

Persiapan COTS



By Dzaky Fazli

SELECT

✓ Kegunaan

Menampilkan beberapa data dari atribut yang dipilih dari satu tabel atau lebih

✓ Struktur

```
SELECT attributes1, attributes2, attributes3 FROM table;
```

Atribut yang akan ditampilkan pada hasil Query

nama tabel asal atribut

SELECT

✓ Contoh

employees
* employeeNumber
lastName
firstName
extension
email
officeCode
reportsTo
jobTitle

Ingin menampilkan data firstName dan lastName yang ada pada tabel employees

```
SELECT firstName, lastName FROM employees;
```

Hasilnya



firstName	lastName
Diane	Murphy
Mary	Patterson
Jeff	Firrelli
William	Patterson
Gerard	Bondur
Anthony	Bow
Leslie	Jennings
Leslie	Thompson
Julie	Firrelli
Steve	Patterson
Foon Yue	Tseng
George	Vanauf

ORDER BY

✓ Kegunaan

Fitur pada query SELECT yang berfungsi untuk mengurutkan data sesuai dengan suatu atribut

✓ Struktur

```
SELECT attributes1, attributes2 FROM table ORDER BY attributes1 ASC;  
DESC
```

ascending
descending

Atribut yang akan digunakan sebagai
paramater pengurutan



ORDER BY

✓ Contoh

employees
* employeeNumber
lastName
firstName
extension
email
officeCode
reportsTo
jobTitle

Ingin menampilkan data firstName dan lastName **sesuai dengan abjad dari firstName** yang ada pada tabel employees

```
SELECT firstName, lastName FROM employees ORDER BY firstName ASC;
```

Hasilnya



firstName ▲ 1	lastName
Andy	Fixter
Anthony	Bow
Barry	Jones
Diane	Murphy
Foon Yue	Tseng
George	Vanauf
Gerard	Bondur
Gerard	Hernandez
Jeff	Firrelli
Julie	Firrelli

WHERE

✓ Kegunaan

Fitur pada query SELECT yang berfungsi untuk menambahkan syarat pada suatu query

✓ Struktur

```
SELECT attribute1, attribute2 FROM table WHERE ...;
```



dapat diisi dengan :

< / > / >= / attributes1 = 10

- persamaan & pertidaksamaan

NOT / AND / OR

- perbandingan

'%a' / 'a%' / attributes1 LIKE '%a%'

- operator LIKE

WHERE

✓ Contoh

customers
* customerNumber
customerName
contactLastName
contactFirstName
phone
addressLine1
addressLine2
city
state
postalCode
country
salesRepEmployeeNumber
creditLimit

Ingin menampilkan data customerNumber, contactFirstName dan contactLastName dengan syarat customerNumber harus kurang dari 129

```
SELECT customerNumber, contactFirstName, contactLastName FROM customers  
WHERE customerNumber < 129;
```

Hasilnya



customerNumber	contactFirstName	contactLastName
103	Carine	Schmitt
112	Jean	King
114	Peter	Ferguson
119	Janine	Labrune
121	Jonas	Bergulfsen
124	Susan	Nelson
125	Zbyszek	Piestrzeniewicz
128	Roland	Keitel

Disclaimer

Suatu Query diakhiri dengan tanda titik koma (;) **BUKAN** *enter* pada keyboard

1 `SELECT attribute1, attribute2 FROM table WHERE attribute1 LIKE 'a%';`

2 `SELECT attribute1, attribute2 FROM table
WHERE attribute1 LIKE 'a%';`

Baik Query 1 ataupun 2 memiliki arti yang **SAMA** dan tetap menjadi **SATU** query

FUNCTION

CONCAT()

✓ Kegunaan

Fungsi untuk menggabungkan dua atribut / data atau lebih dalam satu Kolom yang akan ditampilkan pada hasil Query

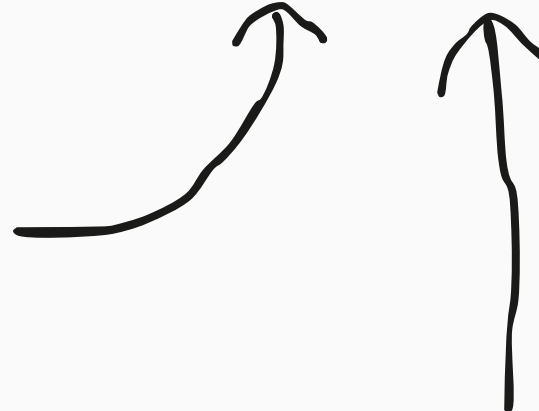

✓ Struktur

```
SELECT CONCAT(attribute1, attribute2) FROM table;
```



2 atribut akan digabungkan menjadi 1 kolom pada query

```
SELECT CONCAT(attribute1, attribute2) AS colName FROM table;
```



memberikan nama pada kolom baru

nama kolom baru

CONCAT()

✓ Contoh

employees
* employeeNumber
lastName
firstName
extension
email
officeCode
reportsTo
jobTitle

Ingin menampilkan data dari
firstName dan lastName menjadi
fullName pada table employees

```
1 SELECT CONCAT(firstName," ",lastName) AS fullname FROM employees;
```

```
2 SELECT CONCAT(firstName,lastName) AS fullname FROM employees;
```

Hasilnya



1

fullname
Diane Murphy
Mary Patterson
Jeff Firrelli
William Patterson
Gerard Bondur
Anthony Bow
Leslie Jennings
Leslie Thompson
Julie Firrelli
Steve Patterson
Foon Yue Tseng

2

fullname
DianeMurphy
MaryPatterson
JeffFirrelli
WilliamPatterson
GerardBondur
AnthonyBow
LeslieJennings
LeslieThompson
JulieFirrelli
StevePatterson
Foon YueTseng

SUM() & COUNT()

✓ Kegunaan

- Sum() untuk menjumlahkan suatu data pada atribut
- Count() untuk menghitung jumlah data pada tabel/atribut

✓ Struktur

```
SELECT COUNT(*) FROM table;
```

```
SELECT SUM(attribute1) FROM table;
```

```
SELECT SUM(attribute1) AS colName FROM table;
```


bisa menggunakan AS

SUM() & COUNT()

✓ Contoh

payments
* customerNumber
* checkNumber
paymentDate
amount

Ingin menghitung jumlah amount sebagai totalAmount keseluruhan dan jumlah data pembayaran sebagai totalPayments pada tabel payments

```
SELECT SUM(amount) AS totalAmount, COUNT(*) AS totalPayment  
FROM payments;
```

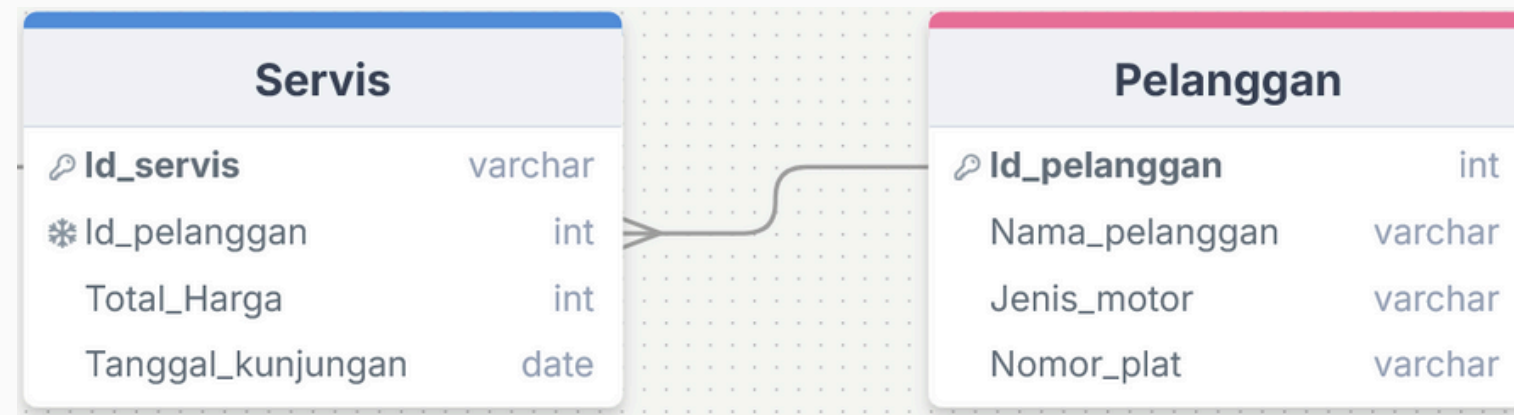
Hasilnya



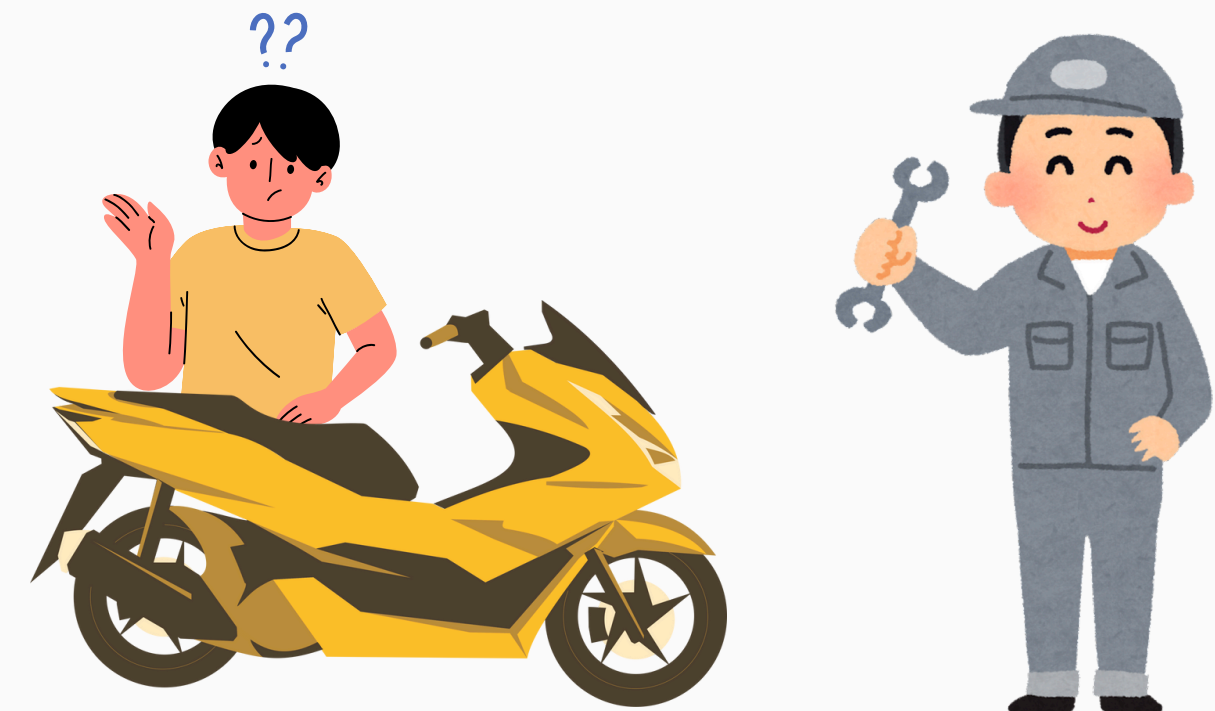
totalAmount	totalPayment
8853839.23	273

RELATIONAL TABLE

Contoh Kasus



tabel Servis ber-relasi dengan tabel Pelanggan, dengan pasangan kunci PK dan FK pada masing-masing tabel. dengan setiap tabel menyimpan data yang berbeda



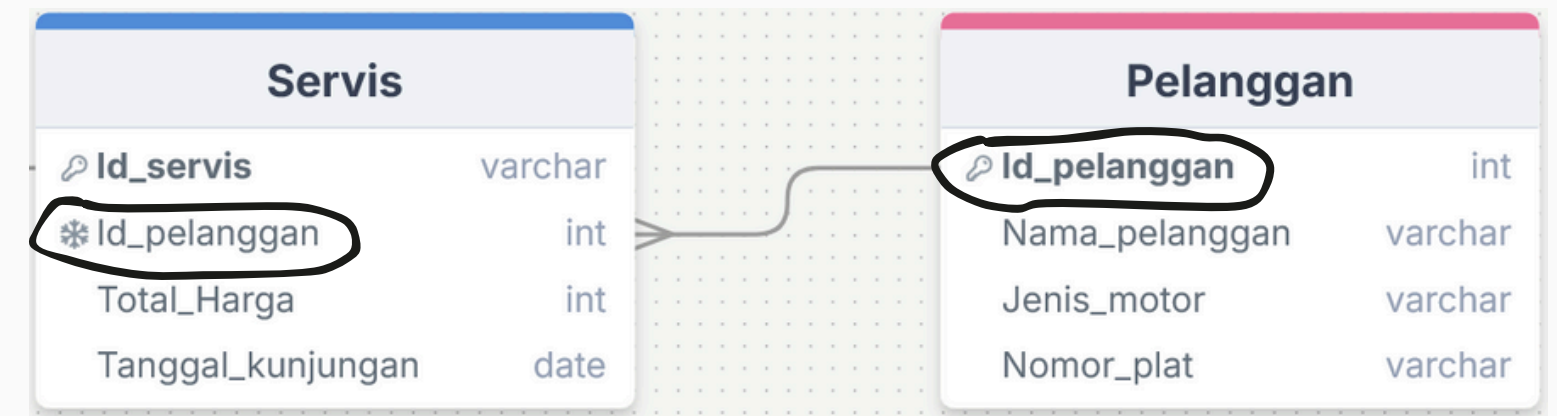
Jika pelanggan hendak melakukan pembayaran setelah servis, bagaimana sistem mengenali bahwa pelanggan itu yang melakukan servis?

Penulisan atribut

Tabel asal atribut

`table1.attribute1`

nama atribut



Berguna agar tidak terjadi konflik jika dua buah tabel menamakan satu atribut yang sama

INNER JOIN

✓ Kegunaan

Mengambil sebuah/banyak data dalam tabel yang berbeda dengan syarat terdapat relasi pada dua atau lebih tabel

✓ Struktur

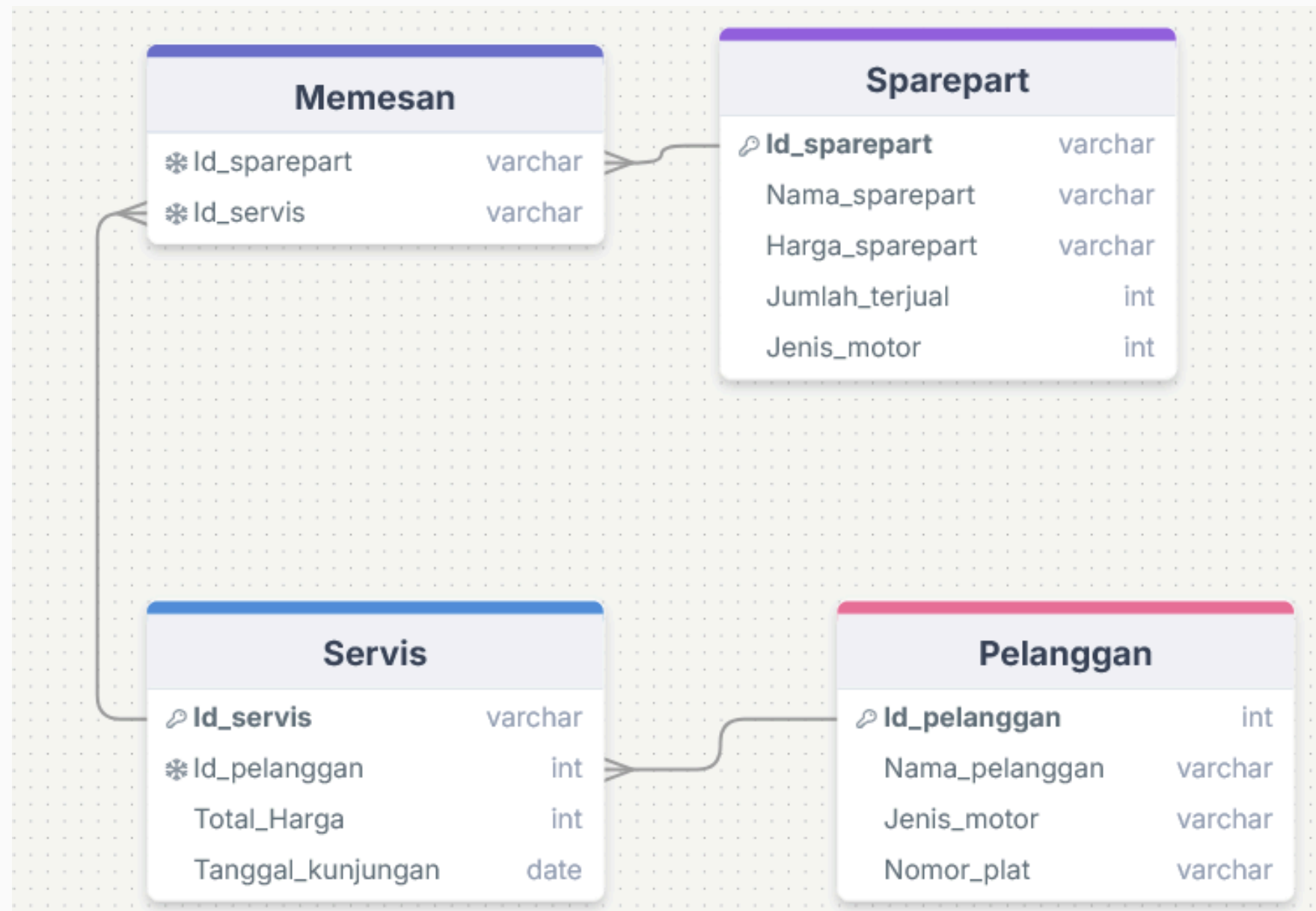
```
SELECT table1.attribute1, table2.attribute2  
FROM table1 INNER JOIN table2  
ON table1.keyAttribute = table2.foreignKeyAttribute;
```

Mengidentifikasi persamaan antara PK
dan FK yang menjadikan relasi antar
tabel

mengidentifikasi tabel yang berelasi

INNER JOIN

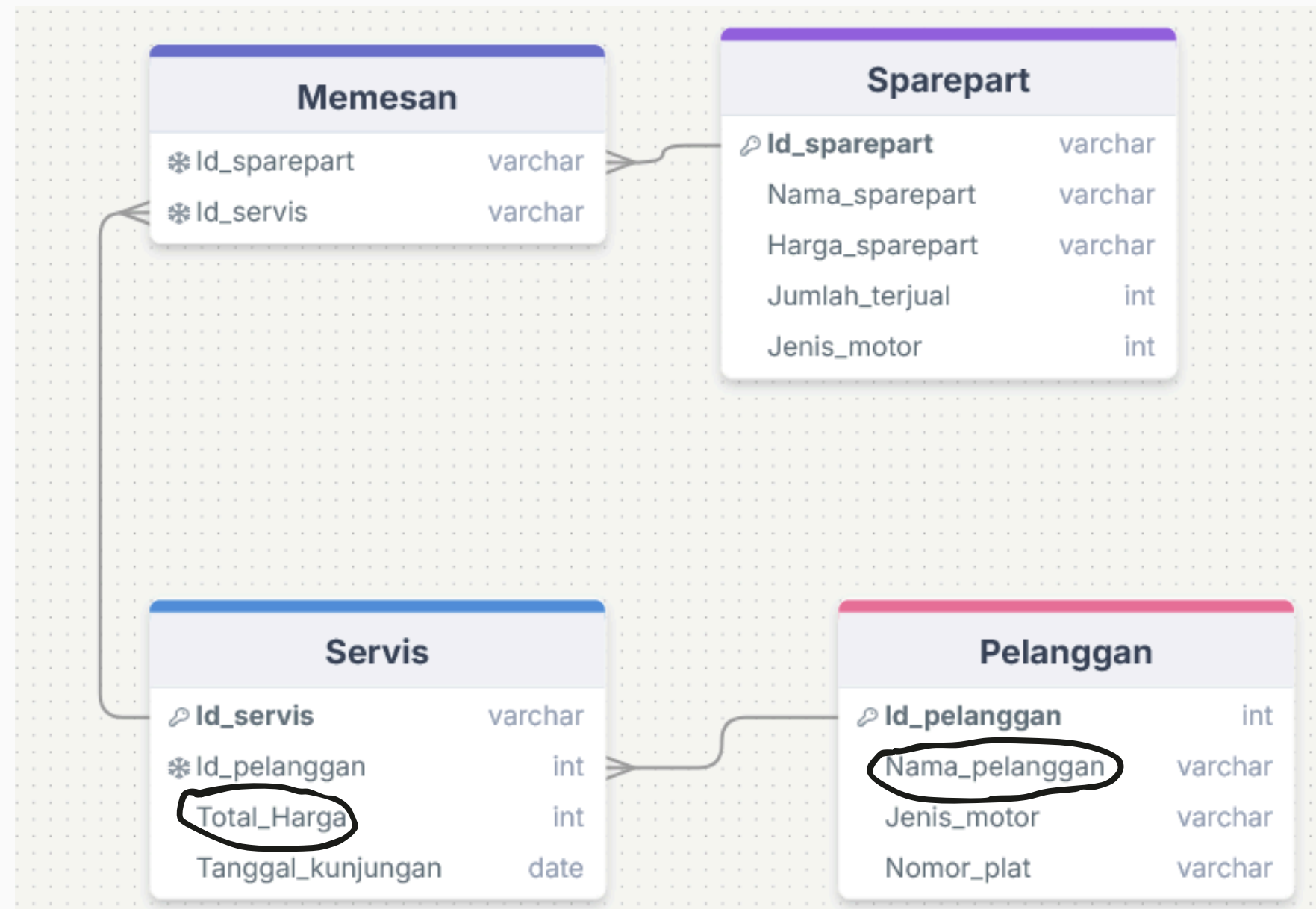
✓ Contoh



Pelanggan bernama “Dzaky” ingin melakukan pembayaran pada setiap servisnya. berapa saja kah total harga yang harus Dzaky bayar?

INNER JOIN

✓ Contoh



Langkah ke-1 :

Identifikasi atribut apa saja yang perlu diambil dan tabel asal atribut tersebut.

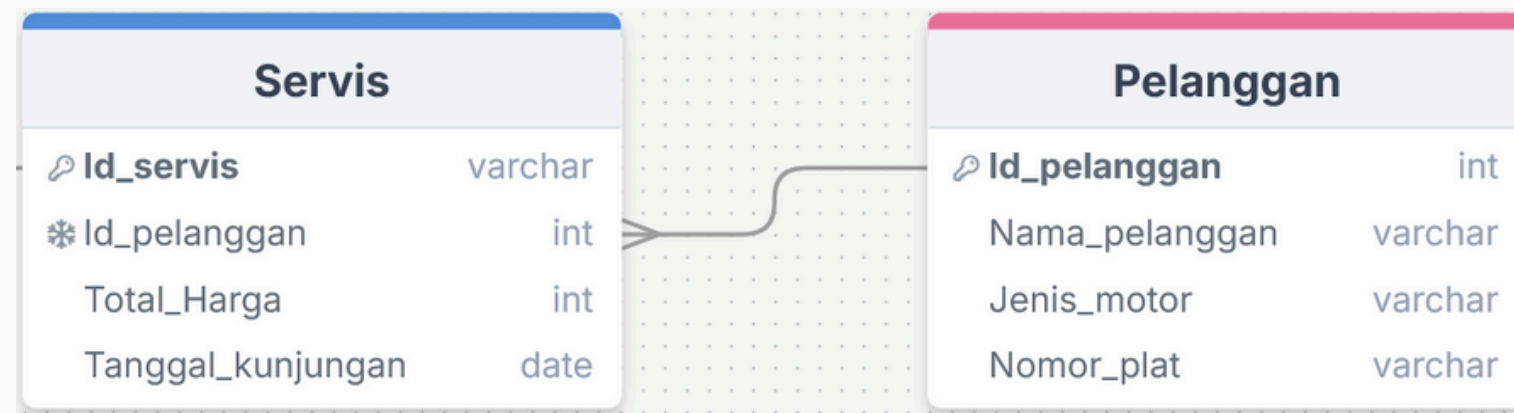
- Total_Harga (tabel Servis)
- Nama_pelanggan (tabel Pelanggan)

Dapat dituliskan sebagai :

```
servis.Total_Harga  
pelanggan>Nama_pelanggan
```


INNER JOIN

✓ Contoh



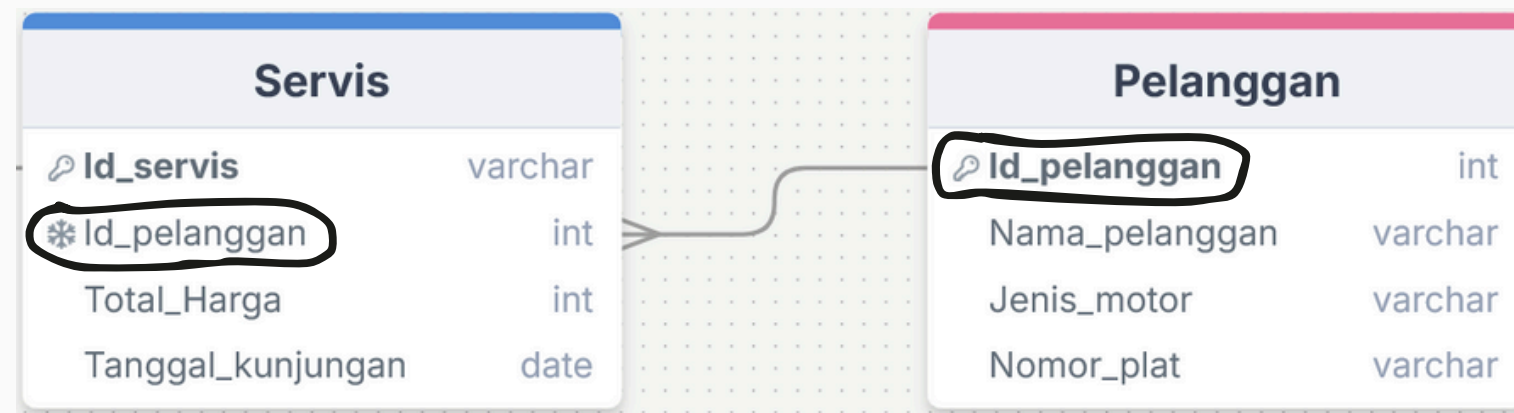
Langkah ke-2 :

Fokus kepada Tabel dimana asal Query berada

- Servis : asal atribut **Total_Harga**
- Pelanggan : asal atribut **Nama_pelanggan**

INNER JOIN

✓ Contoh



Langkah ke-3 :

Identifikasi PK dan FK untuk setiap relasi beserta tabel asal atribut kunci-nya

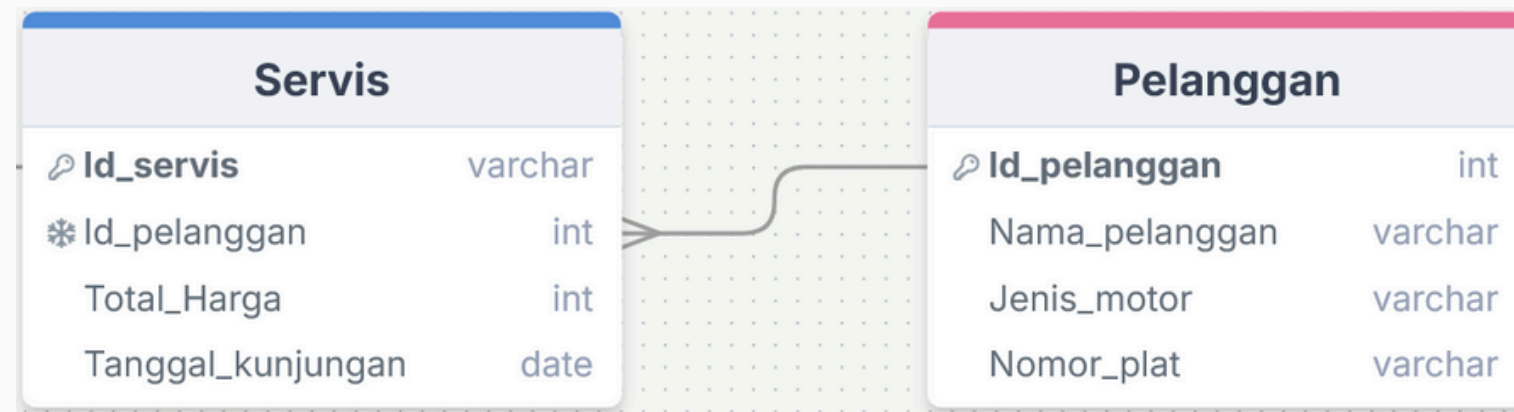
- PK : Id_pelanggan (tabel Pelanggan)
- FK : Id_pelanggan (tabel Servis)

Dapat dituliskan sebagai :

```
pelanggan.Id_pelanggan  
servis.Id_pelanggan
```

INNER JOIN

✓ Contoh



Langkah ke-4 :

Proses desain query sesuai dengan langkah-langkah sebelumnya

```
SELECT servis.Total_Harga
FROM servis INNER JOIN pelanggan
ON servis.Id_pelanggan = pelanggan.Id_pelanggan
WHERE pelanggan>Nama_pelanggan = "Dzaky";
```

Breakdown

✓ Contoh

Pada soal dan langkah ke-1, dinyatakan bahwa pelanggan bernama **Dzaky** ingin membayar biaya **total harga** servis.

Untuk mendapatkan data Total_Harga

```
SELECT servis.Total_Harga
```

Breakdown

✓ Contoh

Pada soal dan langkah ke-2, setiap atribut berasal dari tabel **servis** dan tabel **pelanggan** yang berbeda sehingga dapat dituliskan bahwa dua tabel berelasi

```
FROM servis INNER JOIN pelanggan
```

Breakdown

✓ Contoh

Pada soal dan langkah ke-3, telah teridentifikasi PK dan FK dari setiap tabel yang berelasi

dan dapat dilakukan operasi persamaan sebagai berikut

```
ON servis.Id_pelanggan = pelanggan.Id_pelanggan
```

Breakdown

✓ Contoh

Pada soal dan langkah ke-1, dinyatakan bahwa pelanggan bernama **Dzaky** ingin membayar biaya **total harga** servis.

Untuk mencari pelanggan bernama Dzaky

```
WHERE pelanggan>Nama_pelanggan = "Dzaky"
```

Breakdown

✓ Contoh

Sehingga didapatkan desain Query yang akan dieksekusi sebagai berikut

```
SELECT servis.Total_Harga
FROM servis INNER JOIN pelanggan
ON servis.Id_pelanggan = pelanggan.Id_pelanggan
WHERE pelanggan>Nama_pelanggan = "Dzaky";
```


Hasilnya



Total_Harga
40000
40000
400000
90000
150000
100000
145000
100000
100000
500000
10800000
2300000
430000

Alternatif

Sebuah Query Join yang tidak perlu melakukan persamaan antar Key Attribute

NATURAL JOIN

✓ Struktur

```
SELECT table1.attribute1, table2.attribute2  
FROM table1 NATURAL JOIN table2;
```



mengidentifikasi tabel yang berelasi

Semoga Sukses COTS
