

Terakhir...

- Representasi Pengetahuan
 - Logika Proposisi
 - Logika *First Order*
- Penalaran dengan Inferensi

Kecerdasan Buatan

Pertemuan 07

Sistem Pakar Berbasis Aturan (*Rule-Based Expert System*, RBES)

Kelas 10-S1TI-03, 04, 05

Husni

Lunix96@gmail.com

<http://Komputasi.wordpress.com>

Outline

- Pendahuluan
- Komponen Sistem Berbasis Aturan
- Struktur Sistem Pakar Berbasis Aturan
- Jenis Sistem Pakar Berbasis Aturan
 - *Forward chaining*
 - *Backward chaining*
- Resolusi Konflik
- Keuntungan & Kemampuan RBES
- Jenis Sistem Pakar
- Contoh Sistem Pakar
- Rangkuman
- Tugas

Pendahuluan

- Sistem berbasis Aturan (*Rule-based systems*), dikenal pula sebagai *production systems* atau *expert systems* (Sistem Pakar)
- Bentuk paling sederhana dari aplikasi kecerdasan buatan.
- **Sistem pakar**: sistem yang meniru penalaran pakar (manusia) dalam menyelesaikan suatu masalah intensif pengetahuan
- Sistem berbasis aturan menggunakan himpunan aturan dalam merepresentasi pengetahuan, memberitahukan apa yang harus dilakukan atau apa yang harus disimpulkan. Tidak deklaratif dan statis.

Sistem Berbasis Aturan

- Paling sederhana, dibuat menggunakan himpunan pernyataan dan aturan (*rule*). Aturan diekspresikan dalam pernyataan if-then (disebut aturan IF-THEN atau aturan produksi):
IF P THEN Q Ekuivalen dengan: **$P \Rightarrow Q$**
- Terdiri dari sehimpunan aturan IF-THEN, himpunan fakta dan beberapa interpreter untuk mengendalikan aplikasi dari aturan berdasarkan fakta.
- Area masalah tidak boleh terlalu besar karena dapat menyebabkan jumlah aturan sangat banyak, sehingga penalaran tidak efisien.

Elemen dari Sistem Berbasis Aturan

- **Himpunan fakta:** Pernyataan dan relevan dengan statis awal dari sistem. Berupa data dan kondisi.
Contoh: $suhu < 0$
Suhu adalah data dan kondisi adalah < 0 .
- **Himpunan aturan (rule):** Semua tindakan yang perlu diambil untuk menyelesaikan masalah, berdasarkan fakta. Aturan menghubungkan fakta dalam IF dengan *action* pada bagian THEN.
- **Kriteria berhenti:** kondisi yang menentukan bahwa solusi telah ditemukan atau tidak ada.

Contoh

Data	Kondisi
Musim Dingin	
Temperatur	<0, >0
Gerak Angin	Kencang, sepoi-sepoi
Jalan	Licin, tidak licin
Cuaca	Dingin, hangat, panas

Aturan (*Rule*)

Premis:

IF Temperatur < 0

AND IF Gerak Angin Kencang

OR IF Jalan Licin

Kesimpulan: THEN Cuaca Dingin

Aturan (*Rules*)

- Terdiri dari dua bagian: bagian IF dan bagian THEN.
- Bagian IF disebut *antecedent* atau *premise* (atau *condition*)
- Bagian THEN dinamakan *consequent* atau *conclusion* (atau *action*).
- Ekspresi aturan sederhana :
IF alasan
THEN tindakan
- Contoh:
IF musim adalah dingin
THEN cuaca adalah dingin.

Rule dengan Banyak Alasan

- Antar alasan dihubungkan dengan OR atau AND
- Dihubungkan dengan AND :

```
IF Alasan1  
AND Alasan2  
...  
AND AlasanN  
THEN Tindakan
```

```
IF Musim Dingin  
AND temperatur <0 derajat  
AND Berangin  
THEN Cuaca adalah Dingin
```

- Dihubungkan dengan OR:

```
IF Alasan1  
OR Alasan2  
...  
OR AlasanN  
THEN Tindakan
```

```
IF Musim Dingin  
OR Temperature <0 derajat  
OR Berangin  
THEN Cuaca Dingin
```

Alasan dikombinasikan oleh AND dan OR

- Bentuk dasar:

IF Alasan1

AND Alasan2

...

OR AlasanN

THEN Tindakan

- Contoh:

IF Musim Dingin

AND Temperature < 0 Derajat

OR Berangin

THEN Cuaca Dingin

Banyak Akibat atau Tindakan

- Bentuk dasar:

IF alasan

THEN akibat1

akibat2

...

akibatN

- Contoh:

IF Musim Dingin

THEN Temperature Rendah

Jalan Licin

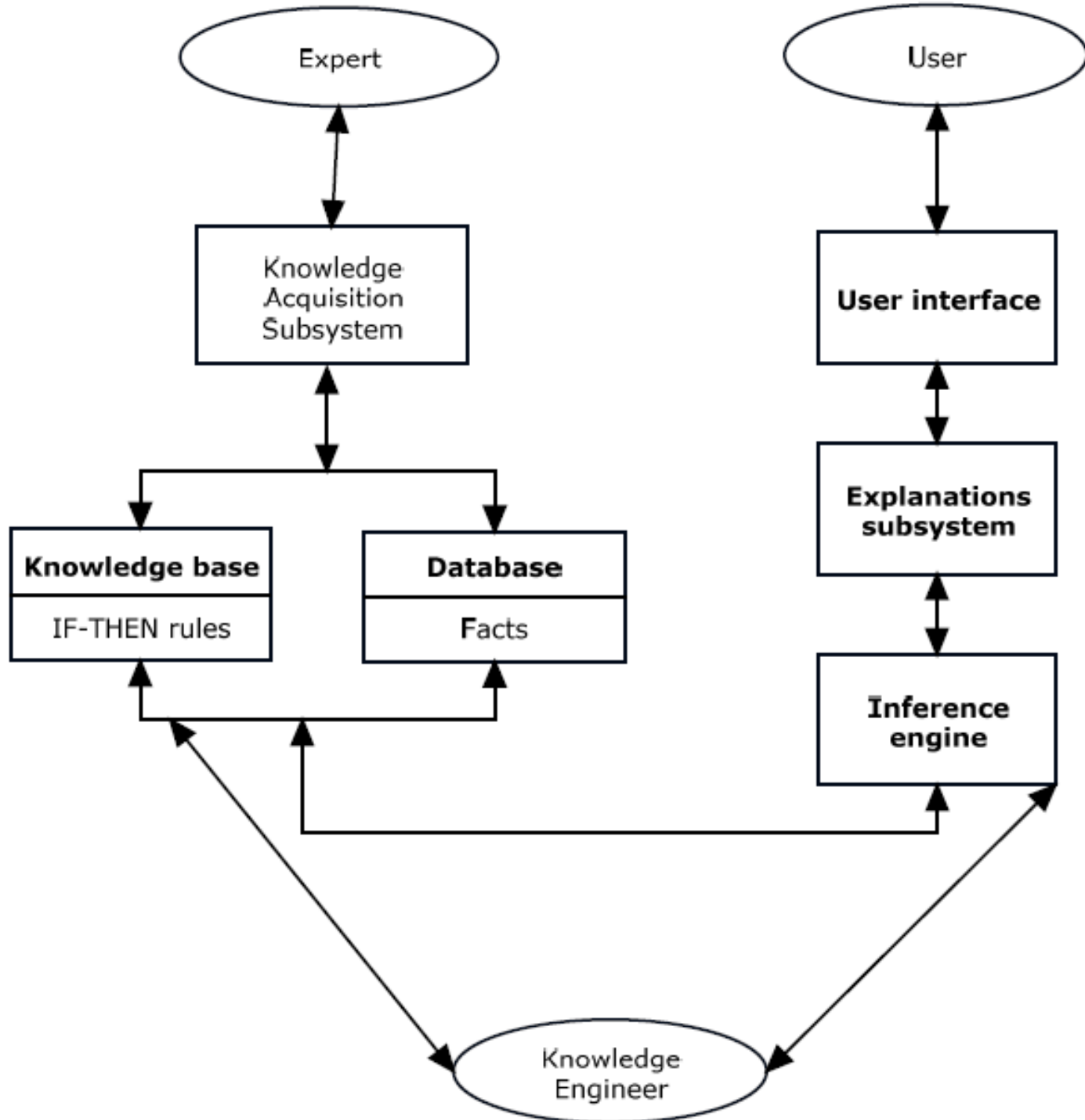
Ramalan Bersalju

Struktur Sistem Pakar Berbasis Aturan

- **Knowledge base**: mengandung pengetahuan domain, berbentuk aturan IF-THEN
- **Database**: fakta kalkulus predikat yang sesuai dengan bagian IF dari aturan dalam basis pengetahuan.
- **Inference Engine**: semua proses yang memanfaatkan basis pengetahuan untuk menyimpulkan apa yang diminta pengguna dan menyusun penalaran yang diperlukan sistem untuk mencapai solusi
- **Explanation subsystem**: menganalisis struktur penalaran yang dikerjakan oleh sistem dan menjelaskan itu kepada pengguna, memberikan pengguna peluang untuk menanyakan sistem mengenai cara kesimpulan dicapai atau mengenai fakta yang digunakan.

Struktur Sistem Pakar Berbasis Aturan

- ***User interface***: komunikasi antara pengguna yang mencari solusi dan sistem pakar. Terdiri dari beberapa jenis sistem pengolahan bahasa alami atau *graphical user interfaces* (GUI) bermenu.
- ***Knowledge engineer***: Biasanya ilmuwan komputer dengan pelatihan AI yang bekerja dengan pakar dalam bidang aplikasi dalam rangka menyediakan pengetahuan relevan dari pakar dalam bentuk yang dapat dimasukkan ke dalam basis pengetahuan.
- ***Knowledge acquisition subsystem***: Memeriksa dan mengupdate basis pengetahuan (yang terus tumbuh) , mungkin ada informasi yang tak-konsisten dan tak-lengkap.



Cara Kerja Penalaran

- Dimulai dengan suatu *rule-base* (basis aturan), mengandung semua pengetahuan yang di-encode ke dalam rule IF-THEN dan suatu *working memory*, dapat berisi data awal atau tidak, *assertions* atau informasi yang diketahui awalnya.
- Sistem memeriksa (menguji) semua kondisi rule (IF) dan menetapkan suatu subset, himpunan konflik, berisi rules yang kondisinya terpenuhi berdasarkan pada *working memory*.
- Dari himpunan konflik ini, salah satu rule dipicu (dijalankan). Mana yang terpilih didasarkan pada strategi resolusi konflik.
- Ketika rule dieksekusi, action yang ditentukan dalam klausa THEN-nya dihasilkan. *Action-action* ini dapat mengubah *working memory*, rule-base itu sendiri, atau hanya mengerjakan sesuatu yang ditetapkan/dimasukkan oleh *system programmer*.
- Loop eksekusi rules dan pengerjaan action ini berlanjut sampai ditemui kriteria berhenti. Karena sudah tidak ada lagi rules yang kondisinya terpenuhi atau dieksekusinya rule yang meminta program berhenti.

Penalaran (*Reasoning*)

- Adalah cara rules dikombinasikan untuk menurunkan pengetahuan baru.
- Adalah bagaimana manusia bekerja dengan pengetahuan, fakta dan strategi penyelesaian masalah untuk menarik kesimpulan.
- Beberapa jenis penalaran:
 - *inductive reasoning*;
 - *deductive reasoning*;
 - *abductive reasoning*;
 - *analogical reasoning*;
 - *common-sense reasoning*;
 - *non-monotonic reasoning*.

Rantai/*Chain* Inferensi

- *Inference engine* membandingkan setiap *rule* dalam basis pengetahuan dengan fakta dalam database.
- Jika bagian IF dari rule cocok dengan fakta maka bagian THEN dieksekusi dan rule berjalan (*fires*).
- Dengan menjalankan suatu rule, suatu hasil baru (fakta baru) diperoleh dan ini akan ditambahkan ke database.
- Dengan menjalankan *rule*, diperoleh rantai (*chain*) inferensi.
- Chain mengindikasikan bagaimana sistem pakar mengaplikasikan rules untuk meraih kesimpulan atau goal.

Jenis Sistem Pakar Berbasis Aturan

- Ada dua cara utama bagaimana rules dieksekusi:
 - ***forward chaining systems***. Dimulai dengan fakta awal dan tetap menggunakan rules untuk menarik kesimpulan baru (atau mengambil tindakan tertentu).
 - ***backward chaining systems***. Dimulai dengan beberapa hipotesis (atau *goal*) untuk membuktikan, dan terus mencari rules yang memungkinkan penyimpulan terhadap hipotesis tersebut, dengan mengatur sub-goal baru untuk dibuktikan.
- Sistem *forward chaining* utamanya bersifat *data-driven*, sedangkan *backward chaining* bersifat *goal-driven*.

Contoh 1

- Database suatu sistem pakar mengandung fakta A, B, C, D, E dan basis pengetahuan diberikan oleh aturan berikut:

**Rule 1: IF A is true
AND C is true
THEN B is true**

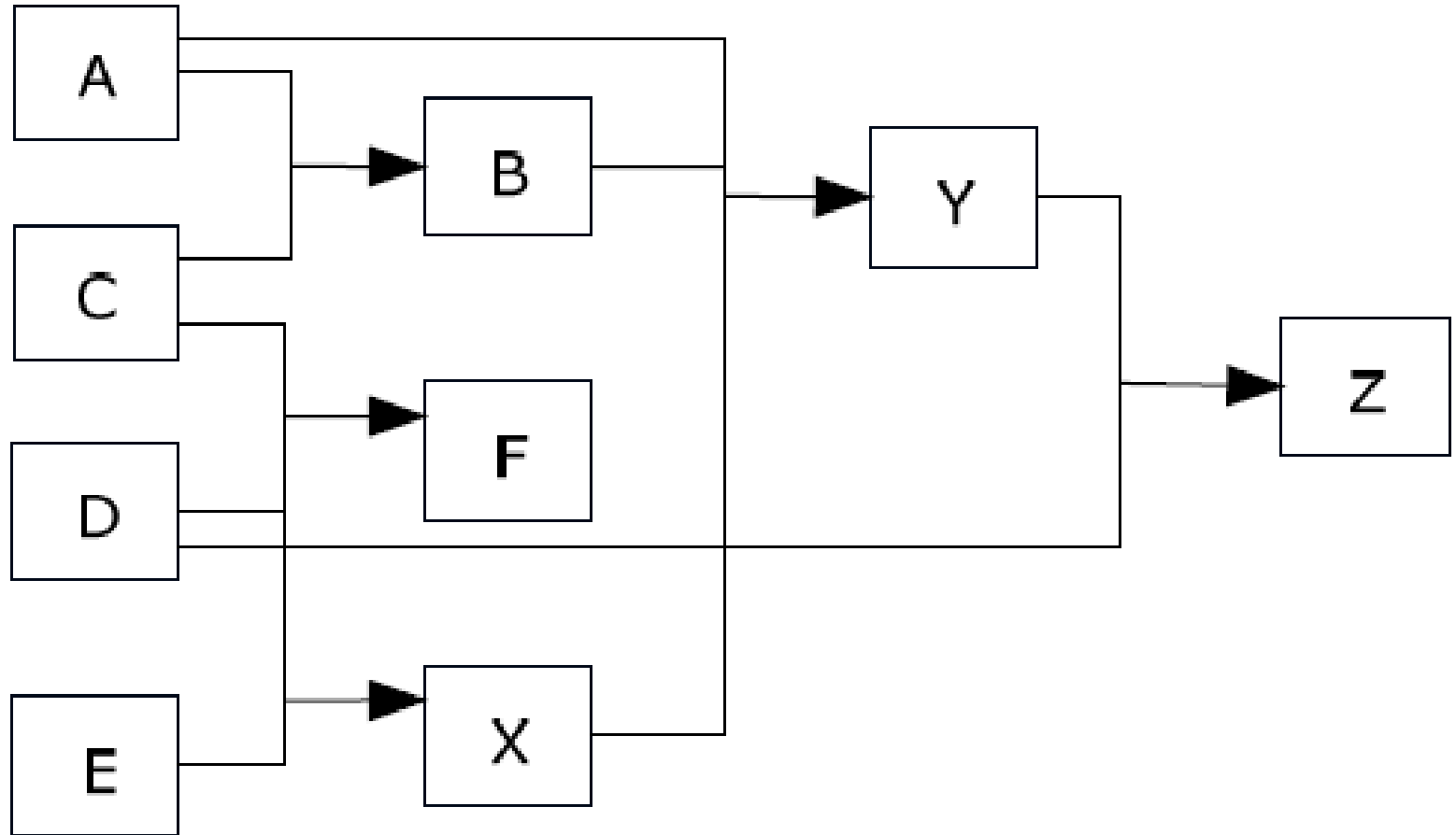
**Rule 2: IF C is true
AND D is true
THEN F is true**

**Rule 3: IF C is true
AND D is true
AND E is true
THEN X is true**

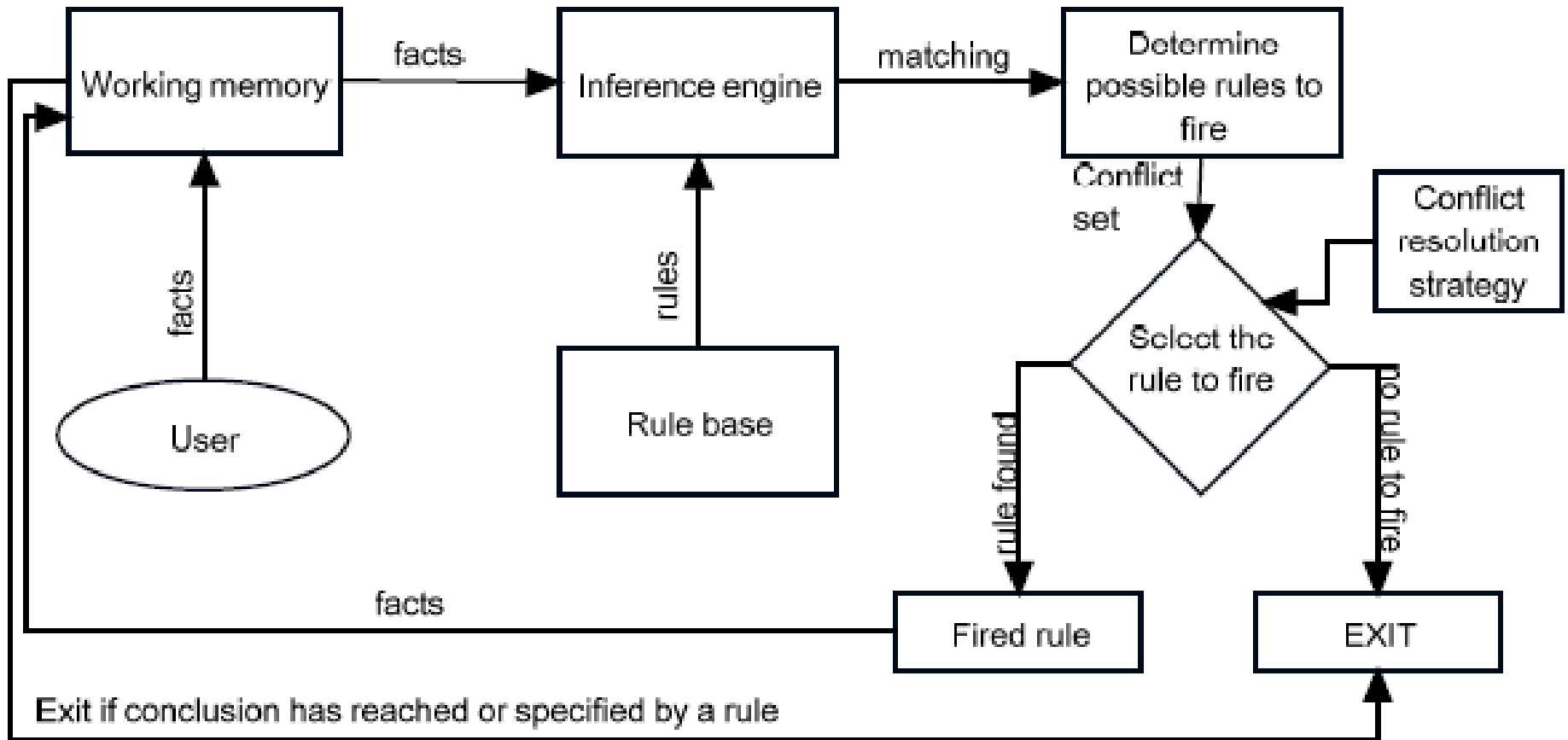
**Rule 4: IF A is true
AND B is true
AND X is true
THEN Y is true**

**Rule 5: IF D is true
AND Y is true
THEN Z is true**

Inference Engine: Contoh 1



System *Forward Chaining*



Sistem *Forward Chaining*

- Diberikan suatu himpunan fakta dalam *working memory*, gunakan rules untuk membangkitkan fakta baru sampai *goal* dicapai.
- Langkah-langkah:
 - 1) Cocokkan bagian IF dari setiap *rule* terhadap fakta-fakta dalam *working memory*.
 - 2) Jika ada lebih dari satu *rule* yang dapat digunakan (lebih dari satu rule yang berjalan), pilih satu yang akan diaplikasikan dengan menggunakan resolusi konflik.
 - 3) Berlakukan rule tersebut. Jika fakta baru diperoleh, tambahkan ke *working memory*.
 - 4) Stop (atau exit) ketika kesimpulan ditambahkan ke *working memory* atau jika ada rule yang menetapkan proses berhenti.

Forward Chaining: Contoh 1

- Perhatikan rantai inferensi sebelumnya. Bagaimana sistem menyimpulkan Z?

1. Cocokkan bagian IF dari setiap *rule* terhadap fakta dalam *working memory*.

Rules berikut dapat dipilih:

- **Rule 1: IF A AND C THEN B** (karena A dan C dalam database);
- **Rule 2: IF C AND D THEN F** (karena C dan D dalam database);
- **Rule 3: IF C AND D AND E THEN X** (C, D dan E dalam database)

Rules 4 dan 5 tidak dapat dipilih karena bagian IF-nya tidak dapat dicocokkan (X pada Rule 4 dan Y dalam Rule 5 tidak ada dalam database saat ini).

Forward Chaining: Contoh 1

- ***2. Jika ada lebih satu rule yang dapat digunakan, pilih satu yang ditetapkan.***

Ada 3 aturan yang dapat digunakan: Rule 1, Rule 2 dan Rule 3. Selalu pilih yang lebih awal dan belum diberlakukan.

Jadi, Rule 1 akan dijalankan pertama.

- ***3. Berlakukan rule tersebut. Jika fakta baru diperoleh, tambahkan itu ke working memory.***

Pada langkah ini, Rule 1 diberlakukan:

Rule 1: IF A AND C THEN B

- Konsekuensinya, B bernilai true. Namun B telah ada dalam database. Jadi tidak ada fakta baru yang diperoleh.

Forward Chaining: Contoh 1

- *4. Stop (atau exit) ketika kesimpulan ditambahkan ke working memory atau jika ada rule yang menetapkan berakhirnya proses tersebut.*
- Z (kesimpulan) belum diraih, sehingga ulangi lagi langkah pertama:
 - 1) Pada saat ini, rules yang dapat dipilih masih sama: Rule 1, Rule 2 dan Rule 3.
 - 2) Karena Rule 1 sudah dijalankan, rule terpilih berikutnya adalah Rule 2.
 - 3) Rule 2 dijalankan: Rule 2: IF C AND D THEN F.
- Fakta baru, F, diperoleh, belum ada dalam database. Tambahkan F ke dalam database. Isi database: **A, B, C, D, E, F.**

Forward Chaining: Contoh 1

4). Kesimpulan Z belum dicapai, proses dimulai lagi.

1). Rule yang dapat dijalankan masih sama: Rule 1, Rule 2, Rule 3.

2). Karena Rule 1 dan Rule 2 sudah dijalankan, sekarang pilih Rule 3:

**Rule 3: IF C is true AND D is true AND E is true
THEN X is true**

3). Rule 3 dieksekusi dan diperoleh fakta baru, X. Karena belum ada di database, X ditambahkan ke database.

4). X bukan kesimpulan, masih harus merestart proses.

Forward Chaining: Contoh 1

Siklus ke dua:

- 1). Database berisi fakta A, B, C, D, E, F, dan X. Jadi, empat rule pertama dapat dicocokkan.
- 2). Karena 3 rule pertama sudah digunakan, hanya Rule 4 yang dapat dipilih:
Rule 4: IF A AND B AND X THEN Y.
- 3). Rule 4 dijalankan dan diperoleh fakta baru, Y. Y ditambahkan ke database.
4. Y bukan kesimpulan, proses berulang kembali.

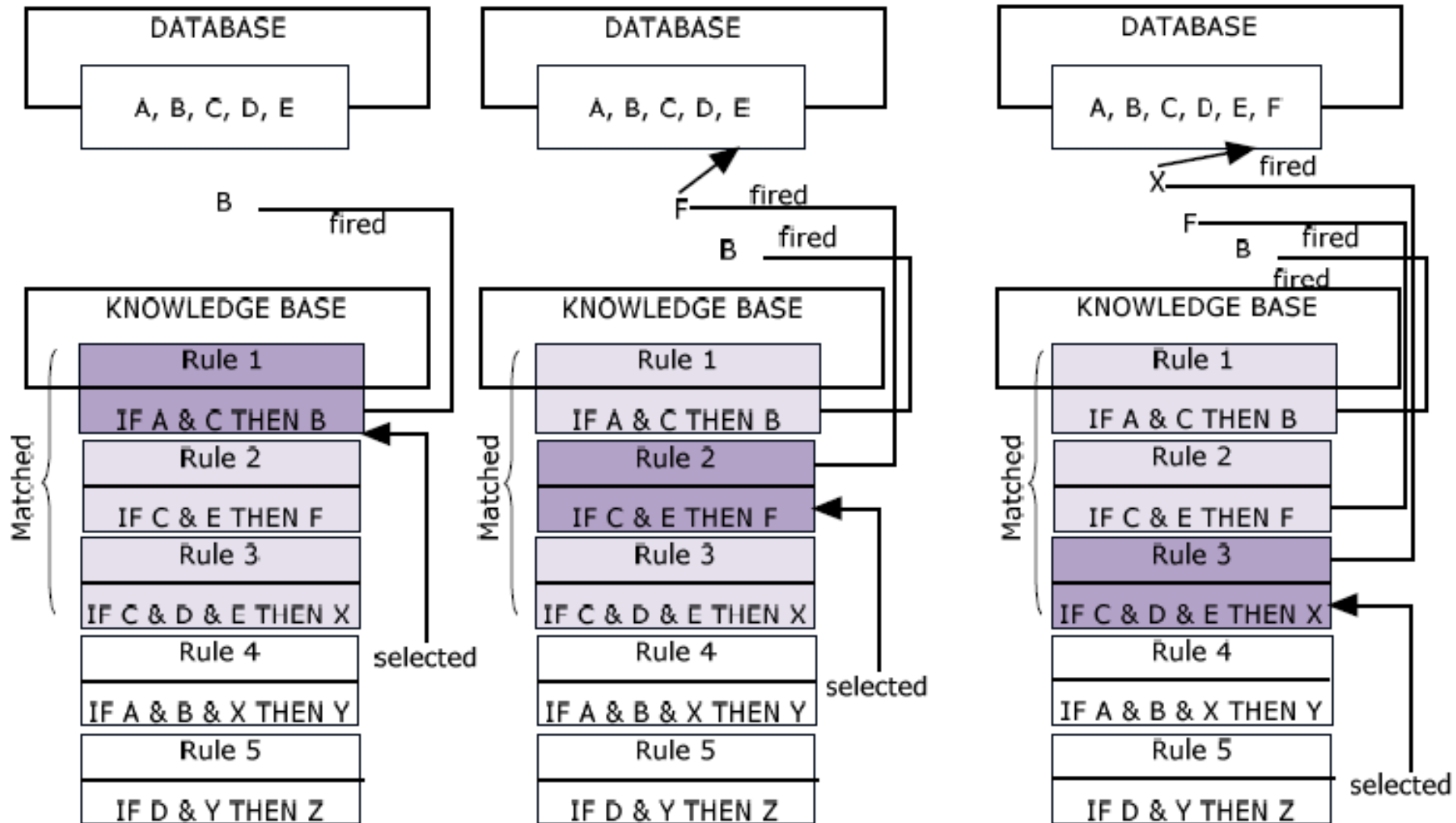
Forward Chaining: Contoh 1

Siklus ketiga:

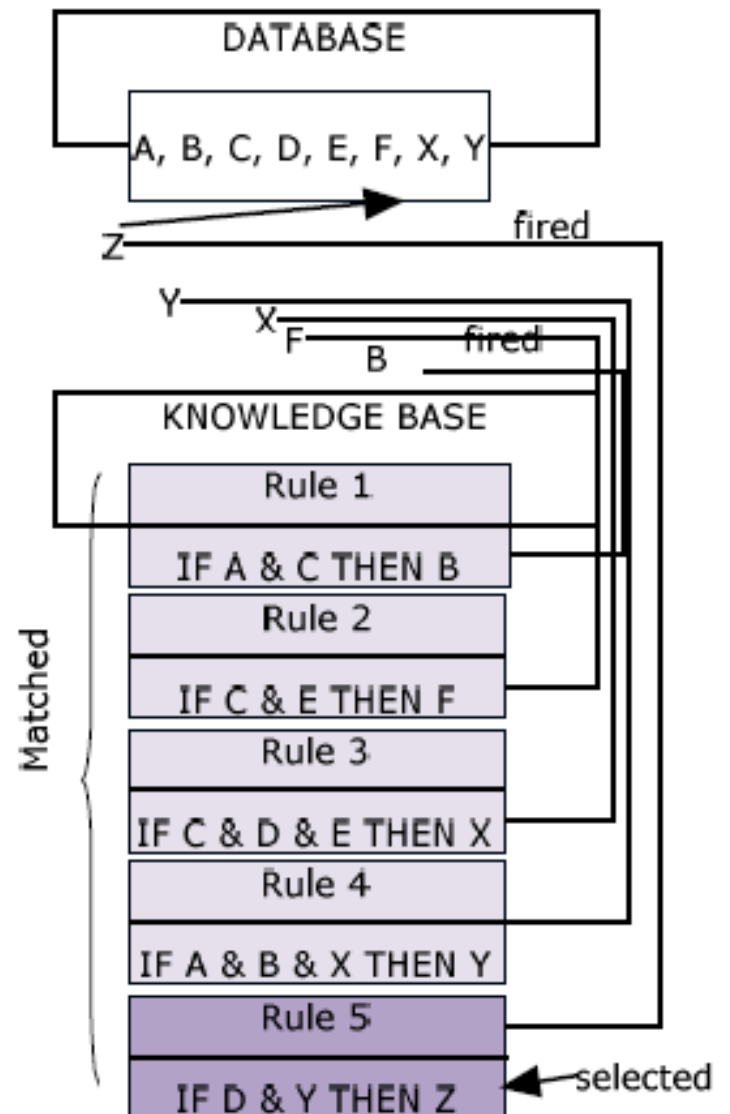
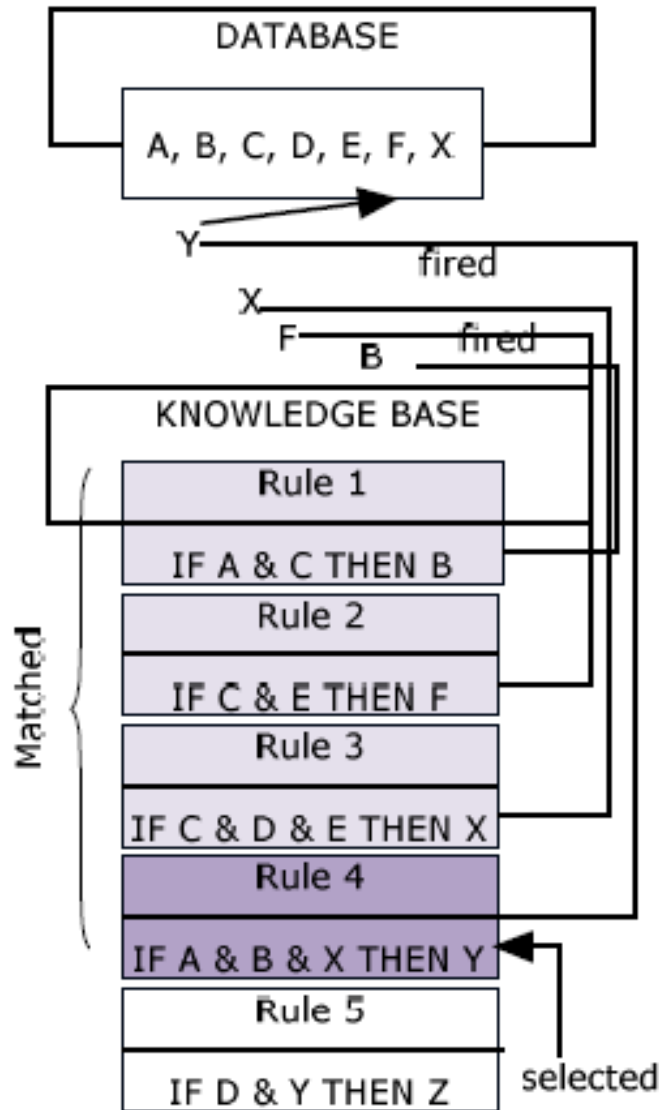
- 1). Semua rule (5 rule) cocok dengan kondisi IF.
- 2). Hanya tersisa Rule 5 yang belum dijalankan:
Rule 5: IF D AND Y THEN Z.
- 3). Rule 5 diberlakukan dan diperoleh fakta baru, Z. Z ditambahkan ke database.
- 4). Z mewakili kesimpulan sehingga proses berhenti sekarang.

Lihat diagram proses pada dua slide berikutnya.

Ilustrasi: Siklus 1



Ilustrasi: Siklus 2 dan 3



Contoh 2

- Diberikan himpunan fakta yang mengandung berbagai informasi mengenai bunga. Disediakan himpunan aturan. Tugasnya adalah menghasilkan solusi yang mengindikasikan **bunga white lily**. Tabel berikut memperlihatkan Variabel linguistik (obyek) dan nilai-nilai yang dibolehkan oleh sistem pakar dan terdapat di dalam database.
- Basis pengetahuan mengandung rule-rule diperlihatkan setelahnya.

Obyek dan Nilainya

Object	Value	Object	Value
Flower name	Iris Anemone Chrysanthemums Freesia Dahlia Narcissus Camellias Lily Begonia Azaleas Anemone Roses White lily	color	blue purple yellow red white pink orange violet pinkish-red

Obyek dan Nilainya

Season	Autumn Summer Spring winter	Size	10-50 cm 50-150 cm >150 cm
Root type	Bulb root	Perfume	True False
Life type	Perennial Annual	Soil	Acidic Loose Fertile Rich Well-drained
Life cycle	One year More than one year	Height	Small Medium Tall

Aturan dalam Basis Pengetahuan

Rule 1: IF size > 10

AND size <50

THEN height is small

Rule 2: IF size > 50

AND size <150

THEN height is medium

Rule 4: IF size > 150

THEN height is tall

Rule 5: IF life cycle is one year

THEN life type is annual

Rule 6: IF life cycle is more than one year

THEN life type is perennial

Rule 7: IF season is summer

AND color is blue

OR color is purple

OR color is yellow

AND life type is perennial

AND root type is bulb

THEN flower name is iris

Rule 8: IF season is autumn

AND color is white

OR color is pink

OR color is pinkish-red

THEN flower name is anemone

Aturan dalam Basis Pengetahuan

Rule 9: IF season is autumn

AND height is medium

AND color is yellow

OR color is white

OR color is purple

OR color is red

THEN flower name is

Chrysanthemum

Rule 10: IF season is spring

AND root type is bulbs

AND color is white

OR color is yellow

OR color is orange

OR color is purple

OR color is red

OR color is blue

AND perfumed is true

THEN flower is Freesia

Aturan dalam Basis Pengetahuan

Rule 11: IF life type is perennial
AND height is tall
AND root type is bulbs
AND season is summer
THEN flower name is Dahlia

Rule 12: IF season is spring
AND root type is bulbs
AND color is yellow
OR color is white
THEN flower name is Narcissus

Rule 13: IF soil is acidic
AND color is white
OR color is pink
OR color is red
AND life type is perennial
AND root type is roots
THEN flower name is Camellia

Rule 14: IF season is spring
AND root type is bulbs
AND perfumed is true
AND height is small
AND life type is perennial
THEN flower name is Lily

Aturan dalam Basis Pengetahuan

Rule 15: IF height is small
AND life type is annual
AND soil is rich
OR soil is loose
OR soil is fertile
THEN flower name is Begonia

Rule 16: IF season is winter
AND color is white
OR color is pink
OR color is red
THEN flower name is Azalea

Rule 17: IF life type is perennial
AND root type is root
AND color is white
OR color is red
OR color is blue
OR color is yellow
THEN flower is Anemone

Rule 18: IF life type is perennial
AND root type is roots
AND color is white
OR color is pink
OR color is red
OR color is yellow
AND perfumed is true
AND soil is well-drained
THEN flower is rose

Aturan dalam Basis Pengetahuan

Rule 19: IF flower name is Lily
AND perfumed is true
THEN flower name is White lily

- Misalnya fakta yang ada dalam database adalah:
**season: spring, root type: bulbs, perfumed: true,
size: 16-18 cm, life cycle more than one year, color:
orange, red, white, pink.**
- Bagaimana proses penalaran menemukan **bunga
white lily?**

Langkah Solusi: Siklus 1

- **Pencocokan:** Rule 1 dan Rule 6 bersifat *applicable*.
- **Pemilihan Rule:** Pilih Rule 1
- **Penerapan Rule:** *height is small* akan ditambahkan ke *working memory*.
- **Pencocokan:** Rule 1 dan Rule 6 *applicable*.
- **Pemilihan Rule:** Hanya dapat memilih Rule 6 karena Rule 1 sudah dijalankan.
- **Penerapan Rule:** *Life type perennial* ditambahkan ke *working memory*.
- Sekarang, memory berisi:
season: spring, **root type:** bulbs, **perfumed:** true, **size:** 16-18 cm, **life cycle** more than one year, **color:** orange, red, white, pink, **height:** small, **life type:** perennial.

Langkah Solusi: Siklus 2

- **Pencocokan** : Rule 1, Rule 6, Rule 10, Rule 12 dan Rule 14 *applicable*.
- **Pemilihan Rule** : Hanya dapat memilih Rule 10, Rule 12 atau Rule 14 karena Rule 1 dan Rule 6 sudah dijalankan. Pilih Rule 10.
- **Penerapan Rule** : **Flower name is Freesia** akan ditambahkan ke *working memory*.
- **Pencocokan** : Rule 1, Rule 6, Rule 10, Rule 12 dan Rule 14 *applicable*.
- **Pemilihan Rule** : Hanya dapat memilih Rule 12 atau Rule 14 karena Rule 1, Rule 6 dan Rule 10 sudah dijalankan. Pilih Rule 12.
- **Penerapan Rule** : **Flower name is Narcissus** akan ditambahkan ke *working memory*.

Langkah Solusi: Siklus 2

- **Pencocokan**: Rule 1, Rule 6, Rule 10, Rule 12 dan Rule 14 *applicable*.
- **Pemilihan Rule** : Hanya dapat memilih Rule 14 karena *rules* lainnya sudah dijalankan.
Pilih Rule 14.
- **Penerapan Rule** : **Flower name is Lily** akan ditambahkan ke *working memory*.
- Working memory sekarang mengandung: **season**: spring, root **type**: bulbs, **perfumed**: true, **size**: 16-18 cm, **life cycle** more than one year, **color**: orange, red, white, pink, **height**: small, **life type**: perennial, flower **name**: freesia, narcissus, lily.

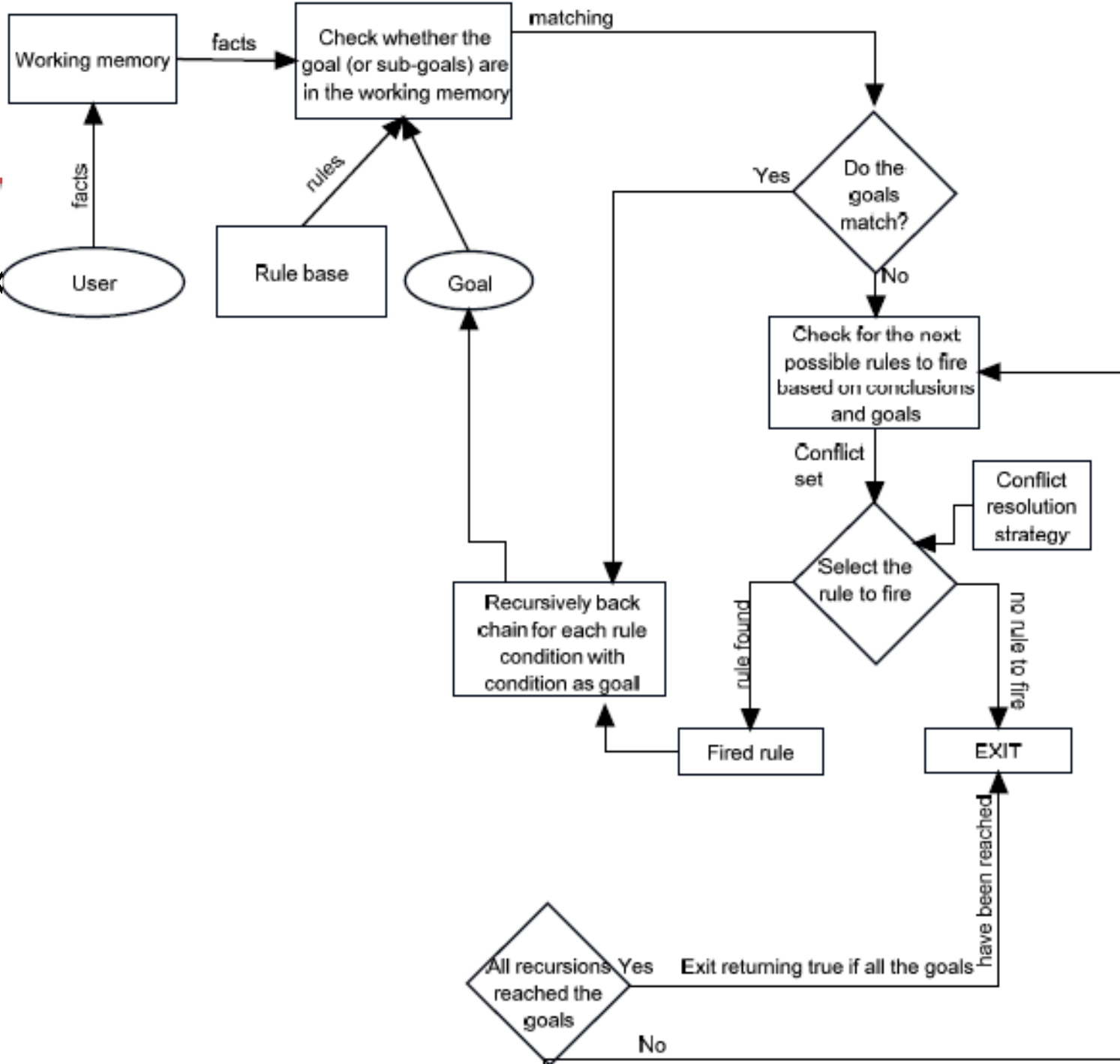
Langkah Solusi: Siklus 3

- **Pencocokan:** Rule 1, Rule 6, Rule 10, Rule 12, Rule 14 dan Rule 19 bersifat *applicable*.
- **Pemilihan Rule:** Hanya dapat memilih Rule 19 karena lainnya sudah dijalankan.
Pilih Rule 19.
- **Penerapan Rule:** Flower name is White lily ditambahkan ke *working memory* dan juga merupakan *goal*.
Proses inferensi berhenti.

Sistem *Backward Chaining*

- Mesin inferensi menjelajah secara mundur (*backward*) rantai inferensi (*chain*) dimulai dari tujuan (*goal*) dalam *working memory*.
- Terdiri dari 3 langkah utama:
 1. Pilih *rules* yang konklusi-nya sesuai dengan goal.
 2. Ganti goal dengan premis dari *rule* terpilih. Jadikan sebagai *sub-goals*.
 3. Kerjakan *backwards* sampai semua *sub-goals* bernilai true. Ini dicapai dengan:
 - Ditemukannya fakta (dalam *working memory*) atau
 - Pengguna menyediakan informasi tersebut.

- X



Backward Chaining: Contoh 1

- Seperti contoh 1 sebelumnya...
- **Rule 1:** IF A is true
AND C is true
THEN B is true
- **Rule 2:** IF C is true
AND D is true
THEN F is true
- **Rule 3:** IF C is true
AND D is true
AND E is true
THEN X is true
- **Rule 4:** IF A is true
AND B is true
AND X is true
THEN Y is true
- **Rule 5:** IF D is true
AND Y is true
THEN Z is true

Goal dari sistem adalah Z

Langkah 1

1). *Pilih rules dengan kesimpulan yang cocok dengan tujuan (goal).*

Hanya kesimpulan dari Rule 5 yang sesuai dengan goal (Z).

2). *Ganti goal dengan premis dari rule. Ini dinamakan sebagai sub-goal.*

D ada dalam database tetapi tidak dengan Y. Sehingga jadikan Y sebagai sub-goal pertama.

3). *Kerjakan secara backward (mundur) sampai semua sub-goals yang diketahui bernilai true.*

- *Belum semua sub-goals sebagai true. Jadi ulangi lagi.*

Langkah 2

1). *Pilih rules yang konklusinya sesuai dengan tujuan.*

Goal adalah Z tetapi sub-goal adalah Y. Rule 4 mempunyai Y sebagai kesimpulan.

2). *Ganti goal dengan premis rulanya. Jadikan sebagai sub-goals.*

Diantara premis pada Rule 4, A dan B terdapat dalam database tetapi X tidak. Jadi, sub-goal adalah X.

3). *Lakukan backwards sampai semua sub-goals diketahui bernilai true.*

Tidak semua sub-goal bernilai true, *restart*.

Langkah 3

1). Pilih rules dengan kesimpulan sesuai goal.

Goal adalah Z. Sub-goalnya adalah Y dan X. X adalah sub-goal terbaru. Rule 3 mempunyai X sebagai kesimpulan.

2). Ganti goal dengan premis dari rule terpilih. Jadikan sebagai sub-goals.

Premis Rule 3 adalah C, D dan E dan semuanya ada dalam database.

Jadi, sub-goal X dapat diperoleh dengan menjalankan Rule 3 terlebih dahulu.

3). Lakukan backwards sampai semua sub-goals diketahui bernilai true.

Lanjutan...

- Telah diperoleh salah satu sub-goals. Belum semua sub-goal bernilai True. Jadi, balik untuk mendapatkan sub-goal lain.
- Pertama adalah Y.
- Kembali ke Langkah 2, diketahui (sekarang) semua premis pada Rule 4. Jadi, Rule 4 dapat dijalankan dan diperoleh Y dan ditambahkan ke database.
- Kembali ke Langkah 1, semua premis dari Rule 5 diketahui dan kesimpulannya adalah goal. Rule 5 dapat dijalankan dan diperoleh goal Z.
- **Diagram pada 2 slide berikut menjelaskan...**

Diagram *Backward Chaining*

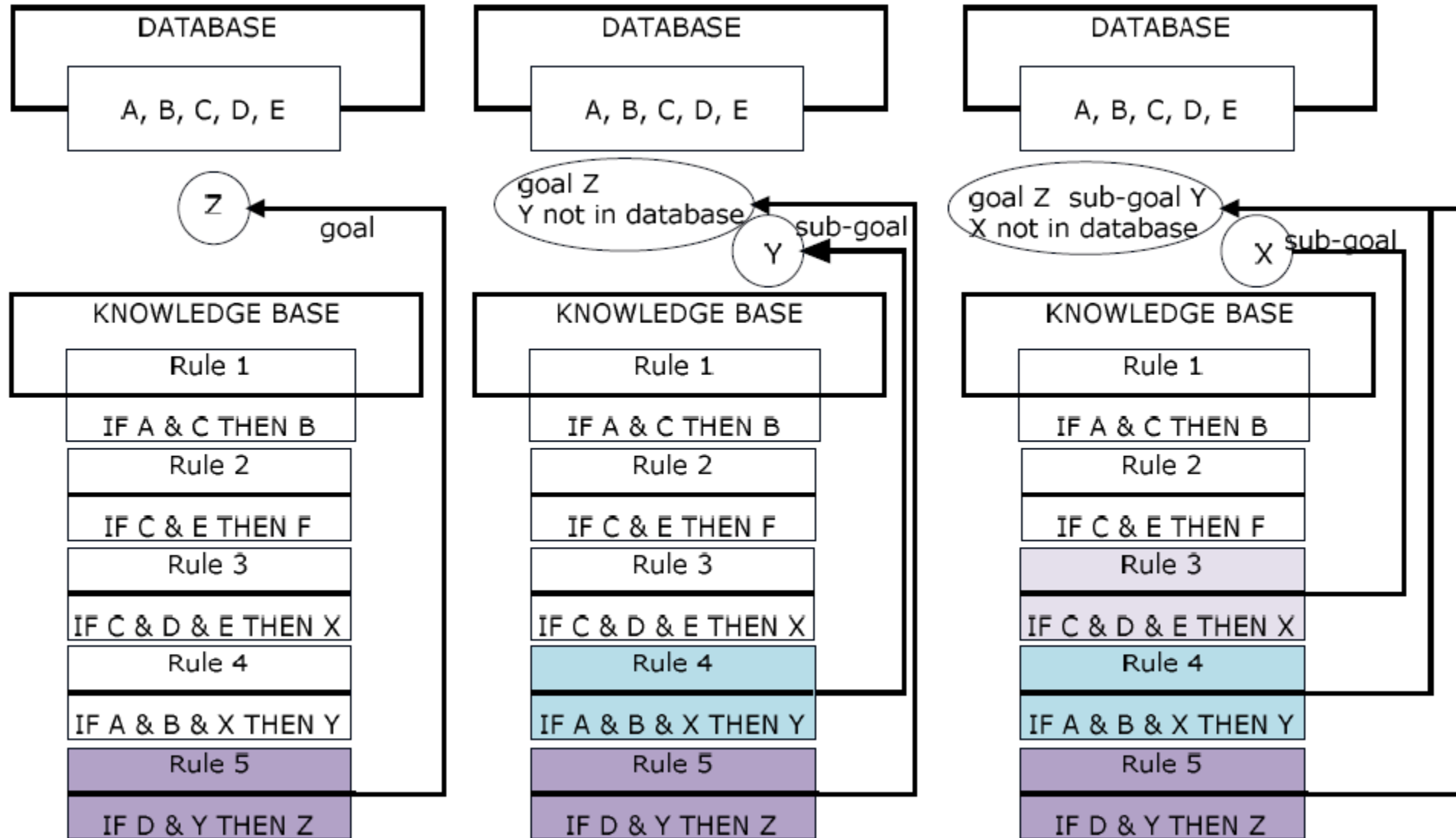
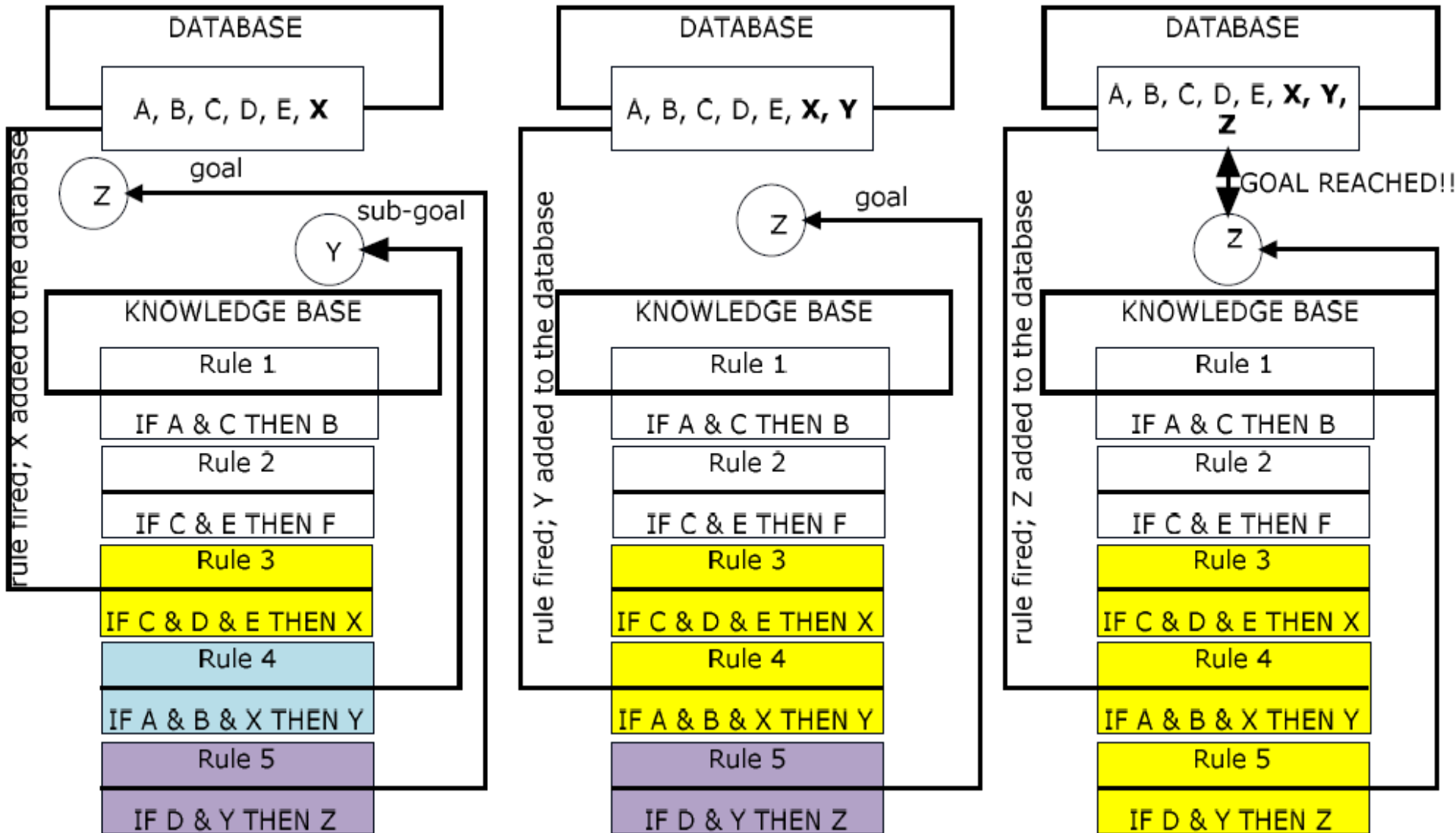


Diagram *Backward Chaining*



Backward Chaining: Contoh 2

Langkah 1

1). *Pilih rules yang konklusinya cocok dengan goal.*

Konklusi yang sesuai dengan *goal* hanya pada Rule 19.

2). *Ganti goal dengan premis dari rule tersebut. Jadikan sebagai sub-goals.*

Premis “perfumed is true” ada dalam database tetapi premis “flower name is Lily” tidak. Kesimpulan “flower name is Lily” adalah sub-goal dan ini adalah consequent dari Rule 14.

3). *Kerjakan backwards sampai semua sub-goals bernilai true.*

Belum semua sub-goals bernilai True, *back-chain* lagi.

Langkah 2

1). *Pilih rules yang konklusinya cocok dengan goal.*

Goal “**flower name is white lily**” (konklusi Rule 19) dan sub-goal “**flower name is lily**” konklusi Rule 14.

2). *Ganti goal dengan premis dari rule tersebut. Jadikan sebagai sub-goals.*

Di antara premis Rule 14, sebagian sudah ada dalam database: “**season is spring**”, “**root type is bulbs**”, “**perfumed is true**”.

Diperlukan dua premis lagi: “**height is small**” dan “**life type is perennial**”.

Sekarang “**height is small**” dan “**life type is perennial**” menjadi sub-goals.

3). *Kerjakan backwards sampai semua sub-goals bernilai true.*

Belum semua sub-goals bernilai true, *back-chain* lagi.

Langkah 3

1). *Pilih rules yang konklusinya cocok dengan goal.*

Goal “**flower name is white lily**” (konklusi Rule 19) dan sub-goals “**flower name is lily**” konklusi Rule 14, “**height is small**”, konklusi Rule 1 dan “**life type is perennial**” konklusi Rule 6.

2). *Ganti goal dengan premis dari rule tersebut. Jadikan sebagai sub-goals.*

Semua premis dari Rule 1 sudah ada dalam database. Jadi, Rule 1 dapat dijalankan dan fakta “**height is small**” ditambahkan ke database.

3). *Kerjakan backwards sampai semua sub-goals bernilai true.*

Belum semua sub-goals bernilai true, *back-chain* lagi .

Langkah 4

1). *Pilih rules yang konklusinya sesuai dengan goal.*

Goal “**flower name is white lily**”, kesimpulan dari Rule 19 dan sub-goals adalah “**flower name is lily**”, kesimpulan dari Rule 14 dan “**life type is perennial**” , kesimpulan dari Rule 6.

2). *Ganti goal dengan premis dari rule. Jadikan sebagai sub-goals.*

Premis dari Rule 1 ada dalam database. Jadi, Rule 6 dapat dijalankan dan fakta “**life type is perennial**” ditambahkan ke database.

3). *Kerjakan secara backwards sampai semua sub-goals bernilai true.*

Belum semua sub-goal bernilai True, kembali *back-chain* lagi.

Langkah 5

1). *Pilih rules yang konklusinya sesuai dengan goal.*

Goal “**flower name is white lily**”, kesimpulan dari Rule 19 dan sub-goalnya “**flower name is lily**”, kesimpulan dari Rule 14.

2). *Ganti goal dengan premis dari rule terpilih. Jadikan sebagai sub-goals.*

Semua premis untuk menjalankan Rule 14 telah dimiliki. Fakta “**flower name is lily**” diperoleh dan dimasukkan ke database.

3). *Kerjakan backwards sampai semua sub-goals bernilai true.*

Belum diperoleh goal, jadi lanjutkan ke *back-chain*.

Langkah 6

1). *Pilih rules yang konklusinya sesuai dengan goal.*

- Goal adalah “**flower name is white lily**”, kesimpulan dari Rule 19. Tidak ada *sub-goals* pada tahapan ini dan semua premis dari Rule 19 bernilai true. Jadi, Rule 19 dapat dijalankan dan tujuan akhir diperoleh.

Resolusi Konflik

- Mendefinisikan urutan rules dijalankan selama proses inferensi.
 - **First applicable**: Jika rules terurut, dahulukan menjalankan rule yang paling dapat diterapkan.
 - **Random**: pilih salah rule yang akan dijalankan secara acak.
 - **Most Specific**: Didasarkan pada sejumlah kondisi dari rule. Rule dengan kondisi paling banyak dan spesifik didahulukan.
 - **Least Recently Used**: setiap rule mempunyai timestamp (jam dan tanggal) menandakan waktu terakhir digunakan.
 - **"Best" Rule**: Setiap rule diberikan bobot yang menentukan seberapa rule itu dipertimbangkan.

Resolusi Konflik: *First Applicable*

- Contoh:

Rule 1: IF color is yellow
THEN fruit is apple;

Rule 2: IF color is yellow
AND shape is long
THEN fruit is banana

Rule 3: IF shape is round
THEN fruit is apple.

- Database mengandung **yellow (color)** dan **round (shape)**.
Rule 1 dan Rule 3 dapat dijalankan. Rule 1 lebih dahulu
kemudian Rule 3.

Resolusi Konflik: *Most Specific*

- Contoh:

Rule 1: IF the weather is cold
THEN the season is winter

Rule 2: IF the weather is cold
AND the temperature is low
AND the wind is blushing
AND the forecast is snow
THEN the season is winter.

- Rule 2 adalah paling signifikan dibandingkan Rule 1. Jadi, Rule 2 dipilih jika digunakan strategi *paling spesifik*.

Resolusi Konflik: *Least Recently Used*

- Contoh:

Rule 1: IF color is yellow [28.02.2012, 13:45]
THEN fruit is apple;

Rule 2: IF color is yellow [01.03.2012, 12:00]
AND shape is long
THEN fruit is banana

Rule 3: IF shape is round [05.03.2012, 20:00]
THEN fruit is apple.

- Rule 1 dipilih untuk dijalankan karena **the least recently introduces**.

Resolusi Konflik: *Best Rule*

- Contoh:

Rule 1: IF color is yellow 30%
THEN fruit is apple;

Rule 2: IF color is yellow 30%
AND shape is long
THEN fruit is banana

Rule 3: IF shape is round 40%
THEN fruit is apple.

- Rule 3 akan dipilih karena mempunyai bobot paling tinggi diantara semua rule.

Tugas

- Sebutkan kelebihan dan kekurangan dari sistem pakar berbasis aturan!
- Sistem pakar, berdasarkan berbagai aspek, dapat dibagi ke dalam beberapa kategori. Sebutkan!
- Sebutkan 5 contoh aplikasi sistem pakar yang dikenal dan banyak digunakan!
- Kerjakan soal No. 7.1 dan 7.2 dari buku text *“Intelligent System: A Modern Approach”*