

PEMROGRAMAN

BERORIENTASI

OBJEK

Pertemuan 12

**PEMROGRAMAN
DATABASE**

KONSEP DATABASE

- » Database atau basis data adalah suatu susunan/kumpulan data operasional lengkap dari suatu organisasi/perusahaan yang diorganisir/dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi yang diperlukan pemakainya.
- » Sistem basis data adalah kumpulan dari program aplikasi yang berinteraksi dengan basis data bersama dengan Database Management System (DBMS) dan basis data itu sendiri.
- » Database Management System (DBMS) merupakan perangkat lunak yang bertugas untuk memanggil query sesuai dengan kebutuhan pemakai.

Database memiliki fungsi sebagai berikut:

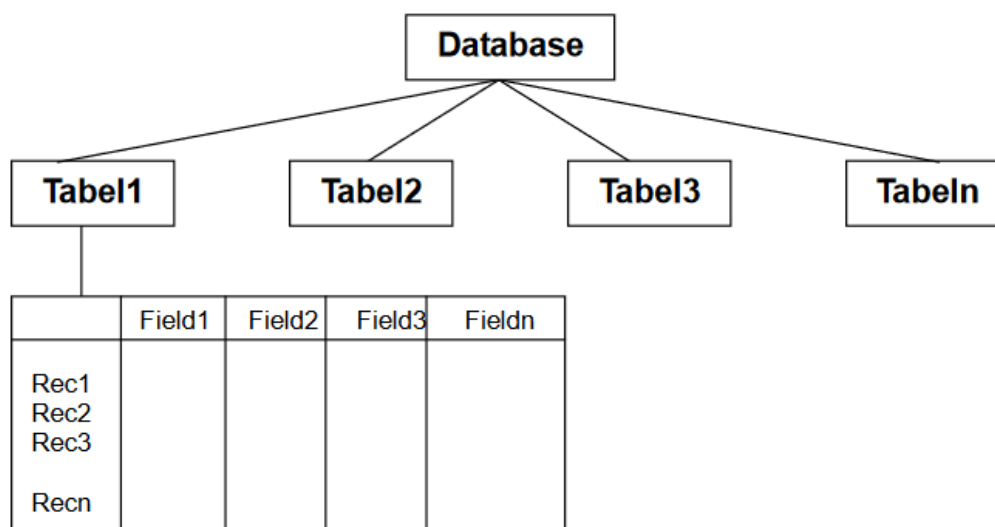
- ★ Mengelompokkan data dan informasi agar mempermudah dalam proses identifikasi data.
- ★ Mencegah inkonsistensi data dan duplikasi data.
- ★ Menyederhanakan proses menyimpan, mengakses, memperbarui, dan menghapus data.
- ★ Menjaga kualitas data dan informasi yang dimasukkan.
- ★ Membantu proses penyimpanan data dengan kapasitas besar.
- ★ Meningkatkan kinerja aplikasi/*software* yang membutuhkan penyimpanan data.

Komponen database

Komponen terbesar dari database adalah database itu sendiri.

- » Data dalam database dapat dikelompokkan kedalam beberapa tabel
- » Tabel merupakan kumpulan data sejenis
- » Tabel terdiri dari beberapa field (kolom) dan record (baris)
- » Field menerangkan sebuah informasi tentang identitas data dalam tabel, sedangkan record menerangkan sebuah data dalam tabel secara lengkap.

Berikut hirarki database:



Perangkat Lunak DBMS

Berikut beberapa perangkat lunak DBMS yang akan digunakan untuk pemrograman database:

1. SQLite

Adalah sebuah engine database SQL yang langsung tertanam pada aplikasi. SQLite tidak memiliki server yang terpisah dari aplikasi sehingga untuk membaca dan menulis dapat dilakukan langsung tanpa konfigurasi.

Berikut contoh program penggunaan SQLite di python:

Membuat database dan tabel di dalam SQLite

```
1  # Membuat database dan tabel dalam SQLite
2
3  import sqlite3
4
5  conn = sqlite3.connect("mydb")
6  cur = conn.cursor()
7  sql = '''
8      CREATE TABLE produk (
9          kode CHAR(4) NOT NULL PRIMARY KEY,
10         nama VARCHAR(25),
11         harga REAL)
12     '''
13  cur.execute(sql)
14  cur.close()
15  conn.close()
```

Menambah baris data di dalam SQLite

```
1 # Mengisi tabel produk dalam SQLite
2
3 import sqlite3
4
5 conn = sqlite3.connect("mydb")
6 cur = conn.cursor()
7
8 # Menambah 3 baris data
9 cur.execute("INSERT INTO produk VALUES ('P001','Pensil',6000)")
10 cur.execute("INSERT INTO produk VALUES('P002','Penghapus',9000)")
11 cur.execute("INSERT INTO produk VALUES('P003','Penggaris',12000)")
12
13 #Commit transaksi
14 conn.commit()
15
16 #Menutup objek cursor
17 cur.close()
18
19 #menutup objek koneksi
20 conn.close()
```

Membaca baris data di dalam SQLite

```
1 # Membaca baris data dalam SQLite
2
3 import sqlite3
4
5 conn = sqlite3.connect("mydb")
6 cur = conn.cursor()
7 result = cur.execute("SELECT * FROM produk")
8
9 # Membaca data
10 for row in result:
11     print('%s,%s,%.0f' % (row[0],row[1],row[2]))
12
13 cur.close()
14 conn.close()
```

Mengubah baris data di dalam SQLite

```
1  # Mengubah baris data dalam SQLite
2
3  import sqlite3
4
5  conn = sqlite3.connect("mydb")
6  cur = conn.cursor()
7
8  # Data sebelum diubah
9  for row in cur.execute("SELECT * FROM produk"):
10     print('%s,%s,%.0f' % (row[0],row[1],row[2]))
11
12  print("-----")
13
14  # Mengubah data
15  cur.execute('''UPDATE produk
16                SET harga = ?
17                WHERE kode = ?''', (3000,'P001'))
18  conn.commit()
19
20  # Data setelah diubah
21  for row in cur.execute("SELECT * FROM produk"):
22     print('%s,%s,%.0f' % (row[0],row[1],row[2]))
23
24  cur.close()
25  conn.close()
```

Menghapus baris data di dalam SQLite

```
1  # Menghapus baris dalam SQLite
2
3  import sqlite3
4
5  conn = sqlite3.connect("mydb")
6  cur = conn.cursor()
7
8  # Data sebelum dihapus
9  for row in cur.execute("SELECT * FROM produk"):
10     print('%s,%s,%.0f' % (row[0],row[1],row[2]))
11
12  print("-----")
13
14  # Menghapus data
15  cur.execute('''DELETE FROM produk
16                WHERE kode = ?''', ('P002',))
17  conn.commit()
18
19  # Data setelah dihapus
20  for row in cur.execute("SELECT * FROM produk"):
21     print('%s,%s,%.0f' % (row[0],row[1],row[2]))
22
23  cur.close()
24  conn.close()
```

Membuat dan mengeksekusi fungsi di dalam SQLite

```
1  # Membuat dan Mengeksekusi Fungsi dalam SQLite
2
3  # Fungsi pertama
4  def toupper(s):
5      return s.upper()
6
7  # Fungsi kedua
8  def tolower(s):
9      return s.lower()
10
11 import sqlite3
12
13 conn = sqlite3.connect("mydb")
14
15 # Memasukkan fungsi ke dalam database
16 conn.create_function("toupper",1,toupper)
17 conn.create_function("tolower",1,tolower)
18
19 # Menggunakan fungsi di dalam SQL
20 cur = conn.cursor()
21 result1 = cur.execute("SELECT toupper(nama) FROM produk")
22 for row in result1:
23     print(row[0])
24
25 print("-----")
26
27 result2 = cur.execute("SELECT tolower(nama) FROM produk")
28 for row in result2:
29     print(row[0])
30
31 cur.close()
32 conn.close()
```

Membuat trigger di dalam SQLite

```
1  # Membuat Trigger di dalam SQLite
2
3  import sqlite3
4
5  conn = sqlite3.connect("mydb")
6  cur = conn.cursor()
7
8  # Membuat tabel log
9  cur.execute('''CREATE TABLE log (
10                id INTEGER PRIMARY KEY,
11                harga_lama REAL,
12                harga_baru REAL,
13                waktu DATE
14            )
15            ''')
16
17  # Membuat trigger
18  cur.execute('''CREATE TRIGGER tr_harga
19                UPDATE OF harga ON produk
20                BEGIN
21                    INSERT INTO log(
22                        harga_lama,
23                        harga_baru,
24                        waktu)
25                    VALUES(
26                        old.harga,
27                        new.harga,
28                        datetime('now'));
29                END;
30            ''')
31
32  # Mengubah data harga pada tabel produk
33  cur.execute('''UPDATE produk SET harga = 7000
34                WHERE kode = 'P003'
35            ''')
36  conn.commit()
37
38  # Menampilkan data pada tabel log
39  result = cur.execute("SELECT * FROM log")
40  for row in result:
41      print('%d,%.0f,%.0f,%s' % (row[0],row[1],row[2],row[3]))
42
43  cur.close()
44  conn.close()
```

Melakukan ekspor data dari SQLite ke MS Excel

```
1 # Melakukan ekspor data dari SQLite ke MS Excel
2
3 import sqlite3, csv
4 conn = sqlite3.connect("mydb")
5 cur = conn.cursor()
6
7 with open("d:\\produk.csv", 'w', newline='', encoding='utf-8') as csvfile:
8     w = csv.writer(csvfile, delimiter=',')
9     for row in cur.execute("SELECT * FROM produk"):
10         w.writerow(row)
11
12 csvfile.close()
13 cur.close()
14 conn.close()
```

Melakukan impor data dari MS Excel ke SQLite

```
1 # Melakukan impor data dari MS Excel ke SQLite
2
3 import csv, sqlite3
4
5 data = [
6     ('P004', 'Tinta', 20000),
7     ('P005', 'Buku Tulis', 10000),
8     ('P006', 'Buku Gambar', 13000)
9 ]
10
11 with open("d:\\produk1.csv", 'w', newline='', encoding='utf-8') as csvfile:
12     w = csv.writer(csvfile, delimiter=',')
13     w.writerows(data)
14     csvfile.close()
15
16 conn = sqlite3.connect("mydb")
17 cur = conn.cursor()
18
19 with open("d:\\produk1.csv", 'r', newline='', encoding='utf-8') as csvfile:
20     r = csv.reader(csvfile, delimiter=',')
21     for row in r:
22         sql = "INSERT INTO produk VALUES(?,?,?)"
23         cur.execute(sql, (row[0], row[1], row[2]))
24
25 conn.commit()
26 csvfile.close()
27
28 for row in cur.execute("SELECT * FROM produk"):
29     print('%s,%s,%.0f' % (row[0], row[1], row[2]))
30
31 cur.close()
32 conn.close()
```


2. MySQL

Adalah sebuah DBMS yang menggunakan perintah SQL (*Structure Query Language*) yang terbagi menjadi dua lisensi yaitu free software dimana perangkat lunak dapat diakses oleh siapa saja, dan shareware dimana perangkat lunak berpelik yang memiliki batasan dalam penggunaannya.

MySQL termasuk kedalam RDBMS (*Relational Database Management System*) yang dalam proses pengambilan data menggunakan metode relasional database dan juga menjadi penghubung antara perangkat lunak dan database server.

Tidak seperti SQLite, paket distribusi python tidak dilengkapi dengan modul untuk mengakses database MySQL. Oleh karena itu modul MySQL harus terpasang secara terpisah dengan melakukan instalasi menggunakan program pip, dengan menuliskan perintah berikut ini:

```
pip install mysql.connector
```

Berikut contoh program penggunaan MySQL di python:

Buatlah database terlebih dahulu di dalam server MySQL dan beri nama **pydb**.

Membuat tabel di dalam MySQL

```
1 # Membuat Tabel di MYSQL
2
3 import mysql.connector
4
5 conn = mysql.connector.connect(
6     user="root",
7     password="",
8     host="localhost",
9     database="pydb"
10 )
11
12 cur = conn.cursor()
13 sql='''
14     CREATE TABLE produk (
15         kode CHAR(4) NOT NULL PRIMARY KEY,
16         nama VARCHAR(25),
17         harga DECIMAL)
18     '''
19 cur.execute(sql)
20 cur.close()
21 conn.close()
```

Menambah baris data di dalam MySQL

```
1  # Menambah Baris Data di dalam MYSQL
2
3  import mysql.connector
4
5  conn = mysql.connector.connect(
6      user="root",
7      password="",
8      host="localhost",
9      database="pydb"
10 )
11
12 cur = conn.cursor()
13
14 # menambah 3 baris data
15 cur.execute("INSERT INTO produk VALUES('P001','Pensil',6000)")
16 cur.execute("INSERT INTO produk VALUES('P002','Spidol',9000)")
17 cur.execute("INSERT INTO produk VALUES('P003','Penggaris',5000)")
18
19 conn.commit()
20 cur.close()
21 conn.close()
```

Membaca baris data di dalam MySQL

```
1  # Membaca Baris Data di dalam MYSQL
2
3  import mysql.connector
4
5  conn = mysql.connector.connect(
6      user="root",
7      password="",
8      host="localhost",
9      database="pydb"
10 )
11
12 cur = conn.cursor()
13
14 # membaca data
15 cur.execute("SELECT * FROM produk")
16 result = cur.fetchall()
17
18 for row in result:
19     print('%s,%s,%.0f' % (row[0],row[1],row[2]))
20
21 cur.close()
22 conn.close()
```

Mengubah baris data di dalam MySQL

```
1  # Mengubah Baris Data di dalam MYSQL
2
3  import mysql.connector
4
5  conn = mysql.connector.connect(
6      user="root",
7      password="",
8      host="localhost",
9      database="pydb"
10 )
11
12 cur = conn.cursor()
13
14 # data sebelum diubah
15 cur.execute("SELECT * FROM produk")
16 result = cur.fetchall()
17
18 for row in result:
19     print('%s,%s,%.0f' % (row[0],row[1],row[2]))
20 print("-----")
21
22 # mengubah data
23 cur.execute('''UPDATE produk SET harga = %s WHERE kode = %s''', (12000,'P002'))
24 conn.commit()
25
26 # data setelah diubah
27 cur.execute("SELECT * FROM produk")
28 result = cur.fetchall()
29
30 for row in result:
31     print('%s,%s,%.0f' % (row[0],row[1],row[2]))
32
33 cur.close()
34 conn.close()
35
```

Menghapus baris data di dalam MySQL

```
1  # Menghapus Baris Data di dalam MYSQL
2
3  import mysql.connector
4
5  conn = mysql.connector.connect(
6      user="root",
7      password="",
8      host="localhost",
9      database="pydb"
10 )
11
12 cur = conn.cursor()
13
14 # data sebelum dihapus
15 cur.execute("SELECT * FROM produk")
16 result = cur.fetchall()
17
18 for row in result:
19     print('%s,%s,%.0f' % (row[0],row[1],row[2]))
20 print("-----")
21
22 # menghapus data
23 cur.execute('DELETE FROM produk WHERE kode = %s', ('P002',))
24 conn.commit()
25
26 # data setelah dihapus
27 cur.execute("SELECT * FROM produk")
28 result = cur.fetchall()
29
30 for row in result:
31     print('%s,%s,%.0f' % (row[0],row[1],row[2]))
32
33 cur.close()
34 conn.close()
```

3. PostgreSQL

Adalah sistem manajemen database relasional (RDBMS) yang bersifat open source, yang memiliki performa stabil, keamanan tinggi, serta fitur melimpah.

postgreSQL menggunakan multi-version concurrency control (MVCC) yang memungkinkan beberapa penulis dan pembaca untuk bekerja pada sistem sekaligus.

Sama halnya dengan MySQL, paket distribusi python tidak dilengkapi dengan modul untuk mengakses database PostgreSQL. Oleh karena itu modul PostgreSQL harus terpasang secara terpisah dengan melakukan instalasi menggunakan program pip, dengan menuliskan perintah berikut ini:

```
pip install psycopg2
```

Berikut contoh program penggunaan PostgreSQL di python:

Buatlah database terlebih dahulu di dalam server PostgreSQL dan beri nama **pypg**.

Membuat tabel di dalam PostgreSQL

```
1 # Membuat Tabel didalam PostgreSQL
2
3 import psycopg2
4
5 conn = psycopg2.connect(
6     user="postgres",
7     password="postgres",
8     host="localhost",
9     database="pypg")
10
11 cur = conn.cursor()
12 sql = '''
13     CREATE TABLE produk (kode CHAR(4) NOT NULL PRIMARY KEY, nama VARCHAR(25), harga NUMERIC(9,2))'''
14
15 cur.execute(sql)
16 conn.commit()
17
18 cur.close()
19 conn.close()
```

Menambah baris data di dalam PostgreSQL

```
1 # Menambah Baris Data didalam PostgreSQL
2
3 import psycopg2
4
5 conn = psycopg2.connect(
6     user="postgres",
7     password="postgres",
8     host="localhost",
9     database="pypg")
10
11 cur = conn.cursor()
12 cur.execute("INSERT INTO produk VALUES('P001', 'Pensil', 6000)")
13 cur.execute("INSERT INTO produk VALUES('P002', 'Spidol', 9000)")
14 cur.execute("INSERT INTO produk VALUES('P003', 'Penggaris', 5000)")
15
16 conn.commit()
17 cur.close()
18 conn.close()
```

Membaca baris data di dalam PostgreSQL

```
1 # Membaca Baris Data didalam PostgreSQL
2
3 import psycopg2
4
5 conn = psycopg2.connect(
6     user="postgres",
7     password="postgres",
8     host="localhost",
9     database="pypg")
10
11 cur = conn.cursor()
12 cur.execute("SELECT * FROM produk")
13 result = cur.fetchall()
14
15 for row in result:
16     print('%s,%s,%.0f' % (row[0], row[1], row[2]))
17
18 cur.close()
19 conn.close()
```

Mengubah baris data di dalam PostgreSQL

```
1  # Mengubah Baris Data didalam PostgreSQL
2
3  import psycopg2
4
5  conn = psycopg2.connect(
6      user="postgres",
7      password="postgres",
8      host="localhost",
9      database="pypg")
10
11  cur = conn.cursor()
12
13  # data sebelum diubah
14  cur.execute("SELECT * FROM produk")
15  result = cur.fetchall()
16
17  for row in result:
18      print('%s,%s,%.0f' % (row[0],row[1],row[2]))
19  print("-----")
20
21  # mengubah data
22  cur.execute('''UPDATE produk SET harga = %s WHERE kode = %s''', (12000,'P002'))
23  conn.commit()
24
25  # data setelah diubah
26  cur.execute("SELECT * FROM produk")
27  result = cur.fetchall()
28
29  for row in result:
30      print('%s,%s,%.0f' % (row[0],row[1],row[2]))
31
32  cur.close()
33  conn.close()
```

Menghapus baris data di dalam PostgreSQL

```
1  # Menghapus Baris Data didalam PostgreSQL
2
3  import psycopg2
4
5  conn = psycopg2.connect(
6      user="postgres",
7      password="postgres",
8      host="localhost",
9      database="pypg")
10
11  cur = conn.cursor()
12
13  # data sebelum dihapus
14  cur.execute("SELECT * FROM produk")
15  result = cur.fetchall()
16
17  for row in result:
18      print('%s,%s,%.0f' % (row[0],row[1],row[2]))
19  print("-----")
20
21  # menghapus data
22  cur.execute('DELETE FROM produk WHERE kode = %s', ('P002',))
23  conn.commit()
24
25  # data setelah dihapus
26  cur.execute("SELECT * FROM produk")
27  result = cur.fetchall()
28
29  for row in result:
30      print('%s,%s,%.0f' % (row[0],row[1],row[2]))
31
32  cur.close()
33  conn.close()
```

4. Oracle

Adalah salah satu RDBMS yang digunakan untuk membangun sebuah sistem yang berskala besar.

Sama halnya dengan MySQL dan PostgreSQL, paket distribusi python tidak dilengkapi dengan modul untuk mengakses database Oracle. Oleh karena itu modul PostgreSQL harus terpasang secara terpisah dengan melakukan instalasi menggunakan program pip, dengan menuliskan perintah berikut ini:

```
pip install cx_Oracle
```

Oracle yang digunakan pada contoh ini adalah Oracle 11g Express Edition. Sebelum dilakukan koneksi pada python dilakukan pembuatan user baru yang bernama **python**

```
Run SQL Command Line

SQL*Plus: Release 11.2.0.2.0 Production on Sun May 22 17:13:39 2022

Copyright (c) 1982, 2014, Oracle. All rights reserved.

SQL> CONNECT SYS as SYSDBA
Enter password:
Connected.
SQL>
SQL> -- membuat user baru
SQL> CREATE USER python IDENTIFIED BY python;

User created.

SQL> -- memberi hak akses penuh terhadap user 'python'
SQL> GRANT DBA TO python;

Grant succeeded.

SQL> DISCONNECT
Disconnected from Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
SQL>
```

Berikut contoh program penggunaan Oracle di python:

Pada Oracle 11g Express Edition tidak diperbolehkan membuat database baru, satu-satunya database yang disediakan yaitu database XE.

Membuat tabel di dalam Oracle

```
1  # Membuat Tabel didalam Oracle
2
3  import cx_Oracle
4
5  conn = cx_Oracle.connect(
6      "python",
7      "python",
8      "127.0.0.1/XE"
9  )
10
11  cur = conn.cursor()
12  sql = '''
13      CREATE TABLE produk (
14          kode CHAR (4) NOT NULL PRIMARY KEY,
15          nama VARCHAR (25),
16          harga DECIMAL
17      )
18      ...
19  cur.execute(sql)
20  cur.close()
21  conn.close()
```


Menambah baris data di dalam Oracle

```
1  # Menambah Baris Data didalam Oracle
2
3  import cx_Oracle
4  #cx_Oracle.init_oracle_client(lib_dir=r"C:\oraclexe\instantclient_19_9")
5
6  conn = cx_Oracle.connect(
7      "python",
8      "python",
9      "127.0.0.1/XE"
10 )
11
12 cur = conn.cursor()
13 cur.execute("INSERT INTO produk VALUES('P001','Pensil',6000)")
14 cur.execute("INSERT INTO produk VALUES('P002','Spidol',9000)")
15 cur.execute("INSERT INTO produk VALUES('P003','Penggaris',5000)")
16
17 conn.commit()
18 cur.close()
19 conn.close()
```

Membaca baris data di dalam Oracle

```
1  # Membaca Baris Data didalam Oracle
2
3  import cx_Oracle
4  #cx_Oracle.init_oracle_client(lib_dir=r"C:\oraclexe\instantclient_19_9")
5
6  conn = cx_Oracle.connect(
7      "python",
8      "python",
9      "127.0.0.1/XE"
10 )
11
12 cur = conn.cursor()
13 cur.execute("SELECT * FROM produk")
14 result = cur.fetchall()
15
16 for row in result:
17     print('%s,%s,%.0f' % (row[0],row[1],row[2]))
18
19 cur.close()
20 conn.close()
```

Mengubah baris data di dalam Oracle

```
1  # Mengubah Baris Data didalam Oracle
2
3  import cx_Oracle
4
5  conn = cx_Oracle.connect(
6      "python", #username
7      "python", #password
8      "127.0.0.1/XE"
9  )
10
11  cur = conn.cursor()
12
13  # data sebelum diubah
14  cur.execute("SELECT * FROM produk")
15  result = cur.fetchall()
16
17  for row in result:
18      print('%s,%s,%.0f' % (row[0],row[1],row[2]))
19  print("-----")
20
21  # Mengubah data
22  cur.execute('''UPDATE produk
23                SET harga = :1
24                WHERE kode = :2''', (12000,'P002'))
25  conn.commit()
26
27  # data setelah diubah
28  cur.execute("SELECT * FROM produk")
29  result = cur.fetchall()
30
31  for row in result:
32      print('%s,%s,%.0f' % (row[0],row[1],row[2]))
33
34  cur.close()
35  conn.close()
```

Menghapus baris data di dalam Oracle

```
1  # Menghapus Baris Data didalam Oracle
2
3  import cx_Oracle
4
5  conn = cx_Oracle.connect(
6      "python", #username
7      "python", #password
8      "127.0.0.1/XE"
9  )
10
11  cur = conn.cursor()
12
13  # data sebelum dihapus
14  cur.execute("SELECT * FROM produk")
15  result = cur.fetchall()
16
17  for row in result:
18      print('%s,%s,%.0f' % (row[0],row[1],row[2]))
19  print("-----")
20
21  # menghapus data
22  cur.execute(''DELETE FROM produk WHERE kode = :1'', ('P002',))
23  conn.commit()
24
25  # data setelah dihapus
26  cur.execute("SELECT * FROM produk")
27  result = cur.fetchall()
28
29  for row in result:
30      print('%s,%s,%.0f' % (row[0],row[1],row[2]))
31
32  cur.close()
33  conn.close()
```