

# SQL server Immediately

---

SQL Server adalah aplikasi buatan Microsoft untuk mengelola database dengan jenis Relational Database Management System (RDBMS) yang sudah sering di gunakan di beberapa kantor. Sesuai dengan namanya yang mengandung kata `SQL` aplikasi ini menggunakan SQL (Structure Query Language) sebagai bahasa dan perintah untuk mengelola database.

Untuk bisa menggunakan SQL server tentunya kita harus paham dan familiar dengan perintah SQL yang digunakan untuk mengelola database bukan, maka dari itu kita akan belajar bagaimana menggunakan SQL server dan bagaimana menggunakan perintah SQL.

## Table of content

---

- [SQL server Immediately](#)
  - [Table of content](#)
  - [Instalasi SQL server](#)
  - [Membuat Database](#)
  - [Membuat Table](#)
  - [Perintah SQL mengelola data](#)
    - [Menambah data pada Table](#)
    - [Mengubah data pada Table](#)
    - [Menghapus data pada Table](#)
    - [Menampilkan data pada Table](#)
      - [Menampilkan seluruh data](#)
      - [Menampilkan semua data dengan filter kolom](#)
      - [Menampilkan data berdasarkan urutan dari kolom](#)
    - [Menampilkan data pada 2 Table yang berbeda](#)

---

## Instalasi SQL server

---

Sebelum menginstal SQL Server pastikan memiliki komponen di bawah ini yang diinstal atau diaktifkan di Windows :

1. Unduh dan instal `.NET Framework 4.0`. Jika komputer memiliki akses Internet, pengaturan SQL Server akan mengunduhnya ketika diperlukan.
2. Aktifkan `.NET Framework 3.5` dengan cara masuk ke Control Panel -> Programs and Features -> Turn on/off features
3. Download SQL server di Link ini atau Silahkan minta ke pemateri.

Jika sudah siap semua, silahkan jalankan aplikasi instalasi SQL server

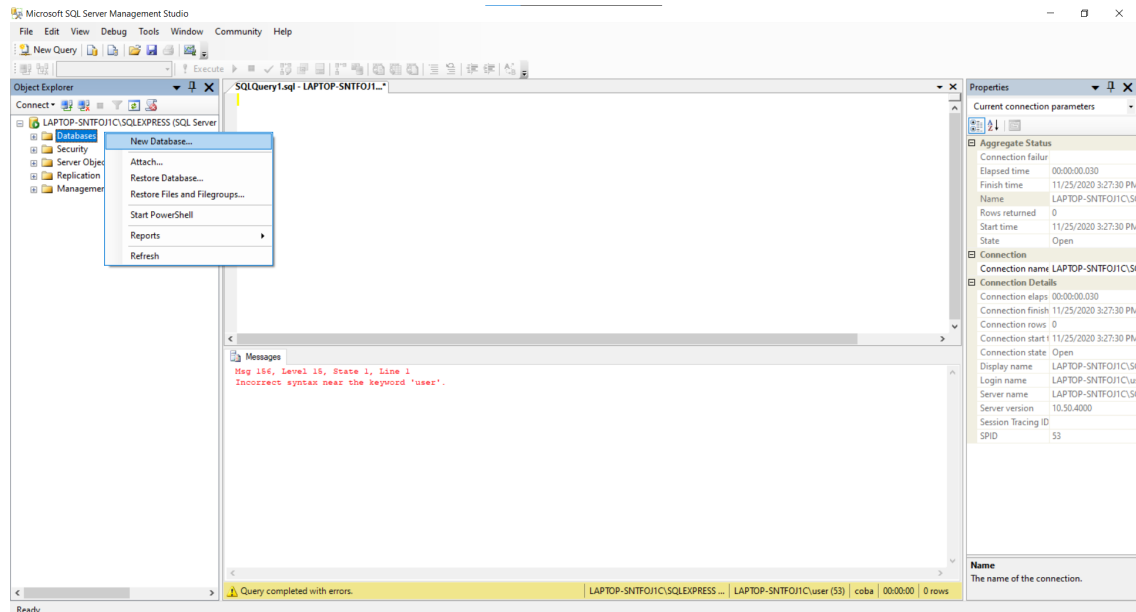
---

## Membuat Database

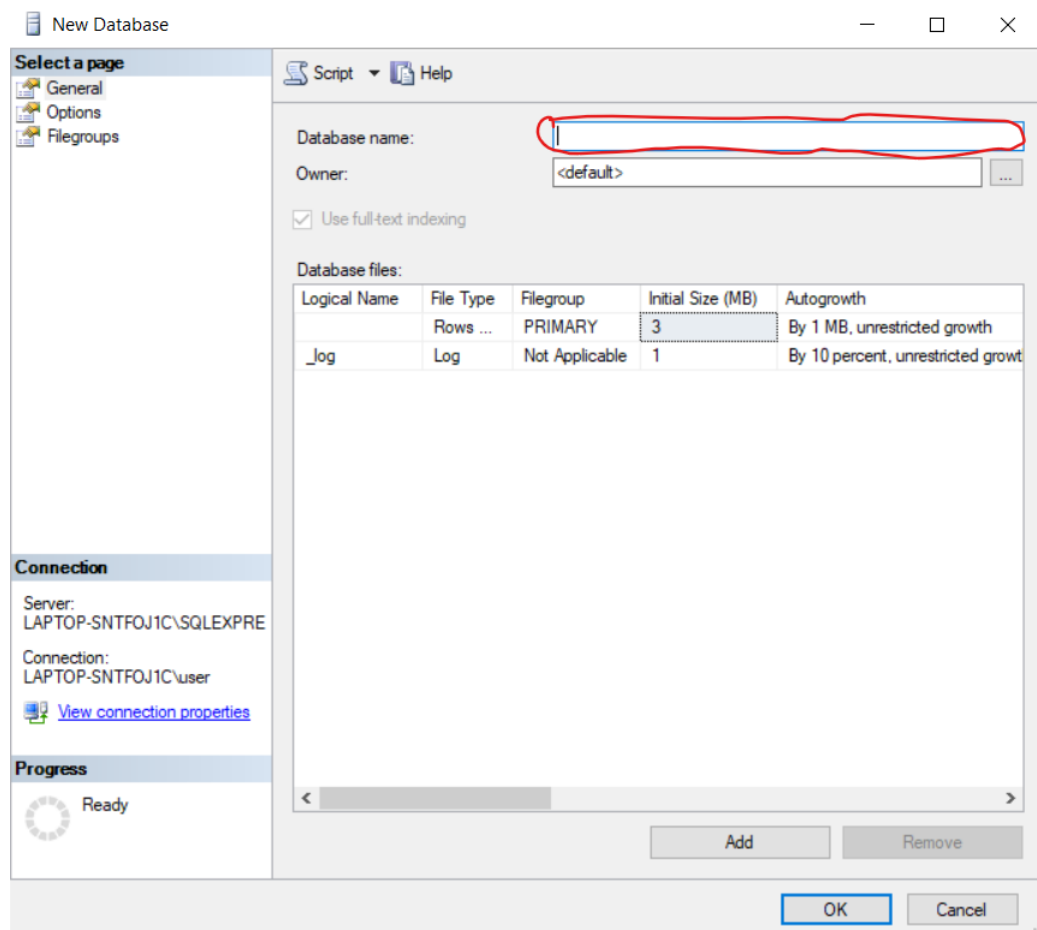
---

Langkah pertama setelah instalasi adalah membuat database, karena ini akan menjadi tempat untuk kita meletakkan data, untuk membuat database ini caranya cukup mudah

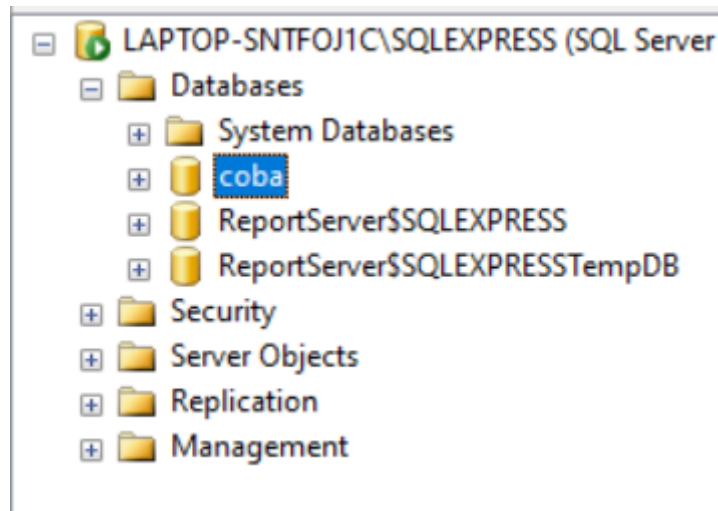
1. Pada toolbar menu pada sebelah kiri klik kanan pada folder database lalu pilih New Database



2. Kemudian akan muncul dialog baru, pada form Database name isi nama database yang diinginkan lalu klik ok



3. Maka database yang sudah dibuat akan muncul pada list folder database



Jika Database tidak muncul silahkan klik kanan folder database lalu pilih refresh jika tidak silahkan restart aplikasi SQL servernya

Kita juga bisa menggunakan perintah SQL untuk

```
create database nama_database;  
  
/* contohnya */  
create database coba;
```

---

## Membuat Table

---

Table adalah sekumpulan data dari object entah itu object mati atau object yang hidup, object yang sering kita gunakan sehari-hari bisa kita ambil contoh misal Handphone, Handphone pastinya akan memiliki ciri-ciri atau atribut yang membedakan antara Handphone dengan lainnya bukan? misalkan saja attribute yang akan digunakan sebagai kolom :

- Nama Handphone
- Merek
- Ukuran Layar
- Ukuran Baterai
- Nama Processor
- Kamera

Atribut yang ada diatas akan menjadi kolom pada Table dan akan disimpan di dalam database, namun setiap kolom Table akan memiliki tipe data yang mempresentasikan atribut/kolom bertipe apa, secara umum ada beberapa tipe data yang digunakan :

Type Data	Keterangan
<code>int</code>	tipe data untuk menyimpan dalam bentuk angka
<code>float</code>	tipe data untuk menyimpan dalam bentuk angka koma
<code>char</code>	tipe data untuk menyimpan dalam bentuk text dengan ukuran tetap
<code>varchar</code>	tipe data untuk menyimpan dalam bentuk text dengan ukuran dinamis
<code>text</code>	tipe data untuk menyimpan dalam bentuk text namun dengan konten yang banyak (biasanya untuk artikel)
<code>boolean</code>	tipe data untuk menyimpan dalam bentuk nilai <code>true</code> atau <code>false</code>
<code>date</code>	tipe data untuk menyimpan dalam bentuk tanggal dengan format Tahun-Bulan-Tanggal(YYYY-MM-DD)
<code>time</code>	tipe data untuk menyimpan dalam bentuk waktu dengan format Jam:Menit:Detik(hh-mm-ss)
<code>datetime</code>	tipe data untuk menyimpan dalam bentuk tanggal dan waktu dengan format Tahun-Bulan-Tanggal Jam:Menit:Detik(YYYY-MM-DD hh:mm:ss)

Jadi dari kasus Handphone diatas kita bisa mendekarasikan `tipe data` pada atribut yang ada, sehingga seperti ini :

- Nama Handphone (`varchar`)
- Merek (`varchar`)
- Ukuran Layar (`int`)
- Ukuran Baterai (`int`)
- Nama Processor (`varchar`)
- Kamera (`int`)

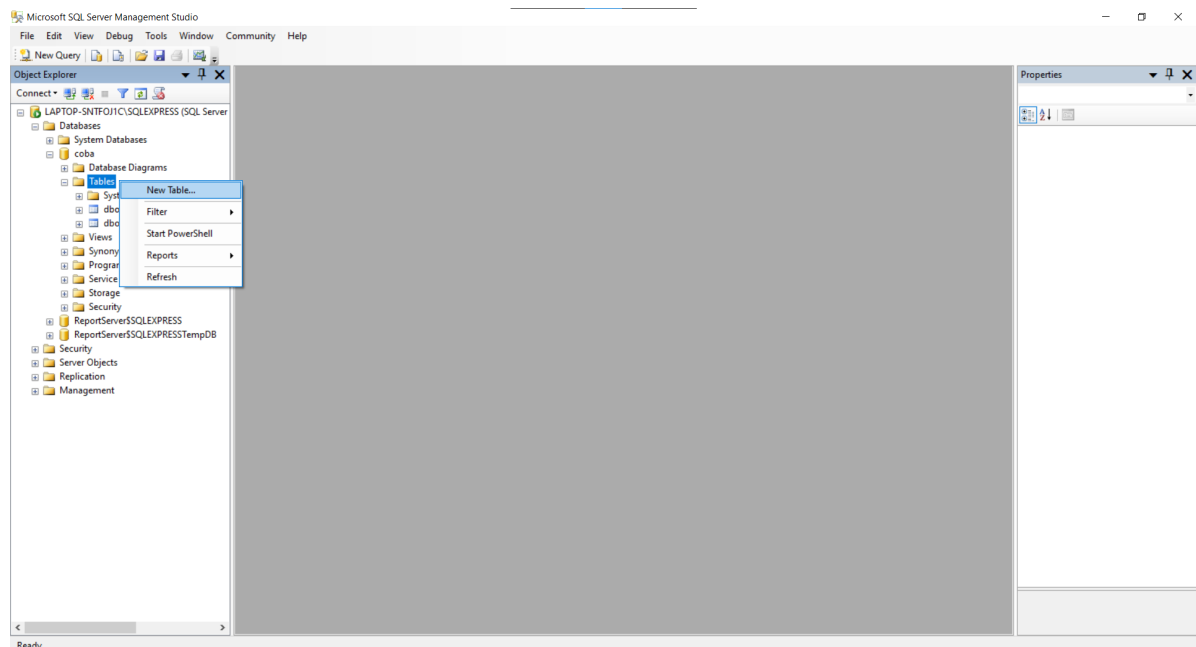
Namun kasus tersebut belum bisa dianggap selesai, kenapa bisa seperti itu? kita sudah menentukan tipe data pada attribute/kolom, namun kita akan menemukan kesulitan apabila ada `Nama handphone` yang sama bukan?

Kita membutuhkan kode unik untuk memberikan identitas meskipun `Nama Handphone` memiliki nama yang sama, kode unik ini disebut dengan `Primary key`, untuk bisa menambah kode unik ini kita akan sederhanakan menjadi `id` saja :

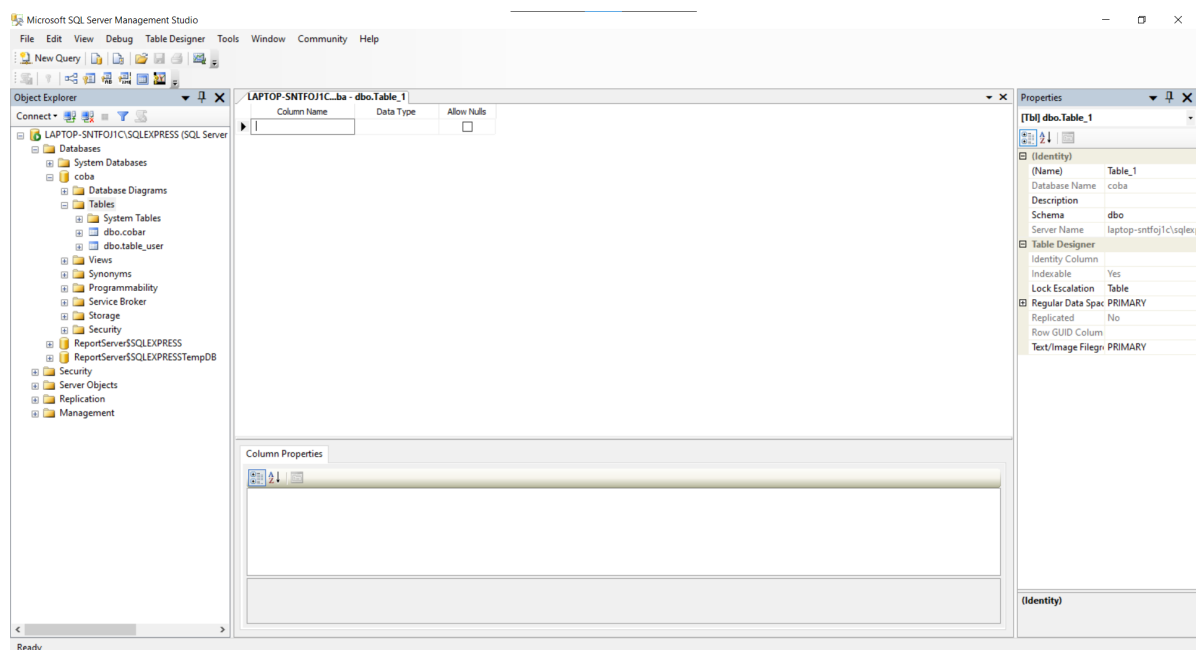
- **`id (int)`** sebagai `Primary key`
- Nama Handphone (`varchar`)
- Merek (`varchar`)
- Ukuran Layar (`int`)
- Ukuran Baterai (`int`)
- Nama Processor (`varchar`)
- Kamera (`int`)

Perlu di ingat setiap table akan memiliki `id` unik masing-masing, dan sangat diwajibkan ketika membuat table segera menambahkan `Primary key` karena akan berpengaruh ketika mengelola data

Langkah selanjutnya adalah kita buat `Table handphone` di SQL server dengan memilih `folder Database` yang sebelumnya kita buat lalu `klik kanan pada folder Table` :



Kemudian akan muncul halaman baru yang menampilkan form pengisian table :



Isi nama kolom pada **Column Name** , Tipe data pada **Data type** dan centang **Allow Nulls** apabila data tersebut diijinkan untuk kosong, dan untuk menambah kolom baru form akan membuat baris baru jika sudah mengisi kolom sebelumnya. Jadi pastikan selesaikan pembuatan kolom lalu bisa lanjut seperti ini :

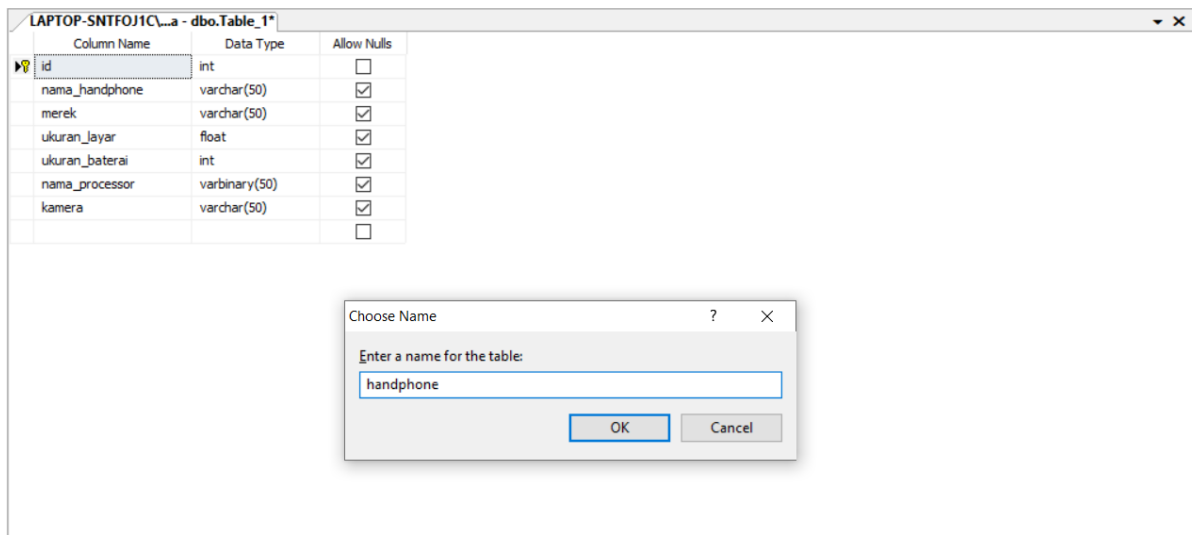
LAPTOP-SNTFOJ1C\...a - dbo.Table_1*			
	Column Name	Data Type	Allow Nulls
▶	id	int	<input type="checkbox"/>
	nama_handphone	varchar(50)	<input checked="" type="checkbox"/>
	merek	varchar(50)	<input checked="" type="checkbox"/>
	ukuran_layar	float	<input checked="" type="checkbox"/>
	ukuran_baterai	int	<input checked="" type="checkbox"/>
	nama_processor	varbinary(50)	<input checked="" type="checkbox"/>
	kamera	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Tidak lupa kita set kolom `id` sebagai `Primary key` dengan cara klik kanan kolom `id` pada form tersebut lalu pilih `Set Primary Key`, dan jika sudah di set maka akan ada `icon kunci` disebelah kolom `id` :

LAPTOP-SNTFOJ1C\...a - dbo.Table_1*			
	Column Name	Data Type	Allow Nulls
▶	id		<input type="checkbox"/>
	nama_handphone		
	merek		
	ukuran_layar		
	ukuran_baterai		
	nama_processor		
	kamera		

- Set Primary Key
- Insert Column
- Delete Column
- Relationships...
- Indexes/Keys...
- Fulltext Index...
- XML Indexes...
- Check Constraints...
- Spatial Indexes...
- Generate Change Script...

Jika sudah kita tinggal menyimpan table tersebut dengan cara tekan `ctrl+s` pada keyboard, kemudian akan muncul form untuk menamai table, kita isi dengan nama `handphone`



Jika sudah kita klik OK lalu nama table akan muncul di list **folder Table**.

Selain itu kita juga bisa menggunakan perintah SQL juga dengan cara statement seperti ini :

```
create table nama_table(  
    kolom1 type_data,  
    kolom2 type_data,  
    kolom3 type_data,  
    ...  
);  
  
/* contohnya */  
create table handphone(  
    id int primary key,  
    nama_handphone varchar(50),  
    merek varchar(50),  
    ukuran_layar float,  
    ukuran_baterai int,  
    nama_processor varchar(50),  
    kamera varchar(50),  
);
```

---

## Perintah SQL mengelola data

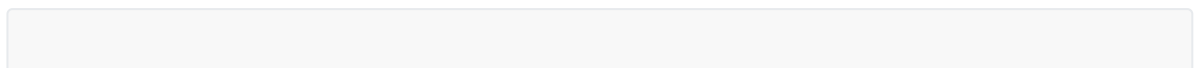
---

Kita sudah membuat **table handphone** maka yang harus kita lakukan adalah kita akan menggunakan perintah SQL untuk mengelola **table handphone**. Ada banyak perintah SQL yang sering digunakan untuk mengelola database, namun perlu diketahui bahwa perintah SQL ini cukup mudah dipahami karena perintah SQL ini menggunakan bahasa inggris sebagai basis perintahnya sehingga jika kita logika dan kita susun akan mudah dipahami.

Kita akan menggunakan beberapa perintah yang sering digunakan untuk mengelola database :

### Menambah data pada Table

Untuk menambah data pada Table kita bisa menggunakan perintah dengan statement dibawah ini :



```

insert into nama_table(kolom1, kolom2, kolom3, ...) values (value1, value2,
value3);

/* contohnya */
insert into handphone(id, nama_handphone, merek, ukuran_layar, ukuran_baterai,
nama_processor, kamera)
values(1, 'Samsung A70', 'Samsung', '4.7', '4500', 'snapdragon 645', '32');

insert into handphone(id, nama_handphone, merek, ukuran_layar, ukuran_baterai,
nama_processor, kamera)
values(2, 'Xiaomi Mi 10', 'Xiaomi', '4.7', '4500', 'snapdragon 645', '64');

insert into handphone(id, nama_handphone, merek, ukuran_layar, ukuran_baterai,
nama_processor, kamera)
values(3, 'Oppo Reno 4', 'Oppo', '5.7', '4500', 'snapdragon 645', '64');

insert into handphone(id, nama_handphone, merek, ukuran_layar, ukuran_baterai,
nama_processor, kamera)
values(4, 'iPhone 12', 'Apple', '5.7', '2500', 'A14', '64');

insert into handphone(id, nama_handphone, merek, ukuran_layar, ukuran_baterai,
nama_processor, kamera)
values(5, 'Samsung A51', 'Samsung', '4.7', '4500', 'snapdragon 645', '32');

```

Jika kita lihat perintah diatas kita harus memasukkan semua nama kolom baru kemudian kita akan memasukkan value/data setelah perintah `value` tentu saja urutan `value` harus disamakan dengan urutan kolom.

## Mengubah data pada Table

Untuk mengubah data pada table tertentu kita bisa mengubah kolom mana saja yang sesuai dengan kebutuhan data yang di update seperti pada statement dibawah ini :

```

update nama_table set kolom1=value1, kolom2=value2, kolom3=value3 where kondisi;

/* contohnya */
update handphone set nama_handphone='A51S', ukuran_layar='5.1', kamera='64'
where id = '5';

update handphone set nama_handphone='A51S', ukuran_layar='5.1', kamera='64'
where merek = 'Samsung';

```

## Menghapus data pada Table

Untuk hapus data kita hanya membutuhka perintah SQL yang pendek dengan statement seperti ini :



```
delete from nama_table where kondisi;

/* contohnya */
delete from handphone where id = '5';

delete from handphone where merek = 'Oppo';
```

## Menampilkan data pada Table

Menampilkan data sudah menjadi kewajiban kita ketika mengelola database, tentu saja ada beberapa perintah dan kondisi yang disediakan SQL untuk menampilkan data sesuai dengan keinginan kita seperti yang ada pada bawah ini :

### Menampilkan seluruh data

Untuk bisa menampilkan data dengan semua kolom kita bisa menggunakan statement seperti ini :

```
select * from nama_table;

/* contohnya */
select * from handphone;
```

Kalau kita lihat pada perintah tersebut terdapat tanda `*` yang dimana tanda `*` ini digunakan untuk mewakili semua kolom. Namun apabila tidak ingin menampilkan semua kita bisa menggunakan statement seperti ini :

```
select kolom1, kolom2, kolom3, ... from nama_table;

/* contohnya */
select merek, ukuran_layar, nama_processor from handphone;
```

### Menampilkan semua data dengan filter kolom

Kadang saat kita mau menampilkan data kita harus mengelompokkan data berdasarkan kondisi, misalkan kita mau cari data handphone yang memiliki merek Samsung. Pada kasus tersebut kita bisa menggunakan kondisi `where` dengan statement seperti ini :

```
select * from nama_table where nama_kolom='kondisi';

/* contohnya */
select * from handphone where merek='Samsung';
```

Lalu Bagaimana kalau kita tidak hanya menampilkan kolom dengan merek samsung saja? misalkan kita mau menampilkan data 2 merek Samsung dan Xiaomi. Jika kondisi seperti itu maka kita bisa menggunakan `or` dengan statement seperti ini :

```
select * from nama_table where nama_kolom='kondisi1' or nama_kolom='kondisi2';

/* contohnya */
select * from handphone where merek='Samsung' or merek='Xiaomi';
```

Ada juga kita harus menampilkan data Handphone kolom bermerek Samsung dan kolom ukuran Layar 5.0 jika menemukan kasus seperti itu maka kita bisa menggunakan `and` dengan statement berikut :

```
select * from nama_table where nama_kolom1='kondisi1' and
nama_kolom2='kondisi2';

/* contohnya */
select * from handphone where merek='Samsung' and ukuran_layar='4.7';
```

Terkadang selain filter kita membutuhkan pencarian juga bukan? misalkan kita mau mencari kata yang berhubungan dengan kata "samsung" kita bisa menggunakan `like`, namun menggunakan `like` ini kita membutuhkan `where` dan kita juga harus menambahkan karakter pengganti seperti :

- `%` Tanda persen mewakili nol, satu, atau beberapa karakter
- `_` Tanda underscore merepresentasikan satu karakter

```
select * from nama_table where nama_kolom like pola

/* contohnya */

/* mencari kata yang dengan awalan `S` */
select * from handphone where merek like 'S%';

/* mencari kata yang dengan akhiran `S` */
select * from handphone where merek like '%S';

/* mencari kata `S` yang ada di tengah-tengah */
select * from handphone where merek like '%S%';

/* mencari kata pada huruf kedua yang berhubungan dengan huruf 'a' */
select * from handphone where merek like '_a%';
```

## Menampilkan data berdasarkan urutan dari kolom

Di SQL kita juga bisa mengurutkan data berdasarkan kebutuhan tentu mengurutkan ini berdasarkan data dari kolom, misalkan saja kita mau mengurutkan data berdasarkan kolom nama processor maka kita bisa melakukan dengan statement seperti ini :

```
select * from nama_table order by nama_kolom;

/* contohnya */
select * from handphone order by nama_processor;
```

pada perintah diatas akan menghasilkan data urutan kecil ke besar, kita juga bisa mengurutkannya dengan menggunakan `desc` untuk mengurutkan dari besar ke kecil dan menggunakan `asc` untuk mengurutkan dari kecil ke besar :

```
/* mengurutkan dari nilai terbesar ke terkecil */
select * from nama_table order by nam_kolom desc;

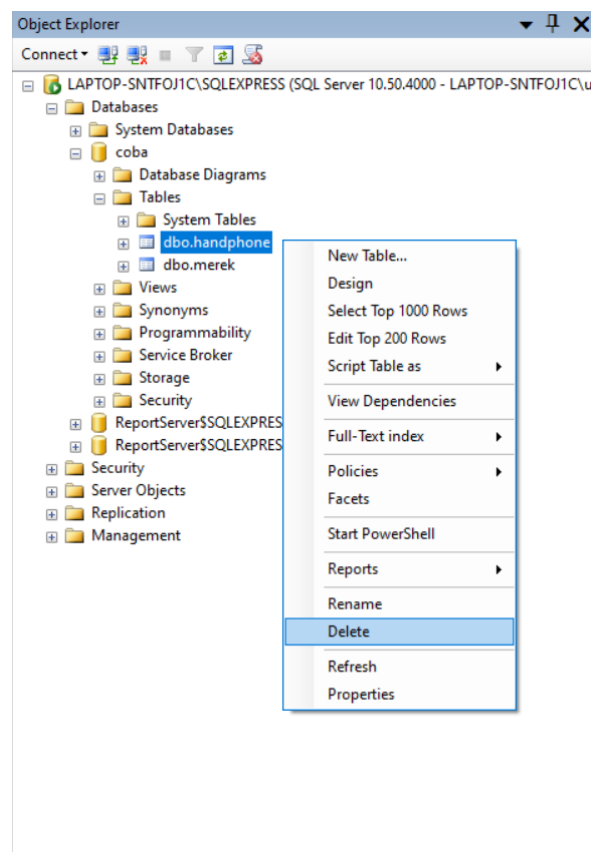
/* mengurutkan dari nilai terkecil ke terbesar */
select * from nama_table order by nam_kolom asc;
```

## Menampilkan data pada 2 Table yang berbeda

Ada beberapa kasus dimana ada dua table yang memiliki hubungan, contohnya pada table sebelumnya yaitu table `handphone` di dalam table tersebut memiliki kolom `merek`, padahal kita bisa memecah kolom `merek` menjadi table `merek`.

Kenapa kita harus memecah kolom `merek` menjadi table? hal ini dilakukan agar data merek memiliki data sendiri mengingat data merek ini juga nantinya akan memiliki data yang banyak maka dari itu kita perlu memisahkan dari kolom dan menjadi table.

Namun sebelum itu kita harus menghapus table `handphone` terlebih dahulu mengingat di table tersebut masih memiliki kolom merek. Cara menghapus tablenya kita ke `fo1der Table` pada toolbar sebelah kiri lalu klik kanan `dbo.handphone` pilih `delete`



Setelah dihapus kita akan membuat 2 table dengan perintah SQL seperti dibawah ini :

```
/* table handpone */
create table handphone(
    id int primary key,
    nama_handphone varchar(50),
    id_merek int,
    ukuran_layar float,
    ukuran_baterai int,
    nama_processor varchar(50),
```

```

        kamera varchar(50),
    );

    /* table merek */
    create table merek(
        id int primary key,
        nama_merek varchar(50),
    );

```

lalu kita Insert datanya

```

    /* insert data handphone */
    insert into handphone(id, nama_handphone, id_merek, ukuran_layar,
    ukuran_baterai, nama_processor, kamera)
    values(1, 'Samsung A70',1,'4.7','4500','snapdragon 645','32');

    insert into handphone(id, nama_handphone, id_merek, ukuran_layar,
    ukuran_baterai, nama_processor, kamera)
    values(2, 'Xiaomi Mi 10',2,'4.7','4500','snapdragon 645','64');

    insert into handphone(id, nama_handphone, id_merek, ukuran_layar,
    ukuran_baterai, nama_processor, kamera)
    values(3, 'Oppo Reno 4',3,'5.7','4500','snapdragon 645','64');

    insert into handphone(id, nama_handphone, id_merek, ukuran_layar,
    ukuran_baterai, nama_processor, kamera)
    values(4, 'iPhone 12',4,'5.7','2500','A14','64');

    insert into handphone(id, nama_handphone, id_merek, ukuran_layar,
    ukuran_baterai, nama_processor, kamera)
    values(5, 'Samsung A51',1,'4.7','4500','snapdragon 645','32');

    /* insert data merek */
    insert into merek (id, nama_merek)values(1, 'Samsung');
    insert into merek (id, nama_merek)values(2, 'Xiaomi');
    insert into merek (id, nama_merek)values(3, 'Oppo');
    insert into merek (id, nama_merek)values(4, 'Apple');

```

Jika sudah coba tampilkan semua datanya :

```

select * from handphone
select * from merek

```

Nah pada saat kita menampilkan data handphone kita melihat kalau kolom `id_merek` menampilkan id dari table `merek`, sedangkan di table `merek` memiliki kolom `nama_merek` yang harusnya digunakan untuk menampilkan nama merek.

Untuk itu kita akan menggabungkan data dari kedua table tersebut dengan menggunakan `join` penggunaan `join` ini cukup sederhana dimana kita akan menyamakan data dari kolom yang berbeda antar table, pada kasus ini kolom yang datanya sama adalah : kolom `id_merek` dari table `handphone` dengan kolom `id` dari table `merek` lalu kita gunakan statement join seperti ini

```
select * from nama_table1 join nama_table2 on nama_table1.kolom =  
nama_table2.kolom
```

```
/* contohnya */
```

```
select * from handphone join merek on handphone.id_merek = merek.id
```

```
/* jika ingin hasilnya lebih rapi */
```

```
select id.handphone, nama_handphone.handphone, nama_merek.merek from handphone  
join merek on handphone.id_merek = merek.id
```