

PEMROGRAMAN BASIS DATA

Subquery

Kamarudin, M.Kom

kamarudin@amikom.ac.id

<http://coding4ever.net/>

<https://github.com/rudi-krsoftware/open-retail>

Subquery

- ✓ Query di dalam query atau nested query.
- ✓ Subquery adalah perintah SELECT yang berada di dalam perintah SELECT lainnya.
- ✓ Subquery (query dalam) dieksekusi terlebih dahulu sebelum query utama (query luar).
- ✓ Hasil subquery digunakan oleh query utama.
- ✓ Perintah yang digunakan

```
SELECT select_list  
FROM table  
WHERE expr operator (SELECT select_list FROM table)
```

```
SELECT select_list, (SELECT single_field FROM table)  
FROM table
```

Panduan Penggunaan Subquery

- ✓ Perintah subquery ditulis diantara tanda kurung “(.....)”.
- ✓ Perintah subquery diletakkan di sebelah kondisi (klausa WHERE atau HAVING).
- ✓ Perintah subquery juga bisa dijadikan sebagai column expression.
- ✓ Self-Contained SubQueries, subquery yang tidak tergantung dengan query utama.
- ✓ Correlated SubQueries, subquery yang mempunyai referensi ke query utama.
- ✓ Single-row operator digunakan dengan single-row subquery.
- ✓ Multiple-row operator digunakan dengan multiple-row subquery.


Self-Contained SubQueries

Subquery yang tidak tergantung dengan query utama.

```
SELECT name, stok, purchasing_price ,selling_price,category_id
FROM Products
WHERE category_id = (
    SELECT category_id FROM Categories
    WHERE description = 'beauty'
)
```



	name	stok	purchasing_price	selling_price	category_id
1	Babyliss Pro Curl	10	675000,00	689000,00	CG-003
2	NXY Soft Matte Lip Cream	50	125000,00	130000,00	CG-003
3	Manicure Set 777	20	195000,00	199000,00	CG-003
4	Rosemary Essential Oil	50	55000,00	60000,00	CG-003
5	Naked 3 Palette	50	235000,00	245000,00	CG-003




	category_id
1	CG-003

Correlated SubQueries

Subquery yang mempunyai referensi ke query utama.

```
SELECT name, stok, purchasing_price ,selling_price,category_id
FROM products
WHERE category_id = (
    SELECT category_id FROM Categories
    WHERE category_id = products.category_id
)
```

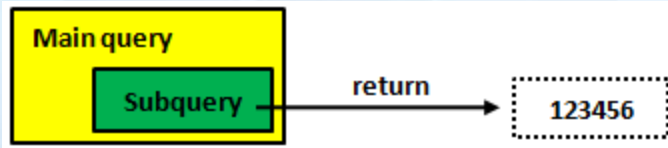


	name	stok	purchasing_price	selling_price	category_id
1	Snapback Hat NY	50	135000.00	140000.00	CG-001
2	Roberto Cavalli Sunglass	100	160000.00	165000.00	CG-001
3	Porsche Design Sunglass	25	265000.00	275000.00	CG-001
4	Giorgio Agnelli Wallet	50	145000.00	150000.00	CG-001
5	Levis Design Walet	50	160000.00	175000.00	CG-001

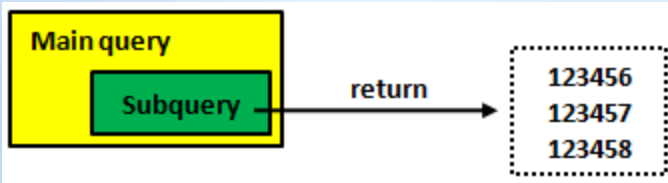
	category_id
1	CG-001

Type Subquery

✓ Single-row subquery



✓ Multiple-row subquery



Single-row Subquery

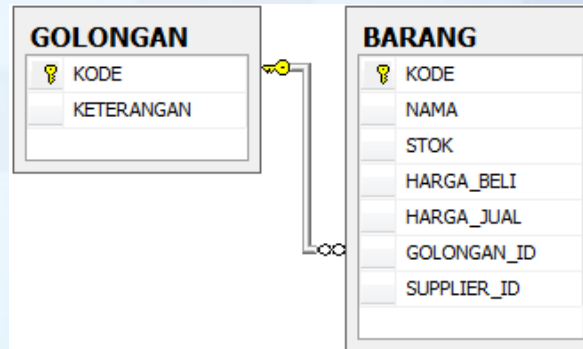
- ✓ Single-row subquery menghasilkan satu baris/record data dari perintah SELECT terdalam.



- ✓ Operator yang digunakan.

Operator	Keterangan
=	Sama dengan
<>	Tidak sama dengan
>	Lebih besar dari
<	Lebih kecil dari
>=	Lebih besar atau sama dengan
<=	Lebih kecil atau sama dengan

Contoh Single-row Subquery



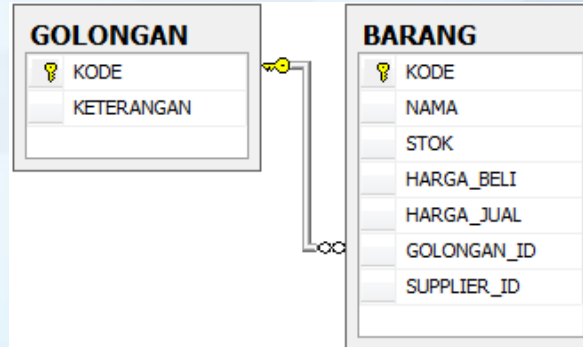
*“Menampilkan data barang (kode, nama, dan stok) yang masuk dalam kategori/golongan **HARDWARE NEW**”*

```
SELECT kode, nama, stok
FROM barang
WHERE golongan_id =
(SELECT kode FROM golongan WHERE keterangan = 'HARDWARE NEW')
```

Query diatas dapat diartikan sebagai berikut :

- ✓ Subquery terdalam menghasilkan “NEW”.
- ✓ Kemudian dari hasil eksekusi subquery terdalam dijadikan syarat untuk query utama.

Contoh Single-row Subquery



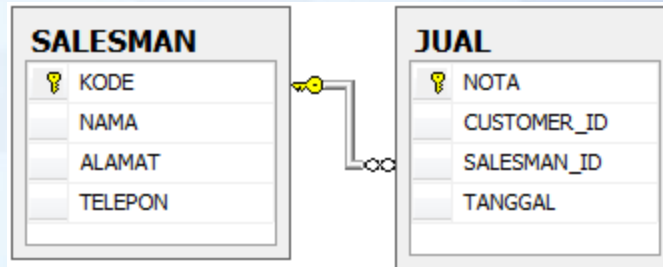
*“Menampilkan data barang (kode, nama, dan stok) yang jumlah stoknya diatas stok barang dengan nama **Printer Epson L360**”.*

```
SELECT kode, nama, stok FROM barang
WHERE stok >
(SELECT stok FROM barang WHERE nama = 'Printer Epson L360')
```

Query diatas dapat diartikan sebagai berikut :

- ✓ Subquery terdalam menghasilkan “21”.
- ✓ Kemudian dari hasil eksekusi subquery terdalam dijadikan syarat untuk query utama.

Contoh Single-row Subquery



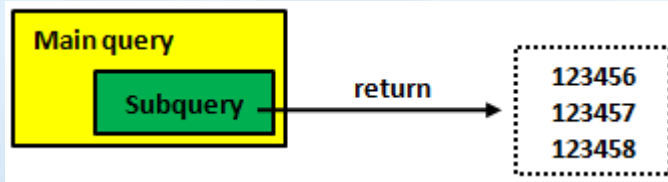
*“Menampilkan daftar salesman yang jumlah transaksi penjualannya di atas penjualan salesman dengan nama **ARIS**”*

	salesman	JUMLAH PENJUALAN
1	ADMIN INNOVA	380
2	CANDRA	15
3	DANIEL	11
4	HAFIS	15
5	KABIHAR	22
6	NIKEN	377
7	YOGI	101

```
SELECT salesman.nama AS salesman, COUNT(*) AS [JUMLAH PENJUALAN]
FROM jual INNER JOIN salesman ON jual.salesman_id = salesman.kode
GROUP BY salesman.nama
HAVING COUNT(*) >
(SELECT COUNT(*) FROM jual INNER JOIN salesman ON jual.salesman_id
= salesman.kode WHERE salesman.nama = 'ARIS')
```

Multiple-row Subquery

- ✓ Menghasilkan lebih dari satu row.



- ✓ Operator yang digunakan

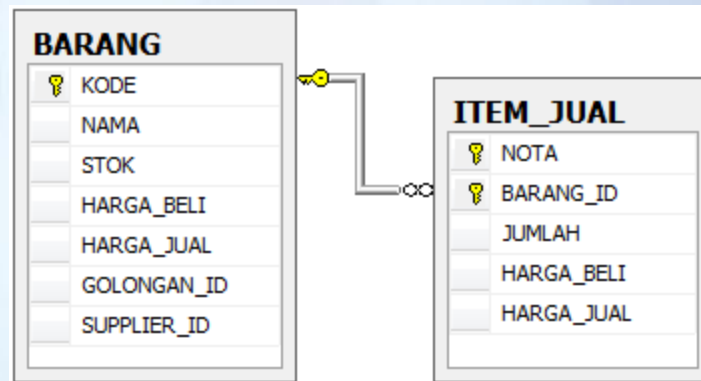
Operator	Keterangan
IN, = ANY	Sama dengan nilai hasil subquery
> ANY	Lebih besar dari nilai minimum hasil subquery. Contoh (1,2,3) berarti lebih besar dari 1
< ANY	Lebih kecil dari nilai maksimum hasil subquery. Contoh (1,2,3) berarti lebih kecil dari 3
> ALL	Lebih besar dari nilai maksimum hasil subquery. Contoh (1,2,3) berarti lebih besar dari 3
< ALL	Lebih kecil dari nilai minimum hasil subquery. Contoh (1,2,3) berarti kecil dari 1

Tabel : XYZ

KODE	NILAI
A	100
B	150
C	200
D	250
E	300
F	350
G	370

```
SELECT kode
FROM xyz
WHERE nilai IN (
    SELECT nilai FROM xyz
    WHERE kode IN ('B', 'D', 'E')
)
```

Contoh Multiple-row Subquery

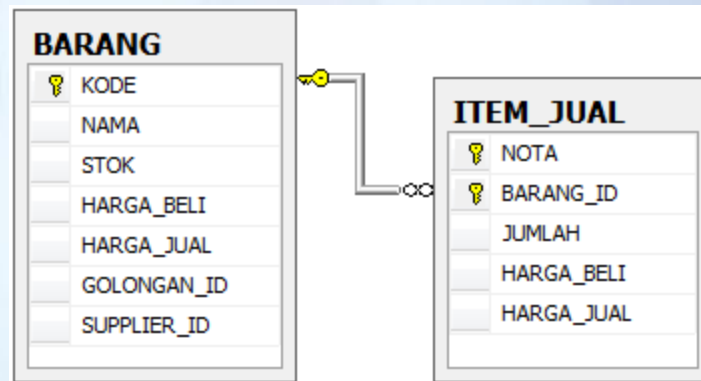


“Menampilkan data barang yang belum pernah terjual”

	barang	stok
1	Tinta 100 ml Blueprint EPSON Biru Muda	0
2	Casing Cpu Samsung	1
3	Keyboard USB Flexible	0
4	LAN Tester	0
5	Crimping Tool 3	0

```
SELECT nama AS barang, stok
FROM barang
WHERE kode NOT IN (
    SELECT barang_id FROM item_jual
)
```

Contoh Multiple-row Subquery



*“Menampilkan data **Flashdisk** yang sudah terjual”*

	barang	stok
1	Flashdisk 4Gb Toshiba	0
2	Flashdisk 4Gb Kingston	0
3	Flashdisk 8Gb Toshiba	0
4	Flashdisk 4Gb PNY All Merk	0

```
SELECT nama AS barang, stok
FROM barang
WHERE nama LIKE '%Flashdisk%' AND kode IN (
    SELECT barang_id FROM item_jual
)
```

Subquery dengan fungsi aggregate

Subquery juga bisa dikombinasikan dengan fungsi aggregate.

staff	
	name
	salary

“Menampilkan semua staf yang gajinya di atas rata”

```
SELECT name, salary
FROM staff
WHERE salary > (
    SELECT AVG(salary) FROM staff
)
```

Subquery menggunakan Operator EXIST

Operator EXIST dibuat khusus untuk subquery. Operator ini digunakan untuk mengecek apakah subquery menghasilkan baris atau tidak. Operator EXIST akan mengembalikan nilai true jika subquery mengembalikan minimal satu baris.

Contoh query di bawah ini akan menampilkan supplier yang mempunyai produk dengan harga kurang dari \$20.

```
SELECT supplier_name
FROM suppliers
WHERE EXISTS (
    SELECT * FROM products
    WHERE supplier_id =
    suppliers.supplier_id AND Price < 20
)
```

	supplier_id	supplier_name	address
1	1	Exotic Liquid	49 Gilbert St.
2	2	New Orleans Cajun Delights	P.O. Box 78934
3	3	Grandma Kelly's Homestead	707 Oxford Rd.
4	4	Tokyo Traders	9-8 Sekimai Musashino-shi

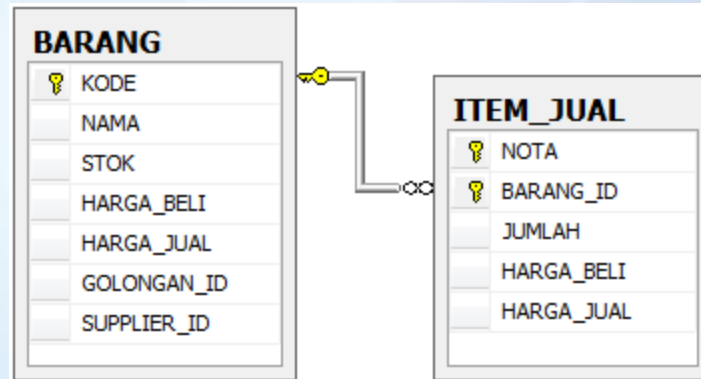
Suppliers

	product_id	product_name	supplier_id	price
1	1	Chais	1	18.00
2	2	Chang	1	19.00
3	3	Aniseed Syrup	1	10.00
4	4	Chef Anton's Cajun Seasoning	2	22.00
5	5	Chef Anton's Gumbo Mix	2	21.35

Products

Subquery sebagai column exp

Subquery juga bisa dijadikan sebagai *column expression*



“Menampilkan jumlah terjual untuk masing-masing barang”

	barang	jumlah terjual
1	Mb 478 DDR2 PCI-E Amptron	400
2	Mb LGA DDR2 PCI-E Amptron G31	315
3	Cooler 478 Cool Master	198
4	Fan Casing Hitam 8in Standart	NULL
5	Kabel HDD ATA 133	NULL

```
SELECT nama AS barang, (SELECT SUM(jumlah) FROM item_jual WHERE
barang_id = barang.kode) AS [jumlah terjual]
FROM barang
```