

## 1. A. Kelebihan dan kekurangan data base system

Kelebihan dari database system:

### - Security yang lebih baik

\*database ini securitynya lebih bagus dibandingkan dari file based system karena dalam DBMS, hanya si pengguna yang bisa memberikan izin ke seseorang dalam menggunakan database mereka, dan perlu adanya otorisasi menggunakan username dan password. DBMS juga unggul di sini karena mereka menyediakan berbagai macam keamanan untuk privasi data sehingga bisa melindungi sebuah data penting yang tidak bisa sembarangan dibuka oleh orang lain(*contoh: database sebuah rahasia perusahaan yang memerlukan tingkat security yang kuat supaya tidak terbobol oleh orang asing/ hacker yang ingin mengambil data tersebut agar nantinya bisa dibocorkan ke pihak lain. Caranya bisa dengan memperketat security mereka, merubah passwordnya setiap 1 bln sekali agar membuat hacker susah untuk meretas dan tidak akan meretas lagi karena sudah susah.*)

### - Backup yang mudah

\*backup data merupakan sebuah tindakan yang penting dalam sebuah sistem. bayangkan saja seperti backup windows, apabila windows kita bermasalah dan tanpa adanya backup file penting, maka file kita akan terhapus selamanya jika tidak di backup, sama seperti DBMS yang menyimpan banyak ribuan data pengguna atau data pasokan dari perusahaan yang perlu dibackup supaya tidak kewalahan. Nah untuk DBMS ini mereka menyediakan fitur auto backup sehingga tidak perlu backup secara manual oleh si user/ admin yang memegang data tersebut. Lalu tidak hanya itu DBMS juga bisa recovery ke state sebelumnya sehingga bisa diakses kembali.(*contohnya: ada sebuah perusahaan ingin backup data, mereka tinggal membackup , dan data tersebut akan kesimpan otomatis, lalu misal sistem mereka ada yang rusak karena gagal update/ atau kelalalaian si user , mereka akan kembali ke state sebelumnya supaya bisa digunakan kembali(checkpoint).*)

### - Data Integrity

\*Integrity berarti data yang akurat, tidak outdated, tidak palsu dan bisa diakses. Dalam DBMS ini sangat penting karena apabila tidak terintegrity maka tidak bisa dilihat oleh pengguna. Makanya perlu data yang konsisten agar DBMS bisa bekerja dengan baik.(*contoh: ada pengguna yang login ke shoppe, nah ketika si pengguna*

*ingin melihat data mereka secara tepat,atau ingin merubah data mereka , mereka bisa ke my accoutn masing-masing untuk melihat data mereka dan melakukann customisasi ke akun mereka.)*

Kekurangan DBMS:

**- Perawatan yang mahal**

Tak bisa dipungkiri bahwa sistem data perlu biaya yang mahal untuk memelihara DBMS agar terawat dengan baik, tidak hanya dalam segi perangkat keras, DBMS perlu dirawat oleh orang yang ahli dalam maintenance sistem, mengelola sistem, dan data yang disimpan dalam jumlah besar. *(Contoh: bandingkan komponen DBMS menggunakan perangkat generasi lama, dengan gen baru tentunya harus menggunakan komponen yang up to date supaya tidak lelet dalam mengakses data, dan sistem lama lebih perlu perhatian ekstra dibandingkan new system karena bisa aja sistem tersbeut fail dalam menyimpan data atau mengakses data. Dan perlu orang yang benar2 ahli bukan asal asalan untuk melakukan hal demikian supaya tidak ada kejadian yang tidak diinginkan, atau merugikan semua data yang ada di DBMS terutama data dari user.)*

**- Sensitif**

\*maksudnya adalah DBMS rentan terhadap kegagalan karena kesalahan tertentu. DBMS ini berbasis data sehingga bisa memberhentikan 1 sistem secara menyeluruh apabila ada 1 komponen yang rusak.*(contoh: misalkan ada user /admin yang sedang membuat code , nah ada 1 error di bagian code maka tidak bisa berjalan dengan baik, dan lebih parahnya ketika sedang membuat table, apabila kita membuat kesalahan di langkah berikutnya kita perlu buat ulang dari awal supaya bisa berjalan dengan baik. Perlu ketelitian dalam menggunakan DBMS.)*

**- Menggunakan size yang besar**

\*DBMS sebagai aplikasi/sistem perangkat lunak mempunyai file base yang besar, bahkan bisa membutuhkan gigabyte disk space untuk semua fitur,atau file update, makanya harus mempersiapkan banyak storage hehe.*(contoh: apabila ada system maintanance dadakkan dan perlu adanya update, namun lupa nih tidak melihat storage maka akan terjadi hang atau error karena storagenya tidak cukup.salah satu untuk menghindari hal tersebut ya perlu beli storage baru.)*

**B. Komponen DBMS**

- Programmers: yang pasti mereka yang membuat kodingnya agar bisa diakses oleh pengguna lain atau untuk bawahan seperti manager.
- User: adalah pengguna dari sebuah database. Mereka hanyalah orang awam yang tidak mengerti apa itu database, namun mempunyai kontribusi besar dalam circle database karena user bisa membantu memberikan saran atau keluhnya kepada komponen DBMS lainnya agar bisa meningkatkan DBMS mereka untuk lebih baik kedepannya.
- DBA: berfungsi untuk mengimplementasikan ide physical DBMSnya mau seperti apa dan membantu programmer agar lebih mempunyai visi yang baik supaya bisa tercapai sesuai harapan.
- Database Manager: database manager menyediakan sebuah UI low lever pada setiap basis data dengan aplikasi dan pada query yang ada di sistem. Bertugas untuk pengujian query dan merekam setiap data yang ada. *(Contoh: menyiapkan data yang mudah dimengerti dan mengatur data yang ada.)*
- Query processor:  
Bertugas untuk mentranslate perintah menjadi instruksi low-level yang dapat dimengerti oleh database manager.
- DDL compiler  
Tugasnya untuk mengubah perintah yang mempunyai bahasa yang tinggi ke bahasa yang tingkat rendah
- Dictionary manager  
\* mengatur , mengakses perpusnya DBMS

2. A. **Views** adalah sebuah komponen dalam Database yang memungkinkan user untuk melihat data mereka dalam bentuk table. Tablenya ini bersifat virtual dan dikombinasikan dengan dipanggilnya menggunakannya SELECT. Tujuannya adalah untuk mempermudah pengguna untuk melihat data(terutama dalam query)

#### **Cara untuk menampilkannya adalah**

Pertama tama dalam database kita buat database terlebih dahulu,

Create database {namabebas}

Lalu di use {namabebas}

Kemudian langkah selanjutnya adalah membuat viewnya caranya dengan CREATE VIEW {namaviewbebas} AS SELECT \* FROM {namatabelbebas};. nah kalau ada

data dokter dibawah kita tinggal ketik create VIEW VIEW1 AS SELECT \* FROM MsDoctor;

```
use UTSGenap
create view view1 as select * from MsDoctor;

select * from view1;
```

Maka nanti hasilnya akan seperti ini:

Results		Messages				
	DoctorID	DoctorName	DoctorAddress	DoctorPhone	DoctorBirthDate	
1	DC001	Yogie Soesanto	Binus Street	+62216600123	1985-11-24	
2	DC002	Nico Natalie	Angrek Street	+628181231345	1992-12-15	
3	DC003	Julian Wesley	Syahdan Street	+628181212	1964-03-07	
4	DC004	Mellisa Pratiwi	Bowl Street	+6221112331	1994-08-08	
5	DC005	Indra Saswita	Sunter Street	+62216600127	1962-09-23	
6	DC006	Franky Quo	Crox Street	+62816609123	1975-10-01	
7	DC007	Livia Ashianti	Syahdan Street	+62816699111	1989-04-03	
8	DC008	Yudy Purnama	Brother Street	+6221316623	1965-05-22	
9	DC009	Budiman	Kijang Street	+6289991221	1954-02-02	
10	DC010	Andy Putra	Angry Street Street	+6287877991	1987-12-06	

Query executed successfully. LAPTOP-ADU8799N\SQLEXPRESS ... LAPTOP-ADU8799N\Darren... UTSGenap 00:00:00 10 rows

### C. Kelemahan dari views

- table depedencies: maksudnya adalah jika di data kita ada tabel yang berubah maka kita harus mengubah nama viewnya agar bisa berjalan dengan baik.(contoh: saya ingin menambahkan data baru(disitu tertulis hanya MsDoctor, saya mau tambahkan {MsDoctor, MobilKeren} saya harus menghapus view yang saya sudah buat supaya tidak ada error,SQL tidak bisa menerima 2 view dengan nama yang sama sehingga ya harus buat barus demi tidak error.)

- performa: cenderung lelet apabila viewnya dibuat secara sekaligus. (Contohnya: saya punya 15view yang mempunyai data yang berbeda dan isinya banyak, maka kalau saya execute secara langsung perlu waktu untuk menjalankan ke 15 view tersbebut, dan belum lagi karena perangkat saya yg belum mendukung multithreaded sehingga bisa memakan waktu lebih lama lagi untuk membuat tabelnya.)

A. Grant dalam SQL adalah sebuah perintah dimana memberikan privileges kepada objek2 di database ke ppihak tertentu yang ingin mengakses database. Hak yang bisa kita kasih bisa berupa INSERT,SELECT,DELETE, REVOKE, CREATE, dan lain lain. Lalu untuk GRANT OPTION yang dimaksud adalah kita bisa memberikan akses user1 ke user lainnya secara luas .(kayak global gitu) .Contohnya ada di bagian B.

3. *Misal: saya mempunyai data yang ingin saya bagi(saya punya teman a,b,c,d,dan e) namun saya kurang percaya terhadap B, dan , C. nah disitu saya bisa mengrestric penggunaannya terhadap B, dan c melainkan hanya A,D, dan e saja.)*

#### **B. Contoh penggunaa dasar SQL dalam grant sebagai berikut:**

Saya masih menggunakan database {UTSGenap}

Pertama tama kita harus buat credentialnya terlebih dahulu yaitu CREATE login user  
Caranya adalah CREATE LOGIN {namauser} WITH PASSWORD = '{namauser}';  
CREATE USER [namauser3] for LOGIN {namauser3}; untuk lebih jelasnya lihat gambar dibawah, setelah itu kita perlu use [database] nya agar di set ke UTSGenap.  
Lalu kita tinggal klik run dan hasilnya command completed(berhasil)

```
--grant--
create login user3 with password = 'user3';
create user [user3] for login user3;

--use grant--
use [UTSGenap]
go
GRANT SELECT
ON MsMedicine
TO user3
```

```
--use grant--
use [UTSGenap]
go
GRANT SELECT
ON MsMedicine
TO user3
```

100 %

Messages

Commands completed successfully.

Completion time: 2022-03-16T09:48:21.9376497+07:00

```
--grant option--
Grant all privileges
ON MsDoctor
TO user3 WITH GRANT OPTION;
```

100 %

Messages

The ALL permission is deprecated and maintained only for compatibility. It DOES NOT imply ALL permissions defined on the entity.

Completion time: 2022-03-16T09:53:42.1426657+07:00

Nah untuk semua privileges dikasih grant khusus untuk mengakses database yang ada.

#### 4. A. Where dan having

```
use UTSGenapnumber4
create table motor
(
Merk char(10) not null PRIMARY KEY,
Tipe varchar(50) not null,
CreatedDate date not null,
Descriptions varchar(50) not null,
);
--INSERT--
INSERT INTO motor
(Merk, Tipe, CreatedDate, Descriptions)
VALUES
('Honda', 'CBR1000RR', '2001-05-03', 'keren'),
('Kawasaki', 'H2', '2002-05-04', 'keren'),
('Yamaha', 'R15', '2003-06-05', 'jelek'),
('Harley', 'gatau', '2004-06-05', 'baja'),
('Suzuki', 'GSX150s', '2005-07-20', 'sombong ');
select "Merk", "Tipe" from dbo.motor
where "CreatedDate" > '2003-01-01';
```

70 %

Results Messages

	Merk	Tipe
1	Harley	gatau
2	Suzuki	GSX150s
3	Yamaha	R15

Where apabila keyfilternya ada di field yang ada di table, bukan fungsi, sedangkan having kalau ada keyfilter dan ada nominal, sayangnya di data saya tidak ada nominal harga sehingga kosong datanya.

```
select "Merk", "Tipe" from dbo.motor
where "CreatedDate" > '2003-01-01';
SELECT Merk, COUNT(*) AS "waduuhhhh"
FROM motor
WHERE CreatedDate > '2002-01-01'
GROUP BY Merk
HAVING COUNT(*) > 1;
```

C. group by adalah untuk membuat group berdasarkan data yang sudah ada kayak dibuat 1 grup agar tidak tercampur ke data lainnya,

Contohnya seperti ini:

```
SELECT Merk, COUNT(10) AS motor
FROM motor
WHERE CreatedDate > '2002-01-01'
GROUP BY Merk
HAVING COUNT(*) > 1;

SELECT Merk
FROM motor
GROUP BY Merk;
```

7 %

Results Messages

	Merk
1	Harley
2	Honda
3	Kawasaki
4	Suzuki
5	Yamaha

Kalau  
aggreg  
ats  
functi  
on  
lebih  
diteka  
nkan  
mengg  
unaka  
n  
COU  
NT,

SUM, MIN, MAX menggunakan angka nominal, namun di data yang saya buat saya tidak tambahkan data nominalnya. Namun contoh dasarnya sebagai berikut:

Bisa juga menggunakan inner JOIN :

Kalau pakai data angka tidak error seharusnya.

```
INNER JOIN Descriptions ON Merk = Create
```

77 %

Messages

Msg 208, Level 16, State 1, Line 35  
Invalid object name 'Descriptions'.

Completion time: 2022-03-16T10:51:43.4089042+07:00

```
SELECT Tipe  
FROM motor  
INNER JOIN Descriptions ON Merk = CreatedDate;
```

Sekian dari saya  
terimakasih.