

# DATABASE SQL



## Pesantren PeTIK II YBM PLN

Jl. KH. Bisri Syansuri RT/01 RW/05, Plosogeneng,  
Kec. Jombang, Kabupaten Jombang, Jawa Timur



# Pertemuan Ke-10





# Materi

1. Pengantar Database
2. Pemodelan Data
3. Model Relasional Database
4. Normalisasi Database
5. Pengantar SQL
6. Perintah SQL SELECT 1
7. Perintah SQL SELECT 2

9. Fungsi Aggregate dan Grouping Data

## **10. Sub Query & SQL Join Table**

11. View dan Analisa Query
12. Store Procedure dan Function
13. Trigger dan Transaction
14. Manajemen User
15. Backup dan Restore



# 10. Sub Query & SQL Join Table





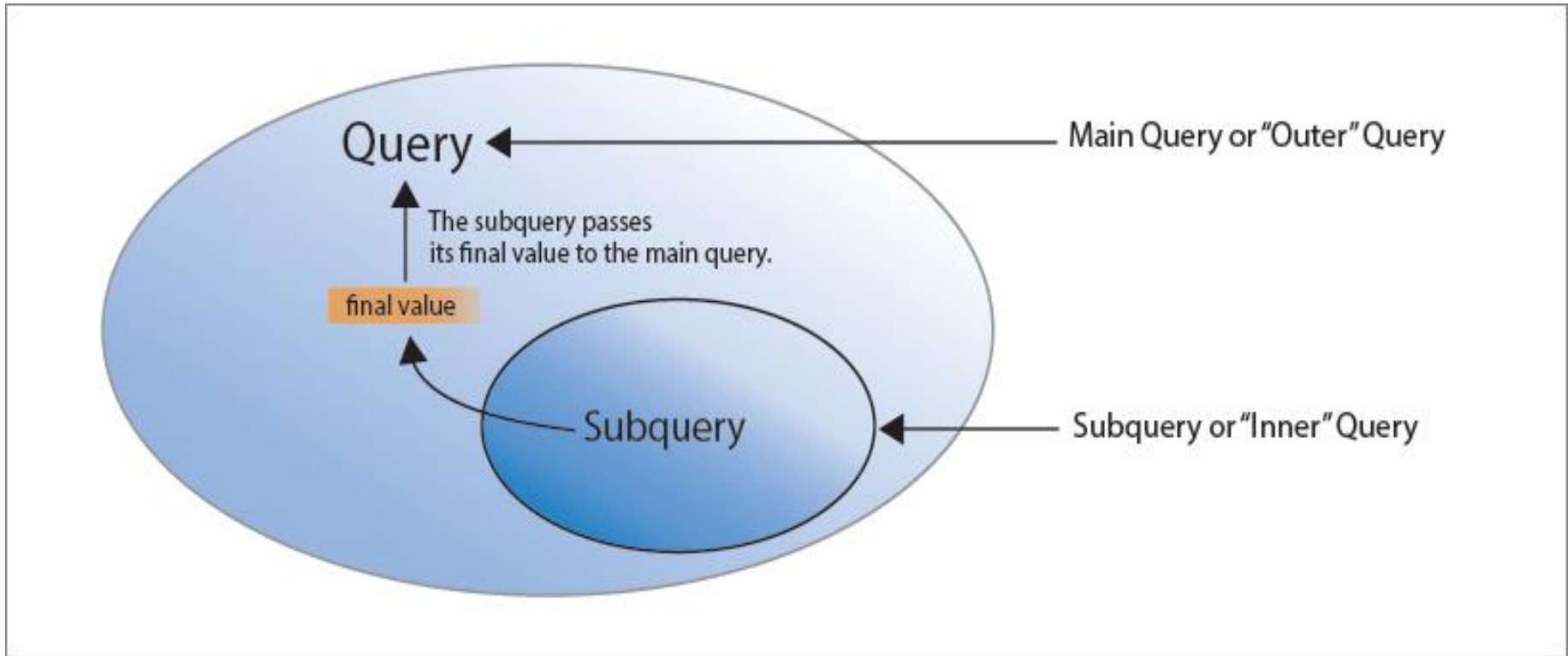
# Subquery

- Query *SELECT* yang berada dalam Query *SELECT* lainnya
- Subquery disebut juga *INNER Query* atau *INNER SELECT*
- Query Utama disebut *Outer Query* atau *Outer SELECT*
- Subquery biasa adalah kriteria yang akan dijadikan kondisi dari Query Utama dengan menggunakan klausa *WHERE*





# Subquery





# Subquery

Outer Query

```
SELECT * FROM Table_1
```

```
WHERE (column_1,column_2) = (SELECT column_1,column_2 FROM Table_2);
```

Inner Query





# Subquery

- ***Terkadang kita perlu menjawab query lain (Query 1) sebelum kita menuliskan query yang kita inginkan (Query 2).***
- ***Query 1 : Inner Query***
- ***Query 2 : Outer Query***
- ***Contohnya kita ingin menampilkan kolom nama, stok pada table produk dimana jumlah stoknya diatas rata-rata stok produk secara keseluruhan***
- ***Query 1 : Query mencari rata-rata stok produk***
- ***Query 2 : Query menampilkan nama, stok table produk***







# Subquery

*Query 1 (Inner Query): Query mencari rata-rata stok produk*

*Query 2 (Outer Query): Query menampilkan nama, stok table produk*

Kita perlu mendefinisikan query dalam dua tahapan

- ❑ 1) `SELECT AVG(stok) FROM produk`
- ❑ 2) `SELECT nama, stok FROM produk WHERE stok > ( HASIL_QUERY_1 )`

Query pertama digunakan sebagai kondisi disebut sub query

- ❑ `SELECT nama, stok FROM produk WHERE stok > ( SELECT AVG(stok) FROM produk )`





# Tabel Relationship

- ☐ One To Many
- ☐ One To One
- ☐ Many To Many





# One To Many Relationship

Tabel agama

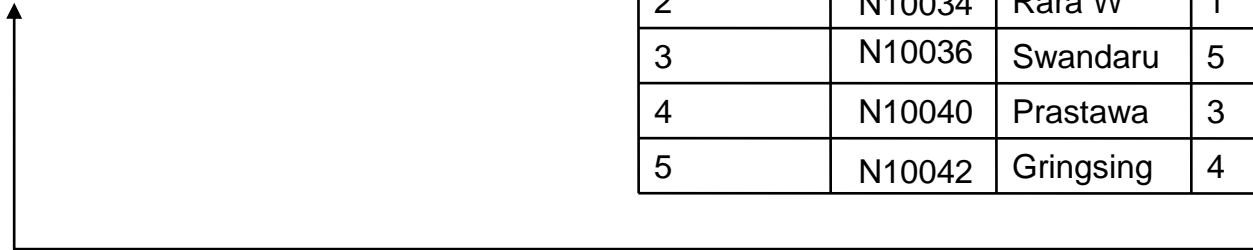
<u>idagama</u>	nama
1	Islam
2	Kristen
3	Khatolik
4	Hindu
5	Budha

Primary  
Key

Foreign  
Key

Tabel pegawai

<u>idpegawai</u>	nip	nama	idagama
1	N10033	Agung S	1
2	N10034	Rara W	1
3	N10036	Swandaru	5
4	N10040	Prastawa	3
5	N10042	Gringsing	4





# One To Many Relationship

Tabel Jenis Produk

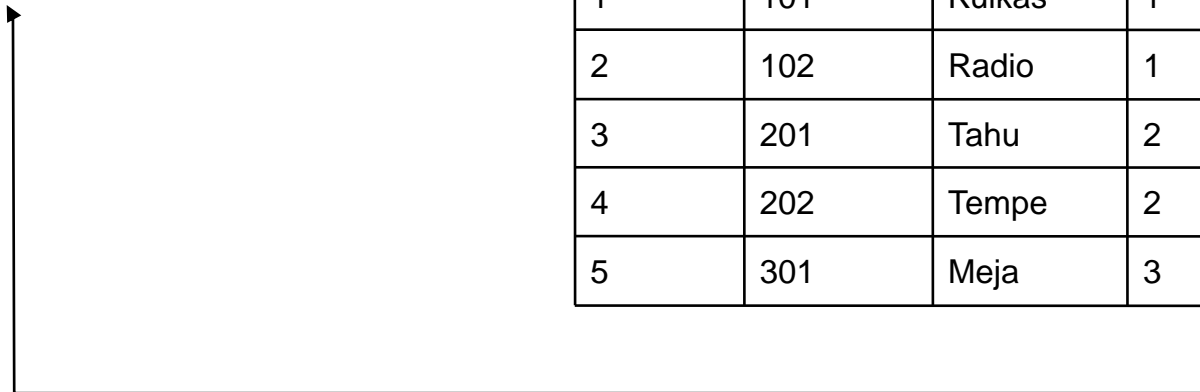
<u>id</u>	nama
1	Elektronika
2	Makanan
3	Furniture
4	Minuman

Primary  
Key

Foreign  
Key

Tabel Produk

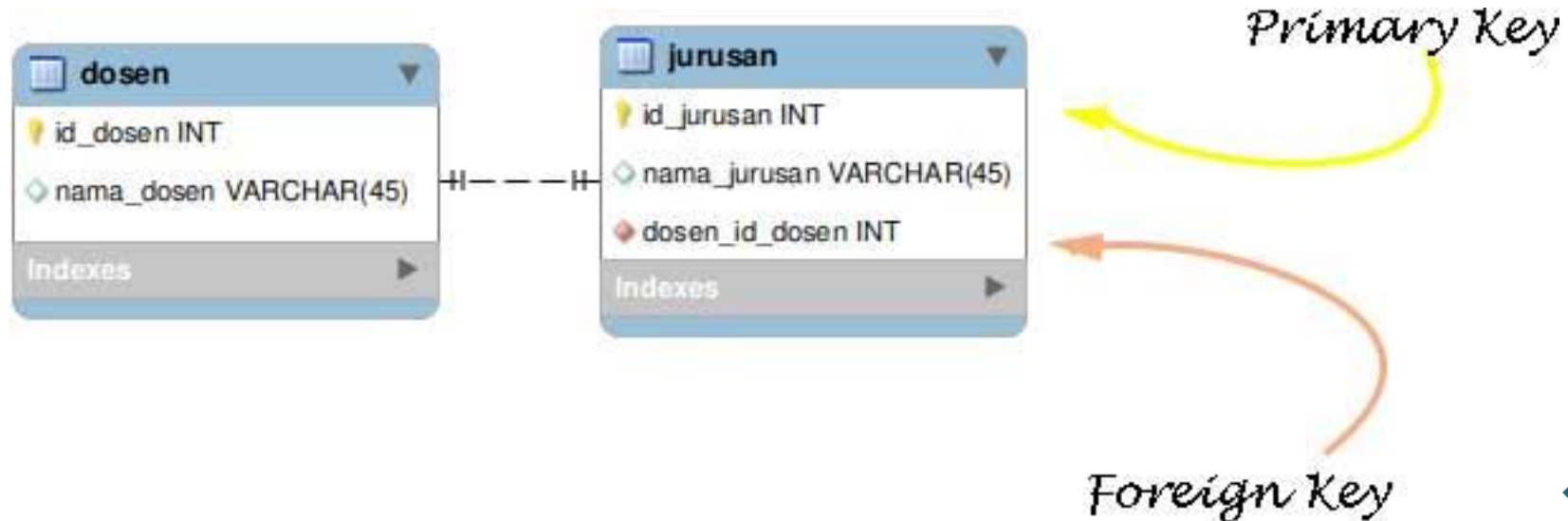
Idprod	Kode	Nama	idjenis
1	101	Kulkas	1
2	102	Radio	1
3	201	Tahu	2
4	202	Tempe	2
5	301	Meja	3





# Karakteristik Foreign Key

- ❑ Foreign key adalah wakil data dari tabel acuan, semua detail data berada pada tabel acuan
- ❑ Field foreign key biasanya adalah field primary key di tabel acuan





# Referential Integrity

```
CREATE TABLE jenis_produk  
( idjenis integer auto_increment primary key,  
  nama varchar(32) unique not null,  
)Engine=InnoDB;
```

Tabel Acuan

```
CREATE TABLE produk  
( id integer auto_increment PRIMARY KEY ,  
  kode varchar(6) unique not null,  
  nama varchar(32) not null, harga double ,  
  stok integer, min_stok integer, idjenis integer ,  
  FOREIGN KEY (idjenis) REFERENCES  
  jenis_produk ( idjenis )  
  ON DELETE CASCADE ON UPDATE CASCADE  
) Engine=InnoDB;
```





# One To One Relationship -1

Data Gaji sensitif diakses oleh user tertentu

Tabel Pegawai



Idpeg	NIP	Nama	Telpon	Gaji
1	11320	Tarom	777556	5000000
2	11201	Ronny	766555	3000000
3	22232	Reza	785656	4500000
4	23322	Nasrul	846555	1500000
5	33222	Edi	754565	1000000

Terdapat alasan untuk memisahkan data dengan alasan : data sensitif (hak akses), optimasi query





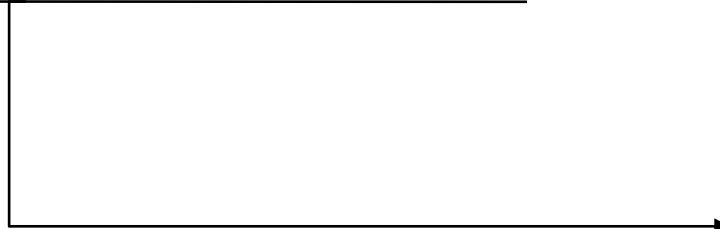
# One To One Relationship

Tabel Gaji

<u>idpeg</u>	gaji
1	5000000
2	3000000
3	4500000
4	1500000
5	1500000

Tabel Pegawai

Idpeg	NIP	Nama	Telpon
1	11320	Tarom	777556
2	11201	Ronny	766555
3	22232	Reza	785656
4	23322	Nasrul	846555
5	33222	Edi	754565







# Many To Many Relationship

Tabel Pegawai

<u>idpeg</u>	nama
1	Tarom
2	Ronny
3	Reza
4	Nasrul
5	Edi

Tabel Kursus

<u>idkursus</u>	nama
1	PHP & MySQL
2	PostgreSQL
3	Linux Basic
4	Open Office
5	Java

Tabel Pegawai\_Kursus

<u>idpeg</u>	<u>idkursus</u>	Nilai
1	1	80
1	2	78
4	1	90
5	5	60





# JOIN TABLE

- A SQL JOIN combines records from two tables.
- A JOIN locates related column values in the two tables.
- A query can contain zero, one, or multiple JOIN operations.
- INNER JOIN is the same as JOIN; the keyword INNER is optional.

## The SQL JOIN syntax

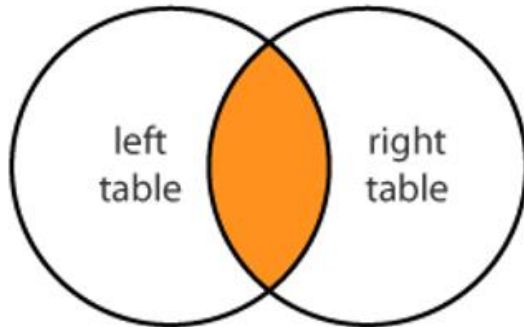
The general syntax is:

```
1. SELECT column-names
2. FROM table-name1 JOIN table-name2
3. ON column-name1 = column-name2
4. WHERE condition
```

The general syntax with INNER is:

```
1. SELECT column-names
2. FROM table-name1 INNER JOIN table-name2
3. ON column-name1 = column-name2
4. WHERE condition
```

INNER JOIN





# JOIN TABLE

- (INNER) JOIN: Select records that have matching values in both tables.
- LEFT (OUTER) JOIN: Select records from the first (left-most) table with matching right table records.
- RIGHT (OUTER) JOIN: Select records from the second (right-most) table with matching left table records.
- FULL (OUTER) JOIN: Selects all records that match either left or right table records.

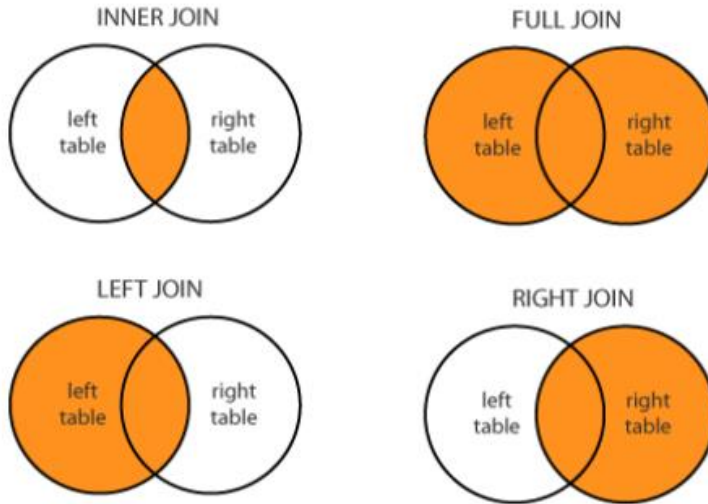
## The SQL JOIN syntax

The general syntax is:

```
1. SELECT column-names
2. FROM table-name1 JOIN table-name2
3. ON column-name1 = column-name2
4. WHERE condition
```

The general syntax with INNER is:

```
1. SELECT column-names
2. FROM table-name1 INNER JOIN table-name2
3. ON column-name1 = column-name2
4. WHERE condition
```



All INNER and OUTER keywords are optional.

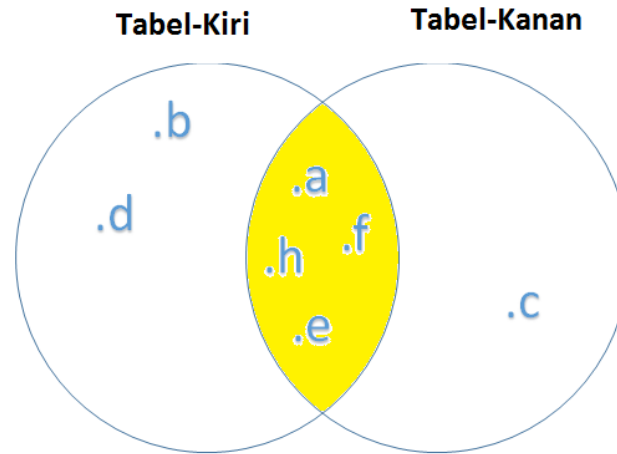
Details about the differences between these JOINS are available in subsequent tutorial pages.





# INNER JOIN

- Menampilkan hanya data yang ada di dua table kiri dan kanan



```
SELECT Tabel-Kiri.* , Tabel-Kanan.*  
FROM Tabel-Kiri INNER JOIN TABLE-Kanan  
ON TABLE-KIRI.FIELD = TABLE-KANAN.FIELD
```





# Query Join Table :: INNER JOIN

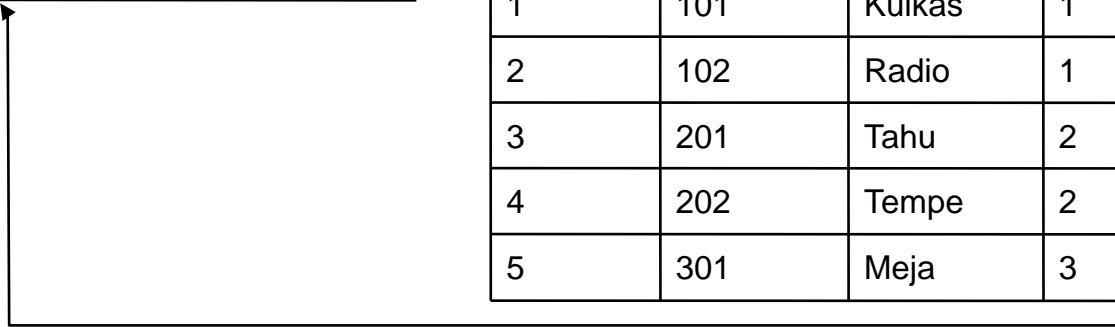
Akan menampilkan hanya data yang terhubung antara dua tabel

Tabel Jenis Produk

<u>id</u>	nama
1	Elektronika
2	Makanan
3	Furniture
4	Minuman

Tabel Produk

Idprod	Kode	Nama	idjenis
1	101	Kulkas	1
2	102	Radio	1
3	201	Tahu	2
4	202	Tempe	2
5	301	Meja	3





# Query Join Table :: INNER JOIN

```
SELECT produk.id, produk.kode, produk.nama,  
jenis_produk.nama as jenis  
from produk inner join jenis_produk  
on produk.idjenis=jenis_produk.id
```

Id	Kode	Nama	jenis
1	101	Kulkas	Elektronika
2	102	Radio	Elektronika
3	201	Tahu	Makanan
4	202	Tempe	Makanan
5	301	Meja	Furniture





# Query Join Table :: INNER JOIN

Role Akan menampilkan hanya data yang terhubung antara dua tabel

<u>id</u>	nama
1	Admin
2	Author
3	Reviewer
4	Guest

Tabel User

id	nama	email	idrole
1	andi	an@nf.co m	1
2	caca	cc@nf.co m	2
3	lina	ina@nf.co m	2
4	mawar	mw@nf.co m	4
5	zidan	zd@nf.com	4

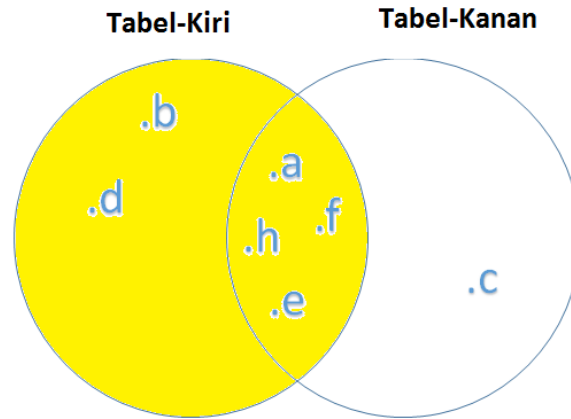


Tampilkan data user beserta rolenya !!



# LEFT JOIN

- Menampilkan semua data di tabel kiri dan yang beririsan dengan data di tabel kanan



```
SELECT Tabel-Kiri.* , Tabel-Kanan.*  
FROM Tabel-kiri LEFT JOIN TABLE-kanan  
ON TABLE-KIRI.FIELD = TABLE-KANAN.FIELD
```







# Query Join Table :: LEFT JOIN

```
SELECT produk.id, produk.kode, produk.nama,  
jenis_produk.nama as jenis  
from jenis_produk LEFT JOIN produk  
on produk.idjenis=jenis_produk.id
```

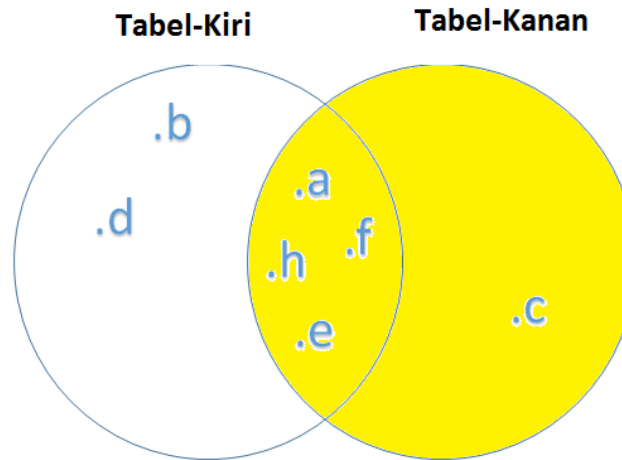
Id	Kode	Nama	jenis
1	101	Kulkas	Elektronika
2	102	Radio	Elektronika
3	201	Tahu	Makanan
4	202	Tempe	Makanan
5	301	Meja	Furniture
			Minuman





# RIGHT JOIN

- Menampilkan semua data di tabel kiri dan yang beririsan dengan data di tabel kanan



```
SELECT Tabel-Kiri.* , Tabel-Kanan.*  
FROM Tabel-Kiri RIGHT JOIN TABLE-Kanan  
ON TABLE-KIRI.FIELD = TABLE-KANAN.FIELD
```





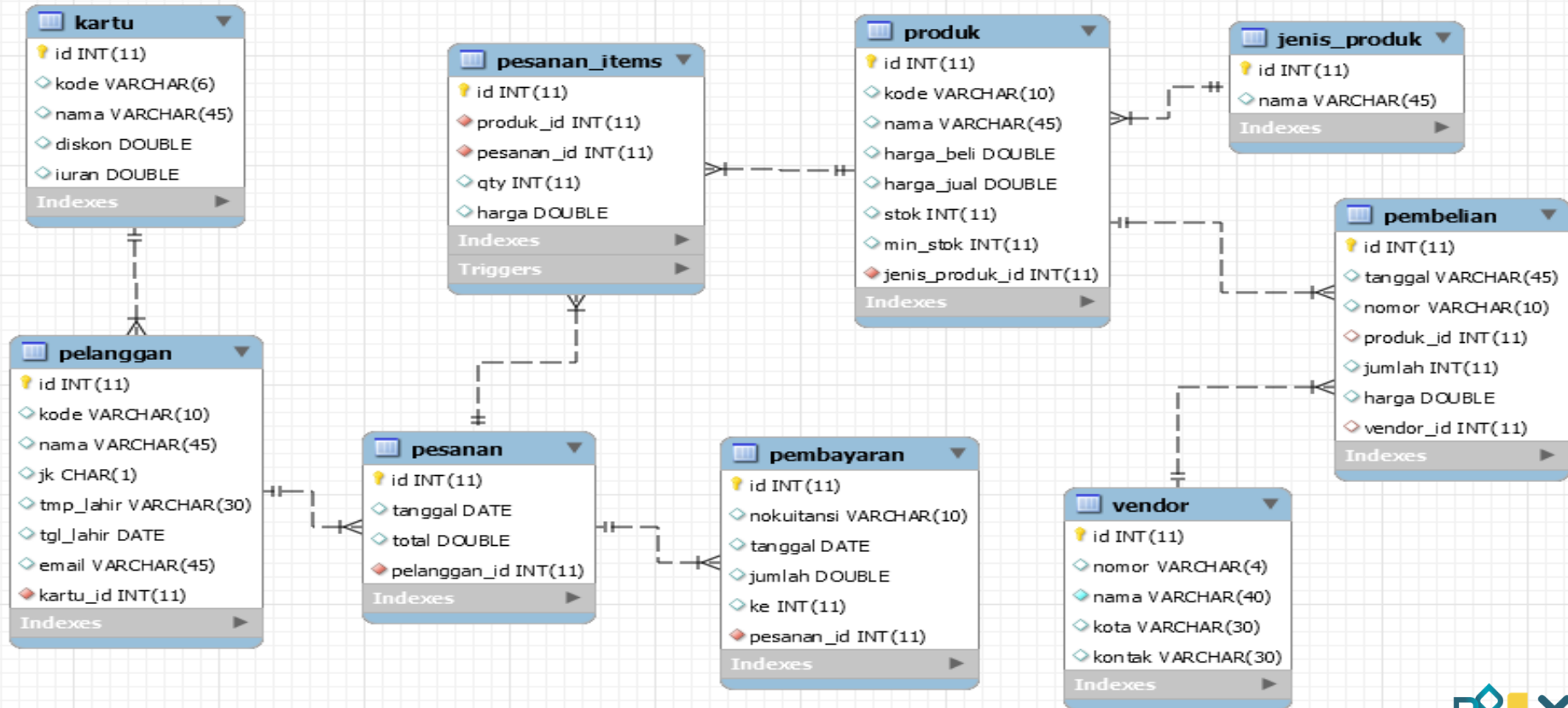
# Query Join Table :: RIGHT JOIN

```
SELECT produk.id, produk.kode, produk.nama,  
jenis_produk.nama as jenis  
from jenis_produk RIGHT JOIN produk  
on produk.idjenis=jenis_produk.id
```

id	Kode	Nama	jenis
1	101	Kulkas	Elektronika
2	102	Radio	Elektronika
3	201	Tahu	Makanan
4	202	Tempe	Makanan
5	301	Meja	Furniture



# ERD Database dbkoperasi



**TERIMA KASIH  
ATAS SEGALA PERHATIAN  
SEMOGA BERMANFAAT...**

