

**LAPORAN DATABASE UNTUK SAINS DATA**  
**“Sistem Belanja *Online*”**



**Disusun oleh:**

**KELOMPOK B**

**KELAS DATABASE UNTUK SAINS DATA (A)**

**Irba Alifa Taqiyya     2006571223**

**Azizah Annurohmah   2006572983**

**Andini Assyahidah    2006571040**

**Agustinus Bravy        2006521300**

**Tulus Setiawan         2006568802**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**  
**UNIVERSITAS INDONESIA**

**2022**

## DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>2</b>
<b>BAB I PENDAHULUAN.....</b>	<b>3</b>
1.1 Latar Belakang.....	3
1.2 Tujuan.....	3
1.3 Rumusan Masalah.....	4
<b>BAB II METODE PENELITIAN.....</b>	<b>5</b>
2.1 Langkah-Langkah Kerja.....	5
2.2 <i>Software</i> Pendukung.....	6
<b>BAB III HASIL DAN PEMBAHASAN.....</b>	<b>7</b>
3.1 Struktur <i>Database</i> .....	7
3.2 ER Diagram.....	9
3.3 Relasi Antar Tabel.....	10
3.4 Simulasi Data.....	10
3.5 Analisis Data.....	18
<b>BAB IV PENUTUP.....</b>	<b>23</b>
4.1 Kesimpulan.....	23
<b>DAFTAR PUSTAKA.....</b>	<b>24</b>

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan bisnis secara *online* di Indonesia saat ini sangat pesat, salah satunya adalah dengan melakukan belanja secara *online*. Belanja *online* atau dikenal dengan sebutan *e-commerce* merupakan salah satu cara berbelanja yang dilakukan melalui alat komunikasi elektronik, dimana pembeli tidak perlu susah payah datang ke toko untuk membeli barang yang diperlukan, hanya perlu memesan barang melalui internet dan melakukan pembayaran, kemudian barang akan dikirim oleh toko *online* ke tempat pembeli.

Sebuah bisnis *online* tentunya memiliki dan memerlukan sebuah sistem yang dapat menggambarkan transaksi yang dilakukan oleh pembelinya. Sistem memerlukan data pembeli (nama, alamat, nomor telepon), data penjualan (tanggal, jam pembelian), produk yang dibeli (lini produk, jumlah, harga), dan data toko (cabang, kota). Dengan adanya sistem ini, sebuah perusahaan *online* dapat mengetahui alur dari proses jual-beli yang terjadi secara detail. Selain itu, perusahaan juga dapat memperoleh *insight* mengenai tren pembelian dan memanfaatkannya untuk melakukan perencanaan perusahaan di masa depan.

Berdasarkan uraian di atas, kami tertarik untuk membuat suatu skema “Sistem Belanja *Online*” yang nantinya dapat digunakan untuk membuat dan mengembangkan sebuah database sebuah perusahaan *e-commerce*.

### 1.2 Tujuan

Tujuan yang ingin dicapai dalam proyek ini adalah sebagai berikut:

1. Untuk membuat dan mengembangkan database sebuah *e-commerce* sehingga mampu memberikan manfaat bagi manajemen.
2. Untuk memudahkan tim data di perusahaan *e-commerce* dalam melakukan pengolahan data dengan cepat dan akurat karena data-data sudah terintegrasi di dalam database.
3. Untuk memberikan kemudahan dalam pelayanan yang lebih baik dan efisien kepada para *customer* ataupun *merchant* melalui otomatisasi.

4. Untuk memudahkan tim dalam proses pencatatan dan pencarian data transaksi serta administrasi.

### **1.3 Rumusan Masalah**

Berdasarkan latar belakang maka dapat dirumuskan suatu pokok masalah yang kemudian disusun dalam bentuk pertanyaan sebagai berikut:

1. Berapa pendapatan per minggu dari ketiga toko?
2. Berapa rata-rata pendapatan toko dari pelanggan yang merupakan member dan non-member?
3. Berapa perbandingan keuntungan antara lini produk?
4. Bagaimana perbandingan keuntungan antar cabang toko?

## BAB II

### METODE PENELITIAN

#### 2.1 Langkah-Langkah Kerja

Dalam proses mengerjakan proyek ini, langkah-langkah kerja yang kami lakukan diantaranya adalah sebagai berikut:

1. Mencari Data

Data pada database proyek ini merupakan kumpulan data yang diperoleh dari sekumpulan dataset yang bersumber dari *KAGGLE*.

2. Pembuatan ER Diagram dan Kriteria Database

Pembuatan ER Diagram dilakukan pada saat dataset yang diperlukan sudah terkumpul, dimana ER Diagram ini digunakan sebagai konsep yang akan mendasari pembuatan database. Adapun kriteria database yang berfungsi untuk memperjelas alur dari ER Diagram.

3. Merapikan Data

Dataset yang sudah terkumpul kemudian akan dirapikan atau dilakukan *data wrangling*, dimana hasil *output*-nya merupakan kumpulan tabel (*Customer*, *Product*, *Store*, *Purchase*) beserta dengan atribut dan *record*-nya masing-masing. Lalu, kumpulan tabel tersebut disimpan ke dalam file dengan format *csv* yang dimana nantinya akan digunakan untuk membuat database.

4. Pembuatan Database

Pembuatan Database ini berdasarkan file *csv* yang telah disimpan sebelumnya dimana file tersebut berisikan tabel-tabel (*Customer*, *Product*, *Store*, *Purchase*) beserta dengan atribut dan *record*-nya masing-masing.

5. Perumusan Masalah

Berdasarkan database yang terbentuk kemudian dibuat rumusan masalah untuk menjawab tujuan dari dilakukannya proyek ini.

6. Analisis Data

Analisis data dalam proyek ini dilakukan dengan menggunakan bantuan *software* SQLite dan juga *Google Colab*.

7. Visualisasi Hasil Analisis

Dibuat visualisasi dari hasil analisis data pada langkah sebelumnya dengan tujuan agar hasil analisis dapat lebih mudah untuk dipahami dan lebih ringkas.

## 8. Pembuatan Kesimpulan

Menyimpulkan hasil yang diperoleh dari penelitian yang dilakukan untuk menjawab tujuan dari proyek yang telah ditentukan sebelumnya.

## 2.2 *Software Pendukung*

Dalam melakukan dan menyelesaikan proyek ini, kami menggunakan bantuan *software* SQLite dan Google Colab dengan menggunakan modul `sqlite3` untuk memudahkan dalam melakukan proses *data wrangling*, pembuatan database dan analisis data.

## BAB III

### HASIL DAN PEMBAHASAN

#### 3.1 Struktur *Database*

Terdapat empat tabel dalam *database* sistem belanja *online* ini, tabel-tabel tersebut yaitu:

1. Tabel *Customer*

Nama tabel : customer

Digunakan : Untuk menyimpan data pelanggan

*Primary key* : Customer\_ID

Nama Field	Tipe Data	Ukuran Field	Keterangan
Customer_ID	Varchar	14	<i>Primary key</i>
Customer_type	Varchar	6	
Phone_Number	Integer	13	
Name	Varchar	28	
Address	Varchar	30	
Gender	Varchar	1	

2. Tabel *Product*

Nama tabel : product

Digunakan : Untuk menyimpan data produk

*Primary key* : Product\_ID

Nama Field	Tipe Data	Ukuran Field	Keterangan
Product_ID	Varchar	12	<i>Primary key</i>
Product_line	Varchar	22	
Unit Price	Number	5,2	
Tax_5	Number	7,4	
Branch	Varchar	1	<i>Foreign key</i> dari store

### 3. Tabel *Purchase*

Nama tabel : purchase

Digunakan : Untuk menyimpan data pembelian yang dilakukan oleh pelanggan

*Primary key* : Invoice\_ID

Nama Field	Tipe Data	Ukuran Field	Keterangan
Invoice_ID	Varchar	11	<i>Primary key</i>
Quantity	Integer	3	
Total	Number	8,4	
Date	Number	7,4	
Time	Varchar	5	
Payment	Varchar	11	
cogs	Number	6,2	
gross_margin_percentage	Number	11,9	
gross_income	Number	7,4	
Rating	Number	4,1	
Product_ID	Varchar	12	<i>Foreign key</i> dari product
Customer_ID	Varchar	14	<i>Foreign key</i> dari customer

### 4. Tabel *Store*

Nama tabel : store

Digunakan : Untuk menyimpan data cabang toko

*Primary key* : Branch

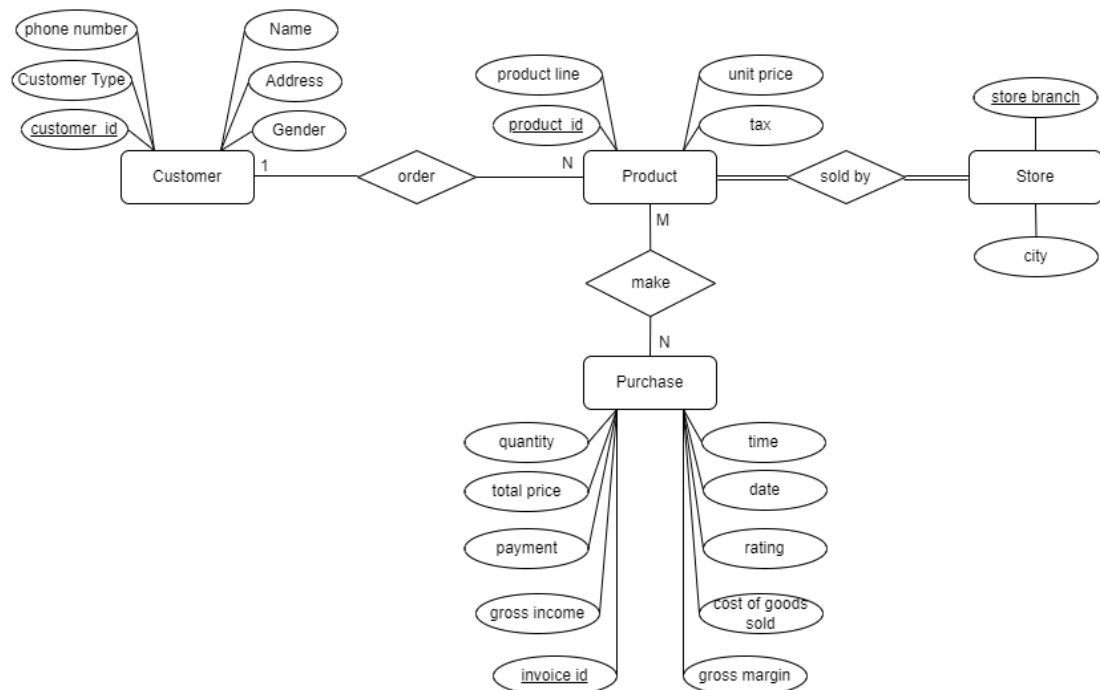
Nama Field	Tipe Data	Ukuran Field	Keterangan
Branch	Varchar	1	<i>Primary key</i>
City	Varchar	9	



### 3.2 ER Diagram

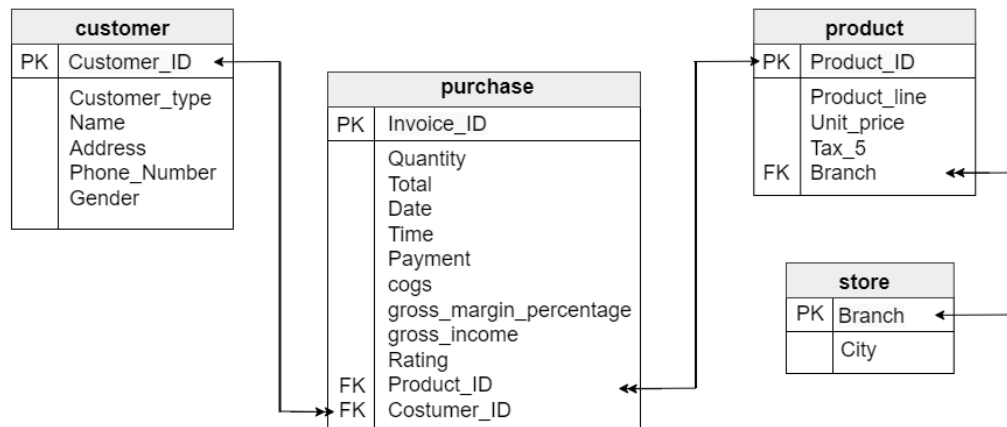
Basis data belanja *online* menyimpan rincian tentang pelanggan, produk yang dipesan oleh pelanggan, alamat pelanggan, toko yang menjual produk, dan waktu pembelian. Berikut adalah informasi tambahan yang diakomodasi oleh basis data sistem belanja *online* ini:

1. Suatu *e-commerce* menyediakan berbagai macam produk
2. Seorang pelanggan bisa memesan satu atau lebih produk
3. Semua produk dijual oleh semua toko
4. Produk dapat dipesan dalam satu atau lebih pembelian
5. Pelanggan memiliki nama, alamat, jenis kelamin, nomor telepon, tipe pelanggan, dan id pelanggan
6. Produk memiliki lini produk, harga satuan, id produk, dan pajak
7. Toko memiliki cabang dan kota
8. Pembelian memiliki jumlah, total harga, jenis pembayaran, invoice id, pendapatan kotor, margin laba kotor, waktu pembelian, tanggal pembelian, rating, dan harga pokok penjualan



### 3.3 Relasi Antar Tabel

Relasi yang terjadi antara tabel ditunjukkan seperti gambar berikut:



### 3.4 Simulasi Data

Dalam pembuatan database *e-commerce* berikut, penulis menggunakan dua dataset publik yang kemudian digabung menjadi satu dataset yang nantinya di input ke dalam sebuah database. Berikut adalah kolom-kolom yang nantinya akan menjadi *field* pada database.

Fitur	Deskripsi
Product ID	ID (unik) untuk setiap produk
Customer ID	ID (unik) untuk setiap customer yang terdaftar pada <i>e-commerce</i> tersebut
Phone Number	Nomor telpon <i>customer</i>
Address	Alamat tempat tinggal <i>customer</i>
Gender	Jenis kelamin <i>customer</i>
Name	Nama <i>customer</i>
Invoice ID	ID (unik) barang/pesanan <i>customer</i>
Branch	Toko penyedia barang barang (ada A, B, dan C)
City	Lokasi dari <i>Branch</i>
Customer Type	Jenis <i>customer</i> (member atau nonmember)
Product line	Kategori umum produk ( <i>Electronic accessories</i> , <i>Fashion accessories</i> , <i>Food</i> )

	<i>and beverages, Health and beauty, Home and lifestyle, Sports and travel)</i>
Unit Price	Harga barang per satuan
Quantity	Jumlah produk yang dibeli <i>customer</i>
Tax 5%	Pajak belanjaan sebesar 5%
Total	Total harga belanjaan sudah termasuk pajak
Date	Tanggal pembelian
Time	Waktu pembelian
Payment	Jenis pembayaran yang dilakukan oleh customer ( <i>Cash, Credit card dan E-wallet</i> )
Cogs	Harga pokok penjualan
Gross Margin Percentage	persentase margin kotor
Gross Income	Pendapatan kotor
Rating	Rating yang diberikan oleh customer

Dari fitur-fitur di atas yang tergabung menjadi satu dataset, kemudian akan di *slicing* atau dipisah-pisah untuk membentuk 4 tabel, yaitu tabel *customer*, tabel *product*, tabel *purchase*, dan tabel *store*.

```
customer = df[['CustomerID', 'Customer type',
              'Phone Number', 'Name', 'Address', 'gender']]
product = df[['ProductID', 'Product line', 'Unit price',
              'Tax 5%', 'Branch']]
purchase = df[['Invoice ID', 'ProductID', 'CustomerID', 'Name',
              'Address', 'Quantity', 'Total', 'Date', 'Time', 'Payment',
              'cogs', 'gross margin percentage',
              'gross income', 'Rating']]
store = df[['Branch', 'City']]
```

Seperti yang terlihat pada gambar di atas, untuk tabel *customer* berisi data-data dari fitur-fitur, seperti *Customer ID*, *Customer Type*, *Phone Number*, *Name*, *Address*, dan *Gender*. Kemudian, untuk tabel *product* berisi data-data dari fitur-fitur, seperti *Product ID*, *Product line*, *Unit Price*, *Tax 5%*, dan *Branch*. Selanjutnya, untuk tabel *purchase* berisi data-data dari fitur-fitur, seperti *Invoice ID*, *Product ID*, *Customer ID*, *Name*, *Address*, *Quantity*, *Total*, *Date*, *Time*, *Payment*, *COGS*, *Gross Margin*

*Percentage*, *Gross Income*, dan *Rating*. Terakhir, untuk tabel *store* hanya berisi data-data dari fitur-fitur, seperti *Branch* dan *City*.

Semua data-data yang disimpan kedalam empat tabel tersebut masih dalam bentuk DataFrame. Sebelum keempat tabel beserta data-datanya di input kedalam database menggunakan modul *sqlite3*, data-data tersebut akan diubah menjadi bentuk *array* atau *list* seperti berikut ini.

customer.head()

	CustomerID	Customer type	Phone Number	Name	Address	gender
0	43-107-331-695	Member	81573975928	Hafizhan Shidqi	777 Brockton Avenue	m
1	75-364-901-710	Normal	81573918390	Gandhi Wibowo	30 Memorial Drive	m
2	93-888-168-812	Normal	81573860852	Aldio Mahendra Purwandarto	250 Hartford Avenue	m
3	56-981-748-179	Member	81573803314	Benny Putra	700 Oak Street	m
4	99-154-337-194	Normal	81573745776	Vicky Vernando Dasta	66-4 Parkhurst Rd	m

Sebagai contoh, dapat terlihat bahwa gambar di atas adalah data-data pada tabel *customer* yang masih dalam bentuk DataFrame. Setelah diubah bentuknya menjadi bentuk *array*, maka akan seperti berikut:

```
customer_array = np.asarray(customer)
print(customer_array)

[['43-107-331-695' 'Member' 81573975928 'Hafizhan Shidqi'
 '777 Brockton Avenue' 'm']
 ['75-364-901-710' 'Normal' 81573918390 'Gandhi Wibowo'
 '30 Memorial Drive' 'm']
 ['93-888-168-812' 'Normal' 81573860852 'Aldio Mahendra Purwandarto'
 '250 Hartford Avenue' 'm']
 ...
 ['90-552-464-712' 'Member' 81516610542 'Diky Handriansyah'
 '454 Highland Ave' 'm']
 ['75-641-975-364' 'Normal' 81516553004 'Giono' '1184 Fall River Avenue'
 'm']
 ['11-836-689-420' 'Member' 81512361845 'Suryadi' '420 Greengrass Road'
 'm']]
```

Hal yang sama atau metode yang sama diterapkan juga ke tabel *product*, tabel *purchase*, dan tabel *store* sehingga data-data pada tabel tersebut berbentuk *array* atau *list*.

```
product_array = np.asarray(product)
purchase_array = np.asarray(purchase)
store_array = np.asarray(store)
```

Setelah semua data-data sudah siap, kemudian akan masuk ke dalam tahap penginputan ke dalam database menggunakan python dengan module sqlite3. Pertama, *import* module yang akan digunakan untuk pembuatan database, di sini akan digunakan module sqlite3 yang sudah ada di python.

```
import sqlite3
```

Kedua, akan digunakan function connect() yang terdapat pada module sqlite3 yang digunakan untuk koneksi file database SQLite yang sudah dibuat sebelumnya.

```
MyDB = sqlite3.connect('e-commerce db.db')
```

Terlihat pada gambar di atas bahwa didefinisikan sebuah variable MyDB dimana terkoneksi dengan SQLite 'e-commerce db.db'. Kemudian, dibuat objek cursor() untuk menjalankan perintah SQL sebagai berikut.

```
c = MyDB.cursor()
```

Kemudian akan digunakan syntax c.execute() untuk mengeksekusi perintah SQL pada cursor di atas.

```
c.execute('''CREATE TABLE customer(  
    "Customer_ID" VARCHAR(14) NOT NULL PRIMARY KEY UNIQUE,  
    "Customer_type" VARCHAR(6) NOT NULL,  
    "Phone_Number" INTEGER NOT NULL,  
    "Name" VARCHAR(28) NOT NULL,  
    "Address" VARCHAR(30) NOT NULL,  
    "Gender" VARCHAR(1) NOT NULL  
)''')
```

Terlihat bahwa dilakukan perintah untuk membuat tabel *customer* yang *field*-nya sudah disesuaikan dengan data pada *array* yang sudah dipersiapkan sebelumnya, dengan 'Customer\_ID' sebagai 'primary key'-nya. Kemudian, dilakukan penginputan data-data kedalam masing-masing *field* yang ada pada tabel *customer* dengan c.executemany() untuk menginput banyak data yang bentuknya *list* atau *array*.

```
[ ] c.executemany("INSERT INTO customer(Customer_ID,Customer_type,Phone_Number,Name,Address,Gender) VALUES(?,?,?,?,?,?)",customer_array)
```

Metode yang sama juga diterapkan untuk membuat tabel *product*, tabel *purchase*, dan tabel *store* yang kemudian juga dilakukan penginputan data-data ke dalam setiap *field* yang berada pada tabel tersebut.

```
c.execute('''CREATE TABLE product(
    "Product_ID" VARCHAR(12) NOT NULL PRIMARY KEY UNIQUE,
    "Product_line" VARCHAR(22) NOT NULL,
    "Unit_price" NUMERIC(5, 2) NOT NULL,
    "Tax_5" NUMERIC(7, 4) NOT NULL,
    "Branch" VARCHAR(1) NOT NULL,
    FOREIGN KEY (Branch) REFERENCES store(Branch)''')

c.executemany("INSERT INTO product(Product_ID,Product_line,Unit_price,Tax_5,Branch) VALUES(?,?,?,?,?)",product_array)
```

Pada tabel *product*, 'Product\_ID' dinyatakan sebagai 'primary key' dan 'Branch' sebagai 'foreign key' terhadap tabel *store*.

```
c.execute('''CREATE TABLE purchase(
    "Invoice_ID" VARCHAR(11) NOT NULL PRIMARY KEY UNIQUE,
    "Product_ID" VARCHAR(12) NOT NULL UNIQUE,
    "Customer_ID" VARCHAR(14) NOT NULL UNIQUE,
    "Name" VARCHAR(28) NOT NULL,
    "Address" VARCHAR(30) NOT NULL,
    "Quantity" INTEGER NOT NULL,
    "Total" NUMERIC(8, 4) NOT NULL,
    "Date" DATE NOT NULL,
    "Time" VARCHAR(5) NOT NULL,
    "Payment" VARCHAR(11) NOT NULL,
    "cogs" NUMERIC(6, 2) NOT NULL,
    "gross_margin_percentage" NUMERIC(11, 9) NOT NULL,
    "gross_income" NUMERIC(7, 4) NOT NULL,
    "Rating" NUMERIC(4, 1) NOT NULL,
    FOREIGN KEY (Product_ID) REFERENCES product(Product_ID),
    FOREIGN KEY (Customer_ID) REFERENCES customer(Customer_ID),
    FOREIGN KEY (Name) REFERENCES customer(Name),
    FOREIGN KEY (Address) REFERENCES customer(Address)''')

c.executemany("INSERT INTO purchase(Invoice_ID,Product_ID,Customer_ID,Name,Address,Quantity,Total,Date,Time,Payment,cogs,gross_margin_percenti
```

Pada tabel *purchase*, 'Invoice\_ID' dinyatakan sebagai 'primary key'. Kemudian untuk 'foreign key'-nya adalah 'Product\_ID' yang terdapat pada tabel *product*, 'Customer\_ID' yang terdapat pada tabel *customer*, 'Name' yang terdapat pada tabel *customer*, dan 'Address' yang terdapat pada tabel *customer*.

```
c.execute('''CREATE TABLE IF NOT EXISTS store(
    "Branch" VARCHAR(1) NOT NULL PRIMARY KEY,
    "City" VARCHAR(9) NOT NULL)''')

c.executemany("INSERT OR IGNORE INTO store(Branch, City) VALUES(?,?)",store_array)
```

Terakhir, pada tabel *store*, 'Branch' dinyatakan sebagai 'primary key'-nya dan tidak ada 'foreign key'.

Setelah proses pembuatan dan penginputan tabel, maka akan dijalankan syntax `commit()` yang bertujuan untuk menyimpan semua perubahan yang terjadi pada database setelah proses pembuatan dan penginputan tabel sehingga data-data yang sudah dimasukkan sebelumnya sudah tersimpan secara terstruktur di dalam database yang dibuat.

```
MyDB.commit()
```

Terakhir, perintah `close()` digunakan untuk menutup koneksi ke database yang sudah kita hubungkan di awal tadi.

```
MyDB.close()
```

Maka, simulasi pembuatan dan penginputan tabel beserta data-datanya menggunakan python dan module `sqlite3` telah selesai. Berikut ini adalah tabel hasil dari penginputan pada colab jika dibuka menggunakan *software* `sqlite`.

### 1. Tabel *customer*

Tabel ini berisi mengenai informasi pelanggan.

```
1 CREATE TABLE "customer" (
2     "Customer_ID"    VARCHAR(14) NOT NULL UNIQUE,
3     "Customer_type"  VARCHAR(6) NOT NULL,
4     "Phone_Number"   INTEGER NOT NULL,
5     "Name"           VARCHAR(28) NOT NULL,
6     "Address"        VARCHAR(30) NOT NULL,
7     "Gender"         VARCHAR(1) NOT NULL,
8     PRIMARY KEY ("Customer_ID")
9 );
```

	Customer_ID	Customer_type	Phone_Number	Name	Address	Gender
	Filter	Filter	Filter	Filter	Filter	Filter
1	36-914-663-853	Member	81573975928	Hafizhan Shidqi	777 Brockton Avenue	m
2	98-743-311-348	Normal	81573918390	Gandhi Wibowo	30 Memorial Drive	m
3	31-539-950-812	Normal	81573860852	Aldio Mahendra Purwandarto	250 Hartford Avenue	m
4	24-936-876-674	Member	81573803314	Benny Putra	700 Oak Street	m
5	97-682-694-598	Normal	81573745776	Vicky Vernando Dasta	66-4 Parkhurst Rd	m
6	98-969-134-387	Normal	81573688238	Jufianto Henri	591 Memorial Dr	m
7	20-436-170-828	Member	81573630700	Aan Nuraini	55 Brooksby Village Way	f
8	62-604-252-592	Normal	81573573162	Abdur Rahman	137 Teaticket Hwy	m
9	93-207-533-545	Member	81573515624	Abdurrahman	42 Fairhaven Commons Way	m
10	94-113-381-125	Member	81573458086	Ade Indra Sukma	374 William S Canning Blvd	f
11	57-523-401-922	Member	81573400548	Ade Irmayani	121 Worcester Rd	f
12	15-698-323-659	Member	81573343010	Bakti Yoga Fiyandana	677 Timpany Blvd	m
13	23-944-458-107	Normal	81573285472	Daniel Septra Fatama	337 Russell St	m
14	14-533-979-353	Normal	81573227934	Dayu M Sandro	295 Plymouth Street	m
15	74-653-291-198	Normal	81573170396	Dean Mareti Hariani	1775 Washington St	f
16	94-927-440-201	Member	81573112858	Edi Kurniawan Wibowo	280 Washington Street	m
17	69-729-427-715	Member	81573055320	Fadil Rahmat Andini	20 Soojian Dr	m
18	84-675-415-538	Normal	81572997782	Fahmi Iqbal Firmananda	11 Jungle Road	m
19	60-579-728-746	Normal	81572940244	Fairuzi	301 Massachusetts Ave	m
20	72-152-610-148	Normal	81572882706	Gustian	780 Lynnway	m
21	72-547-568-657	Member	81572825168	Habil Sabilla Do'A	70 Pleasant Valley Street	f
22	31-767-805-322	Normal	81572767630	Hermawan Syah	830 Curran Memorial Hwy	m
23	35-806-373-542	Normal	81572710092	Ibnuyohanzah Ahmad	1470 S Washington St	m
24	72-152-352-264	Normal	81572652554	Lia Partiwil	506 State Road	f

## 2. Tabel *product*

Tabel ini berisi mengenai informasi produk yang dibeli oleh pelanggan.

```

1 CREATE TABLE "product" (
2     "Product_ID"    VARCHAR(12) NOT NULL UNIQUE,
3     "Product_line"  VARCHAR(22) NOT NULL,
4     "Unit_price"    NUMERIC(5, 2) NOT NULL,
5     "Tax_5"         NUMERIC(7, 4) NOT NULL,
6     "Branch"        VARCHAR(1) NOT NULL,
7     PRIMARY KEY ("Product_ID"),
8     FOREIGN KEY ("Branch") REFERENCES "store" ("Branch")
9 );

```

	Product_ID	Product_line	Unit_price	Tax_5	Branch
	Filter	Filter	Filter	Filter	Filter
1	3724-221-795	Health and beauty	74.69	26.1415	A
2	3917-143-388	Electronic accessories	15.28	3.82	C
3	6370-830-875	Home and lifestyle	46.33	16.2155	A
4	6750-423-981	Health and beauty	58.22	23.288	A
5	6495-125-931	Sports and travel	86.31	30.2085	A
6	7487-804-690	Electronic accessories	85.39	29.8865	C
7	5715-261-677	Electronic accessories	68.84	20.652	A
8	6757-472-230	Home and lifestyle	73.56	36.78	C
9	3462-233-137	Health and beauty	36.26	3.626	A
10	8505-644-260	Food and beverages	54.84	8.226	B
11	1248-643-303	Fashion accessories	14.48	2.896	B
12	4140-319-266	Electronic accessories	25.51	5.102	B
13	4511-605-327	Electronic accessories	46.95	11.7375	A
14	7804-804-101	Food and beverages	43.19	21.595	A
15	8438-252-976	Health and beauty	71.38	35.69	A
16	6492-757-868	Sports and travel	93.72	28.116	B
17	1135-907-813	Health and beauty	68.93	24.1255	A
18	8486-882-410	Sports and travel	72.61	21.783	A
19	6609-887-659	Food and beverages	54.67	8.2005	A
20	2297-800-485	Home and lifestyle	40.3	4.03	B
21	3517-533-936	Electronic accessories	86.04	21.51	C
22	5934-587-506	Health and beauty	87.98	13.197	B
23	1665-756-738	Home and lifestyle	33.2	3.32	B
24	3416-802-735	Electronic accessories	34.56	8.64	A



### 3. Tabel *purchase*

Tabel ini berisi mengenai informasi pembelian pelanggan.

```

1 CREATE TABLE "purchase" (
2     "Invoice_ID"    VARCHAR(11) NOT NULL UNIQUE,
3     "Product_ID"    VARCHAR(12) NOT NULL UNIQUE,
4     "Customer_ID"   VARCHAR(14) NOT NULL UNIQUE,
5     "Name"          VARCHAR(28) NOT NULL,
6     "Address"       VARCHAR(30) NOT NULL,
7     "Quantity"      INTEGER NOT NULL,
8     "Total"         NUMERIC(8, 4) NOT NULL,
9     "Date"          DATE NOT NULL,
10    "Time"           VARCHAR(5) NOT NULL,
11    "Payment"        VARCHAR(11) NOT NULL,
12    "cogs"           NUMERIC(6, 2) NOT NULL,
13    "gross_margin_percentage" NUMERIC(11, 9) NOT NULL,
14    "gross_income"   NUMERIC(7, 4) NOT NULL,
15    "Rating"         NUMERIC(4, 1) NOT NULL,
16    PRIMARY KEY("Invoice_ID"),
17    FOREIGN KEY("Product_ID") REFERENCES "product"("Product_ID"),
18    FOREIGN KEY("Customer_ID") REFERENCES "customer"("Customer_ID"),
19    FOREIGN KEY("Address") REFERENCES "customer"("Address"),
20    FOREIGN KEY("Name") REFERENCES "customer"("Name")
21 );

```

	Invoice_ID	Product_ID	Customer_ID	Name	Address	Quantity	Total	Date	Time	Payment	cogs	_margin_perce	gross_Inco
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
2	226-31-3081	3917-143-388	98-743-311-348	Gandhi Wibowo	30 Memorial Drive	5	80.22	3/8/2019	10:29	Cash	76.4	4.761904762	3.82
3	631-41-3108	6370-830-875	31-539-950-812	Aldio Mahendra Purwandarto	250 Hartford Avenue	7	340.5255	3/3/2019	13:23	Credit card	324.31	4.761904762	16.2155
4	123-19-1176	6750-423-981	24-936-876-674	Benny Putra	700 Oak Street	8	489.048	1/27/2019	20:33	Ewallet	465.76	4.761904762	23.288
5	373-73-7910	6495-125-931	97-682-694-598	Vicky Vernando Dasta	66-4 Parkhurst Rd	7	634.3785	2/8/2019	10:37	Ewallet	604.17	4.761904762	30.2085
6	699-14-3026	7487-804-690	98-969-134-387	Jufianto Henri	591 Memorial Dr	7	627.6165	3/25/2019	18:30	Ewallet	597.73	4.761904762	29.8865
7	355-53-5943	5715-261-677	20-436-170-828	Aan Nuraini	55 Brooksby Village Way	6	433.692	2/25/2019	14:36	Ewallet	413.04	4.761904762	20.652
8	315-22-5665	6757-472-230	62-604-252-592	Abdur Rahman	137 Teaticket Hwy	10	772.38	2/24/2019	11:38	Ewallet	735.6	4.761904762	36.78
9	665-32-9167	3462-233-137	93-207-533-545	Abdurrahman	42 Fairhaven Commons Way	2	76.146	1/10/2019	17:15	Credit card	72.52	4.761904762	3.626
10	692-92-5582	8505-644-260	94-113-381-125	Ade Indra Sukma	374 William S Canning Blvd	3	172.746	2/20/2019	13:27	Credit card	164.52	4.761904762	8.226
11	351-62-0822	1248-643-303	57-523-401-922	Ade Irmayani	121 Worcester Rd	4	60.816	2/6/2019	18:07	Ewallet	57.92	4.761904762	2.896
12	529-56-3974	4140-319-266	15-698-323-659	Bakti Yoga Flyandana	677 Timpany Blvd	4	107.142	3/9/2019	17:03	Cash	102.04	4.761904762	5.102
13	365-64-0515	4511-605-327	23-944-458-107	Daniel Septra Fatama	337 Russell St	5	246.4875	2/12/2019	10:25	Ewallet	234.75	4.761904762	11.7375
14	252-56-2699	7804-804-101	14-533-979-353	Dayu M Sandro	295 Plymouth Street	10	453.495	2/7/2019	16:48	Ewallet	431.9	4.761904762	21.595
15	829-34-3910	8438-252-976	74-653-291-198	Dean Maret Hariani	1775 Washington St	10	749.49	3/29/2019	19:21	Cash	713.8	4.761904762	35.69
16	299-46-1805	6492-757-868	94-927-440-201	Edi Kurniawan Wibowo	280 Washington Street	6	590.436	1/15/2019	16:19	Cash	562.32	4.761904762	28.116
17	656-95-9349	1135-907-813	69-729-427-715	Fadil Rahmat Andini	20 Soojian Dr	7	506.6355	3/11/2019	11:03	Credit card	482.51	4.761904762	24.1255
18	765-26-6951	8486-882-410	84-675-415-538	Fahmi Iqbal Firmananda	11 Jungle Road	6	457.443	1/1/2019	10:39	Credit card	435.66	4.761904762	21.783
19	329-62-1586	6609-887-659	60-579-728-746	Fairuzi	301 Massachusetts Ave	3	172.2105	1/21/2019	18:00	Credit card	164.01	4.761904762	8.2005
20	319-50-3348	2297-800-485	72-152-610-148	Gustian	780 Lynnway	2	84.63	3/11/2019	15:30	Ewallet	80.6	4.761904762	4.03
21	300-71-4605	3517-533-936	72-547-568-657	Habil Sabilla Do'A	70 Pleasant Valley Street	5	451.71	2/25/2019	11:24	Ewallet	430.2	4.761904762	21.51
22	371-85-5789	5934-587-506	31-767-805-322	Hermawan Syah	830 Curran Memorial Hwy	3	277.137	3/5/2019	10:40	Ewallet	263.94	4.761904762	13.197
23	273-16-6619	1665-756-738	35-806-373-542	Ibnuyohanzah Ahmad	1470 S Washington St	2	69.72	3/15/2019	12:20	Credit card	66.4	4.761904762	3.32
24	636-48-8204	3416-892-735	72-157-357-264	Lia Pertiwi	506 State Road	5	181.44	2/17/2019	11:15	Ewallet	172.8	4.761904762	8.64

#### 4. Tabel *store*

Tabel ini berisi mengenai informasi toko yang menjual *product*.

```
1 CREATE TABLE "store" (  
2     "Branch"    VARCHAR(1) NOT NULL,  
3     "City"     VARCHAR(9) NOT NULL,  
4     PRIMARY KEY ("Branch")  
5 );
```

	Branch	City
	Filter	Filter
1	A	Yangon
2	C	Naypyitaw
3	B	Mandalay

### 3.5 Analisis Data

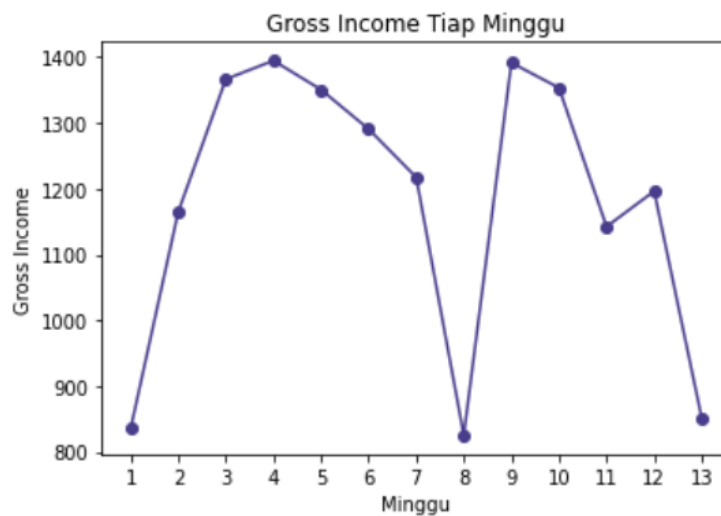
Setelah membuat database, kami melakukan analisis dan visualisasi data sederhana dengan Python. Kami mencoba menelusuri hubungan pendapatan kotor atau *gross income* dengan variabel-variabel lain, yaitu waktu, cabang, lini produk, dan *membership*.

#### 1. Pendapatan Kotor per Minggu

Berikut adalah grafik total pendapatan kotor pada tiap minggu dalam Januari sampai Maret 2019. Atribut *gross income* diambil dari entitas *purchase*. Sedangkan data per minggu didapatkan dari atribut *date* dari entitas *purchase* yang berisi tanggal pembelian, lalu dikelompokkan per minggu. Dapat dilihat bahwa dari awal Januari serta akhir Februari dan Maret, pendapatan kotor relatif rendah. Sedangkan, pada pertengahan tiap bulan pendapatan relatif lebih tinggi.

```
df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
weekly_income = np.asarray(df.groupby(
    df.Date.dt.isocalendar().week)['gross income'].sum())

weeks = [1,2,3,4,5,6,7,8,9,10,11,12,13]
plt.title("Gross Income Tiap Minggu")
plt.xlabel("Minggu")
plt.ylabel("Gross Income")
plt.xticks(weeks)
plt.plot(weeks,weekly_income, color="#483D8B", marker='o')
```



## 2. Pendapatan Kotor per Cabang

Pendapatan kotor per cabang diperoleh dengan melihat *foreign key branch* untuk setiap pembelian dalam tabel *purchase*, lalu menjumlahkan *gross income*-nya untuk setiap cabang. Pada grafik di bawah, label yang digunakan adalah nama kota lokasi tiap cabang. Dapat dilihat bahwa pendapatan dari cabang di Kota Yangon dan Naypyitaw relatif sama, sedangkan pendapatan dari cabang Kota Mandalay sedikit lebih besar.

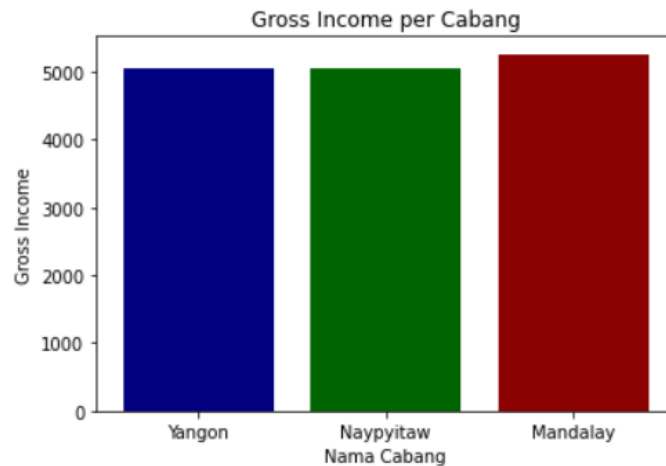
```
cabang = np.asarray(df[['Branch']])
np.unique(cabang)

df_A = df[df['Branch'] == 'A']
df_B = df[df['Branch'] == 'B']
df_C = df[df['Branch'] == 'C']

gross_A = df_A['gross income'].sum()
gross_B = df_B['gross income'].sum()
gross_C = df_C['gross income'].sum()
```

```
plt.title("Gross Income per Cabang")
plt.xlabel("Nama Cabang")
plt.ylabel("Gross Income")
plt.bar(["Yangon", "Naypyitaw", "Mandalay"], [gross_A, gross_B, gross_C],
color=["#000080", "#006400", "#8B0000"])

print(f'Gross Income dari cabang Yangon: {gross_A}')
print(f'Gross Income dari cabang Mandalay: {gross_B}')
print(f'Gross Income dari cabang Naypyitaw: {gross_C}')
```



### 3. Pendapatan Kotor setiap Lini Produk

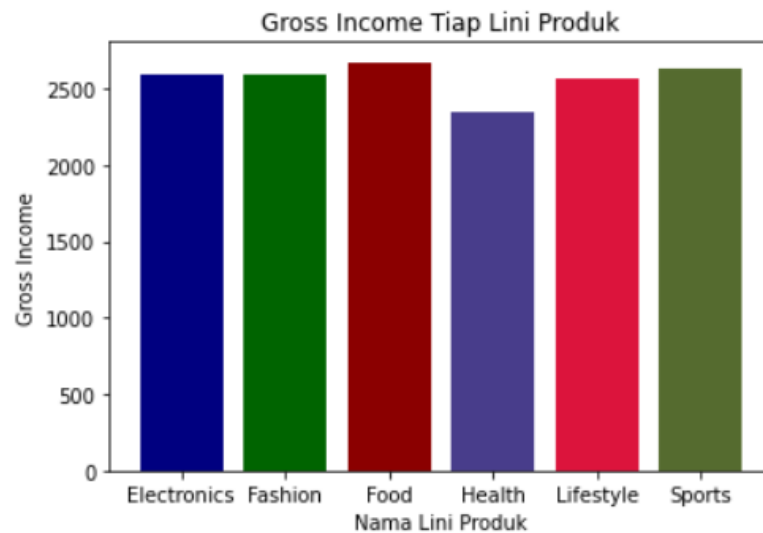
Pendapatan kotor untuk tiap lini produk diperoleh dengan melihat atribut *product line* dari tabel produk untuk setiap *product id* yang berkorespondensi dengan entri-entri pembelian. Dari tabel di bawah, terlihat bahwa *product line* yang membawa keuntungan paling besar adalah *food and drinks*, dan yang keuntungannya paling kecil adalah *health and beauty*.

```
product_line = np.asarray(df[['Product line']])
np.unique(product_line)

df_EA = df[df['Product line'] == 'Electronic accessories']
df_FA = df[df['Product line'] == 'Fashion accessories']
df_FB = df[df['Product line'] == 'Food and beverages']
df_HB = df[df['Product line'] == 'Health and beauty']
df_HL = df[df['Product line'] == 'Home and lifestyle']
df_ST = df[df['Product line'] == 'Sports and travel']

gross_EA = df_EA['gross income'].sum()
gross_FA = df_FA['gross income'].sum()
gross_FB = df_FB['gross income'].sum()
gross_HB = df_HB['gross income'].sum()
gross_HL = df_HL['gross income'].sum()
gross_ST = df_ST['gross income'].sum()
```

```
plt.title("Gross Income Tiap Lini Produk")
plt.xlabel("Nama Lini Produk")
plt.ylabel("Gross Income")
plt.bar(["Electronics", "Fashion", "Food", "Health", "Lifestyle",
        "Sports"],
        [gross_EA, gross_FA, gross_FB, gross_HB, gross_HL, gross_ST],
        color=["#000080", "#006400", "#8B0000", "#483D8B", "#DC143C",
        "#556B2F"])
```



#### 4. Pendapatan Kotor dari Customer Member dan Nonmember

Tabel ini diperoleh dengan membandingkan pendapatan dari pelanggan yang memiliki *membership* dan tidak. Informasi *membership* diambil dari atribut *customer type* di tabel *customer*, dan *gross income* diambil dari atribut *gross income* di tabel *purchase*. *Membership* adalah informasi keanggotaan seorang pelanggan dimana pelanggan yang menjadi member akan memperoleh bonus seperti diskon untuk setiap pembelian yang dilakukan. Pada tabel di bawah, terlihat bahwa rata-rata *customer member* memberikan lebih sedikit *gross income* daripada *customer non-member*.

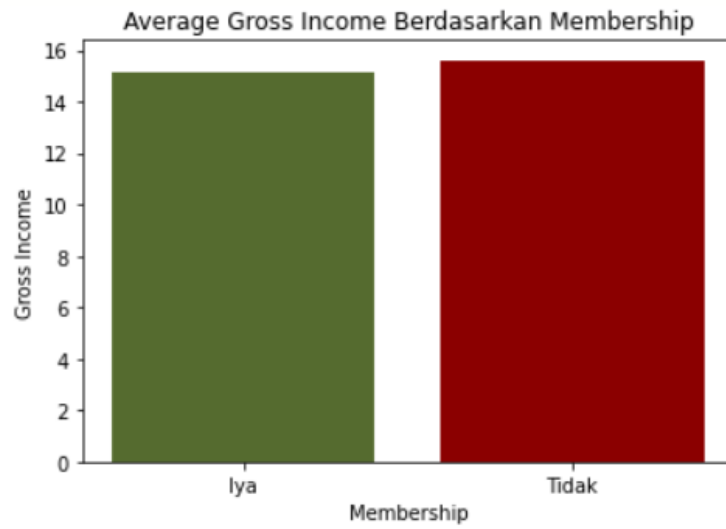
```
df_normal = df[df['Customer type'] == 'Normal']
df_member = df[df['Customer type'] == 'Member']
df_normal.head()

df_member

avg_gross_normal = df_normal['gross income'].sum()/len(df_normal.index)
avg_gross_member = df_member['gross income'].sum()/len(df_member.index)

print(f'Average Gross Income dari customer normal: {avg_gross_normal}')
print(f'Average Gross Income dari customer member: {avg_gross_member}')
```

```
plt.title("Average Gross Income Berdasarkan Membership")
plt.xlabel("Membership")
plt.ylabel("Gross Income")
plt.bar(["Iya", "Tidak"], [avg_gross_normal, avg_gross_member],
color=["#556B2F", "#8B0000"])
```



## **BAB IV**

### **PENUTUP**

#### **4.1 Kesimpulan**

Berdasarkan simulasi yang telah dilakukan sebelumnya maka database yang telah dibuat sebagai penyimpan informasi mengenai data pembelian, data produk yang dibeli, data pembeli, dan data toko yang menjual telah berhasil direalisasikan. Database ini juga dapat membantu perusahaan *e-commerce* dalam melakukan kegiatan operasional, di antaranya adalah memudahkan tim data di perusahaan *e-commerce* dalam melakukan pengelolaan data dengan cepat dan akurat, memberikan kemudahan dalam melakukan pelayanan yang baik dan efisien kepada pembeli, dan memudahkan tim data dalam melakukan proses pencatatan dan pencarian data transaksi dan administrasi. Kemudian, dengan melakukan analisis data sederhana dari database yang telah dibuat didapatkan *insight* mengenai berbagai tren pembelian dan perusahaan dapat memanfaatkannya untuk mengambil keputusan dan melakukan perencanaan perusahaan di masa depan.

## DAFTAR PUSTAKA

- Hermansyah, Dionisius Darryl. 2020. *Indonesian Names*.  
<https://www.kaggle.com/dionisiusdh/indonesian-names>.
- Rahmawaty, Fika. 2009. “Aplikasi Toko Online Berbasis PHP dan MySQL pada Toko Mulya Hadi Craft Pekalongan”. <https://123dok.com/document/q2n4p9pq-fika-rahmawaty.html>.  
Diakses terakhir pada 8 Juni 2022.
- Pyae, Aung. 2019. *Supermarket Sales*.  
<https://www.kaggle.com/aungpyaeap/supermarket-sales>.
- Sakib, Ahmed Shariar. 2022. *List of Real USA Addresses*.  
<https://www.kaggle.com/datasets/ahmedshahriarsakib/list-of-real-usa-addresses>.