

## **About This Course:**

Kamu akan memahami konsep penggabungan tabel menjadi sumber data baru dengan mekanisme INNER JOIN dan UNION. Sumber data baru ini bisa berfungsi untuk pelaporan atau analisa data menjadi informasi yang lebih lengkap dan bermanfaat.

## **Table of Content :**

1. Penggabungan Tabel dari Relasi Kolom
2. Inner Join
3. Union
4. Mini Project

## **START**

## **Module : Fundamental SQL Using Inner Join dan Union**

## Module : Konsep Penggabungan Tabel dari Relasi Kolom

### Penggabungan Tabel dari Relasi Kolom

Memahami bagaimana menghubungkan dua buah tabel pada suatu database yang kolomnya memiliki relasi.

Kemampuan menggabungkan beberapa tabel pada suatu database, merupakan ciri khas dan kekuatan yang dimiliki RDBMS. Contohnya, menggabungkan tabel data penjualan dengan tabel lain di database yang berisi detail produk, mekanisme penggabungan ini disebut JOIN.

**Contoh menggabungkan tabel:** ms\_tabel\_warna & ms\_item\_kategori

**Alasan:** Penggabungan dilakukan karena ingin mengetahui informasi tidak hanya warna tetapi juga jenis/kategori dari setiap nama barang apakah tergolong sayuran atau buah.

Table: ms\_item\_warna

nama_barang	warna
apel	merah
bayam	hijau
daun bawang	hijau
duku	kuning pekat
durian	kuning
gandum	coklat
jambu air	merah
jeruk	oranye

Table: ms\_item\_kategori

nama_item	kategori
bayam	sayuran
belimbing	buah
duku	buah
durian	buah
gandum	buah
jamur	sayuran
jambu air	buah
jeruk	buah

Tabel diatas merupakan data yang digunakan untuk penggabungan, dan dapat terlihat setiap tabel terdiri dari 8 baris data dan 2 kolom. (jumlah baris dan kolom tidak mempengaruhi & dapat dilakukan penggabungan)

Berikut adalah Langkah-langkah untuk melakukan penggabungan Tabel dengan Key Columns:

1. Melakukan Query untuk melihat keseluruhan isi dari kolom tabel ms\_item\_kategori dan ms\_item\_warna.

Penerapan:

```
1. SELECT * FROM ms_item_kategori;  
2. SELECT * FROM ms_item_warna;
```

Hasil:

nama_item	kategori
bayam	sayuran
belimbing	buah
duku	buah
durian	buah
gandum	buah
jamur	sayuran
jambu air	buah
jeruk	buah

nama_barang	warna
apel	merah
bayam	hijau
daun bawang	hijau
duku	kuning pekat
durian	kuning
gandum	coklat
jambu air	merah
jeruk	oranye

2. Hal pertama yang perlu dilakukan sebelum menggabungkan kedua tabel adalah mengidentifikasi kolom dari kedua tabel yang memiliki kecocokan atau informasi yang sama. Dari kedua tabel diatas dapat dilihat bahwa:

**Kolom nama\_barang dan nama\_item memiliki kecocokan**

3. Mecocokkan key column, mana saja isi data di atas yang terdapat pada kedua table?

**Bayam, Gandum, Duku, Jambu Air, Jeruk**

4. Menentukan key column, setelah dicocokkan isi data dari kedua kolom dapat disimpulkan bahwa kedua tabel memiliki hubungan/relasi pada kolom **nama\_barang** dan **nama\_item** karena banyak kecocokan dari isi data.

**Kedua Kolom tersebut dinamakan kolom kunci atau key columns.**

5. Menggabungkan **Tabel dengan Key Columns**

Penggabungan dua tabel baru menggunakan query SELECT, dilakukan dengan cara berikut:

**Cara JOIN seperti ini disebut juga cara JOIN menggunakan operator koma.**



**Penerapan:**

```
1 SELECT * FROM ms_item_kategori, ms_item_warna
2 WHERE nama_barang = nama_item;
```

## Hasil:

Dapat terlihat hasil dari penggabungan dua tabel tersebut adalah tabel baru dengan empat kolom dan enam baris data. Dimana data yang memiliki kecocokan saja yang ditampilkan.

Penting:

penggabungan ini bersifat sementara, artinya tabel asli di database tidak mengalami perubahan tabel baru hasil penggabungan juga tidak serta merta disimpan di database.

nama_item	kategori	nama_barang	warna
bayam	sayuran	bayam	hijau
duku	buah	duku	kuning pekat
durian	buah	durian	kuning
gandum	buah	gandum	coklat
jambu air	buah	jambu air	merah
jeruk	buah	jeruk	oranye

## Penjelasan Hasil Penggabungan Tabel

Berikut adalah gambar untuk memahami bagaimana proses penggabungan tabel terjadi:

ms\_item\_kategori

nama_item	kategori
bayam	sayuran
belimbing	buah
duku	buah
durian	buah
gandum	buah
jamur	sayuran
jambu air	buah
jeruk	buah

ms\_item\_warna

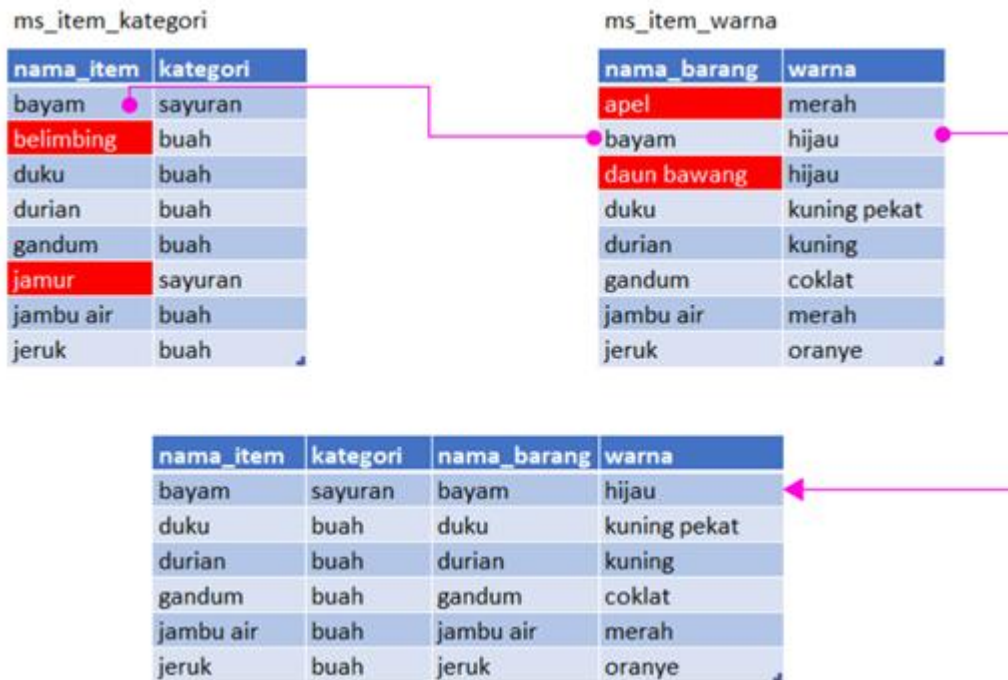
nama_barang	warna
apel	merah
bayam	hijau
daun bawang	hijau
duku	kuning pekat
durian	kuning
gandum	coklat
jambu air	merah
jeruk	oranye

nama_item	kategori	nama_barang	warna
bayam	sayuran	bayam	hijau
duku	buah	duku	kuning pekat
durian	buah	durian	kuning
gandum	buah	gandum	coklat
jambu air	buah	jambu air	merah
jeruk	buah	jeruk	oranye

Berdasarkan tabel diatas terlihat bahwa baris data yang muncul hanyalah baris data yang isi datanya terdapat di kedua key columns, dan data yang isinya tidak cocok akan di-exclude dan tidak akan muncul di tabel hasil.

## Penjelasan Proses Penggabungan Tabel

Perhatikan Gambar berikut ini:



Lihat panah warna magenta, proses dimulai dari baris teratas key column (columns `nama_item`) pada tabel pertama 'tabel `ms_item_kategori`', kemudian dia akan mencari pasangannya pada key columns (`nama_barang`) pada tabel kedua 'tabel `ms_item_warna`'. Setelah ditemukan akan digabungkan menjadi satu tabel baru hasil penggabungan.

Pencocokan ini didasari oleh bagian kondisi (conditional clause) pada query yang ditandai kuning:

```
SELECT * FROM ms_item_kategori, ms_item_warna  
WHERE nama_barang = nama_item;
```

Setelah seluruh data ditemukan dengan kondisi ini, dan jika tidak menentukan spesifik kolom maka seluruh kolom data dari kedua tabel akan dimunculkan dengan isi/baris data yang sama dan terdapat di kedua key column digabungkan menjadi satu baris pada tabel baru.

Isi data yang ditandai dengan warna merah adalah data yang tidak memiliki pasangan sehingga tidak bisa digabungkan, dan dengan demikian tidak memiliki data gabungan pada hasil akhir.

Cara pencocokan dan penggabungan disebut dengan INNER JOIN - dimana isi dari key column kedua tabel harus cocok satu sama lain baru dapat digabungkan.

## Bagaimana jika urutan tabel berubah?

Jika urutan tabel dirubah, tapi dengan kondisi yang sama. Maka urutan kolom akan berbeda, tapi data yang dihasilkan tetap 6 baris.

Dimana dua kolom pertama adalah dari tabel ms\_item\_warna, dan dua kolom berikutnya dari tabel ms\_item\_kategori. Hal ini sesuai dengan urutan nama tabel yang diketikkan setelah FROM.

Contohnya:

```
1 SELECT * FROM ms_item_warna, ms_item_kategori
2 WHERE nama_barang = nama_item;
```

Hasil:

nama_barang	warna	nama_item	kategori
bayam	hijau	bayam	sayuran
duku	kuning pekat	duku	buah
durian	kuning	durian	buah
gandum	coklat	gandum	buah
jambu air	merah	jambu air	buah
jeruk	oranye	jeruk	buah

## Menggunakan Prefix Nama Tabel

Kita akan tetap menggunakan wildcard(\*). Dengan menggunakan prefix nama tabel, dimana dengan merinci prefix nama tabel ini, dimungkinkan untuk menentukan urutan kolom dari tabel mana yang muncul duluan.

Penerapan:

```
1 SELECT ms_item_kategori.*, ms_item_warna.*
2 FROM ms_item_warna, ms_item_kategori
3 WHERE nama_barang = nama_item;
```

Hasil:

nama_item	kategori	nama_barang	warna
bayam	sayuran	bayam	hijau
duku	buah	duku	kuning pekat
durian	buah	durian	kuning
gandum	buah	gandum	coklat
jambu air	buah	jambu air	merah
jeruk	buah	jeruk	oranye

## Menggabungkan Tanpa Kondisi

Dalam beberapa kasus real problem, sering kali terdapat permasalahan tertentu dimana kita ingin menggabungkan tabel tanpa kondisi atau tanpa relasi antar kolom.

Bagaimana data akan terbentuk?

### Penerapan:

```
1 SELECT * FROM ms_item_kategori, ms_item_warna;
```

### Hasil:

Data akan terbentuk sebagai berikut:

data yang terbentuk berjumlah 64 baris data karena 8 baris tabel ms\_item\_kategori dikalikan (X) 8 baris dari tabel ms\_item\_warna.

nama\_item dan nama\_barang menjadi acak. Data sulit sulit dimengerti.

Aku mengumam mengerti. Terlihat banyak sekali hasil yang keluar, ini **dikarenakan setiap baris data pada kedua tabel akan dihubungkan satu sama lain - tanpa ada hubungan.**

Jumlah enam puluh empat baris data ini adalah hasil perkalian dari jumlah data dari kedua tabel, dimana masing-masing memiliki delapan baris data. Cara menggabungkan kedua tabel seperti ini disebut dengan mekanisme cross join.

nama_item	kategori	nama_barang	warna
bayam	sayuran	apel	merah
belimbing	buah	apel	merah
duku	buah	apel	merah
durian	buah	apel	merah
gandum	buah	apel	merah
jamur	sayuran	apel	merah
jambu air	buah	apel	merah
jeruk	buah	apel	merah
bayam	sayuran	bayam	hijau
belimbing	buah	bayam	hijau
duku	buah	bayam	hijau
durian	buah	bayam	hijau
gandum	buah	bayam	hijau



## Kesimpulan

Bahasa SQL diciptakan untuk sistem database relasional atau RDBMS. Aspek Relasional ini yang menjadi fitur penting dan unggul, dimana beberapa tabel dapat digabungkan menjadi satu sumber data baru.

Cara menggabungkan antar tabel disebut sebagai mekanisme JOIN. Terdapat dua tipe JOIN yaitu:

INNER JOIN terjadi jika kedua tabel digabungkan melalui kolom kunci atau key column. Syarat penggabungan adalah dimana isi data dari key column tabel yang satu harus dapat dicocokkan dengan isi data dari key column tabel yang lain.

CROSS JOIN terjadi dari penggabungan tabel tanpa kondisi, dan menghasilkan seluruh penggabungan data seperti proses perkalian.

Untuk bab ini, inner join maupun cross join dilakukan dengan cara menggunakan operator koma dan pengkondisian where. Pada bab selanjutnya, akan terlihat penggunaan inner join dengan menggunakan keyword INNER JOIN ... ON.

## Module : Konsep INNER JOIN

Memahami penggabungan dua tabel yang memiliki relasi pada suatu database dengan menggunakan keyword INNER JOIN.

### Pendahuluan

Menggabungkan dua tabel dengan Keyword INNER JOIN memberi **manfaat** lebih ketika pada suatu query melibatkan beberapa join sehingga penggunaan operator koma menjadi sangat terbatas.

### Syntax INNER JOIN

**Tujuan JOIN** adalah menggabungkan dua atau lebih tabel yang berbeda di dalam database.

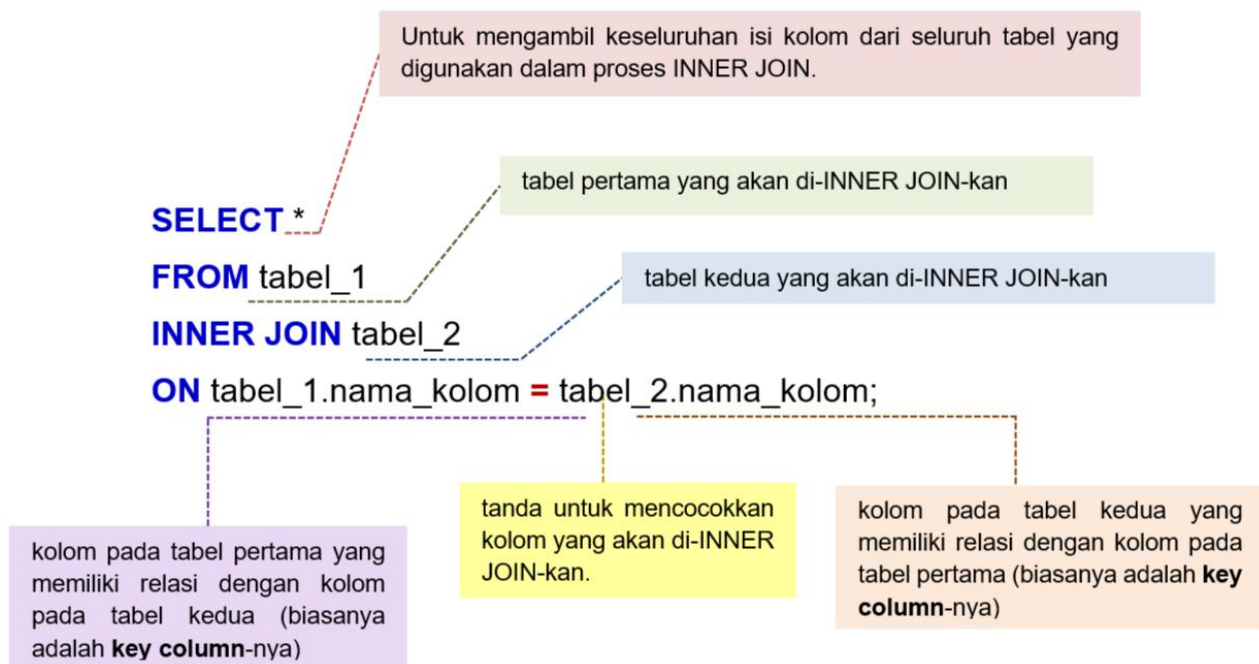
JOIN hanya bisa dilakukan apabila tabel-tabel tersebut memiliki key column yang sama.

Terdapat beberapa tipe JOIN:

- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL JOIN
- SELF JOIN

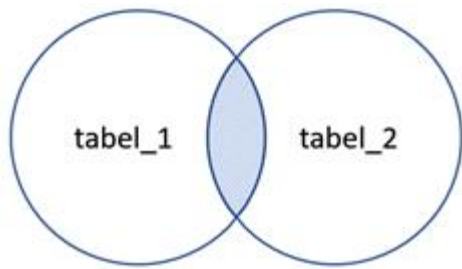
Setiap tipe join memiliki tujuan dan hasil query yang berbeda.

Pada INNER JOIN, baris dari kedua atau lebih tabel akan dibandingkan untuk mengecek baris-baris mana saja yang cocok satu sama lain berdasarkan kondisi JOIN yang ditentukan.



**\*note:** Penggunaan Prefix pada tabel ditujukan agar terdapat kejelasan (clarity) kolom pada tabel mana yang dijadikan acuan dalam proses INNER JOIN.

Jika masing-masing total tabel tersebut dianalogikan sebagai dua himpunan maka proses INNER JOIN ekuivalen dengan INTERSECTION (Irisan) antara dua himpunan.



Proses INNER JOIN memiliki konsep penggabungan untuk records pada kedua tabel yang bernilai sama.

### Tugas Praktek: Menggunakan INNER JOIN (1/3)

**Menggabungkan tabel ms\_item\_warna dengan ms\_item\_kategori** menggunakan Syntaks INNER JOIN.

Hasil yang diperoleh dengan penggunaan SELELCT... FROM... INNER JOIN... ON... adalah sama dengan penerapan SELECT... FROM... WHERE...

**Penerapan:**

```
1 SELECT * FROM ms_item_warna
2 INNER JOIN ms_item_kategori
3 ON ms_item_warna.nama_barang = ms_item_kategori.nama_item;
```

**Hasil:**

nama_barang	warna	nama_item	kategori
bayam	hijau	bayam	sayuran
duku	kuning pekat	duku	buah
durian	kuning	durian	buah
gandum	coklat	gandum	buah
jambu air	merah	jambu air	buah
jeruk	oranye	jeruk	buah

## Tugas Praktek: Menggunakan INNER JOIN (2/3)

### Menggabungkan tabel tr\_penjualan dan ms\_produk

Berikut adalah contoh lanjutan penggunaan INNER JOIN, sama seperti penggabungan dengan koma(,) langkah yang diterapkan juga sama yaitu Langkah penggabungan tabel menggunakan key columns.

#### 1. Melakukan query wildcard untuk melihat isi data dari semua tabel tersebut

Penerapan:

```
1 SELECT * FROM tr_penjualan;
2 SELECT * FROM ms_produk;
```

Hasil:

Console		Graphic				
kode_transaksi	kode_pelanggan	no_urut	kode_produk	nama_produk	qty	harga
tr-001	dqlabcust07	1	prod-01	Kotak Pensil DQLab	5	62500
tr-001	dqlabcust07	2	prod-03	Flash disk DQLab 32 GB	1	100000
tr-001	dqlabcust07	3	prod-09	Buku Planner Agenda DQLab	3	92000
tr-001	dqlabcust07	4	prod-04	Flashdisk DQLab 32 GB	3	40000
tr-002	dqlabcust01	1	prod-03	Gift Voucher DQLab 100rb	2	100000
tr-002	dqlabcust01	2	prod-10	Sticky Notes DQLab 500 sheets	4	55000
tr-002	dqlabcust01	3	prod-07	Tas Travel Organizer DQLab	1	48000
tr-003	dqlabcust03	1	prod-02	Flashdisk DQLab 64 GB	2	55000
tr-004	dqlabcust03	1	prod-10	Sticky Notes DQLab 500 sheets	5	55000
tr-004	dqlabcust03	2	prod-04	Flashdisk DQLab 32 GB	4	40000
tr-005	dqlabcust05	1	prod-09	Buku Planner Agenda DQLab	3	92000
tr-005	dqlabcust05	2	prod-01	Kotak Pensil DQLab	1	62500
tr-005	dqlabcust05	3	prod-04	Flashdisk DQLab 32 GB	2	40000
tr-006	dqlabcust02	1	prod-05	Gift Voucher DQLab 250rb	4	250000
tr-006	dqlabcust02	2	prod-08	Gantungan Kunci DQLab	2	15800

no_urut	kode_produk	nama_produk	harga
1	prod-01	Kotak Pensil DQLab	62500
2	prod-02	Flashdisk DQLab 64 GB	55000
3	prod-03	Gift Voucher DQLab 100rb	100000
4	prod-04	Flashdisk DQLab 32 GB	40000
5	prod-05	Gift Voucher DQLab 250rb	250000
6	prod-06	Pulpen Multifunction + Laser DQLab	92500
7	prod-07	Tas Travel Organizer DQLab	48000
8	prod-08	Gantungan Kunci DQLab	15800
9	prod-09	Buku Planner Agenda DQLab	92000
10	prod-10	Sticky Notes DQLab 500 sheets	55000

#### 2. Mengidentifikasi Key Column tabel tr\_penjualan dan ms\_produk

Berdasarkan kecocokan informasi data dari kolom di kedua tabel maka,

**Key columns:** kode\_produk dari tabel tr\_penjualan dengan kode\_produk dari tabel ms\_produk.

#### 3. Menggabungkan kedua tabel menggunakan INNER JOIN berdasarkan key columns yang teridentifikasi.

## Penerapan:

```
1 SELECT * FROM tr_penjualan
2 INNER JOIN ms_produk
3 ON tr_penjualan.kode_produk = ms_produk.kode_produk;
```

## Hasil:

kode_transaksi	kode_pelanggan	no_urut	kode_produk	nama_produk	qty	harga	no_urut	kode_produk	nama_produk	harga
tr-001	dqlabcust07	1	prod-01	Kotak Pensil DQLab	5	62500	1	prod-01	Kotak Pensil DQLab	62500
tr-001	dqlabcust07	2	prod-03	Flash disk DQLab 32 GB	1	100000	3	prod-03	Gift Voucher DQLab 100rb	100000
tr-001	dqlabcust07	3	prod-09	Buku Planner Agenda DQLab	3	92000	9	prod-09	Buku Planner Agenda DQLab	92000
tr-001	dqlabcust07	4	prod-04	Flashdisk DQLab 32 GB	3	40000	4	prod-04	Flashdisk DQLab 32 GB	40000
tr-002	dqlabcust01	1	prod-03	Gift Voucher DQLab 100rb	2	100000	3	prod-03	Gift Voucher DQLab 100rb	100000
tr-002	dqlabcust01	2	prod-10	Sticky Notes DQLab 500 sheets	4	55000	10	prod-10	Sticky Notes DQLab 500 sheets	55000
tr-002	dqlabcust01	3	prod-07	Tas Travel Organizer DQLab	1	48000	7	prod-07	Tas Travel Organizer DQLab	48000
tr-003	dqlabcust03	1	prod-02	Flashdisk DQLab 64 GB	2	55000	2	prod-02	Flashdisk DQLab 64 GB	55000
tr-004	dqlabcust03	1	prod-10	Sticky Notes DQLab 500 sheets	5	55000	10	prod-10	Sticky Notes DQLab 500 sheets	55000
tr-004	dqlabcust03	2	prod-04	Flashdisk DQLab 32 GB	4	40000	4	prod-04	Flashdisk DQLab 32 GB	40000
tr-005	dqlabcust05	1	prod-09	Buku Planner Agenda DQLab	3	92000	9	prod-09	Buku Planner Agenda DQLab	92000
tr-005	dqlabcust05	2	prod-01	Kotak Pensil DQLab	1	62500	1	prod-01	Kotak Pensil DQLab	62500
tr-005	dqlabcust05	3	prod-04	Flashdisk DQLab 32 GB	2	40000	4	prod-04	Flashdisk DQLab 32 GB	40000
tr-006	dqlabcust02	1	prod-05	Gift Voucher DQLab 250rb	4	250000	5	prod-05	Gift Voucher DQLab 250rb	250000
tr-006	dqlabcust02	2	prod-08	Gantungan Kunci DQLab	2	15000	8	prod-08	Gantungan Kunci DQLab	15000

## 4. Memahami Proses dan Hasil query dengan INNER JOIN

Proses INNER JOIN akan mencocokkan key kolom tabel tr\_penjualan dengan key kolom ms\_produk, JIKA value dari key kolom sama-sama ada di kedua tabel, maka baris itu akan dikembalikan sebagai hasil query dan membentuk satu tabel yang berisi seluruh kolom dari kedua tabel, sedangkan jika value key kolom hanya terdapat di satu kolom, maka baris ini tidak akan ditampilkan.

Contoh prod-06 ada di tabel ms\_produk tetapi tidak ada di tr\_penjualan sehingga saat penggabungan tabel dengan INNER JOIN, data prod-06 dari tabel ms\_produk tidak akan muncul di hasil query.

Sebenarnya juga, pada bagian SELECT, kamu boleh memilih kolom mana saja yang ingin kamu tampilkan setelah kedua tabel di-JOIN.



## Tugas Praktek: Menggunakan INNER JOIN (3/3)

### Melakukan INNER JOIN – Memilih Beberapa Kolom Untuk Ditampilkan

Jika kita ingin menampilkan beberapa kolom saja dari tabel hasil penggabungan, maka Syntax SQL dari INNER JOIN dinyatakan sebagai berikut:

Untuk memilih beberapa kolom dari kedua tabel yang digunakan dalam proses INNER JOIN melalui penggunaan **prefix** masing-masing nama tabel.

**SELECT** tabel1.nama\_kolom1, tabel1.nama\_kolom2, ..., tabel2.nama\_kolom2, ....

**FROM** tabel1

tabel pertama yang akan di-INNER JOIN-kan

**INNER JOIN** tabel2

tabel kedua yang akan di-INNER JOIN-kan

**ON** tabel\_1.nama\_kolom1 = tabel2.nama\_kolom1;

kolom pada tabel pertama yang memiliki relasi dengan kolom pada tabel kedua (biasanya adalah **key column**-nya)

tanda untuk mencocokkan kolom yang akan di-INNER JOIN-kan.

kolom pada tabel kedua yang memiliki relasi dengan kolom pada tabel pertama (biasanya adalah **key column**-nya)

### Bagian query,

```
SELECT tabel1.nama_kolom1, tabel1.nama_kolom2, ..., tabel2.nama_kolom2, ....  
FROM tabel1
```

Menghendaki pemilihan kolom mana saja dari kedua tabel yang akan digabungkan. Tahap ini diperlukan penggunaan prefix nama tabelnya untuk menghindari error.

Error terjadi karena jika kita ingin menampilkan kolom dengan nama yang sama di kedua tabel, tanpa menggunakan prefix maka akan terjadi error karena ambiguitas.

### Bagian query,

```
INNER JOIN tabel2
```

Digunakan untuk menggabungkan tabel1 dengan tabel2

### Bagian query,

```
ON tabel_1.nama_kolom1 = tabel2.nama_kolom1;
```

Adalah acuan penggabungan tabel1 dan tabel2 berdasarkan key column. Kedua kolom dengan tingkat kecocokan yang tinggi pada masing-masing tabel disebut dengan key column.

## 1. Melakukan INNER JOIN – Memilih beberapa kolom yang ingin ditampilkan

Sekarang silakan dipraktikkan pada code editor untuk menggabungkan tabel tr\_penjualan dan ms\_produk. kolom yang ditampilkan dari tabel tr\_penjualan adalah kode\_transaksi, kode\_pelanggan, kode\_produk, qty. Untuk tabel ms\_produk tampilkan kolom nama\_produk dan harga.

Tambahkan kolom total yang merupakan hasil perkalian setiap baris pada kolom harga di tabel ms\_produk dengan kolom qty di tabel tr\_penjualan

Urutan kolom yang ditampilkan adalah kode\_transaksi, kode\_pelanggan, kode\_produk, nama\_produk, harga, qty, dan total.

### Penerapan:

```
1 SELECT tr_penjualan.kode_transaksi, tr_penjualan.kode_pelanggan, tr_penjualan.kode_produk,
ms_produk.nama_produk, ms_produk.harga, tr_penjualan.qty, ms_produk.harga * tr_penjualan.qty AS
total
2 FROM tr_penjualan
3 INNER JOIN ms_produk
4 ON tr_penjualan.kode_produk = ms_produk.kode_produk;
```

### Hasil:

kode_transaksi	kode_pelanggan	kode_produk	nama_produk	harga	qty	total
tr-001	dqlabcust07	prod-01	Kotak Pensil DQLab	62500	5	312500
tr-001	dqlabcust07	prod-03	Gift Voucher DQLab 100rb	100000	1	100000
tr-001	dqlabcust07	prod-09	Buku Planner Agenda DQLab	92000	3	276000
tr-001	dqlabcust07	prod-04	Flashdisk DQLab 32 GB	40000	3	120000
tr-002	dqlabcust01	prod-03	Gift Voucher DQLab 100rb	100000	2	200000
tr-002	dqlabcust01	prod-10	Sticky Notes DQLab 500 sheets	55000	4	220000
tr-002	dqlabcust01	prod-07	Tas Travel Organizer DQLab	48000	1	48000
tr-003	dqlabcust03	prod-02	Flashdisk DQLab 64 GB	55000	2	110000
tr-004	dqlabcust03	prod-10	Sticky Notes DQLab 500 sheets	55000	5	275000
tr-004	dqlabcust03	prod-04	Flashdisk DQLab 32 GB	40000	4	160000
tr-005	dqlabcust05	prod-09	Buku Planner Agenda DQLab	92000	3	276000
tr-005	dqlabcust05	prod-01	Kotak Pensil DQLab	62500	1	62500
tr-005	dqlabcust05	prod-04	Flashdisk DQLab 32 GB	40000	2	80000
tr-006	dqlabcust02	prod-05	Gift Voucher DQLab 250rb	250000	4	1000000
tr-006	dqlabcust02	prod-08	Gantungan Kunci DQLab	15800	2	31600

## Kesimpulan

Pada chapter INNER JOIN ini kita telah mempelajari bagaimana menggabungkan dua tabel berdasarkan key coumn-nya.

- Pada tahap awal kita menggunakan INNER JOIN untuk menggabungkan keseluruhan yang dimiliki oleh kedua tabel tersebut.
- Selanjutnya, kita menerapkan INNER JOIN dengan menggunakan prefix nama tabel untuk memilih kolom-kolom mana saja yang akan ditampilkan pada tabel hasil penggabungan.

## Module : Konsep UNION

Memahami penggabungan dua tabel dengan menggunakan keyword UNION.

### Pendahuluan

Kita akan memahami lebih lanjut bagian yang lebih kompleks, yaitu mengakses data dalam database dengan SQL.

Pembelajaran kita hari ini akan dimulai dengan studi kasus praktik jadi learning by doing. Ini adalah data berupa hasil penjualan suatu store. Akan tetapi ketika coba di cek di database, terdapat 2 tabel penjualan, yaitu:

- Tabel A berisi data transaksi untuk kode transaksi tr-001 sampai tr-003
- Tabel B berisi data transaksi untuk kode transaksi tr-004 sampai tr-006

Bagaimana cara yang cepat dan efektif untuk membuat analisis dari kedua tabel tersebut menggunakan SQL?

### Penggabungan hasil SELECT secara “Vertikal”

Dalam kasus tabel seperti diatas, kita perlu menggunakan metode UNION agar kedua tabel tadi menjadi satu tabel.

UNION adalah operator SQL yang digunakan untuk menggabungkan hasil dari 2 atau lebih SELECT statement secara vertikal. Dengan Catatan:

- a) Setiap hasil dari SELECT statement yang akan digabungkan (UNION) memiliki jumlah kolom yang sama
- b) Kolom tersebut juga harus memiliki tipe data yang sama
- c) Kolom tersebut memiliki urutan posisi yang sama

Berikut format **syntax-nya**:

```
SELECT nama_kolom1, nama_kolom2, ..., nama_kolomn FROM tabel1  
UNION  
SELECT nama_kolom1, nama_kolom2, ..., nama_kolomn FROM tabel2
```



## Tugas Praktek : Menggunakan UNION

### Pengecekan Tabel yang akan Digabungkan

Pertama tampilkan seluruh kolom dalam kedua tabel dengan SELECT seluruh kolom dari tabel\_A dan tabel\_B. berikut adalah Datanya:

#### Penerapan:

```
1 SELECT * FROM tabel_A;  
2 SELECT * FROM tabel_B;
```

#### Hasil:

kode_transaksi	kode_pelanggan	no_urut	kode_produk	nama_produk	qty	harga	total
tr-001	dqlabcust07	1	prod-01	Kotak Pensil DQLab	5	62500	312500
tr-001	dqlabcust07	2	prod-03	Flash disk DQLab 32 GB	1	100000	100000
tr-001	dqlabcust07	3	prod-09	Buku Planner Agenda DQLab	3	92000	276000
tr-001	dqlabcust07	4	prod-04	Flashdisk DQLab 32 GB	3	40000	120000
tr-002	dqlabcust01	1	prod-03	Gift Voucher DQLab 100rb	2	100000	200000
tr-002	dqlabcust01	2	prod-10	Sticky Notes DQLab 500 sheets	4	55000	220000
tr-002	dqlabcust01	3	prod-07	Tas Travel Organizer DQLab	1	48000	48000
tr-003	dqlabcust03	1	prod-02	Flashdisk DQLab 64 GB	2	55000	110000
tr-004	dqlabcust03	1	prod-10	Sticky Notes DQLab 500 sheets	5	55000	275000
tr-004	dqlabcust03	2	prod-04	Flashdisk DQLab 32 GB	4	40000	160000
tr-005	dqlabcust05	1	prod-09	Buku Planner Agenda DQLab	3	92000	276000
tr-005	dqlabcust05	2	prod-01	Kotak Pensil DQLab	1	62500	62500
tr-005	dqlabcust05	3	prod-04	Flashdisk DQLab 32 GB	2	40000	80000
tr-006	dqlabcust02	1	prod-05	Gift Voucher DQLab 250rb	4	250000	1000000
tr-006	dqlabcust02	2	prod-08	Gantungan Kunci DQLab	2	15800	31600

Sesuai dengan syarat untuk penggabungan dengan UNION yang telah dijelaskan tadi bahwa:

- jumlah kolom tabel\_A dan tabel\_B adalah sama
- kolom-kolom pada tabel\_A dan tabel\_B memiliki tipe data yang sama, dan
- kolom-kolom pada tabel\_A dan tabel\_B memiliki urutan posisi yang sama.

Melalui pengecekan pada tabel\_A dan tabel\_B pastikan bahwa ketiga syarat penggabungan dengan UNION yang dinyatakan di atas terpenuhi. Langkah ini kita lakukan sebelum melanjutkan pada praktek berikutnya menggunakan UNION.

## Penggabungan tabel menggunakan UNION

Kedua tabel\_A dan tabel\_B sudah memiliki jumlah kolom yang sama, tipe data dan urutan posisi kolom juga sama, jadi kedua tabel tersebut bisa langsung digabungkan.

Jika belum sama ketiga syarat tersebut, maka harus dilakukan manipulasi ataupun penyelarasan untuk bagian yang belum sesuai.

### Penerapan:

```
1 SELECT * FROM tabel_A
2 UNION
3 SELECT * FROM tabel_B;
```

### Hasil:

kode_transaksi	kode_pelanggan	no_urut	kode_produk	nama_produk	qty	harga	total
tr-001	dqlabcust07	1	prod-01	Kotak Pensil DQLab	5	62500	312500
tr-001	dqlabcust07	2	prod-03	Flash disk DQLab 32 GB	1	100000	100000
tr-001	dqlabcust07	3	prod-09	Buku Planner Agenda DQLab	3	92000	276000
tr-001	dqlabcust07	4	prod-04	Flashdisk DQLab 32 GB	3	40000	120000
tr-002	dqlabcust01	1	prod-03	Gift Voucher DQLab 100rb	2	100000	200000
tr-002	dqlabcust01	2	prod-10	Sticky Notes DQLab 500 sheets	4	55000	220000
tr-002	dqlabcust01	3	prod-07	Tas Travel Organizer DQLab	1	48000	48000
tr-003	dqlabcust03	1	prod-02	Flashdisk DQLab 64 GB	2	55000	110000
tr-004	dqlabcust03	1	prod-10	Sticky Notes DQLab 500 sheets	5	55000	275000
tr-004	dqlabcust03	2	prod-04	Flashdisk DQLab 32 GB	4	40000	160000
tr-005	dqlabcust05	1	prod-09	Buku Planner Agenda DQLab	3	92000	276000
tr-005	dqlabcust05	2	prod-01	Kotak Pensil DQLab	1	62500	62500
tr-005	dqlabcust05	3	prod-04	Flashdisk DQLab 32 GB	2	40000	80000
tr-006	dqlabcust02	1	prod-05	Gift Voucher DQLab 250rb	4	250000	1000000
tr-006	dqlabcust02	2	prod-08	Gantungan Kunci DQLab	2	15800	31600

## Penggabungan dengan Kondisi menggunakan UNION

Bagaimana jika ingin menggabungkan tabel sesuai dengan kondisi yang diinginkan?

**Contoh**, menggabungkan tabel yang isinya kode\_pelanggan untuk pelanggan 'dqlabcust03'

### Penerapan:

Hanya perlu menambahkan WHERE di kedua SELECT statement, seperti berikut ini:

```
1 SELECT * FROM tabel_A
2 WHERE kode_pelanggan = 'dqlabcust03'
3 UNION
4 SELECT * FROM tabel_B
5 WHERE kode_pelanggan = 'dqlabcust03';
```

### Hasil:

kode_transaksi	kode_pelanggan	no_urut	kode_produk	nama_produk	qty	harga	total
tr-003	dqlabcust03	1	prod-02	Flashdisk DQLab 64 GB	2	55000	110000
tr-004	dqlabcust03	1	prod-10	Sticky Notes DQLab 500 sheets	5	55000	275000
tr-004	dqlabcust03	2	prod-04	Flashdisk DQLab 32 GB	4	40000	160000

## Bagaimana jika dari 3 kondisi ada yang tidak terpenuhi?

Permasalahan pertama, bagaimana jika jumlah kolom sama tetapi memiliki urutan posisi yang berbeda?

**Solusi:** Melakukan Penyelarasan (Conforming) Kolom

### Penjelasan:

#### Tabel Customers

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Fransiska Maria	Maria	Jl Sudirman	Jakarta	14450	Indonesia
2	Ana Helena	Ana Helena	Jl Madura	Surabaya	5021	Indonesia
3	Lily Subari	Lili	Jl Sumba	Makassar	5023	Indonesia

#### Tabel Supplier

SupplierID	ContactName	SupplierName	Address	City	PostalCode	Country
1	Yulius	Yulius Syrup	Jl Jayakarta	Jakarta	14450	Indonesia
2	Sherly Ani	Bandung Bakery	Jalan Asia Afrika	Bandung	70117	Indonesia
3	Regina Tara	Tara Pastry	Jalan Merpati	Semarang	48104	Indonesia

Jumlah kolom dari kedua tabel sama yaitu 7 kolom. Tetapi urutan posisi kolom ContactName dari kedua tabel tidak sama. Di tabel Customer, posisi kolom ContactName berada di kolom ke-3 sedangkan di tabel supplier berada di kolom ke-2.

Jika digabungkan langsung, hasilnya akan kacau yaitu seperti ini:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Fransiska Maria	Maria	Jl Sudirman	Jakarta	14450	Indonesia
2	Ana Helena	Ana Helena	Jl Madura	Surabaya	5021	Indonesia
3	Lily Subari	Lili	Jl Sumba	Makassar	5023	Indonesia
1	Yulius	Yulius Syrup	Jl Jayakarta	Jakarta	14450	Indonesia
2	Sherly Ani	Bandung Bakery	Jalan Asia Afrika	Bandung	70117	Indonesia
3	Regina Tara	Tara Pastry	Jalan Merpati	Semarang	48104	Indonesia

Maka, diperlukan **Penyelarasan** yaitu dengan:

Mengurutkan posisi kolom tersebut di SELECT Statement dan juga pilih kolom yang ingin digabungkan saja, sehingga tidak perlu semua kolom dari kedua tabel di – UNION kan.

**Penerapan:**

**Catatan:** Jika terdapat perbedaan nama kolom antara SELECT-statement pertama dan SELECT-statement kedua, maka secara default akan digunakan nama kolom dari SELECT-statement yang pertama.

```
1 SELECT CustomerName, ContactName, City, PostalCode
2 FROM Customers
3 UNION
4 SELECT SupplierName, ContactName, City, PostalCode
5 FROM Suppliers;
```

**Hasil:**

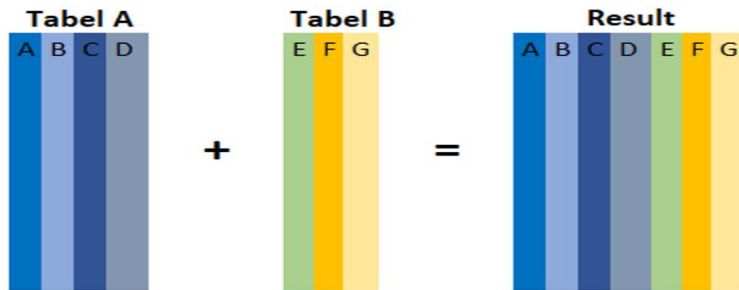
CustomerName	ContactName	City	PostalCode
Fransiska Maria	Maria	Jakarta	14450
Ana Helena	Ana Helena	Surabaya	5021
Lily Subari	Lili	Makassar	5023
Yulius Syrup	Yulius	Jakarta	14450
Bandung Bakery	Sherly Ani	Bandung	70117
Tara Pastry	Regina Tara	Semarang	48104



## Module : Perbedaan UNION dan JOIN

Perbedaan antara UNION dan JOIN adalah bagaimana tabel-tabel itu digabungkan.

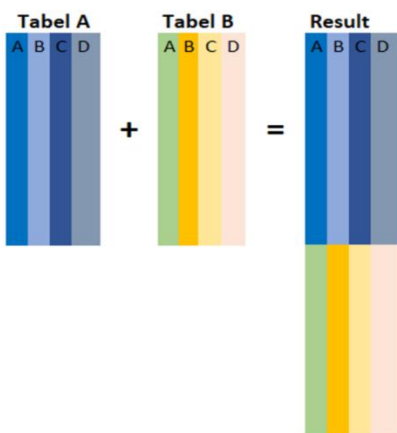
Kita menggunakan JOIN ketika akan menggabungkan tabel secara Horizontal, sehingga hasil join akan memuat kolom-kolom dari kedua atau lebih tabel yang digabungkan. Berikut adalah ilustrasinya:



```
SELECT * FROM Tabel_A  
JOIN  
SELECT * FROM Tabel_B  
ON Tabel_A.A = Tabel_B.E
```

Pada metode JOIN, penggabungan dilakukan berdasarkan key/kolom tertentu yang terdapat di tabel-tabel yang akan digabungkan dan key/kolom ini memiliki nilai yang saling terkait. Seperti yang terlihat pada gambar, Kolom A dan Kolom E merupakan key/kolom yang saling terkait sehingga kedua tabel dapat digabungkan dengan mencocokkan nilai dari kedua kolom ini. Proses JOIN tidak dapat dilakukan jika tidak terdapat key/kolom yang saling terkait di kedua atau lebih tabel yang akan digabungkan.

Kita menggunakan UNION ketika ingin menggabungkan tabel secara Vertikal, yaitu menggabungkan baris/row dari dua atau lebih tabel. Penggabungan UNION mensyaratkan bahwa jumlah kolom dari tabel yang digabungkan adalah sama dan berada diposisi yang sama pula. Berikut adalah ilustrasinya:



```
SELECT * FROM Tabel_A  
UNION  
SELECT * FROM Tabel_B
```

Pada proses UNION, tidak ada penambahan kolom tetapi jumlah baris/rows yang akan bertambah.

## Module : Kesimpulan

Pada chapter UNION ini telah dipelajari bagaimana menggabungkan dua tabel secara vertikal (bertambah barisnya). Tentunya ada syarat yang harus dipenuhi oleh kedua tabel yang digabungkan dengan UNION, yaitu:

- Setiap hasil dari SELECT statement yang akan digabungkan (UNION) memiliki jumlah kolom yang sama
- Kolom tersebut juga harus memiliki tipe data yang sama, dan
- Kolom tersebut memiliki urutan posisi yang sama.

Selain itu, mempelajari bagaimana penyelerasan kolom sehingga record/baris yang ditampilkan pada tabel hasil penggabungan memiliki arti.

Perbedaan mendasar dari JOIN dan UNION adalah JOIN menggabungkan 2 tabel atau lebih berdasarkan baris yang saling berelasi/terkait sedangkan UNION menggabungkan 2 tabel secara vertikal.

## Module : MINI PROJECT

### Project INNER JOIN

#### Soal:

Dalam database, terdapat tabel ms\_pelanggan yang berisi data - data pelanggan yang membeli produk dan tabel tr\_penjualan yang berisi data transaksi pembelian di suatu store.

Suatu hari, departemen marketing & promotion meminta bantuan untuk meng-query data-data pelanggan yang membeli produk Kotak Pensil DQLab, Flashdisk DQLab 32 GB, dan Sticky Notes DQLab 500 sheets.

Buatlah query menggunakan tabel ms\_pelanggan dan tr\_penjualan untuk mendapatkan data - data yang diminta oleh marketing yaitu kode\_pelanggan, nama\_customer, alamat.

NB: Gunakan SELECT DISTINCT untuk menghilangkan duplikasi, jika diperlukan.

Langkah Mengertikan Soal:

1. **INPUT:** table ms\_pelanggan dan tr\_penjualan
2. **OUTPUT:** kode\_pelanggan, nama\_customer, alamat
3. **KONDISI:**  
hanya mengambil data pelanggan yang membeli "Kotak Pensil DQLab, Flashdisk DQLab 32 GB, dan Sticky Notes DQLab 500 sheets"  
lakukan seleksi untuk duplicate
4. **Penggabungan:** INNER JOIN

#### Penerapan:

```
1 SELECT DISTINCT ms_pelanggan.kode_pelanggan, ms_pelanggan.nama_customer, ms_pelanggan.alamat
2 FROM ms_pelanggan
3 INNER JOIN tr_penjualan
4 ON ms_pelanggan.kode_pelanggan = tr_penjualan.kode_pelanggan
5 WHERE nama_produk = 'Kotak Pensil DQLab' OR nama_produk = 'Flashdisk DQLab 32 GB' OR nama_produk =
   'Sticky Notes DQLab 500 sheets';
```

#### Hasil:

kode_pelanggan	nama_customer	alamat
dqlabcust01	Eva Novianti, S.H.	Vila Sempilan, No. 67 - Kota B
dqlabcust03	Unang Handoko	Vila Sempilan, No. 1 - Kota B
dqlabcust05	Tommy Sinaga	Vila Permata Intan Berkilau, Blok A1/2
dqlabcust07	Agus Cahyono	Vila Gunung Seribu, Blok F4 - No. 8

## Project UNION

### Soal:

Persiapkanlah data katalog mengenai mengenai nama - nama produk yang akan dijual di suatu store. Data tersebut akan digunakan dalam meeting untuk mereview produk mana saja yang akan dilanjutkan penjualannya dan mana yang tidak akan dilanjutkan.

Siapkan hanya data produk dengan harga di bawah 100K untuk kode produk prod-1 sampai prod-5; dan dibawah 50K untuk kode produk prod-6 sampai prod-10, tanpa mencantumkan kolom no\_urut.

Saat mengecek data produk di database, terdapat 2 tabel yang sama - sama berisi data katalog, yaitu:

Langkah Mengertikan Soal:

1. **INPUT:** tabel ms\_produk\_1 dan ms\_produk\_2

2. **OUTPUT:**

perhatikan kalimat yang mengandung makna apa yang harus dibuat “saya BOLD”

‘nama\_produk diletakan pada kolom pertama karena katalog yang dibuat adalah mengenai nama produk yang akan dilanjutkan penjualannya atau tidak’

nama\_produk, kode\_produk, harga

3. **Kondisi:**

Prod-1 – prod-2 < 100K

Prod-6 – prod-10 < 50K

4. **Penggabungan:** UNION

Penerapan:

```
1 SELECT nama_produk, kode_produk, harga
2 FROM ms_produk_1
3 WHERE harga < 100000
4 UNION
5 SELECT nama_produk, kode_produk, harga
6 FROM ms_produk_2
7 WHERE harga < 50000;
```

Hasil:

nama_produk	kode_produk	harga
Kotak Pensil DQLab	prod-01	62500
Flashdisk DQLab 64 GB	prod-02	55000
Flashdisk DQLab 32 GB	prod-04	40000
Tas Travel Organizer DQLab	prod-07	48000
Gantungan Kunci DQLab	prod-08	15000



## **Module : HASIL BELAJARKU**

1. Penggabungan dua tabel dengan menggunakan WHERE clause dan teknik cross join.
2. Penggabungan dua tabel relasi dengan menggunakan INNER JOIN.
3. Penggabungan dua tabel secara vertikal dengan menggunakan UNION.
4. Mengerjakan mini project yang merupakan integrasi keseluruhan materi dan tentunya materi-materi pada modul-modul sebelumnya untuk menyelesaikan persoalan bisnis.

Dengan kemampuan ini, aku lebih pede untuk mengolah data dengan SQL. Keterampilan ini sendiri adalah 60% aktivitas awal yang akan dilakukan seorang analis. Keep Fighting