

Roadmap CRUD E-Commerce Python untuk Kelas XI

Bab 1: Pengenalan CRUD & E-Commerce

(sudah dibuat di atas)

Bab 2: Database Dasar untuk E-Commerce

(sudah dibuat di atas)

Bab 3: CRUD Produk Lengkap di Database

(sudah dibuat di atas)

Bab 4: CRUD User (Registrasi & Login)

(sudah dibuat di atas)

Bab 5: Fitur Keranjang Belanja (Shopping Cart)

(sudah dibuat di atas)

Bab 6: Checkout & Transaksi

Tujuan Pembelajaran

- Siswa memahami proses checkout pada aplikasi e-commerce.
- Siswa dapat membuat tabel transaksi dan menyimpan data pembelian.
- Siswa dapat menghitung total harga dari isi keranjang.

Konsep Dasar

Checkout adalah tahap akhir belanja di mana user mengonfirmasi produk di keranjang, total harga, dan menyimpannya ke dalam transaksi.

Langkah Praktik

1. Buat Tabel Transaksi

```
CREATE TABLE transaksi (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    user_id INTEGER,  
    total_harga REAL,  
    tanggal TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
```

```
FOREIGN KEY(user_id) REFERENCES users(id)
);
```

Tabel Detail Transaksi

```
CREATE TABLE detail_transaksi (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    transaksi_id INTEGER,
    produk_id INTEGER,
    jumlah INTEGER,
    harga REAL,
    FOREIGN KEY(transaksi_id) REFERENCES transaksi(id),
    FOREIGN KEY(produk_id) REFERENCES produk(id)
);
```

1. Fungsi Checkout

```
def checkout(user_id):
    conn = sqlite3.connect('ecommerce.db')
    cursor = conn.cursor()

    # Ambil isi keranjang
    cursor.execute("""
        SELECT produk.id, produk.harga, keranjang.jumlah
        FROM keranjang
        JOIN produk ON keranjang.produk_id = produk.id
        WHERE keranjang.user_id = ?
    """, (user_id,))
    items = cursor.fetchall()

    if not items:
        print("Keranjang kosong, tidak bisa checkout!")
        return

    # Hitung total harga
    total = sum(item[1] * item[2] for item in items)

    # Simpan transaksi
    cursor.execute("INSERT INTO transaksi (user_id, total_harga) VALUES (?, ?)", (user_id, total))
    transaksi_id = cursor.lastrowid

    # Simpan detail transaksi
    for produk_id, harga, jumlah in items:
        cursor.execute("INSERT INTO detail_transaksi (transaksi_id, produk_id, jumlah, harga) VALUES (?, ?, ?, ?)", (transaksi_id, produk_id, jumlah, harga))

    # Kosongkan keranjang setelah checkout
```

```
cursor.execute("DELETE FROM keranjang WHERE user_id = ?",
              (user_id,))

conn.commit()
conn.close()
print(f"Checkout berhasil! Total belanja Rp{total}")
```

Tugas Siswa

1. Implementasikan fitur checkout dengan menyimpan transaksi ke database.
2. Uji dengan menambahkan beberapa produk ke keranjang lalu lakukan checkout.
3. Tampilkan riwayat transaksi yang sudah dilakukan user.

Refleksi

Dengan menambahkan fitur **checkout**, aplikasi e-commerce Python semakin menyerupai aplikasi nyata, karena sudah memiliki alur belanja lengkap dari keranjang hingga transaksi tersimpan.