

5/28/2024

# Modul Praktikum

## Pemrograman Mobile

### **Pertemuan 10**

Pokok Bahasan :

RESTful API menggunakan Codeigniter 4

### **Kebutuhan**

Komputer / Laptop

Visual Studio Code

XAMPP

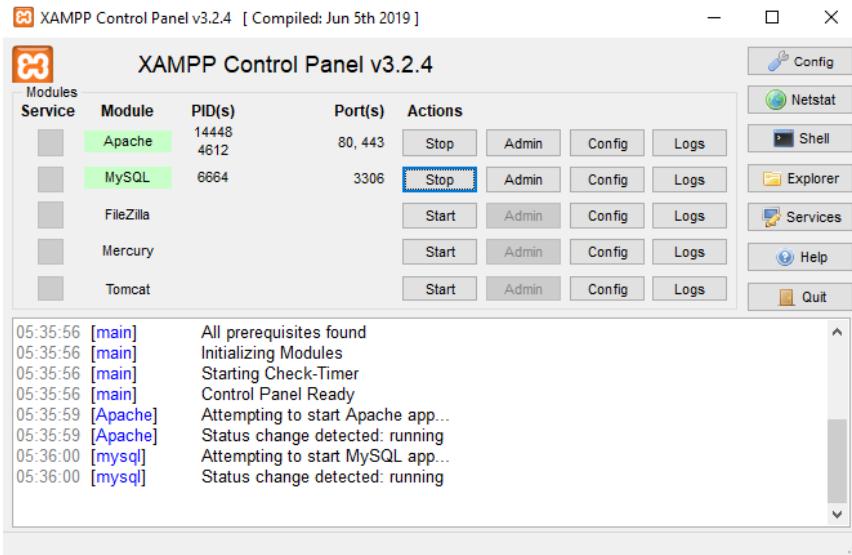
Composer

Postman

Codeigniter 4

Browser (Google Chrome)

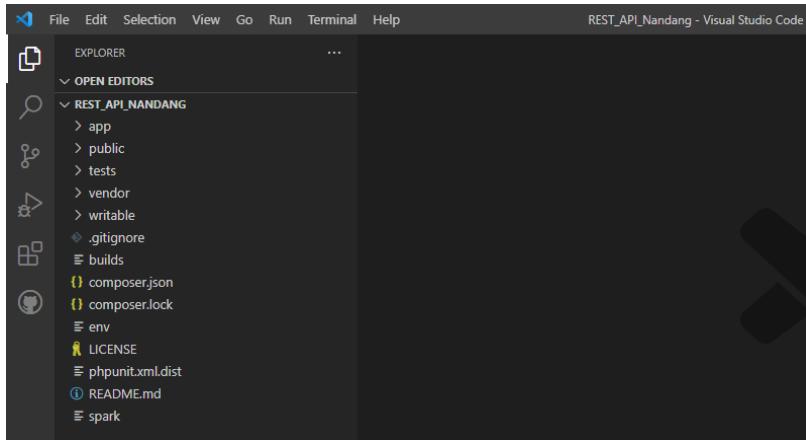
1. Pastikan XAMPP, Composer dan Postman sudah terinstall di computer anda
2. Aktifkan MySQL dan Apache



3. Install CodeIgniter dengan cara :
  - Jalankan Command Prompt
  - Pindahkan ke directory **C:\xampp\htdocs** dengan cara **cd\xampp\htdocs**
  - **composer create-project codeigniter4/appstarter namaproject**

```
C:\xampp\htdocs>composer create-project codeigniter4/appstarter REST_API_Nandang
Creating a "codeigniter4/appstarter" project at "./REST_API_Nandang"
Info from https://repo.packagist.org: #StandWithUkraine
Installing codeigniter4/appstarter (v4.1.9)
- Installing codeigniter4/appstarter (v4.1.9): Extracting archive
Created project in C:\xampp\htdocs\REST_API_Nandang
Loading composer repositories with package information
Info from https://repo.packagist.org: #StandWithUkraine
Updating dependencies
Lock file operations: 42 installs, 0 updates, 0 removals
- Locking codeigniter4/framework (v4.1.9)
- Locking doctrine/instantiator (1.4.1)
- Locking fakerphp/faker (v1.19.0)
- Locking kint-php/kint (4.1.2)
- Locking laminas/laminas-escaper (2.10.0)
```

4. Buka Folder project yang sudah dibuat menggunakan Visual Studio Code dengan cara **Open - Folder**



5. Buat database dengan cara buka phpMyAdmin di browser (<http://localhost/phpmyadmin/>)

## Databases



6. AturKonfigurasi

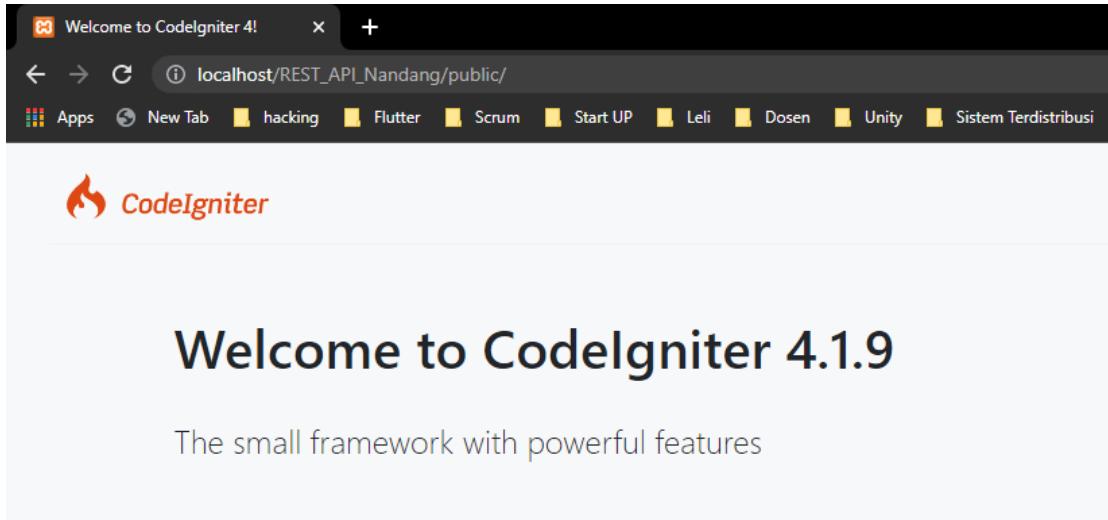
- rename file env menjadi .env
- buka file .env Kemudian ubah CI\_ENVIRONMENT menjadi development

**CI\_ENVIRONMENT = development**

- buka file .env Kemudian atur konfigurasi database

```
database.default.hostname = localhost
database.default.database = db_restapi_nandang
database.default.username = root
database.default.password =
database.default.DBDriver = MySQLi
database.default.DBPrefix =
```

7. Uji Rest API di browser



## About this page

The page you are looking at is being generated dynamically by CodeIgniter.

If you would like to edit this page you will find it located at:

8. Buat database migration dengan cara buka terminal visual studio code kemudian perintahkan  
`php spark migrate:create namamigrationdatabase`

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\xampp\htdocs\REST_API_Nandang> php spark migrate:create Database

CodeIgniter v4.1.9 Command Line Tool - Server Time: 2022-05-16 18:16:25 UTC-05:00

File created: APPPATH\Database\Migrations\2022-05-16-231625_Database.php

PS C:\xampp\htdocs\REST_API_Nandang>
```

9. Buka file **2022-05-16-231625\_Database.php** yang terdapat di **Database-Migrations** kemudian lengkapi isinya seperti dibawah ini

Kode Program

```

1 <?php
2 namespace App\Database\Migrations;
3 use CodeIgniter\Database\Migration;
4 class Pendaftaran extends Migration
5 {
6     public function up()
7     {
8         $this->forge->addField([
9             'id' => [
10                 'type' => 'INT',
11                 'auto_increment' => true,
12             ],
13             'nama' => [
14                 'type' => 'varchar',
15                 'constraint' => 100,
16             ],
17             'email' => [
18                 'type' => 'varchar',
19                 'constraint' => 100,
20             ],
21             'no_telp' => [
22                 'type' => 'varchar',
23                 'constraint' => 20,
24             ],
25             'jenis_kelamin' => [
26                 'type' => 'varchar',
27                 'constraint' => 10,
28             ],
29             'bahasa' => [
30                 'type' => 'varchar',
31                 'constraint' => 1024,
32             ],
33             'agama' => [
34                 'type' => 'varchar',
35                 'constraint' => 20,
36             ],
37             'tanggal_daftar' => [
38                 'type' => 'date',
39                 'null' => false,
40             ],
41             'jam_daftar' => [
42                 'type' => 'time',
43                 'null' => false,
44             ],
45         ]);
46         $this->forge->addKey('id', true);
47         $this->forge->createTable('pendaftaran');
48     }
49     public function down()
50     {
51         $this->forge->dropTable('pendaftaran');
52     }
53 }
54

```

10. Jalankan migration dengan cara buka terminal visual studio code kemudian perintahkan

```
php spark migrate
```

```

Running all new migrations...
    Running: (App) 2022-05-16-231625_App\Database\Migrations\Database
Done migrations.
PS C:\xampp\htdocs\REST_API_Nandang> █

```

11. Buat seeder untuk mengisi data dummy dengan cara buka terminal visual studio code kemudian perintahkan `php spark make:seeder Pendaftaran`
12. Buka file **Pendaftaran.php** yang terdapat di **Database-Seeder** kemudian lengkapi isinya seperti dibawah ini

#### Kode Program

app\Database\Seeds\Pendaftaran.php

```
1 <?php
2
3 namespace App\Database\Seeds;
4
5 use CodeIgniter\Database\Seeder;
6
7 class Pendaftaran extends Seeder
8 {
9     public function run()
10    {
11        $data = [
12            [
13                'nama' => 'Nandang',
14                'email' => 'nandang@gmail.com',
15                'no_telp' => '08961',
16                'jenis_kelamin' => 'Pria',
17                'bahasa' => 'Indonesia,Inggris,Arab,Sunda,Jawa',
18                'agama' => 'Islam',
19                'tanggal_daftar' => '2024-01-01',
20                'jam_daftar' => '08:30:45',
21            ],
22            [
23                'nama' => 'Hermanto',
24                'email' => 'hermanto@gmail.com',
25                'no_telp' => '08962',
26                'jenis_kelamin' => 'Pria',
27                'bahasa' => 'Indonesia,Inggris,Arab',
28                'agama' => 'Islam',
29                'tanggal_daftar' => '2024-02-21',
30                'jam_daftar' => '09:30:45',
31            ],
32            [
33                'nama' => 'Nandang Hermanto',
34                'email' => 'nandanghermanto@gmail.com',
35                'no_telp' => '08963',
36                'jenis_kelamin' => 'Pria',
37                'bahasa' => 'Indonesia,Inggris,Arab,Sunda',
38                'agama' => 'Islam',
39                'tanggal_daftar' => '2024-03-04',
40                'jam_daftar' => '10:35:45',
41            ],
42        ];
43
44        $this->db->table('pendaftaran')->insertBatch($data);
45    }
46 }
47 }
```

13. Jalankan seeder dengan cara buka terminal visual studio code kemudian perintahkan `php spark db:seed Pendaftaran`

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\xampp\htdocs\REST_API_Nandang> php spark db:seed Users

CodeIgniter v4.1.9 Command Line Tool - Server Time: 2022-05-16 18:49:01 UTC-05:00

Seeded: App\Database\Seeds\Users
PS C:\xampp\htdocs\REST_API_Nandang> █
```

14. Buat Model dengan cara buka terminal visual studio code kemudian perintahkan  
`php spark make:model PendaftaranModel -suffix`
15. Buka file **PendaftaranModel.php** yang terdapat di **Models** kemudian lengkapi isinya seperti dibawah ini

**Kode Program**

```
1 | <?php
2 |
3 | namespace App\Models;
4 |
5 | use CodeIgniter\Model;
6 |
7 | class PendaftaranModel extends Model
8 |
9 | {
10 |     protected $DBGroup      = 'default';
11 |     protected $table         = 'pendaftaran';
12 |     protected $primaryKey    = 'id';
13 |     protected $useAutoIncrement = true;
14 |     protected $returnType   = 'App\Entities\Pending';
15 |     protected $useSoftDeletes = false;
16 |     protected $protectFields = true;
17 |     protected $allowedFields = [
18 |         'nama',
19 |         'email',
20 |         'no_telp',
21 |         'jenis_kelamin',
22 |         'bahasa',
23 |         'agama',
24 |         'tanggal_daftar',
25 |         'jam_daftar',
26 |     ];
27 |
28 |     // Dates
29 |     protected $useTimestamps = false;
30 |     protected $dateFormat    = 'datetime';
31 |     protected $createdField  = 'created_at';
32 |     protected $updatedField  = 'updated_at';
33 |     protected $deletedField  = 'deleted_at';
34 |
35 |     // Validation
36 |     protected $validationRules = [
37 |         'nama' => 'required|min_length[5]|is_unique[pendaftaran.nama]',
38 |         'email' => 'required|valid_email',
39 |         'no_telp' => 'required',
40 |         'jenis_kelamin' => 'required',
41 |         'bahasa' => 'required',
42 |         'agama' => 'required',
43 |         'tanggal_daftar' => 'required',
44 |         'jam_daftar' => 'required',
45 |     ];
46 |
47 |     protected $validationMessages = [
48 |         'nama' => [
49 |             'required' => 'silakan masukan nama',
50 |             'min_length' => 'nama minimal 5 karakter',
51 |             'is_unique' => 'nama sudah terdaftar silakan masukan nama yang lain',
52 |         ],
53 |         'email' => [
54 |             'required' => 'silakan masukan email',
55 |             'valid_email' => 'silakan masukan email yang valid',
56 |         ],
57 |         'no_telp' => [
58 |             'required' => 'silakan masukan no_telp',
59 |         ],
60 |     ];
61 |
62 | }
```

```

58     ],
59     'jenis_kelamin' => [
60         'required' => 'silakan masukan jenis_kelamin',
61     ],
62     'bahasa' => [
63         'required' => 'silakan masukan bahasa',
64     ],
65     'agama' => [
66         'required' => 'silakan masukan agama',
67     ],
68     'tanggal_daftar' => [
69         'required' => 'silakan masukan tanggal_daftar',
70     ],
71     'jam_daftar' => [
72         'required' => 'silakan masukan jam_daftar',
73     ],
74 ];
75 protected $skipValidation      = false;
76 protected $cleanValidationRules = true;
77
78 // Callbacks
79 protected $allowCallbacks = true;
80 protected $beforeInsert   = [];
81 protected $afterInsert    = [];
82 protected $beforeUpdate   = [];
83 protected $afterUpdate    = [];
84 protected $beforeFind    = [];
85 protected $afterFind     = [];
86 protected $beforeDelete  = [];
87 protected $afterDelete   = [];
88 }
89

```

16. Buat Entity dengan cara buka terminal visual studio code kemudian perintahkan

`php spark make:entity Pendaftaran`

17. Buka file Entity **Pendaftaran.php** yang terdapat di **Entities** kemudian lengkapi isinya seperti dibawah

**Kode Program**

```

1 <?php
2
3 namespace App\Entities;
4
5 use CodeIgniter\Entity\Entity;
6
7 class Pendaftaran extends Entity
8 {
9     protected $datamap = [];
10    protected $dates   = ['created_at', 'updated_at', 'deleted_at'];
11    protected $casts   = [];
12 }
13

```

18. Buat ControllerUser dengan cara buka terminal visual studio code kemudian perintahkan

`php spark make:controller Pendaftaran --restful`

19. Buka file Controller **Pendaftaran.php** yang terdapat di **Controller** kemudian lengkapi isinya seperti dibawah ini

**Kode Program**

```

1 | <?php
2 | namespace App\Controllers;
3 | use CodeIgniter\HTTP\ResponseInterface;
4 | use CodeIgniter\RESTful\ResourceController;
5 | class Pendaftaran extends ResourceController
6 | {
7 |     protected $modelName = 'App\Models\PendaftaranModel';
8 |     protected $format = 'json';
9 |     public function index()
10 |    {
11 |        return $this->respond($this->model->findAll());
12 |    }
13 |    public function show($id = null)
14 |    {
15 |        if (!$this->model->find($id)) {
16 |            return $this->fail('Data tidak ditemukan');
17 |        }
18 |        return $this->respond($this->model->find($id));
19 |    }
20 |    public function create()
21 |    {
22 |        $data = $this->request->getPost();
23 |        $spendaftaran = new \App\Entities\Pendaftaran();
24 |        $spendaftaran->fill($data);
25 |        if (!$this->validate($this->model->validationRules, $this->model->validationMessages))
26 |        {
27 |            return $this->fail($this->validator->getErrors());
28 |        }
29 |        if ($this->model->save($spendaftaran)) {
30 |            return $this->respondCreated($data);
31 |        }
32 |    }
33 |    public function update($id = null)
34 |    {
35 |        $data = $this->request->getRawInput();
36 |        $data['id'] = $id;
37 |        if (!$this->model->find($id)) {
38 |            return $this->fail('Data tidak ditemukan');
39 |        }
40 |        $spendaftaran = new \App\Entities\Pendaftaran();
41 |        $spendaftaran->fill($data);
42 |        if (!$this->validate($this->model->validationRules, $this->model->validationMessages))
43 |        {
44 |            return $this->fail($this->validator->getErrors());
45 |        }
46 |        if ($this->model->save($spendaftaran)) {
47 |            return $this->respondUpdated($data);
48 |        }
49 |    }
50 |    public function delete($id = null)
51 |    {
52 |        if (!$this->model->find($id)) {
53 |            return $this->fail('Data tidak ditemukan');
54 |        }
55 |        if ($this->model->delete($id)) {
56 |            return $this->respondDeleted("Data dengan id " . $id . " berhasil dihapus");
57 |        }
58 |    }
59 | }

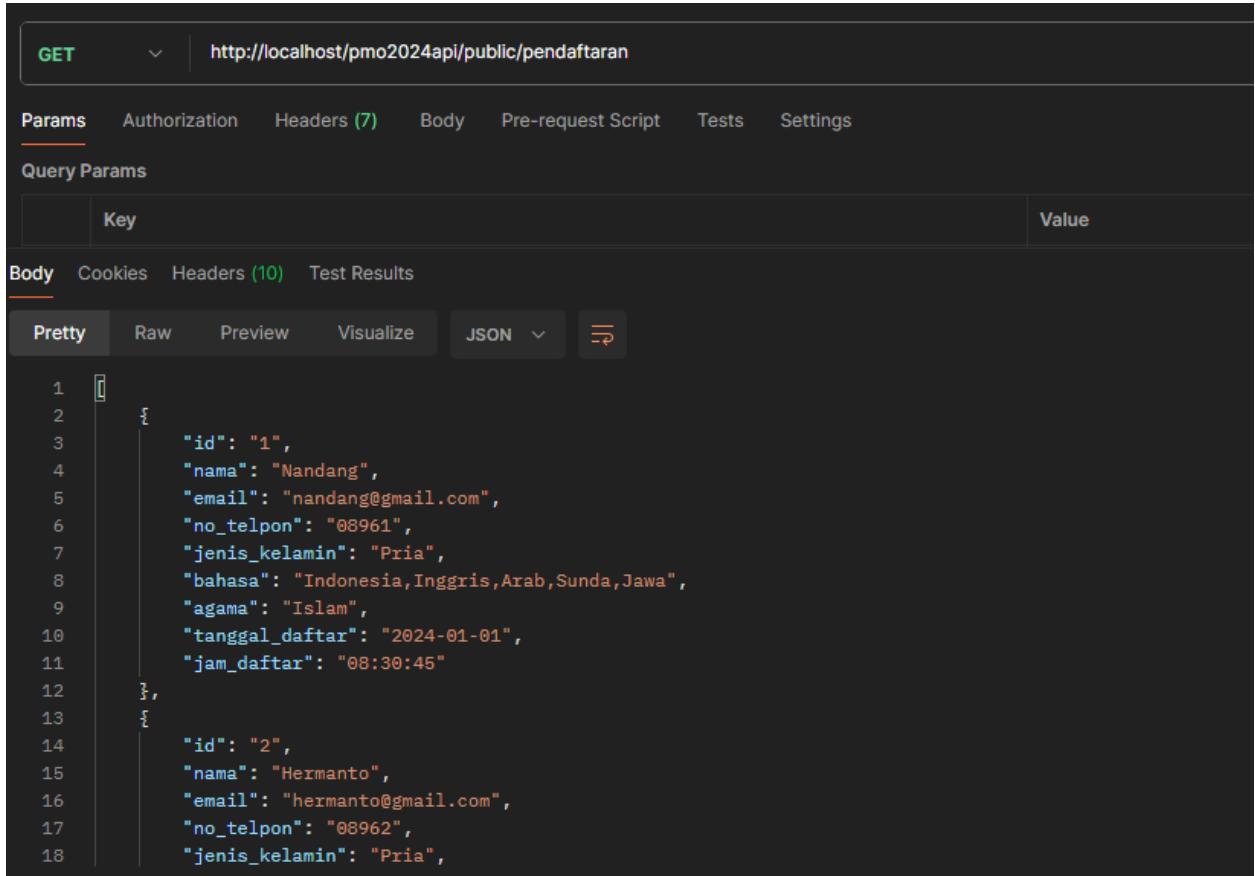
```

20. Atur Route dengan cara buka **Routes.php** yang terdapat di **app\Config** kemudian tambahkan  
`$routes->resource('pendaftaran');`

```
1 <?php
2
3 use CodeIgniter\Router\RouteCollection;
4
5 /**
6  * @var RouteCollection $routes
7 */
8 $routes->get('/', 'Home::index');
9 $routes->resource('pendaftaran');
```

Catatan : untuk melihat semua route bisa gunakan `php spark routes` dari terminal

21. Buka Postman kemudian uji Proses GET



The screenshot shows a Postman interface with a GET request to `http://localhost/pmo2024api/public/pendaftaran`. The response body is displayed in Pretty mode as follows:

```
1 [
2   {
3     "id": "1",
4     "nama": "Nandang",
5     "email": "nandang@gmail.com",
6     "no_telp": "08961",
7     "jenis_kelamin": "Pria",
8     "bahasa": "Indonesia,Inggris,Arab,Sunda,Jawa",
9     "agama": "Islam",
10    "tanggal_daftar": "2024-01-01",
11    "jam_daftar": "08:30:45"
12  },
13  {
14    "id": "2",
15    "nama": "Hermanto",
16    "email": "hermanto@gmail.com",
17    "no_telp": "08962",
18    "jenis_kelamin": "Pria",
19  }
]
```

22. Buka Postman kemudian uji Proses POST

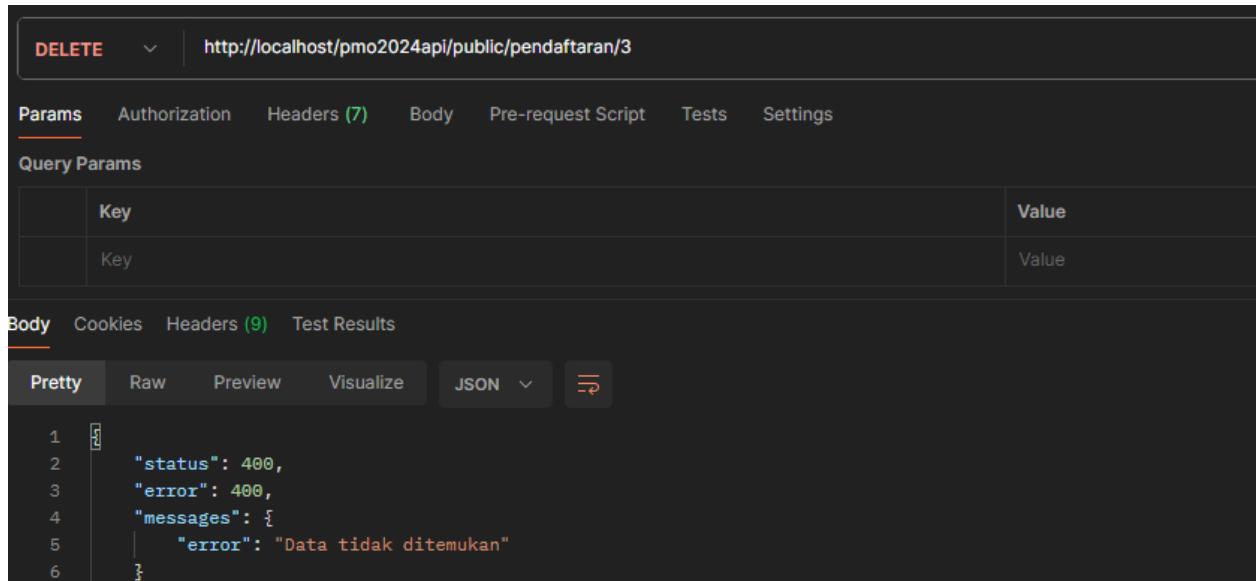
Key	Value
nama	Susi Susanti
email	susi@gmail.com
no_telp	987987
jenis_kelamin	Wanita
bahasa	Indonesia,Inggris
agama	Katolik
tanggal_daftar	2024-02-13
jam_daftar	20:10:30

Catatan : Uji Validation dengan menjalankan perintah post tanpa mengisi data, mengisi data dengan nama yang sama, mengisi data dengan nama kurang dari 5 karakter, mengisi data email yang tidak valid **pastikan semua validation berjalan dengan baik**

23. Buka Postman kemudian uji Proses PUT

Key	Value
nama	Susi Susanti editan
email	susi@gmail.com
no_telp	987987
jenis_kelamin	Wanita
bahasa	Indonesia,Inggris
agama	Katolik
tanggal_daftar	2024-02-13
jam_daftar	20:10:30

24. Buka Postman kemudian uji Proses DELETE



DELETE <http://localhost/pmo2024api/public/pendaftaran/3>

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

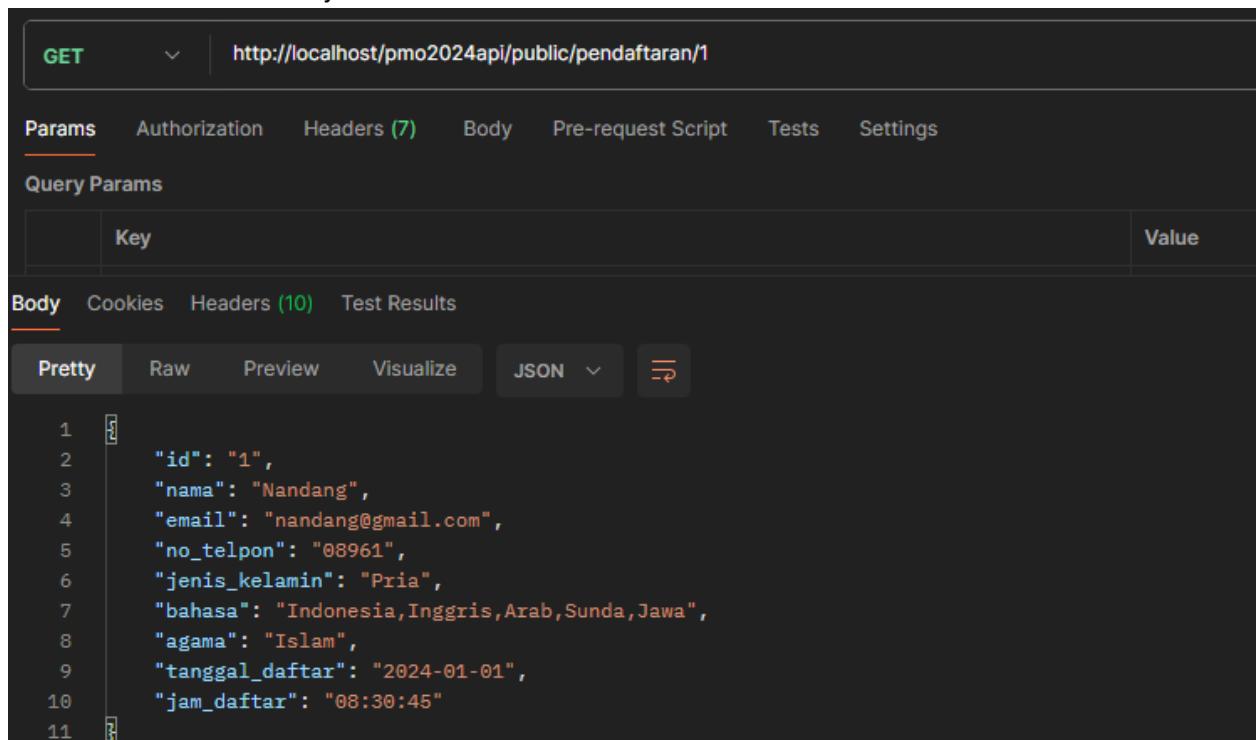
Key	Value
Key	Value

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON ↻

```
1
2     "status": 400,
3     "error": 400,
4     "messages": [
5         "error": "Data tidak ditemukan"
6     ]
```

25. Buka Postman kemudian uji Proses SHOW



GET <http://localhost/pmo2024api/public/pendaftaran/1>

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

Key	Value
-----	-------

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON ↻

```
1 {
2     "id": "1",
3     "nama": "Nandang",
4     "email": "nandang@gmail.com",
5     "no_telp": "08961",
6     "jenis_kelamin": "Pria",
7     "bahasa": "Indonesia,Inggris,Arab,Sunda,Jawa",
8     "agama": "Islam",
9     "tanggal_daftar": "2024-01-01",
10    "jam_daftar": "08:30:45"
11 }
```