LAPORAN PRAKTIKUM
MATA KULIAH PEMROGRAMAN MOBILE

UNIVERSITAS
AMIKOM PURWOKERTO

Oleh :

NAMA: FAJAR RAMADHAN

NIM: 23SA31A065

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER

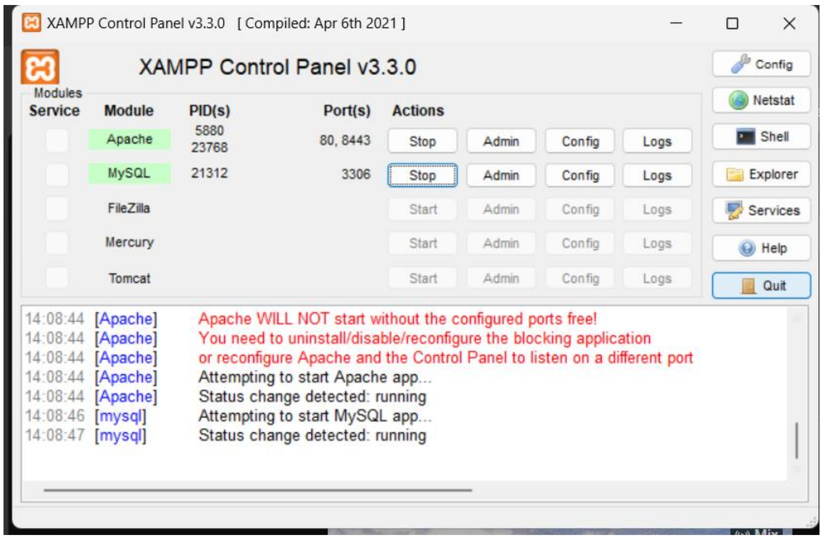
UNIVERSITAS AMIKOM

PURWOKERTO

2025

RESTFUL API MENGGUNAKAN CODEIGNITER 4

1. Start apache dan mysql melalui xampp

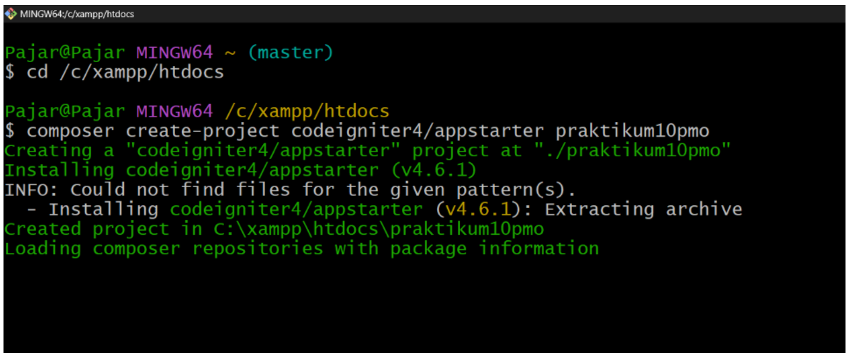


XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

Service	Module	PID(s)	Port(s)	Actions			
	Apache	5880 23768	80, 8443	Stop	Admin	Config	Logs
	MySQL	21312	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

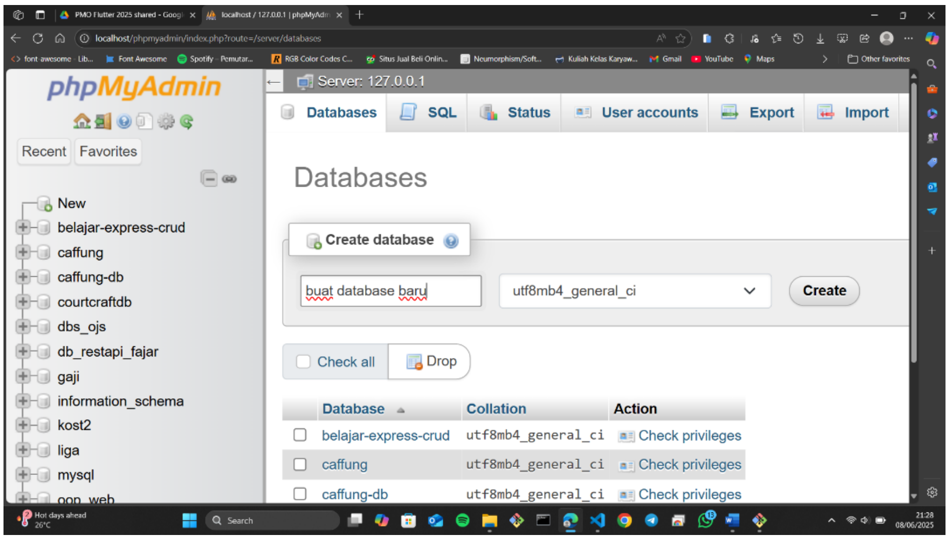
```
14:08:44  [Apache]   Apache WILL NOT start without the configured ports free!
14:08:44  [Apache]   You need to uninstall/disable/reconfigure the blocking application
14:08:44  [Apache]   or reconfigure Apache and the Control Panel to listen on a different port
14:08:44  [Apache]   Attempting to start Apache app...
14:08:44  [Apache]   Status change detected: running
14:08:46  [mysql]    Attempting to start MySQL app...
14:08:47  [mysql]    Status change detected: running
```

2. Menginstall codeignter:
   - Open cmd
   - Masuk ke directory “cd /c/xampp/htdocs”
   - Jalankan composer create-project codeigniter4/appstarter namaproject



```
MINGW64:/c/xampp/htdocs

Pajar@Pajar MINGW64 ~ (master)
$ cd /c/xampp/htdocs

Pajar@Pajar MINGW64 /c/xampp/htdocs
$ composer create-project codeigniter4/appstarter praktikum10pmo
Creating a "codeigniter4/appstarter" project at "./praktikum10pmo"
Installing codeigniter4/appstarter (v4.6.1)
INFO: Could not find files for the given pattern(s).
  - Installing codeigniter4/appstarter (v4.6.1): Extracting archive
Created project in C:\xampp\htdocs\praktikum10pmo
Loading composer repositories with package information
```

3. Buat database baru melalui phpmyadmin



Databases

Create database

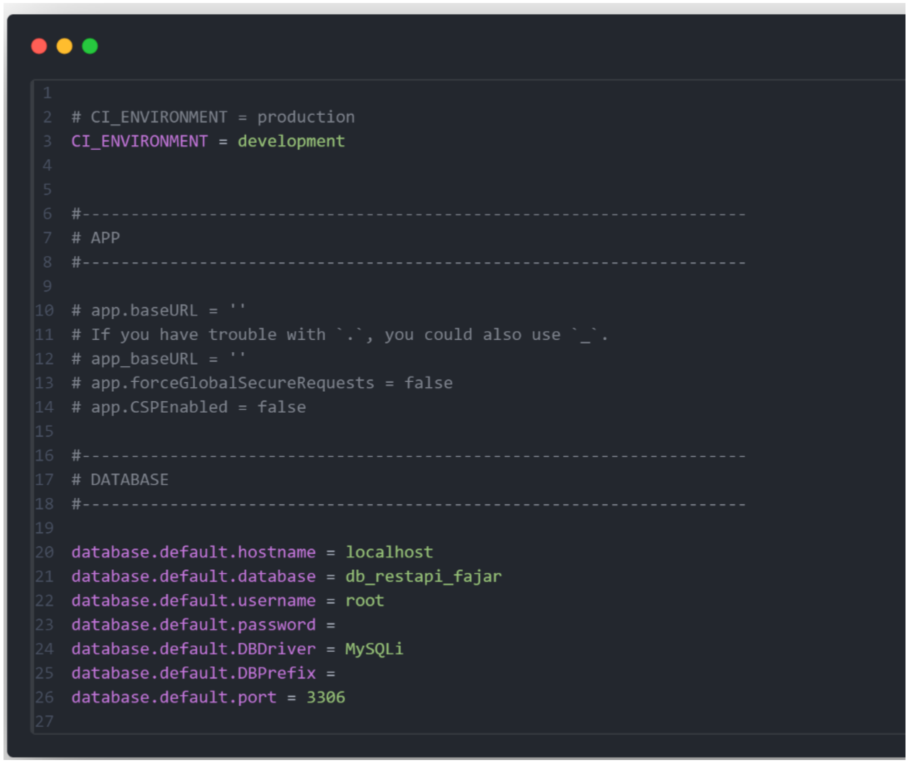
buat database baru — utf8mb4_general_ci — Create

Check all — Drop

New
belajar-express-crud
caffung
caffung-db
courtcraftdb
dbs_ojs
db_restapi_fajar
gaji
information_schema
kost2
liga
mysql
oop_web

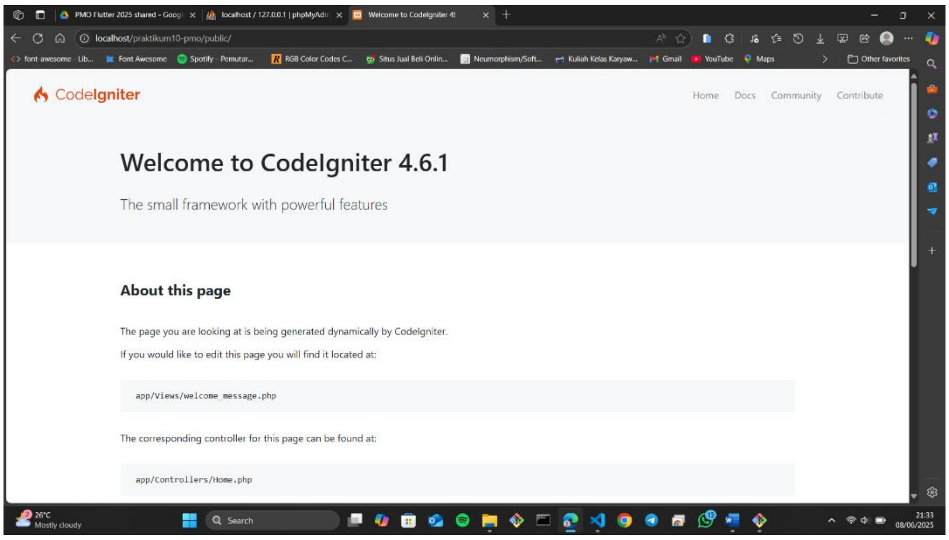
Database	Collation	Action
belajar-express-crud	utf8mb4_general_ci	Check privileges
caffung	utf8mb4_general_ci	Check privileges
caffung-db	utf8mb4_general_ci	Check privileges

4. Konfigurasi env dengan rename file env menjadi .env kemudian set enviroment menjadi devolopment dan atur konfigurasi database



```

# CI_ENVIRONMENT = production
CI_ENVIRONMENT = development


#--------------------------------------------------------------------
# APP
#--------------------------------------------------------------------

# app.baseURL = ''
# If you have trouble with `.`, you could also use `_`.
# app_baseURL = ''
# app.forceGlobalSecureRequests = false
# app.CSPEnabled = false

#--------------------------------------------------------------------
# DATABASE
#--------------------------------------------------------------------

database.default.hostname = localhost
database.default.database = db_restapi_fajar
database.default.username = root
database.default.password =
database.default.DBDriver = MySQLi
database.default.DBPrefix =
database.default.port = 3306

```

5. Menguji rest-api di browser dengan url localhost/namaprojek



PMO Flutter 2025 shared - Google
localhost / 127.0.0.1 | phpMyAdmin
Welcome to CodeIgniter 4!
localhost/praktikum10-pmo/public/
CodeIgniter
Home
Docs
Community
Contribute
26°C Mostly cloudy
Search
21:33 08/06/2025

Welcome to CodeIgniter 4.6.1

The small framework with powerful features

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:

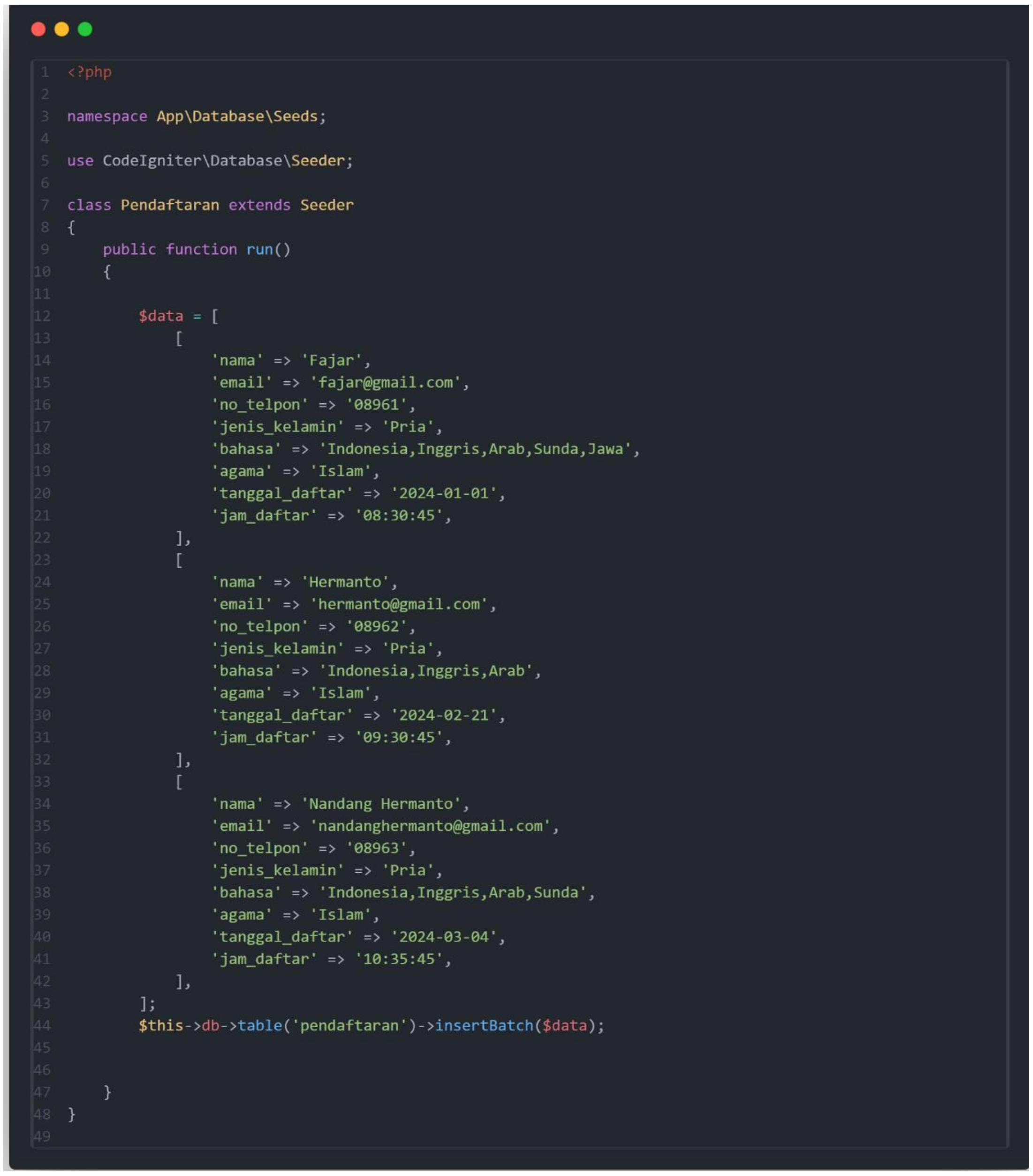
```
app/Views/welcome_message.php
```

The corresponding controller for this page can be found at:

```
app/Controllers/Home.php
```

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

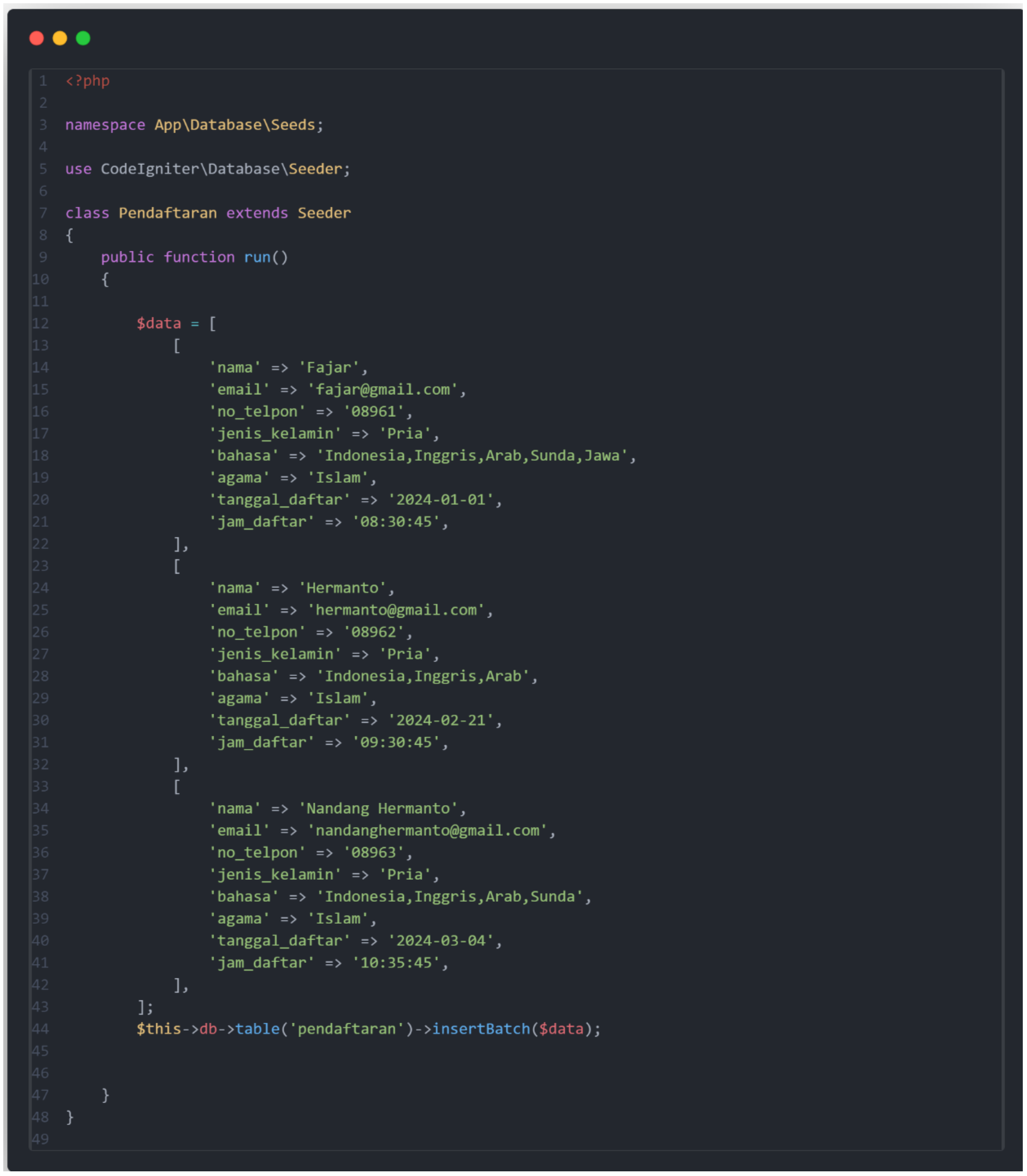
7. Buka file hasil migration pada folder migration dan ubah isi file



```php
<?php

namespace App\Database\Seeds;

use CodeIgniter\Database\Seeder;

class Pendaftaran extends Seeder
{
    public function run()
    {

        $data = [
            [
                'nama' => 'Fajar',
                'email' => 'fajar@gmail.com',
                'no_telpon' => '08961',
                'jenis_kelamin' => 'Pria',
                'bahasa' => 'Indonesia,Inggris,Arab,Sunda,Jawa',
                'agama' => 'Islam',
                'tanggal_daftar' => '2024-01-01',
                'jam_daftar' => '08:30:45',
            ],
            [
                'nama' => 'Hermanto',
                'email' => 'hermanto@gmail.com',
                'no_telpon' => '08962',
                'jenis_kelamin' => 'Pria',
                'bahasa' => 'Indonesia,Inggris,Arab',
                'agama' => 'Islam',
                'tanggal_daftar' => '2024-02-21',
                'jam_daftar' => '09:30:45',
            ],
            [
                'nama' => 'Nandang Hermanto',
                'email' => 'nandanghermanto@gmail.com',
                'no_telpon' => '08963',
                'jenis_kelamin' => 'Pria',
                'bahasa' => 'Indonesia,Inggris,Arab,Sunda',
                'agama' => 'Islam',
                'tanggal_daftar' => '2024-03-04',
                'jam_daftar' => '10:35:45',
            ],
        ];
        $this->db->table('pendaftaran')->insertBatch($data);


    }
}

```

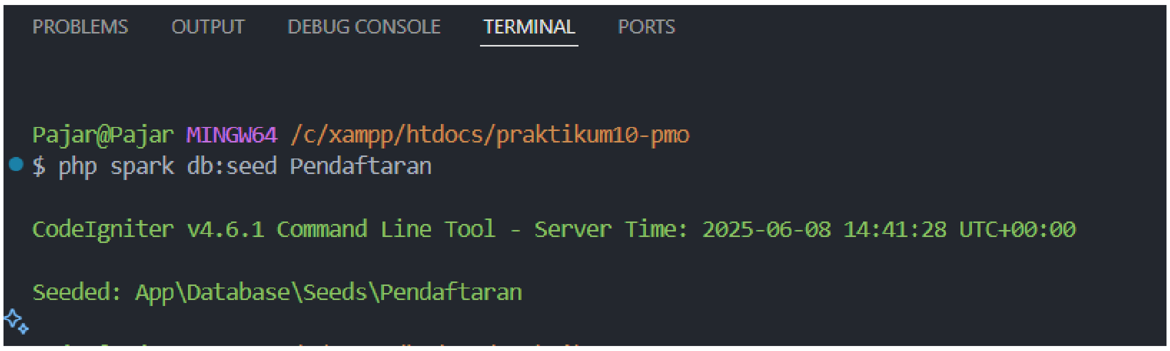
8. Jalankan migration dengan cara buka terminal visual studio code kemudian perintahkan ”php spark migrate”
9. Buat seeder untuk mengisi data dummy dengan cara buka terminal visual studio code kemudian perintahkan ”php spark make:seeder Pendaftaran”

10. Buka file Pendaftaran.php yang terdapat di Database-Seeder kemudian lengkapi isinya atau bisa dibilang buat data dummy dalam bentuk object of array sesuai dengan struktur database yang telah dibuat pada database migration



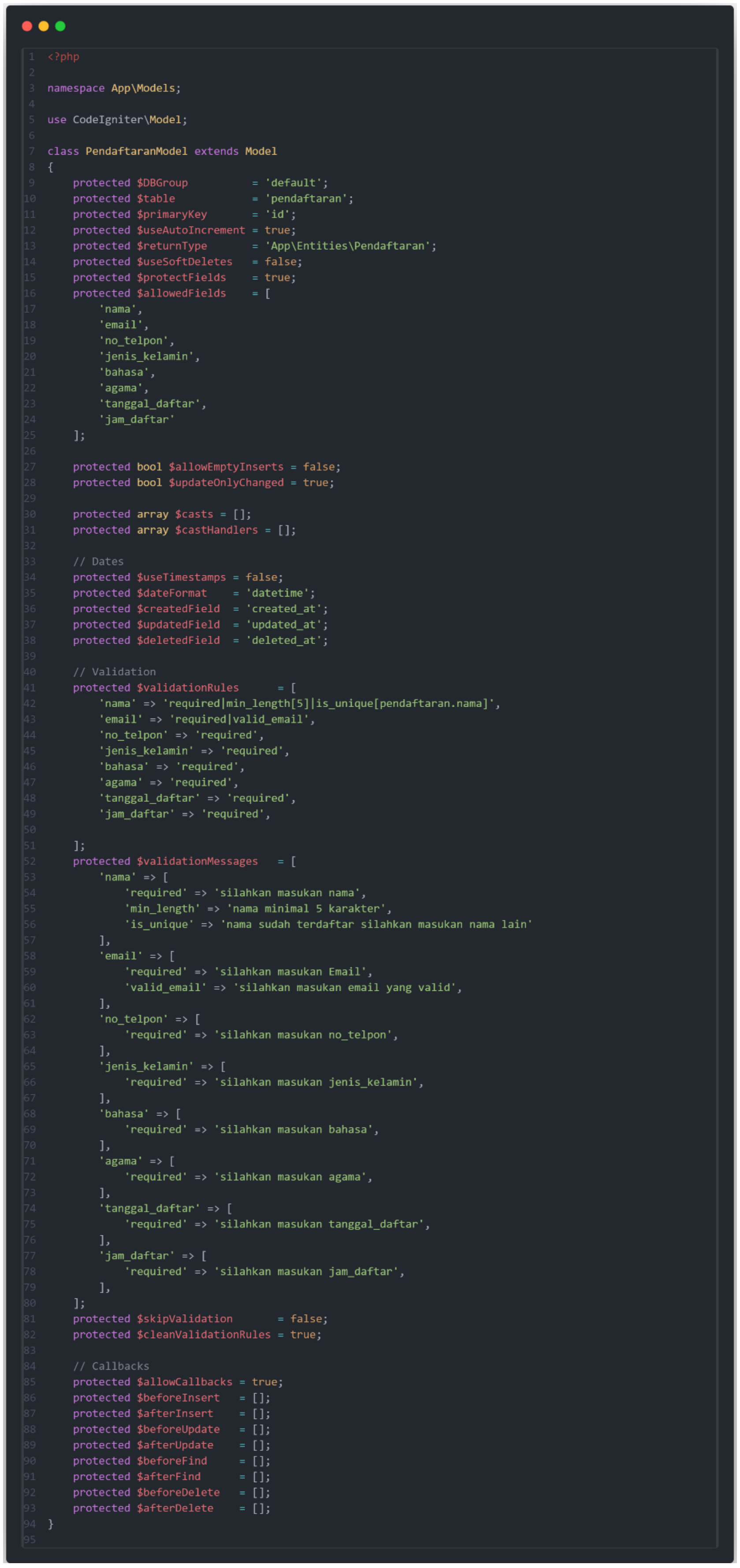
```php
<?php

namespace App\Database\Seeds;

use CodeIgniter\Database\Seeder;

class Pendaftaran extends Seeder
{
    public function run()
    {

        $data = [
            [
                'nama' => 'Fajar',
                'email' => 'fajar@gmail.com',
                'no_telpon' => '08961',
                'jenis_kelamin' => 'Pria',
                'bahasa' => 'Indonesia,Inggris,Arab,Sunda,Jawa',
                'agama' => 'Islam',
                'tanggal_daftar' => '2024-01-01',
                'jam_daftar' => '08:30:45',
            ],
            [
                'nama' => 'Hermanto',
                'email' => 'hermanto@gmail.com',
                'no_telpon' => '08962',
                'jenis_kelamin' => 'Pria',
                'bahasa' => 'Indonesia,Inggris,Arab',
                'agama' => 'Islam',
                'tanggal_daftar' => '2024-02-21',
                'jam_daftar' => '09:30:45',
            ],
            [
                'nama' => 'Nandang Hermanto',
                'email' => 'nandanghermanto@gmail.com',
                'no_telpon' => '08963',
                'jenis_kelamin' => 'Pria',
                'bahasa' => 'Indonesia,Inggris,Arab,Sunda',
                'agama' => 'Islam',
                'tanggal_daftar' => '2024-03-04',
                'jam_daftar' => '10:35:45',
            ],
        ];
        $this->db->table('pendaftaran')->insertBatch($data);


    }
}

```

11. Untuk generate seeder / data dummy buka terminal vscode dengan ctrl+` dan jalankan ” php spark db:seed Pendaftaran”



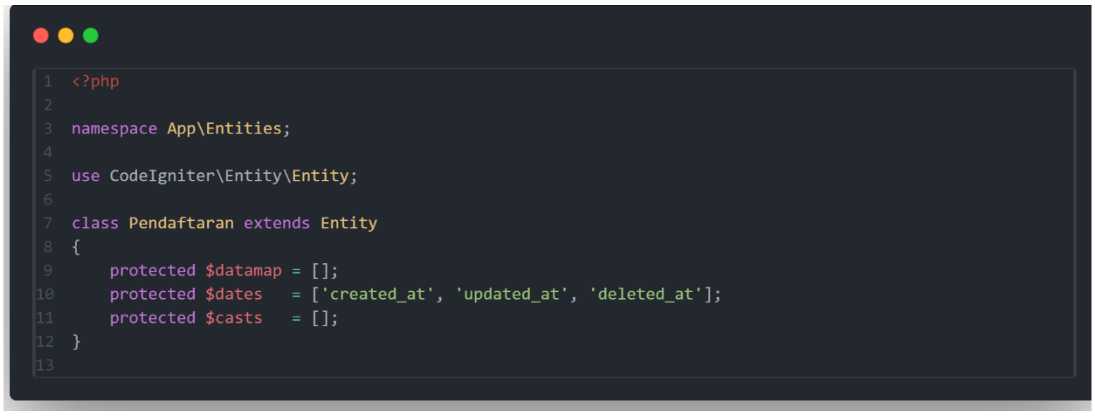
```
Pajar@Pajar MINGW64 /c/xampp/htdocs/praktikum10-pmo
$ php spark db:seed Pendaftaran

CodeIgniter v4.6.1 Command Line Tool - Server Time: 2025-06-08 14:41:28 UTC+00:00

Seeded: App\Database\Seeds\Pendaftaran
```

12. Buat Model dengan cara buka terminal visual studio code kemudian perintahkan ” php spark make:model PendaftaranModel –suffix”



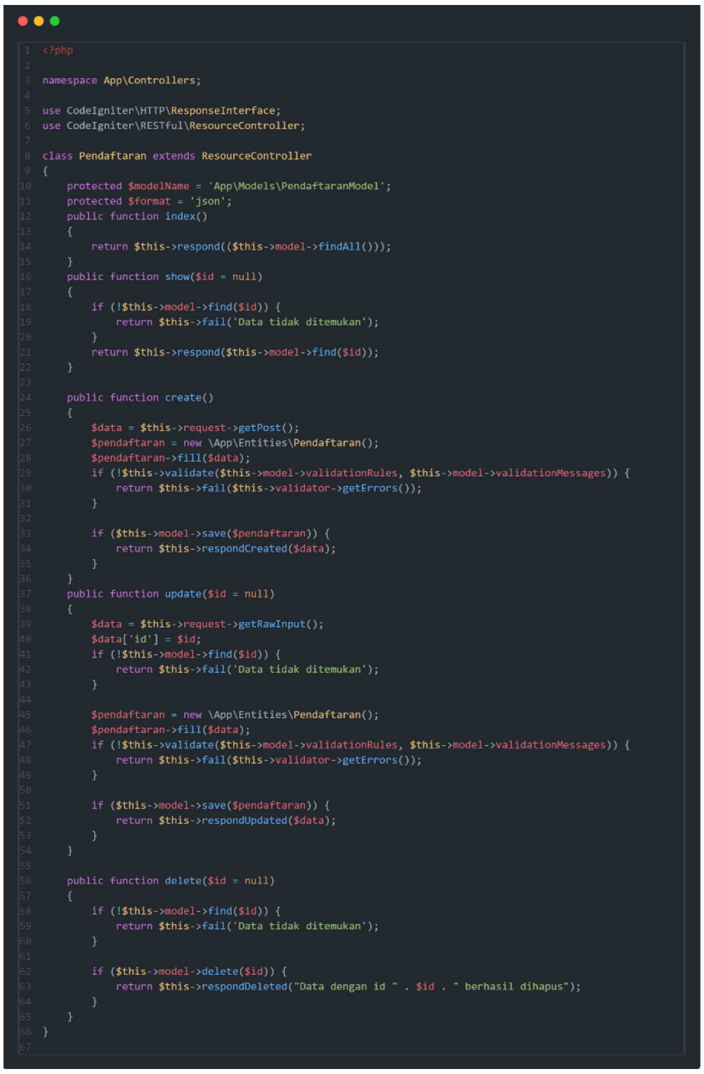
```php
<?php

namespace App\Models;

use CodeIgniter\Model;

class PendaftaranModel extends Model
{
    protected $DBGroup          = 'default';
    protected $table            = 'pendaftaran';
    protected $primaryKey       = 'id';
    protected $useAutoIncrement = true;
    protected $returnType       = 'App\Entities\Pendaftaran';
    protected $useSoftDeletes   = false;
    protected $protectFields    = true;
    protected $allowedFields    = [
        'nama',
        'email',
        'no_telpon',
        'jenis_kelamin',
        'bahasa',
        'agama',
        'tanggal_daftar',
        'jam_daftar'
    ];

    protected bool $allowEmptyInserts = false;
    protected bool $updateOnlyChanged = true;

    protected array $casts = [];
    protected array $castHandlers = [];

    // Dates
    protected $useTimestamps = false;
    protected $dateFormat    = 'datetime';
    protected $createdField  = 'created_at';
    protected $updatedField  = 'updated_at';
    protected $deletedField  = 'deleted_at';

    // Validation
    protected $validationRules      = [
        'nama' => 'required|min_length[5]|is_unique[pendaftaran.nama]',
        'email' => 'required|valid_email',
        'no_telpon' => 'required',
        'jenis_kelamin' => 'required',
        'bahasa' => 'required',
        'agama' => 'required',
        'tanggal_daftar' => 'required',
        'jam_daftar' => 'required',

    ];
    protected $validationMessages   = [
        'nama' => [
            'required' => 'silahkan masukan nama',
            'min_length' => 'nama minimal 5 karakter',
            'is_unique' => 'nama sudah terdaftar silahkan masukan nama lain'
        ],
        'email' => [
            'required' => 'silahkan masukan Email',
            'valid_email' => 'silahkan masukan email yang valid',
        ],
        'no_telpon' => [
            'required' => 'silahkan masukan no_telpon',
        ],
        'jenis_kelamin' => [
            'required' => 'silahkan masukan jenis_kelamin',
        ],
        'bahasa' => [
            'required' => 'silahkan masukan bahasa',
        ],
        'agama' => [
            'required' => 'silahkan masukan agama',
        ],
        'tanggal_daftar' => [
            'required' => 'silahkan masukan tanggal_daftar',
        ],
        'jam_daftar' => [
            'required' => 'silahkan masukan jam_daftar',
        ],
    ];
    protected $skipValidation       = false;
    protected $cleanValidationRules = true;

    // Callbacks
    protected $allowCallbacks = true;
    protected $beforeInsert   = [];
    protected $afterInsert    = [];
    protected $beforeUpdate   = [];
    protected $afterUpdate    = [];
    protected $beforeFind     = [];
    protected $afterFind      = [];
    protected $beforeDelete   = [];
    protected $afterDelete    = [];
}
```

13. Buat Entity yang berguna untuk mengatur format isi suatu data dalam tabel agar isi data memillliki format yang konsisten dalam praktikum ini yang di atur yaitu data yang berhubungan dengan waktu dibuat agar data tersebut diidentifikasi sebagai dates. Buat entiti dengan cara buka terminal visual studio code kemudian perintahkan " php spark make:entity Pendaftaran" lalu ubah isi file yang berada di /Entity/Pendaftaran



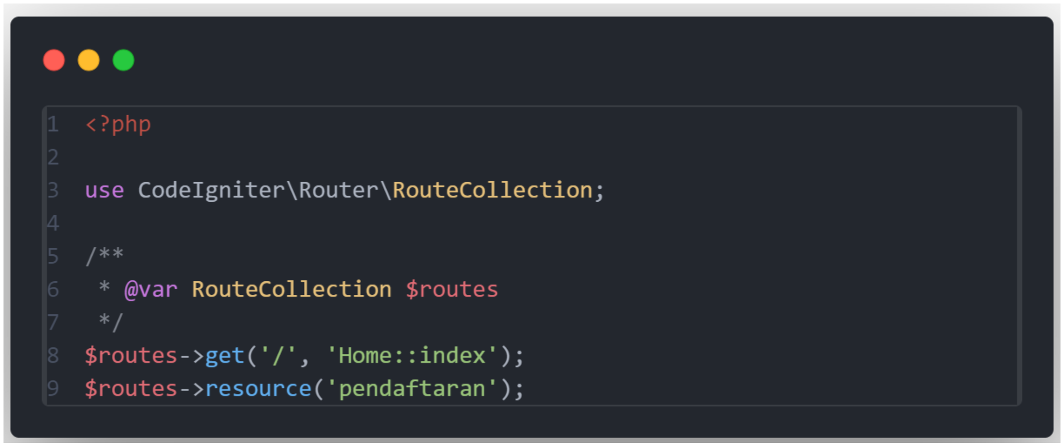
```php
<?php

namespace App\Entities;

use CodeIgniter\Entity\Entity;

class Pendaftaran extends Entity
{
    protected $datamap = [];
    protected $dates   = ['created_at', 'updated_at', 'deleted_at'];
    protected $casts   = [];
}

```

14. Buat ControllerUser yang berguna untuk mengatur request dan respon yang akan diberikan dan di terima dari user. Buat controller dengan cara buka terminal visual studio code kemudian perintahkan " php spark make:controller Pendaftaran --restful" dan edit hasil file di folder /Controlles/Pendaftaran



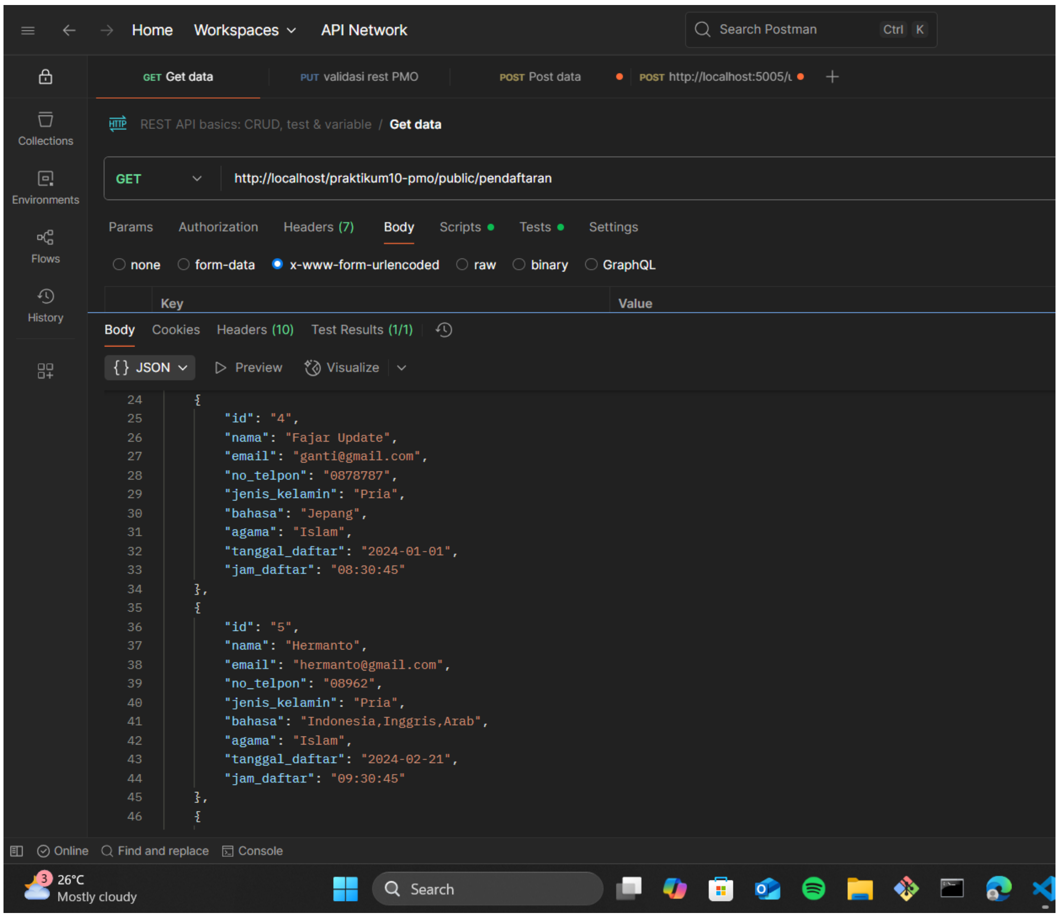
```php
<?php

namespace App\Controllers;

use CodeIgniter\HTTP\ResponseInterface;
use CodeIgniter\RESTful\ResourceController;

class Pendaftaran extends ResourceController
{
    protected $modelName = 'App\Models\PendaftaranModel';
    protected $format = 'json';
    public function index()
    {
        return $this->respond(($this->model->findAll()));
    }
    public function show($id = null)
    {
        if (!$this->model->find($id)) {
            return $this->fail('Data tidak ditemukan');
        }
        return $this->respond($this->model->find($id));
    }

    public function create()
    {
        $data = $this->request->getPost();
        $pendaftaran = new \App\Entities\Pendaftaran();
        $pendaftaran->fill($data);
        if (!$this->validate($this->model->validationRules, $this->model->validationMessages)) {
            return $this->fail($this->validator->getErrors());
        }

        if ($this->model->save($pendaftaran)) {
            return $this->respondCreated($data);
        }
    }
    public function update($id = null)
    {
        $data = $this->request->getRawInput();
        $data['id'] = $id;
        if (!$this->model->find($id)) {
            return $this->fail('Data tidak ditemukan');
        }

        $pendaftaran = new \App\Entities\Pendaftaran();
        $pendaftaran->fill($data);
        if (!$this->validate($this->model->validationRules, $this->model->validationMessages)) {
            return $this->fail($this->validator->getErrors());
        }

        if ($this->model->save($pendaftaran)) {
            return $this->respondUpdated($data);
        }
    }

    public function delete($id = null)
    {
        if (!$this->model->find($id)) {
            return $this->fail('Data tidak ditemukan');
        }

        if ($this->model->delete($id)) {
            return $this->respondDeleted("Data dengan id " . $id . " berhasil dihapus");
        }
    }
}

```

15. Atur Route dengan cara buka Routes.php yang terdapat di app\Config kemudian tambahkan $routes->resource('pendaftaran'); route ini berfungsi sebagai endpoint atau url yang harus diakses untuk mendapat segala hal pada tabel pendaftaran di server



```php
<?php

use CodeIgniter\Router\RouteCollection;

/**
 * @var RouteCollection $routes
 */
$routes->get('/', 'Home::index');
$routes->resource('pendaftaran');
```

16. Untuk menguji hasil rest api digunakan postman dengan berbagai method rest yang telah dibuat
    - GET all data method:



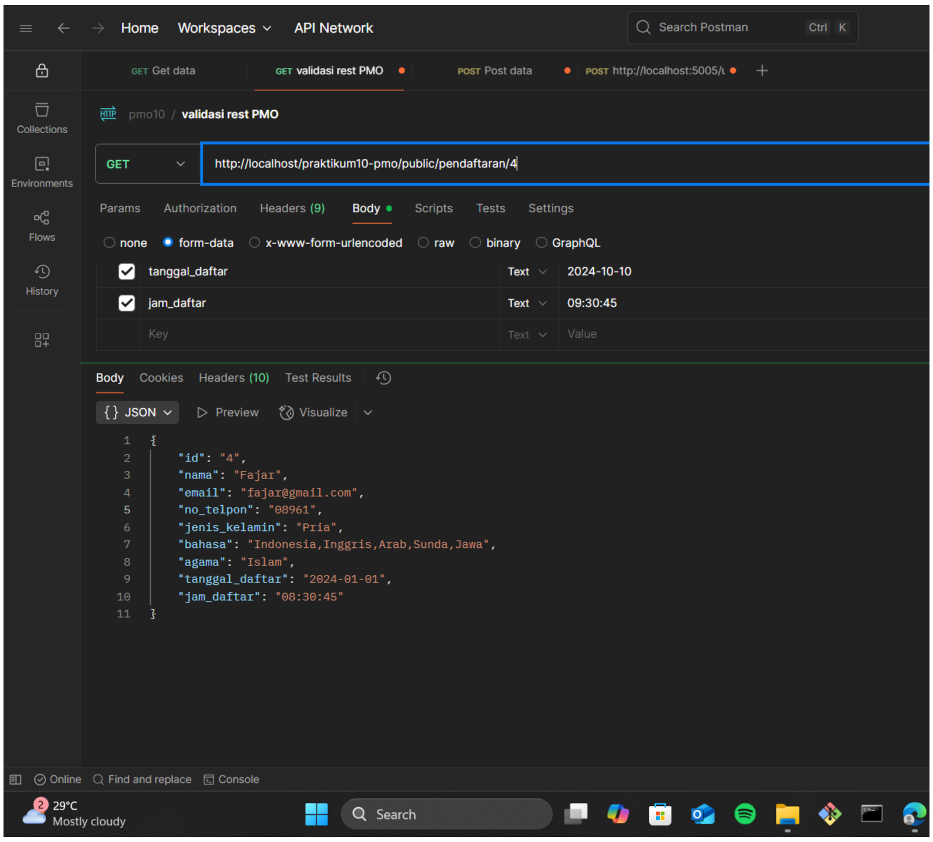
GET http://localhost/praktikum10-pmo/public/pendaftaran

Body: x-www-form-urlencoded

Body | Cookies | Headers (10) | Test Results (1/1)

```
    {
        "id": "4",
        "nama": "Fajar Update",
        "email": "ganti@gmail.com",
        "no_telpon": "0878787",
        "jenis_kelamin": "Pria",
        "bahasa": "Jepang",
        "agama": "Islam",
        "tanggal_daftar": "2024-01-01",
        "jam_daftar": "08:30:45"
    },
    {
        "id": "5",
        "nama": "Hermanto",
        "email": "hermanto@gmail.com",
        "no_telpon": "08962",
        "jenis_kelamin": "Pria",
        "bahasa": "Indonesia,Inggris,Arab",
        "agama": "Islam",
        "tanggal_daftar": "2024-02-21",
        "jam_daftar": "09:30:45"
    },
    {
```

- GET by ID data method



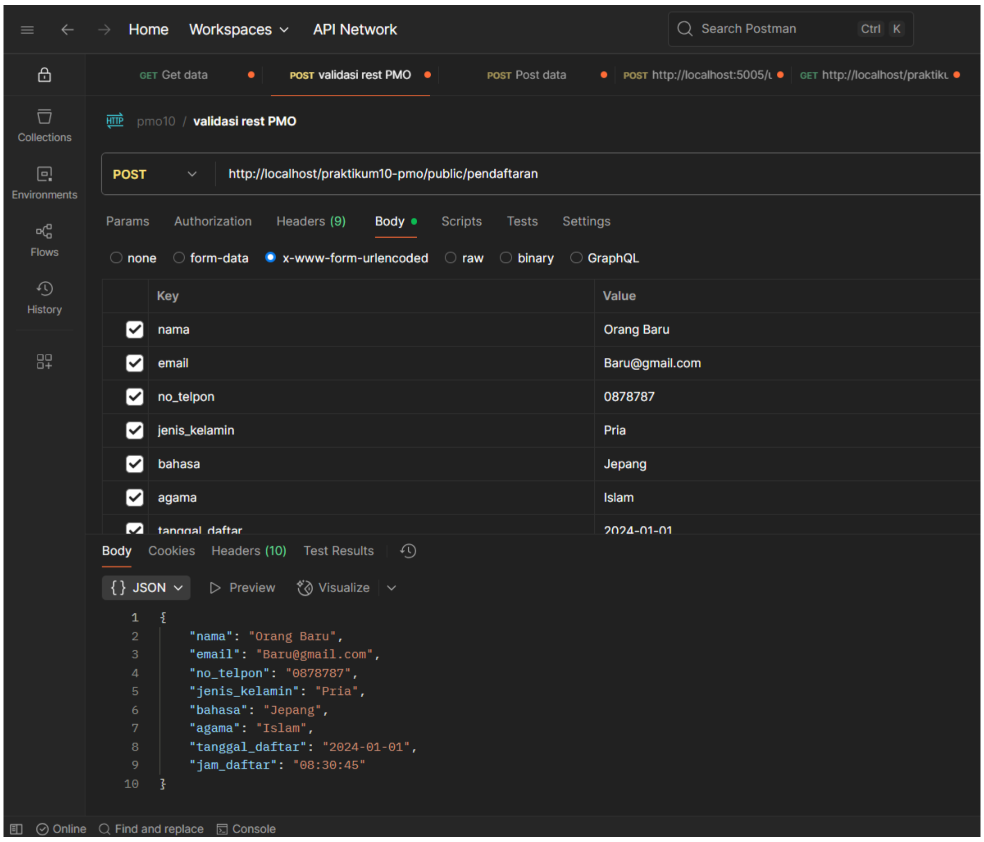
GET http://localhost/praktikum10-pmo/public/pendaftaran/4

Body: form-data

Key		Value
✓ tanggal_daftar	Text	2024-10-10
✓ jam_daftar	Text	09:30:45

```json
{
    "id": "4",
    "nama": "Fajar",
    "email": "fajar@gmail.com",
    "no_telpon": "08961",
    "jenis_kelamin": "Pria",
    "bahasa": "Indonesia,Inggris,Arab,Sunda,Jawa",
    "agama": "Islam",
    "tanggal_daftar": "2024-01-01",
    "jam_daftar": "08:30:45"
}
```

- POST method



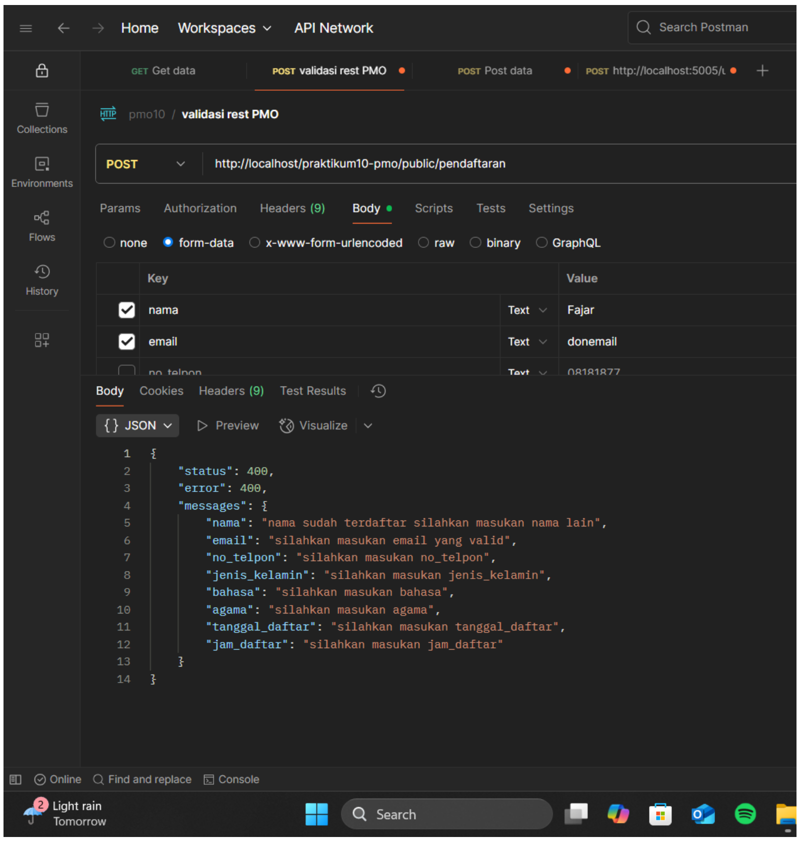
POST http://localhost/praktikum10-pmo/public/pendaftaran

Body: x-www-form-urlencoded

Key	Value
✓ nama	Orang Baru
✓ email	Baru@gmail.com
✓ no_telpon	0878787
✓ jenis_kelamin	Pria
✓ bahasa	Jepang
✓ agama	Islam
✓ tanggal_daftar	2024-01-01

```json
{
    "nama": "Orang Baru",
    "email": "Baru@gmail.com",
    "no_telpon": "0878787",
    "jenis_kelamin": "Pria",
    "bahasa": "Jepang",
    "agama": "Islam",
    "tanggal_daftar": "2024-01-01",
    "jam_daftar": "08:30:45"
}
```

- Coba testing beberapa validasi



pmo10 / validasi rest PMO

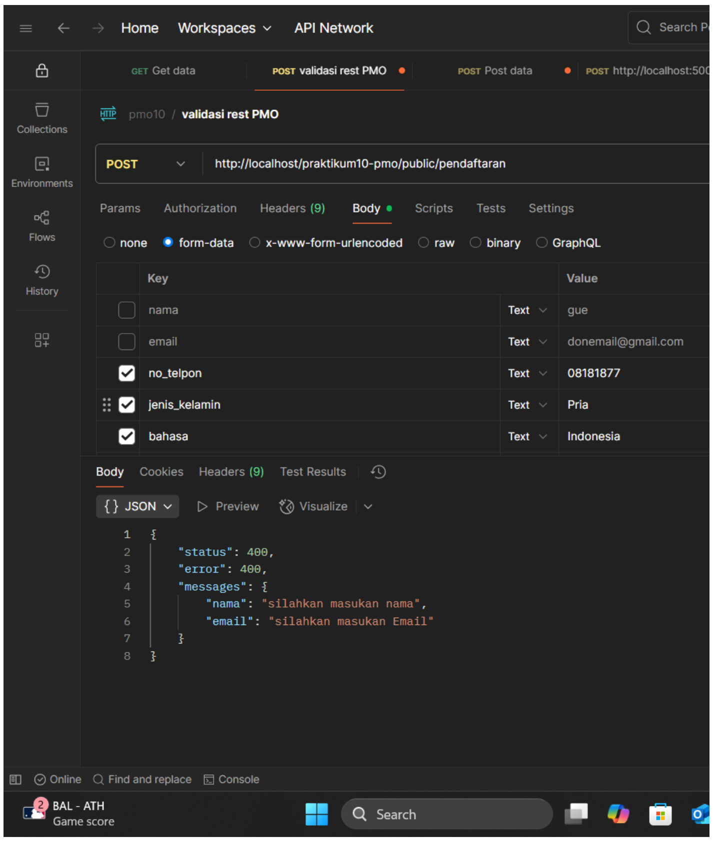
POST http://localhost/praktikum10-pmo/public/pendaftaran

Body: ○ none ◉ form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	Key		Value
✓	nama	Text	Fajar
✓	email	Text	donemail
	no_telpon	Text	08181877

{} JSON

```
{
    "status": 400,
    "error": 400,
    "messages": {
        "nama": "nama sudah terdaftar silahkan masukan nama lain",
        "email": "silahkan masukan email yang valid",
        "no_telpon": "silahkan masukan no_telpon",
        "jenis_kelamin": "silahkan masukan jenis_kelamin",
        "bahasa": "silahkan masukan bahasa",
        "agama": "silahkan masukan agama",
        "tanggal_daftar": "silahkan masukan tanggal_daftar",
        "jam_daftar": "silahkan masukan jam_daftar"
    }
}
```



pmo10 / validasi rest PMO

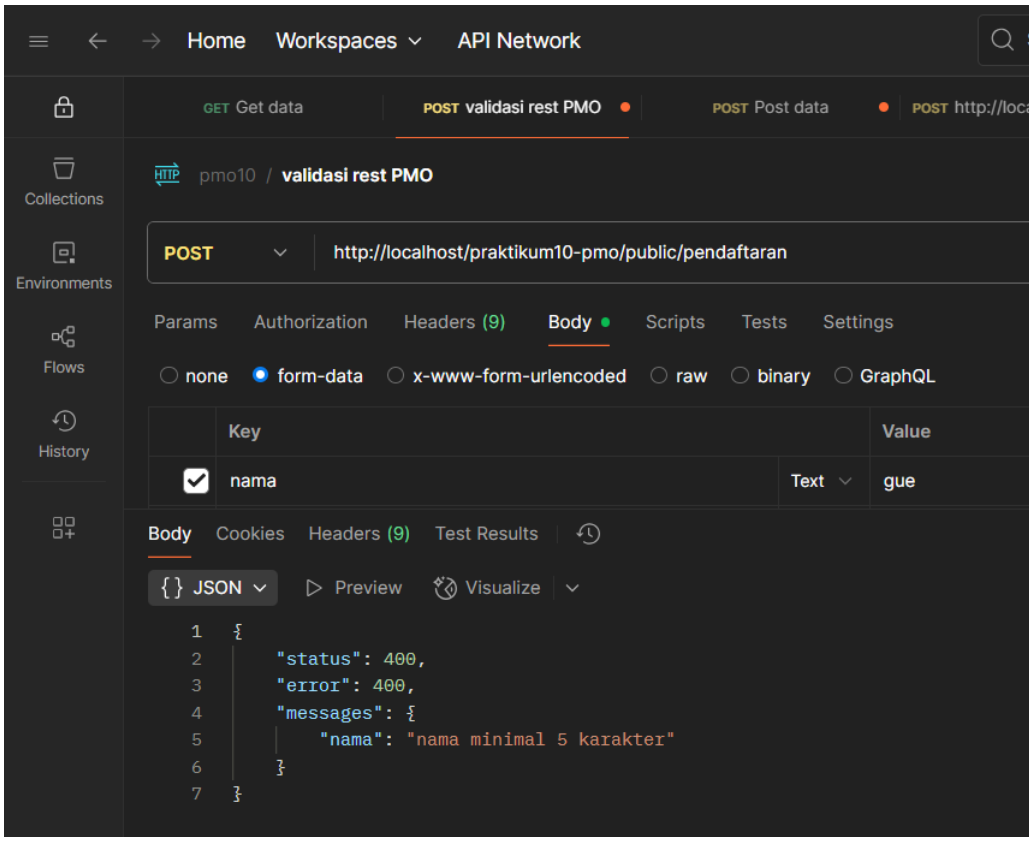
POST http://localhost/praktikum10-pmo/public/pendaftaran

Body: ○ none ◉ form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	Key		Value
	nama	Text	gue
	email	Text	donemail@gmail.com
✓	no_telpon	Text	08181877
✓	jenis_kelamin	Text	Pria
✓	bahasa	Text	Indonesia

{} JSON

```
{
    "status": 400,
    "error": 400,
    "messages": {
        "nama": "silahkan masukan nama",
        "email": "silahkan masukan Email"
    }
}
```



POST http://localhost/praktikum10-pmo/public/pendaftaran

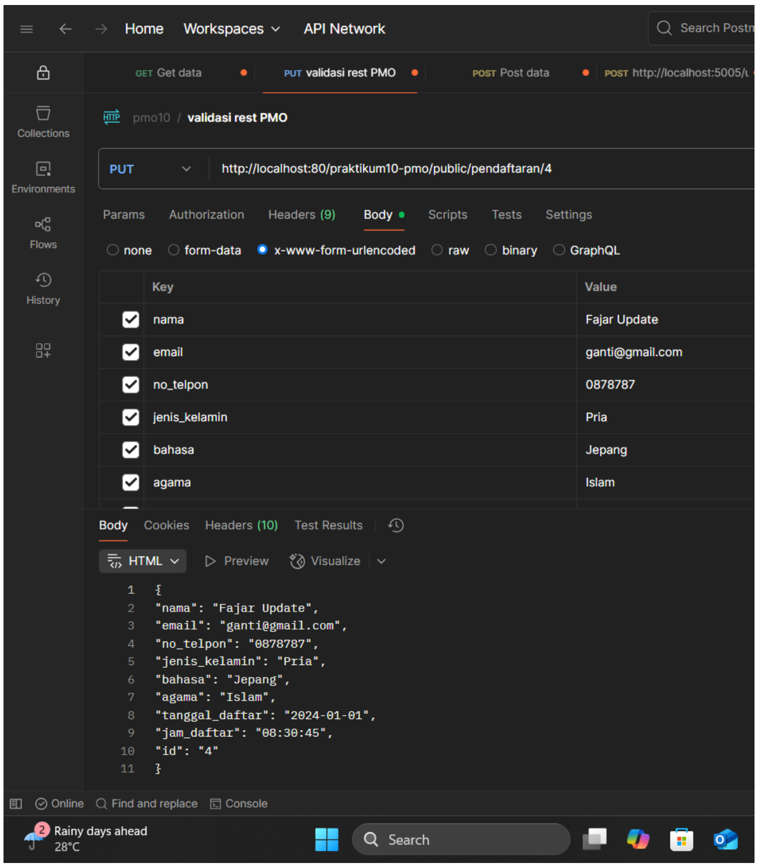
Body: form-data

Key		Value
nama	Text	gue

Response Body: JSON

```
{
    "status": 400,
    "error": 400,
    "messages": {
        "nama": "nama minimal 5 karakter"
    }
}
```

- PUT by ID method:



PUT http://localhost:80/praktikum10-pmo/public/pendaftaran/4

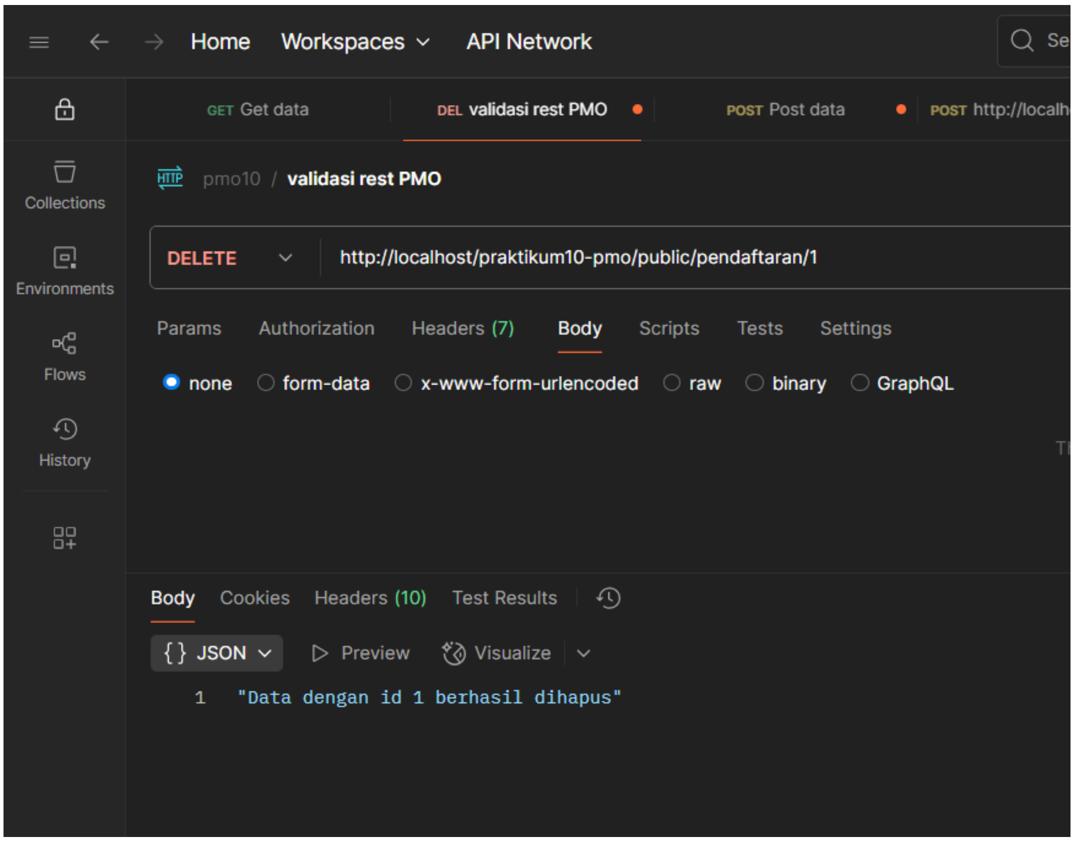
Body: x-www-form-urlencoded

Key	Value
nama	Fajar Update
email	ganti@gmail.com
no_telpon	0878787
jenis_kelamin	Pria
bahasa	Jepang
agama	Islam

Response Body: HTML

```
{
"nama": "Fajar Update",
"email": "ganti@gmail.com",
"no_telpon": "0878787",
"jenis_kelamin": "Pria",
"bahasa": "Jepang",
"agama": "Islam",
"tanggal_daftar": "2024-01-01",
"jam_daftar": "08:30:45",
"id": "4"
}
```

- DELETE by ID method:



DELETE http://localhost/praktikum10-pmo/public/pendaftaran/1

Body: [✓ none] form-data x-www-form-urlencoded raw binary GraphQL

JSON

```
"Data dengan id 1 berhasil dihapus"
```