



CSGE602070 Basis Data
Semester Genap 2023/2024
Tutorial 1
Basic SQL

Deadline: 16 Maret 2024 23:55 WIB (Waktu SCoLE)

I. Mengakses Server

Dalam kuliah Basis Data, *Database Management System* (DBMS) yang digunakan adalah PostgreSQL yang berada pada server Kawung milik Fasilkom UI. Untuk mengakses server tersebut, lakukan langkah-langkah berikut.

Note :

Jika Anda merupakan pengguna sistem operasi Linux/MacOS, cukup lakukan langkah 1.

1. Buka email ui Anda dan cek email masuk berjudul “[FASILKOM12] Private Key - kawung.cs.ui.ac.id”. Pada email tersebut dicantumkan *private key* beserta *username* dan *passphrase*-nya. *Download file private key* yang ada pada email tersebut. Untuk pengguna sistem operasi Linux/MacOS dapat mengikuti instruksi yang tertulis dalam isi email. Untuk pengguna windows, dapat mengikuti langkah selanjutnya.
2. Jika belum memiliki aplikasi PuTTY, download aplikasi dari [link ini](#). Pilih file msi sesuai dengan bit sistem operasi Anda. Setelah aplikasi selesai di-*download*, *install* PuTTY ke komputer Anda.

Package files

You probably want one of these. They include versions of all the PuTTY utilities (except the new and slightly experimental Windows pterm).

(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

We also publish the latest PuTTY installers for all Windows architectures as a free-of-charge download at the [Microsoft Store](#); they usually take a few days to appear there after we release them.

MSI (“Windows Installer”)

64-bit x86:	putty-64bit-0.80-installer.msi	(signature)
64-bit Arm:	putty-arm64-0.80-installer.msi	(signature)
32-bit x86:	putty-0.80-installer.msi	(signature)

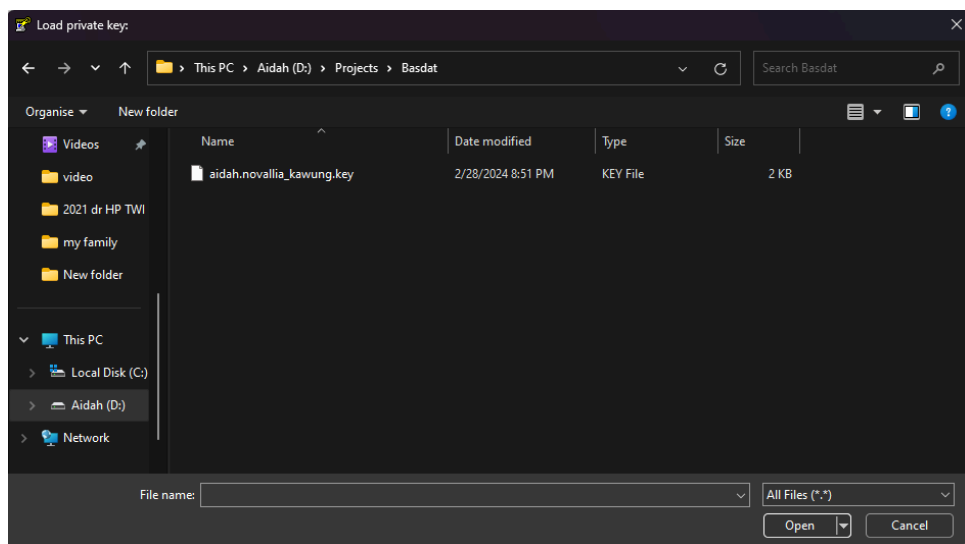
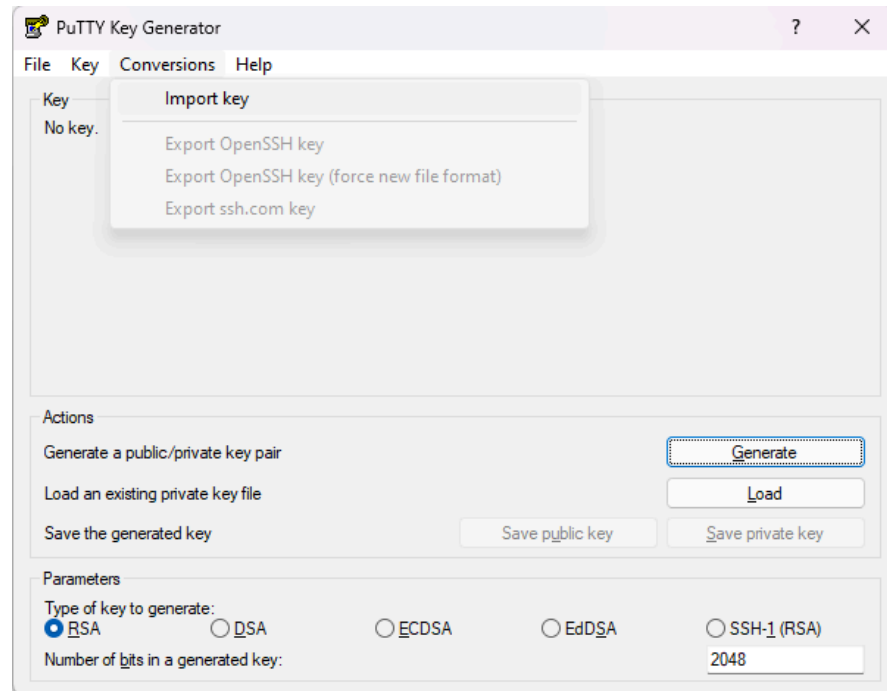
Unix source archive

.tar.gz:	putty-0.80.tar.gz	(signature)
----------	-----------------------------------	-------------------------------

3. Buka aplikasi PuTTYgen melalui komputer Anda. Pilih menu *Conversions* kemudian pilih *Import key*. Akan muncul file explorer dan arahkan explorer ke *private key* yang sebelumnya sudah di-*download*. Kemudian, klik *Open*.

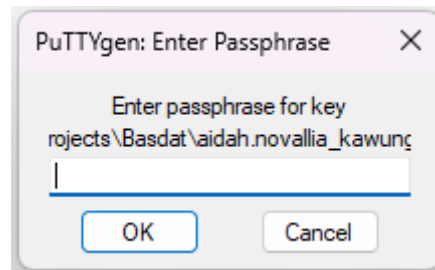


CSGE602070 Basis Data
Semester Genap 2023/2024
Tutorial 1
Basic SQL

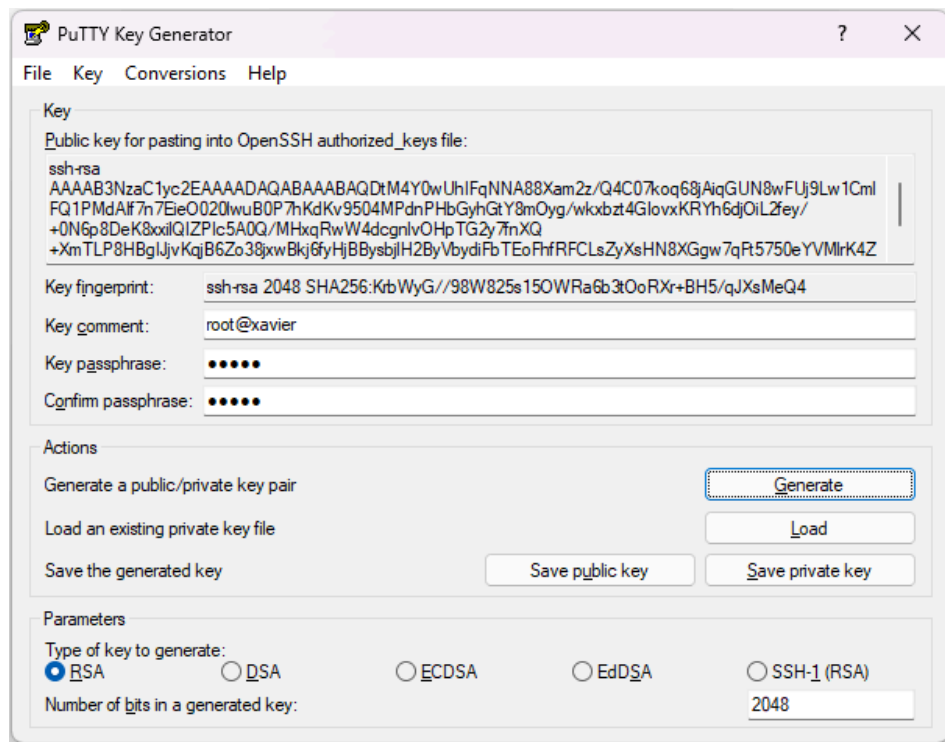


4. Masukkan *passphrase* yang diberikan pada email ke *field* berikut.

CSGE602070 Basis Data Semester Genap 2023/2024 Tutorial 1 Basic SQL

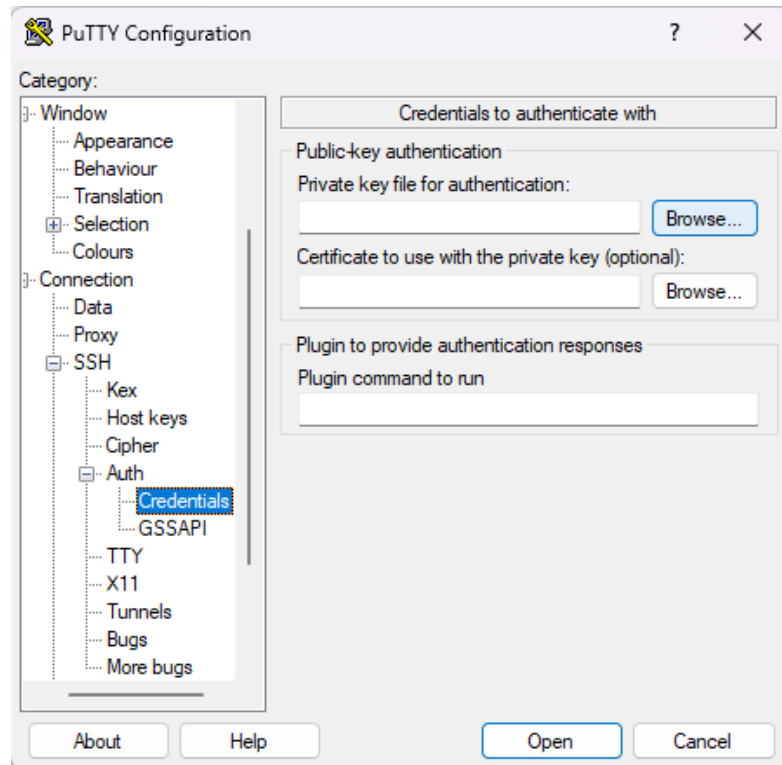


5. Tampilan PuTTYgen akan berubah menjadi seperti ini. Klik tombol *Save private key* dan pilih lokasi dimana file tersebut akan disimpan.



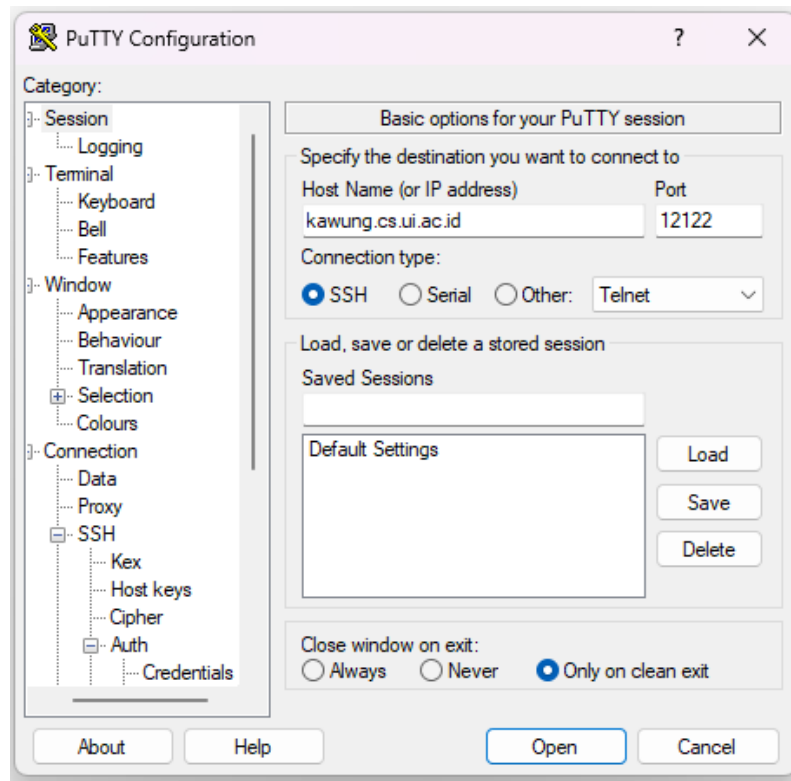
6. Selanjutnya PuTTYgen sudah boleh ditutup dan buka aplikasi PuTTY. Pilih category Connection → SSH → Auth → Credentials. Klik tombol *browse* dan buka *private key* yang sudah dibuat pada langkah nomor 5.

CSGE602070 Basis Data
Semester Genap 2023/2024
Tutorial 1
Basic SQL

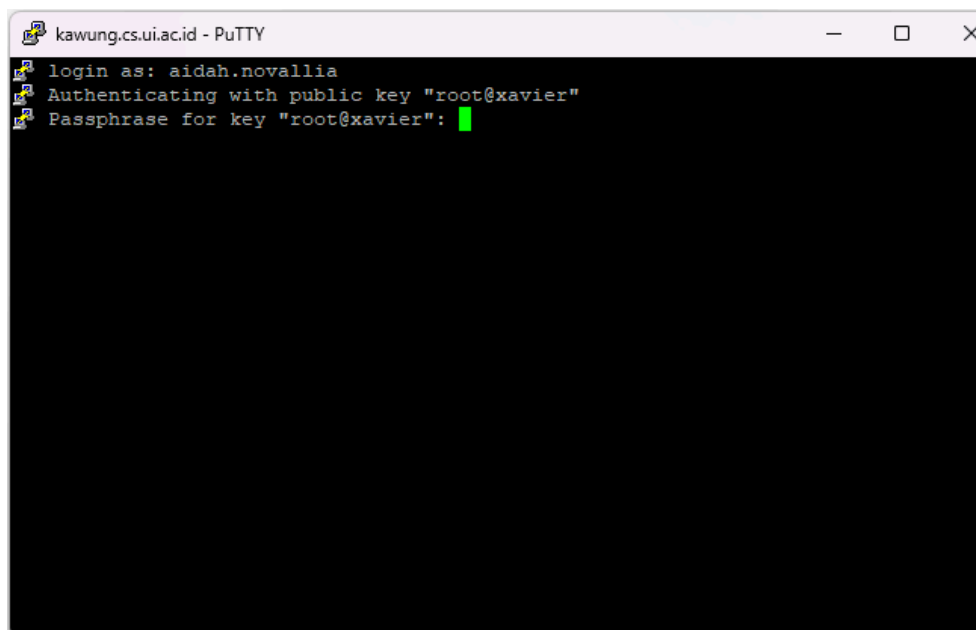


7. Kembali ke category *Session* dan isi *host name* dengan **kawung.cs.ui.ac.id** dan **port 12122**. Lalu, klik **Open**.

CSGE602070 Basis Data
Semester Genap 2023/2024
Tutorial 1
Basic SQL

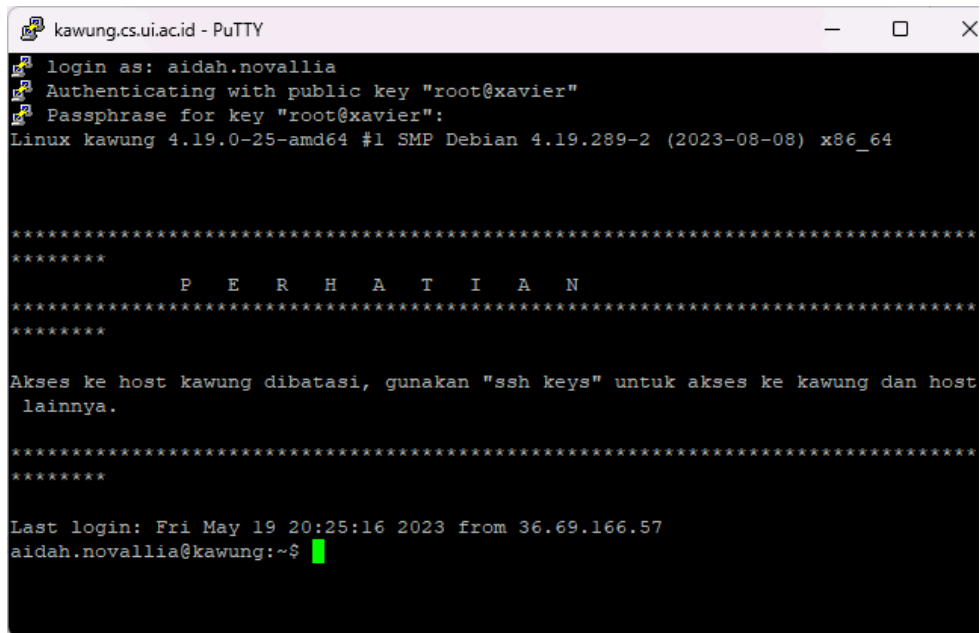


8. PuTTY akan meminta *username* dan *passphrase*. Isi sesuai dengan yang diberikan pada email.



CSGE602070 Basis Data
Semester Genap 2023/2024
Tutorial 1
Basic SQL

9. Setelah memasukkan *username* dan *passphrase*, tekan **enter**. Kemudian akan muncul tampilan seperti di bawah ini



10. Ketikkan perintah:

```
psql -U [USERNAME_SIAKNG] -h db.cs.ui.ac.id
```

Ganti [USERNAME_SIAKNG] dengan *username* Anda sendiri seperti contoh di bawah ini:

```
aidah.novallia@kawung:~$ psql -U aidah.novallia -h db.cs.ui.ac.id
```

Tekan **enter**. Kemudian masukkan *password* SSO Anda

```
Password for user aidah.novallia:
```

Setelah itu akan terdapat tampilan seperti dibawah ini:

```
Password for user aidah.novallia:
psql (11.22 (Debian 11.22-0+deb10u1), server 9.4.26)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
Type "help" for help.
aidah.novallia=>
```

11. *Database* sudah terakses, tabel bisa mulai dibuat.

CATATAN PENTING:



CSGE602070 Basis Data
Semester Genap 2023/2024
Tutorial 1
Basic SQL

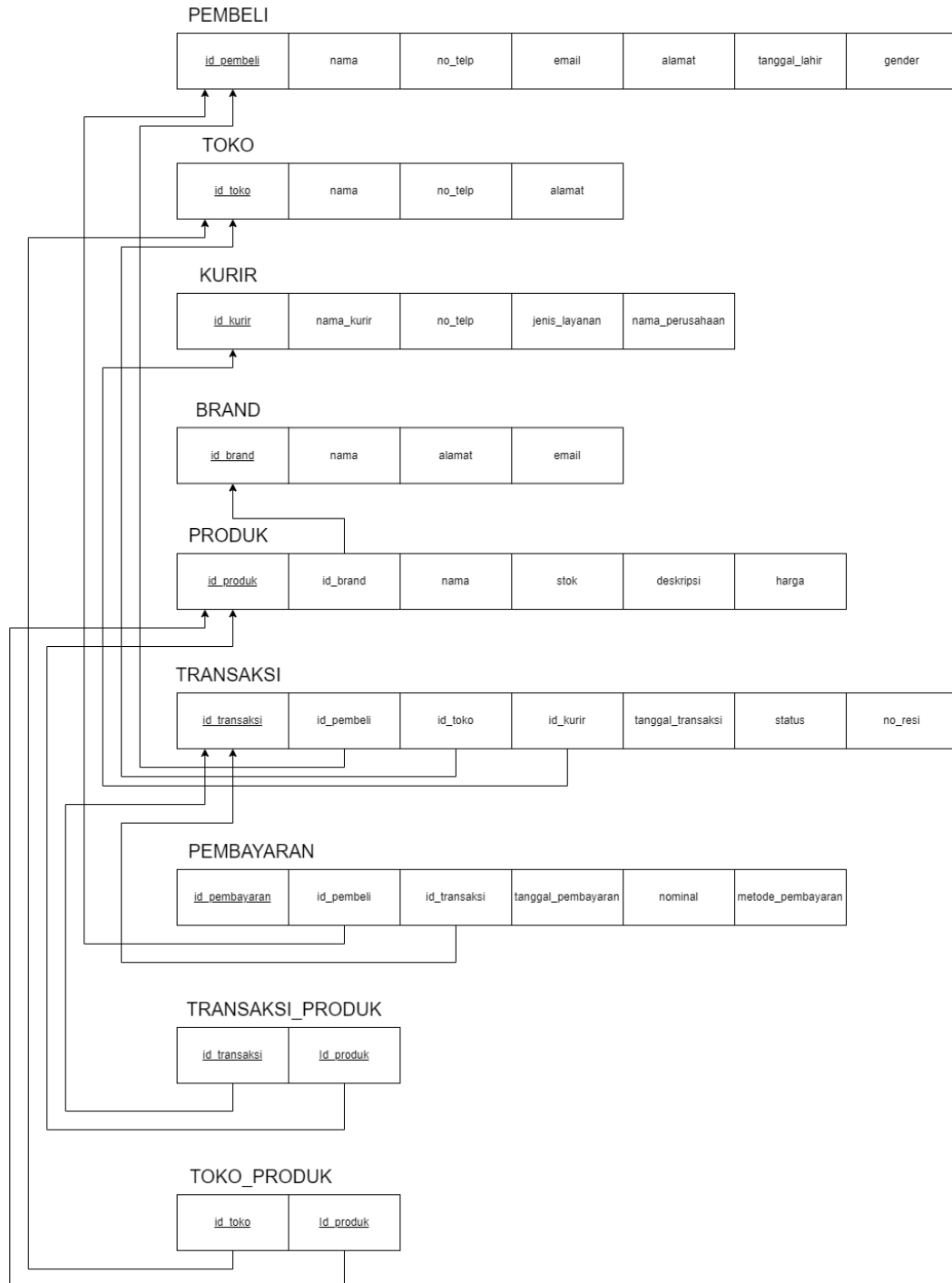
Anda **boleh menggunakan DBMS PostgreSQL yang telah terinstall di lokal komputer** masing-masing, **asalkan diberikan autentikasi sesuai *username* LDAP (*username* SIAK NG)** untuk memudahkan dalam koreksi laporan (juga dapat dengan create database dengan *name* sesuai username SIAK NG.)

II. Definisi Skema (Data Definition Language /DDL)

Untuk latihan dalam tutorial basis data, perhatikan contoh skema relasional berikut.

Misalkan Anda diminta untuk membuat basis data dengan PostgreSQL berdasarkan skema relasional berikut:

CSGE602070 Basis Data
Semester Genap 2023/2024
Tutorial 1
Basic SQL



Dengan keterangan mengenai atribut tiap tabel adalah sebagai berikut:

1. Tabel PEMBELI

CSGE602070 Basis Data
Semester Genap 2023/2024
Tutorial 1
Basic SQL

Tabel PEMBELI menyimpan informasi mengenai data pembeli:

NAMA ATRIBUT	TIPE DATA	Keterangan
id_pembeli	Varchar(10)	PK
nama	Varchar(100)	Not null
no_telp	Varchar(14)	Not null
email	Varchar(50)	Not null
alamat	Text	Not null
tanggal_lahir	Varchar(50)	Not null
gender	Char(1)	Not null

2. Tabel TOKO

Tabel TOKO menyimpan informasi mengenai data toko:

NAMA ATRIBUT	TIPE DATA	Keterangan
id_toko	Varchar(10)	PK
nama	Varchar(100)	Not null
no_telp	Varchar(14)	Not null
alamat	Text	Not null

3. Tabel KURIR

Tabel KURIR menyimpan informasi mengenai data kurir:

NAMA ATRIBUT	TIPE DATA	Keterangan
id_kurir	Varchar(10)	PK
nama_kurir	Varchar(100)	Not null
no_telp	Varchar(14)	Not null
jenis_layanan	Text	Not null
nama_perusahaan	Varchar(50)	Not null

4. Tabel BRAND

Tabel BRAND menyimpan informasi mengenai data *brand*:

CSGE602070 Basis Data
Semester Genap 2023/2024
Tutorial 1
Basic SQL

NAMA ATRIBUT	TIPE DATA	Keterangan
id_brand	Varchar(10)	PK
nama	Varchar(7)	Not null
alamat	Text	Not null
email	Varchar(50)	Not null

5. Tabel PRODUK

Tabel PRODUK menyimpan informasi mengenai data produk:

NAMA ATRIBUT	TIPE DATA	Keterangan
id_produk	Varchar(10)	PK
id_brand	Varchar(10)	FK ke BRAND.id_brand, Not null
nama	Varchar(100)	Not null
stok	Integer	Not null
deskripsi	Text	Not null
harga	Integer	Not null

6. Tabel TRANSAKSI

Tabel TRANSAKSI menyimpan informasi mengenai data transaksi:

NAMA ATRIBUT	TIPE DATA	Keterangan
id_transaksi	Varchar(10)	PK
id_pembeli	Varchar(10)	FK ke PEMBELI.id_pembeli, Not null
id_toko	Varchar(10)	FK ke TOKO.id_toko, Not null
id_kurir	Varchar(10)	FK ke KURIR.id_kurir
tanggal_transaksi	DATE	Not null
status	Varchar(20)	Not null

CSGE602070 Basis Data
Semester Genap 2023/2024
Tutorial 1
Basic SQL

no_resi	Varchar(12)	
---------	-------------	--

7. Tabel PEMBAYARAN

Tabel PEMBAYARAN menyimpan informasi mengenai data pembayaran:

NAMA ATRIBUT	TIPE DATA	Keterangan
id_pembayaran	Varchar(10)	PK
id_pembeli	Varchar(10)	FK ke PEMBELI.id_pembeli, Not null
id_transaksi	Varchar(10)	FK ke TRANSAKSI.id_transaksi, Not null
tanggal_pembayaran	DATE	Not null
nominal	Integer	Not null
metode_pembayaran	Varchar(50)	Not null

8. Tabel TRANSAKSI_PRODUK

Tabel TRANSAKSI_PRODUK menyimpan informasi mengenai transaksi produk:

NAMA ATRIBUT	TIPE DATA	Keterangan
id_transaksi	Varchar(10)	<ul style="list-style-type: none"> ➤ PK (id_transaksi, id_produk) ➤ id_transaksi merupakan FK ke TRANSAKSI.id_transaksi ➤ id_produk merupakan FK ke PRODUK.id_produk
id_produk	Varchar(10)	

9. Tabel TOKO_PRODUK

Tabel TOKO_PRODUK menyimpan informasi mengenai toko produk:

NAMA ATRIBUT	TIPE DATA	Keterangan
id_toko	Varchar(10)	<ul style="list-style-type: none"> ➤ PK (id_toko, id_produk) ➤ id_toko merupakan FK ke TOKO.id_toko ➤ id_produk merupakan FK ke PRODUK.id_produk
id_produk	Varchar(10)	

Agar tabel-tabel yang Anda buat tertata dengan rapi, maka Anda perlu membuat *schema* yang dikhususkan untuk suatu keperluan tertentu. Untuk tabel-tabel di atas, Anda dapat memasukkan ke dalam *schema* SIMART.

CSGE602070 Basis Data
Semester Genap 2023/2024
Tutorial 1
Basic SQL

Note!

Setiap perintah SQL selalu diakhiri dengan tanda titik koma (;).

Sebelum membuat tabel, Anda perlu membuat *schema* untuk menampung tabel-tabel tersebut. Untuk membuat schema pada PostgreSQL, format sintaks SQL yang digunakan adalah sebagai berikut:

```
CREATE SCHEMA [nama_schema];
```

Untuk membuat *schema* SIMART, SQL yang sesuai adalah sebagai berikut:

```
CREATE SCHEMA SIMART;
```

Untuk membuat tabel pada PostgreSQL, format sintaks SQL secara umum adalah sebagai berikut:

```
CREATE TABLE nama_schema.nama_table(  
    nama_atribut1 tipe_atribut1 [NOT NULL],  
    nama_atribut2 tipe_atribut2 [NOT NULL],  
    :  
  
    PRIMARY KEY (nama_atribut1, . . .)  
    FOREIGN KEY (nama_atribut) REFERENCES  
        nama_tabel_yang_di-refer(nama_atribut_yang_di-refer)  
    [ON DELETE RESTRICT | CASCADE | SET NULL | SET DEFAULT] [ON UPDATE  
        RESTRICT | CASCADE ]  
);
```

Keterangan:

1. Tanda "[] " menyatakan **pilihan, boleh tidak digunakan**
2. Tanda ":" " menyatakan baris-baris berikutnya serupa dengan baris sebelumnya
3. Tanda "|" " menyatakan beberapa pilihan yang dapat digunakan
4. Secara lebih lengkap dapat dilihat pada dokumentasi PostgreSQL [disini](#).

CSGE602070 Basis Data
Semester Genap 2023/2024
Tutorial 1
Basic SQL

Sesuai dengan format tersebut, maka SQL untuk membuat tabel **PEMBELI** adalah sebagai berikut:

```
CREATE TABLE SIMART.PEMBELI (  
    id_pembeli VARCHAR(10) PRIMARY KEY,  
    nama VARCHAR(100) NOT NULL,  
    no_telp VARCHAR(14) NOT NULL,  
    email VARCHAR(50) NOT NULL,  
    alamat TEXT NOT NULL,  
    tanggal_lahir DATE NOT NULL,  
    gender CHAR(1) NOT NULL  
);
```

id_pembeli	Varchar(10)
nama	Varchar(100)
no_telp	Varchar(14)
email	Varchar(50)
alamat	Text
tanggal_lahir	Varchar(50)
gender	Char(1)

Pada *default*-nya, *search path* yang dituju adalah *schema public*. Agar anda tidak perlu mengetik ulang nama *schema* sebelum membuat tabel atau mengeksekusi *query* anda dapat mengatur *search path schema* dengan SQL query sebagai berikut :

```
SET search_path to [nama_schema];
```

Untuk mengatur ke *schema* SIMART, SQL yang sesuai adalah sebagai berikut:

```
SET search_path to SIMART;
```

III. Membuat Skema dan Tabel

Sebelum membuat tabel, Anda perlu membuat *schema* untuk menampung tabel-tabel tersebut.

Untuk membuat *schema* pada PostgreSQL, format sintaks SQL yang digunakan adalah sebagai berikut:

CSGE602070 Basis Data
Semester Genap 2023/2024
Tutorial 1
Basic SQL

Setelah search path diatur, SQL untuk membuat tabel **TOKO** sebagai berikut:

```
CREATE TABLE TOKO (  
    id_toko VARCHAR(10) PRIMARY KEY,  
    nama VARCHAR(100) NOT NULL,  
    no_telp VARCHAR(14) NOT NULL,  
    alamat TEXT NOT NULL  
);
```

SQL untuk membuat tabel **KURIR** sebagai berikut:

```
CREATE TABLE KURIR (  
    id_kurir VARCHAR(10) PRIMARY KEY,  
    nama_kurir VARCHAR(100) NOT NULL,  
    no_telp VARCHAR(14) NOT NULL,  
    jenis_layanan TEXT NOT NULL,  
    nama_perusahaan VARCHAR(50) NOT NULL  
);
```

SQL untuk membuat tabel **BRAND** sebagai berikut:

```
CREATE TABLE BRAND (  
    id_brand VARCHAR(10) PRIMARY KEY,  
    nama VARCHAR(7) NOT NULL,  
    alamat TEXT NOT NULL,  
    email VARCHAR(50) NOT NULL  
);
```

SQL untuk membuat tabel **PRODUK** sebagai berikut:

```
CREATE TABLE PRODUK (  
    id_produk VARCHAR(10) PRIMARY KEY,  
    id_brand VARCHAR(10) NOT NULL,  
    nama VARCHAR(100) NOT NULL,  
    stok INTEGER NOT NULL,  
    deskripsi TEXT NOT NULL,  
    harga INTEGER NOT NULL,  
    FOREIGN KEY (id_brand) REFERENCES BRAND (id_brand) ON UPDATE  
    CASCADE ON DELETE CASCADE  
);
```

CSGE602070 Basis Data
Semester Genap 2023/2024
Tutorial 1
Basic SQL

SQL untuk membuat tabel **TRANSAKSI** sebagai berikut:

```
CREATE TABLE TRANSAKSI (  
    id_transaksi VARCHAR(10) PRIMARY KEY,  
    id_pembeli VARCHAR(10) NOT NULL,  
    id_toko VARCHAR(10) NOT NULL,  
    id_kurir VARCHAR(10),  
    tanggal_transaksi DATE NOT NULL,  
    status VARCHAR(20) NOT NULL,  
    no_resi VARCHAR(12),  
    FOREIGN KEY (id_pembeli) REFERENCES PEMBELI (id_pembeli) ON UPDATE  
    CASCADE ON DELETE CASCADE,  
    FOREIGN KEY (id_toko) REFERENCES TOKO (id_toko) ON UPDATE CASCADE  
    ON DELETE CASCADE,  
    FOREIGN KEY (id_kurir) REFERENCES KURIR (id_kurir) ON UPDATE  
    CASCADE ON DELETE CASCADE  
);
```

SQL untuk membuat tabel **PEMBAYARAN** sebagai berikut:

```
CREATE TABLE PEMBAYARAN (  
    id_pembayaran VARCHAR(10) PRIMARY KEY,  
    id_pembeli VARCHAR(10) NOT NULL,  
    id_transaksi VARCHAR(10) NOT NULL,  
    tanggal_pembayaran DATE NOT NULL,  
    nominal INTEGER NOT NULL,  
    metode_pembayaran VARCHAR(50) NOT NULL,  
    FOREIGN KEY (id_pembeli) REFERENCES PEMBELI (id_pembeli) ON UPDATE  
    CASCADE ON DELETE CASCADE,  
    FOREIGN KEY (id_transaksi) REFERENCES TRANSAKSI (id_transaksi) ON  
    UPDATE CASCADE ON DELETE CASCADE  
);
```

SQL untuk membuat tabel **TRANSAKSI_PRODUK** sebagai berikut:

```
CREATE TABLE TRANSAKSI_PRODUK (  
    id_transaksi VARCHAR(10),  
    id_produk VARCHAR(10),  
    PRIMARY KEY (id_transaksi, id_produk),  
    FOREIGN KEY (id_transaksi) REFERENCES TRANSAKSI (id_transaksi) ON  
    UPDATE CASCADE ON DELETE CASCADE,  
    FOREIGN KEY (id_produk) REFERENCES PRODUK (id_produk) ON UPDATE  
    CASCADE ON DELETE CASCADE  
);
```

CSGE602070 Basis Data
Semester Genap 2023/2024
Tutorial 1
Basic SQL

SQL untuk membuat tabel **TOKO_PRODUK** sebagai berikut:

```
CREATE TABLE TOKO_PRODUK (  
    id_toko VARCHAR(10),  
    id_produk VARCHAR(10),  
    PRIMARY KEY (id_toko, id_produk),  
    FOREIGN KEY (id_toko) REFERENCES TOKO (id_toko) ON UPDATE CASCADE  
    ON DELETE CASCADE,  
    FOREIGN KEY (id_produk) REFERENCES PRODUK (id_produk) ON UPDATE  
    CASCADE ON DELETE CASCADE  
);
```

Latihan 1.

Pada latihan ini, soal dengan tanda **[SQL]** harus dijalankan pada database masing-masing dan sertakan screenshot-nya (berupa SQL dan hasilnya) pada laporan. Sedangkan soal dengan tanda **[TRIVIA]**, tuliskan langsung jawaban Anda pada laporan

1. Pelajari sintaks penulisan data type dalam PostgreSQL [disini](#).
2. Pelajari sintaks pembuatan tabel yang ada dalam dokumentasi PostgreSQL [disini](#).
3. **[SQL]** Buatlah skema database SIMART beserta seluruh tabel yang ada sesuai dengan definisi yang telah dijelaskan sebelumnya.
4. **[TRIVIA]** Apa yang terjadi jika kita menghapus tabel yang di-refer tabel lain dengan FOREIGN KEY?
5. **[TRIVIA]** Apakah suatu tabel dapat memiliki lebih dari 1 PRIMARY KEY?

IV. Mengisi Tabel

Setelah schema dan tabel berhasil dibuat, maka tahap berikutnya adalah memasukkan data ke dalam tabel. Untuk mengisi tabel pada PostgreSQL, format sintaks SQL secara umum adalah sebagai berikut:

```
INSERT INTO nama_schema.nama_tabel[(nama_atribut_1, ...,  
nama_atribut_n)]VALUES (  
    nilai_atribut_1, ..., nilai_atribut_n  
);
```

Keterangan:

1. Tanda “[]” menyatakan pilihan, boleh tidak digunakan
2. Tanda “...” menyatakan elemen-elemen berikutnya serupa dengan elemen sebelumnya
3. Secara lebih lengkap dapat dilihat pada dokumentasi PostgreSQL [disini](#).

CSGE602070 Basis Data
Semester Genap 2023/2024
Tutorial 1
Basic SQL

Berdasarkan sintaks SQL diatas, contoh untuk mengisi data pada tabel **PEMBELI** adalah sebagai berikut:

```
INSERT INTO PEMBELI VALUES (  
    'PMB001',  
    'Daloris Beddo',  
    '08817555645',  
    'dbeddo0@gmail.com',  
    'Jl Pradah Kali Kendal I/33 A, Jawa Timur',  
    '1989-05-14',  
    'L'  
);
```

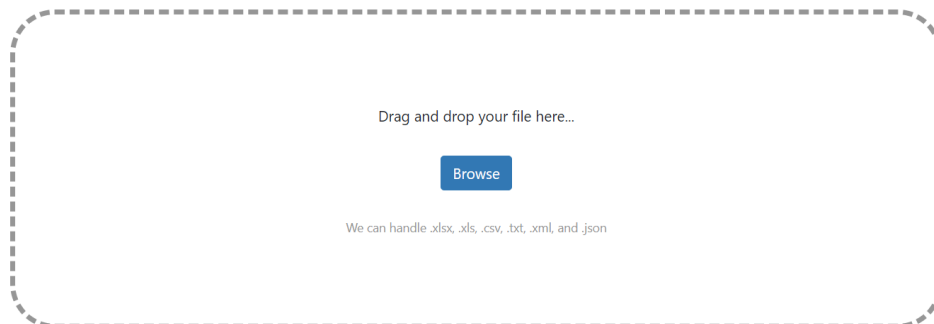
Jika sudah mengatur **search path** sebelumnya, maka kita **tidak perlu** lagi menulis **nama_schema**.

CATATAN PENTING: Anda mungkin saja mengalami error ketika memasukkan format tanggal akibat tidak sesuai format tanggal yang anda masukkan dengan format yang ada pada PostgreSQL. Untuk menyesuaikan format tanggal dengan data yang ingin dimasukkan silahkan jalankan perintah berikut:

```
SET DATESTYLE = "ISO, DMY";
```

TIPS! Setelah mencoba sintaks untuk membuat tabel, Anda perlu memasukkan data yang terdapat di Excel pada Tabel yang telah Anda buat. Untuk dapat memasukkan datanya, Anda dapat menggunakan converter Excel ke PostgreSQL secara online seperti SQLizer dengan langkah-langkah sebagai berikut :
(Link : <https://sqlizer.io/#/>)

1. Memasukkan Excel yang ingin diubah.



2. Pilih PostgreSQL sebagai database yang ingin digunakan.

CSGE602070 Basis Data
Semester Genap 2023/2024
Tutorial 1
Basic SQL

File Types



3. Biarkan “Has Header Row” terceklis. Kemudian Unchecked “Active Worksheet” dan ketikkan nama halaman sheet Excel yang akan diubah. Pada tutorial ini akan dicontohkan mengubah sheet **PEMBELI**.

☒ Has Header Row

☐ Active Worksheet

Worksheet

PEMBELI

4. Unchecked “Whole Sheet” dan masukkan rentang sel dari data yang ingin Anda konversi. Khusus untuk tabel **PEMBELI**, mulai dari record kedua karena record pertama sudah dimasukkan secara manual sebelumnya.

☐ Whole Sheet

Cell Range

A2:G41

5. Pada menu Advanced Settings, isi “Table Name” dengan nama tabel yang ingin Anda konversi. Pada tutorial ini akan dicontohkan mengubah tabel **PEMBELI**.

Advanced Settings ⬆

Table Name

PEMBELI

6. Unchecked “Check Table Exists” dan tekan tombol “Convert”. Maka akan menghasilkan data yang dapat dimasukkan ke Tabel database. Pada bagian “Treat this file as”, pastikan Anda memilih “Excel Spreadsheet”.

☐ Check Table Exists

Treat this file as

Excel Spreadsheet

Convert

7. Hiraukan bagian “CREATE TABLE ...”
Bagian yang perlu Anda salin adalah dari “INSERT INTO ...”

CSGE602070 Basis Data Semester Genap 2023/2024 Tutorial 1 Basic SQL

[Download PEMBELI.sql](#)

[Copy to Clipboard](#)

[Change Settings](#)

[Convert Another File](#)

[Tweet about SQLizer!](#)

```
CREATE TABLE "PEMBELI" (
  "PMB002" TEXT,
  "Clarabelle_Mityashin" TEXT,
  "Column_0885555530" BIGINT,
  "cmityashin1_gmail_com" TEXT,
  "JL_Gemblongan_IV_1_Jawa_Timur" TEXT,
  "Column_2002_08_09_00_00_00" TEXT,
  "P" TEXT
);
INSERT INTO "PEMBELI" VALUES ('PMB003','Katey Seares','08878555645','kseares2@yahoo.com','Jl H Amsir 10, DKI Jakarta','1987-04-03 00:00:00','P'),
('PMB004','Harriette Kippins','08896555395','hkippins3@gmail.com','Jl Mampang Prapatan Raya 17 D, DKI Jakarta','15/10/1980','L'),
('PMB005','Davin Clemenceau','08812555364','dclemenceau4@hotmail.com','Jl Manyar Sabrangan IX/76 F, Jawa Timur','1984-12-03 00:00:00','L'),
('PMB006','Rochell Cornewell','08857555064','rcornewell5@yahoo.com','Jl RS Fatmawati 39 ITC Fatmawati 175, DKI Jakarta','13/05/2001','L'),
('PMB007','Tabbi Poundesford','08818555927','tpoundesford6@gmail.com','Jl Pancoran Birt II 4, DKI Jakarta','1999-01-12 00:00:00','L'),
('PMB008','Correy Castagnasso','08898555205','ccastagnasso7@gmail.com','Jl Kapt Muslim Halvetia, Sumatera Utara','23/11/1997','L'),
('PMB009','Una Guirardin','08814555871','lguirardin8@gmail.com','Jl Wonorejo I/25 A, Jawa Timur','22/12/1993','L'),
('PMB010','Kip Pays','08896555544','kpays9@yahoo.com','Jl Dr Wahidin 62 B, Jawa Tengah','28/07/1993','P'),
('PMB011','Kathleen Gravenell','08838555682','kgravenella@hotmail.com','Jl Marunda Baru 5, DKI Jakarta','1990-06-06 00:00:00','L'),
('PMB012','Brigitte Robinet','08816555389','brolinetb@gmail.com','Jl Daan Mogot Km 1/20 M, DKI Jakarta','18/08/1998','L'),
('PMB013','Dix Somers','08878555864','dsomersc@gmail.com','Jl Halteu Utr 91/74, Jawa Barat','1981-02-09 00:00:00','L'),
('PMB014','Gonzales Whooley','08899555579','gwhooleyd@gmail.com','Jl Bungur Besar 87-8/F, DKI Jakarta','30/05/1998','P'),
('PMB015','Karilynn Aspole','08819555663','kaspolee@hotmail.com','Jl Ibrahim Umar 11, Sumatera Utara','2002-08-03 00:00:00','P');
```

Pastikan untuk menghapus tanda kutip (") pada nama tabel.

8. Anda dapat menyalin hasil convert ke Tabel yang telah Anda buat dan telah Anda sesuaikan.
9. Lakukan langkah yang sama untuk mengonversi tabel lainnya.

Latihan 2.

Pada latihan ini, soal dengan tanda [SQL] harus dijalankan pada database masing-masing dan sertakan screenshot-nya (berupa SQL dan hasilnya) pada laporan. Sedangkan soal dengan tanda [TRIVIA], tuliskan langsung jawaban Anda pada laporan.

1. Pelajari sintaks penulisan data type dalam PostgreSQL [disini](#).
2. Pelajari sintaks pengisian tabel yang ada dalam dokumentasi PostgreSQL [disini](#).
3. [SQL] Masukkan data untuk seluruh tabel menggunakan sqlizer seperti yang telah dicontohkan diatas. Pastikan *screenshot* sintaks untuk melakukan Insert data. Data yang harus dimasukkan pada setiap tabel dapat dilihat dalam lampiran excel: data_SIMART.xlsx
4. [TRIVIA] Bagaimana cara untuk mengisi suatu kolom di tabel dengan suatu **Default Value**. Contoh: memasukkan nilai 0 di semua kolom harga di tabel produk.

V. Melihat Tabel

Untuk mencari tahu apakah tabel yang Anda buat sudah masuk ke dalam DBMS, Anda dapat mencetak daftar tabel yang ada pada DBMS dengan perintah:

```
\d
```

Untuk melihat struktur tabel tertentu, misal tabel **TOKO**, dapat dilakukan dengan perintah:

```
\d TOKO
```

Untuk melihat keseluruhan isi tabel tertentu dapat dilakukan dengan perintah:

CSGE602070 Basis Data
Semester Genap 2023/2024
Tutorial 1
Basic SQL

```
SELECT * FROM [nama_tabel];
```

Untuk memastikan karakteristik tabel dan isi dari tabel, jalankan script di bawah ini.

1. Tampilkan struktur definisi dari semua tabel yang ada pada skema SIMART.

```
\d PEMBELI
\d TOKO
\d KURIR
\d BRAND
\d PRODUK
\d TRANSAKSI
\d PEMBAYARAN
\d TRANSAKSI_PRODUK
\d TOKO_PRODUK
```

2. Tampilkan keseluruhan isi dari semua tabel yang ada pada skema SIMART

```
SELECT * FROM PEMBELI;
SELECT * FROM TOKO;
SELECT * FROM KURIR;
SELECT * FROM BRAND;
SELECT * FROM PRODUK;
SELECT * FROM TRANSAKSI;
SELECT * FROM PEMBAYARAN;
SELECT * FROM TRANSAKSI_PRODUK;
SELECT * FROM TOKO_PRODUK;
```

Latihan 3.

Pada latihan ini, soal dengan tanda **[SQL]** harus dijalankan pada database masing-masing dan sertakan *screenshot*-nya (berupa SQL dan hasilnya) pada laporan. Sedangkan soal dengan tanda **[TRIVIA]**, tuliskan langsung jawaban Anda pada laporan.

1. **[SQL]** Tampilkan struktur definisi dari semua tabel yang ada pada skema SIMART.
2. **[SQL]** Tampilkan keseluruhan isi dari semua tabel yang ada pada skema SIMART.
3. **[TRIVIA]** Apa yang terjadi jika kita menuliskan lebih dari 1 nama tabel pada sintaks SELECT?

CSGE602070 Basis Data Semester Genap 2023/2024 Tutorial 1 Basic SQL

VI. Mengubah Data Tabel

Untuk mengubah data pada suatu tabel, digunakan operasi UPDATE sebagai berikut:

```
UPDATE [Nama_SCHEMA].NAMA_TABEL  
SET nama_atribut_1 = nilai_atribut_1_baru, nama_atribut_2 =  
nilai_atribut_2_baru  
[WHERE CONDITIONAL_STATEMENT];
```

Secara lebih lengkap dapat dilihat pada dokumentasi PostgreSQL [disini](#).

Contoh Sintaks Update:

```
UPDATE KURIR  
SET nama_perusahaan = 'Pos Indonesia'  
WHERE nama_kurir = 'Almira Rahimah';
```

Sintaks diatas akan mengubah nama perusahaan menjadi “Pos Indonesia” untuk setiap kurir yang bernama “Almira Rahimah”. Karena *search_path* sudah diatur sebelumnya, maka langsung ditulis nama tabelnya saja.

VII. Menghapus Data Tabel

Untuk menghapus data dari suatu tabel, digunakan operasi DELETE sebagai berikut:

```
DELETE FROM [Nama_SCHEMA].NAMA_TABLE  
[WHERE CONDITIONAL_STATEMENT];
```

Secara lebih lengkap dapat dilihat pada dokumentasi PostgreSQL [disini](#).

Contoh Sintaks Delete:

```
DELETE FROM PEMBELI  
WHERE id_pembeli = 'PMB040';
```

Sintaks diatas akan menghapus data pembeli yang memiliki id “PMB040”. Karena *search_path* sudah diatur sebelumnya, maka cukup ditulis nama tabelnya saja.

VIII. Modifikasi Definisi Tabel

Modifikasi dapat dilakukan pada tabel yang sudah ada. Salah satunya adalah merubah tipe data kolom.

Eksekusi *command* di bawah ini:

CSGE602070 Basis Data
Semester Genap 2023/2024
Tutorial 1
Basic SQL

```
INSERT INTO BRAND VALUES ('BRND05', 'Motorola', 'JL. Diponegoro No. 3,  
Tarakan', 'motorolaID@motorola.com');
```

Command diatas akan menimbulkan *error*, hal ini dikarenakan kolom nama yang ingin kita masukan pada tabel BRAND memiliki karakter yang lebih dari 7 yang melanggar aturan pada tabel BRAND yang meng-*limit* nama sampai 7 karakter saja karena tipe datanya adalah VARCHAR(7). Oleh karena itu, agar tidak terjadi *error*, kita dapat mengubah definisi tabelnya menjadi seperti dibawah ini:

Contoh Sintaks Modifikasi Definisi Tabel:

```
ALTER TABLE BRAND ALTER COLUMN nama SET DATA TYPE VARCHAR(255);
```

Eksekusi kembali *command* di bawah ini:

```
INSERT INTO BRAND VALUES ('BRND05', 'Motorola', 'JL. Diponegoro No. 3,  
Tarakan', 'motorolaID@motorola.com');
```

Setelah memodifikasi definisi tabel BRAND yang awalnya tipe data kolom nama merupakan VARCHAR(7), sekarang berubah tipe datanya menjadi VARCHAR(255) sehingga *command* diatas dapat dijalankan karena nama *brand* Motorola memiliki karakter yang kurang atau sama dengan 255. Secara lengkap dapat dilihat pada dokumentasi PostgreSQL [disini](#).

IX. SQL Query

Untuk mendapatkan data yang ada di dalam Tabel, dapat digunakan SQL query. Perintahnya adalah sebagai berikut:

```
SELECT <attribute_list>  
FROM <table_list>  
[WHERE <condition>]  
[ORDER by <attribute_list>];
```

Jika kondisi WHERE tidak dicantumkan maka SQL akan menampilkan seluruh hasil yang ada karena dianggap sebagai WHERE TRUE. Kondisi WHERE sering digunakan untuk memfilter data. Secara lengkap dapat dilihat pada dokumentasi PostgreSQL [disini](#).

Contoh Menampilkan Semua Data Tetapi Diurutkan Berdasarkan Suatu Kolom Menggunakan Klausa ORDER BY:

CSGE602070 Basis Data Semester Genap 2023/2024 Tutorial 1 Basic SQL

```
SELECT nama  
FROM PEMBELI  
ORDER BY nama ASC;
```

Klausula ORDER BY dapat digunakan untuk mengurutkan hasil berdasarkan suatu kolom. ASC berarti *ascending* dan DESC berarti *descending*. Jika kolom yang di-order berisi *string* maka akan diurutkan secara abjad, jika menggunakan ASC maka akan diurutkan dari A -> Z sedangkan DESC akan diurutkan dari Z -> A. Sebaliknya, jika kolom yang di-order berisi data numerik maka akan diurutkan seperti biasa. *Command* diatas berguna untuk mendapatkan kolom nama dari tabel pembeli dan di-order berdasarkan nama pembeli secara abjad dan *ascending*.

Secara lengkap dapat dilihat pada dokumentasi PostgreSQL [disini](#).

Contoh Memfilter Kolom Yang Berisi String Menggunakan Klausula LIKE:

```
SELECT alamat FROM PEMBELI WHERE alamat LIKE '%DKI Jakarta';
```

Klausula LIKE dapat digunakan untuk memfilter *string*. Contoh di atas berguna untuk mengambil alamat yang memiliki akhiran DKI Jakarta.

Secara lengkap dapat dilihat pada dokumentasi PostgreSQL [disini](#).

Contoh Mendapatkan Data Yang Unik Menggunakan Klausula DISTINCT:

TANPA DISTINCT:

```
SELECT metode_pembayaran  
FROM PEMBAYARAN;
```

DENGAN DISTINCT:

```
SELECT DISTINCT metode_pembayaran  
FROM PEMBAYARAN;
```

Command diatas berguna untuk mendapatkan nama metode_pembayaran yang unik menggunakan klausula DISTINCT.

Secara lengkap dapat dilihat pada dokumentasi PostgreSQL [disini](#).

CSGE602070 Basis Data
Semester Genap 2023/2024
Tutorial 1
Basic SQL

Sneak Peek SQL Kedepannya (Join dan Alias):

```
SELECT B.nama as nama_brand, select P.nama as nama_produk  
FROM BRAND as B, PRODUK AS P  
WHERE B.id_brand == P.id_brand;
```

Klausa AS berguna untuk memberikan alias. Penggunaan AS pada SELECT akan merubah nama kolom hasil dengan nama alias yang diberikan sedangkan penggunaan AS pada FROM akan membuat nama tabel dengan alias baru sehingga tidak perlu untuk menulis BRAND.id_brand cukup B.id_brand. Untuk *sneak peek* kedepannya, *command* diatas berguna untuk men-*join*-kan tabel BRAND dengan tabel PRODUK berdasarkan id_brand yang akan menge-*list* nama_brand dan nama_produk yang sesuai contoh hasilnya adalah Samsung, Samsung Galaxy S21.

X. Soal

Latihan 4.

Pada latihan ini, soal dengan tanda **[SQL]** harus dijalankan pada database masing-masing dan sertakan *screenshot*-nya (berupa SQL dan hasilnya) pada laporan.

1. **[SQL]** Tampilkan nama, id_pembeli, dan tanggal lahir dari tabel PEMBELI yang memiliki huruf D pada namanya dan urutkan dari yang paling muda ke yang paling tua.
2. **[SQL]** Tampilkan nama, id_toko dari tabel TOKO yang memiliki alamat selain dari kota DKI Jakarta
3. **[SQL]** Tampilkan nama, harga dari tabel PRODUK yang memiliki harga lebih dari 5 juta dan merupakan HP samsung dan urutkan dari yang termahal sampai termurah
4. **[SQL]** Tampilkan nama-nama perusahaan kurir
5. **[SQL]** Tampilkan nama, stok dari tabel PRODUK yang bukan merupakan produk samsung dan memiliki stok kurang dari 100