ASPEK EKSEKUSI DALAM BAHASA PROLOG

Pada bagian ini akan dibahas hal-hal berikut :

1. Penalaran Deduktif
2. Inferensi Resolusi
3. Mesin Inferensi PROLOG
4. Pohon Bukti
5. Pohon Eksekusi
6. Backtrack

PENALARAN DEDUKTIF

Pada Kalkulus Predikat dikenal adanya penalaran deduktir *(Deductive Reasoning)* yang digunakan untuk menghasilkan kalimat baru hasil penurunan *(Derivation)* secara deduktif dari kalimat yang sudah ada.

Penalaran deduktif dapat menghasilkan kalimat baru dengan menerapkan aturan inferensi (pengambilan kesimpulan). Beberapa aturan inferensi yang sangat penting dan akan digunakan pada bahasan tentang PROLOG adalah :

1. Substitution (substitusi)
2. Modusponens : dari kalimat A dan kalimat (if A then B) dapat dihasilkan kalimat baru B.
3. Modustollens : dari kalimat (not A) dan (if B then A) dapat dihasilkan kalimat baru (not B).
4. Resolution (resolusi) : dari kalimat A dan kalimat ((not A) or B) dapat dihasilkan B.

Aturan inferensi modustollens dan resolusi merupakan pengembangan dari modusponens untuk bentuk kalimat yang berbeda.

Penalaran deduktif terdiri dari jenis :

1. Bottom-Up Reasoning : bekerja maju dari asumsi ke konklusi (kesimpulan).
2. Top-Down Reasoning : bekerja mundur dari goal (tujuan) ke subgoal.

Berdasarkan hal tersebut diatas maka juga dikenal dua jenis aturan inferensi :

1. Bottom-Up Inference : melakukan sintesis informasi baru berdasarkan informasi lama.
2. Top-Down Inference : melakukan analisis goal ke dalam subgoal.

Contoh : menggunakan modusponens dan modustollens

Bottom-Up	**Top-Down**
Asersi : if A1 then B1 Asumsi : A1 Kesimpulan : B1	Klausa : if A1 then B1 Goal : (not B1) Subgoal : (not A1)

INFERENSI RESOLUSI

Karena PROLOG menyatakan kalimat-kalimatnya (fakta, aturan, query) dalam bentuk klausa Horn (Horn Clause), maka PROLOG menggunakan aturan inferensi resolusi. Aturan inferensi resolusi menghasilkan bukti dengan cara **refutasi** *(Refutation)*, yaitu : untuk membuktikan sebuah kalimat, resolusi berusaha menunjukkan bahwa negasi dari kalimat tersebut berkontradiksi dengan kalimat-kalimat yang sudah ada sebelumnya.

Bottom-Up Refutation melakukan sintesis informasi (dalam bentuk asersi) baru berdasarkan informasi (asersi) lama. Sehingga, bermula dari asersi sebagai himpunan klausa awal, akan berakhir dengan penurunan asersi yang secara eksplisit berkontradiksi dengan konklusi/kesimpulan.

Top-Down Refutation melakukan analisis goal (dalam bentuk negasi kalimat atau denials) ke dalam subgoal (denials). Sehingga, bermula dari denials sebagai himpunan klausa awal, akan berakhir dengan penurunan denials berupa klausa kosong (nil).

Contoh : sama dengan sebelumnya, tetapi menggunakan resolusi

Bottom-Up	**Top-Down**
Asersi : (not A1) or B1 Asumsi : A1 Kesimpulan : B1	Klausa : (not A1) or B1 Goal : (not B1) Subgoal : (not A1)

MESIN INFERENSI PROLOG

Mesin PROLOG bekerja berdasarkan Top-Down Refutation, yang melakukan komputasi yes/no, yang memerlukan masukan program P dan goal berupa query Q untuk menghasilkan keluaran :

- yes, jika Q dapat dideduksi dari P
- no, jika Q tidak dapat dideduksi dari P

Mesin PROLOG juga mungkin tidak berhasil mencapai terminasi jika Q tidak dapat dideduksi dari P, sehingga tidak menghasilkan keluaran apapun. Jika Q tidak dapat dideduksi dari P, maka dapat berarti dua hal:

1. Q bukan konsekuensi logik dari P, atau
2. fakta tidak cukup untuk membuktikan bahwa Q dapat dideduksi dari P.

Pada setiap tahap komputasi akan dihasilkan subgoal baru *(Current Goal)* yang disebut resolvent. *Trace* dari kerja mesin berupa deretan resolvent yang dihasilkan selama komputasi berlangsung. Trace itu secara implisit berisi bukti dari query Q terhadap program P, yang dapat direpresentasikan dalam bentuk pohon (selanjutnya disebut **Pohon Bukti**).

Contoh 1.3 : Pohon bukti berdasarkan program Contoh 1.2.

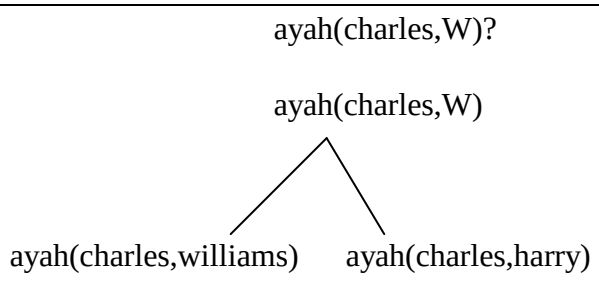
Query (beserta jawabannya)

```
Goal: ayah(charles,W)
yes
W=williams
W=harry
```

Hasil Trace

```
CALL:   ayah(charles,_)
RETURN: ayah(charles,williams)
REDO:   ayah(charles,williams)
RETURN: ayah(charles,harry)
```

Pohon Bukti



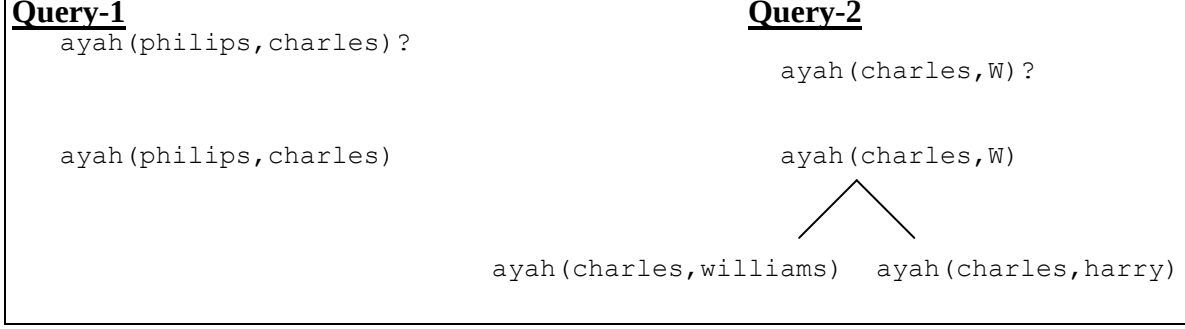
POHON BUKTI

Pohon bukti terdiri dari simpul dan busur yang menyatakan goal yang direduksi menjadi subgoal-subgoal selama komputasi berlangsung. Akar dari pohon bukti adalah goal berupa query Q, sedangkan simpulnya adalah goal-goal yang direduksi selama komputasi berlangsung, serta busurnya menyatakan penurunan subgoal dari sebuah goal. Jika sebuah goal menghasilkan lebih dari satu subgoal, maka busurnya berupa busur **or**. Tetapi jika goal berupa conjunctive goal, maka busurnya berupa busur **and**.

Pohon bukti dari conjunctive query adalah kumpulan pohon bukti dari individual goal (dalam konjungsi). Untuk kegunaan dan kemudahan penggambarannya, umumnya pohon bukti hanya berisi resolvent yang menyebabkan tujuan komputasi (penurunan klausa kosong) tercapai (jika mungkin).

Contoh 1.3 : Pohon bukti berdasarkan program Contoh 1.1.

Pohon Bukti



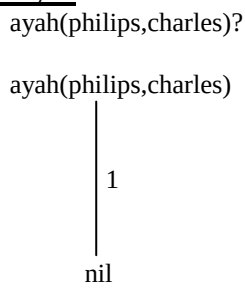
POHON EKSEKUSI

Pohon eksekusi adalah pengembangan dari pohon bukti untuk kepentingan yang berbeda. Akar dari pohon eksekusi menyatakan goal berupa query Q. Simpul pada pohon eksekusi menyatakan klausa yang cocok dengan goal (denials), dan resolvent. yang dihasilkan (denials baru) beserta substitusi yang perlu dilakukan. Jika goal (denials) berupa fakta, maka resolvent nil. Tetapi jika goal (denials) berupa aturan, maka resolvent berupa bagian kondisi dari aturan. Busur or digunakan untuk menyatakan penurunan resolvent dari denials, sedangkan busur and digunakan untuk menyatakan conjunctive denials.

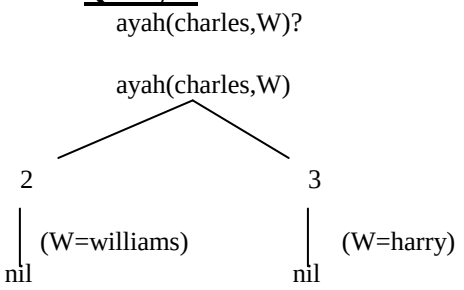
Contoh : Pohon eksekusi berdasarkan program Contoh 1.1, yang faktanya diberi nomor dari 1 sampai dengan 7 dan aturan bernomor 8.

Pohon Eksekusi

Query-1



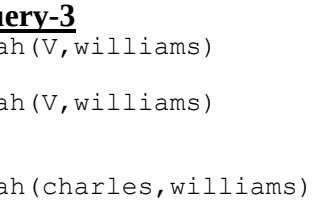
Query-2



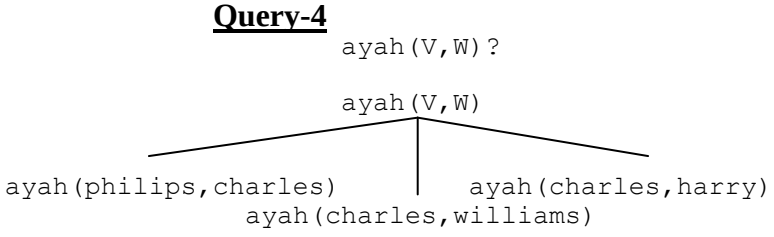
Beberapa contoh hasil eksekusi query lain, masih berdasarkan program Contoh 1.1, memperlihatkan kedua pohon - baik pohon bukti maupun pohon eksekusinya.

Pohon Bukti

Query-3

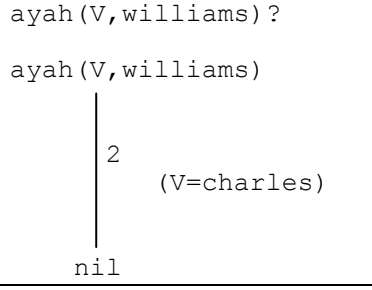


Query-4

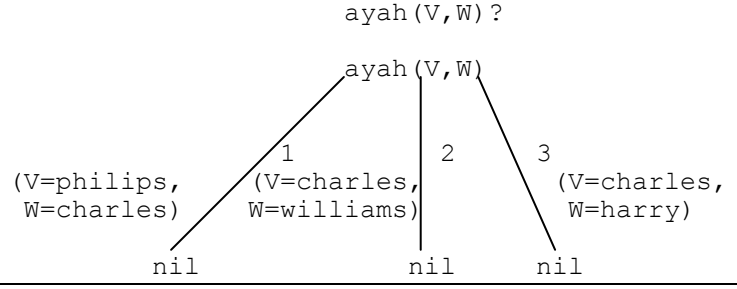


Pohon Eksekusi

Query-3

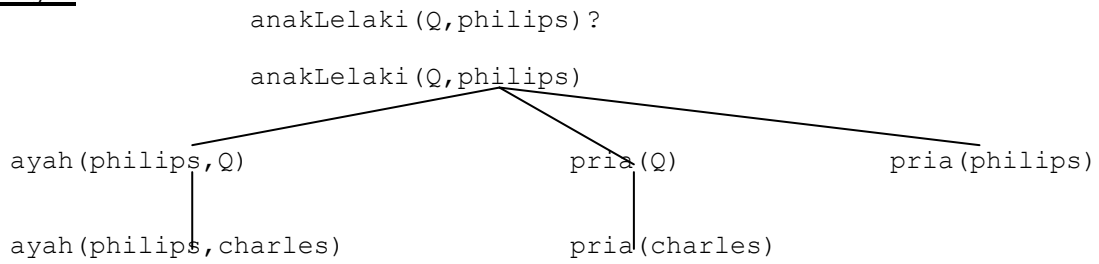


Query-4



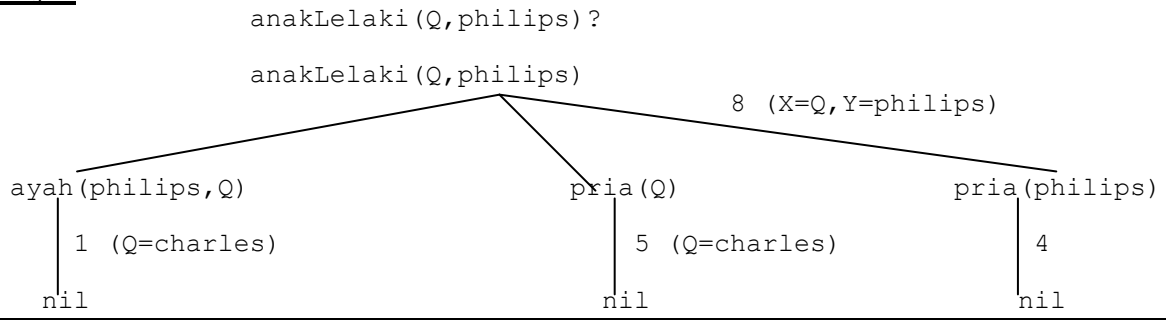
Pohon Bukti

Query-5



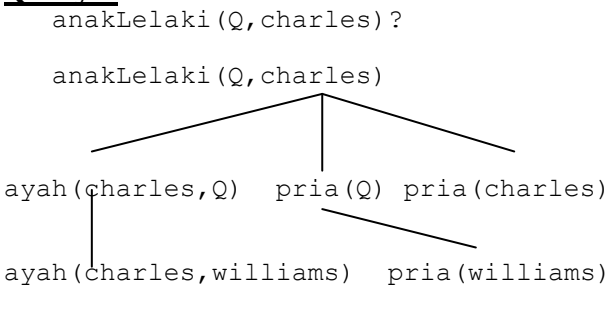
Pohon Eksekusi

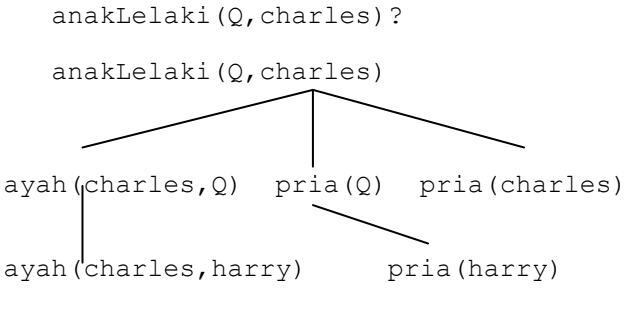
Query-5



Pohon Bukti

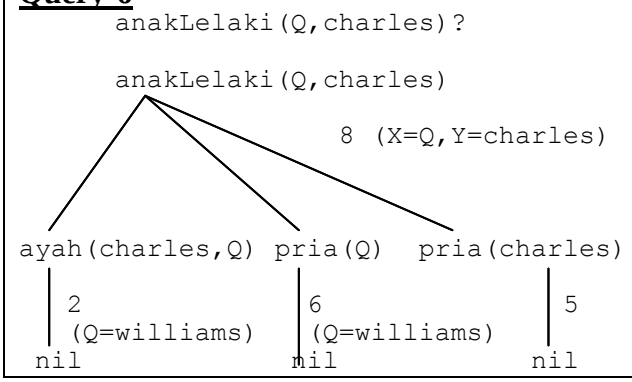
Query-6

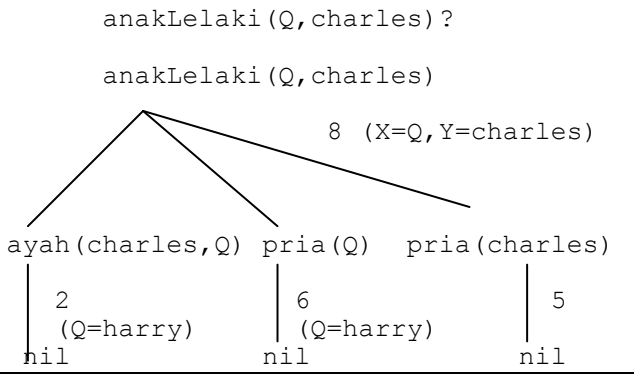




Pohon Eksekusi

Query-6





BACKTRACK

Dalam melakukan inferensi (yang dapat digambarkan dalam bentuk pohon eksekusi) digunakan strategi pencarian jenis DFS atau Depth First Search. Jika pada proses pencapaian klausa kosong (nil) digunakan path yang salah sehingga tujuan tidak tercapai, maka pencarian simpul berikutnya (untuk mendapatkan path yang benar) dilakukan dengan cara bactracking.

Contoh : Pohon eksekusi beserta backtrack yang dilakukan berdasarkan program Contoh 1.1, yang faktanya diberi nomor dari 1 sampai dengan 7 dan aturan bernomor 8.

Pohon Eksekusi

Query