

**HANDS ON OBJECT ORIENTED DATABASE**  
**SISTEM BASIS DATA 2**



Disusun Oleh :

**Kelompok 14**

Lucky Nugroho (10121670)

Raqa Andala (11121077)

Rafi Amirul Linarto (11121039)

Depok

2023

UNIVERSITAS GUNADARMA

Studi Kasus :

Membuat pendataan buku perpustakaan menggunakan Object Oriented Programming pada database.

Batasan Masalah :

- Perancangan ini dibuat untuk perpustakaan pribadi yang jumlah bukunya tidak terlalu banyak
- Perancangan ini berfokus pada user pribadi untuk mendata serta menampilkan judul, penulis, dan tahun terbit.

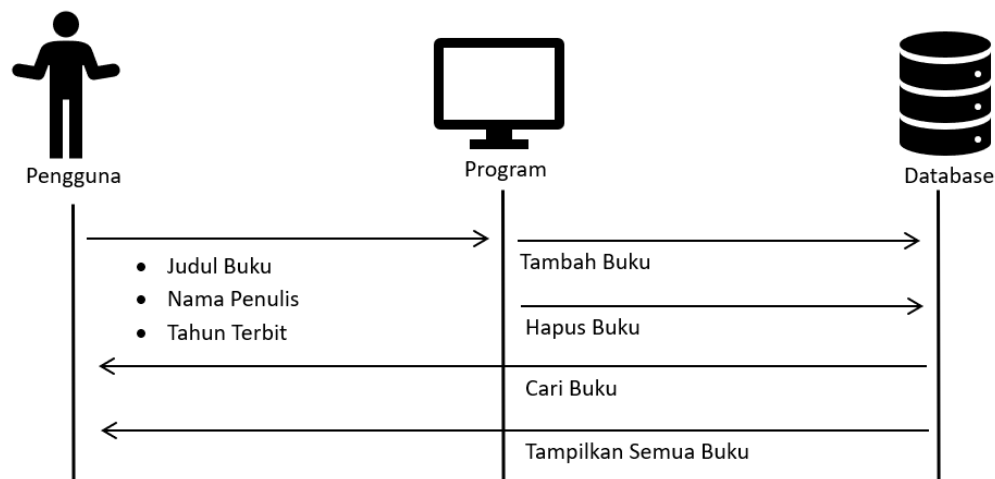
Fase 1 Pengumpulan data dan analisis

User/Pengguna : Orang/Individu pribadi

Data yang dipakai : Judul buku, nama penulis, tahun terbit.

Data yang dipakai hanya sedikit karena pada OOP database data tersebut nantinya dapat digunakan oleh beberapa fungsi. OOP tidak mendukung RDBMS dan tidak mendukung penggunaan object yang kompleks.

Data Flow Diagram



Fase 2 Perancangan Data Secara Konseptual

1. User memasukkan judul buku, nama penulis, dan tahun terbit
2. Program akan menyimpan ketiga variabel tersebut
3. User menentukan apakah variabel tersebut akan ditambahkan atau dihapus
4. User dapat mencari buku dengan judul tertentu
5. User dapat menampilkan seluruh judul buku yang ada di dalam database

### Fase 3 Perancangan Data Secara Logical

1. Import library python sqlite3

```
1 import sqlite3
2
```

2. Definiskan kelas untuk buku

```
3 # Definisikan kelas Buku
4 class Buku:
5     def __init__(self, judul, penulis, tahun_terbit):
6         self.judul = judul
7         self.penulis = penulis
8         self.tahun_terbit = tahun_terbit
9
10    def __str__(self):
11        return f"{self.judul} oleh {self.penulis} ({self.tahun_terbit})"
12
```

3. Definiskan kelas untuk database buku

- a. Membuat Table buku

```
13 # Definisikan kelas DatabaseBuku
14 class DatabaseBuku:
15     def __init__(self, nama_database):
16         self.nama_database = nama_database
17         self.conn = sqlite3.connect(nama_database)
18         self.cursor = self.conn.cursor()
19         self.cursor.execute("CREATE TABLE IF NOT EXISTS buku (judul TEXT, penulis TEXT, tahun_terbit INTEGER)")
20
```

- b. Funtion untuk menambahkan buku ke dalam table

```
21 def tambah_buku(self, buku):
22     self.cursor.execute("INSERT INTO buku VALUES (?, ?, ?)", (buku.judul, buku.penulis, buku.tahun_terbit))
23     self.conn.commit()
24     print("Buku berhasil ditambahkan ke database.")
```

- c. Funtion untuk menghapus buku dari table berdasarkan judul

```
26 def hapus_buku(self, judul):
27     self.cursor.execute("DELETE FROM buku WHERE judul = ?", (judul,))
28     self.conn.commit()
29     print("Buku berhasil dihapus dari database.")
```

- d. Function untuk mencari buku dalam table berdasarkan judul buku

```
31 def cari_buku(self, judul):
32     self.cursor.execute("SELECT * FROM buku WHERE judul = ?", (judul,))
33     result = self.cursor.fetchone()
34     if result:
35         buku = Buku(result[0], result[1], result[2])
36         print("Buku ditemukan dalam database.")
37         print(buku)
38     else:
39         print("Buku tidak ditemukan dalam database.")
```

- e. Function untuk menampilkan semua buku yang ada dalam database

```
41     def tampilkan_semua_buku(self):
42         self.cursor.execute("SELECT * FROM buku")
43         results = self.cursor.fetchall()
44         if results:
45             print("Daftar Buku:")
46             for result in results:
47                 buku = Buku(result[0], result[1], result[2])
48                 print(buku)
49         else:
50             print("Database kosong.")
```

- f. Function close supaya tidak ada lagi akses dari database sehingga tidak proses jalan terus menerus

```
52     def __del__(self):
53         self.cursor.close()
54         self.conn.close()
55
```

- g. Memanggil function untuk menggunakan database

```
56 # Contoh penggunaan
57 db_buku = DatabaseBuku("perpustakaan.db")
58
59 # Menambahkan buku ke database
60 buku1 = Buku("Python Programming", "John Smith", 2020)
61 db_buku.tambah_buku(buku1)
62
63 # Menampilkan semua buku dalam database
64 db_buku.tampilkan_semua_buku()
65
66 # Mencari buku dalam database
67 db_buku.cari_buku("Python Programming")
68
69 # Menghapus buku dari database
70 db_buku.hapus_buku("Python Programming")
71
72 # Menampilkan semua buku setelah penghapusan
73 db_buku.tampilkan_semua_buku()
```

#### Fase 4 Perancangan Data Secara Fisik

##### 1. Response time :

Dalam contoh studi kasus sederhana di atas, menggunakan SQLite sebagai sistem manajemen basis data dalam bahasa pemrograman Python, kita dapat memperkirakan bahwa respons time relatif cepat karena jumlah data yang disimpan tidak terlalu besar.

Respons time pada operasi seperti tambah\_buku, cari\_buku, hapus\_buku, dan tampilkan\_semua\_buku biasanya akan sangat cepat, dalam rentang milidetik hingga beberapa puluh milidetik, tergantung pada kinerja perangkat keras dan ukuran database.

Namun, penting untuk dicatat bahwa perkiraan ini sangat kasar dan hanya berdasarkan pada contoh studi kasus sederhana yang diberikan. Faktor-faktor seperti kinerja perangkat keras, ukuran database, jumlah pengguna, dan jenis operasi yang dilakukan akan mempengaruhi respons time yang sebenarnya pada sistem basis data yang lebih kompleks.

## 2. Space utility :

jumlah ruang penyimpanan yang digunakan oleh file-file database dan struktur jalur akses. Dalam database ini penyimpanan yang diperlukan sebesar kurang lebih 84Mb untuk 1.000.000 (Menurut perhitungan AI)

## Fase 5 Pemilihan bahasa pemrograman dan DBMS

Python dengan library sqlite3 dipilih karena bahasa pemrograman python sudah terkenal andal dalam menjalankan perhitungan data ataupun kalkulasi data. Selain itu juga python dipilih karena bahasa pemrograman ini termasuk bahasa pemrograman dengan kategori sintaks yang mudah dipelajari dan dipahami. Selain itu python ini juga bisa dijalankan di banyak IDE seperti Visual Studio Code sehingga memudahkan pengguna individu untuk memakainya.

## Fase 6 Implementasi Sistem Database

Hasil dalam Visual Studio Code menggunakan bahasa pemrograman Python

### 1. Insert data buku baru ke dalam database

```
56 # Contoh penggunaan
57 db_buku = DatabaseBuku("perpustakaan.db")
58
59 # Menambahkan buku ke database
60 buku1 = Buku("Manajemen Sistem Informasi", "Raymond McLeod", 2004)
61 db_buku.tambah_buku(buku1)
62 buku2 = Buku("Python Programming", "John Smith", 2020)
63 db_buku.tambah_buku(buku2)
64 buku3 = Buku("Pengantar Akuntansi", "Kimmel Kieso", 2018)
65 db_buku.tambah_buku(buku3)
```

Hasil :

```
PS C:\Users\Lucky Nugroho\Documents\Tugas\SBD 2>
sers/Lucky Nugroho/Documents/Tugas/SBD 2/OOP_db.p
Buku berhasil ditambahkan ke database.
Buku berhasil ditambahkan ke database.
Buku berhasil ditambahkan ke database.
```

2. Menampilkan semua buku yang ada pada database

```
59 # Menampilkan semua buku dalam database
60 db_buku.tampilkan_semua_buku()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\Lucky Nugroho\Documents\Tugas\SBD 2> & "C:/Users/Lucky Nugroho/Documents/Tugas/SBD 2/OOP\_db.py"  
Daftar Buku:  
Manajemen Sistem Informasi oleh Raymond McLeod (2004)  
Python Programming oleh John Smith (2020)  
Pengantar Akuntansi oleh Kimmel Kieso (2018)

3. Mencari buku berdasarkan judul buku

```
59 # Mencari buku dalam database
60 db_buku.cari_buku("Python Programming")
61 db_buku.cari_buku("Pengantar Akuntansi")
62 db_buku.cari_buku("Manajemen Sistem Informasi")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\Lucky Nugroho\Documents\Tugas\SBD 2> & "C:/Users/Lucky Nugroho/Documents/Tugas/SBD 2/OOP\_db.py"  
Buku ditemukan dalam database.  
Python Programming oleh John Smith (2020)  
Buku ditemukan dalam database.  
Pengantar Akuntansi oleh Kimmel Kieso (2018)  
Buku ditemukan dalam database.  
Manajemen Sistem Informasi oleh Raymond McLeod (2004)

4. Menghapus buku dan menampilkan list buku setelah melakukan penghapusan

```
59 # Menghapus buku dari database
60 db_buku.hapus_buku("Python Programming")
61
62 # Menampilkan semua buku setelah penghapusan
63 db_buku.tampilkan_semua_buku()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\Lucky Nugroho\Documents\Tugas\SBD 2> & "C:/Users/Lucky Nugroho/Documents/Tugas/SBD 2/OOP\_db.py"  
Buku berhasil dihapus dari database.  
Daftar Buku:  
Manajemen Sistem Informasi oleh Raymond McLeod (2004)  
Pengantar Akuntansi oleh Kimmel Kieso (2018)  
PS C:\Users\Lucky Nugroho\Documents\Tugas\SBD 2> █

User yang dapat memakai :

- Lulusan D3/S1 Komputer
- Berpengalaman memakai visual studio code
- Pernah menggunakan python sebagai bahasa pemrograman