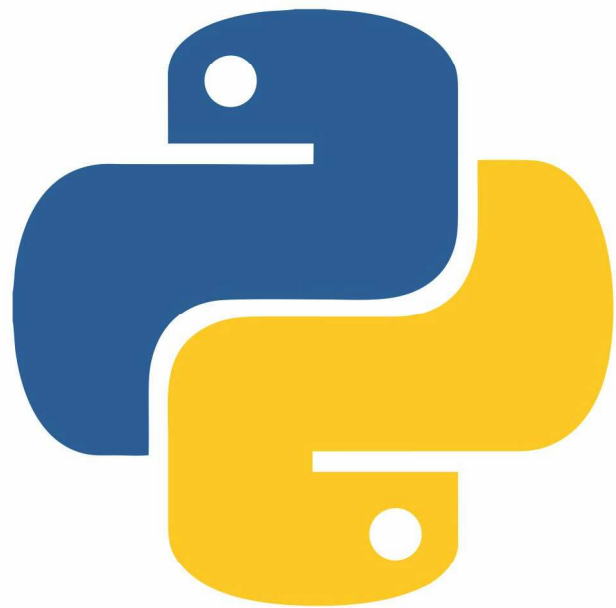


PYTHON

django

web framework

by Kevin Perdana

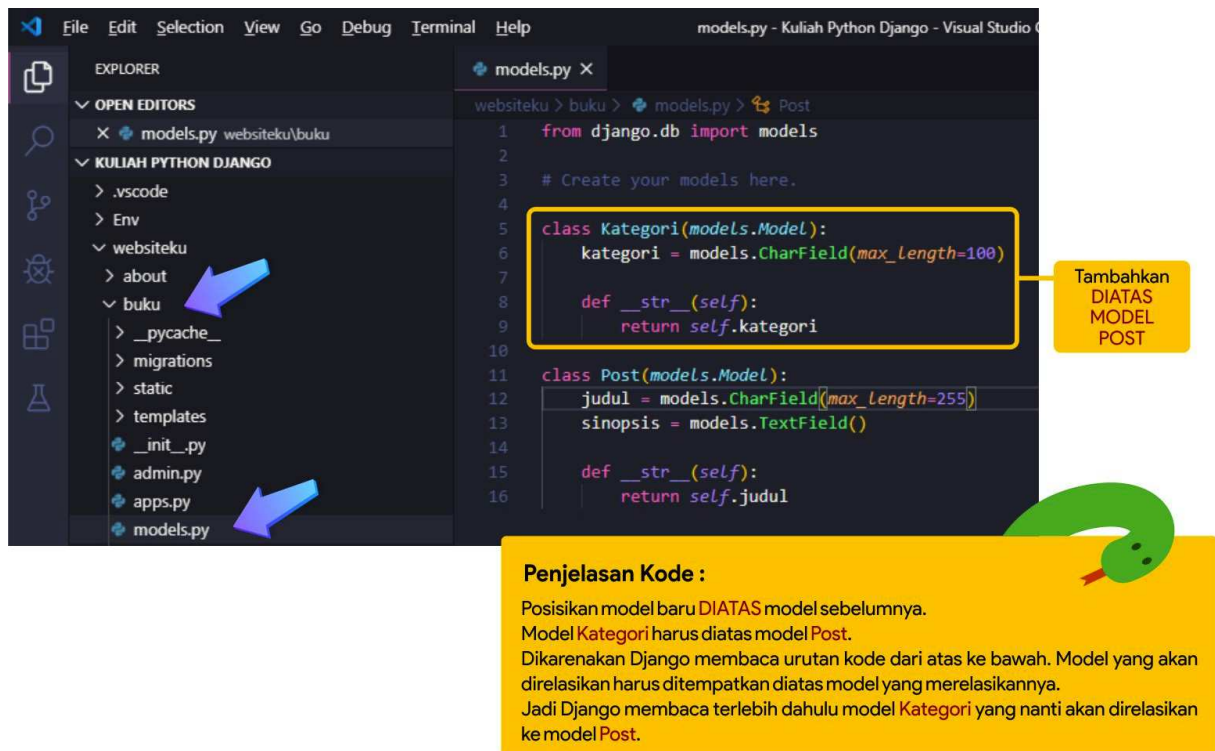


Relasi Model Django

11.1 Relasi Dua Model pada App

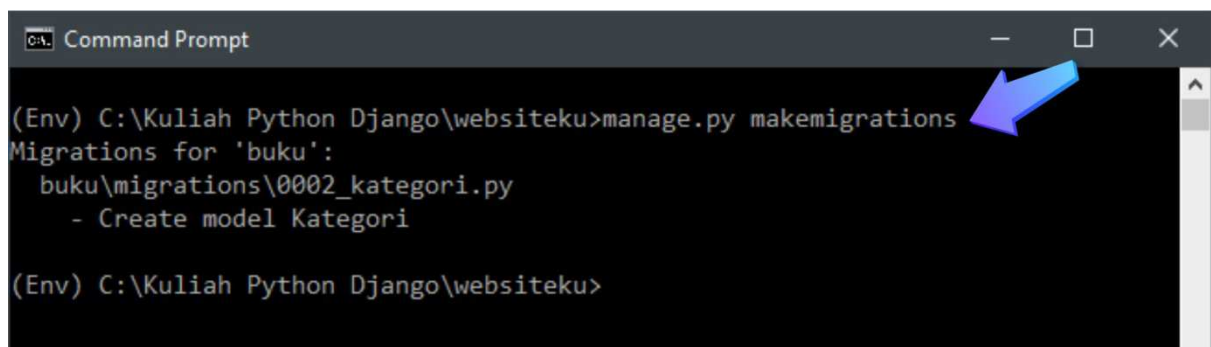
Langkah-langkah berikut ini adalah lanjutan dari modul sebelumnya.

1. Buka file **models.py** pada app **buku**, tambahkan kode seperti gambar di bawah.



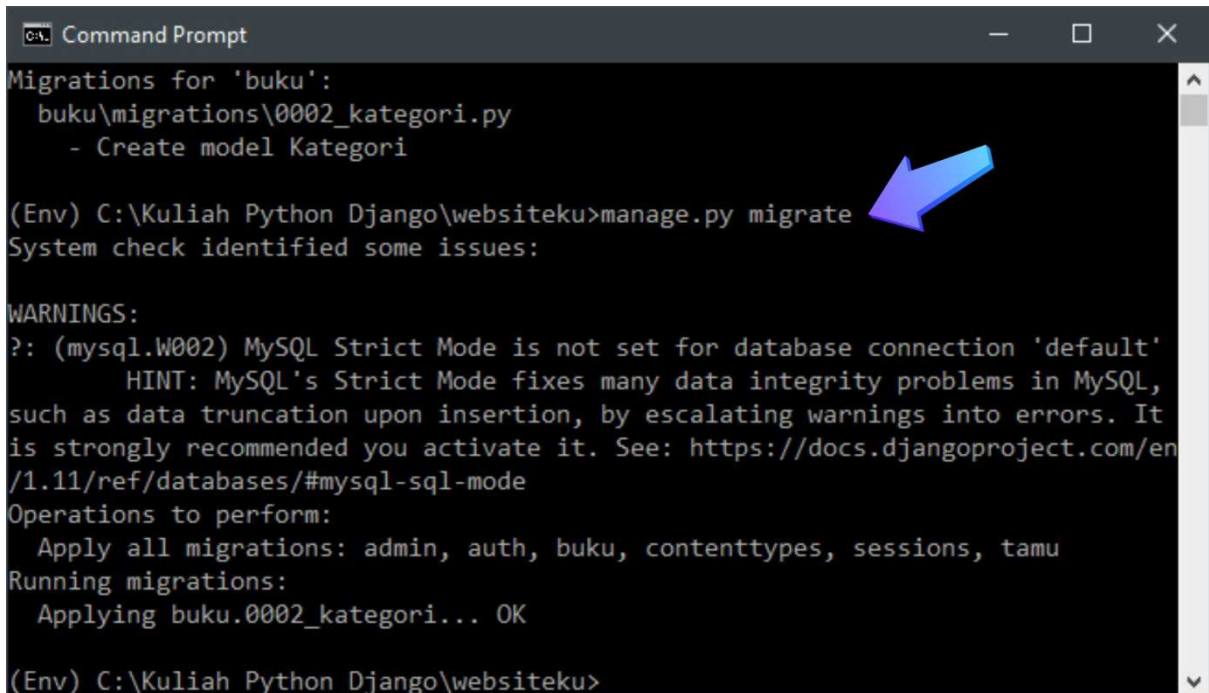
Gambar 156 models.py app buku

2. Buka cmd, ketik **manage.py makemigrations**. Lakukan ini ketika ada penambahan atau perubahan pada model.



Gambar 157 Migrasi model

3. Ketik **manage.py migrate** untuk eksekusi migrasi.



```
Command Prompt
Migrations for 'buku':
  buku\migrations\0002_kategori.py
    - Create model Kategori

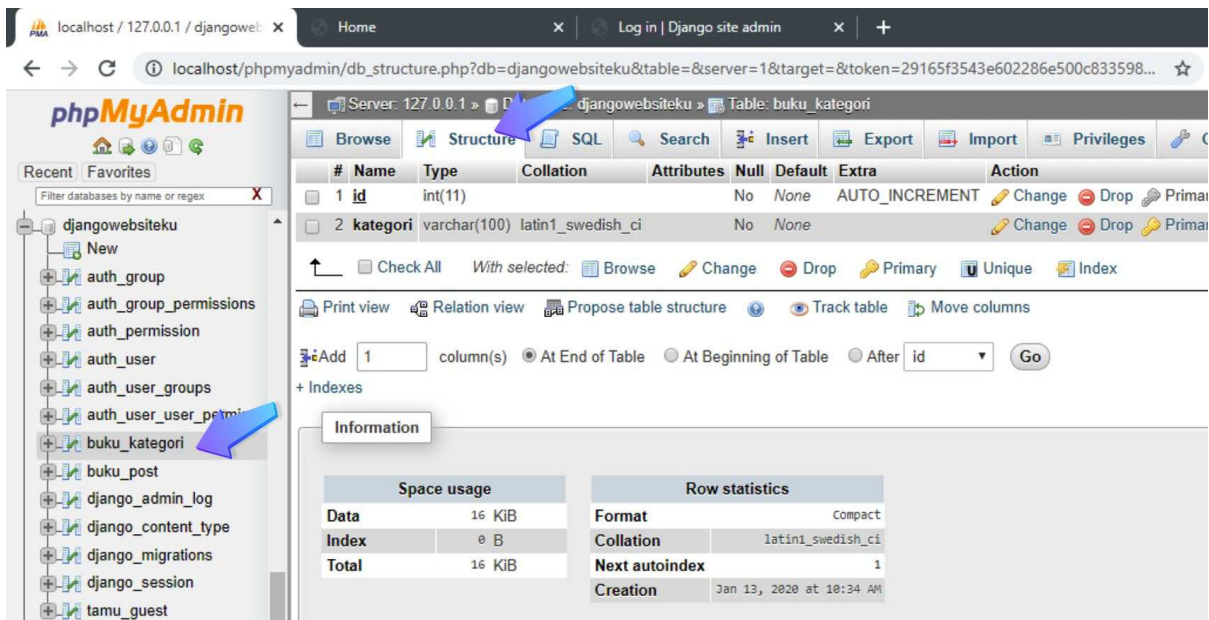
(Erv) C:\Kuliah Python Django\websiteku>manage.py migrate
System check identified some issues:

WARNINGS:
?: (mysql.W002) MySQL Strict Mode is not set for database connection 'default'
   HINT: MySQL's Strict Mode fixes many data integrity problems in MySQL,
   such as data truncation upon insertion, by escalating warnings into errors. It
   is strongly recommended you activate it. See: https://docs.djangoproject.com/en
   /1.11/ref/databases/#mysql-sql-mode
Operations to perform:
  Apply all migrations: admin, auth, buku, contenttypes, sessions, tamu
Running migrations:
  Applying buku.0002_kategori... OK

(Erv) C:\Kuliah Python Django\websiteku>
```

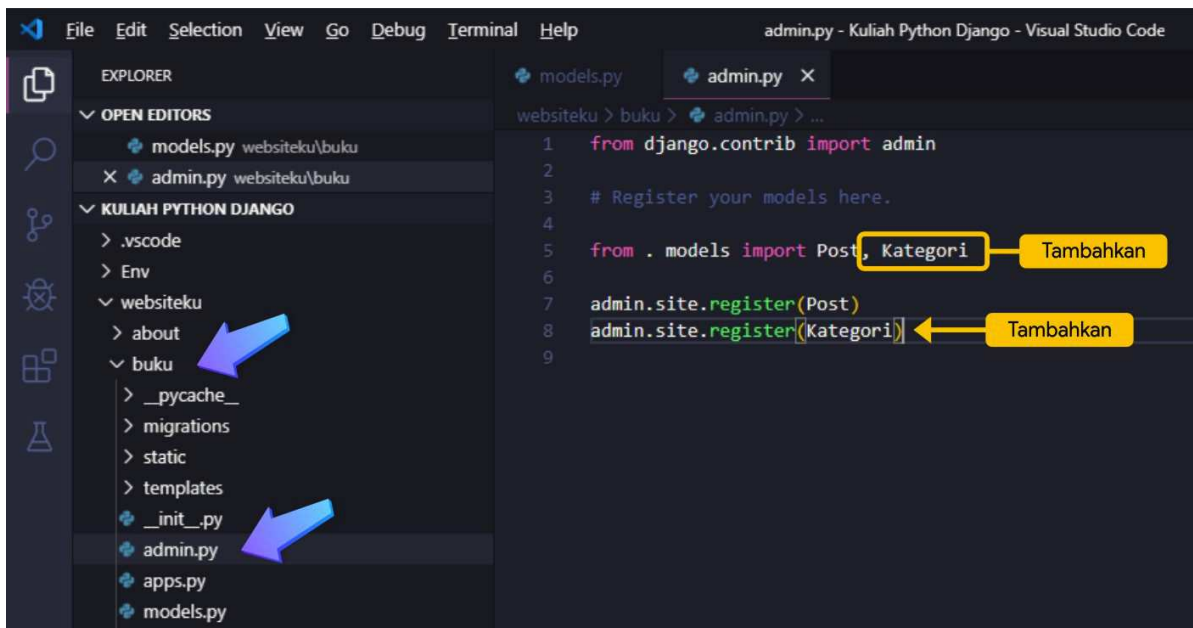
Gambar 158 Eksekusi migrasi

4. Cek database seperti gambar di bawah. Terdapat tambahan nama tabel **buku_kategori**.



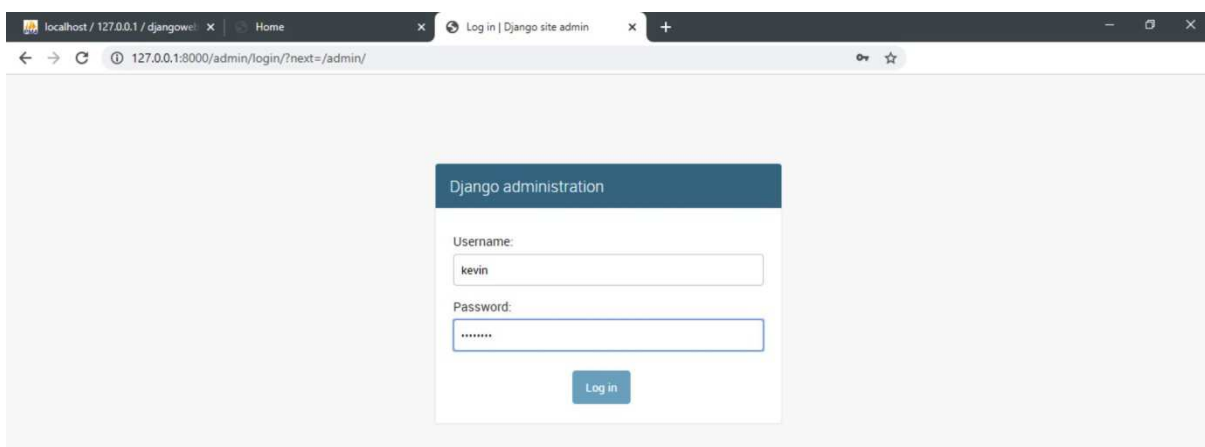
Gambar 159 Cek database

5. Buka file **admin.py** pada app **buku**, tambahkan kode sebagai berikut.



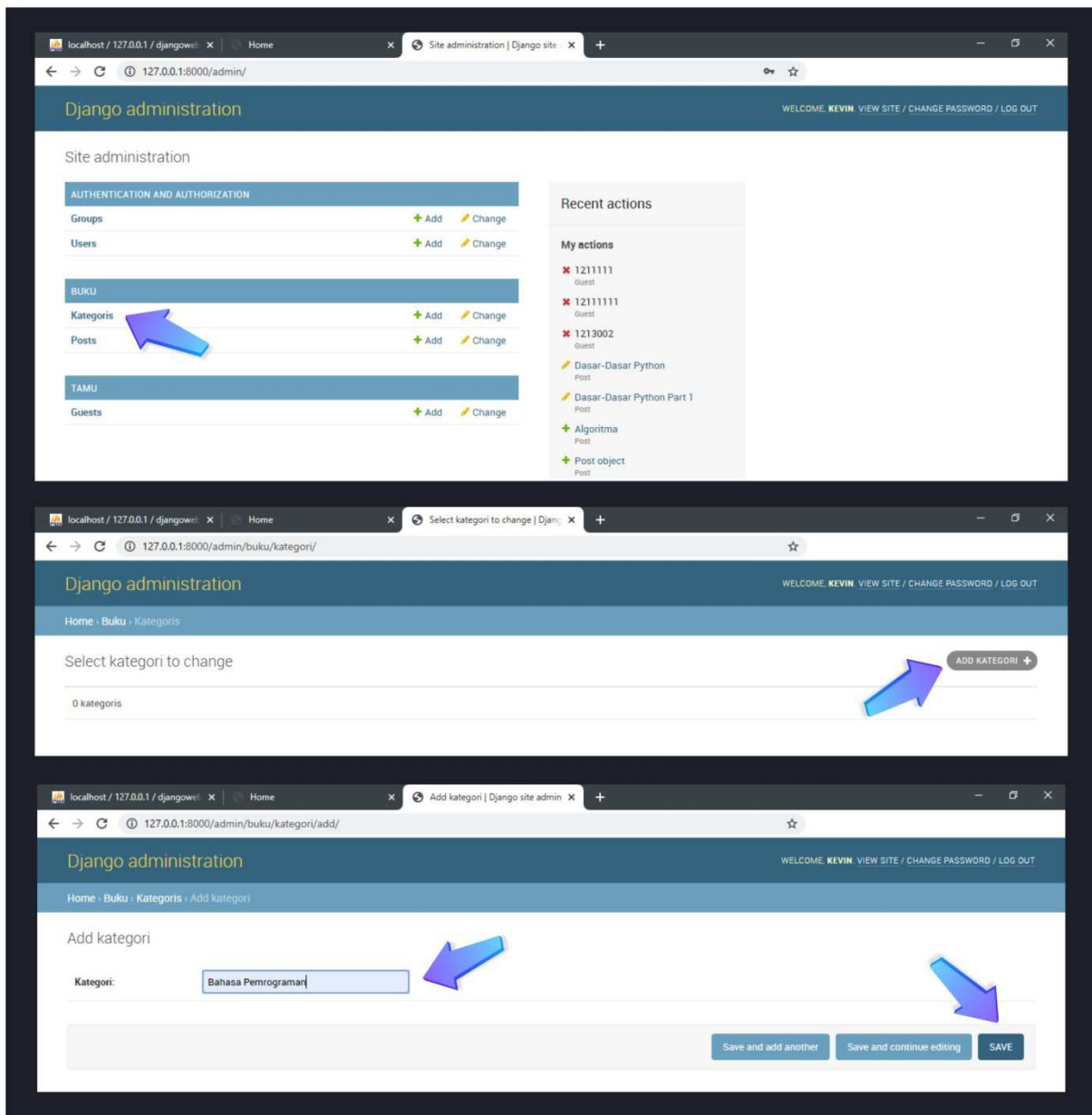
Gambar 160 admin.py app buku

6. Lakukan login.



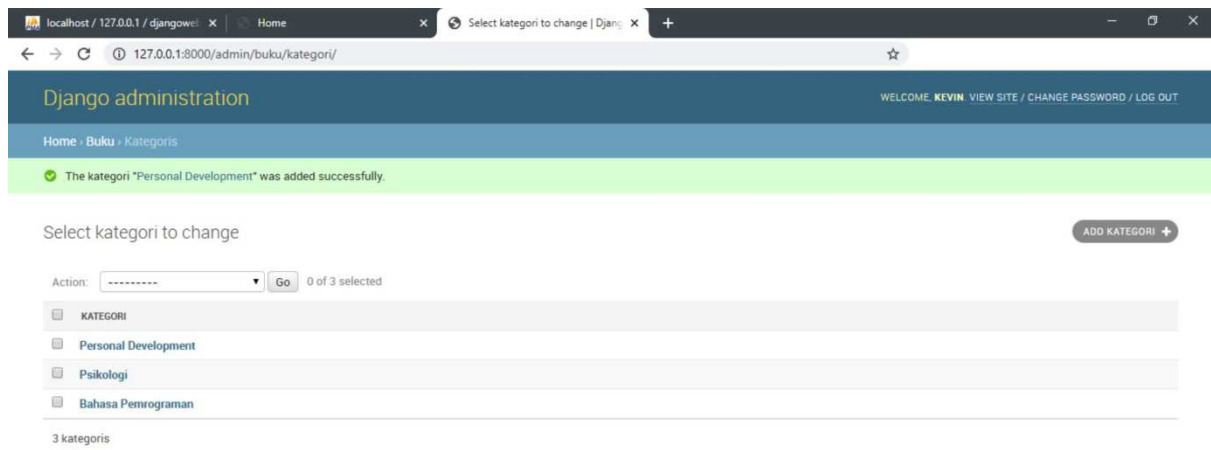
Gambar 161 Login

7. Tambahkan kategori dengan nama **Bahasa Pemrograman**. Contoh seperti gambar di bawah.



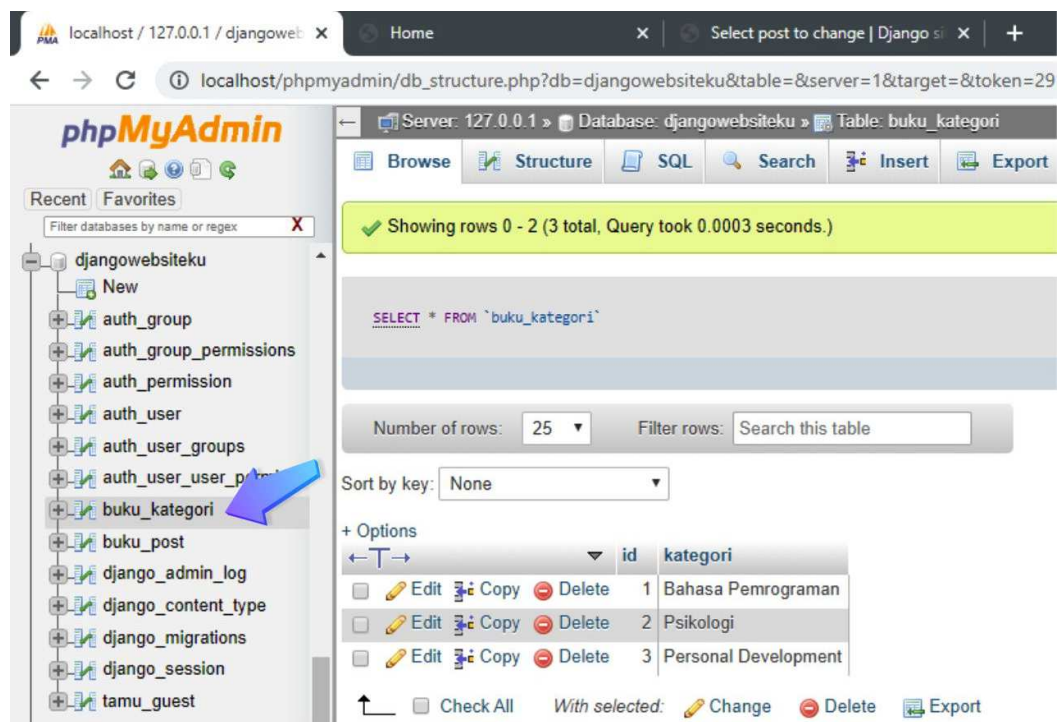
Gambar 162 Menambah kategori

8. Tambah kategori lain.



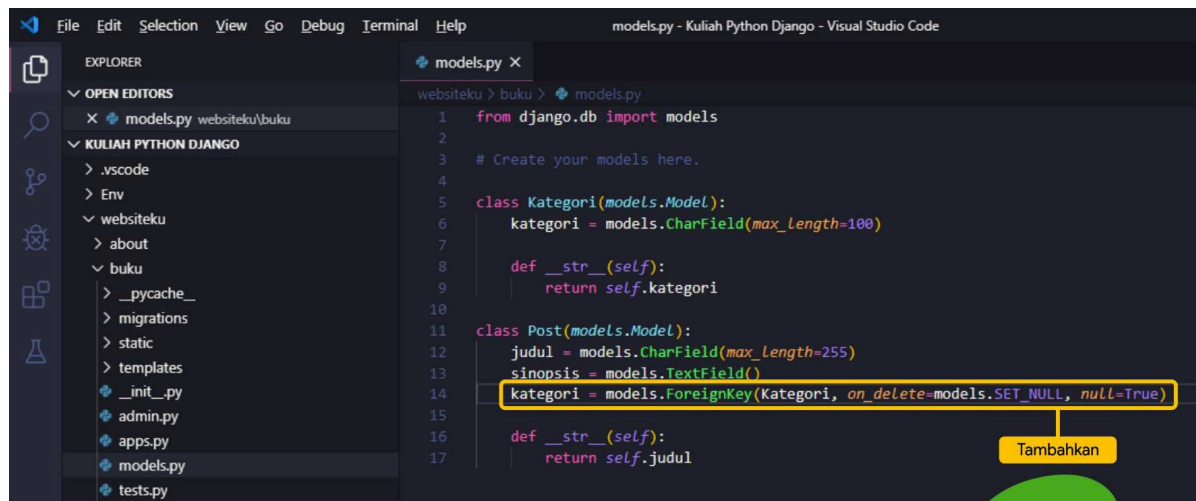
Gambar 163 Tambah kategori lain

9. Cek record yang masuk pada database.



Gambar 164 Cek database

10. Buka kembali file **models.py** pada app **buku**. Tambahkan kode di bawah ini.



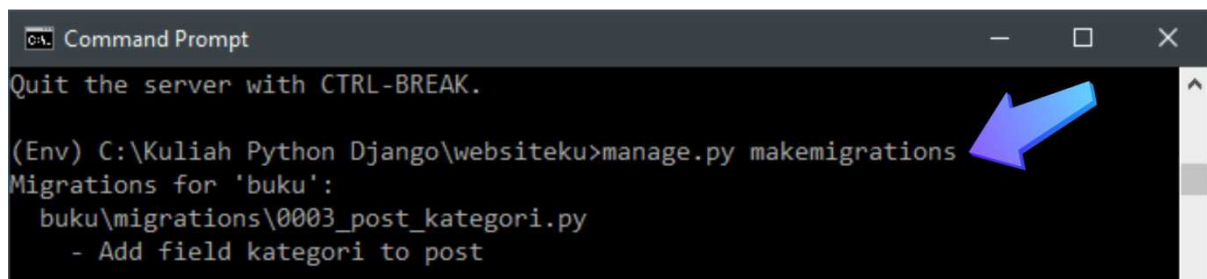
```
1 from django.db import models
2
3 # Create your models here.
4
5 class Kategori(models.Model):
6     kategori = models.CharField(max_length=100)
7
8     def __str__(self):
9         return self.kategori
10
11 class Post(models.Model):
12     judul = models.CharField(max_length=255)
13     sinopsis = models.TextField()
14     kategori = models.ForeignKey(Kategori, on_delete=models.SET_NULL, null=True)
15
16     def __str__(self):
17         return self.judul
```

Penjelasan Kode :

Atribut kategori yang ditambahkan akan menjadi Foreign Key pada tabel buku_post yang mengacu pada tabel buku_kategori. on_delete=models.SET_NULL apabila sebuah kategori dihapus, maka record yang berelasi dengan kategori tersebut tidak akan terhapus. Ini berlawanan dengan on_delete=models.CASCADE yang apabila sebuah kategori dihapus, maka record yang berelasi dengan kategori tersebut ikut terhapus. null=True artinya jika record pada field kategori (Foreign Key) tidak ada nilai, aplikasi masih bisa berjalan, (membuat elemen Null pada Database menjadi Yes).

Gambar 165 models.py app buku

11. Lakukan migrasi.



```
C:\Kuliah Python Django\websiteku>manage.py makemigrations
Quit the server with CTRL-BREAK.
Migrations for 'buku':
  buku\migrations\0003_post_kategori.py
  - Add field kategori to post
```

Gambar 166 Migrasi model

12. Eksekusi migrasi.

```
- Add field kategori to post

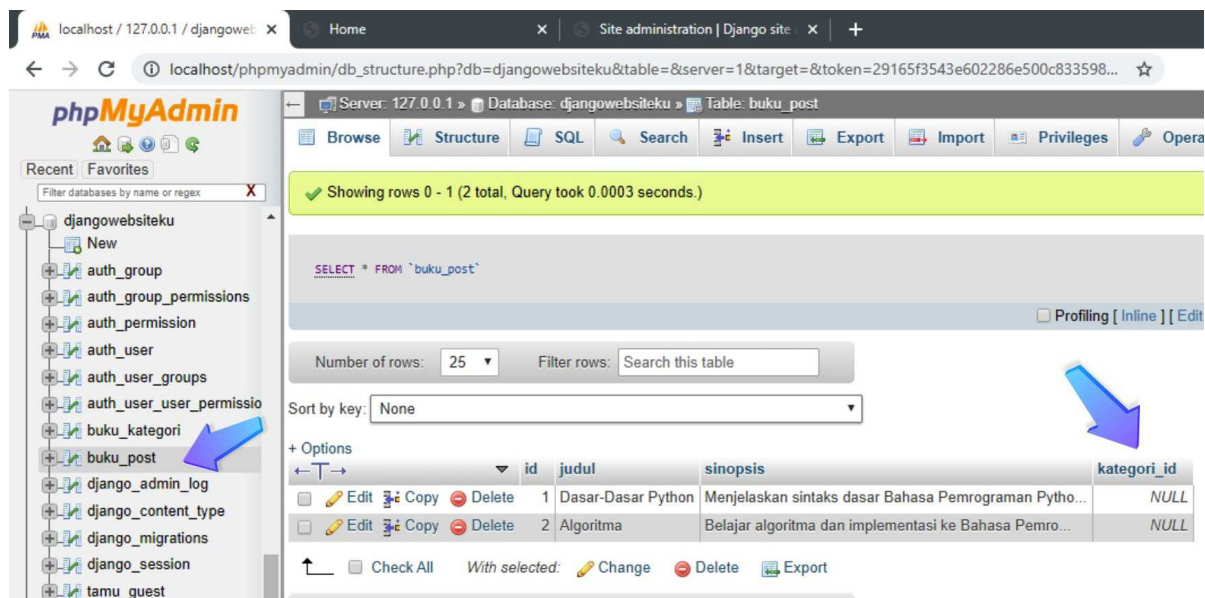
(Env) C:\Kuliah Python Django\websiteku>manage.py migrate
System check identified some issues:

WARNINGS:
?: (mysql.W002) MySQL Strict Mode is not set for database connection 'default'
   HINT: MySQL's Strict Mode fixes many data integrity problems in MySQL,
   such as data truncation upon insertion, by escalating warnings into errors. It
   is strongly recommended you activate it. See: https://docs.djangoproject.com/en
   /1.11/ref/databases/#mysql-sql-mode
Operations to perform:
  Apply all migrations: admin, auth, buku, contenttypes, sessions, tamu
Running migrations:
  Applying buku.0003_post_kategori... OK

(Env) C:\Kuliah Python Django\websiteku>
```

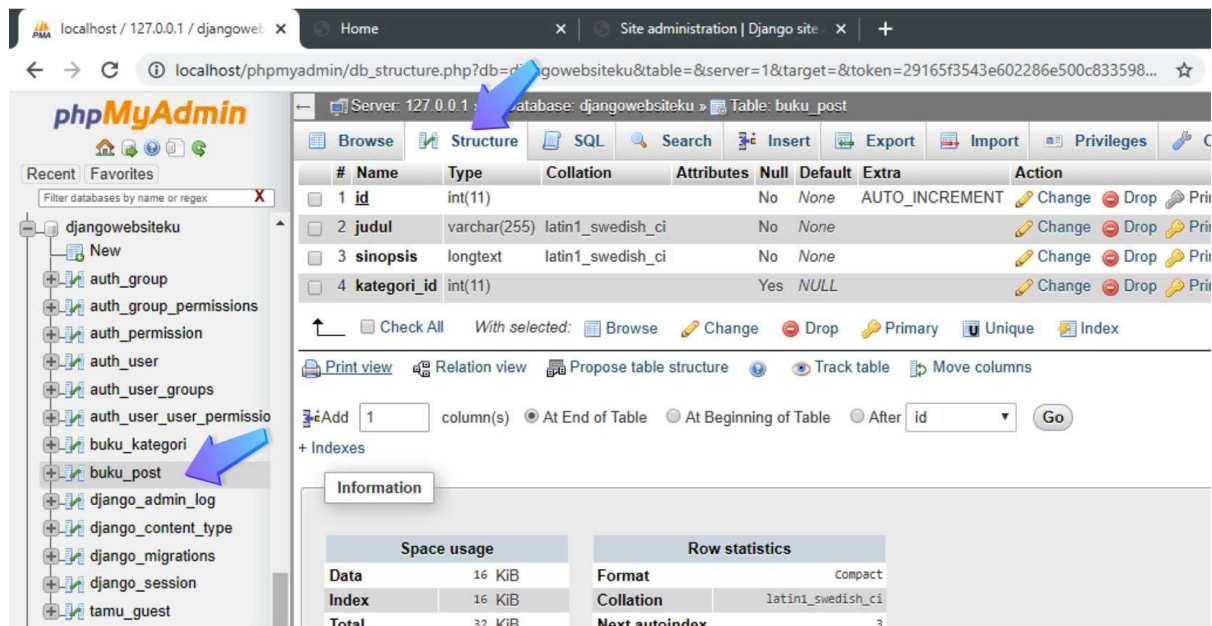
Gambar 167 Eksekusi migrasi

13. Cek database, terdapat penambahan field.



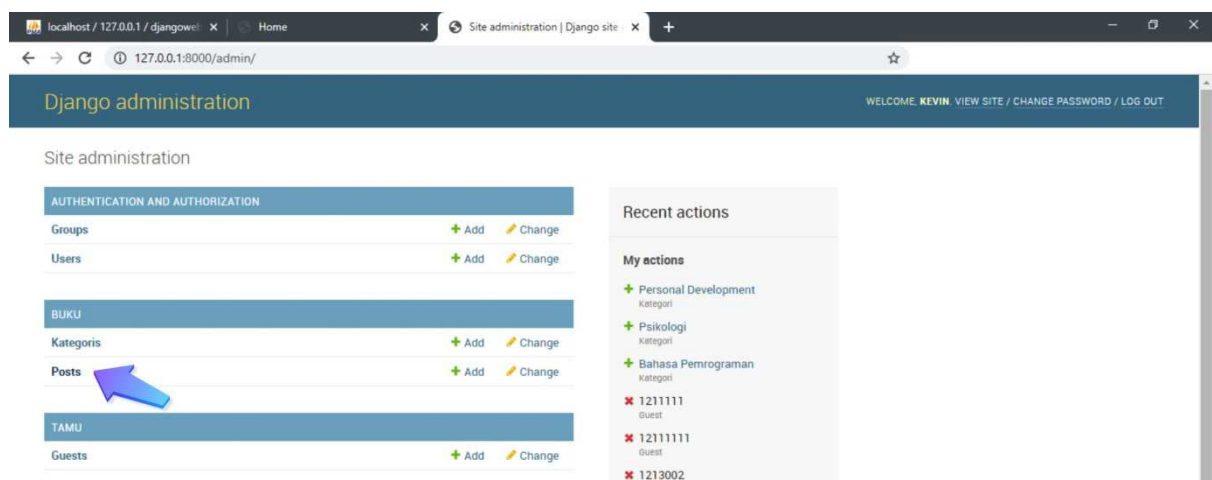
Gambar 168 Cek database

14. Lihat tipe datanya seperti gambar di bawah.



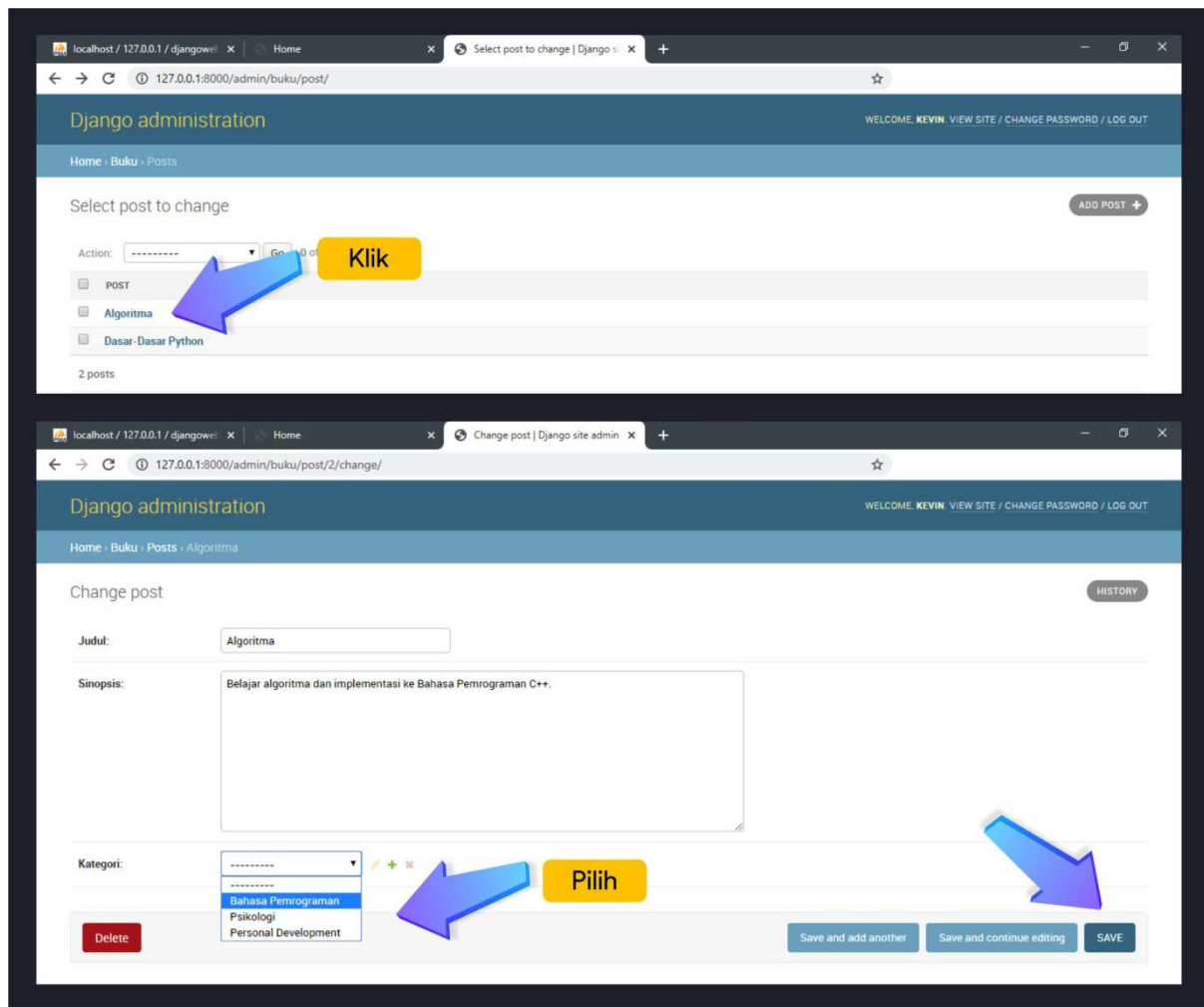
Gambar 169 Cek tipe data field database

15. Klik model **Posts**.



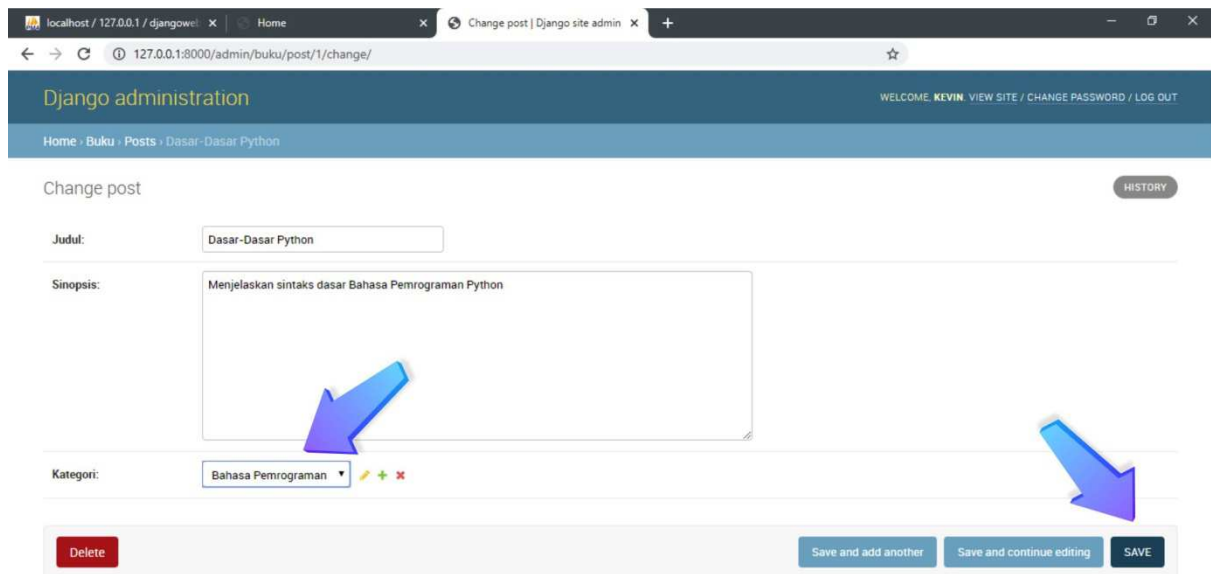
Gambar 170 Model Posts

16. Klik **judul buku** yang telah diinput sebelumnya, dan lakukan edit pada **kategori**, kemudian **Save**.



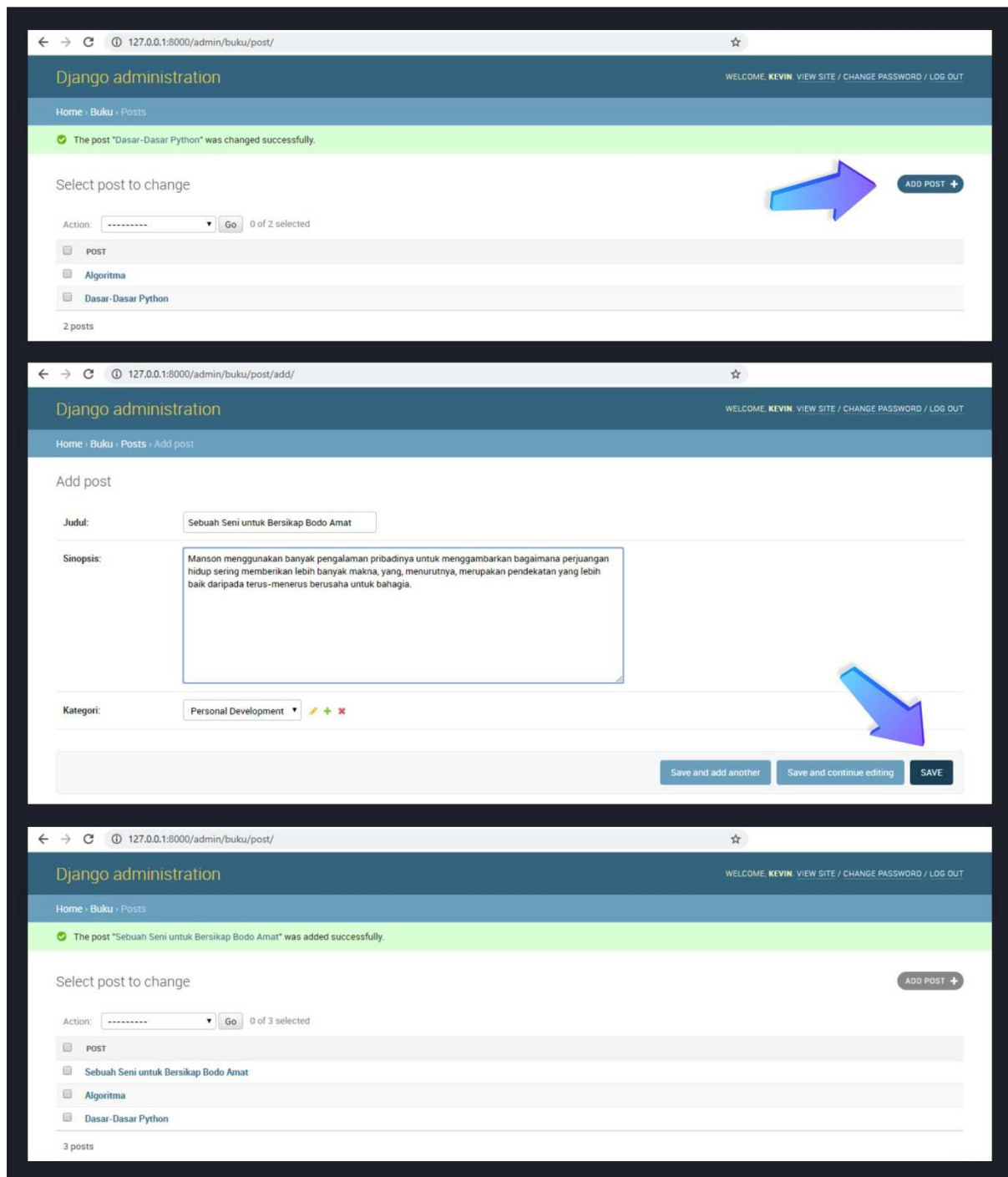
Gambar 171 Edit kategori buku

17. Lakukan hal yang sama untuk buku/object yang lain.



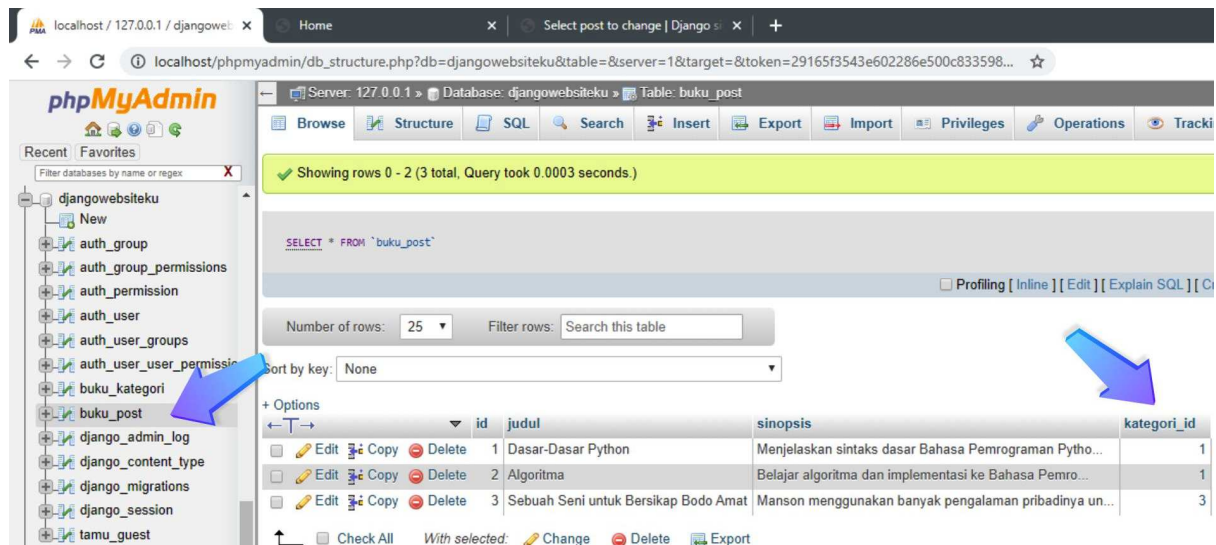
Gambar 172 Edit kategori buku kedua

18. Tambah **buku lain** dengan **kategori lain**. Contoh di bawah dengan kategori **Personal Development**.



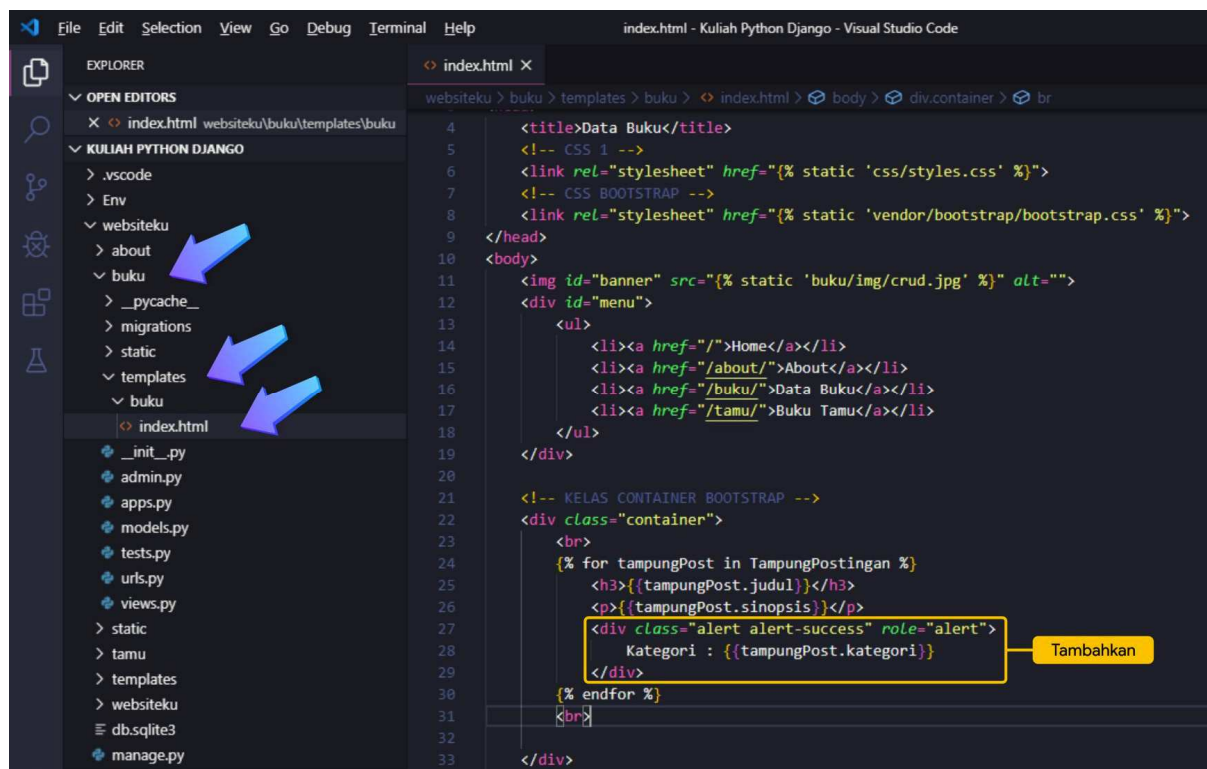
Gambar 173 Tambah buku lain

19. Cek database untuk melihat **Foreign Key** sudah terisi.



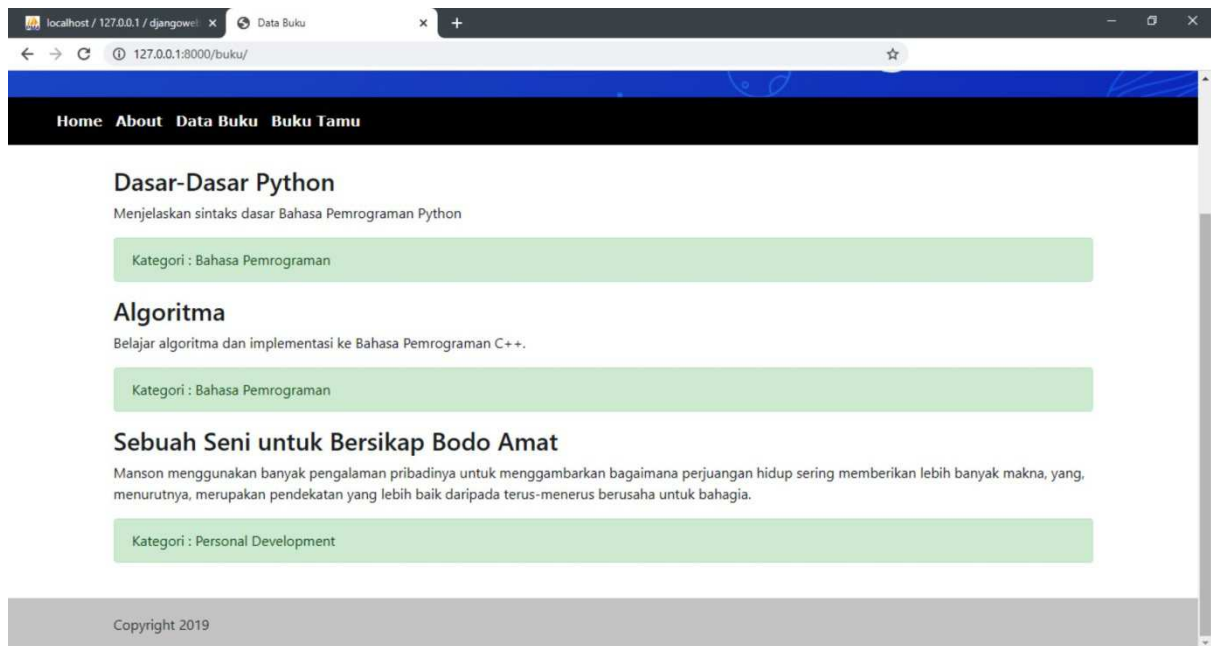
Gambar 174 Cek Foreign Key

20. Buka **index.html** pada **templates buku**. Lakukan penambahan kode seperti gambar di bawah.



Gambar 175 index.html buku

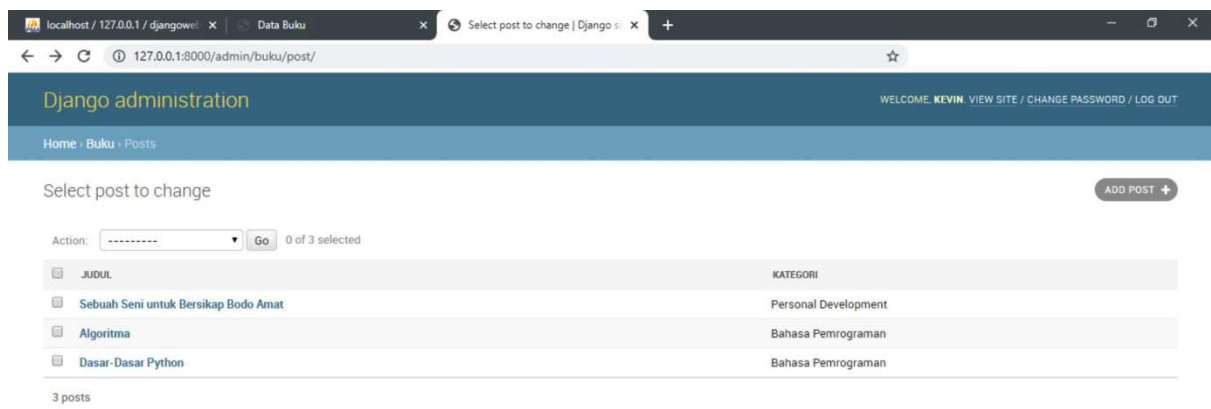
21. Tampilan Data Buku.



Gambar 176 Tampilan Data Buku

11.2 Tampilkan Relasi Model pada Halaman Admin (Latihan)

Buatlah tampilan Halaman Admin pada model buku berelasi dengan kategori yang telah disesuaikan. Contoh seperti gambar di bawah ini.



Gambar 177 Relasi pada Halaman Admin