



ESTIMASI QUERY

Sistem Basis Data

Gentisya Tri Mardiani, M.Kom

Estimasi Biaya Query

- Optimizer query akan membuat informasi statistik yang tersimpan dalam katalog DBMS untuk memperkirakan besarnya biaya dari sebuah rencana query.
- Informasi yang tersimpan meliputi:
 - Banyaknya baris data (record) dalam sebuah tabel
 - Banyaknya blok berisi baris data dalam tabel
 - Ukuran setiap baris data dari tabel (satuan byte)
 - Banyaknya nilai unik dalam tabel untuk suatu atribut

Pengukuran Biaya Query

- Biaya evaluasi query dapat diukur dari banyaknya sumber daya (resource) sistem yang terpakai, meliputi pengaksesan disk, waktu CPU mengerjakan query, dan untuk sistem basis data paralel atau terdistribusi.
- Pada sistem basis data yang besar dapat memberikan kontribusi biaya query yang dominan.
- Perhitungan waktu CPU lebih sukar untuk diprediksikan, maka biaya pengekseskuan sebuah rencana query yaitu waktu yang dibutuhkan untuk pengaksesan ke disk.

Ekivalensi Ekspresi Relasional

π Nama ($\sigma_{\text{nama_bag}='Akunting'}$ (**bagian** \bowtie **pekerjaan** \bowtie **pribadi**))

Karena atribut nama_bag itu terdapat di tabel bagian, maka dapat direduksi terlebih dahulu banyaknya baris data yang akan dilibatkan dalam operasi join, dengan mengubah ekspresi di atas menjadi ekspresi yang ekivalen:

π Nama ($\sigma_{\text{nama_bag}='Akunting'}$ (**bagian**) \bowtie (**pekerjaan** \bowtie **pribadi**))

Ekivalensi ekspresi operasi Seleksi

- Aturan:

- Jalankan operasi seleksi seawal mungkin (prioritaskan operasi seleksi)

contoh:

$\sigma_{\text{jenis_kelamin}='P' \wedge \text{kota}='Yogya'}$ (pribadi)

- Ganti ekspresi yang berbentuk

$\sigma_{P1 \wedge P2}(E)$ menjadi $\sigma_{P1}(\sigma_{P2}(E))$

Ekivalensi ekspresi operasi natural join

- Memilih urutan operasi Join yang optimal, untuk semua relasi r_1, r_2, r_3 maka:

$$(r_1 \bowtie r_2) \bowtie r_3 \text{ menjadi } r_1 \bowtie (r_2 \bowtie r_3)$$

Contoh:

$(\text{pribadi} \bowtie \text{pekerjaan}) \bowtie \text{bagian}$

$\text{pribadi} \bowtie (\text{pekerjaan} \bowtie \text{bagian})$

meskipun ekspresi di atas sama, namun secara komputasi operasi bisa berbeda

Aturan Ekuivalensi

1. Operasi seleksi konjungtif dapat direkonstruksi ke dalam sebuah sekuen seleksi individual

$$\sigma_{P1 \wedge P2} (E) = \sigma_{P1}(\sigma_{P2}^{(E)})$$

2. Operasi seleksi bersifat komutatif

$$\sigma_{P1}(\sigma_{P2}^{(E)}) = \sigma_{P2}(\sigma_{P1}^{(E)})$$

3. Hanya operasi final dalam sekuen operasi proyeksi yang diperlukan

$$\pi_{L1} (\pi_{L2} (E)) = \pi_{L1}(E)$$

4. Seleksi dapat dikombinasikan dengan cartesian product dan theta join

$$\sigma_{\theta}(E1 \times E2) = E1 \bowtie_{\theta} E2$$

Aturan Ekivalensi

5. Operasi theta join bersifat komutatif

$$E1 \bowtie_{\theta} E2 = E2 \bowtie_{\theta} E1$$

6. Operasi natural join bersifat asosiatif

$$(E1 \bowtie E2) \bowtie E3 = E1 \bowtie (E2 \bowtie E3)$$

7. Operasi union dan intersection bersifat komutatif

$$E1 \cup E2 = E2 \cup E1, \quad E1 \cap E2 = E2 \cap E1$$

8. Operasi union dan intersection bersifat asosiatif

$$(E1 \cup E2) \cup E3 = E1 \cup (E2 \cup E3)$$

$$(E1 \cap E2) \cap E3 = E1 \cap (E2 \cap E3)$$

Aturan Ekuivalensi

9. Operasi seleksi dapat didistribusikan ke operasi union, intersection, dan set difference

$$\sigma_P(E1 - E2) = \sigma_P(E1) - E2 = \sigma_P(E1) - \sigma_P(E2)$$

10. Operasi proyeksi dapat didistribusikan ke operasi union

$$\pi_L(E1 \cup E2) = \pi_L(E1) \cup \pi_L(E2)$$

Ekuivalensi aljabar relasional dan SQL

	Aljabar Relasional	SQL
Selection	$\sigma_P(E)$ Contoh: $\sigma_{\text{kota}='Bantul'}(\text{Pribadi})$	Select * from E where P Contoh: Select * from pribadi where kota='Bantul'
Projection	$\pi_{\text{column}}(E)$ Contoh: $\pi_{\text{NIP, Nama}}(\text{Pribadi})$	Select column from E Contoh: Select NIP, Nama from Pribadi

	Aljabar Relasional	SQL
Union	<p>$E1 \cup E2$</p> <p>Contoh: Pribadi \cup Pekerjaan</p>	<p>select * from E1 union select * from E2</p> <p>Contoh: Select * from pribadi union select * from pekerjaan</p>
Set Difference	<p>$E1 - E2$</p> <p>Contoh: $\pi_{NIP}(\text{Pribadi}) - \pi_{NIP}(\text{Pekerjaan})$</p>	<p>Select * from E1 except select * from E2</p> <p>Contoh: Select NIP from pribadi except select NIP from pekerjaan</p>
Cartesian Product	<p>$E1 \times E2$</p> <p>Contoh: $\pi_{NIP, Nama, Gaji}$ $(\sigma_{\text{Pribadi.NIP}=\text{Pekerjaan.NIP}}$ (Pribadi x Pekerjaan))</p>	<p>Select * from E1, E2</p> <p>Contoh: Select Pribadi.NIP, Pribadi>Nama, Pekerjaan.Gaji from Pribadi, Pekerjaan where Pribadi.NIP = Pekerjaan.NIP;</p>

	Aljabar Relasional	SQL
Set Intersection	<p>$E1 \cap E2$</p> <p>Contoh: Pribadi \cap Pekerjaan</p>	<pre>select * from E1 intersect select * from E2</pre> <p>Contoh: Select * from pribadi intersect select * from pekerjaan</p>
Join	<p>$E1 \bowtie E2$</p> <p>Contoh: Pribadi \bowtie Pribadi.NIP=Pekerjaan.NIP Pekerjaan</p>	<pre>Select * from E1 join E2</pre> <p>Contoh: Select * from pribadi join Pekerjaan on Pribadi.NIP=Pekerjaan. NIP</p>

Struktur Sistem Basis Data

- Tujuan utama dari sistem basis data adalah untuk memudahkan dan memfasilitasi akses ke data.
- Faktor utama yang menjadi parameter kepuasan user terhadap sistem basis data adalah performansinya.
- Performansi sistem tergantung pada:
 - Efisiensi struktur data (penyimpanan) yang digunakan/ dipilih
 - Seberapa efisien sistem tersebut dapat beroperasi pada struktur data tersebut

Struktur DBMS untuk Pemrosesan Query

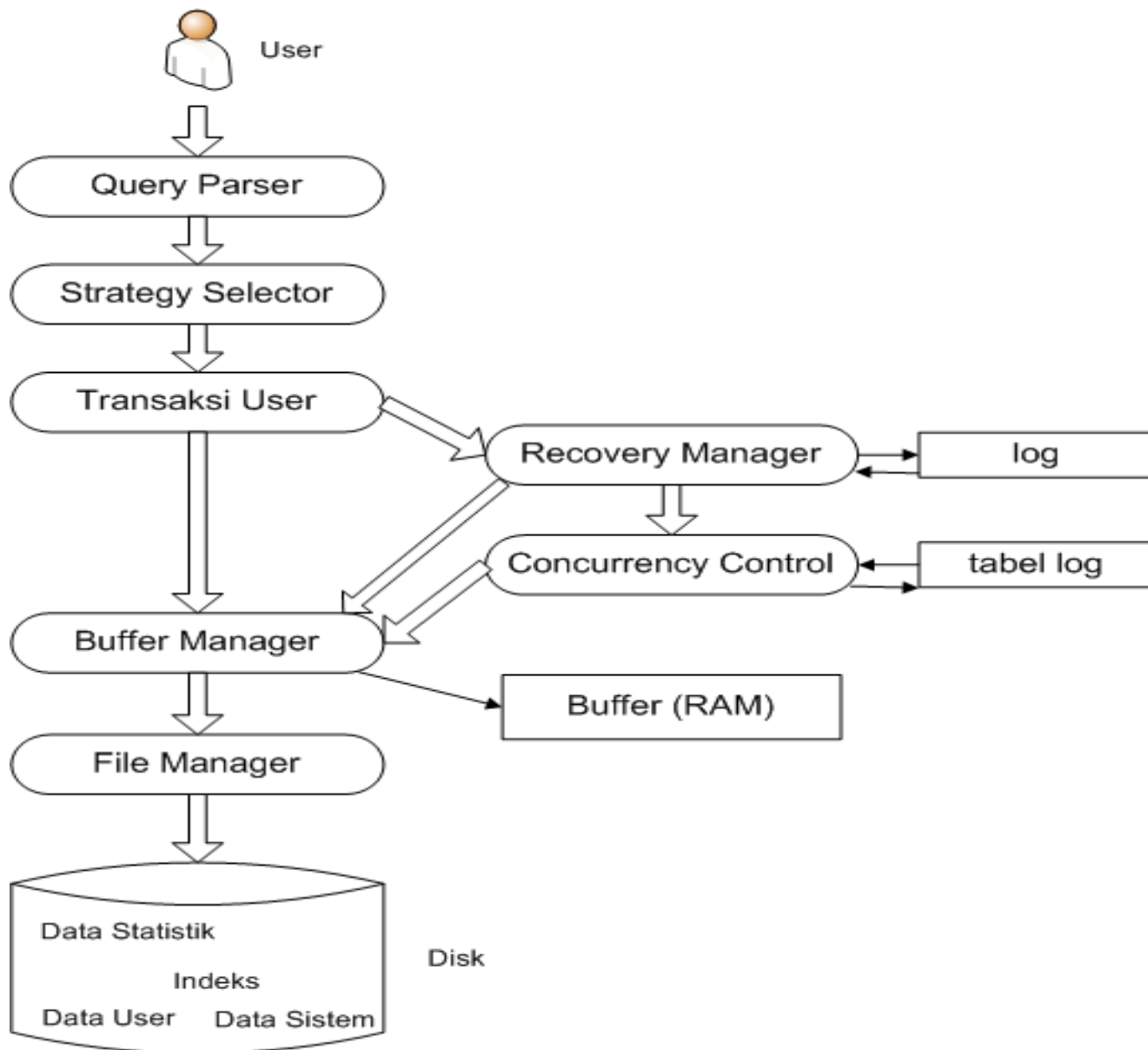
- **File manager**, yang mengelola alokasi dalam disk dan struktur data yang digunakan untuk merepresentasikan informasi yang tersimpan dalam disk
- **Buffer manager**, yang bertanggung jawab dalam pentransferan informasi antara disk dan memori utama
- **Query parser**, yang menerjemahkan perintah dalam query language ke dalam bahasa mesin
- **Strategy selector**, yang mentransformasikan permintaan user ke dalam bentuk lain yang sama tetapi lebih efisien, kemudian menentukan strategi terbaik untuk menjalankan query

Struktur DBMS untuk Pemrosesan Query

- **Authorization / integrity manager**, yang memeriksa pemenuhan batasan-batasan integritas dan otoritas user untuk mengakses data
- **Recovery manager**, yang menjamin bahwa basis data dapat tetap konsisten setelah kegagalan/ kerusakan sistem insidental
- **Concurrency controller**, yang menjamin interaksi pada basis data secara konkuren dilaksanakan tanpa adanya konflik antar user

Struktur data yang dibutuhkan dalam implementasi fisik

- **File Data**, yang merupakan basis data itu sendiri
- **File Data Sistem**, yang menyimpan informasi tentang struktur basis data, contoh isi file data sistem adalah kamus data
- **Data Statistik**, yang menyimpan informasi spesifik tentang data dalam basis data. Informasi ini bermanfaat bagi pemilihan strategi operasi yang diminta user



Tugas Kelompok

- Membangun Database untuk sistem informasi suatu organisasi dari dokumen manual

Sistem informasi:

Penjualan/Point Of Sale (POS), perpustakaan, rumah sakit, koperasi, akademik, HRM/kepegawaian, inventory, perbankan, rental/penyewaan, perhotelan

- Database minimal terdiri dari 3 tabel
- Gunakan prinsip Normalisasi (1, 2, 3 ..dsb)
- Buat ERD, Skema Relasi, dan Struktur tabel

- Create Database dan tabel (type data disesuaikan dengan kebutuhan)
- Query:
- Aljabar relasional (5 soal)
- Insert, update, delete dan select (dari 1 tabel, 2 tabel dsb) (5 soal)

- Dikumpulkan minggu depan dalam bentuk laporan dan presentasi
- Waktu sampai pertemuan ke-6 (sebelum UTS)