

---

**Soal Proyek: Membuat Aplikasi Web CRUD dengan Python, Django dan PostgreSQL****Deskripsi Proyek:**

Buatlah sebuah aplikasi web sederhana menggunakan **Python Framework Django**. Aplikasi ini harus memiliki fitur CRUD (Create, Read, Update, Delete) untuk dua entitas yang memiliki relasi **one-to-many**. Backend bisa menggunakan **PostgreSQL** yang dihubungkan melalui API (REST API atau GraphQL). Sebagai contoh, buatlah aplikasi manajemen **Buku** dan **Kategori**, di mana setiap buku terkait dengan satu kategori.

**Langkah-langkah:****Instalasi Python, Django dan PostgreSQL**

1. Buat proyek baru menggunakan **Django**.
2. Instal dependensi yang dibutuhkan untuk interaksi dengan API dan Frontend.
3. Backend dapat menggunakan **Django** yang mendukung PostgreSQL untuk menyediakan endpoint API yang berhubungan dengan database.

**Membuat Database dan API**

1. Buat backend API yang menggunakan **PostgreSQL** sebagai database. API ini harus menyediakan endpoint untuk entitas Kategori dan Buku.
  1. **Kategori**: Endpoint CRUD untuk mengelola kategori.
  2. **Buku**: Endpoint CRUD untuk mengelola buku dan relasinya dengan kategori.

Struktur tabel bisa sebagai berikut:

1. **Kategori**
  1. id: Primary key.
  2. nama: Nama kategori buku.
  3. createdAt: Waktu ketika kategori dibuat.
  4. updatedAt: Waktu ketika kategori diubah.
2. **Buku**
  1. id: Primary key.
  2. judul: Nama buku.
  3. kategoriId: Foreign key yang merujuk ke entitas Kategori.
  4. penulis: Nama penulis buku.
  5. createdAt: Waktu ketika buku dibuat.
  6. updatedAt: Waktu ketika buku diubah.

---

### Membuat Frontend dengan Python, Django

1. **Home Page:** Halaman berisi daftar buku lengkap dengan kategori buku.
2. **Halaman Tambah/Edit Buku:** Halaman untuk menambah dan mengedit data buku.
3. **Halaman Daftar Kategori:** Halaman yang menampilkan daftar kategori buku.
4. **Halaman Tambah/Edit Kategori:** Halaman untuk menambah dan mengedit kategori.

Buat halaman **Python, Django** berikut:

1. **/buku:** Menampilkan daftar buku dengan fitur CRUD.
2. **/buku/tambah:** Form untuk menambah buku baru.
3. **/buku/edit/[id]:** Form untuk mengedit data buku berdasarkan ID.
4. **/kategori:** Menampilkan daftar kategori dengan fitur CRUD.
5. **/kategori/tambah:** Form untuk menambah kategori baru.
6. **/kategori/edit/[id]:** Form untuk mengedit data kategori berdasarkan ID.

### Validasi Form dan Pesan Error

1. Implementasikan validasi pada form input ketika menambah dan mengedit buku serta kategori.
2. Tampilkan pesan error jika ada kesalahan dalam input form.

### Uji Coba Aplikasi

1. Pastikan aplikasi dapat melakukan operasi CRUD pada kedua entitas (Kategori dan Buku), dan data tersimpan di database PostgreSQL.

### Tugas Tambahan:

1. Implementasikan **pagination** pada halaman daftar buku.
2. Tambahkan fitur pencarian buku berdasarkan judul menggunakan query API.
3. Tambahkan fitur **sorting** pada daftar buku (misalnya, berdasarkan judul atau tanggal).

### Kriteria Penilaian:

- Aplikasi dapat menampilkan daftar buku dan kategori dengan relasi yang benar.
- Fitur CRUD berfungsi dengan baik pada kedua entitas.
- Validasi input diterapkan dengan baik.
- Aplikasi memiliki tampilan antarmuka yang responsif dan mudah digunakan.
- Penggunaan API untuk komunikasi antara frontend dan backend terimplementasi dengan baik.