

**LAPORAN DATA WRANGLING C
PYTHON DAN SQL
T05**



DISUSUN OLEH:

Mohamad Ibnu Fajar Maulana (21083010106)

DOSEN PENGAMPU:

KARTIKA MAULIDA HINDRAYANI, S.Kom, M.Kom

**PROGRAM STUDI SAINS DATA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"
JAWA TIMUR
2022**

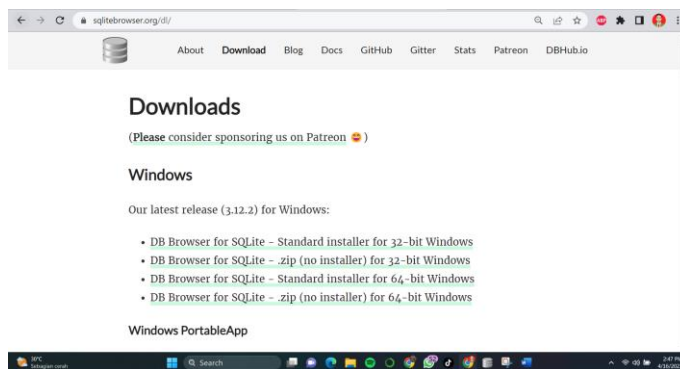
Jawablah pertanyaan dari soal-soal berikut ini:

1. Buat Database yang kamu tentukan menggunakan perintah pada python
2. Buat minimal 2 tabel dan tambahkan data menggunakan perintah pada python
3. Lakukan pemutakhiran data pada tabel yang telah dibuat menggunakan perintah pada python
4. Lakukan pengelompokan data pada tabel yang telah dibuat menggunakan perintah pada python
5. Lakukan join data pada tabel yang telah dibuat menggunakan perintah pada python

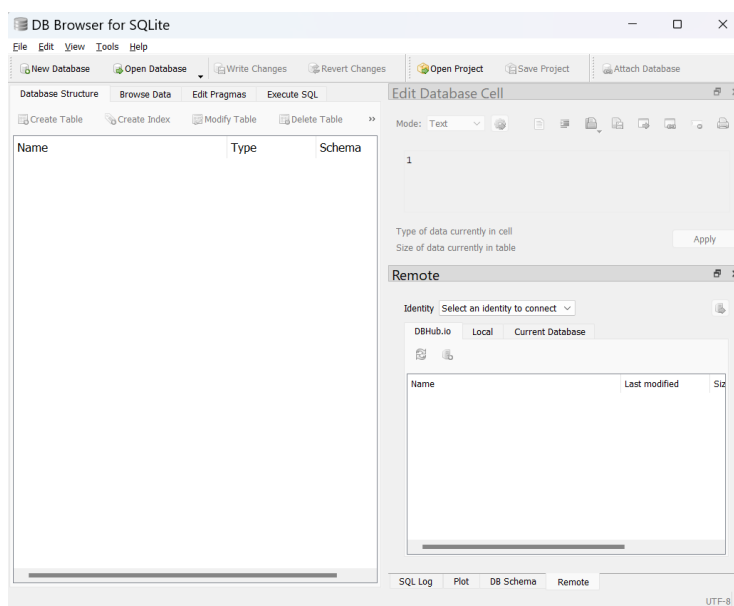
Jawab

Sebelum menjawab pertanyaan diatas ada beberapa hal yang perlu dilakukan terlebih dahulu

- Mendownload aplikasi sqlite pada web atau link : <https://sqlitebrowser.org/dl/>



- Sesuaikan laptopmu jika ingin mendownload, jika sudah mendownload aplikasi *sqlite*, kemudian instalasi aplikasi tersebut dan akan menampilkan seperti gambar dibawah ini:



Selanjutnya kita menjawab pertanyaan

1. Buat Database yang kamu tentukan menggunakan perintah pada python

Jawab:

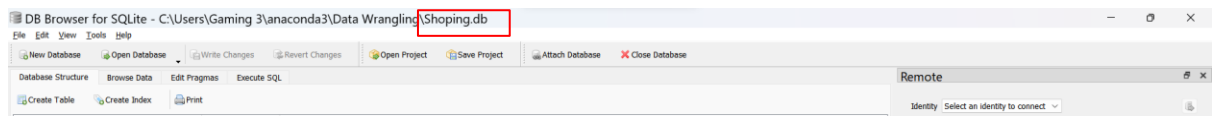
- Pada project yang saya lakukan saya membuat database Shopping.db yang kemudian dikoneksikan kepada aplikasi *sqlite* dengan kode skrip berikut:

```
[1]: import sqlite3

# membuat koneksi ke database
conn = sqlite3.connect('Belanja.db')

# membuat cursor
c = conn.cursor()
```

- Selanjutnya kita lihat apakah database sudah terbuat:



- Gambar diatas menunjukkan database sudah dibuat, bernama Shopping.db, Selanjutnya kita membuat tabel pada database untuk menjawab pertanyaan nomor 2

2. Buat minimal 2 tabel dan tambahkan data menggunakan perintah pada python

Jawab:

- Pertama-tama lakukan pengkodean di python untuk membuat tabel pertama, seperti gambar berikut ini:

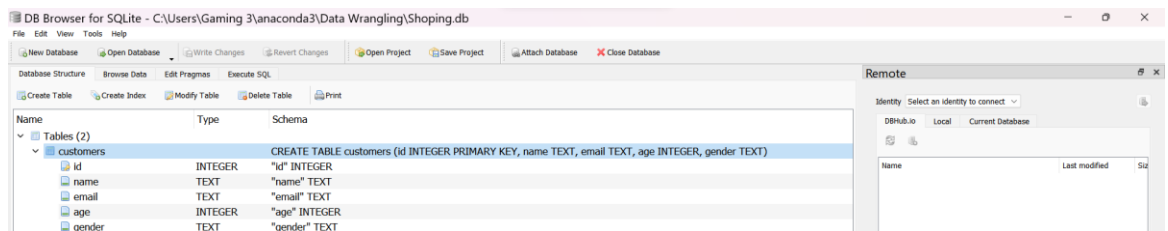
```
# membuat tabel pertama
c.execute('CREATE TABLE IF NOT EXISTS customers
          (id INTEGER PRIMARY KEY,
           name TEXT,
           email TEXT,
           age INTEGER,
           gender TEXT)')
```

Gambar diatas membuat tabel customer yang berisikan “*id, name, email, age, gender*” yang nantinya akan dibantu dengan data-data customers untuk tabel customers, dengan kodingan dibawah ini:

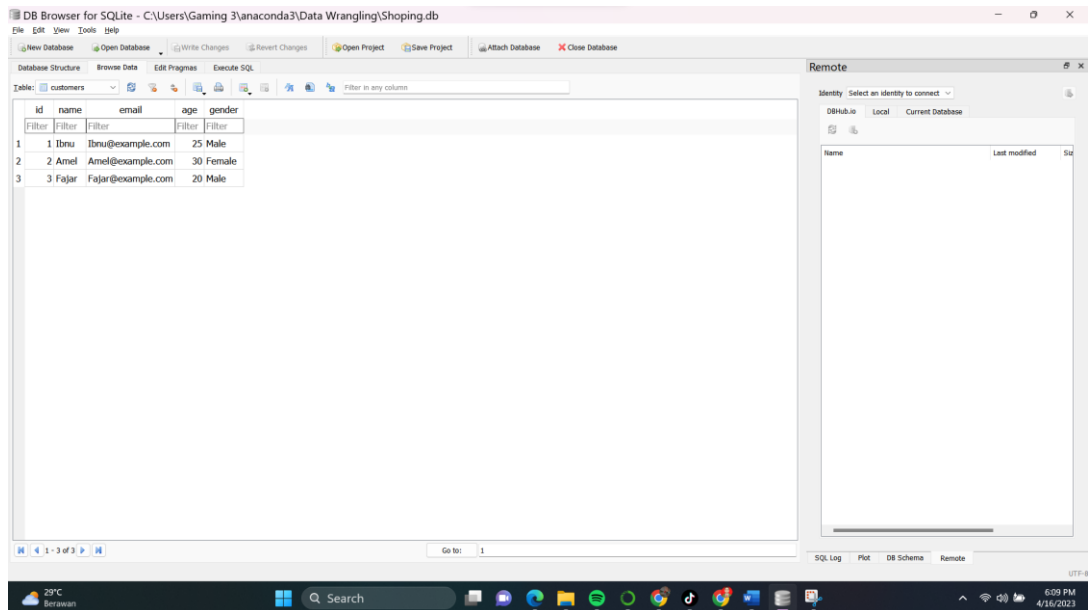
```
# menambahkan data ke tabel customers
data_customers = [(1, 'Ibnu', 'Ibnu@example.com', 25, 'Male'),
                  (2, 'Amel', 'Amel@example.com', 30, 'Female'),
                  (3, 'Fajar', 'Fajar@example.com', 20, 'Male')]
c.executemany('INSERT INTO customers VALUES (?, ?, ?, ?, ?)', data_customers)
```

Gambar diatas melakukan penambahan data customers ke tabel customers yang mana data tersebut berisikan “*id, nama, email, umur, jenis kelamin*” customer.

- Setelah itu didapatkan tabel customers sebagai berikut:



- Kemudian kita mendapatkan data-data customers, sebagai berikut:



- Selanjutnya kita akan membuat tabel kedua yaitu tabel order, dengan kode berikut:

```
# membuat tabel kedua
c.execute('''CREATE TABLE IF NOT EXISTS orders
            (id INTEGER PRIMARY KEY,
             customer_id INTEGER,
             product TEXT,
             quantity INTEGER,
             FOREIGN KEY(customer_id) REFERENCES customers(id))''')
```

Pada gambar diatas merupakan tabel orders yang memiliki “*id, customer_id, product, quantity*”

- Setelah itu, menambah data-data ke tabel orders dengan kode berikut:

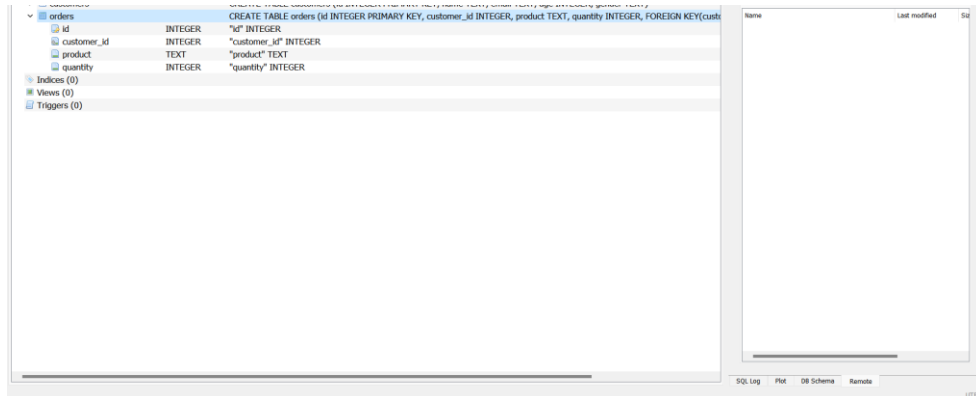
```
# menambahkan data ke tabel orders
data_orders = [(1, 1, 'Sepatu', 2),
                (2, 1, 'Sendal', 1),
                (3, 2, 'Celana Panjang', 3)]
c.executemany('INSERT INTO orders VALUES (?, ?, ?, ?)', data_orders)
```

Gambar diatas tersebut berisikan “*id customers, id orders, nama barang, dan kuantitas(jumlah)*”.

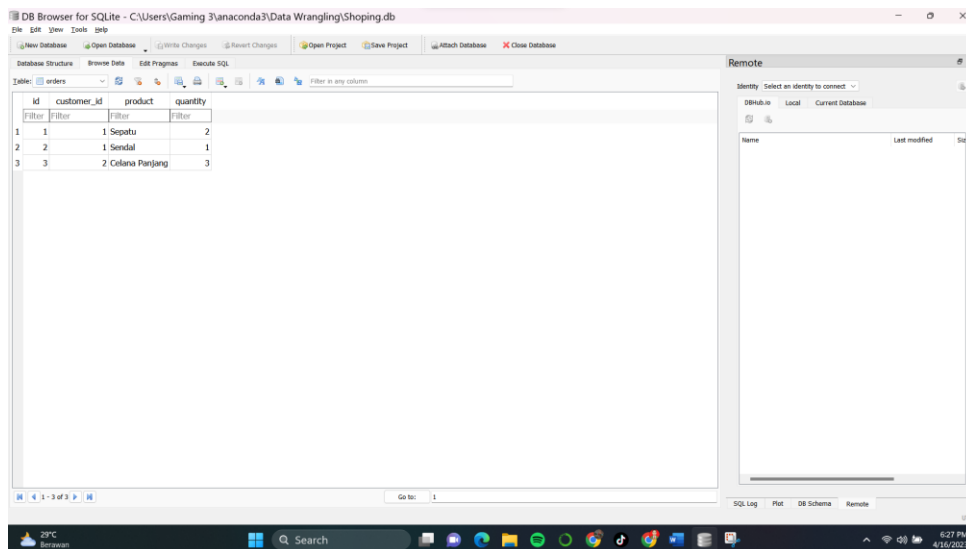
- Setelah menjalani pemrosesan pengkodean diatas maka lakukanlah kodingan dibawah ini, untuk mengcommit:

```
# commit perubahan dan menutup koneksi
conn.commit()
conn.close()
```

- Selanjutnya akan didapatkan tabel orders pada *sqlite* sebagai berikut:



- Setelah itu, didapatkan data-data tabel orders pada *sqlite* sebagai berikut:



3. Lakukan pemutakhiran data pada tabel yang telah dibuat menggunakan perintah pada python

Jawab:

- Untuk melakukan pemutakhiran data pada tabel menggunakan perintah Python, kita dapat menggunakan perintah UPDATE. Berikut adalah contoh penggunaan perintah UPDATE pada tabel customers, Berikut kode python:

```
import sqlite3

# membuat koneksi ke database
conn = sqlite3.connect('Shopping.db')

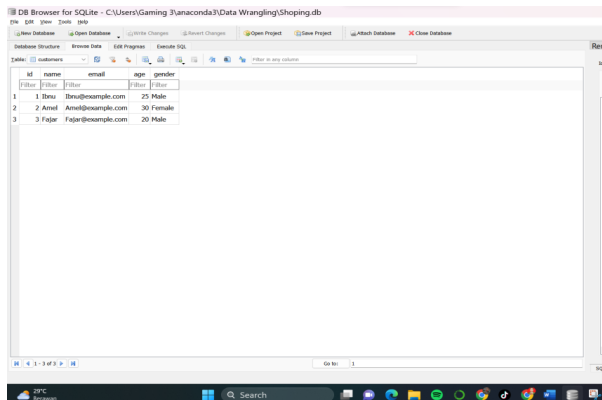
# membuat cursor
c = conn.cursor()

# update data di tabel customers
c.execute('UPDATE customers SET email = 'Ibnu@gmail.com' WHERE id = 1')
c.execute('UPDATE customers SET age = 19 WHERE id = 1')
c.execute('UPDATE customers SET email = 'Ame1@gmail.com' WHERE id = 2')
c.execute('UPDATE customers SET age = 19 WHERE id = 2')
c.execute('UPDATE customers SET email = 'Fajar@gmail.com' WHERE id = 3')
c.execute('UPDATE customers SET age = 19 WHERE id = 3')

# commit perubahan dan menutup koneksi
conn.commit()
conn.close()
```

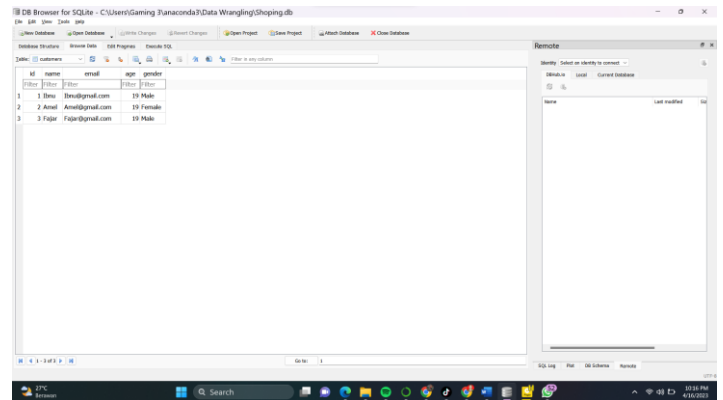
Gambar diatas mengubah email dan umur customers pada *sqlite* berdasarkan *id* *customers*, mari kita lihat perbedaan sebelum di Update dan setelah di update

Sebelum UPDATE:



id	name	email	age	gender
1	Ibnu	Ibnu@sample.com	25	Male
2	Ame1	Ame1@sample.com	30	Female
3	Fajar	Fajar@sample.com	20	Male

Setelah UPDATE:



id	name	email	age	gender
1	Ibnu	Ibnu@gmail.com	19	Male
2	Ame1	Ame1@gmail.com	19	Female
3	Fajar	Fajar@gmail.com	19	Male

- Lakukan pengelompokan data pada tabel yang telah dibuat menggunakan perintah pada python

Jawab:

- Untuk melakukan pengelompokan data pada tabel, Anda dapat menggunakan perintah SQL GROUP BY. Berikut adalah contoh kode Python untuk mengelompokkan data pada tabel orders berdasarkan produk, berikut kode skrip pythonnya:

```
[2]: # membuat koneksi ke database
conn = sqlite3.connect('Shopping.db')
c = conn.cursor()

# melakukan query dengan group by
c.execute('SELECT product, SUM(quantity)
          FROM orders
          GROUP BY product')
data = c.fetchall()
print(data)

# menutup koneksi
conn.close()
```

Kemudian mendapatkan output sebagai berikut:

```
[('Celana Panjang', 3), ('Sendal', 1), ('Sepatu', 2)]
```

Gambar diatas mengelompokkan data barang dan kuantitas barang yang dibeli, jadi pada gambar diatas Celana Panjang dibeli dengan jumlah 3, Sendal dibeli dengan jumlah 1, Sepatu dibeli dengan jumlah 2

5. Lakukan join data pada tabel yang telah dibuat menggunakan perintah pada python

Jawab:

- Untuk melakukan join data pada tabel, Anda dapat menggunakan perintah SQL JOIN. Berikut adalah contoh kode Python untuk menggabungkan data pada tabel customers dan orders, berikut kode skrip pythonnya:

```
[5]: # membuat koneksi ke database
conn = sqlite3.connect('Shopping.db')
c = conn.cursor()

# melakukan join tabel
c.execute('''SELECT customers.name, orders.product, orders.quantity
            FROM customers
            JOIN orders ON customers.id = orders.customer_id''')
data = c.fetchall()
print(data)

# menutup koneksi
conn.close()
```

- Kemudian mendapatkan output:

```
[('Ibnu', 'Sepatu', 2), ('Ibnu', 'Sendal', 1), ('Amel', 'Celana Panjang', 3)]
```

Pada gambar diatas menghasilkan output bahwa Ibnu memesan 2 sepatu, Ibnu memesan 1 Sendal dan Amel memesan 3 Celana Panjang.