

Join Berantai

Djoni Darmawikarta

djoni_darmawikarta@yahoo.ca

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Dikantor saya, salah satu bagian dari sistem data warehouse-nya menerapkan database struktur ternormalisasi (*normalized*) yang setiap tabelnya menggunakan surrogate key sebagai primary key-nya (baca tulisan saya tentang Surrogate Key yang sudah diterbitkan di IKC)

Konsekuensi dari kedua teknik ini (*normalized* dan *surrogate key*) adalah hubungan berantai yang diperlukan untuk menghubungkan suatu data dengan data lain.

Sebagai contoh, untuk mencari informasi polis dari suatu premi, misalnya pemegang polis, kita harus menghubungkan 5 tabel sebagai berikut:

Polis (*policy*), Versi Polis (*policy version*, sebuah polis bisa memiliki beberapa versi), Unit Resiko (*risk unit*, barang yang diasuransikan, misalnya mobil, bisa lebih dari satu untuk sebuah versi), Pencakupan (*coverage*, jenis resiko yang diasuransikan, bisa lebih dari satu untuk sebuah mobil), dan Premium (transaksi premi, umumnya lebih dari satu)

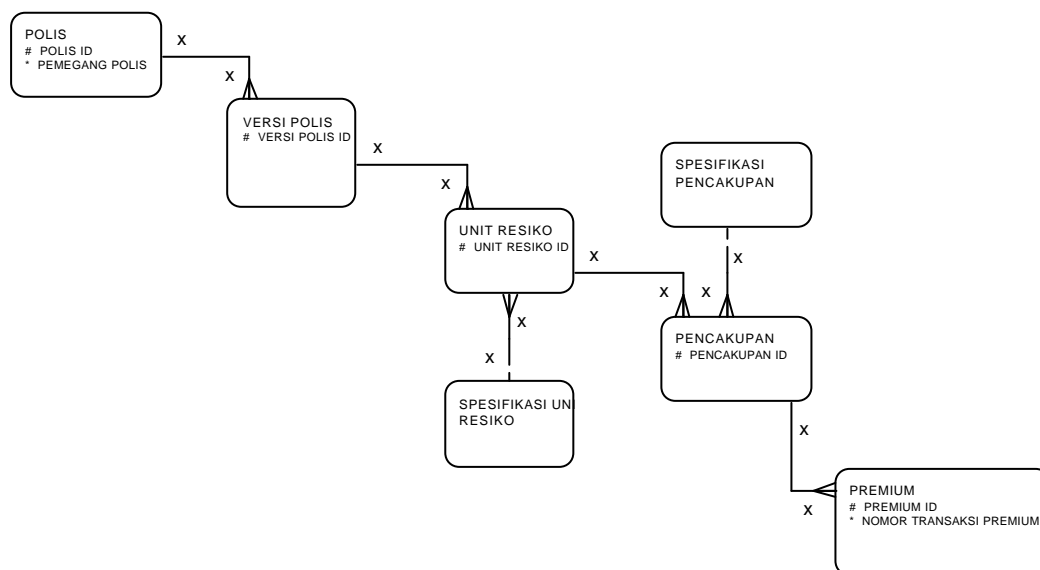


Diagram data model yang menunjukkan entity dari ke 5 tabel tersebut terlihat digambar diatas ini (ada 2 entity lain yang sebenarnya ada, tetapi tidak dibahas dicontoh tulisan ini)

Atribut berakhiran _id adalah primary key berupa surrogate key.

Fasilitas join di database relasional memudahkan kita untuk melaksanakan rantai hubungan yang diperlukan. Untuk kasus contoh diatas, join yang diperlukan di SQL-nya saya tuliskan berwarna **biru** dibawah ini; **xxxxxx** adalah nomor transaksi premium yang kita ingin ketahui pemegang polis-nya.

```
SELECT polis.pemgang_polis
FROM polis, versi_polis, unit_resiko, pencakupan, premium
WHERE
nomor_transaksi_premium = xxxxxx AND
premium.pencakupan_id = pencakupan.pencakupan_id AND
pencakupan.unit_resiko_id = unit_resiko.unit_resiko_id AND
unit_resiko.versi_polis_id = versi_polis.versi_polis_id AND
versi_polis.polis_id = polis.polis_id
```

Rantai join ini dimulai dari entity PREMIUM bergerak keatas menuju POLIS dengan menggunakan primary key-nya (atributnya yang berakhiran _id)

Dalam pengalaman saya, **visualisasi diagram data model** sangat membantu untuk merangkai join, bahkan diperlukan bila tabel-tabel-nya lebih banyak jumlahnya dan kompleks (sejumlah aplikasi yang pernah saya tangani database-nya memiliki lebih dari 250 entity/tabel yang saling berhubungan dan tidak mungkin menuliskan rantai join SQL melibatkan 5 s/d 10 entity/tabel dengan tepat dan efisien tanpa melihat diagram data modelnya)