

OPTIMASI QUERY

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

Teknik Informatika – UNIKOM BD2 [2020]



Konsep 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
- **Concurrecy controller**, yang menjamin interaksi pada basis data secara konkuren dilaksanakan tanpa adanya konflik antar user

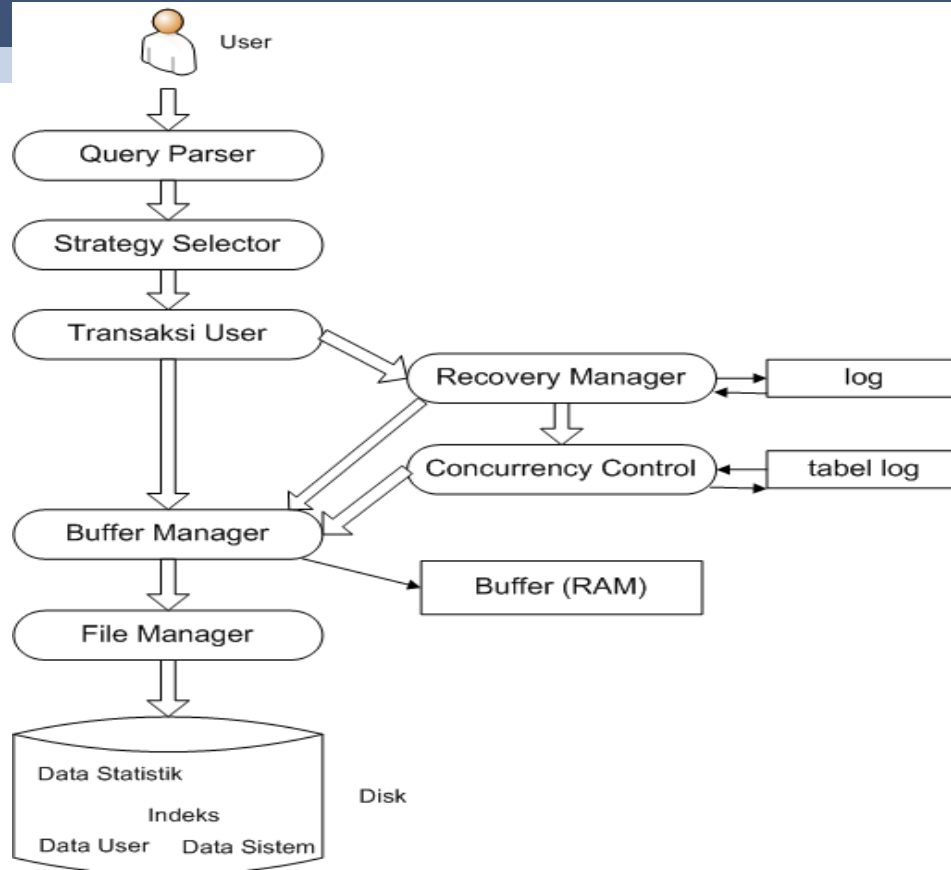


Struktur data yang dibutuhkan dalam implementasi fisik

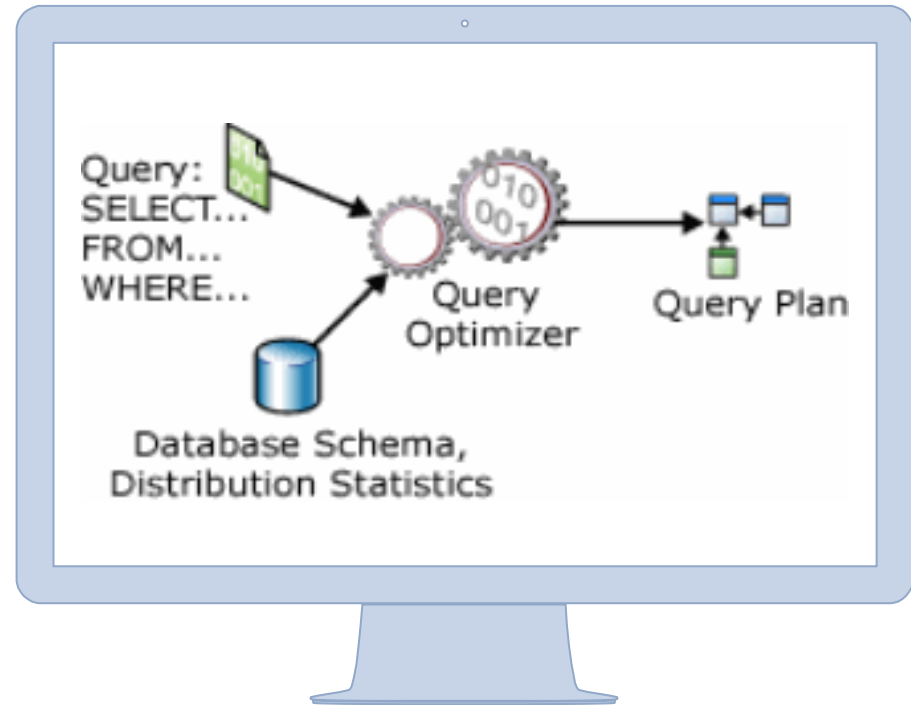
- **File Data**, yang merupakan basis data itu sendiri
- **File Data Sistem**, yang menyimpan informasi tentang struktur basis data, contoh isi file data sistem adalah kamus data
- **Data Statistik**, yang menyimpan informasi spesifik tentang data dalam basis data. Informasi ini bermanfaat bagi pemilihan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



Struktur DBMS untuk Pemrosesan Query



ESTIMASI QUERY





ESTIMASI QUERY

- Query Optimizer 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 $V(A,r) = n_r$

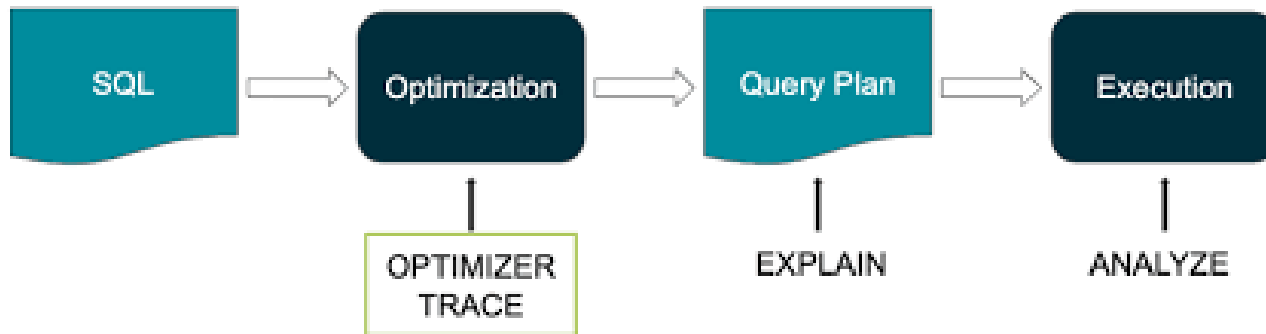


Pengukuran Biaya QUERY

- 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.



GAMBARAN KONSEP OPTIMASI QUERY





EKIVALENSI EKSPRESI RELASIONAL

Contoh:

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

1. $\pi_{\text{Nama_jabatan, Gaji_pokok}}(\sigma_{\text{Gaji_pokok} < 2000000}(\text{jabatan}))$
2. $\sigma_{\text{Gaji_pokok} < 2000000}(\pi_{\text{Nama_jabatan, Gaji_pokok}}(\text{jabatan}))$

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

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



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

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

2. Operasi seleksi bersifat komutatif

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

3. Hanya operasi final dalam sekuen operasi proyeksi yang diperlukan

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

4. Operasi theta join bersifat komutatif

$$E_1 \bowtie_{\theta} E_2 = E_2 \bowtie_{\theta} E_1$$



ATURAN EKIVALENSI

5. Operasi natural join berifat asosiatif
$$(E1 \bowtie E2) \bowtie E3 = E1 \bowtie (E2 \bowtie E3)$$
6. Seleksi dapat dikombinasikan dengan cartesian product dan theta join
$$\sigma_{\theta}(E1 \times E2) = E1 \bowtie_{\theta} E2$$
7. Operasi union dan intersection bersifat komutatif
$$E1 \cup E2 = E2 \cup E1, E1 \cap E2 = E2 \cap E1$$
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

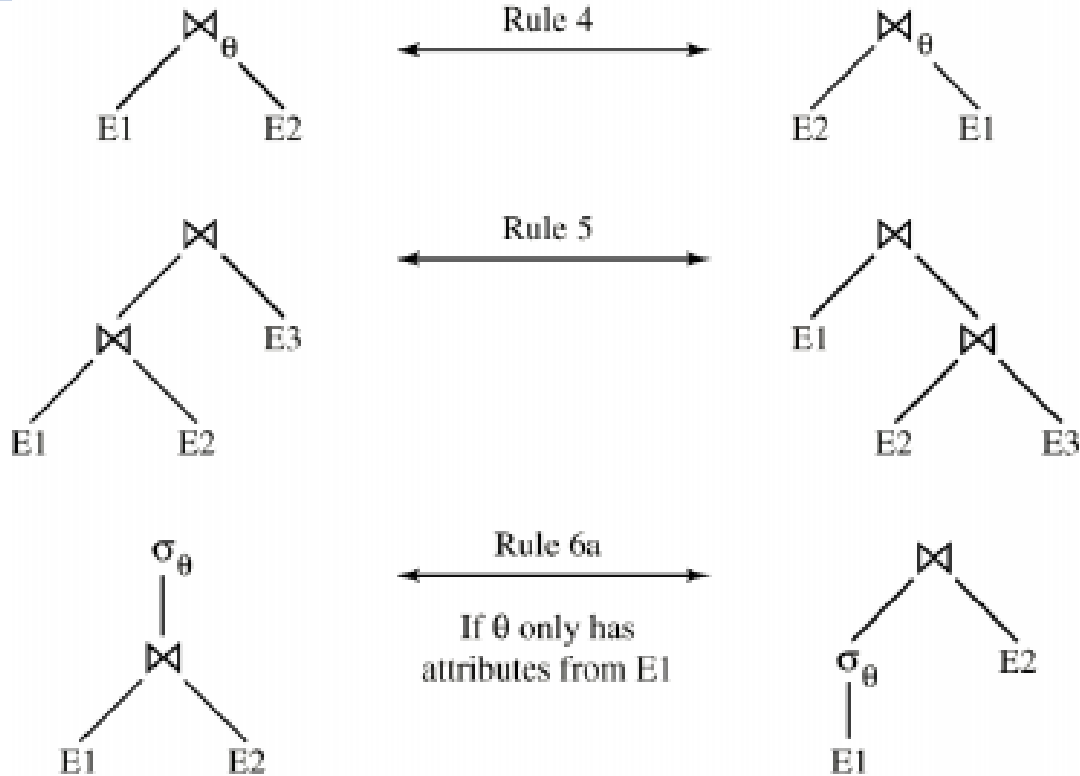
$$\sigma_P(E1 - E2) = \sigma_P(E1) - E2 = \sigma_P(E1) - \sigma_P(E2)$$

10. Operasi proyeksi dapat didistribusikan ke operasi union

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

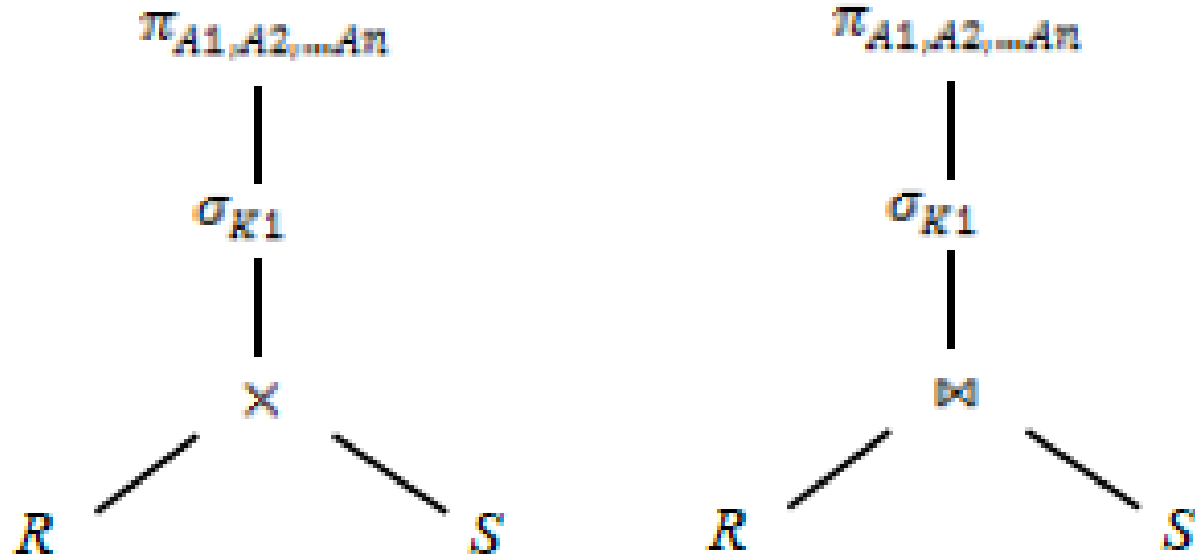


Struktur Tree Query





Struktur Tree Operasi Cartesian Product dan Join





EKIVALENSI ALJABAR RELASIONAL DAN SQL

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



EKIVALENSI ALJABAR RELASIONAL DAN SQL

Operasi	Aljabar Relasional	SQL
Union	$E1 \cup E2$ Contoh: $Pegawai \cup \text{HistoryPeg}$	<code>select * from E1 union select * from E2</code> Contoh: <code>Select * from pegawai union select * from historypeg</code>
Set Difference	$E1 - E2$ Contoh: $\pi_{\text{NIP}}(\text{Pegawai}) - \pi_{\text{NIP}}(\text{HistoryPeg})$	<code>Select * from E1 except select * from E2</code> Contoh: <code>Select NIP from pegawai except select NIP from historypeg</code>
Cartesian Product	$E1 \times E2$ Contoh: $\pi_{\text{NIP, Nama, Gaji_pokok}}(\sigma_{\text{Pegawai.Kode_jabatan}=\text{Jabatan.Kode_jabatan}}(\text{Pegawai} \times \text{Jabatan}))$	<code>Select * from E1, E2</code> Contoh: <code>Select Pegawai.NIP, Pegawai>Nama, Jabatan.Gaji_pokok from Pegawai, Jabatan where Pegawai.Kode_jabatan=Jabatan.Kode_jabatan;</code>



EKIVALENSI ALJABAR RELASIONAL DAN SQL

Operasi	Aljabar Relasional	SQL
Set Intersection	$E1 \cap E2$ Contoh: Pegawai \cap HistoryPeg	<code>select * from E1 intersect select * from E2</code> Contoh: <code>Select * from pegawai intersect select * from historypeg</code>
Join	$E1 \bowtie E2$ Contoh: Pegawai \bowtie Pegawai.Kode_area=Area.Kode_area Area	<code>Select * from E1 join E2</code> Contoh: <code>Select * from Pegawai join Area on Pegawai.Kode_area=Area.Kode_area;</code>



LATIHAN

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



LATIHAN- CONTOH

Nama Tabel	Jumlah Kolom	Jumlah data
Pegawai	9	5
Area	3	4
Jabatan	3	5

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



LATIHAN SOAL

Tuliskan SQL, aljabar relasional, gambarkan tree, dan hitung estimasinya:

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



LATIHAN SOAL

Latihan dikerjakan dengan ditulis di kertas, tulis juga Nim, Nama, Kelas, kemudian discan/ difoto

Latihan dikirimkan ke email: gentisya.tri.mardiani@email.unikom.ac.id

Subject: Latihan BD2-Kelas-NIM

Nama file: Lat2-Kelas-NIM-Nama

Latihan dikirimkan maksimal hari ini (sampai jam 24.00)



TUGAS KELOMPOK

Membangun Database untuk sistem informasi suatu organisasi dari dokumen manual

Tema:

Penjualan, rental/penyewaan, perpustakaan, sekolah, transportasi, apotek, inventory, restaurant, laundry, kosan

Ketentuan:

1. Scan dokumen manual
2. Lakukan Normalisasi (1, 2, 3 ..dst), Buat ERD, Skema relasi, dan struktur tabel
3. Create Database dan tabel (tabel harus sama berdasarkan hasil no.2) (type data disesuaikan dengan kebutuhan)
4. Syarat: Database minimal terdiri dari 3 tabel. Setiap tahap di-screenshot. Lakukan contoh insert, update, dan delete.
5. Buatlah minimal 5 soal Query (dari 1 tabel, 2 tabel, dsb dengan berbagai operasi), kemudian jawablah soal:
6. Tuliskan SQL, Screenshot hasil, Aljabar relasional, serta struktur tree dari setiap soal



TUGAS KELOMPOK

- Tugas dikirimkan ke email maksimal sampai tanggal 21 April 2020, sesuai jam awal kuliah (karena akan dijadikan konfirmasi kehadiran, jika telat/ tidak mengumpulkan berarti tidak hadir)
- Subject: Tugas kelompok BD2-Kelas-NIM
- Nama File: Kelas-BD2-Tema kelompok.docx
- NIM merupakan NIM salah satu anggota yang mengirim
- Tema kelompok: silakan atur dengan ketua Kelasnya (tidak boleh ada tema yang sama)

Kuliah Online:

BD2 [2020]

