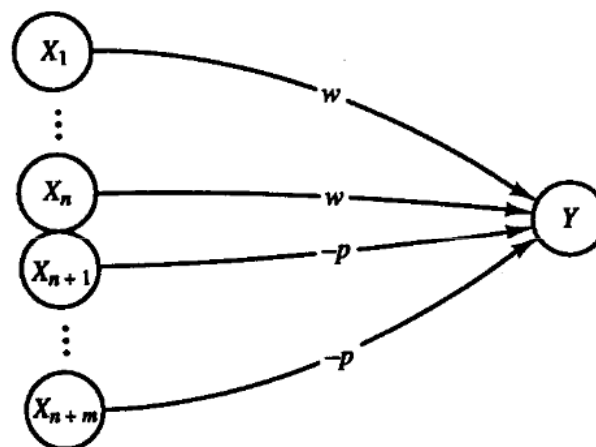
Contoh FUZZY



Jaringan Syaraf Tiruan (JST)

- Berusaha meniru mekanisme syaraf biologis (*Artificial Neural Network*).
- Terdiri dari susunan: Neuron dengan fungsi aktivasi antara lain: *linear* dan *sigmoid*.
- Pembelajaran diperlukan untuk merubah bobot (yg sesuai diperkuat yang tidak sesuai diperlemah).
- Terdiri dari satu lapis masukan, satu lapis keluaran dan beberapa lapis tersembunyi.

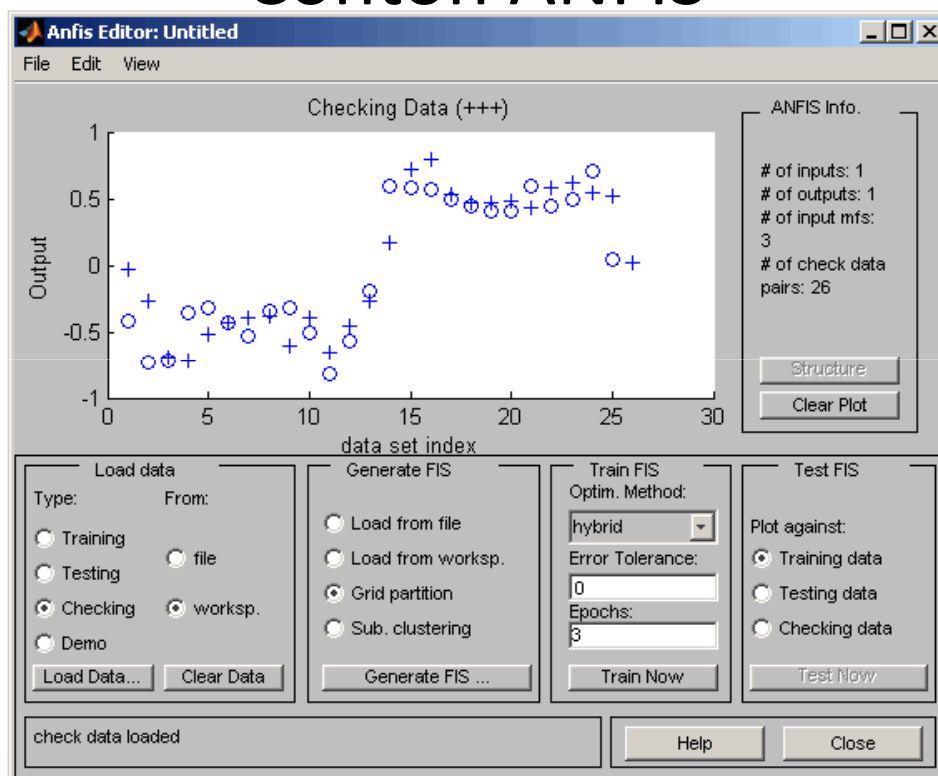
Model JST Pertama



Adaptive Neuro Fuzzy Inference System (ANFIS)

- Gabungan antara FIS dengan Jaringan Syaraf Tiruan (JST).
- Karena rumitnya membuat rule-based, dengan JST rule dibuat oleh sistem itu sendiri dengan mekanisme "*Learning*".

Contoh ANFIS



Genetic Algorithm

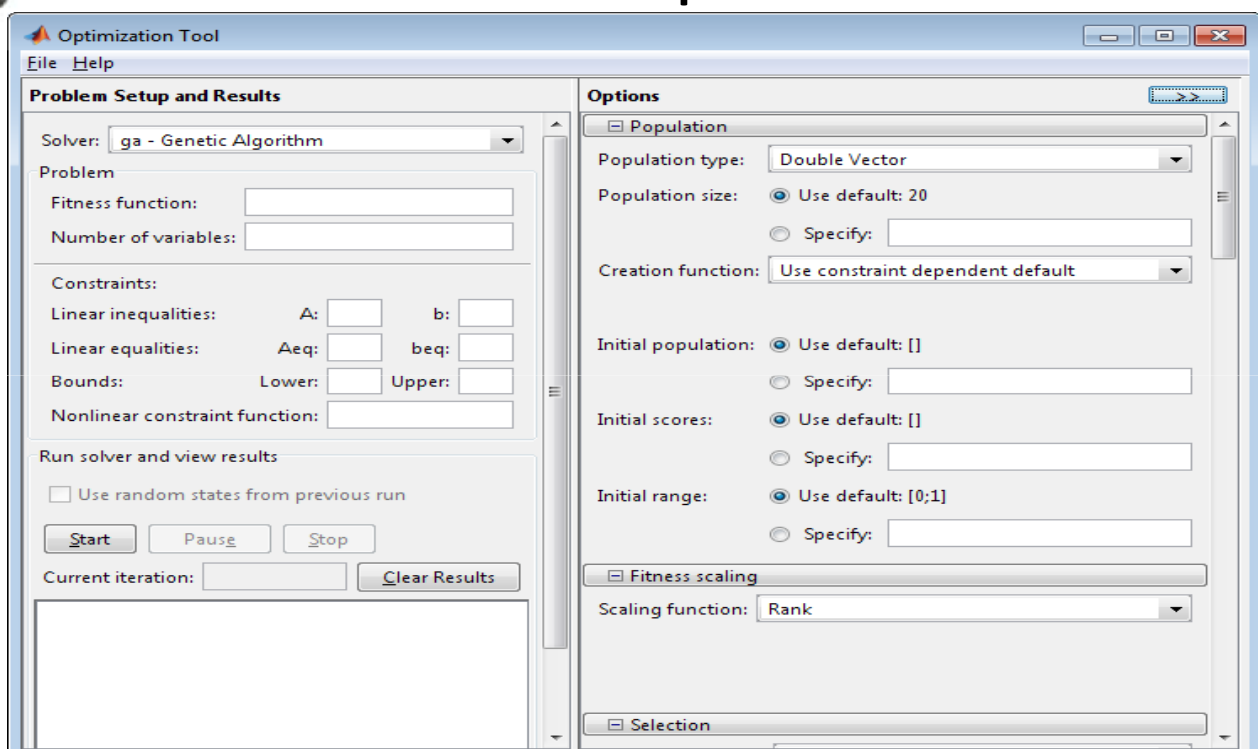
- Diperkenalkan pertama kali oleh Holland tahun 60-an.
- Berusaha meniru mekanisme evolusi makhluk hidup dalam menyelesaikan kasus tertentu.
- Kasus yang diselesaikan biasanya dalam optimasi dimana nilai minimum/maksimumnya banyak sehingga jika diselesaikan dengan aljabar biasa, sistem akan terjebak dalam local minimum/maksimum.

Struktur Umum

- Populasi, istilah pada teknik pencarian yang dilakukan sekaligus atas sejumlah solusi yang mungkin
- Kromosom, individu yang terdapat dalam satu populasi dan merupakan suatu solusi yang masih berbentuk simbol.
- Generasi, populasi awal dibangun secara acak sedangkan populasi selanjutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi

- Fungsi Fitness, alat ukur yang digunakan untuk proses evaluasi kromosom. Nilai fitness dari suatu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut.
- Generasi berikutnya dikenal dengan anak (*offspring*) terbentuk dari gabungan dua kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyilang (*crossover*).
- Mutasi, operator untuk memodi-fikasi kromosom.

Toolbox GA pada MATLAB



Pengenalan CLIPS

Pertemuan ke empat belas

CLIPS adalah:

- ❖ Salah satu bahasa pemrograman untuk sistem pakar
- ❖ Dibuat dari Bahasa C
- ❖ Tipe inference yang dipakai rangkaian forward
- ❖ Dikembangkan oleh NASA

Eleman Pokok CLIPS:

- Daftar Fakta (Fact List)
- Basis Pengetahuan (Knowledge Base)
- Mesin Inference (Inference Engine)

FAKTA:

- Fakta dibuat dari field yang berupa kata, string, atau angka.
- Field pertama dari fakta secara normal digunakan untuk menunjukkan tipe informasi yang disimpan dalam fakta dan disebut dengan relasi. Template fakta dapat digunakan untuk dokumen tipe informasi yang disimpan dalam fakta.

BARIS:

- Baris merupakan komponen kedua dari sistem CLIPS.
- Suatu baris dibagi menjadi LHS dan RHS.
- LHS dari baris dapat dianggap sebagai porsi IF dan RHS dapat dianggap sebagai porsi THEN.
- Baris dapat mempunyai pola multiple dan aksi.

MESIN INFERENSI:

- Baris yang mempunyai pola yang dipenuhi dengan fakta akan membuat aktivasi yang ditampilkan pada agenda.
- Refraction akan mencegah baris dari pengaktifan secara konstant oleh fakta yang lama.

NOTASI

()	:	Diisi Fakta
[]	:	Pilihan
<>	:	Harus diisi angka
{ }	:	Menyatakan Himpunan Fakta
" "	:	Berarti String

Mengisi Daftar Fakta

Sintaks:

(assert <<<fact>>>)

Contoh:

CLIPS> (assert (emergency fire)) enter
CLIPS>

Note: Fakta selain diketik langsung, bisa juga diimport dari text editor lainnya.

Melihat Isi List Fakta

```
CLIPS> (facts) enter  
f-1      (emergency fire)  
CLIPS>
```

Perintah-perintah Dasar CLIPS

- WATCH: digunakan untuk debugging program
- MATCHES: debugging pola baris.
- SET BREAK: Mengeset batas debugging

- Pernyataan dan penarikan kembali fakta, baris string dan aktivasi dapat dilakukan dengan menggunakan perintah WATCH.
- Perintah MATCHES akan menampilkan fakta yang telah menggabungkan pola baris termasuk daftar gabungan bagian untuk suatu baris.
- Perintah SET BREAK memungkinkan pembuatan/pelaksanaan untuk diakhiri sebelum baris di-fired.
- Perintah PRINTOUT dapat digunakan untuk output informasi dan RHS baris. Perintah CLEAR digunakan untuk memberi initial kembali suatu pernyataan dari lingkungan CLIPS.