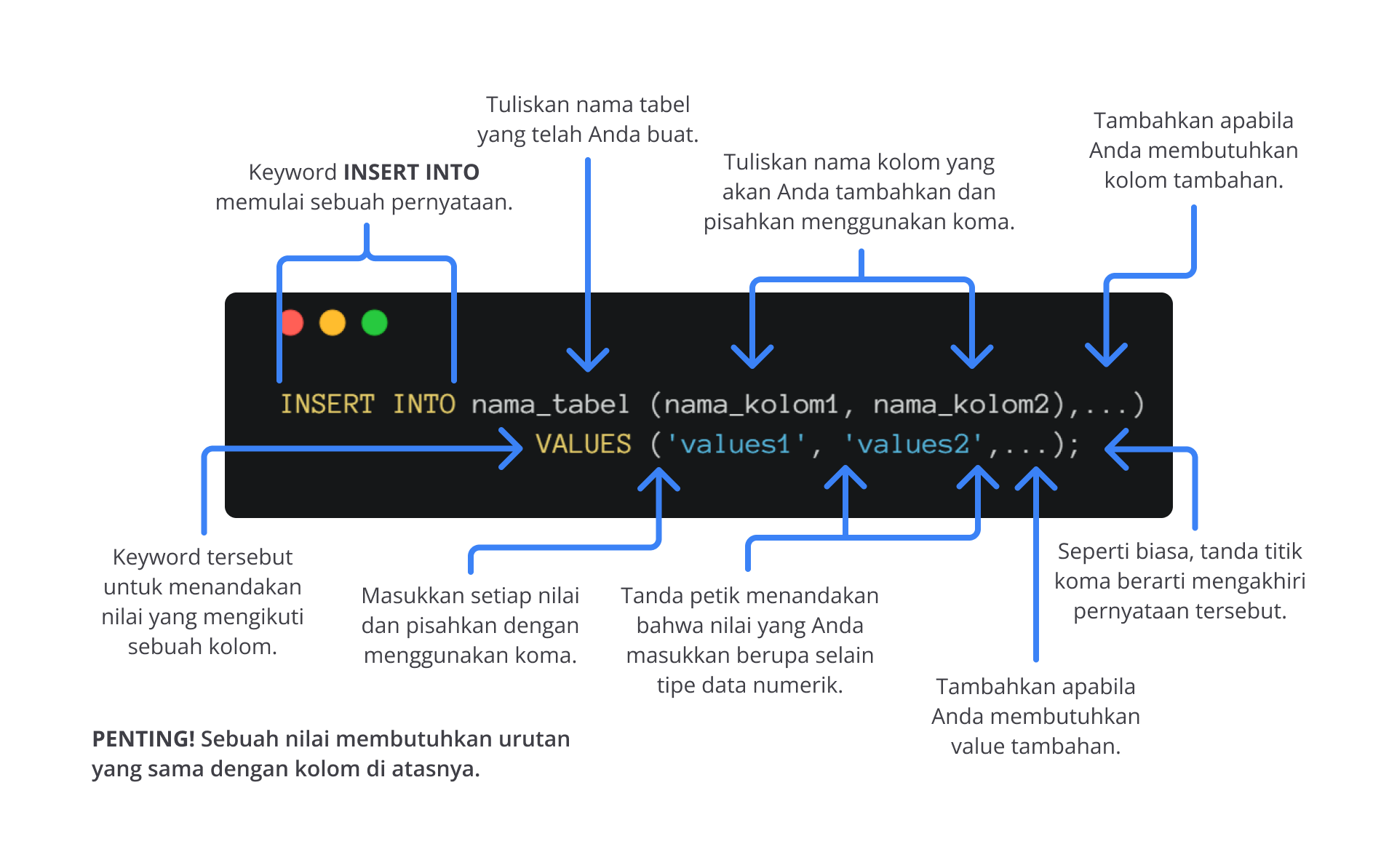
**INSERT**

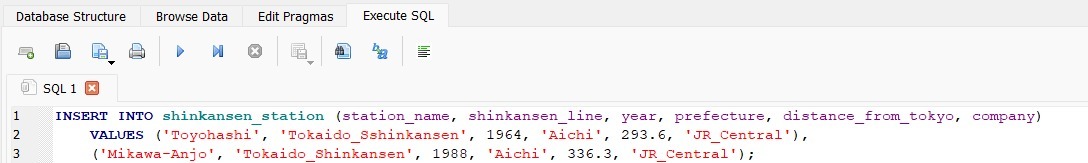
**Pengantar INSERT**



Pada CREATE TABLE yang telah dibuat sebelumnya, Anda memiliki sepuluh baris tabel dengan nilai berbeda-beda. Langkah selanjutnya adalah coba Anda menambahkan dua baris baru dari dataset (no. 11 dan no. 12) menggunakan INSERT dan bandingkan hasilnya dengan gambar di bawah ini. *Good luck!*

**Latihan Implementasi INSERT**

Langkah pertama, tambahkan dua baris baru dari dataset menggunakan INSERT INTO pada tabel yang telah Anda buat dengan VALUES yang terdapat pada dataset sebelumnya.



**SELECT & FROM**

Layaknya persamaan matematika yang telah dijelaskan dalam modul sebelumnya, SELECTyang dilambangkan oleh simbol *sigma*() adalah operasi untuk menyeleksi tupel-tupel yang memenuhi suatu predikat. Bagaimanakah cara kerjanya? Yuk, kita simak penjelasannya.

**Latihan SELECT & FROM**

Setelah memasukkan data menggunakan INSERT INTO, selanjutnya adalah Anda mempelajari SELECT & FROM untuk dapat melihat seluruh data yang tersimpan dalam basis data tanpa perlu membuka *Browse Data.*Perhatikan langkah-langkahnya di bawah ini.

1. Sebelum menggunakan pernyataan berikut, Anda dapat menghapus query INSERT INTO sebelumnya untuk memudahkan Anda menggunakan konsol atau menambah tab baru dengan klik ***open tab***atau **CTRL + T**.
2. Agar Anda dapat melihat seluruh data, cukup masukkan tanda \* (bintang) bersamaan dengan SELECT dan FROM yang diakhiri dengan tanda titik koma (;),

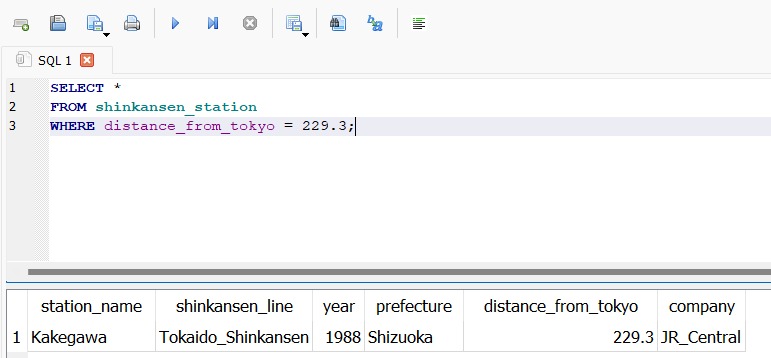
**WHERE**

WHEREyaitu spesifikasi kata yang Anda cari di konsol tersebut.Tentunya dengan menggunakan pernyataan WHERE,pencarian data yang lebih spesifik akan lebih mudah. Lalu, bagaimanakah cara kerja dari WHERE*?*

Pada dasarnya pernyataan WHERE dapat digunakan pada kondisi apa pun serta dapat dipadukan dengan pernyataan lainnya. Pada modul sebelumnya, Anda telah dikenalkan dengan penggunaan WHERE untuk setiap pernyataan. Anda akan mempelajari lebih lanjut terkait WHERE dengan beberapa kondisi, di antaranya *numeric, non-numeric,*AND, OR, NOT, dan LIKE. Simak penjelasannya di bawah ini dengan saksama, ya!

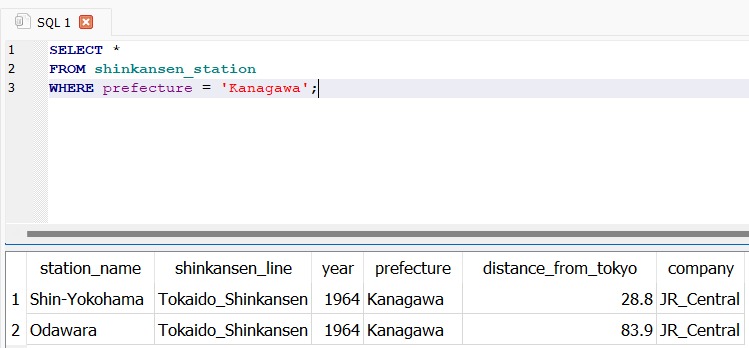
**Numeric**

Pada saat membuat tabel (CREATE TABLE) di awal, Anda pasti sudah menentukan kolom dengan tipe datanya masing-masing. Apabila Anda mengikuti langkah-langkah yang persis seperti di atas, pada modul ini yang termasuk dalam tipe data numerik adalah kolom distance\_from\_tokyo*.*Maka dari itu, mari simak implementasinya di bawah ini, ya.



**Non-numeric**

Setelah sebelumnya berlatih mengenai penggunaan WHERE dengan tipe data numeric, saat ini Anda akan berlatih berkebalikannya, yaitu non-numeric alias semua yang terdapat pada basis data dan tidak berupa bilangan, contohnya kolom distance\_from\_tokyoseperti di atas.



**AND, OR, NOT**

Nah, setelah Anda mempelajari *numeric*dan *non-numeric*, Anda akan kami kenalkan dengan operator AND, OR, dan NOT. Bagaimana sajakah operator yang bekerja untuk ketiga pertanyaan tersebut? Yuk, simak penjelasannya.

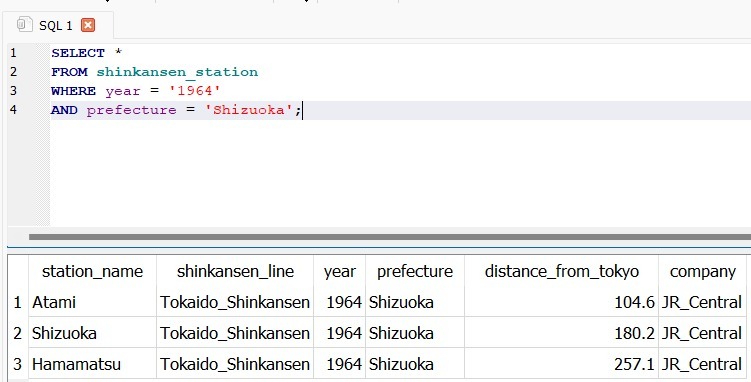
Sebelum Anda melaksanakan ketiga operator tersebut, mari kita pisahkan satu per satu antara AND, OR, dan NOT.

Anda bisa menggunakan AND atau OR untuk memadukan dua buah pernyataan yang mirip seperti logika matematis yang pernah kita pelajari saat SMA dulu. Anda masih ingat bukan? Nah, sedangkan NOT merupakan operator yang digunakan untuk menghasilkan negasi dari suatu pernyataan.

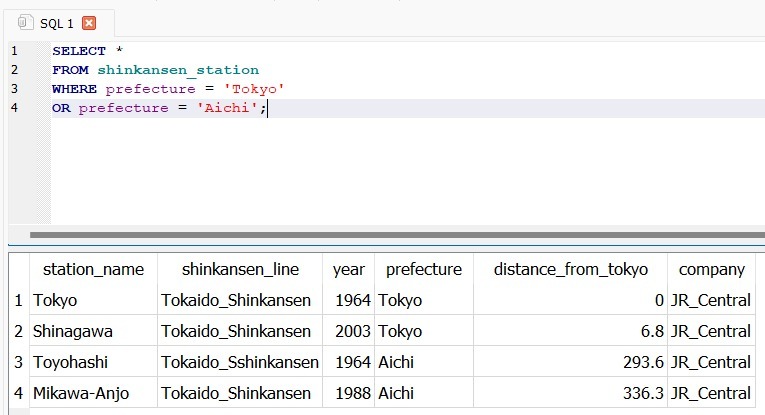
Ketika membuat sebuah pernyataan WHERE, kita membutuhkan beberapa operator seperti pada tabel berikut.

Simak contoh di bawah ini yang dapat Anda gunakan untuk melatih kemampuan dalam menjalankan WHERE dengan operator AND, OR, dan NOT.

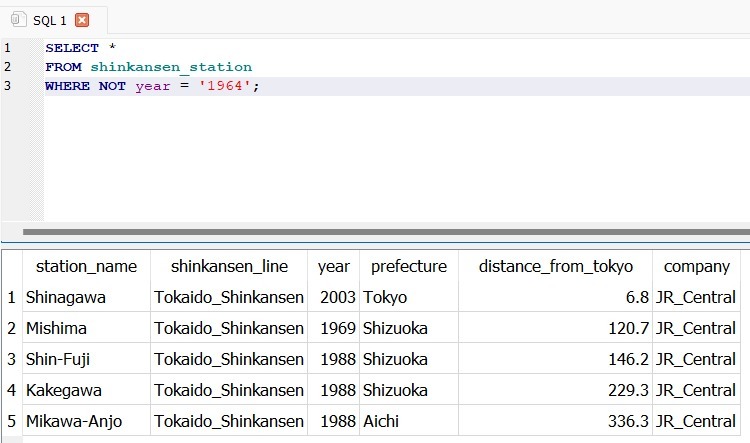
1. **WHERE - AND**  
   Gambar berikut menunjukkan cara untuk meng-query data yang memiliki nilai **year=1964** dan **prefecture=“Shizuoka”**. Untuk menggabungkan kedua pernyataan tersebut, kita perlu menggunakan operator AND. Ia hanya akan menampilkan data yang memenuhi kedua kondisi tersebut.  
     
   **Catatan:** Untuk operator AND, kolom yang digunakan tidak boleh sama, contohnya WHERE year = … AND year = …  
     
   Jika dilakukan, konsol tidak akan menampilkan hasil apa pun.



1. **WHERE - OR**  
   Gambar di bawah ini menampilkan data yang memiliki nilai **prefecture=“Tokyo”** atau **prefecture=“Aichi”**. Pada kasus ini, kita bisa menggunakan operator OR. Ia akan menampilkan data yang memenuhi salah satu kondisi tersebut. Berbeda dengan sebelumnya, yakni operator AND, operator OR dapat menggunakan nama kolom yang sama pada konsol. Contohnya tertera pada gambar berikut.



1. **WHERE-NOT**  
   Terakhir, katakanlah Anda ingin menampilkan seluruh data selain data yang memiliki nilai **year=1964**. Untuk melakukan hal ini, Anda bisa menggunakan operator NOT seperti pada gambar di bawah ini.



**LIKE**

Pada modul sebelumnya, Anda telah mengetahui dan mempraktikkan operator AND, OR, dan NOT. Sama halnya dengan LIKE, ia memiliki operator dalam mencari pola dan kolom yang diperlukan.

Operator LIKE digunakan dalam pernyataan WHERE untuk mencari pola yang ditentukan pada kolom.

Ada dua pola yang sering digunakan bersama operator LIKE.

* Tanda persen (%) mewakili nol, satu, atau beberapa karakter.
* Tanda garis bawah ( \_ ) mewakili satu karakter tunggal.

Sebelum Anda latihan langsung di SQL, berikut adalah pola yang dapat digunakan bersamaan dengan LIKE.

* WHERE (nama kolom) LIKE ‘a%’

Mencari nilai apa pun pada field kolom yang dimulai huruf “a” atau Anda bisa menggunakan huruf lainnya.

* WHERE (nama kolom) LIKE ‘%a’

Mencari nilai apa pun pada field kolom nama yang diakhiri huruf “a” atau Anda bisa menggunakan huruf lainnya.

* WHERE (nama kolom) LIKE ‘%or%’

Mencari nilai apa pun pada field kolom yang di dalamnya terdapat huruf “or”. Anda juga bisa menggunakan huruf lainnya.

* WHERE (nama kolom) LIKE ‘\_r%’

Mencari nilai apa pun pada field kolom yang karakter keduanya huruf “r” atau Anda bisa menggunakan karakter lainnya.

* WHERE (nama kolom) LIKE ‘a\_%’

Mencari nilai apa pun pada field kolom yang dimulai dengan huruf “a” dan panjangnya minimal dua karakter atau Anda bisa menggunakan karakter lainnya.

* WHERE (nama kolom) LIKE ‘a\_\_%’

Mencari nilai apa pun pada field kolom yang dimulai dengan huruf “a” dan panjangnya minimal tiga karakter atau Anda bisa menggunakan karakter lainnya.

* WHERE (nama kolom) LIKE ‘a%o’

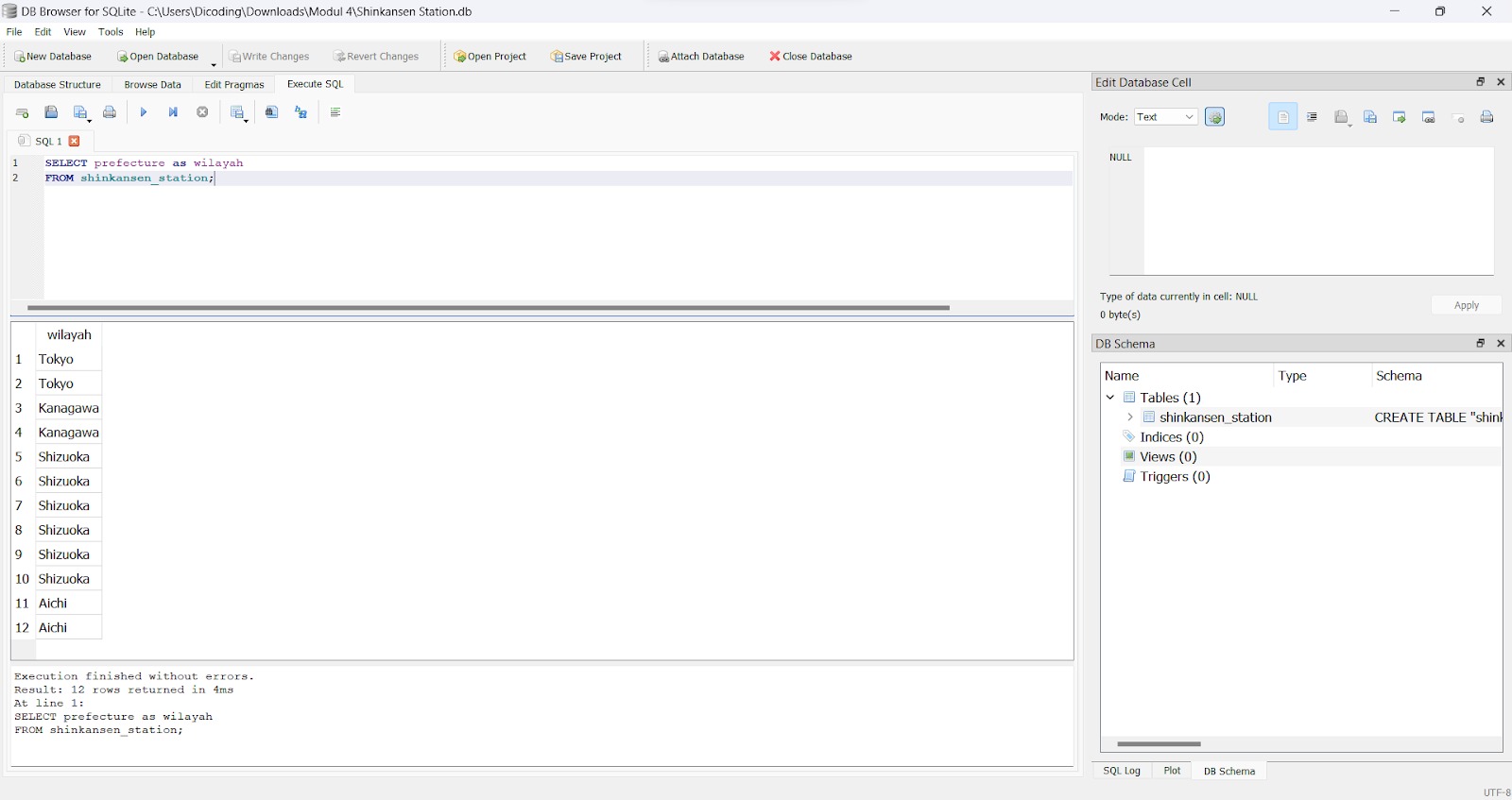
Mencari nilai apa pun pada field kolom yang dimulai huruf “a” dan diakhiri huruf “o” atau Anda bisa menggunakan karakter lainnya.

**ALIAS**

ALIAS pada SQL digunakan untuk memberi nama lain yang bersifat sementara pada tabel atau kolom dalam tabel. Maksud dari sementara adalah pernyataan ALIAS hanya berlaku pada proses query tersebut. ALIAS sering digunakan untuk membuat nama kolom lebih mudah dibaca.

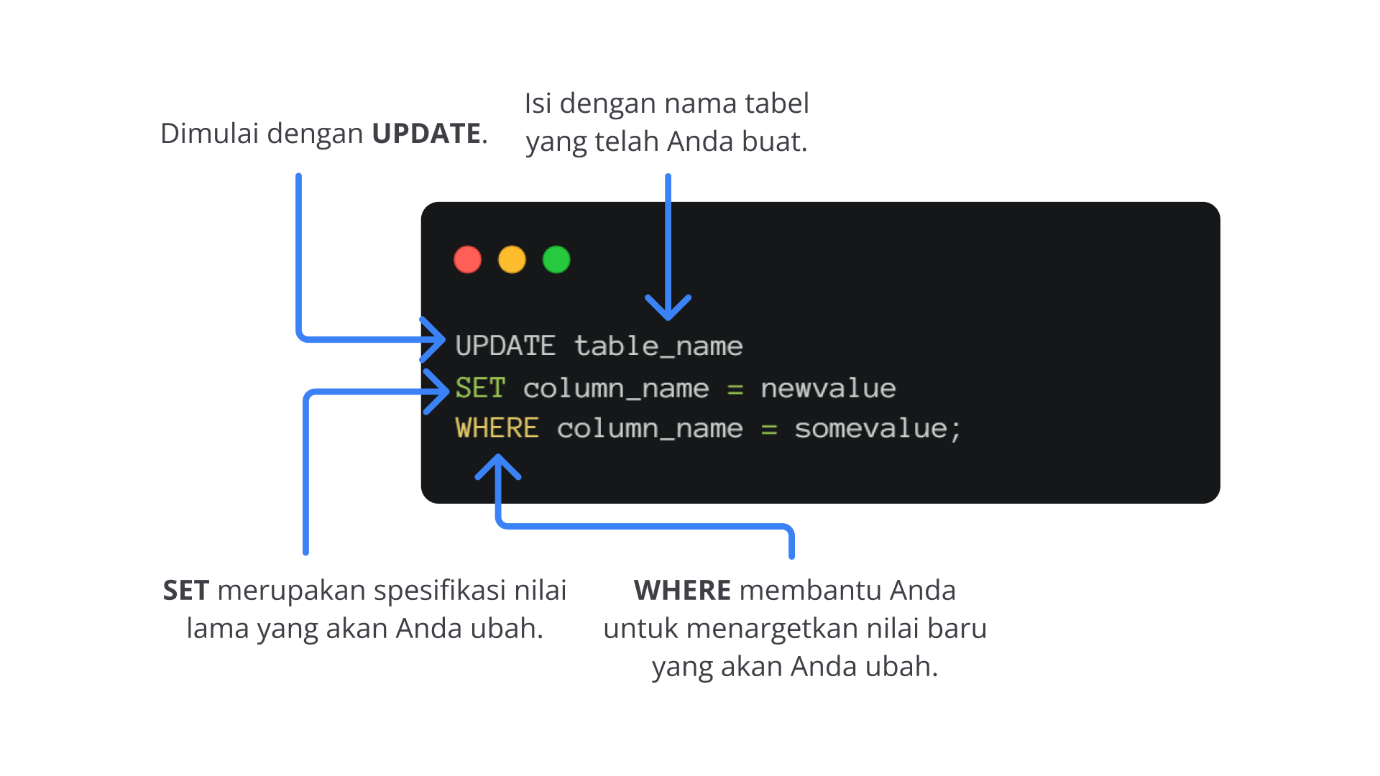


Pada tabel shinkansen\_station terdapat kolom bernama prefecture yang berarti wilayah. Sebab dirasa pelafalan *prefecture*cukup sulit, kita akan ubah namanya menggunakan ALIAS menjadi wilayah agar lebih mudah diingat dalam latihan.



**UPDATE**

Salah satu fungsi dan tujuan pernyataan UPDATE adalah untuk mengubah nilai lama menjadi sebuah nilai baru pada sebuah basis data SQL. UPDATE disandingkan dengan pernyataan WHERE untuk mengetahui nilai yang akan diubah. Bagaimanakah cara kerjanya? Yuk, simak penjelasannya.



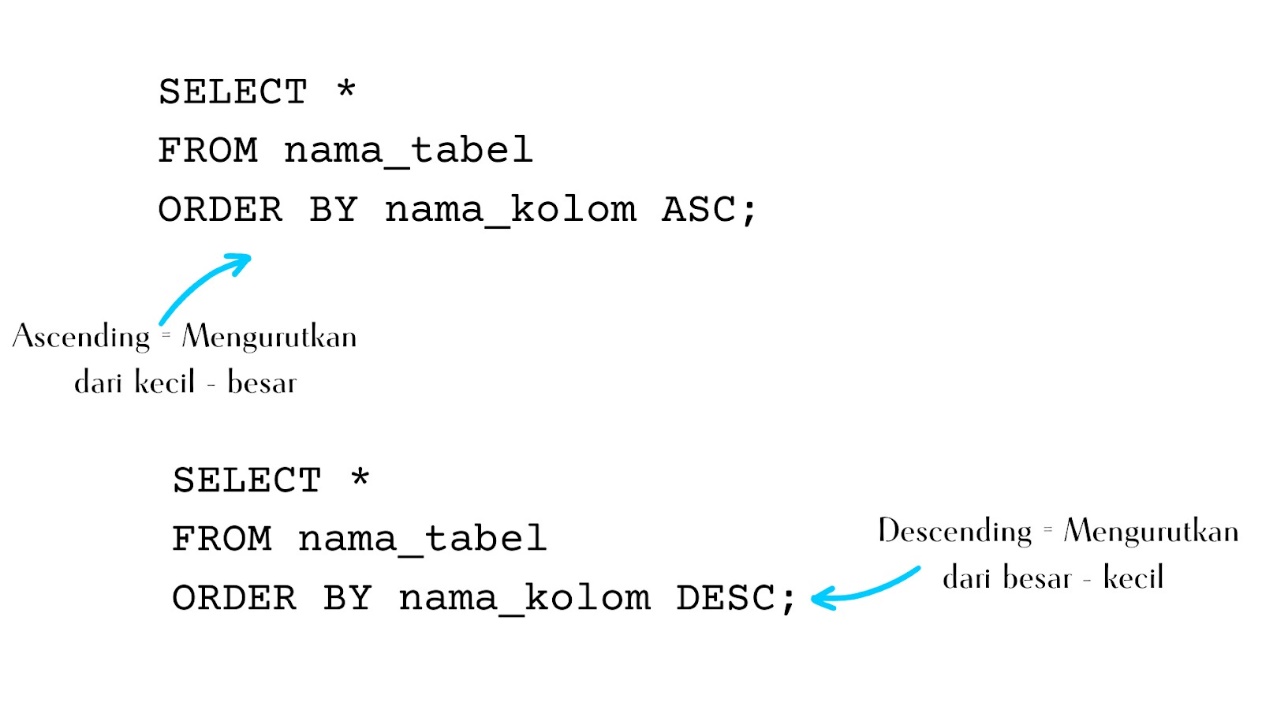
**ORDER BY**

Setiap *user*tentunya memiliki tujuan yang berbeda-beda dalam mengeksekusi data menggunakan SQL. Salah satunya adalah seorang *user*perlu mengurutkan sebuah data baik dari urutan kecil ke besar ataupun sebaliknya. ORDER BY akan membantu *user*dalam proses pengurutan data. Bagaimanakah cara kerjanya? Yuk, simak penjelasannya.

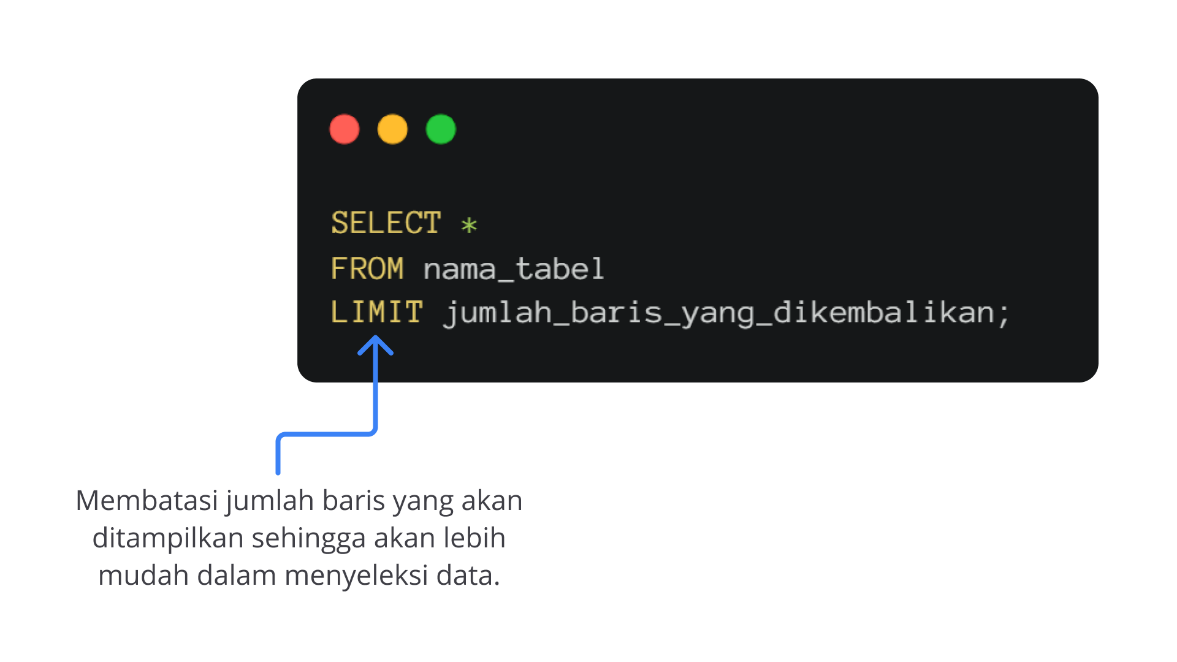
ORDER BY digunakan sebagai operator untuk menyusun data berurutan dari kecil ke besar atau besar ke kecil, atau biasanya disebut *ascending*(ASC) dan *descending*(DESC). Contohnya:

A - Z = *Ascending*

Z - A = *Descending*



**LIMIT**

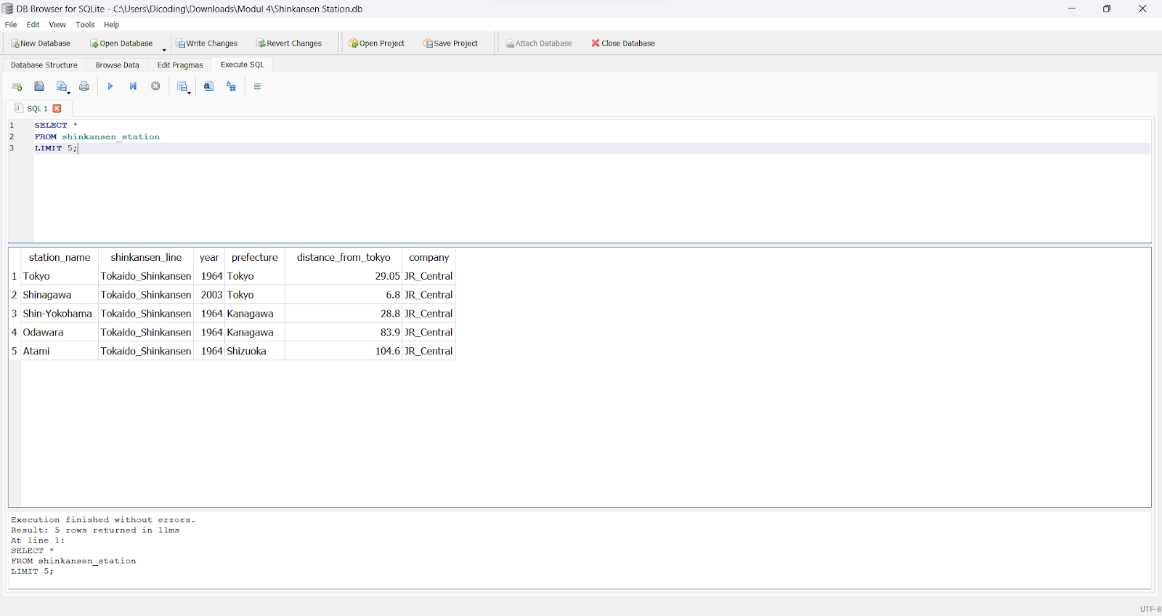


Sempat disebutkan dalam Modul 1 Pengenalan Data dan Basis Data, bahwa pengertian dari data salah satunya adalah kumpulan dari sebuah informasi. Kumpulan tersebut baik berskala kecil maupun berskala besar akan dianggap sebagai sebuah data.

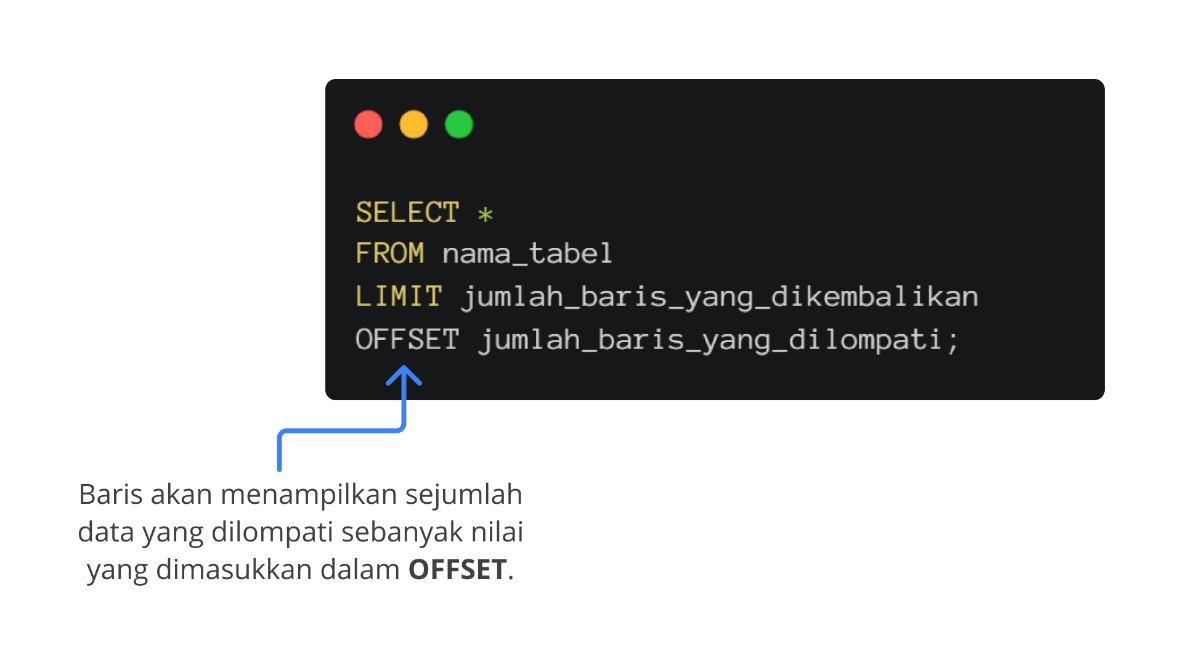
Apabila sebuah data berskala kecil, kemungkinan untuk mencari sebuah data yang diperlukan akan mudah, tetapi bagaimana jika ingin mencari data berskala besar? Maka dari itu, Anda memerlukan LIMIT untuk membatasi data-data yang tidak Anda perlukan. Bagaimanakah cara kerjanya? Yuk, simak penjelasannya.

Salah satu basic queryyang biasanya sering digunakan, yaitu LIMIT. Pernyataan tersebut berguna sebagai pembatas data yang ingin ditampilkan pada konsol. Apabila terjadi suatu kondisi Anda hanya ingin menampilkan beberapa baris dari data-data yang banyak, penggunaan LIMIT sangatlah tepat.

Sebagaimana yang Anda lihat, pada data tersebut terdapat 12 baris data yang terekam pada basis data. Namun, apabila hanya ingin menampilkan 5 baris data, Anda dapat menggunakan LIMIT untuk membatasi nilai-nilai tersebut. Simak implementasinya di bawah ini.



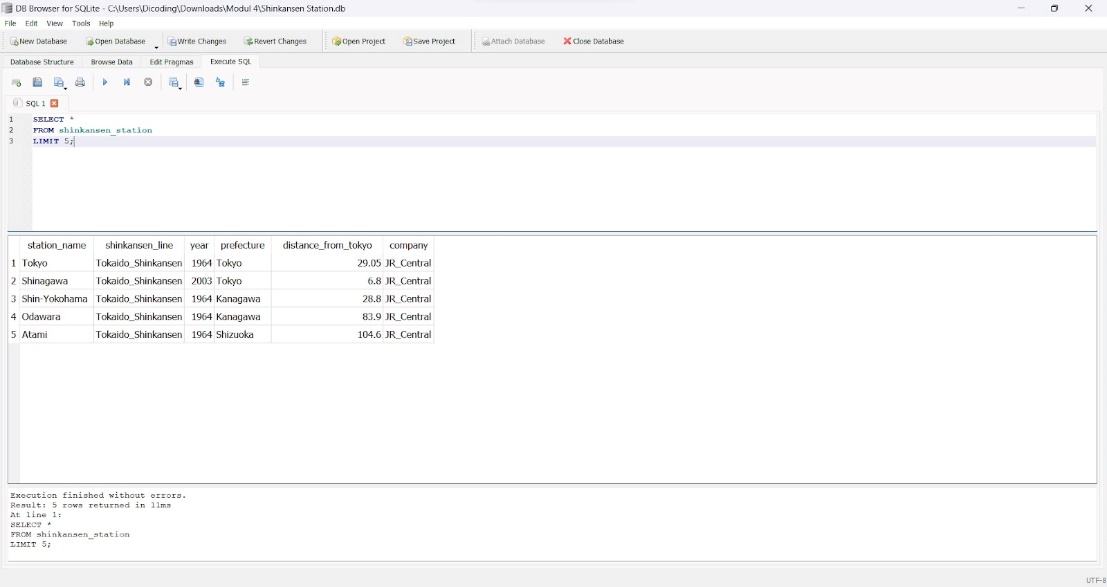
**OFFSET**



OFFSET membantu Anda dalam menemukan sebuah data yang bisa saja berada di awal, tengah, atau bahkan akhir dalam sebuah basis data. Bagaimanakah cara kerjanya? Yuk, simak penjelasannya.

Pada umumnya, penggunaan OFFSET disandingkan dengan penggunaan LIMIT dan dapat saling melengkapi. Cara kerja OFFSET adalah untuk melompati sejumlah baris dimulai dari yang pertama sampai yang ditentukan.

Masih sama dengan sebelumnya, terdapat sebuah tabel yang terdiri dari empat baris dan tiga kolom dengan nilai yang berbeda-beda. Apabila Anda menampilkan sebuah data tanpa menggunakan fungsi OFFSET, maka data tersebut akan tampil seluruhnya seperti tabel berikut.

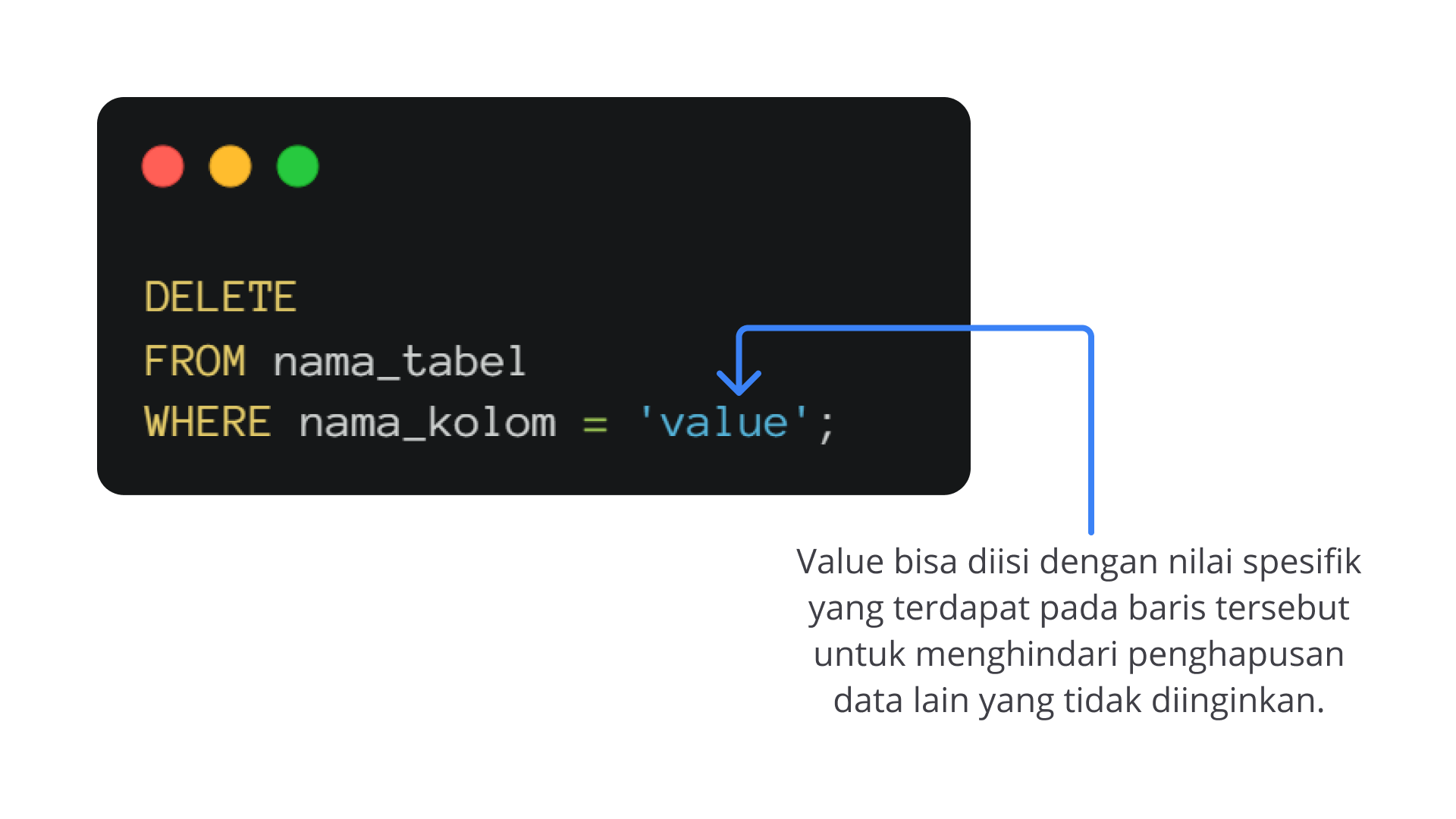


**DELETE**

Seperti judulnya, DELETE berarti hapus. Pada submodul terakhir ini, Anda akan mempelajari bagaimana cara menghapus tabel dari basis data yang sudah tidak diperlukan lagi. Bagaimanakah cara kerjanya? Yuk, simak penjelasannya.

*Last but not least!*Perjalanan Anda sebentar lagi sampai pada tujuan akhir. Di akhir materi ini, seperti pada modul sebelumnya, Anda kembali mempelajari materi terkait DELETE.

*Hayoo*! Anda masih ingat kan, perbedaan DROP TABLE dengan DELETE? Apabila ingin membaca kembali, Anda bisa membuka Modul 3 Pengenalan Structured Query Language (SQL), ya!



**Latihan Basic Query**

Pada akhir modul ini, Anda diharapkan untuk dapat latihan secara mandiri. Tujuannya sebagai tolok ukur pemahaman Anda terkait materi yang telah disampaikan sebelumnya, mulai dari membuat tabel hingga menghapus kembali tabel yang telah dibuat.

Maka dari itu, pada akhir modul ini, Anda dapat menggunakan Latihan Basic Query sebagai evaluasi atas seluruh materi yang telah dipelajari dalam modul ini, mulai dari membuat tabel hingga menghapus kembali seluruh tabel.