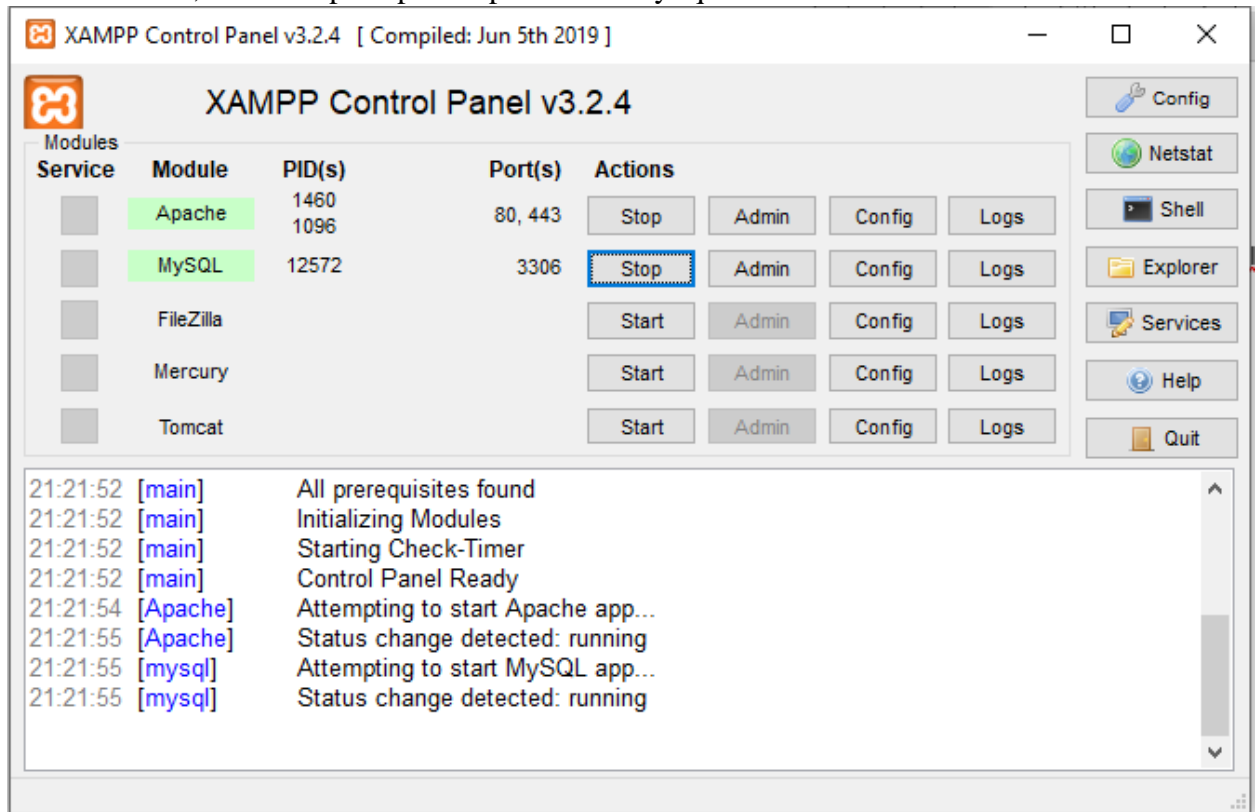


Repository : <https://github.com/ronny12062000/first-restful-api>

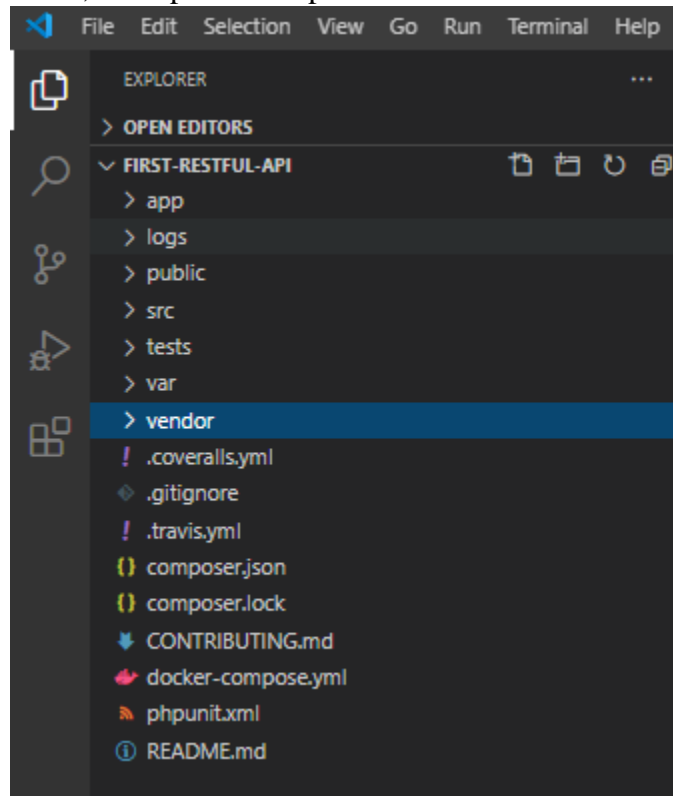
1. Buka XAMPP, klik start pada pada Apache dan MySQL



2. Masuk ke folder htdocs lalu buka command prompt. Ketikan “composer create-project slim/slim-skeleton first-restful-api”

```
C:\Windows\System32\cmd.exe
- Installing sebastian/comparator (3.0.3): Extracting archive
- Installing phpunit/php-timer (2.1.3): Extracting archive
- Installing phpunit/php-text-template (1.2.1): Extracting archive
- Installing phpunit/php-file-iterator (2.0.3): Extracting archive
- Installing theseer/tokenizer (1.2.0): Extracting archive
- Installing sebastian/code-unit-reverse-lookup (1.0.2): Extracting archive
- Installing phpunit/php-code-coverage (7.0.14): Extracting archive
- Installing doctrine/instantiator (1.4.0): Extracting archive
- Installing phpspec/prophecy (1.12.2): Extracting archive
- Installing phar-io/version (3.0.4): Extracting archive
- Installing phar-io/manifest (2.0.1): Extracting archive
- Installing myclabs/deep-copy (1.10.2): Extracting archive
- Installing phpunit/phpunit (8.5.13): Extracting archive
- Installing symfony/polyfill-php80 (v1.20.0): Extracting archive
- Installing ralouphie/getallheaders (3.0.3): Extracting archive
- Installing psr/http-message (1.0.1): Extracting archive
- Installing psr/http-factory (1.0.1): Extracting archive
- Installing fig/http-message-util (1.1.5): Extracting archive
- Installing slim/psr7 (1.3.0): Extracting archive
- Installing psr/http-server-handler (1.0.1): Extracting archive
- Installing psr/http-server-middleware (1.0.1): Extracting archive
- Installing nikic/fast-route (v1.3.0): Extracting archive
- Installing slim/slim (4.7.1): Extracting archive
19 package suggestions were added by new dependencies, use `composer suggest` to see details.
Package phpunit/php-token-stream is abandoned, you should avoid using it. No replacement was suggested.
Generating autoload files
25 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
```

3. Buka Visual Studi Code, lalu open folder pada folder FIRST-RESTFUL-API



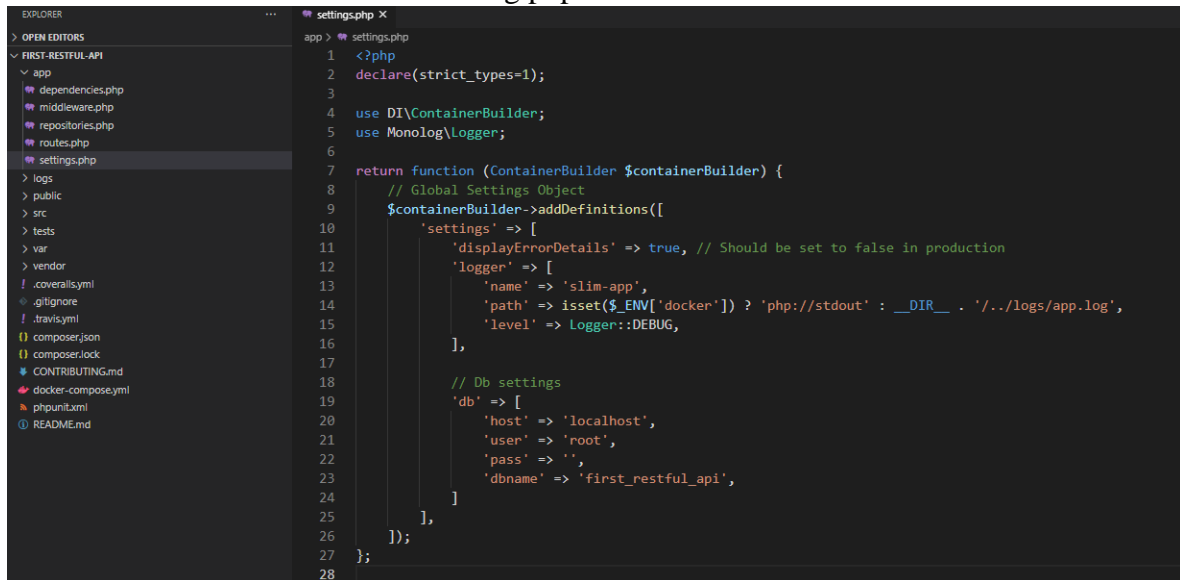
4. Buat database dengan nama first-rstful-api
5. Keudian buat table dengan nama buah, berikut daftar kolomnya

Table name: Add column(s)

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index
<input type="text" value="id"/> <small>Pick from Central Columns</small>	INT	3	None			<input type="checkbox"/>	PRIMARY <input checked="" type="checkbox"/>
<input type="text" value="nama"/> <small>Pick from Central Columns</small>	VARCHAR	100	None			<input type="checkbox"/>	--- <input type="checkbox"/>
<input type="text" value="created_at"/> <small>Pick from Central Columns</small>	TIMESTAMP		CURRENT_TIME			<input type="checkbox"/>	--- <input type="checkbox"/>
<input type="text" value="updated_at"/> <small>Pick from Central Columns</small>	TIMESTAMP		CURRENT_TIME		on update CURR	<input type="checkbox"/>	--- <input type="checkbox"/>

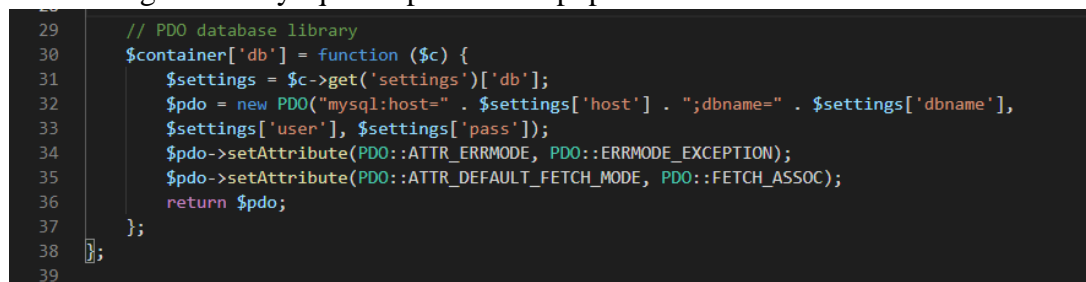
Table comments: Collation: Storage Engine:

6. Membuat konstanta database di setting.php



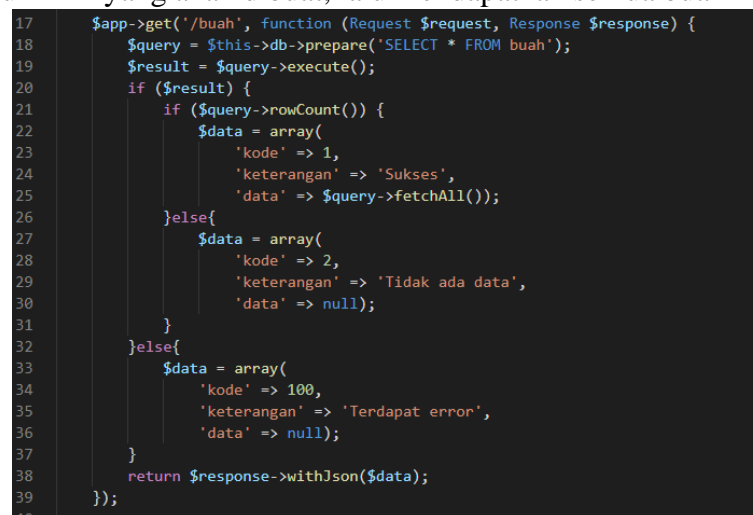
```
1 <?php
2 declare(strict_types=1);
3
4 use DI\ContainerBuilder;
5 use Monolog\Logger;
6
7 return function (ContainerBuilder $containerBuilder) {
8     // Global Settings Object
9     $containerBuilder->addDefinitions([
10         'settings' => [
11             'displayErrorDetails' => true, // Should be set to false in production
12             'logger' => [
13                 'name' => 'slim-app',
14                 'path' => isset($_ENV['docker']) ? 'php://stdout' : __DIR__ . '/../logs/app.log',
15                 'level' => Logger::DEBUG,
16             ],
17
18             // Db settings
19             'db' => [
20                 'host' => 'localhost',
21                 'user' => 'root',
22                 'pass' => '',
23                 'dbname' => 'first_restful_api',
24             ],
25         ],
26     ]);
27 };
28
```

7. Lalu sambungkan ke MySQL di dependencies.php



```
29 // PDO database library
30 $container['db'] = function ($c) {
31     $settings = $c->get('settings')['db'];
32     $pdo = new PDO("mysql:host=" . $settings['host'] . ";dbname=" . $settings['dbname'],
33         $settings['user'], $settings['pass']);
34     $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
35     $pdo->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);
36     return $pdo;
37 };
38
39
```

8. Mendaftarkan url API yang akan dibuat, lalu mendapatkan semua buah



```
17 $app->get('/buah', function (Request $request, Response $response) {
18     $query = $this->db->prepare('SELECT * FROM buah');
19     $result = $query->execute();
20     if ($result) {
21         if ($query->rowCount() > 0) {
22             $data = array(
23                 'kode' => 1,
24                 'keterangan' => 'Sukses',
25                 'data' => $query->fetchAll();
26             );
27         } else {
28             $data = array(
29                 'kode' => 2,
30                 'keterangan' => 'Tidak ada data',
31                 'data' => null;
32             );
33         }
34     } else {
35         $data = array(
36             'kode' => 100,
37             'keterangan' => 'Terdapat error',
38             'data' => null;
39         );
40     }
41     return $response->withJson($data);
42 });
43
```

9. Mendapatkan 1 data buah berdasarkan id

```
41 $app->get('/buah/{id}', function (Request $request, Response $response, array $args) {
42     $query = $this->db->prepare('SELECT * FROM buah WHERE id = :id');
43     $query->bindParam(':id', $args['id']);
44     $result = $query->execute();
45     if ($result) {
46         if ($query->rowCount()) {
47             $data = array(
48                 'kode' => 1,
49                 'keterangan' => 'Sukses',
50                 'data' => $query->fetch());
51         }else{
52             $data = array(
53                 'kode' => 2,
54                 'keterangan' => 'Tidak ada data',
55                 'data' => null);
56         }
57     }else{
58         $data = array(
59             'kode' => 100,
60             'keterangan' => 'Terdapat error',
61             'data' => null);
62     }
63     return $response->withJson($data);
64 });
```

10. Menambahkan 1 buah

```
66 $app->post('/buah', function (Request $request, Response $response) {
67     $params = $request->getParsedBody();
68     $nama = filter_var($params['nama'], FILTER_SANITIZE_STRING);
69     $query = $this->db->prepare('INSERT INTO buah (nama) VALUES(:nama)');
70     $query->bindParam(':nama', $nama);
71     $result = $query->execute();
72     if ($result) {
73         $data = array(
74             'kode' => 1,
75             'keterangan' => 'Sukses',
76             'data' => null);
77     }else{
78         $data = array(
79             'kode' => 100,
80             'keterangan' => 'Terdapat error',
81             'data' => null);
82     }
83     return $response->withJson($data);
84 });
```

11. Update nama buah

```
86     $app->put('/buah/{id}', function (Request $request, Response $response, array $args) {
87         $params = $request->getParsedBody();
88         $nama = filter_var($params['nama'], FILTER_SANITIZE_STRING);
89         $query = $this->db->prepare('UPDATE buah SET nama=:nama WHERE id = :id');
90         $query->bindParam(':nama', $nama);
91         $query->bindParam(':id', $args['id']);
92         $result = $query->execute();
93         if ($result) {
94             $data = array(
95                 'kode' => 1,
96                 'keterangan' => 'Sukses',
97                 'data' => null);
98         }else{
99             $data = array(
100                 'kode' => 100,
101                 'keterangan' => 'Terdapat error',
102                 'data' => null);
103         }
104         return $response->withJson($data);
105     });
```

12. Menghapus 1 buah

```
107     $app->delete('/buah/{id}', function (Request $request, Response $response, array $args) {
108         $query = $this->db->prepare('DELETE FROM buah WHERE id = :id');
109         $query->bindParam(':id', $args['id']);
110         $result = $query->execute();
111         if ($result) {
112             $data = array(
113                 'kode' => 1,
114                 'keterangan' => 'Sukses',
115                 'data' => null);
116         }else{
117             $data = array(
118                 'kode' => 100,
119                 'keterangan' => 'Terdapat error',
120                 'data' => null);
121         }
122         return $response->withJson($data);
123     });
```