

Praktikum 8

Table 1

```
MariaDB [company]> Select * from OrderDetails;
```

OrderID	ProductID	UnitPrice	Quantity	Discount
10260	70	12.00	21	0.25
10260	62	39.40	15	0.00
10260	57	15.60	1	0.00
10260	41	7.70	16	0.00
10259	37	20.80	10	0.25
10259	21	8.00	50	0.00
10258	32	25.60	6	0.00
10258	5	17.00	65	0.20
10257	2	15.20	15	0.20
10257	77	10.40	6	0.00
10257	39	14.40	6	0.00
10256	27	35.10	15	0.00
10256	77	10.40	12	0.00
10256	53	26.20	15	0.00

14 rows in set (0.002 sec)

Table 2

```
MariaDB [company]> select * from Products;
```

ProductID	ProductName	SupplierID	QuantityPerUnit	UnitPrice	UnitsInStock
2	Chang	1	24 - 12 oz bottles	19.00	17
5	Chef Anton's Gumbo Mix	2	36 boxes	21.35	50
21	Sir Rodney's Scones	8	24 pkgs. x 4 pieces	10.00	17
27	Schoggi Schokolade	11	100 - 100 g pieces	43.90	49
32	Mascarpone Fabioli	14	24 - 200 g pkgs.	32.00	9
37	Gravad lax	17	12 - 500 g pkgs.	26.00	11
39	Chartreuse verte	18	750 cc per bottle	18.00	69
41	Jack's Clam Chowder	19	12 - 12 oz cans	9.65	85
53	Perth Pasties	24	48 pieces	32.80	36
57	Ravioli Angelo	26	24 - 250 g pkgs.	19.50	17
62	Tarte au sucre	29	48 pies	49.30	76
65	Hot Pepper Sauce	2	24 - 355 ml bottles	21.05	15
70	Longlife Tofu	4	5 Kg pkg.	10.00	4
74	Original Frankfurter	12	12 boxes	13.00	32
77	Outback Lager	7	32 - 8 oz bottles	15.00	15

15 rows in set (0.001 sec)

QUERY 1

```
SELECT p.ProductID, p.ProductName, od.Quantity, p.UnitsInStock  
-> FROM Products p
```

```
-> INNER JOIN OrderDetails od ON p.ProductID = od.ProductID
-> ORDER BY p.ProductID ASC;
```

HASIL

```
ERROR 1052 (23000): Column 'ProductID' in on clause is ambiguous
MariaDB [company]> SELECT p.ProductID, p.ProductName, od.Quantity, p.UnitsInStock
-> FROM Products p
-> INNER JOIN OrderDetails od ON p.ProductID = od.ProductID
-> ORDER BY p.ProductID ASC;
```

ProductID	ProductName	Quantity	UnitsInStock
2	Chang	15	17
5	Chef Anton's Gumbo Mix	65	50
21	Sir Rodney's Scones	50	17
27	Schoggi Schokolade	15	49
32	Mascarpone Fabioli	6	9
37	Gravad lax	10	11
39	Chartreuse verte	6	69
41	Jack's Clam Chowder	16	85
53	Perth Pasties	15	36
57	Ravioli Angelo	1	17
62	Tarte au sucre	15	76
70	Longlife Tofu	21	4
77	Outback Lager	6	15
77	Outback Lager	12	15

14 rows in set (0.102 sec)

ANALISIS

1. SELECT p.ProductID, p.ProductName, od.Quantity, p.UnitsInStock

- Bagian ini menentukan kolom-kolom yang akan ditampilkan dalam hasil query. Kolom-kolom tersebut adalah:
 - p.ProductID: ID produk dari tabel Products.
 - p.ProductName: Nama produk dari tabel Products.
 - od.Quantity: Jumlah produk yang dipesan dari tabel OrderDetails.
 - p.UnitsInStock: Jumlah unit produk yang tersedia di stok dari tabel Products

2. FROM Products p

- Bagian ini menentukan tabel utama yang digunakan dalam query, yaitu tabel Products. Alias p digunakan untuk mempermudah referensi ke tabel ini.

3. INNER JOIN OrderDetails od ON p.ProductID = od.ProductID

- Bagian ini menghubungkan tabel Products dengan tabel OrderDetails menggunakan klausa INNER JOIN. Hubungan dibuat berdasarkan kolom ProductID yang ada di kedua tabel tersebut.
- Alias od digunakan untuk tabel OrderDetails.

4. ORDER BY p.ProductID ASC

- Bagian ini menentukan urutan hasil query berdasarkan kolom ProductID dari tabel Products secara ascending (menaik).

Penjelasan Fungsionalitas

Query ini mengambil data dari dua tabel: `Products` dan `OrderDetails`. Untuk setiap produk yang memiliki entri dalam tabel `OrderDetails`, query ini akan menampilkan:

- ID produk (`ProductID`)
- Nama produk (`ProductName`)
- Jumlah produk yang dipesan (`Quantity`)
- Jumlah unit produk yang tersedia di stok (`UnitsInStock`)

Hasil query kemudian diurutkan berdasarkan `ProductID` secara ascending.

QUERY 2

```
SELECT p.ProductID, p.ProductName, p.UnitPrice, od.Quantity, p.UnitsInStock
-> FROM Products p
-> LEFT OUTER JOIN OrderDetails od
-> ON p.ProductID = od.ProductID
-> ORDER BY p.ProductID ASC;
```

HASIL

14 rows in set (0.102 sec)

```
MariaDB [company]> SELECT p.ProductID, p.ProductName, p.UnitPrice, od.Quantity, p.UnitsInStock
-> FROM Products p
-> LEFT OUTER JOIN OrderDetails od
-> ON p.ProductID = od.ProductID
-> ORDER BY p.ProductID ASC;
```

ProductID	ProductName	UnitPrice	Quantity	UnitsInStock
2	Chang	19.00	15	17
5	Chef Anton's Gumbo Mix	21.35	65	50
21	Sir Rodney's Scones	10.00	50	17
27	Schoggi Schokolade	43.90	15	49
32	Mascarpone Fabioli	32.00	6	9
37	Gravad lax	26.00	10	11
39	Chartreuse verte	18.00	6	69
41	Jack's Clam Chowder	9.65	16	85
53	Perth Pasties	32.80	15	36
57	Ravioli Angelo	19.50	1	17
62	Tarte au sucre	49.30	15	76
65	Hot Pepper Sauce	21.05	NULL	15
70	Longlife Tofu	10.00	21	4
74	Original Frankfurter	13.00	NULL	32
77	Outback Lager	15.00	6	15
77	Outback Lager	15.00	12	15

16 rows in set (0.124 sec)

ANALISIS

1. `SELECT p.ProductID, p.ProductName, p.UnitPrice, od.Quantity, p.UnitsInStock`

- Bagian ini menentukan kolom-kolom yang akan ditampilkan dalam hasil query. Kolom-kolom tersebut adalah:
 - `p.ProductID`: ID produk dari tabel `Products`.
 - `p.ProductName`: Nama produk dari tabel `Products`.
 - `p.UnitPrice`: Harga unit produk dari tabel `Products`.
 - `od.Quantity`: Jumlah produk yang dipesan dari tabel `OrderDetails`. Jika tidak ada pesanan untuk produk tertentu, nilai ini akan `NULL`.
 - `p.UnitsInStock`: Jumlah unit produk yang tersedia di stok dari tabel `Products`.

2. FROM Products p

- Bagian ini menentukan tabel utama yang digunakan dalam query, yaitu tabel `Products`. Alias `p` digunakan untuk mempermudah referensi ke tabel ini.

3. LEFT OUTER JOIN OrderDetails od ON p.ProductID = od.ProductID

- Bagian ini menghubungkan tabel `Products` dengan tabel `OrderDetails` menggunakan klausa `LEFT OUTER JOIN`. Hubungan dibuat berdasarkan kolom `ProductID` yang ada di kedua tabel tersebut.
- Alias `od` digunakan untuk tabel `OrderDetails`.
- `LEFT OUTER JOIN` memastikan semua baris dari tabel `Products` akan ditampilkan, bahkan jika tidak ada entri yang sesuai di tabel `OrderDetails`. Jika tidak ada kecocokan, kolom dari tabel `OrderDetails` (seperti `Quantity`) akan mengandung nilai `NULL`.

4. ORDER BY p.ProductID ASC

- Bagian ini menentukan urutan hasil query berdasarkan kolom `ProductID` dari tabel `Products` secara ascending (menaik).

Penjelasan Fungsionalitas

Query ini mengambil data dari dua tabel: `Products` dan `OrderDetails`. Untuk setiap produk, query ini akan menampilkan:

- ID produk (`ProductID`)
- Nama produk (`ProductName`)
- Harga unit produk (`UnitPrice`)
- Jumlah produk yang dipesan (`Quantity` , jika ada; jika tidak, akan bernilai `NULL`)
- Jumlah unit produk yang tersedia di stok (`UnitsInStock`)

Hasil query kemudian diurutkan berdasarkan `ProductID` secara ascending.

QUERY 3

```
SELECT p.ProductID, p.ProductName, p.UnitPrice, od.Quantity, p.UnitsInStock
-> FROM Products p
-> RIGHT OUTER JOIN OrderDetails od
-> ON p.ProductID = od.ProductID
-> ORDER BY p.ProductID ASC;
```

HASIL

```
14 rows in set (0.102 sec)
```

```
MariaDB [company]> SELECT p.ProductID, p.ProductName, p.UnitPrice, od.Quantity, p.UnitsInStock
-> FROM Products p
-> LEFT OUTER JOIN OrderDetails od
-> ON p.ProductID = od.ProductID
-> ORDER BY p.ProductID ASC;
```

ProductID	ProductName	UnitPrice	Quantity	UnitsInStock
2	Chang	19.00	15	17
5	Chef Anton's Gumbo Mix	21.35	65	50
21	Sir Rodney's Scones	10.00	50	17
27	Schoggi Schokolade	43.90	15	49
32	Mascarpone Fabioli	32.00	6	9
37	Gravad lax	26.00	10	11
39	Chartreuse verte	18.00	6	69
41	Jack's Clam Chowder	9.65	16	85
53	Perth Pasties	32.80	15	36
57	Ravioli Angelo	19.50	1	17
62	Tarte au sucre	49.30	15	76
65	Hot Pepper Sauce	21.05	NULL	15
70	Longlife Tofu	10.00	21	4
74	Original Frankfurter	13.00	NULL	32
77	Outback Lager	15.00	6	15
77	Outback Lager	15.00	12	15

```
16 rows in set (0.124 sec)
```

ANALISIS

1. SELECT p.ProductID, p.ProductName, p.UnitPrice, od.Quantity, p.UnitsInStock

- Bagian ini menentukan kolom-kolom yang akan ditampilkan dalam hasil query. Kolom-kolom tersebut adalah:
 - p.ProductID: ID produk dari tabel Products.
 - p.ProductName: Nama produk dari tabel Products.
 - p.UnitPrice: Harga unit produk dari tabel Products.
 - od.Quantity: Jumlah produk yang dipesan dari tabel OrderDetails.
 - p.UnitsInStock: Jumlah unit produk yang tersedia di stok dari tabel Products.

2. FROM Products p

- Bagian ini menentukan tabel utama yang digunakan dalam query, yaitu tabel Products. Alias p digunakan untuk mempermudah referensi ke tabel ini.

3. RIGHT OUTER JOIN OrderDetails od ON p.ProductID = od.ProductID

- Bagian ini menghubungkan tabel Products dengan tabel OrderDetails menggunakan klausa RIGHT OUTER JOIN. Hubungan dibuat berdasarkan kolom ProductID yang ada di kedua tabel tersebut.
- Alias od digunakan untuk tabel OrderDetails.
- RIGHT OUTER JOIN memastikan semua baris dari tabel OrderDetails akan ditampilkan, bahkan jika tidak ada entri yang sesuai di tabel Products. Jika tidak ada kecocokan, kolom dari tabel

`Products` (seperti `ProductID`, `ProductName`, `UnitPrice`, dan `UnitsInStock`) akan mengandung nilai `NULL`.

4. **ORDER BY p.ProductID ASC**

- Bagian ini menentukan urutan hasil query berdasarkan kolom `ProductID` dari tabel `Products` secara ascending (menaik). Namun, karena `ProductID` dari tabel `Products` dapat bernilai `NULL` jika tidak ada kecocokan, hasil urutannya mungkin tidak sepenuhnya intuitif.

Penjelasan Fungsionalitas

Query ini mengambil data dari dua tabel: `Products` dan `OrderDetails`. Untuk setiap entri di tabel `OrderDetails`, query ini akan menampilkan:

- ID produk (`ProductID`)
- Nama produk (`ProductName`)
- Harga unit produk (`UnitPrice`)
- Jumlah produk yang dipesan (`Quantity`)
- Jumlah unit produk yang tersedia di stok (`UnitsInStock`)

Jika tidak ada kecocokan di tabel `Products` untuk entri di tabel `OrderDetails`, kolom dari tabel `Products` akan bernilai `NULL`.

Hasil query kemudian diurutkan berdasarkan `ProductID` dari tabel `Products` secara ascending. Namun, karena `RIGHT OUTER JOIN` digunakan, urutan `ProductID` dapat mengandung nilai `NULL`.

Query 4

```
SELECT ProductsID FROM products UNION SELECT ProductID FROM OrderDetails;
```

Extra options

ProductID

2
5
65
74
70
21
27
77
32
37
39
41
53
57
62

- SELECT untuk memilih kolom mana saja Yang ingin ditampilkan dan dari tabel mana kolom tersebut dipilih.
- Product ID: Adalah nama kolom yang dipilih untuk digabungkan .
- From Products = untuk memilih dari tabel mana saja data kolomnya akan digabung.
- UNION = untuk melakukan dua SELECT, data Yang tampil adalah hasil gabungan dari tabel Products dengan OrderDetails. Tapi tampilannya tidak menampilkan duplikat dan hanya tampilan Distinct (Hanya menampilkan masing-masing Satu).
- SELECT untuk memilih kolom apa saja yang ingin ditampilkan
- ProductID= adalah nama kolom yang dipilih untuk digabungkan.
- From sales untuk memilih dari tabel mana saja data kolomnya akan diambil Hasilnya merupakan hasil gabungan dari tabel Productsdummy dan Sales dari ProductID merupakan kolom Yang datanya diambil namun data duplikat terhapus/tidak ditampilkan karena tampilan undan adalah Distinct

Query 5

```
SELECT ProductID FROM products UNION ALL SELECT ProductID FROM OrderDetails;
```

Hasil

+-----+	
ProductID	
+-----+	
2	
5	
65	
74	
70	
21	
27	
77	
32	
37	
39	
41	
53	
57	
62	
53	
77	
27	
39	
77	
2	
32	
21	
37	
41	
57	
62	

65
74
70
21
27
77
32
37
39
41
53
57
62
53
77
27
39
77
2
32
21
37
41
57
62
70

- SELECT = untuk memilih kolom mana saja yang ingin ditampilkan/digabung.
- FROM Products = untuk memilih dari tabel mana saja yang data kolomnya akan digabung
- ProductID = adalah nama kolom yang dipilih untuk digabungkan
- UNION ALL = untuk menggabung suatu kolom dan tabel. Dan menampilkan semua Data termasuk yang duplikat.
- SELECT = untuk memilih kolom mana saja yang ingin di tampilkan/digabung
- ProductID = adalah nama kolom yang dipilih untuk digabungkan
- FROM OrderDetails = untuk memilih dari tabel mana saja yang data kolomnya akan digabung
- Hasilnya = semua data termasuk data duplikat akan di tampilkan semua. Dari tabel Products dan OrderDetails

Query 6

Kode Program

```
SELECT ProductID FROM products INTERSECT SELECT ProductID FROM OrderDetails;
```

Hasil

ProductID
2
70
21
27
77
32
37
39
41
53
57
62

- SELECT = untuk memilih kolom mana saja yang ingin ditampilkan/digabung
- Product ID = adalah nama kolom yang dipilih untuk digabungkan.
- From Products = untuk memilih dari tabel mana saja yang data kolomnya akan digabung
- INTERSECT = untuk menampilkan gabungan dari tabel Products dan OrderDetails tapi hanya data yang sama/identik
- SELECT = untuk memilih kolom mana saja yang ingin ditampilkan / digabung
- ProductID= adalah nama kolom yang dipilih untuk digabungkan.
- From OrderDetails = untuk memilih dari tabel mana saja yang data kolomnya akan digabung
- Hasilnya = hanya data yang sama dengan antarkolom pada dua tabel yang tampil.

nomor 7

Query

```
SELECT productID FROM productsdummy
EXCEPT
SELECT productID FROM sales;
```

Hasil query

```

MariaDB [company]> SELECT productID FROM productsdummy
-> EXCEPT
-> SELECT productID FROM sales;
+-----+
| productID |
+-----+
|          4 |
+-----+
1 row in set (0.016 sec)

```

Analisis

SELECT : untuk memilih kolom mana saja yang ingin ditampilkan / digabung.

productID : nama kolom yang dipilih untuk digabungkan.

FROM productsdummy : untuk memilih dari tabel mana saja yang data kolomnya akan digabungkan.

EXCEPT : untuk menampilkan gabungan dari productsdummy dan sales tapi hanya data yang tidak sama.

SELECT : untuk memilih kolom mana saja yang ingin ditampilkan / digabung.

productID : nama kolom yang dipilih untuk digabungkan.

FROM sales : untuk memilih dari tabel mana saja yang data kolomnya akan di gabung.

HASILNYA : hanya 4 yang tidak memiliki data yang sama.

TABEL productsdummy

```

MariaDB [company]> select * from productsdummy;
+-----+-----+-----+
| productID | ProductName          | unitsInStock |
+-----+-----+-----+
|          1 | Chang                | 17           |
|          2 | Chef Anton's Gumbo Mix | 50           |
|          3 | Sir Rodney's Scones  | 17           |
|          4 | Schoggi Schokolade  | 49           |
|          6 | Mascarpone Fabioli   | 9            |
+-----+-----+-----+
5 rows in set (0.001 sec)

```

TABEL Sales

```
MariaDB [company]> select * from sales;
```

salesID	productID	unitPrice	quantity	discount
10255	1	26.20	15	0
10255	2	10.40	12	0.1
10257	3	35.10	25	0
10258	1	26.20	25	0
10258	6	15.20	50	0.2
10259	2	10.40	24	0.1

6 rows in set (0.001 sec)

nomor 8

Query

```
SELECT pegawai.ID, pegawai>Nama, departemen.IdDep, departemen.Departemen
FROM pegawai
CROSS JOIN departemen
ORDER BY pegawai.ID ASC;
```

Hasil query

```
MariaDB [company]> SELECT pegawai.ID, pegawai>Nama, departemen.IdDep, departemen.D
epartemen
-> FROM pegawai
-> CROSS JOIN departemen ORDER BY pegawai.ID ASC;
```

ID	Nama	IdDep	Departemen
3121	Jejio Abimanyu	102	Marketing
3121	Jejio Abimanyu	101	HRD
3215	Gianna Nitya	102	Marketing
3215	Gianna Nitya	101	HRD

4 rows in set (0.001 sec)

Analisis

SELECT : untuk memilih kolom mana saja yang ingin ditampilkan / digabung.

pegawai.ID, pegawai>Nama : kolom ID dan nama dari tabel pegawai dipilih.

departemen.IdDep, departemen.Departemen : kolom IdDep dan Departemen dari tabel departemen dipilih untuk digabungkan.

FROM pegawai : untuk memilih dari tabel mana data kolomnya dipilih yaitu pegawai.

CROSS JOIN : untuk menggabung data kolom secara bersilang (masing-masing kolom digabungkan).

departemen : nama tabel yang digabung.

ORDER BY pegawai.ID ASC : untuk mengurutkan data berdasarkan kolom ID dari pegawai secara

(Ascending) terkecil ke terbesar.

HASILNYA : masing-masing data kolom digabung satu sama lain, Jejo dengan HRD dan Marketing, Gianna dengan HRD dan Marketing.

TABEL pegawai

```
MariaDB [company]> select * from pegawai;
+-----+-----+
| ID    | Nama          |
+-----+-----+
| 3121  | Jejo Abimanyu |
| 3215  | Gianna Nitya  |
+-----+-----+
2 rows in set (0.001 sec)
```

TABEL departemen

```
MariaDB [company]> select * from departemen;
+-----+-----+
| IdDep | Departemen    |
+-----+-----+
| 101   | HRD           |
| 102   | Marketing     |
+-----+-----+
2 rows in set (0.001 sec)
```