



# OPTIMASI QUERY

Sistem Basis Data

Gentisya Tri Mardiani, S.Kom., M.Kom

# 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

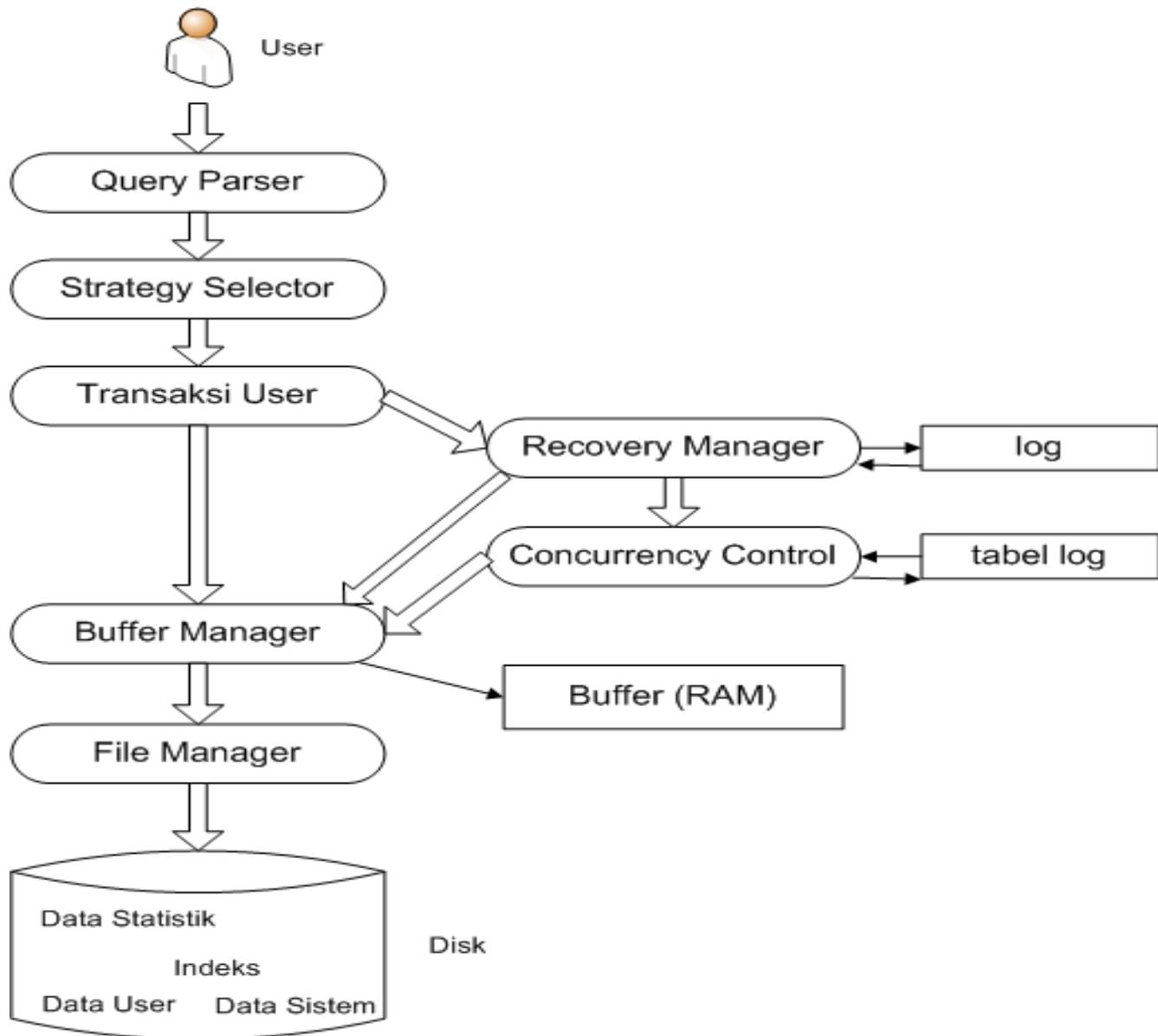
- **File manager**, yang mengelola alokasi dalam disk dan struktur data yang digunakan untuk merepresentasikan informasi yang tersimpan dalam disk
- **Buffer manager**, yang bertanggung jawab dalam pentransferan informasi antara disk dan memori utama
- **Query parser**, yang menerjemahkan perintah dalam 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



# Estimasi 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 ( $n_r$ )
  - Banyaknya blok berisi baris data dalam tabel ( $b_r$ )
  - Ukuran setiap baris data dari tabel (satuan byte)( $s_r$ )
  - Banyaknya baris data di tabel  $r$  yang bisa masuk ke dalam blok ( $f_r$ )
  - Banyaknya nilai unik dalam tabel  $r$  untuk atribut  $A$  ( $V(A,r)$ )
  - Jika  $A$  merupakan key dari tabel  $r$ , maka  $V(A,r) = n_r$

# 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.

# Ekivalensi Ekspresi Relasional

Contoh:

```
SELECT Nama_jabatan, Gaji_pokok FROM jabatan  
WHERE gaji_pokok < 2000000;
```

1.  $\pi$  Nama\_jabatan, Gaji\_pokok ( $\sigma_{\text{Gaji\_pokok} < 2000000}$  (jabatan))
2.  $\sigma_{\text{Gaji\_pokok} < 2000000}$  ( $\pi$  Nama\_jabatan, Gaji\_pokok (jabatan))

Apa perbedaannya?

## Tabel Pegawai

NIP	Nama	Tgl_lahir	Jenis_kelamin	Alamat	Kota	Tgl_masuk	Kode_jabatan	Kode_area
12346	Udin	22/01/1978	P	Jl.Masjid 47	Sleman	02/02/1999	02	G1
12347	Arum Dian	14/03/1980	W	Jl.Sawo 108	Yogya	02/02/1999	01	G2
12348	Sueb	04/07/1971	P	Jl.Astina 4A	Yogya	02/02/1999	02	G1
12349	Bagus	13/05/1969	P	Jl.Karangwaru 3	Magelang	02/02/1999	04	G1
12350	Intan	01/02/1975	W	Jl.Karyacita 9	Bantul	02/02/1999	05	G3

## Tabel Area

Kode_area	Nama_area	Alamat_area
G1	Gedung 1 Pusat	Jl. Dipatiukur 35
G2	Gedung 2	Jl. Ir.H.Juanda 100
G3	Gedung 3	Jl. Ir.H.Juanda 21
G4	Gedung 4	Jl. Siliwangi 50

## Tabel Jabatan

Kode_jabatan	Nama_jabatan	Gaji_pokok
01	EDP	1500000
02	Pemasaran	1200000
03	Produksi	2000000
04	SDM	2500000
05	Akunting	1200000

# Ekivalensi ekspresi operasi Seleksi

- Aturan:

- Jalankan operasi seleksi seawal mungkin (prioritaskan operasi seleksi)

contoh:  $\pi_{\text{Nama}}(\sigma_{\text{nama\_jabatan}='Akunting' \wedge \text{gaji\_pokok} < 200000}(\text{pegawai} \bowtie \text{jabatan}))$

- 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:

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

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

# Aturan Ekivalensi

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

$$\sigma_{P_1 \wedge P_2}(E) = \sigma_{P_1}(\sigma_{P_2}(E))$$

2. Operasi seleksi bersifat komutatif

$$\sigma_{P_1}(\sigma_{P_2}(E)) = \sigma_{P_2}(\sigma_{P_1}(E))$$

3. Hanya operasi final dalam sekuen operasi proyeksi yang diperlukan

$$\pi_{L_1}(\pi_{L_2}(E)) = \pi_{L_1}(E)$$

4. Seleksi dapat dikombinasikan dengan cartesian product dan theta join

$$\sigma_{\theta}(E_1 \times E_2) = E_1 \bowtie_{\theta} E_2$$

# 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 Ekivalensi

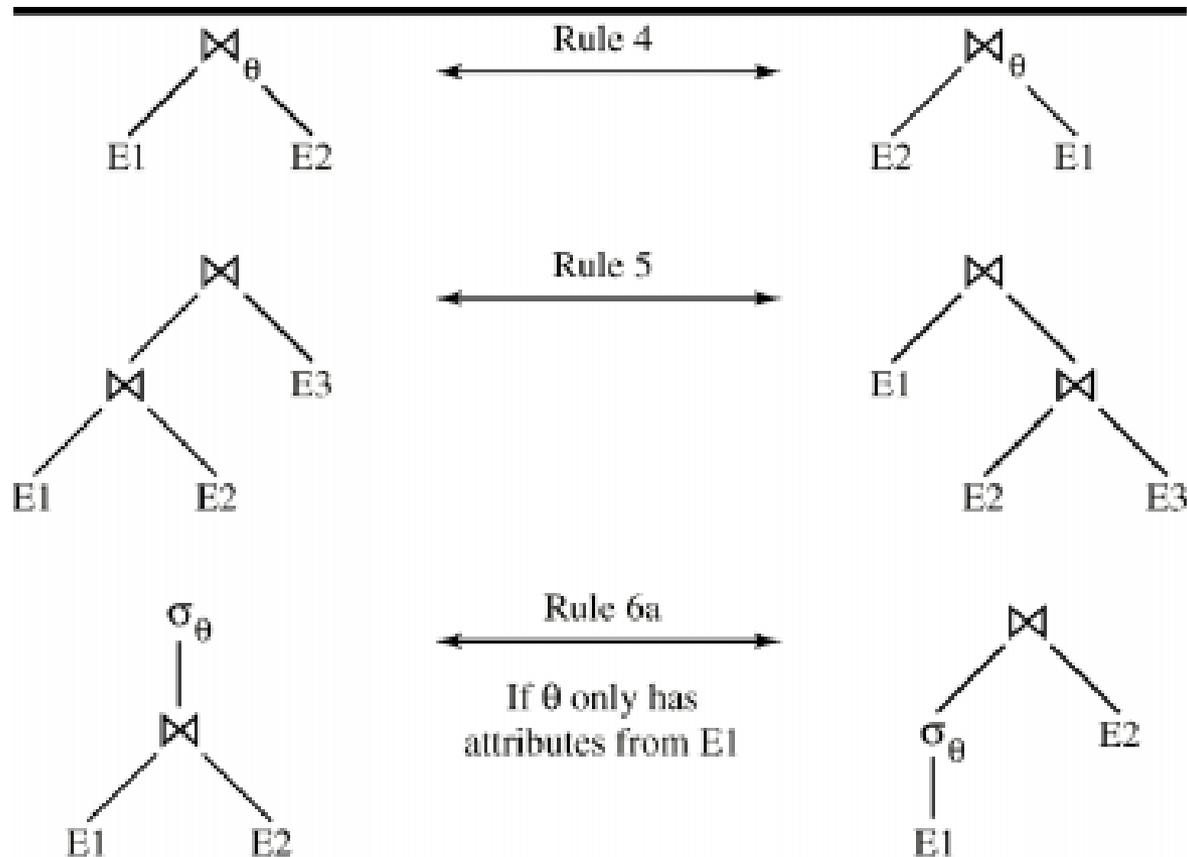
9. Operasi seleksi dapat didistribusikan ke operasi union, intersection, dan set difference

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

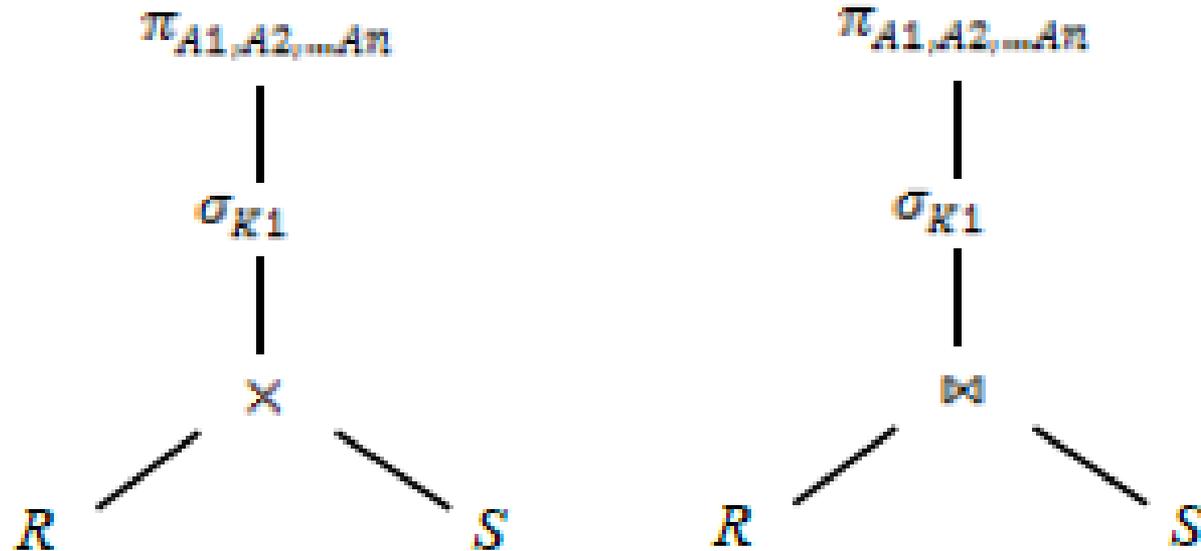
10. Operasi proyeksi dapat didistribusikan ke operasi union

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

# Struktur tree dari operasi query



# Operasi cartesian product dan join



# Ekuivalensi aljabar relasional dan SQL

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

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

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

## Tabel Pegawai

NIP	Nama	Tgl_lahir	Jenis_kelamin	Alamat	Kota	Tgl_masuk	Kode_jabatan	Kode_area
12346	Udin	22/01/1978	P	Jl.Masjid 47	Sleman	02/02/1999	02	G1
12347	Arum Dian	14/03/1980	W	Jl.Sawo 108	Yogya	02/02/1999	01	G2
12348	Sueb	04/07/1971	P	Jl.Astina 4A	Yogya	02/02/1999	02	G1
12349	Bagus	13/05/1969	P	Jl.Karangwaru 3	Magelang	02/02/1999	04	G1
12350	Intan	01/02/1975	W	Jl.Karyacita 9	Bantul	02/02/1999	05	G3

## Tabel Area

Kode_area	Nama_area	Alamat_area
G1	Gedung 1 Pusat	Jl. Dipatiukur 35
G2	Gedung 2	Jl. Ir.H.Juanda 100
G3	Gedung 3	Jl. Ir.H.Juanda 21
G4	Gedung 4	Jl. Siliwangi 50

## Tabel Jabatan

Kode_jabatan	Nama_jabatan	Gaji_pokok
01	EDP	1500000
02	Pemasaran	1200000
03	Produksi	2000000
04	SDM	2500000
05	Akunting	1200000

# Contoh

<b>Nama Tabel</b>	<b>Jumlah Kolom</b>	<b>Jumlah data</b>
Pegawai	9	5
Area	3	4
Jabatan	3	5

- Tampilkan NIP, Nama, dan tanggal masuk dari pegawai yang berada di area Gedung 3

# Latihan

1. Tampilkan NIP, Nama, tgl masuk dari pegawai pria yang memiliki jabatan Pemasaran
2. Tampilkan Nip, Nama, Alamat, jabatan, dan area kerja, dari pegawai yang jenis kelaminnya wanita
3. Tampilkan NIP, Nama, Area kerja, dan jabatan dari pegawai yang memiliki Gaji pokok lebih dari 1 juta

# Tugas Kelompok

- Membangun Database untuk sistem informasi suatu organisasi dari dokumen manual  
Sistem informasi:  
**Penjualan, rental/penyewaan, perpustakaan, sekolah, kepegawaian, apotek, inventory, restaurant, laundry, kosan**
- 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 beserta struktur tree dari query masing2 (5 soal)
- SQL: Insert, update, delete dan select (dari 1 tabel, 2 tabel dsb) (5 soal)
- Dikumpulkan minggu depan dalam bentuk laporan dan presentasi

- Urutan Isi makalah:
  - Cover judul sistem yg akan dibangun
  - Scan dokumen manual
  - Normalisasi (jelaskan setiap tahapan sampai hasilnya)
  - ERD, skema relasi, struktur tabel
  - Implementasi database (create db, tabel, insert, update, delete)
  - 5 Pertanyaan/ soal query
  - Jawaban SQL, Aljabar Relasional, dan tree nya lengkap