

# Perbandingan *Cross-Product* dan *Subset Query* pada *Multiple Relasi* dengan Metode *Cost-Based*

Metta Santiputri<sup>1)</sup> Mira Chandra Kirana<sup>1)</sup> Anni<sup>2)</sup>

1) Program Studi Teknik Informatika, Politeknik Batam

E-mail: metta@polibatam.ac.id.ac.id, mira@polibatam.ac.id

2) Divisi Information System, PT Panasonic Shikoku Electronics Batam

E-mail: anni@mkpi.panasonic.co.id

**Abstrak** - Ada beberapa model query yang digunakan untuk mengakses data pada 2 tabel atau lebih dalam basis data relasi. Dua model query yang umum antara lain adalah *cross product* dan *subset query*, dimana kedua model ini dapat menghasilkan data yang sama. Namun perlu diperhatikan cara mana yang lebih optimal sehingga pada akhirnya didapatkan query dengan waktu akses yang paling minimum. Dengan menggunakan basis data Oracle 10g Express Edition akan dilakukan penelitian untuk mencari model query yang lebih optimal dengan metode *cost-based*. Parameter yang akan dibandingkan adalah harga/biaya dan waktu yang dihasilkan pada perencanaan eksekusi. Penelitian dilakukan dengan pengelompokan uji data seperti jumlah data, jumlah relasi, akses tabel penuh atau sebagian dan pengindeksan.

**Keywords** : basis data, query, cross product, subset, cost, optimasi

## 1 PENDAHULUAN

*Database Management System* (DBMS) pada dasarnya adalah suatu cara memelihara basis data sehingga pengguna dapat menjalankan operasi seperti memasukkan, mengubah, menghapus, memanipulasi, dan mengambil data dengan mudah. *Structured Query Language* (SQL) merupakan standar bahasa untuk operasi-operasi umum pada basis data yang dapat digunakan secara luas, baik oleh aplikasi internal, aplikasi bahasa pemrograman yang mendukung basis data, dan juga aplikasi manajemen buatan dari pihak diluar pengembang aplikasi manajemen basis data itu sendiri.

Kehandalan dari DBMS dapat diketahui dari cara kerja *optimizer* dalam memproses SQL yang dibuat oleh pengguna. Di dalam *optimizer*, query yang ada diproses dengan berbagai cara sehingga mendapatkan perencanaan query yang optimal.

Kemajuan teknologi DBMS saat ini telah mencapai penggunaan *multiuser*, *multi processing* dan aplikasi sistem lebih cenderung ke *client-server*, *web base* dan bahkan ke arah *mobile application*. Aplikasi-aplikasi ini akan mengakses basis data secara bersama-sama. Dalam sistem basis data relasional sering juga dibutuhkan pengaksesan data terhadap beberapa tabel sekaligus (*multiple relasi*). Ada beberapa cara dalam

melakukan akses ini, namun perlu diperhatikan cara mana yang lebih optimal sehingga pada akhirnya didapatkan query dengan waktu akses yang paling minimum. Sehingga perlu dirumuskan bagaimana model query yang lebih optimal dalam melakukan pencarian data.

## 2 DASAR TEORI

### 2.1 Metode Optimasi *Cost-Based*

Teknik ini mengoptimasikan *cost* yang dipergunakan dari beberapa alternatif untuk kemudian dipilih salah satu yang menjadi *cost* terendah. Teknik ini mengoptimalkan urutan *join* terbalik yang dimungkinkan pada relasi-relasi  $r_1 \rightarrow r_2 \rightarrow \dots r_n$ . Teknik ini dipergunakan untuk mendapatkan pohon *left-deep join* yang akan menghasilkan sebuah relasi sebenarnya pada *node* sebelah kanan yang bukan hasil dari sebuah *intermediate join*. Komponen-komponen harga yang digunakan untuk mengeksekusi query adalah:

a. *Access cost* untuk *secondary storage*

Harga ini adalah harga untuk pencarian, pembacaan dan penulisan blok-blok data yang terletak pada *secondary storage*, terutama pada disk. Harga dari pencarian untuk *record-record* dalam sebuah file tergantung pada tipe dari bentuk-bentuk akses pada file tersebut, seperti pengurutan (*ordering*), *hashing* dan pengindeksan *primary* ataupun *secondary*. Sebagai tambahan, faktor-faktor seperti disediakan atau tidaknya blok-blok file yang berdekatan pada silinder disk yang sama atau tersebar pada disk juga dapat mempengaruhi harga akses.

b. *Storage cost*

Harga ini adalah harga dari penyimpanan file-file menengah yang dihasilkan oleh sebuah strategi eksekusi untuk query.

c. *Computation cost*

Harga ini adalah harga dari pelaksanaan operasi-operasi memori pada *buffer* data selama eksekusi query. Seperti operasi-operasi pencarian dan pengurutan *record*, penggabungan *record* untuk sebuah *join* dan melakukan perhitungan-perhitungan pada nilai-nilai *field*.

d. *Memory usage cost*

Harga ini adalah harga mengenai jumlah dari *buffer* memori yang diperlukan selama eksekusi query.

e. *Communication cost*

Harga ini adalah harga dari pengiriman *query* dan hasilnya dari tempat basisdata atau terminal dimana *query* berasal.

Untuk melihat kinerja sistem basis data dalam pencarian data, ada 2 model *query* yang akan digunakan, yaitu *cross product* dan *subset query*.

## 2.2 Cross-Product Query

*Cross product* atau disebut juga *cross join* digunakan saat mengkombinasikan data pada dua tabel atau lebih. Model *cross product* diwakili oleh *query* berikut ini:

```
select [nama_kolom1],..., [nama_kolomN]
from [nama_tabel1], [nama_tabel2]
where [nama_tabel1].[nama_kolom1] =
[nama_tabel2].[nama_kolom2]
```

*Query* tersebut melakukan *select* sejumlah kolom dari 2 tabel atau lebih dimana pengkondisian masing-masing tabel di *join*.

## 2.3 Subset Query

*Subset query* dibagi atas 3 macam bentuk *query*, antara lain sebagai berikut:

### 1. Scalar

Yaitu melakukan *select* sejumlah kolom dari satu tabel dimana kondisi suatu kolom terpenuhi pada satu sub *query*. *Scalar query* diwakili oleh *query* berikut:

```
select [nama_kolom1],..., [nama_kolomN]
from [nama_tabel1]
where [nama_tabel1].[nama_kolom1] in
(select [nama_kolom1] from [nama_tabel2])
```

### 2. Correlated

Yaitu melakukan *select* sejumlah kolom pada satu tabel dimana kondisi suatu kolom terpenuhi pada satu sub *query*, dan sub *query* tersebut pengkondisiannya masih berhubungan dengan *super query*. *Correlated query* diwakili oleh *query* berikut:

```
select [nama_kolom1],..., [nama_kolomN]
from [nama_tabel1]
where [nama_tabel1].[nama_kolom1] in
(select [nama_kolom1] from [nama_tabel2]
where [nama_tabel1].[nama_kolom2]
=[nama_tabel2].[nama_kolom2])
```

### 3. Kombinasi

Bentuk kombinasi ini adalah menggabungkan bentuk *cross product* dengan *subset query*. Kombinasi tersebut adalah:

#### • Cross Product dengan Scalar

Melakukan *select* sejumlah kolom pada beberapa tabel dimana pengkondisian antar tabel menggunakan *join* dan pengkondisian suatu kolom dipenuhi oleh satu sub *query*. *Query* tersebut diwakili oleh:

```
select [nama_kolom1],..., [nama_kolomN]
from [nama_tabel1], [nama_tabel2]
where [nama_tabel1].[nama_kolom1]
=[nama_tabel2].[nama_kolom2]
AND [nama_tabel1].[nama_kolom1] in
(select [nama_kolom1]
from [nama_tabel3])
```

#### • Multi Scalar

Melakukan *select* sejumlah kolom pada satu tabel dimana kondisi suatu kolom terpenuhi

pada satu sub *query* dan sub *query* tersebut terdapat satu sub *query* lagi. *Query* ini diwakili oleh:

```
select [nama_kolom1],..., [nama_kolomN]
from [nama_tabel1]
where [nama_tabel1].[nama_kolom1] in
(select [nama_kolom1] from [nama_tabel2]
where [nama_tabel2].[nama_kolom2] in
(select [nama_kolom2]
from [nama_tabel3]))
```

## 3 PENGUJIAN

### 3.1 Parameter Perbandingan

Optimasi *query* pada Oracle 10g XE akan dihitung dengan menggunakan *Explain Plan* sebagai fungsi yang terdapat dalam Oracle. Perintah *Explain Plan* akan menampilkan rencana eksekusi suatu *query*. Adapun 6 parameter yang akan dihasilkan adalah sebagai berikut:

1. *Cardinality*, perkiraan dengan pendekatan *cost-based* berdasarkan jumlah baris yang diakses oleh operasi.
2. *Bytes*, perkiraan oleh pendekatan *cost-based* berdasarkan jumlah *byte* yang diakses oleh operasi.
3. *Cost*, perkiraan oleh pendekatan *cost-based* berdasarkan harga/biaya yang diakses oleh operasi. Nilai tersebut tidak memiliki satuan khusus. Fungsi *cost* didefinisikan dengan *time units*, merujuk pada sumber daya seperti alokasi ruang penyimpanan, I/O (*input/output*) *disk*, *buffer space*, CPU (*Control Process Unit*) *cost*, dan *communication cost*. *Cost* tersebut merupakan gabungan dari seluruh I/O, CPU dan *communication cost*.
4. *CPU Cost*, perkiraan oleh pendekatan *cost-based* berdasarkan jumlah siklus mesin yang dibutuhkan untuk operasi. Rumusan hitungan adalah sebagai berikut:

$$\text{CPUCost} = (\#SRDs * \text{sreadtim} + \#MRDs * \text{mreadtim} + \#CPUCycles / \text{cpuspeed}) / \text{sreadtim}$$

- dengan #SRDs adalah jumlah *single* blok yang dibaca, #MRDs adalah jumlah *multi* blok yang dibaca, #CPUCycles adalah jumlah CPU *cycles*, Sreadtim adalah waktu membaca *single* blok, Mreadtim adalah waktu membaca *multi* blok, dan CPUSpeed adalah CPU *cycles* per detik
5. *IO Cost*, perkiraan oleh pendekatan *cost-based* berdasarkan jumlah data blok yang dibaca untuk operasi.
  6. *Time*, waktu yang diperkirakan oleh pendekatan *cost-based* berdasarkan jumlah waktu yang dibutuhkan untuk operasi dalam satuan detik.

Dalam penelitian ini parameter yang akan dibandingkan ada 3, yaitu *cardinality*, *cost* dan *time*.

### 3.2 Batasan

Model *query* yang dieksekusi pada saat perbandingan memenuhi kriteria sebagai berikut:

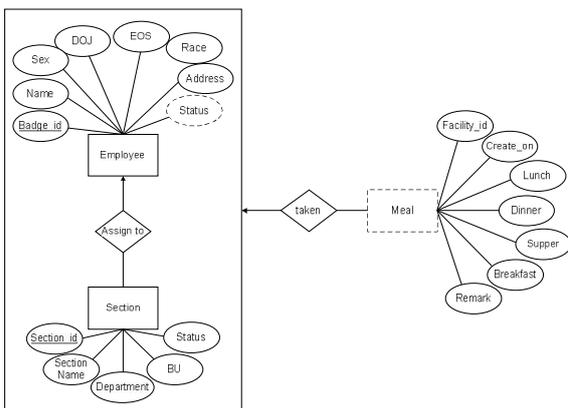
- a. Dijalankan pada struktur tabel yang sama, seperti

- jenis tipe data, pengindeksan dan *constraints*.
- b. Data yang dihasilkan oleh *query-query* tersebut adalah sama.
- c. Tabel yang diuji merupakan tabel non partisi.
- d. Model *subset query* yang diujikan antara lain: *scalar*, kombinasi antara *cross product* dengan *subset*, dan kombinasi *scalar (multi scalar)*. *Correlated* tidak diujikan karena model ini tidak dapat menghasilkan data yang sama dengan model-model *query* yang lain.

**3.3 Deskripsi Basis Data**

Studi kasus yang diambil sebagai contoh uji coba perhitungan *cost query* adalah sistem kantin perusahaan. Sistem kantin ini bertujuan untuk melakukan perhitungan jumlah karyawan yang makan pada saat jam makan. Sebelum pengambilan makan, karyawan akan melakukan *swap* pada mesin yang terhubung pada suatu aplikasi. Rata-rata jumlah karyawan yang makan berkisar antara 1500-2000 orang per waktu makan, yaitu pada saat siang, malam, tengah malam dan subuh.

Jam kerja karyawan dibagi atas 2 *shift* (siang dan malam). Karyawan hanya mengambil 2 kali jatah makan seperti untuk *shift* siang pada makan siang dan malam, sedangkan untuk *shift* malam pada makan tengah malam dan subuh. Karyawan dapat berpindah departemen sesuai kebijakan perusahaan, oleh karena itu setiap jatah makan yang diambil perlu diikuti keterangan departemen. Hal ini bertujuan dapat menghitung jumlah karyawan yang makan per departemen. Sistem kantin digambarkan dalam diagram E-R di gambar 1.



**Gambar 1: Diagram ER**

**3.4 Skenario Pengujian**

Pengujian model *query* dibagi atas beberapa group. Group dengan perbedaan jumlah data dan group dengan atau tanpa menggunakan indeks. Sebagai uji coba penelitian, tabel Meal akan dibagi menjadi 4 tabel, hal ini dilakukan untuk memudahkan uji eksekusi *query* dengan jumlah data yang berbeda-beda seperti terlihat pada tabel 1.

**Tabel 1. Skala data pengujian**

No	Nama	Jumlah Data	Group Data
1	Meal_500	500	Kecil
2	Meal_5K	5,000	Sedang
3	Meal_50K	50,000	Besar
4	Meal_500K	500,000	Sangat Besar

**3.5 Benchmark Query**

*Query-query* yang akan digunakan sebagai uji coba dalam penelitian ini akan dibagi dalam beberapa relasi, yaitu dengan 2 relasi dan 3 relasi. Dengan group yang sama tersebut dibagi lagi atas 2 pengindeksan, pertama dengan pengujian menggunakan indeks dan yang kedua tanpa indeks. Pengujian *query* pada 2 relasi dilakukan 2 tahap, pertama dengan akses tabel secara penuh, dengan tujuan pengekseskuan *query* agar membaca seluruh blok data dalam perhitungan *cost*. Kedua, dengan akses tabel sebagian. Pengujian 3 relasi dilakukan dengan akses tabel secara penuh saja. *Query* yang digunakan dalam pengujian ini berdasarkan group-group yang telah dibagi berjumlah 72 *query*. Secara rinci dapat dilihat pada lampiran.

**4 HASIL DAN PEMBAHASAN**

**4.1 Cardinality**

Pada parameter *cardinality*, nilai yang dihasilkan adalah sama. Hal ini menunjukkan bahwa hasil eksekusi *query* yang diujikan pada model *query* yang berbeda menghasilkan data (*output*) yang sama. Namun pada model *Multi Scalar*, nilai *cardinality* pada tabel tanpa indeks adalah berjumlah 1. Nilai pengujian untuk parameter *cardinality* dapat dilihat pada lampiran.

**4.2 Cost dan Time**

Pengujian dilakukan pada *query* yang menggunakan 2 tabel dengan indeks pada tahapan akses tabel secara penuh menghasilkan data yang menunjukkan bahwa model *Scalar* memiliki nilai *cost* dan *time* paling kecil mulai dari group data kecil hingga group data sangat besar. Perbedaan nilai parameter tersebut antara *Cross Product* dan *Scalar* mengalami perubahan yang sangat jauh jika jumlah data semakin besar.

Sedangkan pengujian yang dilakukan pada *query* dengan 2 tabel tanpa menggunakan indeks pada tahapan akses tabel secara penuh menunjukkan bahwa model *Scalar* memiliki nilai *cost* dan *time* paling kecil. Namun untuk group data kecil, *cost* dan *time* pada kedua model ini memiliki nilai yang sama. Perubahan terjadi ada pada group data yang lebih tinggi.

Pada pengujian dilakukan untuk *query* yang menggunakan 2 tabel dengan indeks pada tahapan

akses tabel sebagian, hasil pengujian menunjukkan bahwa nilai *time* yang dihasilkan oleh model *Cross Product* dan *Scalar* hampir mendekati, kecuali dengan skala data yang sangat besar. Namun perbedaan tersebut tidaklah terlalu jauh.

Pada pengujian berikutnya yaitu pada *query* dengan 2 tabel tanpa menggunakan indeks pada tahapan akses tabel sebagian, ternyata hasil dari pengujian ini tidak jauh beda dengan hasil pengujian menggunakan indeks. Nilai *time* yang dihasilkan oleh model *Cross Product* dan *Scalar* hampir mendekati, kecuali dengan skala data yang sangat besar. Namun perbedaan tersebut tidaklah terlalu jauh.

Jika kedua parameter dibandingkan antar sesama model pada pengujian dengan indeks maupun tanpa indeks, baik akses tabel secara penuh maupun sebagian, dapat dilihat bahwa tidak ada perubahan yang berarti pada model *Cross Product*, namun berbeda pada model *Scalar*, mengalami perubahan yang besar. Walaupun demikian model *Scalar* tetap unggul dalam pengujian ini.

Pengujian berikutnya adalah pengujian untuk *query* dengan 3 tabel yang menggunakan indeks. Perbandingan 3 relasi dengan 5 model *query* ternyata menunjukan hasil yang menarik. Model *Scalar*, Kombinasi 2 dan *Multi Scalar* memiliki nilai yang cenderung sama. Sedangkan *Cross Product* memiliki nilai yang cenderung sama dengan model Kombinasi 1. Sama halnya pada pengujian 2 relasi, nilai model *Scalar*, Kombinasi 2 dan *Multi Scalar* memiliki nilai yang paling kecil di setiap group data.

Sedangkan jika pengujian dilakukan pada *query* dengan 3 tabel yang menggunakan indeks, nilai yang dihasilkan pada model *Cross Product*, Kombinasi 1 dan Kombinasi 2 tidak jauh berbeda dengan pengujian dengan indeks. Model *Scalar* mengalami perubahan nilai *cost* yang tinggi jika dibandingkan dengan pengujian dengan indeks. Hal yang menjadi perhatian ada pada model *Multi Scalar*, nilai *cost* dan *time* pada pengujian ini menghasilkan nilai yang tidak terduga. Nilai tersebut terlalu jauh dibandingkan dengan pengujian-pengujian sebelumnya.

Data hasil pengujian dapat dilihat secara lengkap pada lampiran.

Berdasarkan data-data hasil pengujian, maka dapat diurutkan model-model mana yang lebih optimal dalam pencarian data berdasarkan nilai *cost* dan *time*.

**Tabel 2. Peringkat Optimasi**

Relasi	Akses Tabel	Group Data	Index	Peringkat Optimasi	
				Pertama	Teraakhir
2 Relasi	Penuh	Kecil	Ya	Scalar	Cross Product
			Tidak	Scalar	Cross Product
		Sedang	Ya	Scalar	Cross Product
			Tidak	Scalar	Cross Product

Relasi	Akses Tabel	Group Data	Index	Peringkat Optimasi	
				Pertama	Teraakhir
		Besar	Ya	Scalar	Cross Product
			Tidak	Scalar	Cross Product
		Sangat Besar	Ya	Scalar	Cross Product
			Tidak	Scalar	Cross Product
2 Relasi	Sebagian	Kecil	Ya	Scalar	Cross Product
			Tidak	Scalar	Cross Product
		Sedang	Ya	Scalar	Cross Product
			Tidak	Scalar	Cross Product
		Besar	Ya	Scalar	Cross Product
			Tidak	Scalar	Cross Product
		Sangat Besar	Ya	Scalar	Cross Product
			Tidak	Scalar	Cross Product
3 Relasi	Penuh	Kecil	Ya	Scalar, Multi Scalar	Cross Product
			Tidak	Scalar, Kombinasi 2	Multi Scalar
		Sedang	Ya	Scalar, Multi Scalar	Cross Product
			Tidak	Scalar, Kombinasi 2	Multi Scalar
		Besar	Ya	Scalar, Multi Scalar	Cross Product
			Tidak	Scalar, Kombinasi 2	Multi Scalar
		Sangat Besar	Ya	Scalar, Multi Scalar	Cross Product
			Tidak	Scalar, Kombinasi 2	Multi Scalar

## 5. KESIMPULAN

*Subset query* pada bentuk *scalar* dan kombinasi 1 merupakan model yang lebih baik dalam pencarian data jika dibandingkan dengan *cross product*, baik dalam 2 relasi maupun 3 relasi. Jika menggunakan model *cross product* dalam pencarian data, maka pada bagian kolom yang dikondisi harus diindekskan.

Perlu diperhatikan bahwa penelitian ini dilakukan pada tabel non partisi, sehingga untuk penelitian lebih lanjut perlu dilakukan juga pada tabel yang menggunakan partisi.

### REFERENSI

- [1] Kusriani, S. Kom., *Optimasi Query Untuk Pencarian Data dengan Subset Query*, Bandung, 2006.
- [2] Setiawan, M.A., *Optimasi SQL Query untuk Informasi Retrieval pada Aplikasi Berbasis Web*, Proceedings Seminar Nasional Aplikasi Teknologi Informasi UII, Yogyakarta, 2004.
- [3] Immanuel Chan, *Oracle Database Performance Tuning Guide, 10g Release 2 (10.2)*, Redwood City, CA, Maret 2008, pp. 379-403.
- [4] Tom Best dan M.J. Billings, *Oracle Database 10g: Administration Workshop I*, Electronic Presentation, Redwood Shores, California USA, November 2005, pp. 41-42.
- [5] Sandra Cheevers, *Oracle Database Product Family*, An Oracle White Paper, Redwood Shores, CA USA, Agustus 2006, pp 3-4.
- [6] Sagi Arsyad, *Pengenalan .NET dan C#*, Microsoft Innovation Center, Universitas Indonesia, Jakarta, 2008.
- [7] Aris A.b, *Optimisasi Query Pada Sistem Database Paralel*, (2008 September 20). Tersedia: <http://xuyas.wordpress.com>
- [8] Richard Foote, *OPTIMIZER\_INDEKS\_CACHING Parameter*, (2009 September 1). Tersedia: <http://richardfoote.wordpress.com>

pada jurusan yang sama. Pada tahun 2009 beliau mulai mengajar di Program Studi Teknik Informatika Politeknik Batam sampai sekarang.

Anni dilahirkan di Batam pada tanggal 13 Mei 1984. Menyelesaikan diploma di Politeknik Batam pada tahun 2005 dan menyelesaikan studi sarjana di Universitas Maritim Raja Ali Haji (UMRAH) pada tahun 2010. Saat ini beliau bekerja di Divisi Information System, PT Panasonic Shikoku Electronics Batam.

### BIOGRAFI

Metta Santiputri dilahirkan di Yogyakarta pada tanggal 20 Juli 1977. Menyelesaikan studi sarjana di Jurusan Teknik Informatika, Institut Teknologi Bandung (ITB) pada tahun 2000 dan program master di Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS) pada university of Twente (Belanda) pada tahun 2006. Sejak lulus dari ITB, langsung mengajar pada Program Studi Teknik Informatika di Politeknik Batam sampai sekarang.

Mira Chandra Kirana dilahirkan di Surabaya pada tanggal 30 Mei 1979. Menyelesaikan studi sarjana pada tahun 2006 di Jurusan Teknik Elektro, Program Studi Sistem Komputer, Institut Teknologi Sepuluh November, Surabaya (ITS). Beliau mulai mengajar sejak tahun 2006 di Universitas Muhammadiyah Malang pada Jurusan Teknik Informatika, kemudian dilanjutkan mengajar pada Politeknik Kota Malang

## LAMPIRAN

## A. Query yang digunakan dalam pengujian

Kode Query	Relasi	Model Query	Group Data	Indeks
Q1	2, akses tabel penuh	Cross Product	Kecil	Ya
Q2	2, akses tabel penuh	Scalar	Kecil	Ya
Q3	2, akses tabel penuh	Cross Product	Sedang	Ya
Q4	2, akses tabel penuh	Scalar	Sedang	Ya
Q5	2, akses tabel penuh	Cross Product	Besar	Ya
Q6	2, akses tabel penuh	Scalar	Besar	Ya
Q7	2, akses tabel penuh	Cross Product	Sangat Besar	Ya
Q8	2, akses tabel penuh	Scalar	Sangat Besar	Ya
Q9	2, akses tabel penuh	Cross Product	Kecil	Tidak
Q10	2, akses tabel penuh	Scalar	Kecil	Tidak
Q11	2, akses tabel penuh	Cross Product	Sedang	Tidak
Q12	2, akses tabel penuh	Scalar	Sedang	Tidak
Q13	2, akses tabel penuh	Cross Product	Besar	Tidak
Q14	2, akses tabel penuh	Scalar	Besar	Tidak
Q15	2, akses tabel penuh	Cross Product	Sangat Besar	Tidak
Q16	2, akses tabel penuh	Scalar	Sangat Besar	Tidak
Q17	2, akses tabel sebagian	Cross Product	Kecil	Ya
Q18	2, akses tabel sebagian	Scalar	Kecil	Ya
Q19	2, akses tabel sebagian	Cross Product	Sedang	Ya
Q20	2, akses tabel sebagian	Scalar	Sedang	Ya
Q21	2, akses tabel sebagian	Cross Product	Besar	Ya
Q22	2, akses tabel sebagian	Scalar	Besar	Ya
Q23	2, akses tabel sebagian	Cross Product	Sangat Besar	Ya
Q24	2, akses tabel sebagian	Scalar	Sangat Besar	Ya
Q25	2, akses tabel sebagian	Cross Product	Kecil	Tidak
Q26	2, akses tabel sebagian	Scalar	Kecil	Tidak
Q27	2, akses tabel sebagian	Cross Product	Sedang	Tidak
Q28	2, akses tabel sebagian	Scalar	Sedang	Tidak
Q29	2, akses tabel sebagian	Cross Product	Besar	Tidak
Q30	2, akses tabel sebagian	Scalar	Besar	Tidak
Q31	2, akses tabel sebagian	Cross Product	Sangat Besar	Tidak
Q32	2, akses tabel sebagian	Scalar	Sangat Besar	Tidak
Q33	3, akses tabel penuh	Cross Product	Kecil	Ya
Q34	3, akses tabel penuh	Scalar	Kecil	Ya
Q35	3, akses tabel penuh	Kombinasi 1	Kecil	Ya
Q36	3, akses tabel penuh	Kombinasi 2	Kecil	Ya
Q37	3, akses tabel penuh	Multi Scalar	Kecil	Ya
Q38	3, akses tabel penuh	Cross Product	Sedang	Ya
Q39	3, akses tabel penuh	Scalar	Sedang	Ya
Q40	3, akses tabel penuh	Kombinasi 1	Sedang	Ya
Q41	3, akses tabel penuh	Kombinasi 2	Sedang	Ya
Q42	3, akses tabel penuh	Multi Scalar	Sedang	Ya
Q43	3, akses tabel penuh	Cross Product	Besar	Ya
Q44	3, akses tabel penuh	Scalar	Besar	Ya
Q45	3, akses tabel penuh	Kombinasi 1	Besar	Ya
Q46	3, akses tabel penuh	Kombinasi 2	Besar	Ya
Q47	3, akses tabel penuh	Multi Scalar	Besar	Ya
Q48	3, akses tabel penuh	Cross Product	Sangat Besar	Ya
Q49	3, akses tabel penuh	Scalar	Sangat Besar	Ya
Q50	3, akses tabel penuh	Kombinasi 1	Sangat Besar	Ya
Q51	3, akses tabel penuh	Kombinasi 2	Sangat Besar	Ya
Q52	3, akses tabel penuh	Multi Scalar	Sangat Besar	Ya
Q53	3, akses tabel penuh	Cross Product	Kecil	Tidak

Kode Query	Relasi	Model Query	Group Data	Indeks
Q54	3, akses tabel penuh	Scalar	Kecil	Tidak
Q55	3, akses tabel penuh	Kombinasi 1	Kecil	Tidak
Q56	3, akses tabel penuh	Kombinasi 2	Kecil	Tidak
Q57	3, akses tabel penuh	Multi Scalar	Kecil	Tidak
Q58	3, akses tabel penuh	Cross Product	Sedang	Tidak
Q59	3, akses tabel penuh	Scalar	Sedang	Tidak
Q60	3, akses tabel penuh	Kombinasi 1	Sedang	Tidak
Q61	3, akses tabel penuh	Kombinasi 2	Sedang	Tidak
Q62	3, akses tabel penuh	Multi Scalar	Sedang	Tidak
Q63	3, akses tabel penuh	Cross Product	Besar	Tidak
Q64	3, akses tabel penuh	Scalar	Besar	Tidak
Q65	3, akses tabel penuh	Kombinasi 1	Besar	Tidak
Q66	3, akses tabel penuh	Kombinasi 2	Besar	Tidak
Q67	3, akses tabel penuh	Multi Scalar	Besar	Tidak
Q68	3, akses tabel penuh	Cross Product	Sangat Besar	Tidak
Q69	3, akses tabel penuh	Scalar	Sangat Besar	Tidak
Q70	3, akses tabel penuh	Kombinasi 1	Sangat Besar	Tidak
Q71	3, akses tabel penuh	Kombinasi 2	Sangat Besar	Tidak
Q72	3, akses tabel penuh	Multi Scalar	Sangat Besar	Tidak

#### B. Data Cardinality pada 2 Relasi – Akses Tabel Penuh

Group Data	Indeks	Cross Product	Scalar
Kecil	Ya	500	500
Kecil	Tidak	500	500
Sedang	Ya	5,000	5,000
Sedang	Tidak	5,000	5,000
Besar	Ya	43,037	43,037
Besar	Tidak	43,037	43,037
Sangat Besar	Ya	423,522	423,522
Sangat Besar	Tidak	423,522	423,522

#### C. Data Cardinality pada 2 Relasi – Akses Tabel Sebagian

Group Data	Indeks	Cross Product	Scalar
Kecil	Ya	10	10
Kecil	Tidak	10	10
Sedang	Ya	109	109
Sedang	Tidak	109	109
Besar	Ya	944	944
Besar	Tidak	753	753
Sangat Besar	Ya	8,995	8,995
Sangat Besar	Tidak	8,995	8,995

#### D. Data Cardinality pada 3 Relasi

Group Data	Indeks	Cross Product	Scalar	Kombinasi 1	Kombinasi 2	Multi Scalar
Kecil	Ya	500	500	500	500	500
Kecil	Tidak	500	500	500	500	1
Sedang	Ya	5,000	5,000	5,000	5,000	5,000
Sedang	Tidak	5,000	5,000	5,000	5,000	1
Besar	Ya	43,037	43,037	43,037	43,037	43,037
Besar	Tidak	43,037	43,037	43,037	43,037	1
Sangat Besar	Ya	423,522	423,522	423,522	423,522	423,522
Sangat Besar	Tidak	423,522	423,522	423,522	423,522	1

**E. Hasil Pengujian 2 Relasi Akses Tabel Secara Penuh**

Group Data	Indeks	Kode Query	Model Query	Cost	Time (detik)
Kecil	Ya	Q2	Scalar	5	1
Kecil	Ya	Q1	Cross Product	93	2
Kecil	Tidak	Q10	Scalar	93	2
Kecil	Tidak	Q9	Cross Product	93	2
Sedang	Ya	Q4	Scalar	20	1
Sedang	Ya	Q3	Cross Product	676	9
Sedang	Tidak	Q12	Scalar	105	2
Sedang	Tidak	Q11	Cross Product	1,060	13
Besar	Ya	Q6	Scalar	175	3
Besar	Ya	Q5	Cross Product	1,635	20
Besar	Tidak	Q14	Scalar	246	3
Besar	Tidak	Q13	Cross Product	1,635	20
Sangat Besar	Ya	Q8	Scalar	1,527	19
Sangat Besar	Ya	Q7	Cross Product	7,332	88
Sangat Besar	Tidak	Q16	Scalar	1,591	20
Sangat Besar	Tidak	Q15	Cross Product	7,332	88

**F. Hasil Pengujian 2 Relasi Akses Tabel Sebagian**

Group Data	Indeks	Kode Query	Model Query	Cost	Time (detik)
Kecil	Ya	Q18	Scalar	3	1
Kecil	Ya	Q17	Cross Product	13	1
Kecil	Tidak	Q26	Scalar	92	2
Kecil	Tidak	Q25	Cross Product	93	2
Sedang	Ya	Q20	Scalar	17	1
Sedang	Ya	Q19	Cross Product	106	2
Sedang	Tidak	Q28	Scalar	105	2
Sedang	Tidak	Q27	Cross Product	106	2
Besar	Ya	Q22	Scalar	157	2
Besar	Ya	Q21	Cross Product	245	3
Besar	Tidak	Q30	Scalar	244	3
Besar	Tidak	Q29	Cross Product	245	3
Sangat Besar	Ya	Q24	Scalar	1,485	18
Sangat Besar	Ya	Q23	Cross Product	2,338	29
Sangat Besar	Tidak	Q32	Scalar	1,569	19
Sangat Besar	Tidak	Q31	Cross Product	2,568	31

**G. Hasil Pengujian 3 Relasi**

Group Data	Indeks	Kode Query	Model Query	Cost	Time (detik)
Kecil	Ya	Q37	Multi Scalar	5	1
Kecil	Ya	Q34	Scalar	5	1
Kecil	Ya	Q36	Kombinasi 2	9	1
Kecil	Ya	Q35	Kombinasi 1	94	2
Kecil	Ya	Q33	Cross Product	97	2
Kecil	Tidak	Q54	Scalar	96	2
Kecil	Tidak	Q56	Kombinasi 2	96	2
Kecil	Tidak	Q55	Kombinasi 1	97	2
Kecil	Tidak	Q53	Cross Product	97	2
Kecil	Tidak	Q57	Multi Scalar	1,729	21
Sedang	Ya	Q39	Scalar	21	1
Sedang	Ya	Q42	Multi Scalar	21	1
Sedang	Ya	Q41	Kombinasi 2	23	1
Sedang	Ya	Q40	Kombinasi 1	677	9
Sedang	Ya	Q38	Cross Product	679	9

Group Data	Indeks	Kode Query	Model Query	Cost	Time (detik)
Sedang	Tidak	Q59	Scalar	109	2
Sedang	Tidak	Q61	Kombinasi 2	109	2
Sedang	Tidak	Q60	Kombinasi 1	1,063	13
Sedang	Tidak	Q58	Cross Product	1,063	13
Sedang	Tidak	Q62	Multi Scalar	17,670	213
Besar	Ya	Q44	Scalar	178	3
Besar	Ya	Q47	Multi Scalar	178	3
Besar	Ya	Q46	Kombinasi 2	180	3
Besar	Ya	Q45	Kombinasi 1	1,638	20
Besar	Ya	Q43	Cross Product	1,640	20
Besar	Tidak	Q66	Kombinasi 2	251	4
Besar	Tidak	Q64	Scalar	251	4
Besar	Tidak	Q65	Kombinasi 1	1,640	20
Besar	Tidak	Q63	Cross Product	1,640	20
Besar	Tidak	Q67	Multi Scalar	43,615	524
Sangat Besar	Ya	Q52	Multi Scalar	1,538	19
Sangat Besar	Ya	Q49	Scalar	1,538	19
Sangat Besar	Ya	Q51	Kombinasi 2	1,540	19
Sangat Besar	Ya	Q50	Kombinasi 1	7,342	89
Sangat Besar	Ya	Q48	Cross Product	7,344	89
Sangat Besar	Tidak	Q71	Kombinasi 2	1,604	20
Sangat Besar	Tidak	Q69	Scalar	1,604	20
Sangat Besar	Tidak	Q70	Kombinasi 1	7,344	89
Sangat Besar	Tidak	Q68	Cross Product	7,344	89
Sangat Besar	Tidak	Q72	Multi Scalar	49,307	592