

## ANALISA EFEKTIFITAS METODE FORWARD CHAINING DAN BACKWARD CHAINING PADA SISTEM PAKAR

Ibnu Akil

Program Studi Manajemen Administrasi  
ASM BSI Jakarta

Jl. Jatiwaringin Raya No.18, Jakarta Timur  
Ibnu.ial@bsi.ac.id

**Abstract** — *The use of forward-chaining and backward-chaining in expert system is something usual. Where forward-chaining is a data-driven algorithm, while backward-chaining is goal-driven. Both methods are usually used for reasoning. Here, the writer will analyze effective the both method in their using for expert system.*

**Intisari** — Penggunaan algoritma *forward-chaining* dan *backward-chaining* pada sistem pakar adalah hal yang umum. Dimana proses metode *forward chaining* adalah data-driven, sedangkan metode *backward chaining* adalah *goal-driven*. Kedua metode tersebut umumnya digunakan untuk pencarian dengan kriteria tertentu (*reasoning*). Disini penulis akan menganalisa sejauh mana efektifitas kedua metode tersebut dalam penggunaannya pada sistem pakar.

**Kata kunci:** *Backward-chaining, Forward-chaining, Sistem pakar.*

### PENDAHULUAN

Dalam pembuatan program system pakar, algoritma memiliki kedudukan yang penting, bahkan menjadi jantung dari sistem tersebut. Algoritma system pakar berbeda dengan algoritma system pada umumnya. Dalam system pakar diperlukan suatu *Inference Mechanism* (mekanisme inferensi). Mekanisme inferensi adalah suatu algoritma yang tidak bergantung pada suatu masalah tertentu yang digunakan untuk menentukan kesimpulan-kesimpulan atau menjalankan tindakan-tindakan menggunakan *knowledge base* (dasar pengetahuan).

Mekanisme inferensi yang biasa digunakan untuk system pakar yang bertujuan mencari jawaban dari beberapa kriteria atau dalam terminology artificial intelligence disebut sebagai *Horn clause* – yang merupakan pemisahan dari literal-literal yang paling benar, adalah *forward chaining* dan *backward chaining*. Meskipun masih banyak metode-metode yang lain, namun kedua metode inilah yang paling mudah digunakan. Tetapi sejauh manakah kedua metode tersebut efektif diterapkan dalam system pakar? Disini

penulis akan menganalisa keefektifan kedua metode tersebut dengan melakukan perbandingan metode-metode tersebut pada satu kasus sistem pakar.

### BAHAN DAN METODE

#### *Artificial Intelligent*

Selama ribuan tahun, kita telah mencoba untuk memahami bagaimana cara kita berpikir, yaitu bagaimana segelintir materi dapat melihat, memahami, memprediksi, dan memanipulasi dunia yang lebih besar dan lebih rumit dari dirinya sendiri. Bidang *artificial intelligent* (AI) masih lebih jauh lagi, AI bukan hanya mencoba untuk memahami akan tetapi untuk membuat entitas yang pintar. (Russel & Norvig, 2010). Artificial Intelligent kadang kala juga disebut juga kecerdasan mesin atau pemrograman heuristik, adalah teknologi yang muncul yang belakangan ini menarik publisitas. Banyak aplikasi yang sekarang ini sedang dikembangkan. Salah satu hal yang terlihat dari bidang ini adalah bahwa AI menekankan kepada bagaimana membuat program komputer menjadi lebih pintar (Hunt, 1986).

Dalam bukunya Russel dan Norvig membagi definisi AI menjadi empat kategori, yaitu;

- *Thinking Humanly*; suatu usaha yang luar biasa untuk membuat bagaimana sebuah mesin dapat berpikir seperti layaknya manusia.
- *Acting Humanly*; sebuah seni dari membuat sebuah mesin yang menjalankan fungsi yang membutuhkan kecerdasan apabila dilaksanakan oleh manusia.
- *Thinking Rationally*; sebuah kajian tentang komputasi yang membuatnya menjadi mampu mempersepsikan, berpikir dan bertindak.
- *Acting Rationally*; adalah suatu kajian dari merancang agen (mesin) yang cerdas.

#### *Expert System (Sistem Pakar)*

Selama beberapa dekade terakhir, Sistem Pakar telah menjadi aplikasi praktek yang utama dari riset AI. Dewasa ini, ada banyak system yang berguna dalam hampir setiap bidang operasional diseluruh dunia. Mulai dari gadget sederhana seperti handphone sampai robot-robot dalam

industri manufaktur dan medis. Menurut F. Bobillo dalam Al Ajlan, Sistem Pakar adalah program-program yang memberi saran secara otomatis yang mencoba untuk meniru proses-proses berpikir dan pengetahuan dari ahli-ahli untuk meraih sasaran dari masalah tertentu (Ajlan, 2015). Sebuah Sistem Pakar adalah program computer yang cerdas yang menggunakan pengetahuan dan prosedur inferensi untuk menyelesaikan masalah-masalah yang cukup sulit sehingga membutuhkan kepakaran manusia untuk solusinya (Hunt, 1986).

**Metode Forward Chaining**

Algoritma *forward-chaining* adalah satu dari dua metode utama *reasoning* (pemikiran) ketika menggunakan *inference engine* (mesin pengambil keputusan) dan bisa secara logis dideskripsikan sebagai aplikasi pengulangan dari *modus ponens* (satu set aturan inferensi dan argumen yang valid). Lawan dari *forward-chaining* adalah *backward-chaining*.

*Forward-chaining* mulai bekerja dengan data yang tersedia dan menggunakan aturan-aturan inferensi untuk mendapatkan data yang lain sampai sasaran atau kesimpulan didapatkan. Mesin inferensi yang menggunakan *forward-chaining* mencari aturan-aturan inferensi sampai menemukan satu dari *antecedent* (dalil hipotesa atau klausa IF - THEN) yang benar. Ketika aturan tersebut ditemukan maka mesin pengambil keputusan dapat membuat kesimpulan, atau konsekuensi (klausa THEN), yang menghasilkan informasi tambahan yang baru dari data yang disediakan. Mesin akan mengulang melalui proses ini sampai sasaran ditemukan. *Forward-chaining* adalah contoh konsep umum dari pemikiran yang dikendalikan oleh data

(*data-driven*) yaitu, pemikiran yang mana focus perhatiannya dimulai dari data yang diketahui. *Forward-chaining* bisa digunakan didalam agen untuk menghasilkan kesimpulan dari persepsi-persepsi yang datang, seringkali tanpa query yang spesifik.

**Algoritma Forward Chaining**

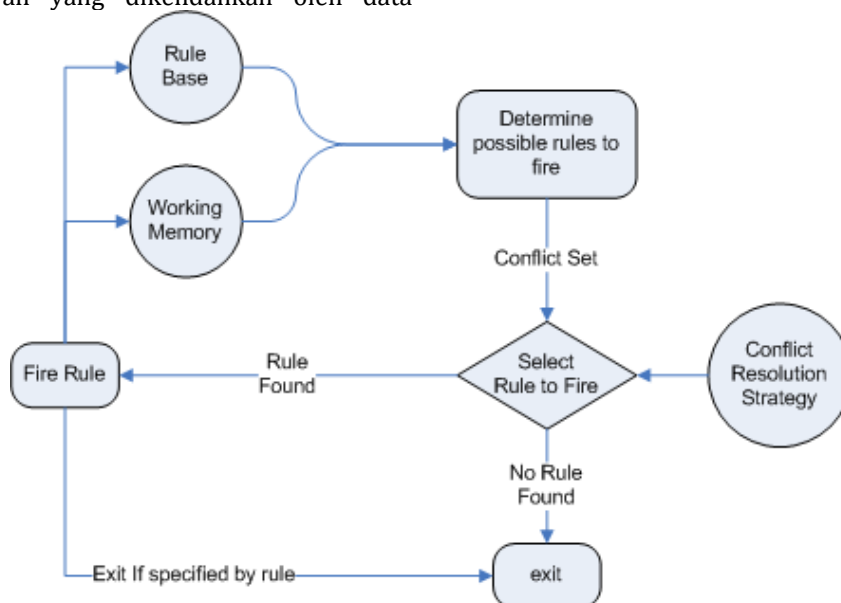
Berikut ini adalah algoritma dari metode *forward-chaining*:

```

function PL-FC-ENTAILS? (KB, Q) returns True
or False
inputs: KB, the knowledge base, a set of
propositional definite clauses q, the
query, a proposition symbol
count ← a table, where count[ e ] is the
number of symbols in c's premise
inferred ← a table, where inferred[ s ] is
initially false for all symbols
agenda ← a queue of symbols, initially
symbols known to be true in KB

while agenda is not empty do
  p ← PoP(agenda)
  if p = q then return true
  if inferred[ p ] = false then
    inferred[ p ] ← true
    for each clause c in KB where p
is in c.PREMISE do
      decrement count[ c ]
      if count[ c ] = 0 then
        add
c.CONCLUSION to agenda
return false
    
```

Jika algoritma diatas ditransformasikan dalam bentuk flow chart adalah seperti gambar 1.



Gambar 1. Flow chart forward chaining (Sharma, Tiwari, & Kelkar, 2012)

**Metode Backward Chaining**

Algoritma *backward-chaining*, sesuai namanya bekerja mundur dari query-nya. Jika query  $q$  diketahui adalah benar, maka tak ada yang perlu dikerjakan selanjutnya. Selain itu, algoritmanya akan mencari implikasi-implikasi di dalam basis data pengetahuan atau *Knowledge Base* (KB) yang kesimpulannya adalah  $q$ . Jika semua premis-premis dari salah satu implikasi-implikasi tersebut bisa dibuktikan benar, maka  $q$  adalah benar (Russel & Norvig, 2010). Jelas sekali disini bahwa *backward-chaining* menggunakan algoritma pencarian *depth-first*.

*Backward-chaining* adalah sebuah bentuk pemikiran yang dikendalikan oleh tujuan atau goal. *Backward-chaining* berguna untuk menjawab pertanyaan-pertanyaan yang spesifik seperti "Apa yang harus aku lakukan sekarang?" dan "Dimana kunci-kunci ku?". Seringkali, harga dari metode *backward-chaining* lebih sedikit dari pada pencarian linear didalam KB, karena prosesnya hanya menyentuh fakta-fakta yang terkait.

**Algoritma Backward Chaining**

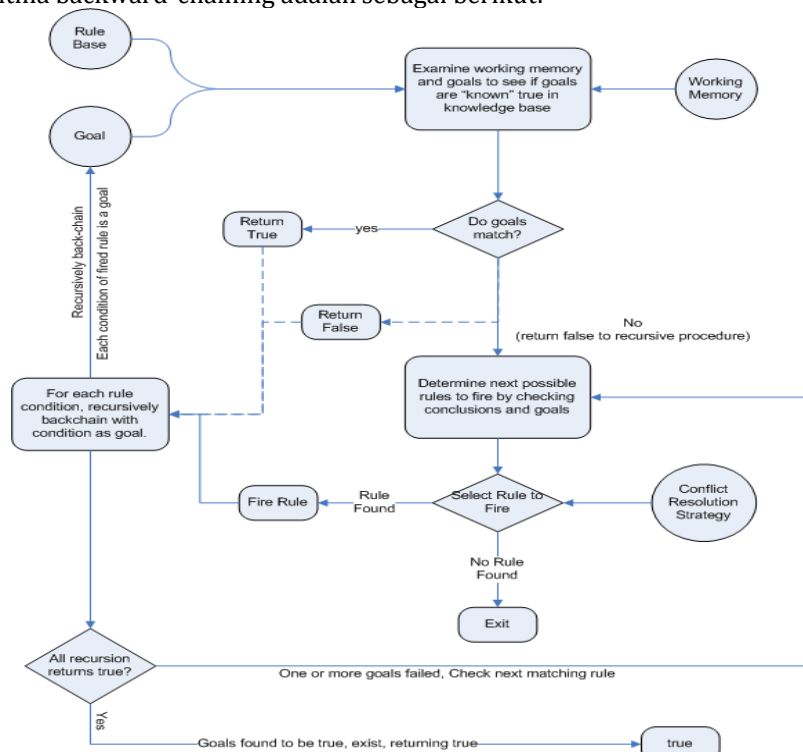
Berikut adalah pseudocode dari algoritma metode *backward-chaining*.

**function** FOL-BC-ASK(KB, query) returns a generator of substitutions **return** FOL-BC-OR ( KB, query, F)

generator **FOL-BC-OR**(KB, goal, 0) yields a substitution  
**for each** rule (/ /is rhs) in **FETCH-RULES-FOR-GOAL**(KB, goal) **do**  
 (lhs, rise)  $\leftarrow$  **STANDARDIZE VARIABLES**((lhs, rhs))  
**for each** 0' in **FOL-BC-AND**(KB, the, **UNIFY**(rhs, goad, 0)) **do**  
 yield 9'

generator **FOL-BC-AND**(KB, goals, 9) yields a substitution  
**if** 0 = failure then return  
**else if** **LENGTH**(goals) = 0 then yield 0  
**else do**  
 first,rest **FIRST**(goals). **REST**(goals)  
**for each** 0' in **FOL-BC-OR**(KB, **SUBST**(0, first), 0) **do**  
**for each** 0'' in **FOL-BC-AND**(KB, rest, 0') **do**  
 yield 8''

flowchart algoritma *backward-chaining* adalah sebagai berikut:



Gambar 2. Flowchart Backward-chaining (Sharma, Tiwari, & Kelkar, 2012)

## HASIL DAN PEMBAHASAN

Apa perbedaan mendasar antara metode forward-chaining dan backward-chaining? Kita akan ilustrasikan dengan rules berikut:

**R1: Jika jagung tumbuh di tanah yang tidak subur, maka jagung tersebut dihindangi lalat hitam.**

**R2: Jika tanah rendah kandungan nitrogen, maka tanah itu tidak subur.**

**Dengan forward-chaining:**

Tanah ini rendah kandungan nitrogen, karena itu tanah ini tidak subur, karena itu jagung yang tumbuh di tanah ini dihindangi lalat hitam.

Reasoning;

- Ada masalah dengan jagung ini.
- Maka uji tanahnya.
- Terbukti bahwa tanahnya rendah kandungan nitrogen.
- Jika itu masalahnya, jagung yang tumbuh pada tanah tersebut akan dihindangi lalat hitam.
- Karena itu masalahnya adalah jagung dihindangi lalat hitam dikarenakan rendahnya kandungan nitrogen pada tanah.

Bagaimana cara kerja metode forward-chaining?

- Proses dikendalikan oleh data (data-driven)
- Mulai dari data/fakta yang diketahui dan proses data tersebut
- Hanya rule paling atas yang diproses setiap saat
- Rule menambah fakta baru ke database ketika diproses
- Setiap rule hanya boleh dieksekusi sekali
- Putaran akan berhenti ketika tidak ada lagi rule yang akan diproses

**Dengan backward-chaining:**

Jagung ini dihindangi lalat hitam, karena itu jagung ini pasti tumbuh di tanah yang tidak subur, karena itu tanahnya pasti rendah kandungan nitrogen.

Reasoning;

- Ada masalah dengan jagung ini.
- Mungkin jagung ini dihindangi lalat hitam.
- Jika iya, maka jagung ini pasti tumbuh di tanah yang tidak subur.
- Jika iya, pasti tanahnya rendah kandungan nitrogen.
- Maka uji kandungan nitrogen pada tanah, maka kita akan tahu apakah benar masalahnya adalah lalat hitam pada jagung.

Bagaimana cara kerja metode backward-chaining?

- Proses dikendalikan oleh tujuan (goal-driven)
- System memiliki tujuan, mekanisme inferensi berusaha mencari fakta untuk membuktikannya

- Mencari di database untuk rule-rule yang mungkin mencapai tujuan atau memiliki tujuan dalam sebagian fakta mereka
- Jika kondisi dari rule tersebut cocok dengan fakta pada database maka rule diproses dan goal dapat dibuktikan

### Studi Kasus

Diketahui rule-rule sebagai berikut:

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

$B \wedge C \Rightarrow M$

$A \wedge B \Rightarrow L$

$Q \Rightarrow C$

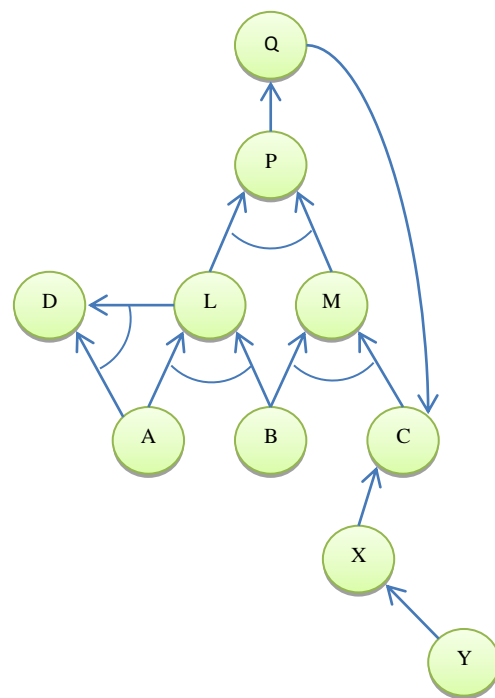
$X \Rightarrow C$

$Y \Rightarrow X$

$L \wedge A \Rightarrow D$

A

B



Gambar 3. Pohon Pakar

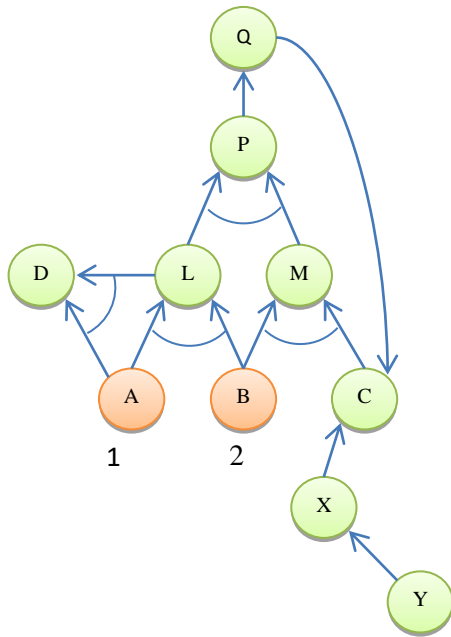
Sedangkan fakta yang diketahui adalah: L, Q, D, P, C

**Proses dengan metode forward-chaining:**

Fact: L

$A, B \Rightarrow L$  Precedent: A, B + L

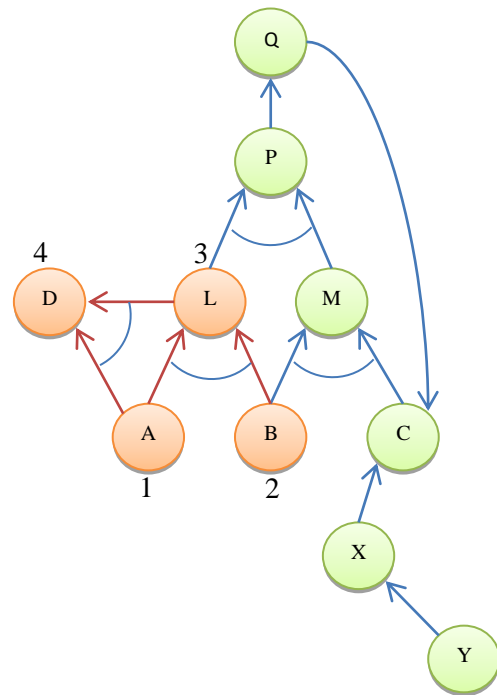
YES



Gambar 4. Proses Fakta L

Fact: Q  
 A, B => L Precedent: A,B + L  
 A, B, L => D Precedent: L,A + D  
 NO

Fact: D  
 A, B => L Precedent: A,B + L  
 A, B, L => D Precedent: L,A + D  
 YES



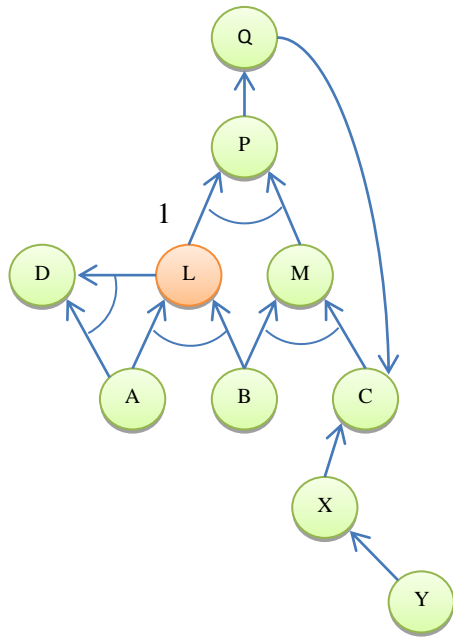
Gambar 5. Proses Fakta D

Fact: P  
 A, B => L Precedent: A,B + L  
 A, B, L => D Precedent: L,A + D  
 NO

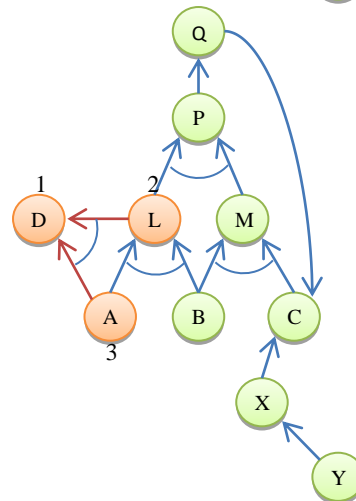
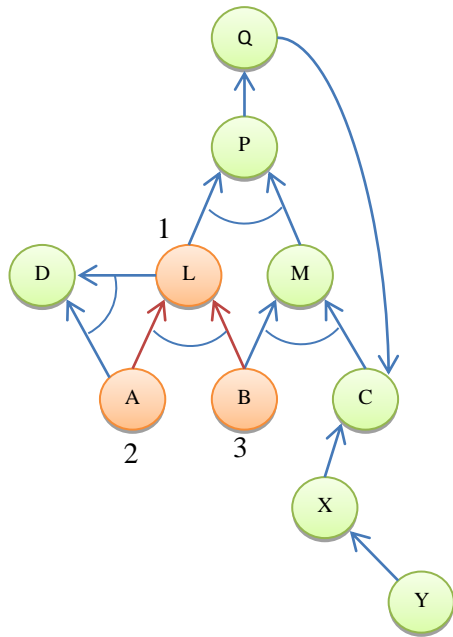
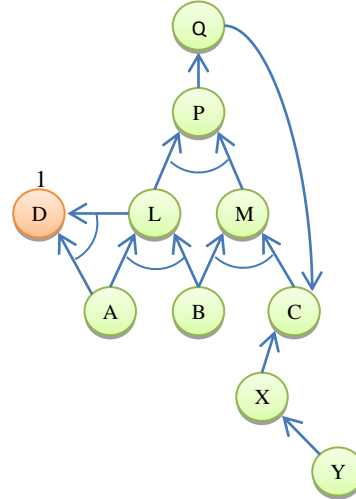
Fact: C  
 A, B => L Precedent: A,B + L  
 A, B, L => D Precedent: L,A + D  
 NO

**Proses dengan metode Backward-chaining:**

Fact: L  
 L : L <= A,B : A, B  
 A : A : None  
 B : B : None  
 YES

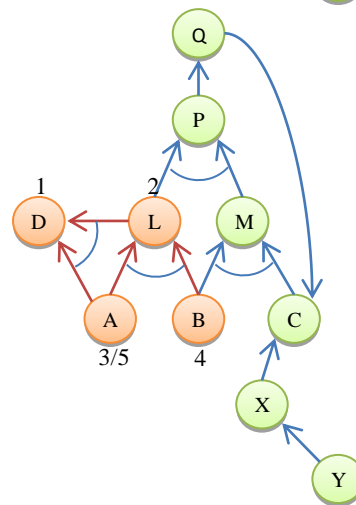


Fact: D  
 D : D <= L, A : L, A  
 L : L <= A, B : A, B  
 A : A : None  
 B : B : None  
 A : A : None  
 YES



Gambar 6. Fakta L

Fact: Q  
 Q : Q <= P : P  
 P : P <= L, M : L, M  
 L : L <= A, B : A, B  
 A : A : None  
 B : B : None  
 M : M <= B, C : B, C  
 B : B : None  
 C : C <= Q : Q  
 Q : CYCLE DETECTED : None  
 C : C <= X : X  
 X : X <= Y : Y  
 Y : None  
 NO



Gambar 7. Fakta D

Fact: P  
 P : P <= L,M : L, M  
 L : L <= A,B : A, B  
 A : A : None  
 B : B : None  
 M : M <= B,C : B, C  
 B : B : None  
 C : C <= Q : Q  
 Q : Q <= P : P  
 P : CYCLE DETECTED : None  
 C : C <= X : X  
 X : X <= Y : Y  
 Y : None  
 NO

Fact: C  
 C : C <= Q : Q  
 Q : Q <= P : P  
 P : P <= L,M : L, M  
 L : L <= A,B : A, B  
 A : A : None  
 B : B : None  
 M : M <= B,C : B, C  
 B : B : None  
 C : CYCLE DETECTED : None  
 C : C <= X : X  
 X : X <= Y : Y  
 Y : None  
 NO

**KESIMPULAN**

Dari hasil analisa diketahui beberapa fakta mengenai metode forward-chaining dan backward-chaining sebagai berikut:

1. Eksplorasi dari pengetahuan memiliki mekanisme yang berbeda antara forward-chaining dan backward-chaining. Backward-chaining lebih terfokus dan mencoba untuk menghindari eksplorasi jalur-jalur yang tidak perlu dari reasoning. Sementara forward chaining seperti pencarian yang melelahkan.
2. Backward-chaining baik untuk tugas-tugas klasifikasi dan diagnosa, tetapi tidak cukup baik untuk perencanaan, perancangan, dan pemantauan proses. Namun forward-chaining cocok untuk tugas-tugas tersebut.
3. Forward-chaining sistem melibatkan penulisan rule-rule untuk mengatur sub goal – sub goal. Sementara backward-chaining secara otomatis mengatur sub goal – sub goal.
4. Gunakan metode forward-chaining untuk system yang memiliki banyak hipotesa keluaran dan data, sebaliknya gunakan backward-chaining untuk system yang sedikit memiliki hipotesa keluaran dan data.
5. Mesin backward-chaining melakukan query untuk fakta baru, sedangkan forward-chaining

- bergantung kepada aplikasi yang menerima input fakta-fakta baru untuk mesin rule.
6. Dalam backward-chaining diarahkan oleh tujuannya/goalnya, jadi rule-rule bisa diterapkan yang dibutuhkan untuk meraih goal. Akan tetapi dalam forward-chaining keseluruhan proses tidak diarahkan kepada tujuan, jadi waktu untuk menghentikan rule-rule tidak diketahui.
  7. Jika fakta-fakta yang diberikan menuntun kepada kesimpulan yang banyak, tetapi cara untuk meraih kesimpulan tertentu sedikit, maka akan lebih banyak informasi yang keluar dari pada informasi yang masuk, maka gunakanlah backward-chaining. Disisi lain, apabila cara untuk meraih kesimpulan tertentu banyak, tetapi jumlah kesimpulan untuk diraih dengan menggunakan fakta adalah sedikit, maka lebih baik menggunakan forward-chaining.

Table 1. Komparasi Forward-chaining dan Backward-chaining

Atribut	Backward-chaining	Forward-chaining
Dikenal juga sebagai	Goal-driven	Data-driven
Dimulai dari	Kesimpulan yang memungkinkan	Data baru
Pemrosesan	Efisien	Melelahkan
Ditujukan untuk	Data yang diperlukan	Kesimpulan apapun
Pendekatan	Konservatif	Oportunistik
Praktis jika	Jumlah jawaban yang mungkin cukup beralasan	Ledakan kombinasi membuat jumlah jawaban menjadi tak tertentu
Cocok untuk	Diagnose, resep, dan debug aplikasi	Perencanaan, pemantauan, kendali dan aplikasi interpretasi
Reasoning	Atas-bawah	Bawah-atas
Jenis pencarian	Depth-first search	Breadth-first search
Siapa yang menentukan pencarian	Konsekuensi menentukan pencarian	Antecedent menentukan pencarian
Alur	Consequent to antecedent	Antecedent to consequent

### REFERENSI

- Ajlan, A. (2015). *The Comparison Between Forward and Backward Chaining*. *International Journal of Machine Learning and Computing*, 106-113.
- Hunt, D. (1986). *Artificial Intelligence & Expert System Source Book*. New York: Chapman and Hall.
- Russel, S., & Norvig, P. (2010). *Artificial Intelligent A Modern Approach Third Edition*. New Jersey: Pearson Education.
- Sharma, T., Tiwari, N., & Kelkar, D. (2012). *Study of Difference Between Forward and Backward Reasoning*. *Ijetae*, 271-273.

Paper Ilmiah yang pernah di hasilkan

- ANALISIS DAN DESAIN SISTEM BERBASIS WEB DENGAN WEB APPLICATION EXTENSION-UML DAN FRAMEWORK MVC
- RANCANG BANGUN SISTEM INFORMASI PERPUSTAKAAN BERBASIS WEB MENGGUNAKAN MVC (MODEL VIEW CONTROLER)
- IMPLEMENTASI PERSISTENCE DENGAN FRAMEWORK HIBERNATE UNTUK MENINGKATKAN EFEKTIFITAS PEMROGRAMAN
- METODE PEMBELAJARAN OBJECT ORIENTED PROGRAMMING (OOP) DENGAN PENDEKATAN HEMISPHERIC COGNITIVE STYLE COLLABORATION

### BIODATA PENULIS



**Ibnu Akil, M.Kom.** Jakarta 15 Januari 1980. Magister Ilmu Komputer Program Pasca Sarjana Nusamandiri. Bekerja sebagai Dosen di AMIK BSI dan Konsultan IT.