

# LAPORAN PRAKTIKUM



**NIM** : 2003071

**Nama** : Hakim Asrori

**Kelas** : D3TI.2C

**Mata Kuliah** : **Pemrograman Perangkat Bergerak  
(TIU3403)**

**Praktikum ke / Judul** : 8 / NETWORKING & ASYNCHRONOUS

**Tanggal Praktikum** : 14 April 2022

**Dosen Pengampu** : Fachrul Pralienka Bani Muhamad, S.ST.,  
M.Kom

**PROGRAM STUDI D3 TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
POLITEKNIK NEGERI INDRAMAYU  
2022**



## A. Tujuan Praktikum

### Tujuan Umum

Mahasiswa

### Tujuan Khusus

Mahasiswa mampu:

1. Menjelaskan
2. ...

## B. Rangkuman Teori Singkat

### API

API adalah singkatan dari Application Programming Interface. Api sendiri merupakan interface penghubung antara satu aplikasi dengan aplikasi lainnya. Tujuan penggunaan API adalah untuk mempercepat proses development dengan menyediakan function secara terpisah sehingga developer tidak perlu membuat fitur yang serupa. Penerapan *API* akan sangat terasa jika fitur yang diinginkan sudah sangat kompleks, tentu membutuhkan waktu untuk membuat yang serupa dengannya.

### Lumen Micro-Framework

Lumen adalah Micro Framework yang diciptakan pengembang Laravel untuk mengakomodasi kebutuhan developer yang ingin membuat aplikasi dalam skala lebih kecil dari Laravel. Karena banyak library yang dihilangkan dalam bundle source code, Lumen bisa dijadikan framework untuk membuat REST API.

### Asynchronous

Asynchronous merupakan sebuah konsep dimana hasil eksekusi tidak selalu berdasarkan urutan, tetapi berdasarkan waktu proses. Jadi proses jalannya program bisa dilakukan secara bersamaan tanpa harus menunggu urutan eksekusi.

### Class Future

Class future memungkinkan kita untuk menjalankan kode asynchronous dan kita juga dapat menghindari callback dengan bantuannya. Class Future juga menyediakan fungsionalitas tingkat rendah yang lebih terarah untuk mengakses hasil yang dilengkapi dengannya. Kata kunci `async` dan `await` dibangun di atas fungsi ini.

### Kata kunci `async` dan `await`

`Async/Await` merupakan sebuah syntax khusus yang digunakan untuk menangani Promise agar penulisan code lebih efisien dan rapih. Function `async` berfungsi untuk menjadikan suatu function asynchronous. Dan `await` berfungsi menunda eksekusi hingga proses asynchronous selesai, `await` juga bisa digunakan berkali-kali di dalam function.

### CRUD Flutter



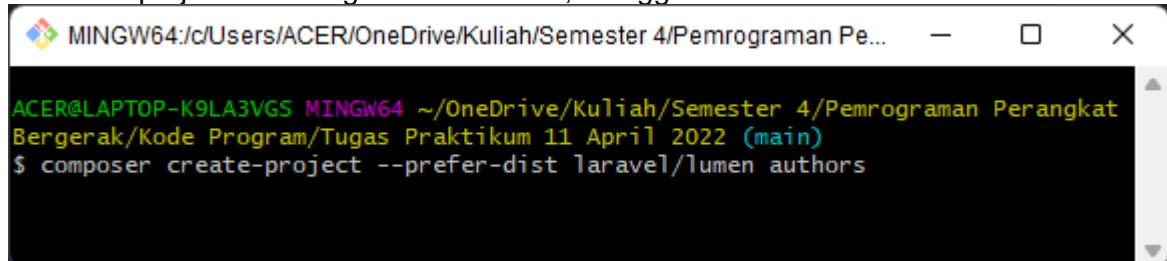
CRUD merupakan kepanjangan dari Create, Read, Update dan Delete. Jadi CRUD Flutter adalah sebuah operasi yang dapat menjalankan masing-masing dari CRUD pada framework Flutter.



## C. Langkah dan Hasil Pelaksanaan Praktikum

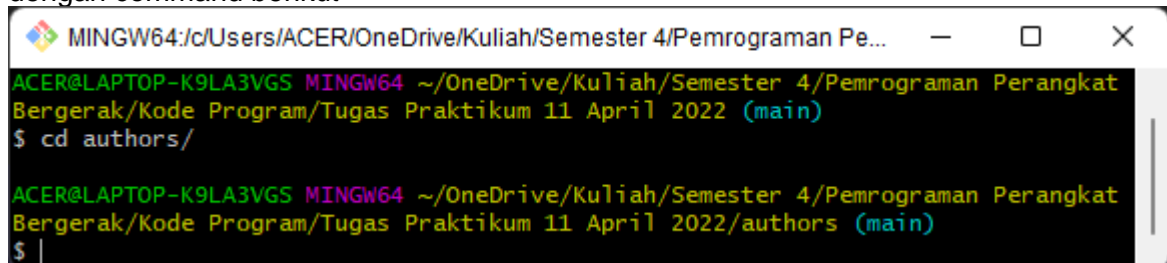
### Langkah-langkah praktikum Membuat Restful API

1. Dalam praktikum ini saya menggunakan Lumen Micro-Framework.
2. Membuat proyek baru dengan nama author, menggunakan command terminal berikut



```
ACER@LAPTOP-K9LA3VGS MINGW64 ~/OneDrive/Kuliah/Semester 4/Pemrograman Perangkat Bergerak/Kode Program/Tugas Praktikum 11 April 2022 (main)
$ composer create-project --prefer-dist laravel/lumen authors
```

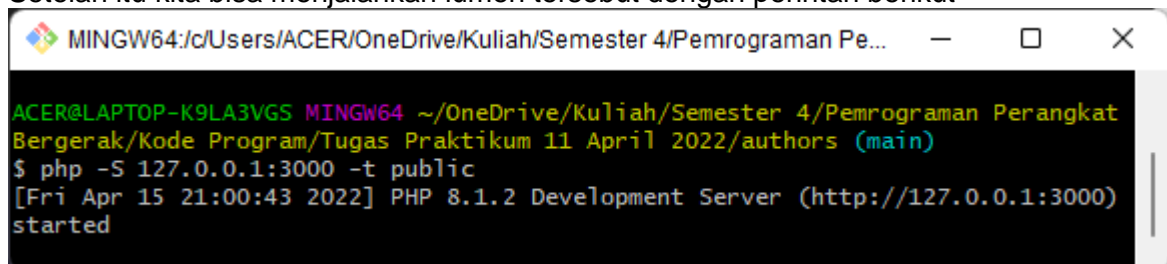
3. Tunggu prosesnya hingga selesai, lalu masuk ke folder proyek yang sudah dibuat dengan command berikut



```
ACER@LAPTOP-K9LA3VGS MINGW64 ~/OneDrive/Kuliah/Semester 4/Pemrograman Perangkat Bergerak/Kode Program/Tugas Praktikum 11 April 2022 (main)
$ cd authors/

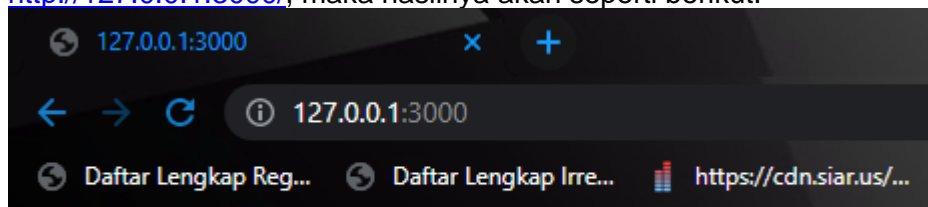
ACER@LAPTOP-K9LA3VGS MINGW64 ~/OneDrive/Kuliah/Semester 4/Pemrograman Perangkat Bergerak/Kode Program/Tugas Praktikum 11 April 2022/authors (main)
$ |
```

4. Setelah itu kita bisa menjalankan lumen tersebut dengan perintah berikut



```
ACER@LAPTOP-K9LA3VGS MINGW64 ~/OneDrive/Kuliah/Semester 4/Pemrograman Perangkat Bergerak/Kode Program/Tugas Praktikum 11 April 2022/authors (main)
$ php -S 127.0.0.1:3000 -t public
[Fri Apr 15 21:00:43 2022] PHP 8.1.2 Development Server (http://127.0.0.1:3000)
started
```

5. Kita coba akses melalui website dengan mengetik di mesin pencarian url yaitu <http://127.0.0.1:3000/>, maka hasilnya akan seperti berikut.



Lumen (9.0.2) (Laravel Components ^9.0)

6. Selanjutnya konfigurasi pada file app.php yang berada di folder bootstrap.
7. Hapus komentar pada sintaks berikut.

```
// $app->withFacades();

// $app->withEloquent();
```

**Penjelasan:** dengan kita hapus komentar pada sintaks diatas, dengan otomatis kita dapat menjalankan fungsi-fungsi dari facades dan eloquent itu sendiri.



8. Selanjutnya yaitu setting database pada file .env seperti berikut. Pastikan database, username, dan passwordnya menyesuaikan dengan server yang kita punya

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=db_ppb
DB_USERNAME=root
DB_PASSWORD=
```

9. Selanjutnya yaitu membuat table dengan command berikut

```
MINGW64:/c/Users/ACER/OneDrive/Kuliah/Semester 4/Pemrograman Pe...
ACER@LAPTOP-K9LA3VGS MINGW64 ~/OneDrive/Kuliah/Semester 4/Pemrograman Perangkat
Bergerak/Kode Program/Tugas Praktikum 11 April 2022/authors (main)
$ php artisan make:migration create_authors_table
```

10. Selanjutnya kita isi kolom-kolom untuk menyimpan data yang kita butuhkan kedalam database seperti berikut.



```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('authors', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email');
            $table->string('github');
            $table->string('twitter');
            $table->string('location');
            $table->string('latest_article_published');
            $table->string('image')->default('https://via.placeholder.com/480x480.png/00dd88?text=ducimus');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('authors');
    }
};
```

11. Selanjutnya kita buat sebuah file model dengan nama Author.php, di dalam folder app/Models
12. Lalu tambahkan kode berikut ini



```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Factories\HasFactory;

class Author extends Model
{
    use HasFactory;
    /**
     * The attributes that are mass assignable.
     *
     * @var string[]
     */
    protected $guarded = ['id', 'created_at', 'updated_at'];

    /**
     * The attributes excluded from the model's JSON form.
     *
     * @var string[]
     */
    protected $hidden = [];
}
```

13. Selanjutnya buat sebuah file factory dengan nama AuthorFactory.php didalam folder database/factories

14. Lalu isikan kode berikut ini



```
<?php

namespace Database\Factories;

use App\Models\Author;
use Illuminate\Database\Eloquent\Factories\Factory;

class AuthorFactory extends Factory
{
    /**
     * The name of the factory's corresponding model.
     *
     * @var string
     */
    protected $model = Author::class;

    /**
     * Define the model's default state.
     *
     * @return array
     */
    public function definition()
    {
        return [
            'name' => $this->faker->name,
            'email' => $this->faker->unique()->safeEmail,
            'github' => preg_replace('/[^\a-z0-9]+/i', '-', trim(strtolower($this->faker->userName))),
            'twitter' => preg_replace('/[^\a-z0-9]+/i', '-', trim(strtolower($this->faker->userName))),
            'location' => $this->faker->city,
            'latest_article_published' => date('d F Y'),
            'image' => $this->faker->imageUrl(480, 480),
        ];
    }
}
```

**Penjelasan:** faker merupakan sebuah library yang berguna untuk mengenerate ratusan data hingga ribuan data palsu untuk testing secara acak.

15. Selanjutnya kita upload table yang kita buat menggunakan command berikut

```
MINGW64:/c/Users/ACER/OneDrive/Kuliah/Semester 4/Pemrograman Pe...
ACER@LAPTOP-K9LA3VGS MINGW64 ~/OneDrive/Kuliah/Semester 4/Pemrograman Perangkat
Bergerak/Kode Program/Tugas Praktikum 11 April 2022/authors (main)
$ php artisan migrate --seed
```

16. Dapat dilihat hasilnya melalui phpMyAdmin/aplikasi mysql client





	id	name	email	github	twitter	location	latest_article_published	image
<input type="checkbox"/>	1	Laron Thiel	mjohns@example.net	nkris	klegros	Barrowsborough	15 April 2022	https://via.placeholder.com/480tex...
<input type="checkbox"/>	2	Pansy Rolfson	scottie.parker@example.com	sschimmel	sandrine99	Goyettfurt	15 April 2022	https://via.placeholder.com/480tex...
<input type="checkbox"/>	3	Lexus Jerde	arturo.skiles@example.com	jaunita-feest	rolfson-verda	Vellamouth	15 April 2022	https://via.placeholder.com/480tex...
<input type="checkbox"/>	4	Beau Donnelly	lschultz@example.org	emard-kennith	mara35	Myahburgh	15 April 2022	https://via.placeholder.com/480tex...
<input type="checkbox"/>	5	Belle Bosco DVM	hagenes.coleman@example.net	korbin-bogisich	durgan-griffin	Julienland	15 April 2022	https://via.placeholder.com/480tex...
<input type="checkbox"/>	6	Rey Zboncak PhD	weldon98@example.com	zemlak-phyllis	nicholaus-schmeler	Rosellaland	15 April 2022	https://via.placeholder.com/480tex...
<input type="checkbox"/>	7	Tobin Adams	lonnie20@example.org	skylasipes	rolando-considine	Tarynburgh	15 April 2022	https://via.placeholder.com/480tex...
<input type="checkbox"/>	8	Clare Rempel	drake.denesik@example.com	trey74	bechtelar-alfred	Genesisfort	15 April 2022	https://via.placeholder.com/480tex...
<input type="checkbox"/>	9	Belle Roob	daniel.benjamin@example.com	xwalter	kub-duane	O'Keefetown	15 April 2022	https://via.placeholder.com/480tex...
<input type="checkbox"/>	10	Dr. Ola Hoppe	lenore78@example.com	ferry-hollie	rhoda25	Kuphalview	15 April 2022	https://via.placeholder.com/480tex...

17. Selanjutnya membuat perutean yang berada di routes/web.php

18. Lalu masukan kode berikut ini

```
<?php

use App\Http\Controllers\AuthorController;
use App\Http\Controllers\AuthController;

/** @var \Laravel\Lumen\Routing\Router $router */

/*
|-----
| Application Routes
|-----
|
| Here is where you can register all of the routes for an application.
| It is a breeze. Simply tell Lumen the URIs it should respond to
| and give it the Closure to call when that URI is requested.
|
|*/

$router->get('/', function () use ($router) {
    return $router->app->version();
});

$router->group(['prefix' => 'api'], function () use ($router) {
    $router->get('authors', ['uses' => 'AuthorController@showAllAuthors']);

    $router->get('authors/{id}', ['uses' => 'AuthorController@showOneAuthors']);

    $router->post('authors', ['uses' => 'AuthorController@create']);

    $router->delete('authors/{id}', ['uses' => 'AuthorController@delete']);

    $router->put('authors/{id}', ['uses' => 'AuthorController@update']);
});
```



19. Selanjutnya kita buat sebuah controller dengan nama AuthorController.php yang disimpan didalam folder app/Http/Controller
20. Lalu masukan kode berikut ini



```
<?php

namespace App\Http\Controllers;

use App\Models\Author;
use Illuminate\Http\Request;

class AuthorController extends Controller
{
    public function showAllAuthors()
    {
        $author = Author::orderBy('name', 'asc')->get();

        $data = [];

        foreach ($author as $a) {
            $data[] = [
                'id' => $a->id,
                'name' => $a->name,
                'email' => $a->email,
                'github' => $a->github,
                'twitter' => $a->twitter,
                'location' => $a->location,
                'image' => $a->image,
                'latest_article_published' => $a->latest_article_published,
                'created_at' => date('d F Y', strtotime($a->updated_at)),
            ];
        }

        return response()->json($data);
    }

    public function showOneAuthors($id)
    {
        return response()->json(Author::find($id));
    }

    public function create(Request $request)
    {
        $this->validate($request, [
            'name' => 'required',
            'email' => 'required|email|unique:authors',
            'location' => 'required|alpha'
        ]);

        $authors = Author::create($request->all());

        return response()->json($authors, 201);
    }

    public function update($id, Request $request)
    {
        $authors = Author::findOrFail($id);
        $authors->update($request->all());

        return response()->json($authors, 200);
    }

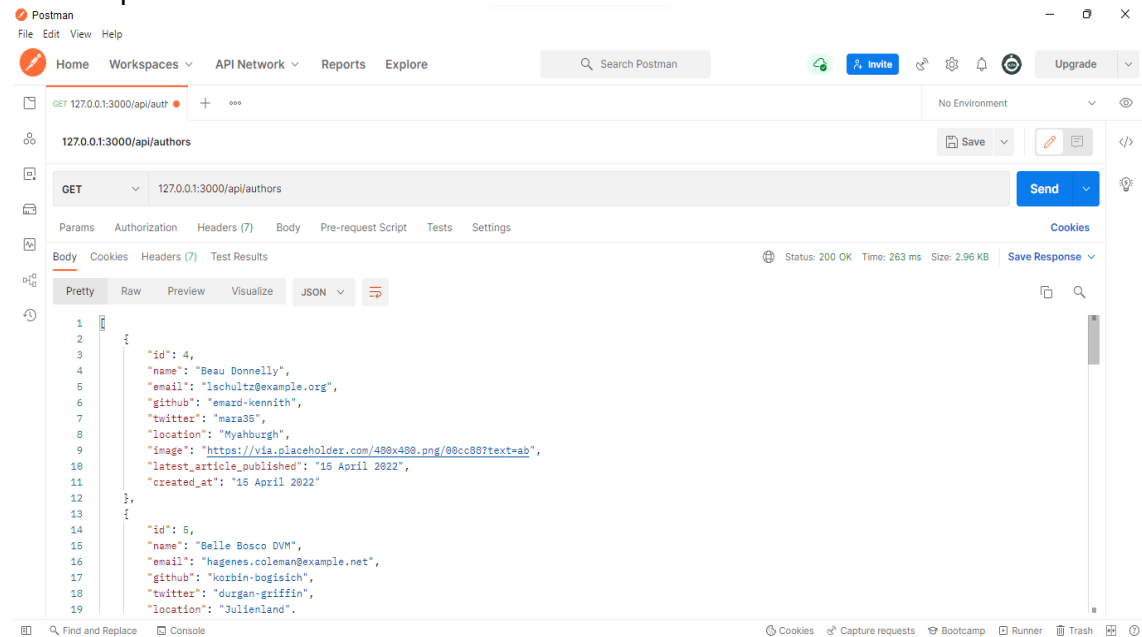
    public function delete($id)
    {
        $authors = Author::findOrFail($id)->delete();

        return response()->json('Deleted Successfully', 200);
    }
}
```

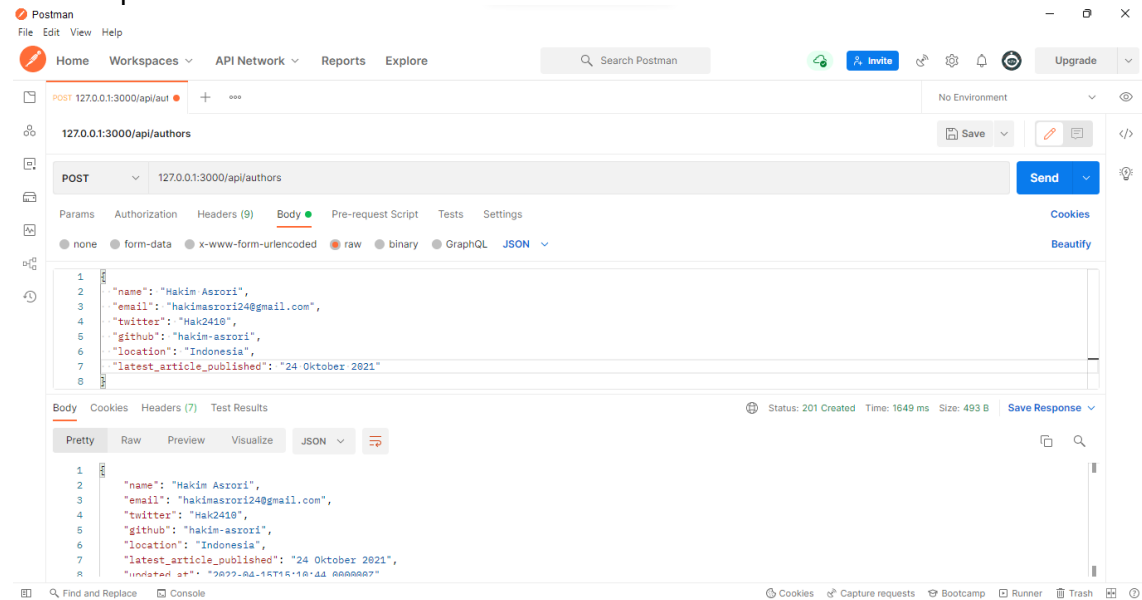


## 21. Selanjutnya kita lakukan percobaan dengan menggunakan postman seperti berikut

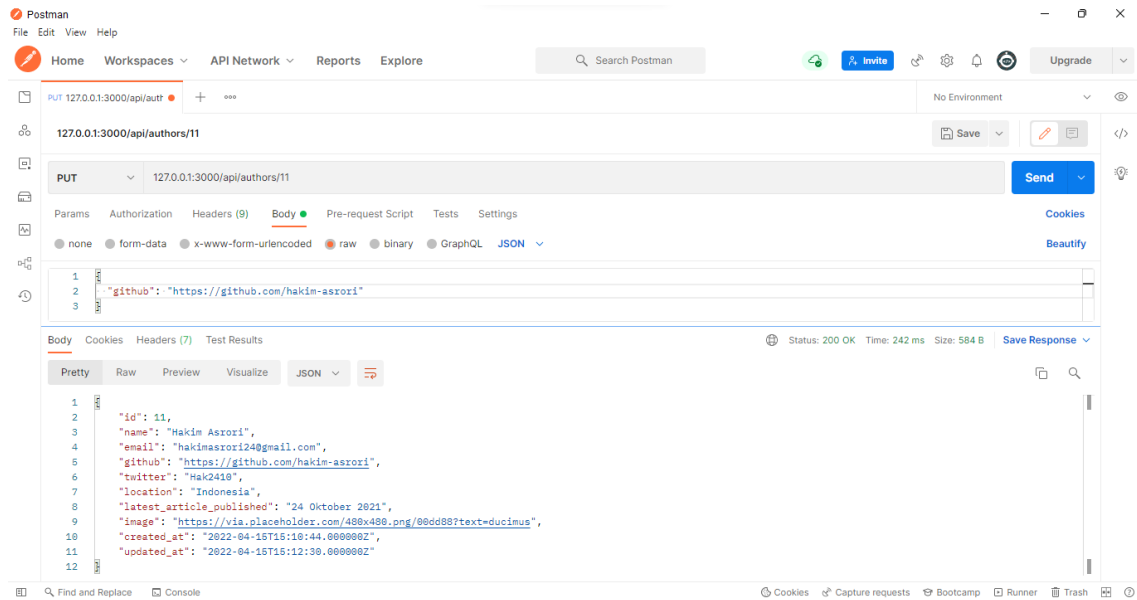
### 21.1. Operasi GET



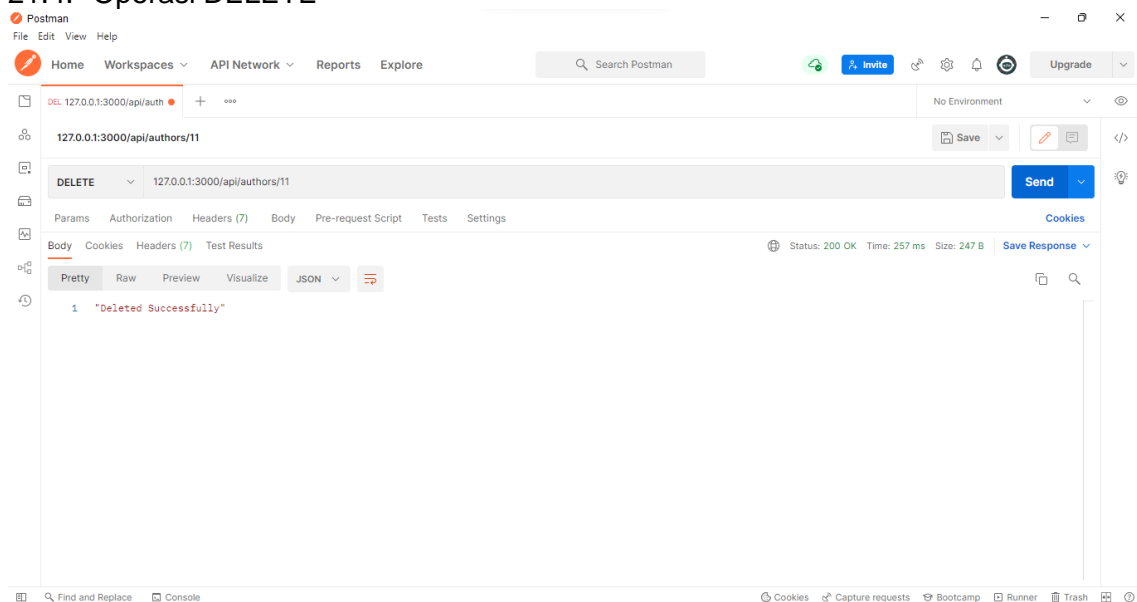
### 21.2. Operasi POST



### 21.3. Operasi PUT



## 21.4. Operasi DELETE



21.5. Selesai. Untuk kode program yang lengkapnya bisa diakses di link github berikut:  
<https://github.com/hakim-asrori/Pemrograman-Perangkat-Bergerak/tree/main/Tugas%20Praktikum%2011%20April%202022/authors>



## Langkah-langkah praktikum Future, async, & await

### Kode Program Future Bad

```
String createOrderMessage() {  
    var order = fetchUserOrder();  
    return 'Your order is: $order';  
}  
  
Future<String> fetchUserOrder() =>  
    Future.delayed(  
        const Duration(seconds: 2),  
        () => 'Large Latte',  
    );  
  
void main() {  
    print(createOrderMessage());  
}
```

### Hasil Kode Program Future Bad

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  
  
Your order is: Instance of 'Future<String>'  
Exited
```

### Kode Program Future Good

```
Future<void> fetchUserOrder() {  
    return Future.delayed(const Duration(seconds: 2), () => print('Large Latte'));  
}  
  
void main() {  
    fetchUserOrder();  
    print('Fetching user order...');  
}
```

### Hasil Kode Program Future Good

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  
  
Fetching user order...  
Large Latte  
Exited
```

### Kode Program Async & Await



```
Future<void> printOrderMessage() async {  
    print('Awaiting user order...');  
    var order = await fetchUserOrder();  
    print('Your order is: $order');  
}  
  
Future<String> fetchUserOrder() {  
    return Future.delayed(const Duration(seconds: 4), () => 'Large Latte');  
}  
  
void main() async {  
    countSeconds(4);  
    await printOrderMessage();  
}  
  
void countSeconds(int s) {  
    for (var i = 1; i <= s; i++) {  
        Future.delayed(Duration(seconds: i), () => print(i));  
    }  
}
```

Hasil Kode Program Async & Await

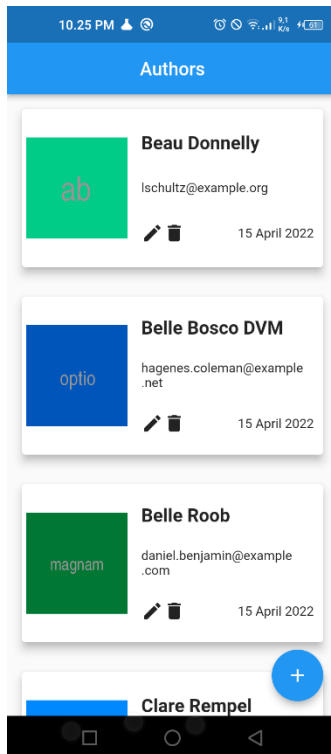
PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL
		Awaiting user order...	
		1	
		2	
		3	
		4	
		Your order is: Large Latte	
		Exited	



#### D. Langkah dan Hasil Latihan

Berikut hasil Latihan yang sudah dibuat dimana datanya menggunakan API yang sudah dibuat dipraktikum, untuk kode program dari flutter nya sendiri sudah tersedia digithub. Berikut link nya:

[https://github.com/hakim-asrori/Pemrograman-Perangkat-Bergerak/tree/main/Tugas%20Praktikum%202014%20April%202022/crud\\_api](https://github.com/hakim-asrori/Pemrograman-Perangkat-Bergerak/tree/main/Tugas%20Praktikum%202014%20April%202022/crud_api)







10.26 PM

← Add Author

Name  
Hakim Asrori

Email  
saneglos005@gmail.com

Github  
hakim-asrori

Twitter  
hak2410

Latest Article Publish  
24 Oktober 2020

Location  
Indramayu

Save

10.26 PM

← Edit Author Beau Donnelly

Name  
Beau Donnelly

Email  
lschultz@example.org

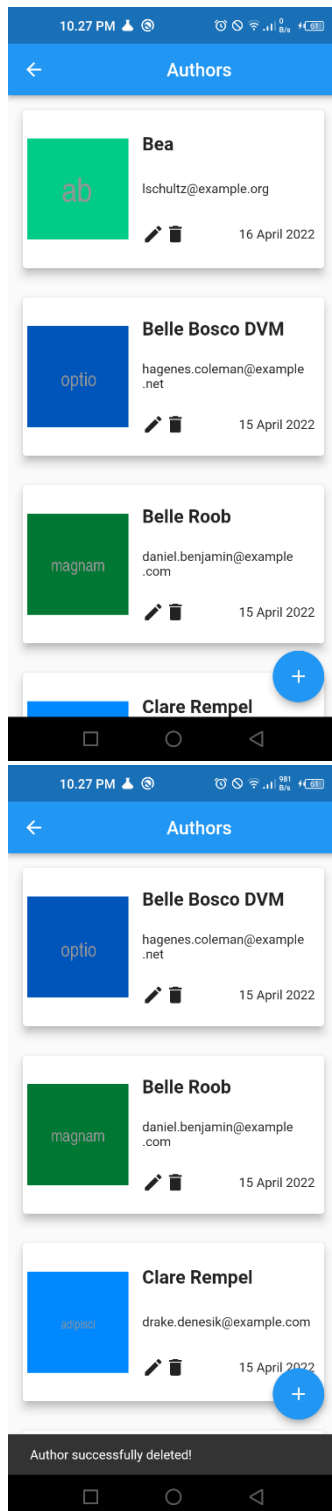
Github  
emard-kennith

Twitter  
mara35

Latest Article Publish  
15 April 2022

Location  
Myahburgh

Save





#### E. Hasil dan Tugas

1. Integrasikan ke mata kuliah proyek
  - a. Perubahan kode program harus dapat terlacak pada suatu repository terpusat (github, gitlab, atau bitbucket)
  - b. Lampirkan tautan ke repository tersebut

Link Github: <https://github.com/hakim-asrori/Implementasi-PPB-Di-Projek>



## F. Kesimpulan

API sangat dibutuhkan di dunia programming, karena dengan adanya API kita dapat mengolah data dengan proses yang sama dapat digunakan oleh Bahasa pemrograman yang berbeda. Dalam studi kasus yang saya buat ialah menghubungkan antara WEB dengan Mobile menggunakan perantara API yang.

## G. Referensi

- <https://www.niagahoster.co.id/blog/api-adalah/>
- <https://www.codepolitan.com/mengenal-apa-itu-web-api-5a0c2855799c8/>
- <https://www.codepolitan.com/pengenalan-lumen-framework-micro-framework-berbasis-php-59f19fe6ea010/>
- <https://medium.com/coderupa/panduan-komplit-asynchronous-programming-pada-javascript-part-1-fca22279c056>
- <https://api.dart.dev/stable/2.16.2/dart-async/Future-class.html>
- <https://www.tutorialspoint.com/future-class-in-dart-programming>
- <https://www.matawebsite.com/blog/mengenal-async-await-javascript>
- <https://auth0.com/blog/developing-restful-apis-with-lumen-a-php-micro-framework/>