

DEDUCTIVE DATABASE

Rachmat Selamat

Sekolah Tinggi Manajemen Informatika dan Komputer LIKMI

Jl. Ir. H. Juanda 96 Bandung 40132

E-mail: if25005@students.itb.ac.id

Abstrak

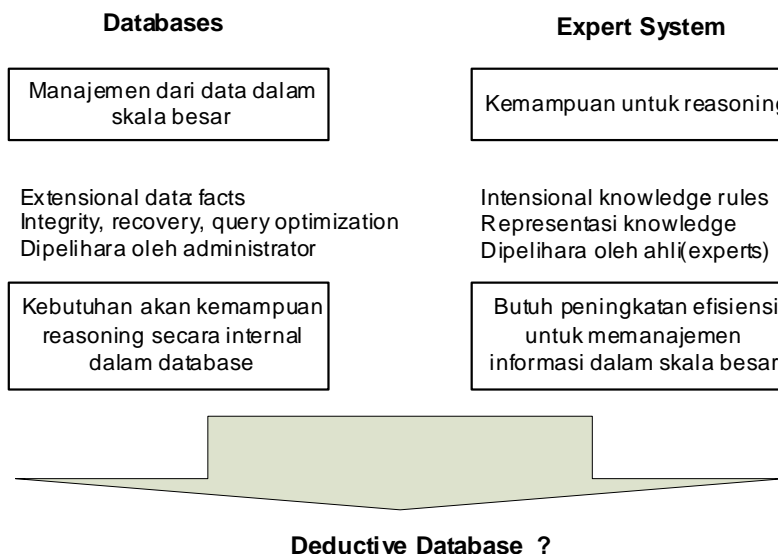
Perkembangan teknologi database sangat cepat sejak DBMS pertama kali muncul pada tahun 1960-an. DBMS sangat berperan penting dalam manufaktur, bank, administrasi umum dan industri lainnya. Sebuah DBMS umumnya digunakan untuk menangani data dalam skala yang besar, walaupun biasanya DBMS melakukan operasi-operasi yang sederhana untuk memanipulasi data.

Pada saat yang bersamaan dengan perkembangan DBMS, perkembangan expert system juga mengalami perkembangan. Expert system biasa digunakan untuk proses pengambilan keputusan, namun expert system tidak dapat menangani informasi dalam skala yang besar.

Kata-kata kunci: Database, expert system, deductive

1. PENDAHULUAN

Ilustrasi perkembangan DBMS dan perkembangan expert system digambarkan pada Gambar 1.



Gambar 1. Perkembangan teknologi database dengan expert system

Deductive database diusulkan sebagai sebuah solusi untuk mengatasi keterbatasan dari DBMS dengan memasukkan fitur unik yang disediakan oleh expert system. Deductive database adalah integrasi dari data (seperti dalam DBMS) dan pengetahuan (seperti dalam expert system).

Sebuah deductive database terdiri dari tiga himpunan finite: himpunan facts (fakta), himpunan deductive rules (aturan deduktif), himpunan integrity constraints (kendala integritas). Facts menyatakan informasi dasar yang diketahui oleh database secara benar. Deductive rules memperbolehkan derivation (penurunan) dari facts yang telah disimpan dalam database menjadi facts yang baru. Integrity constraints menyangkut aturan yang harus dijaga oleh deductive database.

Data yang digambarkan dengan ekstensi dari predikat database (biasanya berupa facts) disebut juga EDB, sedangkan pengetahuan digambarkan dengan instensi dari predikat database disebut juga IDB. Pengetahuan yang didefinisikan dengan rule akan dapat menarik informasi baru dari semua yang tersimpan secara eksplisit di DB. Rule yang dimaksud ada 2 macam, yaitu: deductive rule, yang dapat membantu mendefinisikan facts baru dari facts yang tersimpan, dan integrity constraint, yang menggambarkan kondisi yang tersimpan di database.

Contoh deductive database adalah sebagai berikut:

Facts:

Ayah(Anthony, John)

Ayah(Anthony, Mary)

Ayah(Jack, Anthony)

Ayah(Jack, Rose)

Ibu(Susan, Anthony)

Ibu(Susan, Rose)

Ibu(Rose, Jennifer)

Ibu(Jennifer, Monica)

Deductive Rules:

OrangTua(x,y) <- Ayah(x,y)

OrangTua(x,y) <- Ibu(x,y)

Nenek(x,y) <- Ibu(x,z) ^ Orangtua(z,y)

Kakek(x,y) <- Ayah(x,z) ^ Orangtua(z,y)

Pendahulu(x,y) <- Orangtua(x,y)

Pendahulu(x,y) <- Orangtua(x,z) ^ Pendahulu(z,y)

Pendahulu_tdk_lsg(x,y) <- Pendahulu(x,y) ^ ¬Orangtua(x,y)

Integrity Constraints:

IC1(x) <- OrangTua(x,x)

IC2(x) <- Ayah(x,y) ^ Ibu(x,z)

Penjelasan:

Terdapat sebuah kesepakatan bersama yaitu bahwa aturan pembacaan Ayah(x,y) adalah x adalah ayah dari y. Terdapat 8 fakta tentang ayah dan ibu sebagai ekstensi data. Terdapat 4 deductive rule sebagai notasi instensi, yaitu: Orangtua, Orangtua, Nenek dan Kakek. Terdapat 2 integrity constraint yang menggambarkan bahwa tidak ada orang yang bisa menjadi orangtua bagi dirinya sendiri dan tidak ada orang yang dapat menjadi ayah dan ibu dalam waktu yang bersamaan.

2. DEDUCTIVE DALAM DATABASE MANAGEMENT SYSTEM (DBMS)

Cara penyimpanan deductive database adalah sebagai berikut.

- Memecah tabel menjadi beberapa relasi sesuai dengan facts.

Contoh:

Tabel Keluarga

Nama	JenisKelamin
Anthony	L
Jack	L
Jennifer	P
John	L
Mary	P
Monica	P
Rose	P
Susan	P

Tabel Ayah

NamaAyah	NamaAnak
Anthony	John
Anthony	Mary
Jack	Anthony
Jack	Rose

Tabel Ibu

NamaIbu	NamaAnak
Jennifer	Monica
Rose	Jennifer
Susan	Anthony
Susan	Rose

- Menambahkan field sebagai penghubung (relasi) antar tabel dimana harus sesuai dengan facts. Contoh:

Tabel Keluarga

Nama	JenisKelamin	NamaAyah	NamaIbu
Anthony	L	Jack	Susan
Jack	L	(null)	(null)
Jennifer	P	(null)	Rose
John	L	Anthony	(null)
Mary	P	Anthony	(null)
Monica	P	(null)	Jennifer
Rose	P	Jack	Susan
Susan	P	(null)	(null)

- Deduksi rule dapat disimpan sebagai view. Contoh:

```
CREATE VIEW OrangTua AS
SELECT NamaIbu as Nama, NamaAnak
FROM Ibu
UNION
SELECT NamaAyah as Nama, NamaAnak
FROM Ayah
```

```
CREATE VIEW Nenek AS
SELECT Ibu>NamaIbu, OrangTua>NamaAnak
FROM Ibu, OrangTua
WHERE Ibu>NamaAnak = OrangTua>Nama
```

- Integrity constraint dapat disimpan sebagai assertion. Contoh:

```
CREATE ASSERTION ic2 CHECK (
  NOT EXISTS (
    SELECT Ayah>NamaAnak
    FROM Ayah, Ibu
    WHERE Ayah>NamaAyah = Ibu>NamaIbu
  )
)
```

3. QUERY DEDUCTIVE DATABASE

Cara menggunakan query untuk penerapan deductive database akan dijelaskan sebagai berikut.

➤ Bottom-up (Forward Chaining)

Query mengevaluasi procedure yang dimulai dari facts dasar dan menerapkan semua deductive rule sampai tidak ada suatu informasi yang baru yang dapat ditarik.

➤ Top-down (Backward Chaining)

Query mengevaluasi procedure yang dimulai dari sebuah query Q yang menerapkan deductive rule backward dengan mencoba menarik semua kondisi yang dibutuhkan untuk membuat Q menjadi true.

➤ Magic-set

Gabungan antara Bottom-up dan Top-down, dengan mengambil keuntungan dari Top-down dan ketika mengevaluasi deductive rule digunakan Bottom-up.

Perhatikan contoh di bab 1 (Pendahuluan), Jawab query: Nenek(x, Mary) ?

Berikut ini akan dijelaskan bagaimana jawaban query dikerjakan dengan menggunakan masing-masing penerapan.

3.1 BOTTOM-UP (FORWARD CHAINING)

Langkah-langkah dalam bottom-up adalah sebagai berikut.

Langkah 1.

Semua informasi yang dapat ditarik dari database (dari facts) dihitung dengan iterasi sebagai berikut:

Iterasi 0: Semua fakta ditarik

Iterasi 1: Gunakan rule R1 dari hasil iterasi 0, hasilnya adalah

OrangTua(Anthony, John)

OrangTua(Anthony, Mary)

OrangTua(Jack, Anthony)

OrangTua(Jack, Rose)

Iterasi 2: Gunakan rule R2 dari hasil iterasi 0, hasilnya adalah

OrangTua(Susan, Anthony)

OrangTua(Susan, Rose)

OrangTua(Rose, Jennifer)

OrangTua(Jennifer, Monica)

Iterasi 3: Gunakan rule R3 pada hasil iterasi 0 sampai 2, hasilnya adalah

Nenek(Rose, Monica)
 Nenek(Susan, Jennifer)
 Nenek(Susan, Mary)
 Nenek(Susan, John)

Langkah pertama selesai dengan tidak ada lagi yang dapat ditarik dari rule R1, R2 dan R3, sehingga hasil yang digunakan adalah Iterasi 3.

Langkah 2.

Semua facts sudah dicari melalui iterasi 0 sampai 3 pada Langkah 1. Karena faktanya Nenek(Susan, Mary), maka hasil $x = \text{Susan}$.

Beberapa hal yang harus diperhatikan dalam bottom-up:

- Bottom-up ikut menarik fact yang tidak dibutuhkan oleh query.

Contoh:

OrangTua(Jennifer, Monica), OrangTua(Rose, Jennifer), OrangTua(Jack, Anthony), Nenek(Susan, Jennifer) tidak dibutuhkan tetapi tetap ditarik fact-nya.

- Urutan pemilihan rule yang dievaluasi oleh query harus efisien.

Contoh:

Jika semua fakta ditarik menggunakan R3 (sebelum R1), maka tidak ada jawaban yang dapat diberikan.

- Penghitungan informasi negatif harus dilaksanakan berdasarkan tiap strata.

3.2 TOP-DOWN (BACKWARD CHAINING)

Langkah-langkah dalam top-down adalah sebagai berikut.

Langkah 1.

Query Q dikurangi menjadi Q1 : $\text{Ibu}(x, z) \wedge \text{Orangtua}(z, \text{Mary})$ dengan menggunakan rule R3.

Langkah 2.

Q1 dikurangi menjadi 2 subquery menggunakan R1 atau R2. Hasilnya adalah sebagai berikut:

Q2a : $\text{Ibu}(x, z) \wedge \text{Ayah}(z, \text{Mary})$

Q2b : Ibu(x, z) ^ Ibu(z, Mary)

Langkah 3.

Q2a dikurangi untuk Q3: Ibu(x, Anthony) karena DB memiliki fakta Ayah(Anthony, Mary).

Langkah 4.

Q2b tidak menghasilkan jawaban karena tidak ada fakta yang sesuai dengan Ibu(z, Mary).

Langkah 5.

Q3 dievaluasi dengan EDB dan memberikan x = Susan sebagai hasilnya.

Beberapa hal yang harus diperhatikan pada Top-down:

- Top-down biasanya satu tuple pada satu waktu. Setiap fakta diperiksa oleh top-down satu persatu, sehingga top-down kurang efisien.
- Top-down mungkin tidak diterminasi. Untuk rule yang bersifat rekursif, pengevaluasian top-down dapat menyebabkan loop tidak terhingga dan tidak pernah diterminasi.

Contoh:

Menggunakan top-down untuk menghitung pendahulu (pendahulu harus meng-query pendahulu lagi)

- Tidak mungkin untuk memperkirakan bahwa selalu, pada waktu pendefinisian, apakah sebuah algoritma top-down terminasi. Dengan pendekatan top-down, kita tidak akan pernah tahu kapan metode ini akan menyelesaikan pengeksesusiannya, dan ternyata memakan banyak waktu untuk menjawab.
- Subquery yang berulang. Sejak proses pengurangan query yang sebenarnya menjadi subquery yang lebih sederhana akan menghasilkan hasil yang sama, kadang-kadang subquery harus digunakan beberapa kali. Di beberapa kasus mungkin dapat menyebabkan pengevaluasian ulang subquery yang dapat menyebabkan pengurangan efisiensi dalam evaluasi.

3.3 MAGIC SET

Contoh:

```
Pendahulu(x, y) <- OrangTua(x, y)
```

```
Pendahulu(x, y) <- OrangTua(x, z) ^ Pendahulu(z, y)
```

Deductive rule yang digunakan untuk Pendahulu(Rose,x):

```
Magic_dhl(Rose)
```

```
Pendahulu(x, y) <- Magic_dhl(x) ^ OrangTua(x, y) (R1)
```

```
Magic_dhl(z) <- Magic_dhl(x) ^ OrangTua(x, z) (R2)
```

```
Pendahulu(x, y) <- Magic_dhl(x) ^ OrangTua(x, z) ^ Pendahulu(z, y) (R3)
```

Asumsikan semua fact tentang orangtua sudah dihitung, hasilnya OrangTua(Rose, Jennifer) dan OrangTua(Jennifer, Monica).

Langkah-langkah dalam magic set dengan menggunakan bottom-up:

Langkah 1.

Iterasi 1: Pendahulu(Rose, Jennifer) ditarik dengan rule R1
 Iterasi 2: Magic_dhl(Jennifer) ditarik dengan rule R2
 Iterasi 3: Tidak ada yang bisa ditarik dengan rule R3
 Iterasi 4: Pendahulu(Jennifer, Monica) ditarik dengan rule R1
 Iterasi 5: Magic_dhl(Monica) ditarik dengan rule R2
 Iterasi 6: Pendahulu(Rose, Monica) ditarik dengan rule R3
 Iterasi 7: Tidak ada lagi yang dapat ditarik rule R1, R2 dan R3

Langkah 2.

Hasilnya adalah {Pendahulu(Rose, Jennifer), Pendahulu(Rose, Monica)}.

Catatan: Penarikan fact diperoleh menggunakan bottom-up dengan hanya menarik fact yang sesuai dengan permintaan query. Hal ini disebabkan adanya predikat Magic_dhl, yang ditambahkan di setiap rule. Hanya dengan Magic_dhl(Rose), kita dapat menghitung keturunan dari Rose.

4. KESIMPULAN

Deductive database dapat dilihat sebagai gabungan data (seperti DBMS) dan pengetahuan (seperti expert system). Data yang digambarkan dengan ekstensi dari predikat database (biasanya berupa facts), sedangkan pengetahuan digambarkan dengan instensi dari predikat database. Pengetahuan yang didefinisikan dengan rule akan dapat menarik informasi baru dari semua yang tersimpan secara eksplisit di database. Rule yang dimaksud ada 2 macam, yaitu: deductive rule, yang dapat membantu mendefinisikan fact baru dari fact yang tersimpan, dan integrity constraint, yang menggambarkan kondisi data yang diperbolehkan untuk disimpan di database.

5. DAFTAR PUSTAKA

1. Piattini, Mario (Editorial) dan Oscar Diaz (Editorial). *Advanced Database Technology and Design*. Artech House, Inc. 2000.
2. <http://www.cs.mu.oz.au/research/aditi>
3. <http://www.vldb.org/journal>
4. <http://www.wsmo.org/wsmo/papers/presentation>
5. <http://www.ncbi.nlm.nih.gov/entrez>