TUGAS UTS PEMROGRAMAN

Hendra wijaya hadi

TI 6C

API IMSAKIYAH 1444 H

A. Keterangan atau Penjelasan Kode

Pada kode berikut ada beberapa fungsi yang dibuat diantaranya Create, Read, Update, Delete (**CRUD**) API Imsak menggunakan Django Rest Framework.

```
# menampilkan semua data
@api_view(['GET'])
def readJadwal(request):
    jadwalimsyak = JadwalModels.objects.all()
    serializer = jadwalSerializer(jadwalimsyak, many=True)
    return Response(serializer.data)
```

@api_view(['GET'])
Decorator ini hanya dapat menerima request dengan method **HTTP GET**

Pada fungsi **readJadwal(request)**, **Jadwal.objects.all()** untuk mengambil *semua data Jadwal yang tersimpan pada database*. Kemudian, data tersebut di-serialisasi menggunakan jadwalSerializer dengan parameter **many=True**, karena data yang diambil berupa *banyak* data. Selanjutnya, data yang sudah di-serialisasi dikembalikan sebagai HTTP response menggunakan Response(serializer.data).

```
# menampilkan 1 data atau detail
@api_view(['GET'])
def detailJadwal(request,id):
    jadwalimsyak = JadwalModels.objects.get(pk=id)
    serializer = jadwalSerializer(jadwalimsyak, many=False)
    return Response(serializer.data)
```

@api_view(['GET'])
Decorator ini hanya dapat menerima request dengan method **HTTP GET**

Pada fungsi **detailJadwal(request, id)**, parameter id digunakan untuk mencari data Jadwal berdasarkan id-nya yang diterima dari HTTP request. untuk mencari data Jadwal dengan id yang sama dengan id yang diterima menggunakan
Jadwal.objects.get(pk=id)

Setelah itu, data Jadwal yang telah ditemukan tersebut di-serialisasi menggunakan jadwalSerializer dengan parameter **many=False**, karena data yang diambil hanya berupa *satu data*. Kemudian, data yang sudah di-serialisasi dikembalikan sebagai HTTP response menggunakan **Response(serializer.data).**

```
# menambah/input data
@api_view(['POST'])
def createJadwal(request):
    serializer = jadwalSerializer(data=request.data)
    if serializer.is_valid():
        serializer.save()
    return Response(serializer.data)
```

@api_view(['POST'])
Decorator ini digunakan untuk memberi tanda bahwa view function yang didekorasi adalah sebuah view yang hanya dapat menerima request dengan method **HTTP POST**

def createJadwal(request):
fungsi createJadwal(request), data yang diterima dari HTTP request di-serialisasi menggunakan jadwalSerializer. Data yang diterima berupa data baru yang akan ditambahkan ke database. pada saat inisialisasi jadwalSerializer, digunakan data=request.data untuk mengambil data baru yang dikirimkan oleh client melalui HTTP request.
Setelah itu, dilakukan pengecekan apakah data yang diterima oleh **jadwalSerializer** adalah *valid atau tidak*, menggunakan method **is_valid().** Jika valid, data tersebut disimpan ke dalam database menggunakan method **save()**, yang secara otomatis akan memanggil create() pada jadwalSerializer.
Setelah data tersimpan kemudian akan dikirimkan kembali sebagai HTTP response menggunakan Response(serializer.data)

```
# mengedit/update data
@api_view(['PUT'])
def updateJadwal(request,id):
    jadwalimsyak = JadwalModels.objects.get(pk=id)
    serializer=jadwalSerializer(instance=jadwalimsyak, data=request.data)
    if serializer.is_valid():
        serializer.save()
    return Response(serializer.data)
```

@api_view(['PUT'])
Decorator ini hanya dapat menerima request dengan method **HTTP PUT**

Pada fungsi **updateJadwal(request, id),** data yang diterima dari HTTP request di-serialisasi menggunakan jadwalSerializer. pada saat inisialisasi jadwalSerializer, digunakan **instance=jadwalimsyak** untuk menentukan data yang akan **di-update**.
Data yang dikirimkan oleh client melalui HTTP request diambil menggunakan **data=request.data**. Setelah itu, dilakukan pengecekan apakah data yang diterima oleh jadwalSerializer adalah valid atau tidak. Jika valid, data tersebut disimpan ke dalam database

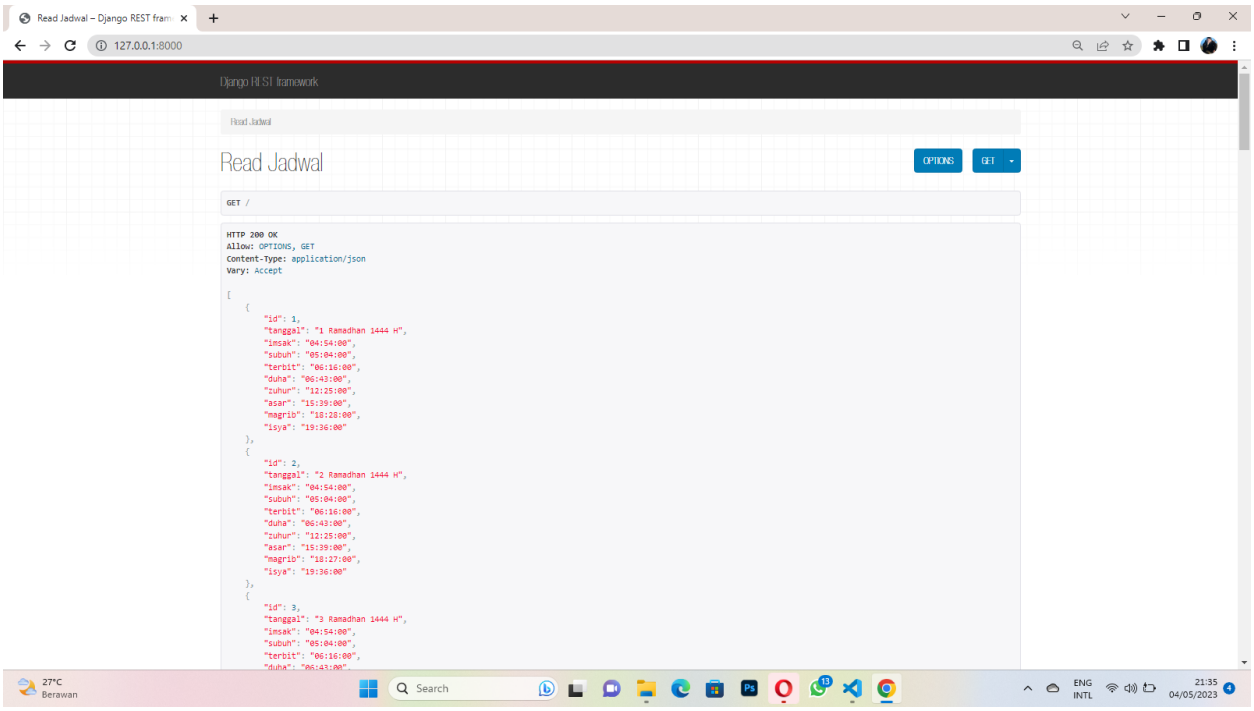
menggunakan method save(), yang secara otomatis akan memanggil update() pada jadwalSerializer.

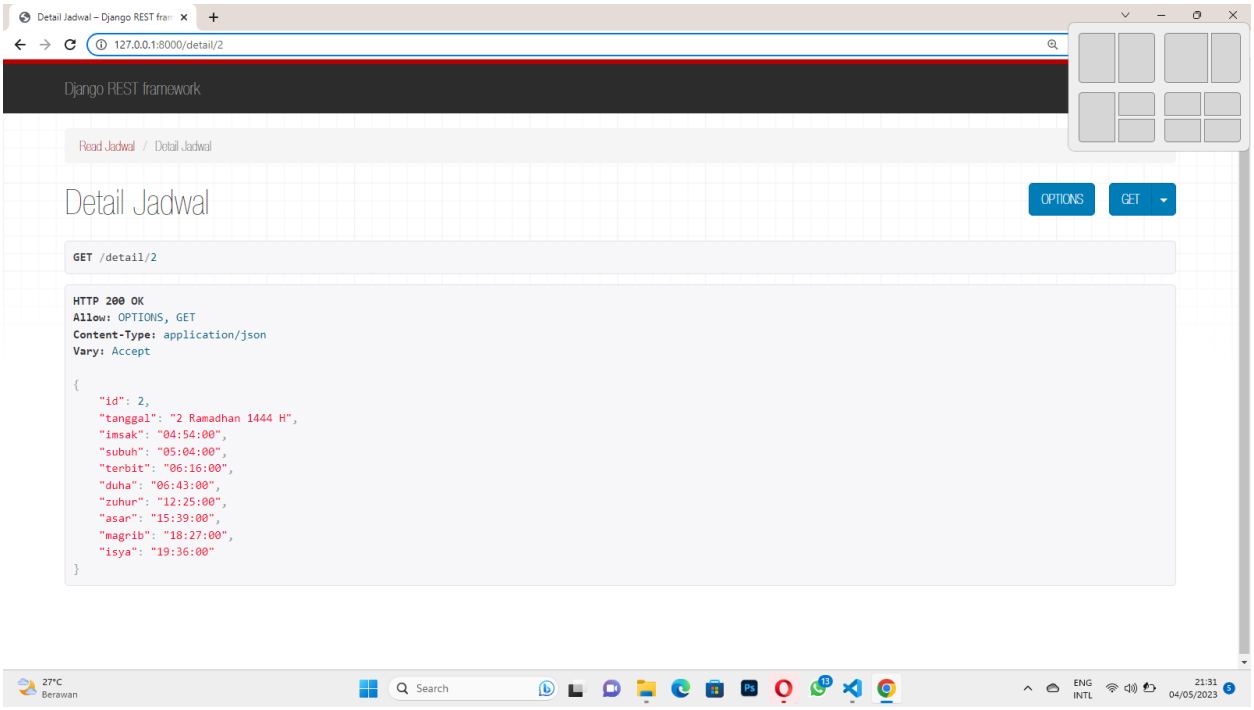
```
# meghapus/delet
@api_view(['DELETE'])
def deletJadwal(request,id):
    jadwalimsyak = JadwalModels.objects.get(pk=id)
    jadwalimsyak.delete()

    return Response('Data sudah dihilangkan')
```

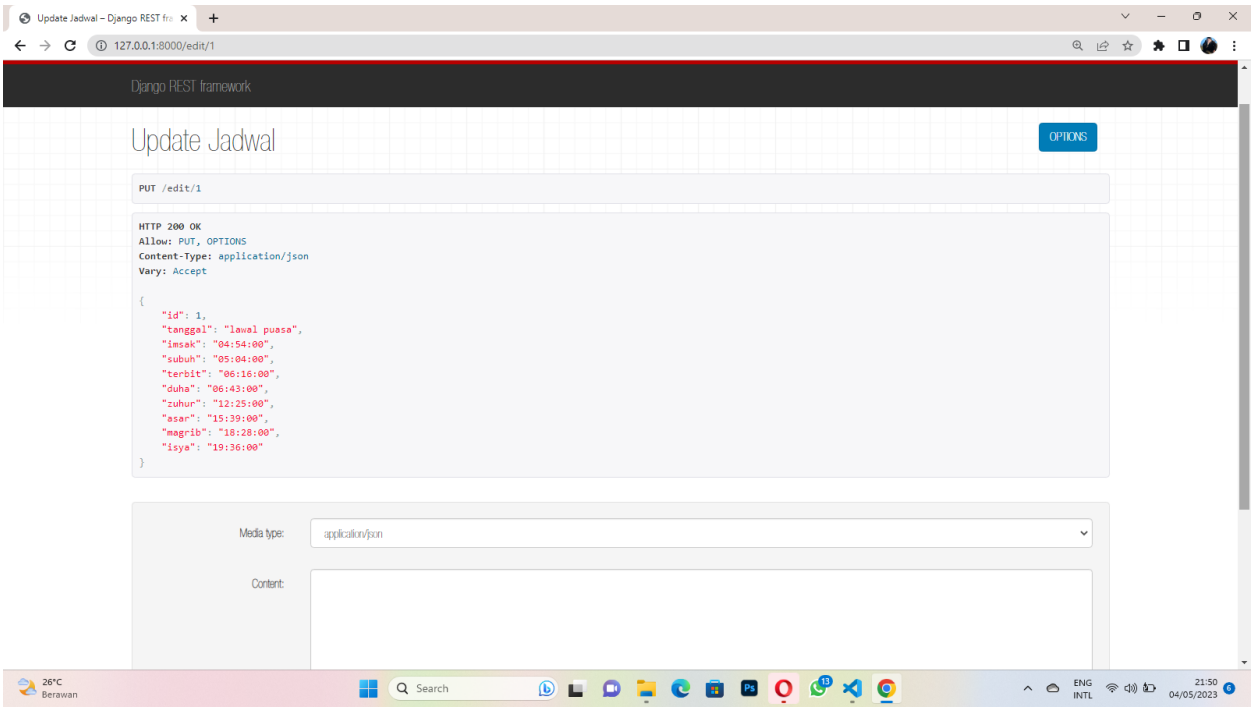
@api_view(['DELETE'])
Decorator ini hanya dapat menerima request dengan method **HTTP DELETE**

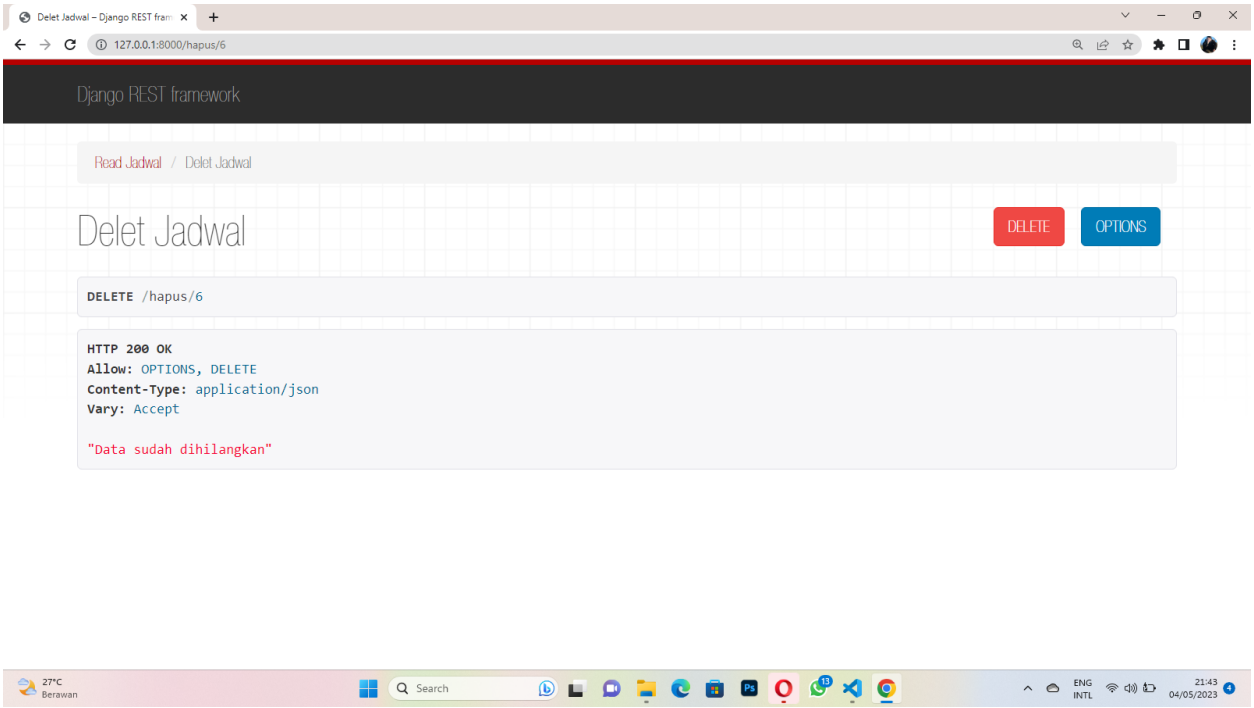
Pada fungsi **deletJadwal(request, id),** data yang akan dihapus diambil dari database menggunakan
Jadwal.objects.get(pk=id), dengan pk=id sebagai primary key dari data yang akan dihapus. Kemudian, data tersebut dihapus dari database menggunakan **method delete(),** yang akan menghapus data dari database sesuai dengan primary key yang diberikan.
Setelah data berhasil dihapus, fungsi akan **mengembalikan response** dengan pesan *'Data berhasil di Hapus'*.

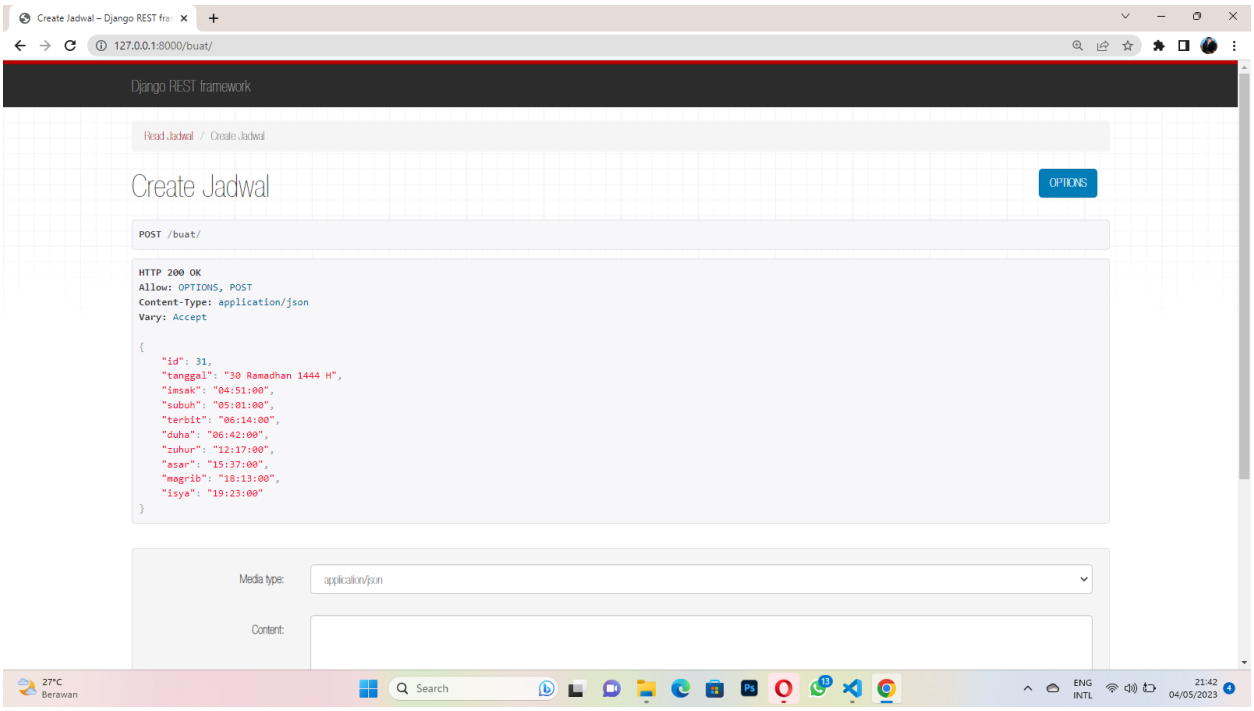
Respon API


Read Jadwal – Django REST fram
127.0.0.1:8000
Django REST framework
Read Jadwal
Read Jadwal
OPTIONS
GET
GET /
HTTP 200 OK
Allow: OPTIONS, GET
Content-Type: application/json
Vary: Accept
[
{
"id": 1,
"tanggal": "1 Ramadhan 1444 H",
"imsak": "04:54:00",
"subuh": "05:04:00",
"terbit": "06:16:00",
"duha": "06:43:00",
"zuhur": "12:25:00",
"asar": "15:39:00",
"magrib": "18:28:00",
"isya": "19:36:00"
},
{
"id": 2,
"tanggal": "2 Ramadhan 1444 H",
"imsak": "04:54:00",
"subuh": "05:04:00",
"terbit": "06:16:00",
"duha": "06:43:00",
"zuhur": "12:25:00",
"asar": "15:39:00",
"magrib": "18:27:00",
"isya": "19:36:00"
},
{
"id": 3,
"tanggal": "3 Ramadhan 1444 H",
"imsak": "04:54:00",
"subuh": "05:04:00",
"terbit": "06:16:00",
27°C
Berawan
Search
ENG
INTL
21:35
04/05/2023


Detail Jadwal – Django REST fram
127.0.0.1:8000/detail/2
Django REST framework
Read Jadwal / Detail Jadwal
Detail Jadwal
OPTIONS
GET
GET /detail/2
HTTP 200 OK
Allow: OPTIONS, GET
Content-Type: application/json
Vary: Accept
{
"id": 2,
"tanggal": "2 Ramadhan 1444 H",
"imsak": "04:54:00",
"subuh": "05:04:00",
"terbit": "06:16:00",
"duha": "06:43:00",
"zuhur": "12:25:00",
"asar": "15:39:00",
"magrib": "18:27:00",
"isya": "19:36:00"
}
27°C
Berawan
Search
ENG
INTL
21:31
04/05/2023


Update Jadwal – Django REST fra
127.0.0.1:8000/edit/1
Django REST framework
Update Jadwal
OPTIONS
PUT /edit/1
HTTP 200 OK
Allow: PUT, OPTIONS
Content-Type: application/json
Vary: Accept
{
"id": 1,
"tanggal": "lawal puasa",
"imsak": "04:54:00",
"subuh": "05:04:00",
"terbit": "06:16:00",
"duha": "06:43:00",
"zuhur": "12:25:00",
"asar": "15:39:00",
"magrib": "18:28:00",
"isya": "19:36:00"
}
Media type:
application/json
Content:
26°C
Berawan
Search
ENG
INTL
21:50
04/05/2023


Delet Jadwal – Django REST fram
127.0.0.1:8000/hapus/6
Django REST framework
Read Jadwal / Delet Jadwal
Delet Jadwal
DELETE
OPTIONS
DELETE /hapus/6
HTTP 200 OK
Allow: OPTIONS, DELETE
Content-Type: application/json
Vary: Accept
"Data sudah dihilangkan"
27°C
Berawan
Search
ENG
INTL
21:43
04/05/2023


Create Jadwal – Django REST fra
127.0.0.1:8000/buat/
Django REST framework
Read Jadwal / Create Jadwal
Create Jadwal
OPTIONS
POST /buat/
HTTP 200 OK
Allow: OPTIONS, POST
Content-Type: application/json
Vary: Accept
{
"id": 31,
"tanggal": "30 Ramadhan 1444 H",
"imsak": "04:51:00",
"subuh": "05:01:00",
"terbit": "06:14:00",
"duha": "06:42:00",
"zuhur": "12:17:00",
"asar": "15:37:00",
"magrib": "18:13:00",
"isya": "19:23:00"
}
Media type:
application/json
Content:
27°C
Berawan
Search
ENG
INTL
21:42
04/05/2023