

## LEARNING PROGRESS REVIEW WEEK-2



## OPTIMISTIC TEAM

**ALDIVA WIBOWO** 

LUTFIA HUMAIROSI

ASPRIZAL RIZKY

MILLENIA WINADYAP

GILANG RAHMAT R



01

# INTRODUCTION TO DATA AND DATABASE





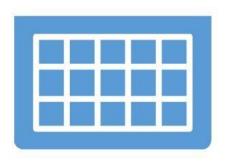
Data Data merupakan bentuk jamak dari datum, atau suatu pengukuran oleh sesorang yang kemudian bisa dipahami oleh orang yang melakukan proses pengukuran tersebut maupun orang lain. Misalnya seseorang yang suka menulis diary, kegiatan apa saja atau mengalami apa aja, nah diary ini dianggap juga sebagai data karena menuangkan informasi-informasi ke dalam diary daa tesebut adalah data yang benar terjadi.

Data adalah semua fakta apapun yang bisa dilihat kembali atau atau tercatat yang di mana seseorang bisa diakses kembali dan diterima kembali informasinya.

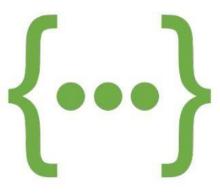








**Structured** 



**Semi-structured** 



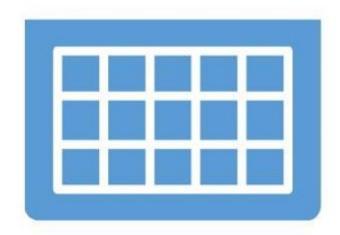




#### Structured

(data yang terstruktur)

- Ditampilkan dalam bentuk baris dan kolom, mis: excel (kita dapat membaca informasi dengan mudah)
- Tersimpan di relational database (ada hubungan data yang satu dengan yang lain)
- Memiliki struktur data yang strip (isi data akan mengikuti dengan dayta dari awal)
- Mudah untuk di manage dan di analisis, karena sudah tersusun dalam bentuk kolom dan baris maka untuk proses analisis pun akan lebih mudah.
- Contoh struktur data: excel, csv, mysql, postragesql, data warehouse.



**Structured** 

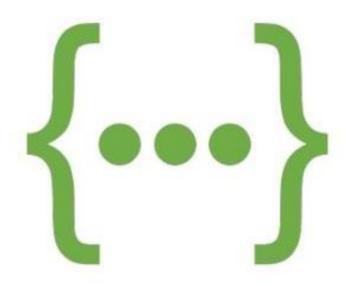




#### Semi structured

( semi struktur)

- Ciri utamanya ditampilkan dalam bentuk key-pared values
- Disimpan dalam database semi terstruktur
- Tipe data transaksional umum (untuk API)
- Fleksibel artinya kita bisa mesukan nilainya berupa type data struktur lainnya
- Data struktur yang sudah rapi dan format sendiri hanya saja perlu dilakukan 1 step lagi agar mudah dibaca.
- Contoh ;json, xml, parquet, avro



**Semi-structured** 





#### Unstructured

(tidak terstruktur sama sekali)

- Hampir semua hal yang sudah kita miliki dapat ditemukan di hampir semua kehidupan
- disimpan di folder
- itu mewakili objek tertentu
- sulit untuk dikelola, diamankan, dan diproses.
- Text, image, video, music, dibutuhkan beberapa informasi dalam file tersebut







Database database adalah media untuk menyimpan iuntuk menyimpan data atau kumpulan dta yang terorganisir sehingga data tersebut digunakan sehingga orang lain mudah mengakses sebuah data yang dikumpulkan dalam format tertentu dengan rapi di sebuah sistem komputer.







#### KENAPA MENGGUNAKAN DATABASE

Apa keuntungannya menggunakan database jika dibandingkan dengan spreadsheet seperti Excel?









#### KENAPA MENGGUNAKAN DATABASE

Spreadsheet seperti Ms. Excel mempunyai nilai limit baris maupun kolom sementara database tidak memiliki limit dan pemakaian sesuai kebutuhan

Spreadsheet seperti Ms. Excel memiliki kesulitan untuk akses beberapa file secara bersamaan sementara database tidak kesulitan dalam mengakses file secara bersamaan.







#### KENAPA MENGGUNAKAN DATABASE

Mudah dalam menyimpan data berukuran besar dan integrasi antar tabel.

Mudah untuk mengakses data secara cepat, modifikasi dan mengolah data besar dengan waktu yang lebih cepat.







Tipe database,berdasarkan fungsi dan pengggunaannya:

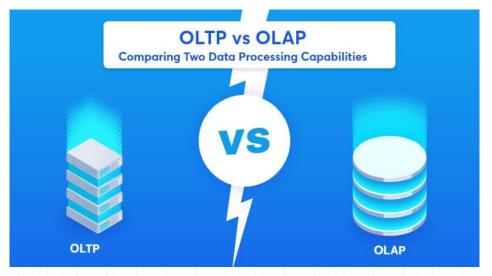
- OLTP
- OLAP

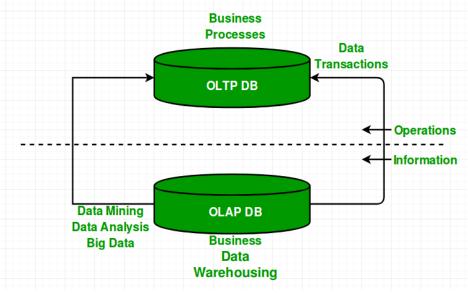
#### **OLTP** (on line transaction processing)

- Digunakan untuk data-data yang sifatnya transaksional atau yang ada terus-menerus.
- Contoh: mysql, mariadb, mongodb, postgesql.

#### **OLAP** (on line analytical processing)

- Sifatnya menganalisis ke dalam datase
- Lebih cepat untuk READ, catatan berbasis kolom, terbatas pada penggunaan internal, dan data yang didenormalisasi.
- Contoh: data warehouse, redshift (aws), bigquery (gcp),teradata.





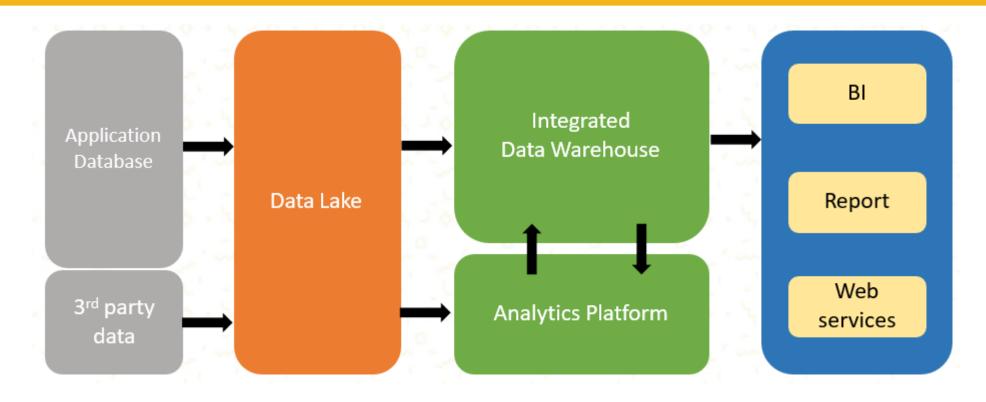
## Perbedaan OLTP dan OLAP



	OLTP	OLAP
Karakteristik	Cenderung menghandle transaksi-transaksi kecil	Menghandle data yang besar
Type kueri	Sederhana	Lebih kompleks
Operasi	Kegiatannyamenulis, menghapus atau update	Mengambil dan melakukan agregasi data/kesimpulan suatu data
Respon Waktu	Milidetik	Detik, menit, atau jam tergantung pada jumlah data yang akan diproses.
Desain	khusus industri, seperti ritel, manufaktur, atau perbankan.	khusus subjek, seperti penjualan, inventaris, atau pemasaran.
Sumber	Transaksi	Kumpulan data dari transaksi
Tujuan	Mengontrol dan menjalankan operasi bisnis penting secara real time.	merencanakan, memecahkan masalah, mendukung keputusan, menemukan wawasan yang tersembunyi.
Pembaruan data	pembaruan singkat dan cepat yang diprakarsai oleh pengguna	ata diperbarui secara berkala dengan tugas batch yang telah berjalan lama dan terjadwal.

## Arsitektur Data





Data architecture adalah proses yang terjadi dalam suatu usaha atau suatu project bekerja dengan data





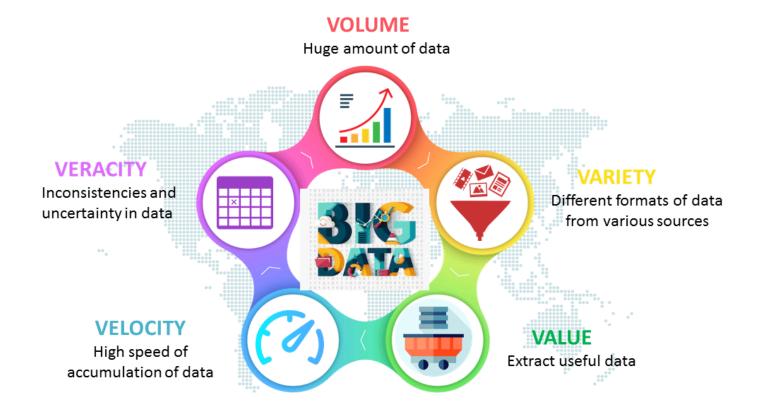


Terdapat banyak data baik dari data structur maupun data unstructure misalnya data yang berasal dari media sosial, yuotube, tiktok terdapat beberapa data gambar, video inilah yang dinamakan dunia big data. Misalnya 1 huruf = ukuran 1 byte yang di mana dalam 1 halaman terdapat dari banyaknya huruf ukuran 1 kb, lalu 1 buku ukurannya 1 mb., belum lagi dengandata laiinya maka semakin hari semakin banyak data di zaman sekarang.





5 Karakteristik Big Data







#### 5 Karakteristik Big Data

### Velocity

• Data dapat diakses dengan kecepatan yang sangat cepat sehingga dapat langsung digunakan pada detik itu juga (lebih *real time*).

#### Volume

• Mengacu pada sejumlah big data yang dihasilkan setiap detiknya

## Variety

• Memiliki karakteristik yang bermacammacam dan tidak homogen.

## Veracity

• Mengarah kepada seberapa akurat dan dapat dipercaya suatu data.

#### Value

 Memiliki nilai yang sangat tinggi apabila diolah dengan cara yang tepat guna atau dapat juga dikatakan seberapa bernilainya atau bermaknanya suatu data.



02

## BASIC SQL





#### Introduction to SQL

#### **Database**

Database merupakan tempat penyimpanan data yang tersusun dengan baik dan dapat diakses secara sistem/komputer. Didalam database terdapat skema yang berisi tables, dan didalam tables terdapat baris dan kolom.

Kelebihan database

- 1. Lebih mudah untuk menggabungkan data
- 2. Dapat berinteraksi dengan banyak device

#### SQL (Structured Query Language)

SQL merupakan bahasa yang digunakan untuk berkomunikasi dengan database, dikatakan structured karena bentuk datanya tersusun. SQL dirancang untuk pengambilan informasi tertentu dari database. Fungsi pasa SQL tidak sensitif (MaMa=MAMA=mama).

Cara kerja SQL: Masukkan query/pertanyaan, Query akan dikirim ke database, Database akan mengirim jawaban dari query tersebut.

Macam-macam SQL yang sering digunakan: MySQL, PostgreSQL, MariaDB





#### Mengapa menggunakan PostgreSQL

- 1. Penyimpanan tipe datanya bervariasi
- 2. Fungsi untuk memanipulasi data lebih lengkap
- 3. Dapat digunakan secara bersamaan
- 4. Sistem keamananya baik







**SELECT**: Digunakan untuk memanggil data. Select sebagai nama kolom yang akan dipilih, dan From sebagai nama table yang akan diplih.

#### Query:

- 1. Select semua kolom : SELECT\*FROM "TABLE NAME"
- 1. Select lebih dari 1 kolom :
  SELECT "COLUMN NAME 1","COLUMN NAME2"
  FROM "TABLE NAME"





**CONTOH**: Memanggil data regions dari datasource

#### Query

select \*from datasource.regions

	¹ãregion_id ∜‡	region_name 📆
1	1	Europe\r
2	2	Americas\r
3	3	Asia\r
4	4	Middle East and Africa\r





CREATE: digunakan untuk membuat tabel baru, harus menuliskan nama kolom dan tipe data.

#### Tipe data yang umum digunakan:

- INT : tipe data untuk bilangan bulat (+/-).
- BIGINT: tipe data untuk bilangan bulat (+/-) yang ukuran dan jangkauan lebih besar dari INT.
- VARCHAR: tipe data untuk karakter/teks, contoh ('abc', '123').
- FLOAT: tipe data untuk bilangan pecahan (+/-).
- DATE : tipe data untuk tanggal.





**CONTOH:** Membuat table dengan namakmb\_basic\_Lutfia\_Humairosi

#### Query

```
---create table
create table kmb_basic_Lutfia_Humairosi (
   id int
   , nama varchar
   , tanggal_lahir date
   , gaji float
)
```

- Emb\_basic\_linda\_kushernawati
- kmb\_basic\_louis\_madaerdo
- 🕟 🎫 kmb\_basic\_lutfia\_humairosi
- kmb\_basic\_margaretha\_gok\_asi\_naiba
- kmb\_basic\_masayu\_fany\_shapura
- kmb\_basic\_melania\_panggabean
- kmb\_basic\_millenia\_winadya\_putri





INSERT: Digunakan untuk memasukkan nilai pada table

**CONTOH:** Setelah membuat table selanjutnya mengisi data kolom

yang sudah dibuat

#### Query

# insert data to table insert into kmb\_basic\_lutfia\_humairosi values (1,'Lutfia Humairosi','2000-02-27s',300000000), (2,'Shofiyyah Fauziah','2001-03-21',20000000) select \*from kmb\_basic\_lutfia\_humairosi

<u> </u>	<sup>123</sup> id ∜‡	nama 👣 🔭	atanggal_lahir 👣	<sup>123</sup> gaji <b>T</b> ‡
1	1	Lutfia Humairosi	2000-02-27	30,000,000
2	2	Shofiyyah Fauziah	2001-03-21	20,000,000





**UPDATE**: Digunakan untuk memperbarui isi table

**CONTOH:** Memperbarui data dengan nama Shoffiyah Fauziah,

dimana gaji yang sebelumnya 20.000.000 menjadi 30.000.000

#### Query

```
update data
update kmb_basic_lutfia_humairosi
set gaji = 30000000
where nama = 'Shofiyyah Fauziah';
select *from kmb_basic_lutfia_humairosi
```

	<sup>123</sup> id ∜‡	nama 👣 🕽	anggal_lahir 🟗	123 gaji 🏋
1	1	Lutfia Humairosi	2000-02-27	30,000,000
2	2	Shofiyyah Fauziah	2001-03-21	30,000,000





ALTER: Digunakan untuk menambahkan kolom

**CONTOH**: Menambahkan kolom alamat di table

kmb\_basic\_lutfia\_humairosi

#### Query

---alter
alter table kmb\_basic\_lutfia\_humairosi add address varchar;

	<sup>123</sup> id 📆	nama T:	atanggal_lahir 🟗	<sup>123</sup> gaji <b>\(\frac{1}{4}\)</b>	address 📆
1	1	Lutfia Humairosi	2000-02-27	30,000,000	[NULL]
	-				
	1				





**DELETE:** Digunakan untuk menghapus baris pada table

**CONTOH**: Menghapus baris apabila id = 2 di table table

kmb\_basic\_lutfia\_humairosi

#### Query

---delete row
delete from kmb\_basic\_lutfia\_humairosi
where id >=2

	123 id 📆	nama 👣	atanggal_lahir 👯	<sup>123</sup> gaji <b>T</b> ‡
1	1	Lutfia Humairosi	2000-02-27	30,000,000





TRUNCATE: Ngosongkan table (Menghapus semua record/baris

yang ada didalam tabel tertentu)

**CONTOH**: Mengosongkan isi table mahasiswa

Query

truncate table mahasiswa;





**DROP**: digunakan untuk menghapus tabel dalam sebuah database

**CONTOH:** Menghapus table mahasiswa

Query

drop table mahasiswa;



03

# INTERMEDIATE SQL





LIMIT: Digunakan untuk membatasi jumlah baris yang ditampilkan.

QUERY: select column\_name from table\_name limit no\_of\_rows

**CONTOH:** Melihat table order\_details\_csv dan membatasi jumlah baris yang

ditampilkan dengan limit 5

#### Query

```
select * from sandbox.batch_kmb.order_details_csv limit 5;
```

	ABC order_id 🏋‡	123 sales 🏋‡	123 profit 🏋‡	123 quantity 🏋‡	ABC category 🏋
1	AZ-2011-1029887	85	15	2	Furniture
2	AZ-2011-1029887	26	7	2	Office Supplies
3	AZ-2011-107716	294	109	7	Technology
4	AZ-2011-1087704	76	11	3	Furniture
5	AZ-2011-1087704	252	15	5	Office Supplies





**DISTINCT**: Digunakan untuk mengeliminasi data yang berduplikasi dan hanya

mengambil karakter unik.

QUERY: select distinct column\_name from table\_name

**CONTOH:** Melihat table dan hanya mengambil karakter unik dari order\_details\_csv

#### Query

select distinct \* from sandbox.batch kmb.order details csv

	ABC order_id \(\frac{7}{3}\)	123 sales 📆 🕻	123 profit 🏋 🔭	123 quantity	T:	ABC category	ABC sub_category
1	AZ-2014-3909532	129	46		3	Technology	Accessories
2	AZ-2013-7312557	53	15		2	Office Supplies	Art
3	AZ-2012-4265765	407	29		6	Technology	Phones
4	AZ-2014-586948	1,101	319		9	Technology	Machines
5	BN-2011-4913858	9	-3		3	Office Supplies	Binders





WHERE: Digunakan untuk menentukan kondisi saat mengambil data dari suatu tabel atau beberapa tabel.

QUERY: select column\_name from table\_name where column\_name = 'condition'

**CONTOH:** Melihat table dan kondisinya hanya menampilkan Furniture dari

order\_details\_csv

#### Query

select \* from sandbox.batch\_kmb.order\_details\_csv where category = 'Furniture';

123 sales	T:	123 profit	T‡	123 quantity	T‡	ABC category	T:
	85		15		2	Furniture	
	76		11		3	Furniture	
	88		3		2	Furniture	
	541		156		4	Furniture	
	307		99		5	Furniture	
	87		-78		3	Furniture	





**ORDER BY:** Digunakan untuk mengurutkan data dalam urutan menaik atau menurun berdasarkan satu atau beberapa kolom.

QUERY: select \* from table\_name order by column\_name

CONTOH: Menampilkan data dengan urutan dari yang paling kecil ke paling besar

pada kolom profit pada order\_details\_csv

#### Query

select \* from sandbox.batch\_kmb.order\_details\_csv order by profit

123 sales 🏋 🕽	123 profit 🏋 🕻	123 quantity	7‡	RBC category 17	RBC sub_category 🏋
3,400	-3,060	1	12	Office Supplies	Appliances
3,400	-3,060	1	12	Office Supplies	Appliances
2,571	-2,211	1	11	Furniture	Chairs
2,571	-2,211	1	11	Furniture	Chairs
2,830	-1,981	1	13	Furniture	Bookcases
2,830	-1,981	1	13	Furniture	Bookcases





**GROUP BY:** Digunakan untuk mengelompokkan baris berdasarkan kolom tertentu.

QUERY: select column1, column2 from table\_name group by column1

**CONTOH:** Menampilkan data dengan mengelompokkan baris

#### Query

select \* from sandbox.batch kmb.order details csv order by profit

123 sales 🏋 🕽	123 profit 🏋 🕻	123 quantity	7‡	RBC category 17	RBC sub_category 🏋
3,400	-3,060	1	12	Office Supplies	Appliances
3,400	-3,060	1	12	Office Supplies	Appliances
2,571	-2,211	1	11	Furniture	Chairs
2,571	-2,211	1	11	Furniture	Chairs
2,830	-1,981	1	13	Furniture	Bookcases
2,830	-1,981	1	13	Furniture	Bookcases





Fungsi Agregasi, merupakan fungsi yang akan menghitung nilai yang diberikan dan menampilkan hasilnya sesuai dengan fungsi yang dipilih.

Fungsi yang biasa atau sering digunakan:

- SUM, untuk menjumlahkan semua nilai yang ada pada data yang dipilih. Syntax : SUM(expression).
- AVG, untuk menghitung nilai rata-rata dari semua nilai yang ada pada data yang dipilih. Syntax : AVG(expression).
- MIN, untuk mencari nilai paling kecil dari semua nilai yang ada pada data yang dipilih. Syntax : MIN(expression).
- MAX, untuk mencari nilai paling besar dari semua nilai yang ada pada data yang dipilih. Syntax : MAX(expression).
- COUNT, untuk menghitung jumlah baris yang terisi (bukan null) pada data yang dipilih. Syntax : COUNT(expression).





**CONTOH**: Memakai Fungsi Agregasi

#### Query

```
select sum(total_profit), avg(sales), min(quantity), max(quantity), count(quantity)
from sandbox.batch_kmb.order_details_csv
```

<u> </u>	123 sum 🏋 📜	123 avg 🔭 🔭	123 min 🏋 🔭	123 max 🏋 🔭	123 count 🏋 📜
1	3,009,204	314.6833712984	1	21	14,926





**LEAD**: Digunakan untuk membuat query lebih dari satu baris dalam tabel pada satu waktu tanpa harus menggabungkan tabel ke tabel itu sendiri atau mengembalikan nilai dari baris berikutnya dalam tabel.

QUERY: lead(expression, offset) over(partition by partition\_expression order by order\_expression)

**CONTOH**: Menggunakan fungsi lead

#### Query

select category, sales, lead(sales, 1) over(partition by category)
as previous\_sales from batch\_kmb.order\_details\_csv

	ABC category \\\ \‡	123 sales 📆 🔭	123 previous_sales 🏋 🕽
1	Furniture	76	689
2	Furniture	689	209
3	Furniture	209	40
4	Furniture	40	859
5	Furniture	859	1,235





LAG: LAG menyediakan akses ke baris pada offset fisik tertentu yang muncul sebelum baris saat ini. Gunakan fungsi analitik ini dalam pernyataan SELECT untuk membandingkan nilai di baris saat ini dengan nilai di baris sebelumnya.

QUERY: lag(expression, offset) over(partition by partition\_expression order by order\_expression)

**CONTOH**: Menggunakan fungsi lag

#### Query

select category, sales, lag(sales, 1) over(partition by category)
as next\_sales from batch\_kmb.order\_details\_csv

	ABC category \\\ \\ \\ \\ \\ \\ \	123 sales 📆	123 next_sales 🏋‡
1	Furniture	76	[NULL]
2	Furniture	689	76
3	Furniture	209	689
4	Furniture	40	209
5	Furniture	859	40



