

TUGAS UTS PEMROGRAMAN

API IMSAKIYAH 1444 H

A. Sekilas tentang Api Imsakiyah

Data waktu Imsak sangat penting bagi umat Muslim yang sedang menjalankan ibadah puasa selama bulan Ramadhan. Rest API Imsakiyah Ramadhan umumnya menyediakan informasi dalam format JSON yang dapat diakses melalui HTTP request. Informasi yang biasanya tersedia dalam Rest API Imsakiyah Ramadhan meliputi Tanggal, waktu Imsak, Subuh, Terbit, Duha, waktu Dzuhur, waktu Ashar, waktu Maghrib, dan waktu Isya' untuk setiap hari selama bulan Ramadhan. Pada pembuatan Rest Api ini menggunakan bahasa pemrograman Python dengan Framework Django.

B. Keterangan atau Penjelasan Kode

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

KODE	KETERANGAN
<pre># menambah/input data @api_view(['POST']) def createJadwal(request): serializer = jadwalSerializer(data=request.data) if serializer.is_valid(): serializer.save() return Response(serializer.data) # endpoint : http://127.0.0.1:8000/create/</pre>	<p>@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</p> <p>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 <i>valid atau tidak</i>, menggunakan method is_valid(). Jika valid, data tersebut disimpan ke dalam database menggunakan method save(), yang</p>

	<p>secara otomatis akan memanggil create() pada jadwalSerializer. Setelah data tersimpan kemudian akan dikirimkan kembali sebagai HTTP response menggunakan Response(serializer.data)</p>
<pre># menampilkan semua data @api_view(['GET']) def readJadwal(request): jadwalimsyak = Jadwal.objects.all() serializer = jadwalSerializer(jadwalimsyak, many=True) return Response(serializer.data) # endpoint: http://127.0.0.1:8000/</pre>	<p>@api_view(['GET']) Decorator ini hanya dapat menerima request dengan method HTTP GET</p> <p>Pada fungsi readJadwal(request), Jadwal.objects.all() untuk mengambil <i>semua data Jadwal yang tersimpan pada database</i>. Kemudian, data tersebut di-serialisasi menggunakan jadwalSerializer dengan parameter many=True, karena data yang diambil berupa <i>banyak data</i>. Selanjutnya, data yang sudah di-serialisasi dikembalikan sebagai HTTP response menggunakan Response(serializer.data).</p>
<pre># menampilkan 1 data atau detail @api_view(['GET']) def detailJadwal(request, id): jadwalimsyak = Jadwal.objects.get(pk=id) serializer = jadwalSerializer(jadwalimsyak, return Response(serializer.data) # endpoint: http://127.0.0.1:8000/detail/</pre>	<p>@api_view(['GET']) Decorator ini hanya dapat menerima request dengan method HTTP GET</p> <p>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 <i>satu data</i>. Kemudian, data yang sudah di-serialisasi dikembalikan sebagai HTTP response menggunakan Response(serializer.data).</p>

<pre># mengedit/update data @api_view(['PUT']) def updateJadwal(request,id): jadwalimsyak = Jadwal.objects.get(pk=id) serializer=JadwalSerializer(instance=jadwalimsyak,data=request.data) if serializer.is_valid(): serializer.save() return Response(serializer.data) # endpoint : http://127.0.0.1:8000/update/ path('delete/<int:id>',views.deletJadwal),</pre>	<p>@api_view(['PUT']) Decorator ini hanya dapat menerima request dengan method HTTP PUT</p> <p>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.</p>
<pre># meghapus/delet @api_view(['DELETE']) def deletJadwal(request,id): jadwalimsyak = Jadwal.objects.get(pk=id) jadwalimsyak.delete() return Response('Data berhasil di Hapus') #endpoint : http://127.0.0.1:8000/delete/</pre>	<p>@api_view(['DELETE']) Decorator ini hanya dapat menerima request dengan method HTTP DELETE</p> <p>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 <i>'Data berhasil di Hapus'</i>.</p>

Respon API

localhost: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": "06:16:42",
    "duha": "06:44:00",
    "zuhur": "12:26:00",
    "asar": "15:39:00",
    "magrib": "18:28:00",
    "isya": "19:37:00"
  },
  {
    "id": 2,
    "tanggal": "2 RAMADHAN 1444 H",
    "imsak": "04:54:00",
    "subuh": "05:04:00",
    "terbit": "06:16:42",
    "duha": "06:44:00",
    "zuhur": "12:26:00",
    "asar": "15:39:00",
    "magrib": "18:28:00",
    "isya": "19:37:00"
  }
]
```

localhost:8000/detail/1

Django REST framework

Read Jadwal / Detail Jadwal

Detail Jadwal

OPTIONS GET

GET /detail/1

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": "06:16:42",
  "duha": "06:44:00",
  "zuhur": "12:26:00",
  "asar": "15:39:00",
  "magrib": "18:28:00",
  "isya": "19:37:00"
}
```

← ↻ ⓘ localhost:8000/create/ 🔍 🏠 ⚙️ ⭐ 🗂️ 👤 ⋮

Django REST framework

[Read Jadwal](#) / Create Jadwal

Create Jadwal

OPTIONS

POST /create/

HTTP 200 OK
Allow: OPTIONS, POST
Content-Type: application/json
Vary: Accept

```
{
  "id": 30,
  "tanggal": "30 RAMADHAN 1444 H",
  "imsak": "04:54:00",
  "subuh": "05:04:00",
  "terbit": "06:16:42",
  "duha": "06:44:00",
  "zuhur": "12:26:00",
  "asar": "15:39:00",
  "magrib": "18:28:00",
  "isya": "19:37:00"
}
```

Media type:

← ↻ ⓘ localhost:8000/delete/31 🔍 🏠 ⚙️ ⭐ 🗂️ 👤 ⋮

Django REST framework

[Read Jadwal](#) / Delet Jadwal

Delet Jadwal

DELETE /delete/31

DELETE OPTIONS

HTTP 200 OK
Allow: OPTIONS, DELETE
Content-Type: application/json
Vary: Accept

"Data berhasil di Hapus"

←↻🔍localhost:8000/update/1

🔍🔖⚙️🔖🔖👤⋮

Django REST framework

Read Jadwal / Update Jadwal

Update Jadwal

OPTIONS

PUT /update/1

HTTP 200 OK
Allow: OPTIONS, PUT
Content-Type: application/json
Vary: Accept

```
{
  "id": 1,
  "tanggal": "1 RAMADHANssss 1444 H",
  "imsak": "04:54:00",
  "subuh": "05:04:00",
  "terbit": "06:16:42",
  "duha": "06:44:00",
  "zuhur": "12:26:00",
  "asar": "15:39:00",
  "magrib": "18:28:00",
  "isya": "19:37:00"
}
```

Media type:

application/json