SISTEM - BASIS - DATA

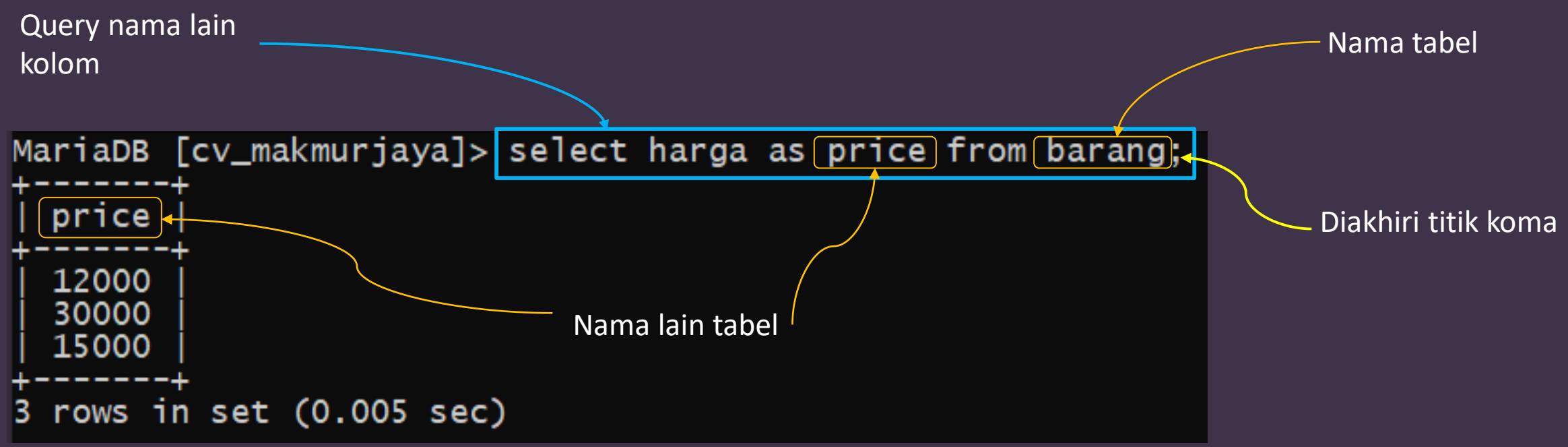
DML & Retrieve Data (Bagian 2)

My Structured Query Language

Penggunaan :: SELECT

A : Menampilkan nama lain pada kolom:

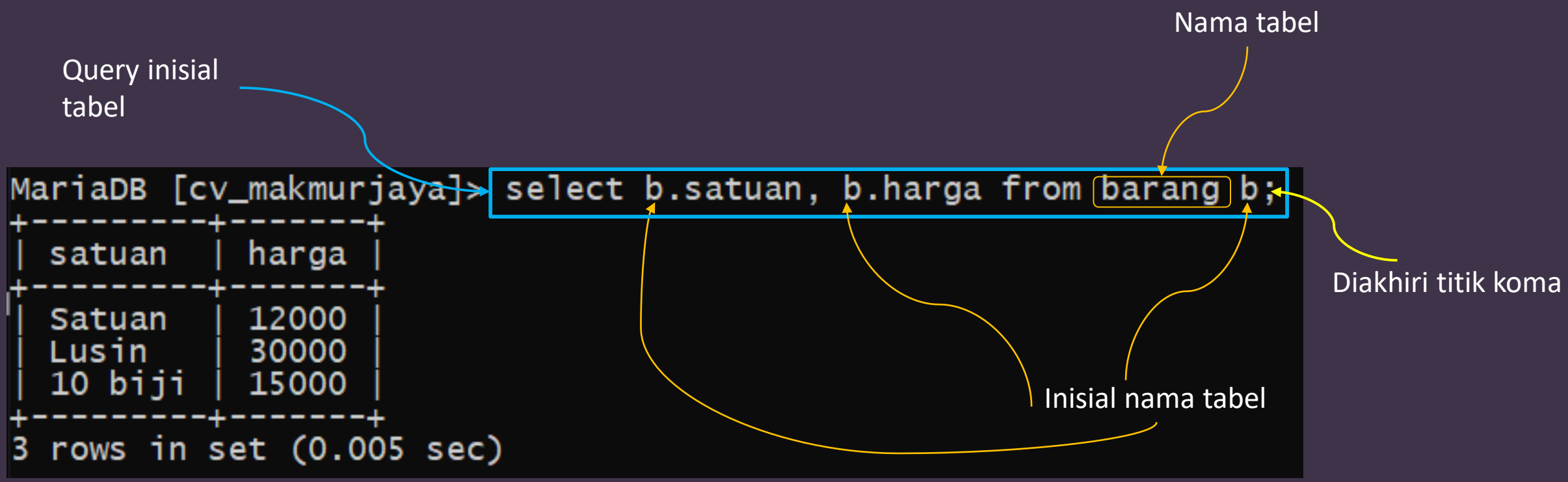
A SELECT namakolomlama AS namakolombaru FROM namatabel;



Penggunaan :: SELECT

B : Menampilkan alias untuk nama tabel:

B SELECT namalias.satuan, namalias.harga FROM namatabel namalias;



Penggunaan :: SELECT

C : Menampilkan data lebih dari dua tabel:

C SELECT * FROM namatabel1, namatabel2, namatabel-n;

Query lebih dari dua tabel

Diakhiri titik koma

```
MariaDB [cv_makmurjaya]> select * from user, barang;
+--------+----------+----------+----------------+-----------------------+----------+-------------+---------+-------+
| idUser | username | password | nama           | alamat                | KDBARANG | NAMA_BARANG | SATUAN  | HARGA |
+--------+----------+----------+----------------+-----------------------+----------+-------------+---------+-------+
|      1 | ahmad    | admin    | Ahmad Maulana  | Jl. ismail Marzuki    | A01      | Pensil      | Satuan  | 12000 |
|      2 | jayadi   | user     | Jayadi nugroho | JI. Sriwijaya mataram | A01      | Pensil      | Satuan  | 12000 |
|      1 | ahmad    | admin    | Ahmad Maulana  | Jl. ismail Marzuki    | A02      | Buku Tulis  | Lusin   | 30000 |
|      2 | jayadi   | user     | Jayadi nugroho | JI. Sriwijaya mataram | A02      | Buku Tulis  | Lusin   | 30000 |
|      1 | ahmad    | admin    | Ahmad Maulana  | Jl. ismail Marzuki    | A03      | Penggaris   | 10 biji | 15000 |
|      2 | jayadi   | user     | Jayadi nugroho | JI. Sriwijaya mataram | A03      | Penggaris   | 10 biji | 15000 |
+--------+----------+----------+----------------+-----------------------+----------+-------------+---------+-------+
6 rows in set (0.000 sec)
```

user

barang

Nested Queries / Subquery :: SELECT

D : Subquery ada (**IN**, **NOT IN**, **EXISTS**, **NOT EXISTS**)

Subquery berarti **Query** di dalam **Query**

Dengan menggunakan subquery, hasil dari query akan menjadi bagian dari query di atasnya

Subquery terletak di dalam klausa **WHERE** atau **HAVING**

Pada klausa **WHERE** : subquery digunakan untuk memilih baris-baris tertentu yang kemudian digunakan oleh query

Sedangkan pada klausa **HAVING** : subquery digunakan untuk memilih kelompok baris yang kemudian digunakan oleh query

Nested Queries / Subquery :: SELECT

D : Subquery ada (**IN**, **NOT IN**, **EXISTS**, **NOT EXISTS**)

A. Perintah untuk menampilkan data pada tabel **user** yang mana data pada kolom **idUser-nya** tercantum pada tabel **penjualan** menggunakan **IN** :

A

```
SELECT * FROM user WHERE idUser IN (SELECT idUser FROM penjualan);
```

B. perintah untuk menampilkan data pada tabel jenisfilm yang mana data pada kolom jenis-nya tidak tercantum pada tabel film menggunakan **NOT IN**:

B

```
SELECT * FROM user WHERE NOT IN (SELECT idUser FROM penjualan);
```

C. Menggunakan **EXISTS**

C

```
SELECT * FROM user WHERE EXISTS (SELECT * FROM penjualan WHERE user.idUser=penjualan.idUser);
```

D. Menggunakan **NOT EXISTS**

D

```
SELECT * FROM user WHERE NOT EXISTS (SELECT * FROM penjualan WHERE user.idUser=penjualan.idUser);
```

Contoh Data:

Nested Queries / Subquery :: SELECT

Nama tabel

```
MariaDB [cv_makmurjaya]> select*from user;
+--------+----------+----------+-----------------+-----------------------+
| idUser | username | password | nama            | alamat                |
+--------+----------+----------+-----------------+-----------------------+
|      1 | ahmad    | admin    | Ahmad Maulana   | Jl. ismail Marzuki    |
|      2 | jayadi   | user     | Jayadi nugroho  | JI. Sriwijaya mataram |
|      3 | tanwir   | 4444     | Muhammad Tanwir | Jl.Yosudarso lembar   |
+--------+----------+----------+-----------------+-----------------------+
3 rows in set (0.000 sec)
```

Nama tabel

```
MariaDB [cv_makmurjaya]> select*from penjualan;
+--------------+-------+---------------------+--------+
| id_penjualan | total | timestamp           | idUser |
+--------------+-------+---------------------+--------+
|          111 | 10000 | 2021-02-01 10:27:54 |      1 |
|          222 | 20000 | 2021-02-11 10:29:06 |      2 |
|          333 | 30000 | 2021-02-03 10:33:02 |      1 |
+--------------+-------+---------------------+--------+
3 rows in set (0.000 sec)
```

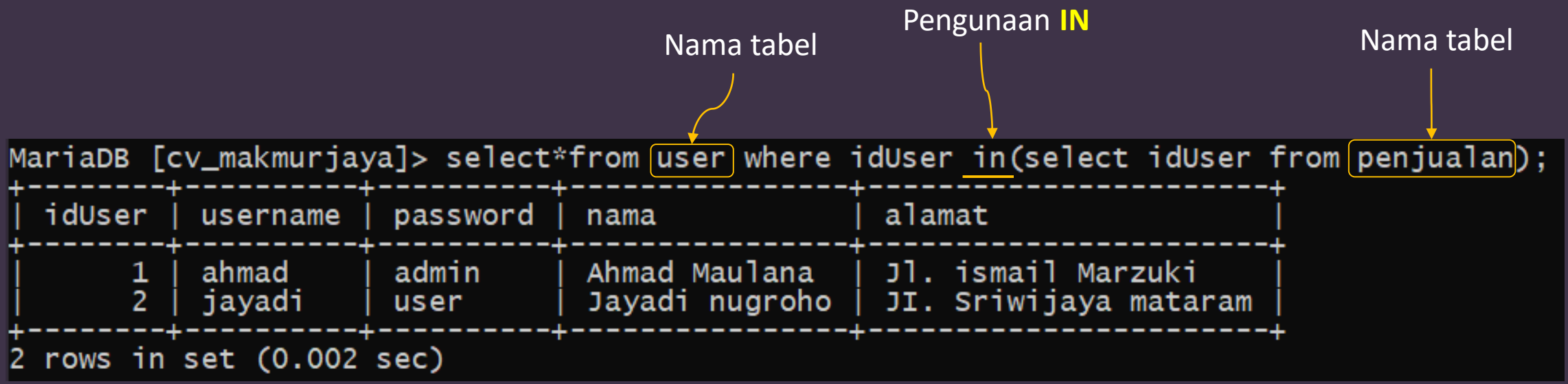
Nested Queries / Subquery :: SELECT

D : Subquery ada (**IN**, **NOT IN**, **EXISTS**, **NOT EXISTS**)

A. Perintah untuk menampilkan data pada tabel **user** yang mana data pada kolom **idUser-nya** tercantum pada tabel **penjualan** menggunakan **IN** :

A

```
SELECT * FROM user WHERE idUser IN (SELECT idUser FROM penjualan);
```



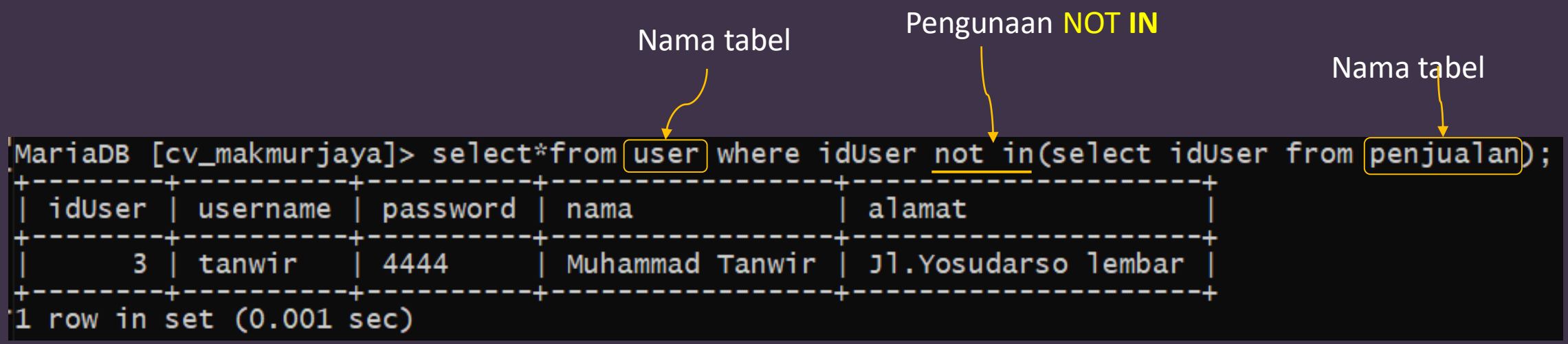
Nested Queries / Subquery :: SELECT

D : Subquery ada (**IN**, **NOT IN**, **EXISTS**, **NOT EXISTS**)

B. perintah untuk menampilkan data pada tabel **user** yang mana data pada kolom **idUser-nya** tidak tercantum pada tabel **penjualan** menggunakan **NOT IN**:

B

```
SELECT * FROM user WHERE NOT IN (SELECT idUser FROM penjualan);
```



Nested Queries / Subquery :: SELECT

D : Subquery ada (**IN**, **NOT IN**, **EXISTS**, **NOT EXISTS**)

C. Menggunakan **EXISTS**

C SELECT * FROM user WHERE EXISTS (SELECT * FROM penjualan WHERE user.idUser=penjualan.idUser);

Nama tabel

Pengunaan **EXISTS**

Nama tabel

```
MariaDB [cv_makmurjaya]> select*from user where exists(select * from penjualan where user.idUser=penjualan.idUser);
+--------+----------+----------+----------------+----------------------+
| idUser | username | password | nama           | alamat               |
+--------+----------+----------+----------------+----------------------+
|      1 | ahmad    | admin    | Ahmad Maulana  | Jl. ismail Marzuki   |
|      2 | jayadi   | user     | Jayadi nugroho | JI. Sriwijaya mataram |
+--------+----------+----------+----------------+----------------------+
2 rows in set (0.000 sec)
```

Nested Queries / Subquery :: SELECT

D : Subquery ada (**IN**, **NOT IN**, **EXISTS**, **NOT EXISTS**)

D. Menggunakan **NOT EXISTS**

D

```
SELECT * FROM user WHERE NOT EXISTS (SELECT * FROM penjualan WHERE user.idUser=penjualan.idUser);
```

Nama tabel

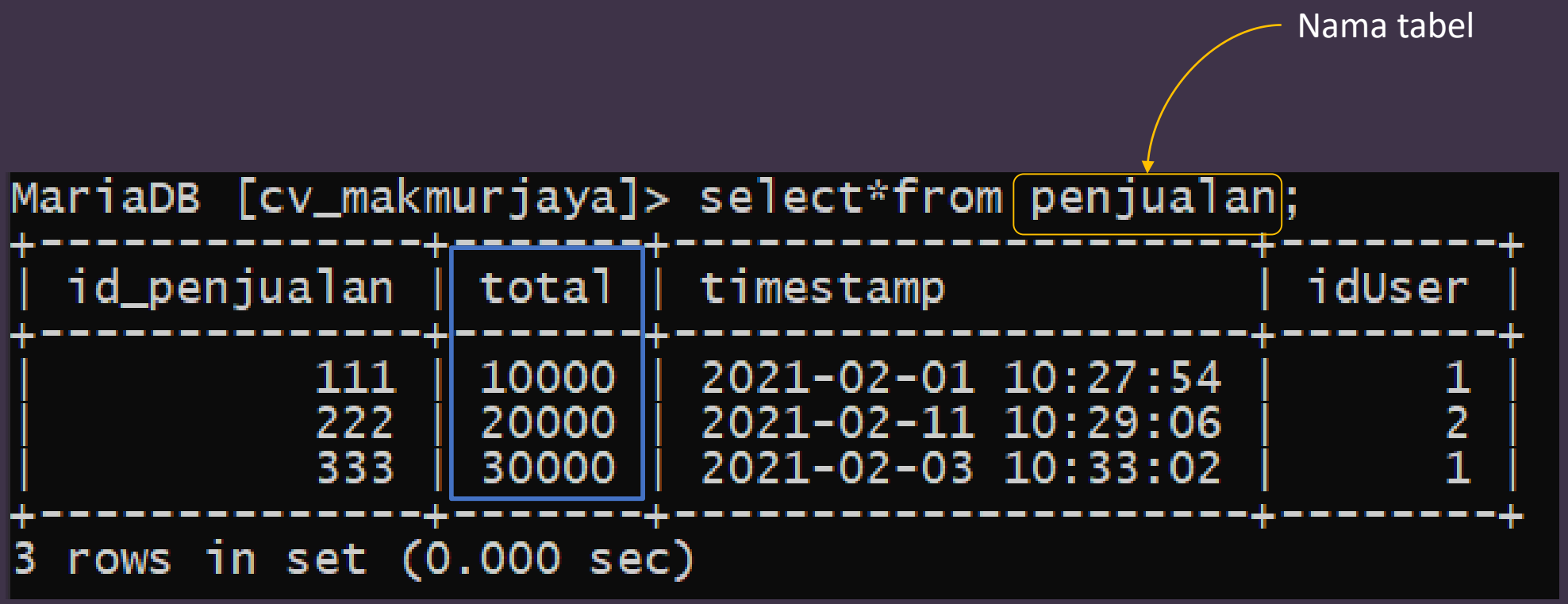
Pengunaan NOT **EXISTS**

Nama tabel

```
MariaDB [cv_makmurjaya]> select*from user where not exists(select * from penjualan where user.idUser=penjualan.idUser);
+--------+----------+----------+-----------------+---------------------+
| idUser | username | password | nama            | alamat              |
+--------+----------+----------+-----------------+---------------------+
|      3 | tanwir   | 4444     | Muhammad Tanwir | Jl.Yosudarso lembar |
+--------+----------+----------+-----------------+---------------------+
1 row in set (0.001 sec)
```

Contoh Data:

ANY & ALL :: SELECT



ANY & ALL :: SELECT

A. Operator **ANY** digunakan berkaitan dengan subquery

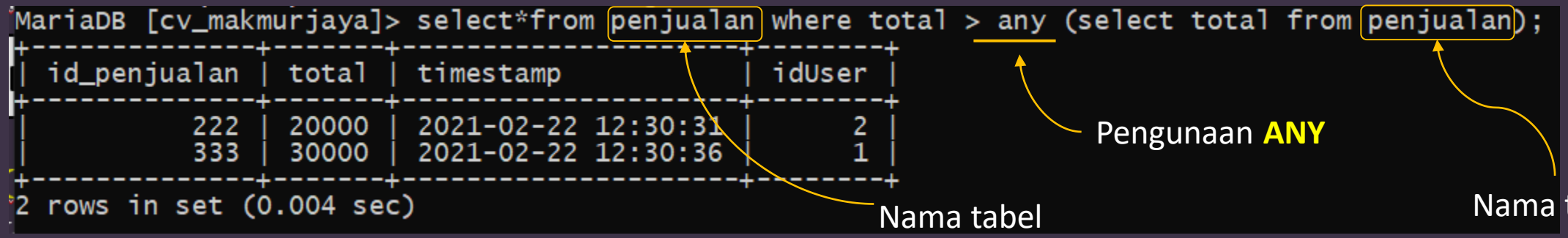
Operator ini menghasilkan **TRUE** (benar) jika paling tidak salah satu perbandingan dengan hasil subquery menghasilkan nilai **TRUE**

A SELECT * FROM penjualan WHERE total > ANY (SELECT total FROM penjualan);

Gaji > ANY (S)

Jika subquery S menghasilkan G1, G2, ..., Gn, maka kondisi di atas identik dengan:

(gaji > G1) OR (gaji > G2) OR ... OR (gaji > Gn)

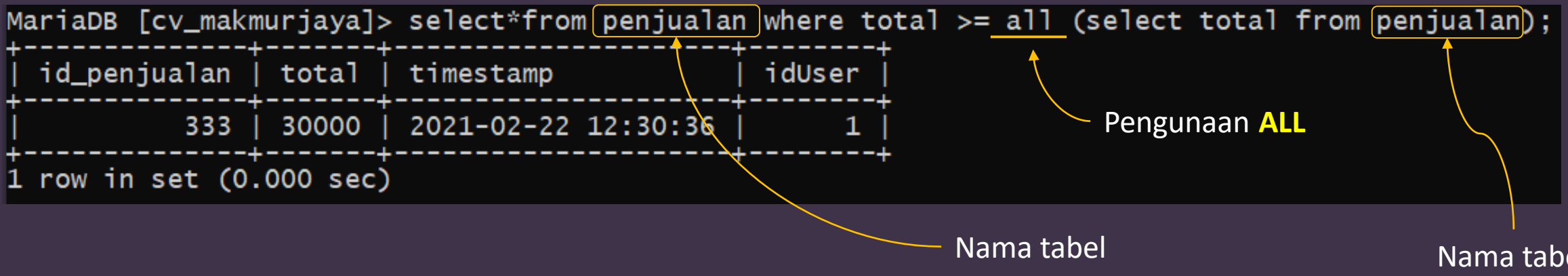


ANY & ALL :: SELECT

B. Operator **ALL** digunakan untuk melakukan perbandingan dengan subquery

Kondisi dengan ALL menghasilkan nilai **TRUE** (benar) jika subquery tidak menghasilkan apapun atau jika perbandingan menghasilkan **TRUE** untuk setiap nilai query terhadap hasil subquery

B SELECT * FROM penjualan WHERE total >= ALL (SELECT total FROM penjualan);



ORDER BY :: SELECT

Klausa **ORDER BY** digunakan untuk mengurutkan data berdasarkan kolom tertentu sesuai dengan tipe data yang dimiliki

Perintah untuk mengurutkan data barang berdasarkan kolom satuan:

A SELECT * FROM barang ORDER BY satuan;

Perintah untuk mengurutkan dengan menampilkan data secara tertentu

B SELECT * FROM barang ORDER BY satuan LIMIT 2;

Perintah untuk mengurutkan dengan menampilkan data secara ascending(Menaik)

C SELECT * FROM barang ORDER BY satuan ASC;

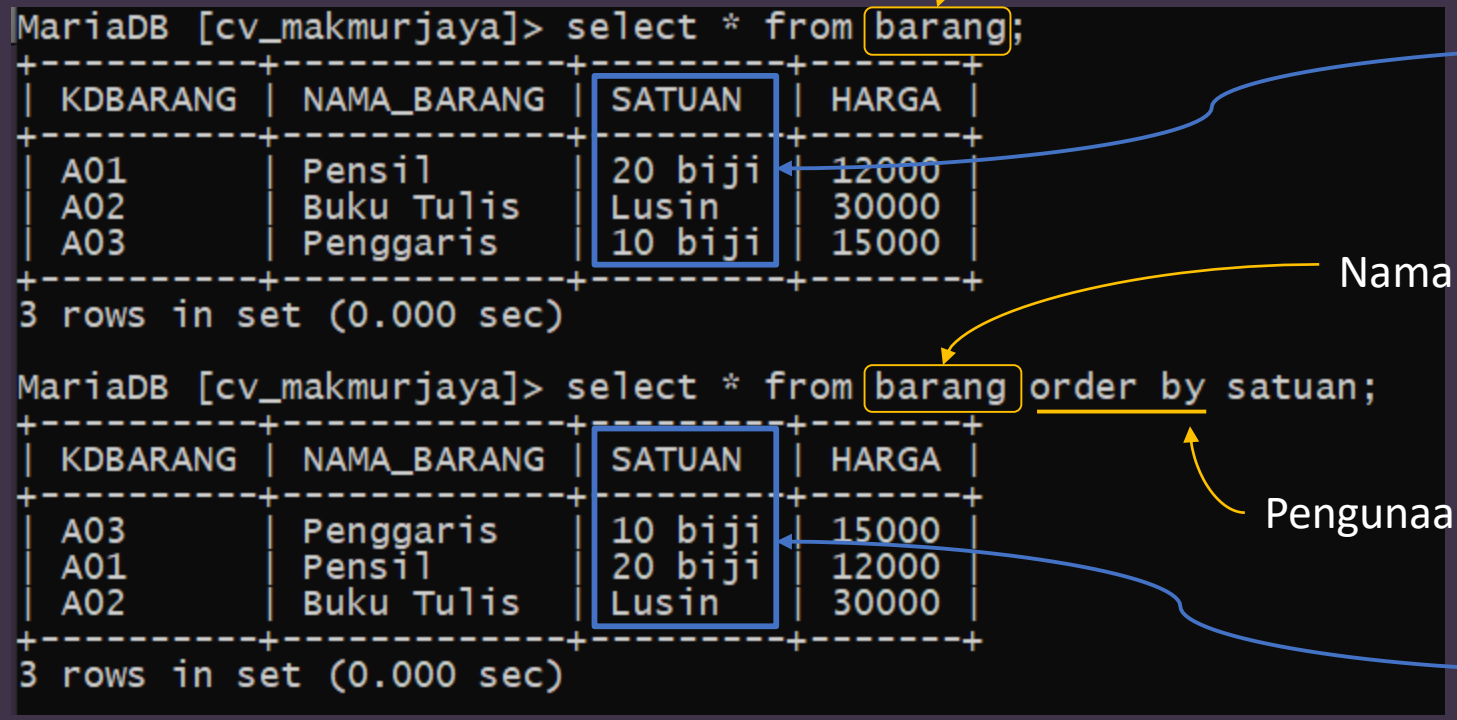
Perintah untuk mengurutkan dengan menampilkan data secara descending(Menurun)

D SELECT * FROM barang ORDER BY satuan DESC;

ORDER BY :: SELECT

Perintah untuk mengurutkan data barang berdasarkan kolom satuan

A SELECT * FROM barang ORDER BY satuan;



Nama tabel

Sebelum melakukan **order by**

Nama tabel

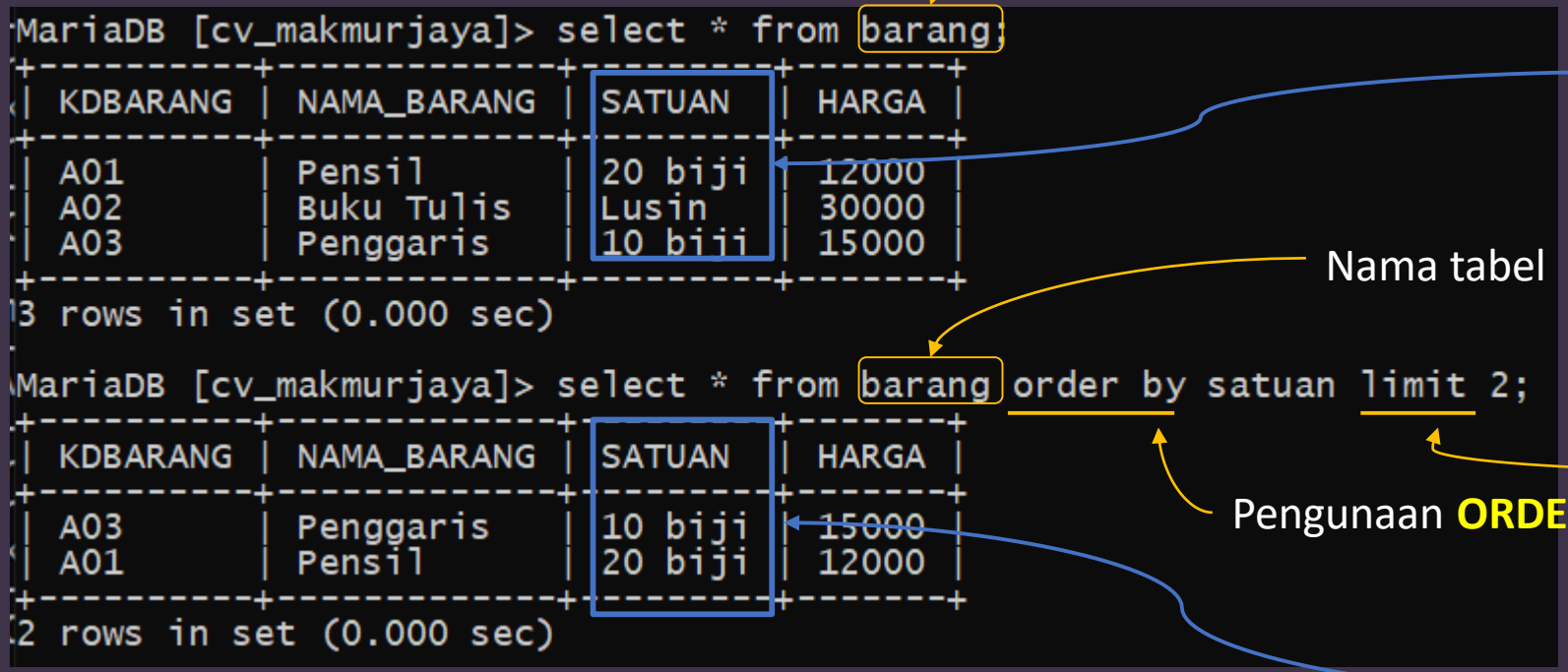
Pengunaan **ORDER BY**

Sesudah melakukan **order by** berdasarkan table satuan

ORDER BY :: SELECT

Perintah untuk mengurutkan dengan menampilkan data secara tertentu

B SELECT * FROM barang ORDER BY satuan LIMIT 2;



Nama tabel

Sebelum melakukan **order by**

Nama tabel

Pengunaan **LIMIT**

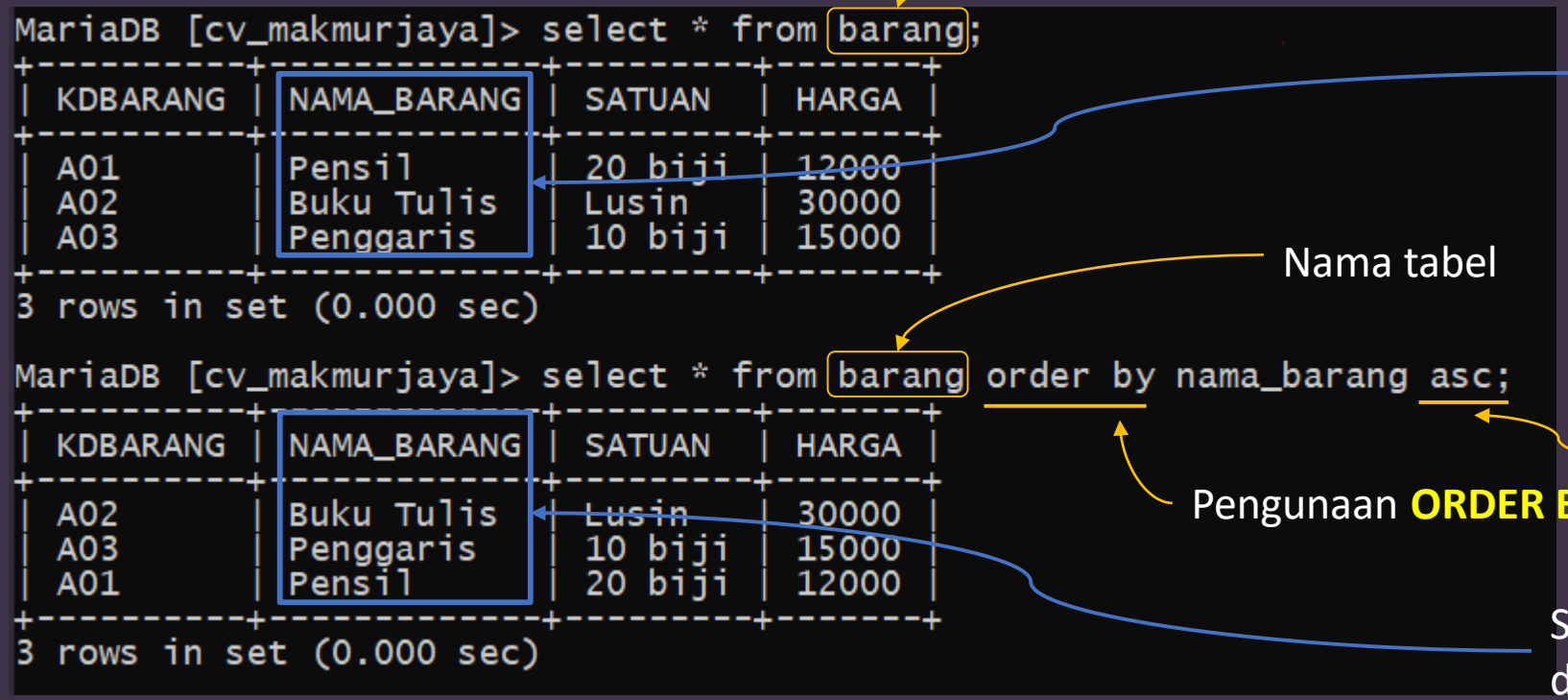
Pengunaan **ORDER BY**

Sesudah melakukan **order by** berdasarkan table satuan dengan 2 data yang ditampilkan

ORDER BY :: SELECT

Perintah untuk mengurutkan dengan menampilkan data secara ascending(Menaik)

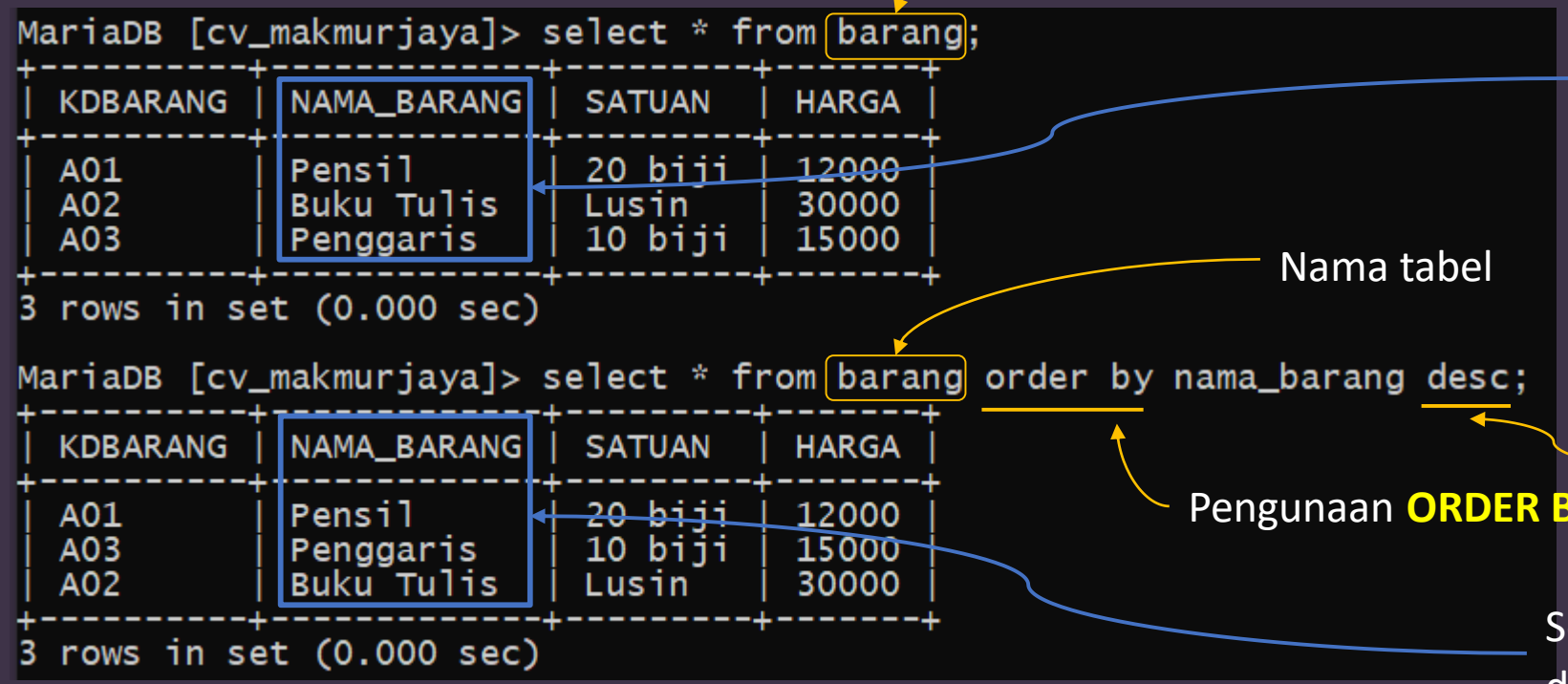
```
C   SELECT * FROM barang ORDER BY satuan ASC;
```



ORDER BY :: SELECT

Perintah untuk mengurutkan dengan menampilkan data secara descending(Menurun)

D SELECT * FROM barang ORDER BY satuan DESC;



Nama tabel

Sebelum melakukan **ASC**

Nama tabel

Pengunaan **DESC**

Pengunaan **ORDER BY**

Sesudah melakukan **DESC** data mulai dari huruf **Z - A**

DISTINCT :: SELECT

Distinct adalah kata kunci ini untuk menghilangan duplikasi

```
SELECT DISTINCT satuan FROM barang;
```

```
MariaDB [cv_makmurjaya]> select * from barang;
+----------+-------------+---------+-------+
| KDBARANG | NAMA_BARANG | SATUAN  | HARGA |
+----------+-------------+---------+-------+
| A01      | Pensil      | 20 biji | 12000 |
| A02      | Buku Tulis  | Lusin   | 30000 |
| A03      | Penggaris   | 10 biji | 15000 |
| A04      | Buku gambar | Lusin   | 40000 |
+----------+-------------+---------+-------+
4 rows in set (0.000 sec)

MariaDB [cv_makmurjaya]> select distinct satuan from barang;
+---------+
| satuan  |
+---------+
| 20 biji |
| Lusin   |
| 10 biji |
+---------+
3 rows in set (0.000 sec)
```

Sebelum melakukan **DISTINCT**

Pengunaan **DISTINCT**

Sesudah melakukan **DISTINCT**

UNION, INTERSECT, EXCEPT:: SELECT

UNION merupakan operator yang digunakan untuk menggabungkan hasil query, dengan ketentuan jumlah, nama dan tipe kolom dari masing-masing tabel yang akan ditampilkan datanya harus sama

```
MariaDB [db_organ_manusia]> select*from tbl_organ_dalam;
+----+-------------+
| id | nama        |
+----+-------------+
|  1 | Jantung     |
|  2 | Paru-paru   |
|  3 | Tenggorokan |
+----+-------------+
3 rows in set (0.000 sec)

MariaDB [db_organ_manusia]> select*from tbl_organ_luar;
+----+-------------+
| id | nama        |
+----+-------------+
|  1 | Hidung      |
|  2 | Mata        |
|  3 | Tenggorokan |
+----+-------------+
3 rows in set (0.000 sec)

MariaDB [db_organ_manusia]> select nama from tbl_organ_dalam union select nama from tbl_organ_luar;
+-------------+
| nama        |
+-------------+
| Jantung     |
| Paru-paru   |
| Tenggorokan |
| Hidung      |
| Mata        |
+-------------+
5 rows in set (0.000 sec)
```

Pengunaan **UNION**

Sesudah melakukan **UNION**

UNION, INTERSECT, EXCEPT:: SELECT

INTERSECT merupakan operator yang digunakan untuk memperoleh data dari dua buah query dimana data yang ditampilkan adalah yang memenuhi kedua query tersebut dengan ketentuan jumlah, nama dan tipe kolom dari masing-masing tabel yang akan ditampilkan datanya harus sama

```
MariaDB [db_organ_manusia]> select*from tbl_organ_dalam;
+----+-------------+
| id | nama        |
+----+-------------+
|  1 | Jantung     |
|  2 | Paru-paru   |
|  3 | Tenggorokan |
+----+-------------+
3 rows in set (0.000 sec)

MariaDB [db_organ_manusia]> select*from tbl_organ_luar;
+----+-------------+
| id | nama        |
+----+-------------+
|  1 | Hidung      |
|  2 | Mata        |
|  3 | Tenggorokan |
+----+-------------+
3 rows in set (0.000 sec)

MariaDB [db_organ_manusia]> select nama from tbl_organ_dalam intersect select nama from tbl_organ_luar;
+-------------+
| nama        |
+-------------+
| Tenggorokan |
+-------------+
1 row in set (0.000 sec)
```

Sesudah melakukan **INTERSECT**

Pengunaan **INTERSECT**

UNION, INTERSECT, EXCEPT:: SELECT

EXCEPT / Set Difference merupakan operator yang digunakan untuk memperoleh data dari dua buah query dimana data yang ditampilkan adalah data yang ada pada hasil query 1 dan tidak terdapat pada data dari hasil query 2 dengan ketentuan jumlah, nama dan tipe kolom dari masing-masing tabel yang akan ditampilkan datanya harus sama

```
MariaDB [db_organ_manusia]> select*from tbl_organ_dalam;
+----+-------------+
| id | nama        |
+----+-------------+
|  1 | Jantung     |
|  2 | Paru-paru   |
|  3 | Tenggorokan |
+----+-------------+
3 rows in set (0.000 sec)

MariaDB [db_organ_manusia]> select*from tbl_organ_luar;
+----+-------------+
| id | nama        |
+----+-------------+
|  1 | Hidung      |
|  2 | Mata        |
|  3 | Tenggorokan |
+----+-------------+
3 rows in set (0.000 sec)

MariaDB [db_organ_manusia]> select nama from tbl_organ_dalam except select nama from tbl_organ_luar;
+-----------+
| nama      |
+-----------+
| Jantung   |
| Paru-paru |
+-----------+
2 rows in set (0.000 sec)
```

Sesudah melakukan **EXCEPT**

Pengunaan **EXCEPT**

LATIHAN

1. Buat tabel pegawai sebagai berikut:

Field	Type	Null	Key	Default	Extra
idpegawai	char(6)	NO		NULL	
namadepan	varchar(20)	YES		NULL	
namabelakang	varchar(25)	NO		NULL	
email	varchar(25)	NO		NULL	
telepon	varchar(20)	YES		NULL	
tglkontrak	date	NO		NULL	
idjob	varchar(10)	NO		NULL	
gaji	int(8)	YES		NULL	
tunjangan	int(8)	YES		NULL	
idmanajer	char(6)	YES		NULL	
iddepartemen	char(4)	YES		NULL	

2. Isi data tabel (data dapat dilihat pada halaman terakhir)!
3. Tampilkan semua kolom di tabel!
4. Tampilkan kolom idpegawai, namabelakang dan gaji saja!
5. Tampilkan kolom idpegawai, namabelakang, gaji, tunjangan dan sebuah kolom baru yaitu tunjangan+gaji yang berisi jumlah tunjangan dan gaji !
6. Ubah tunjangan menjadi NULL untuk pegawai dengan idpegawai = E003. Kemudian lakukan kembali percobaan 5.
7. Seperti percobaan 5, tampilkan kolom idpegawai, namabelakang, gaji, tunjangan dan sebuah kolom baru (gunakan alias) yaitu total_pendapatan yang berisi jumlah tunjangan dan gaji!
8. Tambahkan record baru dengan value: E006,lincoln, burrows, linc@yahoo.com, 085275384544, 2008-09-01, L0006, 1750000, NULL, ex, coml.
9. Untuk pegawai yang ber-id E004 dan E005 ubah idmanajernya menjadi al!
10. Sekarang tampilkan kolom idmanajer saja!
11. Dari percobaan 10, terdapat 3 idmanajer yang sama dengan total record 6, sekarang tampilkan
12. idmanajer tanpa duplikasi idmanajer sehingga akan tampil 4 record dengan idmanajer yang berbeda!
13. Tampilkan pegawai yang gajinya antara 1750000 - 1250000!
14. Tampilkan tabel pegawai yang terurut berdasarkan namabelakang (dari a ke z)!
15. Tampilkan tabel pegawai yang diurutkan berdasarkan namadepan dengan urutan terbalik (dari z ke a)!