PRAKTIKUM 05: UNIT TESTING DENGAN PHPUNIT PART 2

by Nurul Firdaus, S.Kom., M.Inf.

TUJUAN PEMBELAJARAN

Mahasiswa mampu menjelaskan dasar-dasar pengujian sistem pada komponen sistem

Mahasiswa mampu menjelaskan pengujian komponen sistem

Mahasiswa mampu menjelaskan dan melakukan pengujian unit/*unit testing* yang merupakan bagian dari komponen sistem

Mahasiswa mampu menjelaskan dan melakukan *unit testing* dengan PHPUnit di CI3/CI4 dan Laravel

REFERENSI

Link Penting

https://phpunit.de/

https://phpunit.de/getting-started/phpunit-9.html

https://phpunit.readthedocs.io/en/9.3/index.html

https://codeigniter4.github.io/userguide/testing/overview.html

https://semaphoreci.com/community/tutorials/getting-started-with-phpunit-in-laravel

https://laravel.com/docs/8.x/testing

Buku Referensi PHPUnit

Bergmann, S., 2005. *PHPUnit Pocket Guide: Test-Driven Development in PHP*. " O'Reilly Media, Inc."

Basic Ilmu Pengetahuan:

- Basic pemrograman PHP
- Konsep pemrograman berorientasikan objek (OOP) dengan PHP
- Sintaks dasar SQL

Tools:

1. Teks Editor: Visual Studio Code, Sublime Text, dll
2. Web Browser: Google Chrome atau Firefox
3. Web Server: PHP, MySQL, Phpmyadmin

   Requirement server untuk Codeigniter 4:

   - PHP Versi 7.2+
   - MySQL Versi 5.1+
   - Phpmyadmin

   Aktifkan extension *intl* dan *mbstring* dengan menghapus **;** (semicolon) yang ada di depan nama extension pada XAMPP Control Panel, lalu pada bagian apache klik **Config->PHP (php.ini)**

4. Composer
5. File Project Codeigniter

MATERI

CodeIgniter has been built to make testing both the framework and your application as simple as possible. Support for PHPUnit is built in, and the framework provides a number of convenient helper methods to make testing every aspect of your application as painless as possible.

Laravel is built with testing in mind. In fact, support for testing with PHPUnit is included out of the box and a `phpunit.xml` file is already set up for your application. The framework also ships with convenient helper methods that allow you to expressively test your applications.

By default, your application's tests directory contains two directories: Feature and Unit. Unit tests are tests that focus on a very small, isolated portion of your code. In fact, most unit tests probably focus on a single method. Feature tests may test a larger portion of your code, including how several objects interact with each other or even a full HTTP request to a JSON endpoint.

An `ExampleTest.php` file is provided in both the Feature and Unit test directories. After installing a new Laravel application, run `.\vendor\bin\phpunit` on the command line to run your tests.

PRAKTIKUM

1. Instalasi CI 4 dengan Composer

   Untuk menginstal CI 4 dengan composer, ketikkan perintah berikut:

   ```
   composer create-project codeigniter4/appstarter ci-project
   ```

   Tunggu sampai prosesnya selesai.

   Ada beberapa argumen yang kita berikan pada perintah ini:

   - `create-project` adalah perintah untuk membuat proyek baru dengan composer;
   - `codeigniter4/appstarter` adalah file CI yang akan di-download;
   - `ci4` adalah nama proyek yang akan kita buat;

   Setelah prosesnya selesai, kita akan mendapatkan folder baru dengan nama `ci-project`. ***(nama proyek/folder sesuai dengan preferensi masing-masing)***

   Untuk mengupdate project Codeigniter secara otomatis, ketikkan perintah berikut:

   ```
   D:\#Semester Ganjil 2020\Bahan Ajar\Praktikum RPL\latihan-php-unit\ci4>composer update
   ```

   ```
   D:\#Semester Ganjil 2020\Bahan Ajar\Praktikum RPL\latihan-php-unit>composer create-project codeigniter4/appstarter ci4
   Creating a "codeigniter4/appstarter" project at "./ci4"
   Installing codeigniter4/appstarter (v4.0.4)
     - Installing codeigniter4/appstarter (v4.0.4): Loading from cache
   Created project in D:\#Semester Ganjil 2020\Bahan Ajar\Praktikum RPL\latihan-php-unit\ci4
   Loading composer repositories with package information
   Updating dependencies (including require-dev)
   Package operations: 36 installs, 0 updates, 0 removals
     - Installing psr/log (1.1.3): Downloading (100%)
     - Installing laminas/laminas-zendframework-bridge (1.1.1): Downloading (100%)
     - Installing laminas/laminas-escaper (2.6.1): Downloading (100%)
     - Installing kint-php/kint (3.3): Downloading (100%)
     - Installing codeigniter4/framework (v4.0.4): Downloading (100%)
     - Installing fzaninotto/faker (dev-master ac73e52): Cloning ac73e52870
   ```

2. Jalankan perintah `php spark serve` untuk menjalankan CI.

   ```
   D:\#Semester Ganjil 2020\Bahan Ajar\Praktikum RPL\latihan-php-unit\ci4> php spark serve
   ```
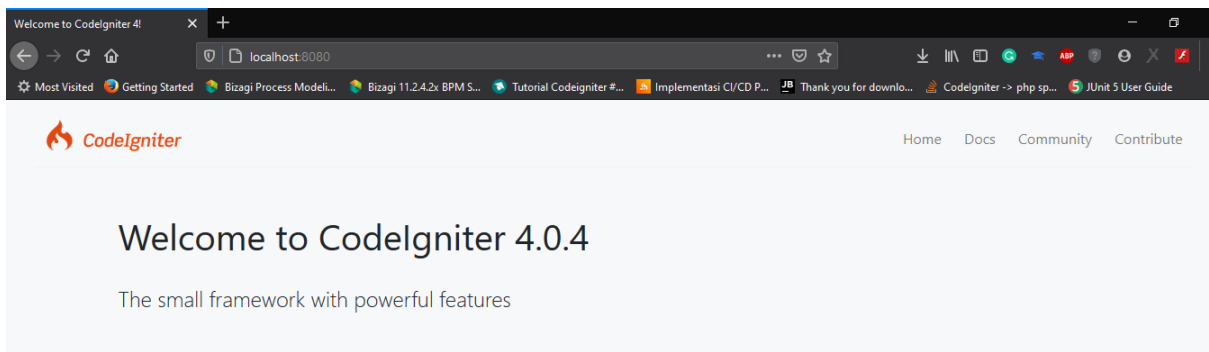
   Jika ada error, silahkan copy folder ci4 ke htdocs kemudian jalankan perintah berikut.

   ```
   C:\xampp\htdocs\ci4>php spark serve

   CodeIgniter CLI Tool - Version 4.0.4 - Server-Time: 2020-11-25 19:56:05pm

   CodeIgniter development server started on http://localhost:8080
   Press Control-C to stop.
   [Thu Nov 26 08:56:06 2020] PHP 7.4.12 Development Server (http://localhost:8080) started
   ```

   Kemudian buka http://localhost:8080 di browser

3. Instalasi Laravel dengan composer

```
composer create-project laravel/laravel --prefer-dist phpunit-tests
```

```
D:\#Semester Ganjil 2020\Bahan Ajar\Praktikum RPL\latihan-php-unit>composer create-project laravel/laravel --prefer-dist phpunit-tests
Creating a "laravel/laravel" project at "./phpunit-tests"
Installing laravel/laravel (v8.4.2)
  - Installing laravel/laravel (v8.4.2): Downloading (100%)
Created project in D:\#Semester Ganjil 2020\Bahan Ajar\Praktikum RPL\latihan-php-unit\phpunit-tests
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies (including require-dev)
```

Tunggu sampai prosesnya selesai.

Ada beberapa argumen yang kita berikan pada perintah ini:

- `create-project` adalah perintah untuk membuat proyek baru dengan composer;
- `laravel/laravel` adalah file Laravel yang akan di-download;
- `--prefer-dist` adalah kebalikan dari `--prefer-source`; composer akan menginstall dari *dist* jika mungkin. Ini dapat mempercepat penginstalan secara substansial pada pembangunan *server* dan kasus penggunaan lain di mana kita biasanya tidak menjalankan pembaruan dari *vendor*. Ini juga merupakan cara untuk menghindari masalah dengan *Git* jika kita tidak memiliki pengaturan yang tepat.
- `phpunit-tests` adalah nama proyek yang akan kita buat;

Creating a New Test in Laravel

1. Langkah pertama ketika menggunakan PHPUnit adalah membuat *test class.* Setiap *test class* diberi nama `<nama>Test.php`. Dengan format ini memperbolehkan PHPUnit untuk menemukan *test class* dan akan mengabaikan apapun yang tidak berakhiran `Test.php`.
2. Pada Laravel, ada 2 files di direktori `.\tests\`:

- `Feature/ExampleTest.php` dan `Unit/ExampleTest.php:ExampleTest.php` adalah contoh *test class* yang memuat kasus pengujian dasar *(basic test case)* menggunakan *application testing helpers* – abaikan untuk sekarang.

-

`TestCase.php` adalah file *bootstrap* untuk mensetting *environment* Laravel didalam pengujian kita. Hal ini memungkinkan kita untuk menggunakan Laravel *facades* dalam pengujian dan menyediakan kerangka kerja *(framework)* untuk membantu pengujian *(testing helpers)*.

3. Untuk membuat test class baru, kita bisa membuat file baru secara manual atau mengeksekusi perintah *Artisan* `make:test` yang disediakan oleh Laravel.

`php artisan make:test BasicTest` adalah *artisan command* yang digunakan untuk membuat *test class* bernama **BasicTest.php.** Sedangkan untuk melakukan pengujian gunakan *PHPUnit test command* sebagaimana berikut:

Laravel akan membuat *basic test class* sebagaimana berikut:

```php
BasicTest.php                                    ●
1   <?php
2
3   namespace Tests\Feature;
4
5   use Illuminate\Foundation\Testing\RefreshDatabase;
6   use Illuminate\Foundation\Testing\WithFaker;
7   use Tests\TestCase;
8
9   class BasicTest extends TestCase
10  {
11      /**
12       * A basic feature test example.
13       *
14       * @return void
15       */
16      public function testExample()
17      {
18          $response = $this->get('/');
19
20          $response->assertStatus(200);
21      }
22  }
```

4. Buat file **Box.php** di direktori **.\app** sebagaimana berikut **:**

```php
BasicTest.php        ●        Box.php              ●
1   <?php
2   # app/Box.php
3   namespace App;
4   class Box
5   {
6       /**
7        * @var array
8        */
9       protected $items = [];
10      /**
11       * Construct the box with the given items.
12       *
13       * @param array $items
14       */
15      public function __construct($items = [])
16      {
17          $this->items = $items;
18      }
19      /**
20       * Check if the specified item is in the box.
21       *
22       * @param string $item
23       * @return bool
24       */
25      public function has($item)
26      {
27          return in_array($item, $this->items);
28      }
29      /**
30       * Remove an item from the box, or null if the box is empty.
31       *
32       * @return string
33       */
34      public function takeOne()
35      {
36          return array_shift($this->items);
37      }
38      /**
39       * Retrieve all items from the box that start with the specified letter.
40       *
41       * @param string $letter
42       * @return array
43       */
44      public function startsWith($letter)
45      {
46          return array_filter($this->items, function ($item) use ($letter) {
47              return stripos($item, $letter) === 0;
48          });
49      }
50  }
```

5. Kemudian buka `.\tests\Feature\BasicTest.php` class yang dibuat pada no.3, dan hilangkan `testExample` method yang dibuat *by default*. Kemudian buatlah method/function `testBoxContents()` sebagaimana berikut:

```php
<?php

namespace Tests\Feature;

use Illuminate\Foundation\Testing\RefreshDatabase;
use Illuminate\Foundation\Testing\WithFaker;
use Tests\TestCase;
#Import Box class
use App\Box;

class BasicTest extends TestCase
{
    /**
     * A basic feature test example.
     *
     * @return void
     */
    /**
     *public function testExample()
     *{
     *      $response = $this->get('/');
     *
     *      $response->assertStatus(200);
     *}
     */
    #Test function for Box class
    public function testBoxContents()
    {
        $box = new Box(['toy']);
        $this->assertTrue($box->has('toy'));
        $this->assertFalse($box->has('ball'));
    }
}
```

Kemudian dieksekusi dengan *command testing* sebagaimana berikut:

```
D:\#Semester Ganjil 2020\Bahan Ajar\Praktikum RPL\latihan-php-unit\phpunit-tests>.\vendor\bin\phpunit

PHPUnit 9.4.3 by Sebastian Bergmann and contributors.

...                                                                 3 / 3 (100%)

Time: 00:25.824, Memory: 20.00 MB

OK (3 tests, 4 assertions)
```

```
D:\#Semester Ganjil 2020\Bahan Ajar\Praktikum RPL\latihan-php-unit\phpunit-tests>.\vendor\bin\phpunit
 --testdox tests\feature\BasicTest.php
PHPUnit 9.4.3 by Sebastian Bergmann and contributors.

Basic (Tests\Feature\Basic)
 Box contents

Time: 00:01.373, Memory: 18.00 MB

OK (1 test, 2 assertions)
```

6. Tambahkan method/function `testTakeOneFromTheBox()` di **BasicTest.php** sebagaimana berikut:

```php
36    public function testTakeOneFromTheBox()
37    {
38        $box = new Box(['torch']);
39        $this->assertEquals('torch', $box->takeOne());
40
41        # Null, now the box is empty
42        $this->assertNull($box->takeOne());
43    }
```

kemudian dieksekusi dengan *command testing* sebagaimana berikut:

```
D:\#Semester Ganjil 2020\Bahan Ajar\Praktikum RPL\latihan-php-unit\phpunit-tests>.\vendor\bin\phpunit
tests\feature\BasicTest.php
PHPUnit 9.4.3 by Sebastian Bergmann and contributors.

..                                                              2 / 2 (100%)

Time: 00:00.357, Memory: 18.00 MB

OK (2 tests, 4 assertions)

D:\#Semester Ganjil 2020\Bahan Ajar\Praktikum RPL\latihan-php-unit\phpunit-tests>.\vendor\bin\phpunit
PHPUnit 9.4.3 by Sebastian Bergmann and contributors.

....                                                            4 / 4 (100%)

Time: 00:00.414, Memory: 20.00 MB

OK (4 tests, 6 assertions)
```

7. Tambahkan method/function `testTakeOneFromTheBox()` di **BasicTest.php** sebagaimana berikut:

```php
47    public function testStartsWithALetter()
48    {
49        $box = new Box(['toy','torch','ball','cat','tissue']);
50        $results = $box->startsWith('t');
51
52        $this->assertCount(3,$results);
53        $this->assertContains('toy', $results);
54        $this->assertContains('torch', $results);
55        $this->assertContains('tissue', $results);
56
57        #Empty arry if passed even
58        $this->assertEmpty($box->startsWith('s'));
59    }
```

kemudian dieksekusi dengan *command testing* sebagaimana berikut:

```
D:\#Semester Ganjil 2020\Bahan Ajar\Praktikum RPL\latihan-php-unit\phpunit-tests>.\vendor\bin\phpunit
tests\feature\BasicTest.php
PHPUnit 9.4.3 by Sebastian Bergmann and contributors.

...                                                              3 / 3 (100%)

Time: 00:00.524, Memory: 18.00 MB

OK (3 tests, 9 assertions)

D:\#Semester Ganjil 2020\Bahan Ajar\Praktikum RPL\latihan-php-unit\phpunit-tests>.\vendor\bin\phpunit
PHPUnit 9.4.3 by Sebastian Bergmann and contributors.

.....                                                            5 / 5 (100%)

Time: 00:00.406, Memory: 20.00 MB

OK (5 tests, 11 assertions)
```

TUGAS

1. Screenshoot, analisa, dan jelaskan praktikum diatas (termasuk method PHPUnit yang digunakan untuk testing) !
2. Buatlah 1 file class dan 1 file testing menggunakan method PHPUnit *assertion* pada folder **test** CI4 kemudian lakukan unit test !
3. Screenshoot, analisa, dan jelaskan instalasi CI3 dan instalasi CI-PHPUnit-test!

   Instalasi dan membuat project CI3

```
D:\#Semester Ganjil 2020\Bahan Ajar\Praktikum RPL\latihan-php-unit>composer create-project codeigniter/framework dope
Creating a "codeigniter/framework" project at "./dope"
Installing codeigniter/framework (3.1.11)
  - Installing codeigniter/framework (3.1.11): Downloading (100%)
Created project in D:\#Semester Ganjil 2020\Bahan Ajar\Praktikum RPL\latihan-php-unit\dope
Deprecation warning: require-dev.mikey179/vfsStream is invalid, it should not contain uppercase characters. Please use mike
y179/vfsstream instead. Make sure you fix this as Composer 2.0 will error.
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 27 installs, 0 updates, 0 removals
  - Installing mikey179/vfsstream (v1.1.0): Downloading (100%)
  - Installing myclabs/deep-copy (1.10.1): Downloading (100%)
  - Installing sebastian/version (2.0.1): Downloading (100%)
  - Installing sebastian/resource-operations (1.0.0): Downloading (100%)
  - Installing sebastian/recursion-context (2.0.0): Downloading (100%)
  - Installing sebastian/object-enumerator (2.0.1): Downloading (100%)
  - Installing sebastian/global-state (1.1.1): Downloading (100%)
  - Installing sebastian/exporter (2.0.0): Downloading (100%)
  - Installing sebastian/environment (2.0.0): Downloading (100%)
  - Installing sebastian/diff (1.4.3): Downloading (100%)
  - Installing sebastian/comparator (1.2.4): Downloading (100%)
  - Installing symfony/polyfill-ctype (v1.20.0): Downloading (100%)
  - Installing symfony/yaml (v4.4.16): Downloading (100%)
  - Installing doctrine/instantiator (1.3.1): Downloading (100%)
  - Installing webmozart/assert (1.9.1): Downloading (100%)
  - Installing phpdocumentor/reflection-common (2.2.0): Downloading (100%)
  - Installing phpdocumentor/type-resolver (1.4.0): Downloading (100%)
  - Installing phpdocumentor/reflection-docblock (5.2.2): Downloading (100%)
  - Installing phpspec/prophecy (v1.10.3): Downloading (100%)
  - Installing phpunit/php-text-template (1.2.1): Downloading (100%)
  - Installing phpunit/phpunit-mock-objects (3.4.4): Downloading (100%)
  - Installing phpunit/php-timer (1.0.9): Downloading (100%)
  - Installing sebastian/code-unit-reverse-lookup (1.0.1): Downloading (100%)
  - Installing phpunit/php-token-stream (2.0.2): Downloading (100%)
  - Installing phpunit/php-file-iterator (1.4.5): Downloading (100%)
  - Installing phpunit/php-code-coverage (4.0.8): Downloading (100%)
  - Installing phpunit/phpunit (5.7.27): Downloading (100%)
codeigniter/framework suggests installing paragonie/random_compat (Provides better randomness in PHP 5.x)
sebastian/global-state suggests installing ext-uopz (*)
symfony/yaml suggests installing symfony/console (For validating YAML files using the lint command)
phpunit/phpunit-mock-objects suggests installing ext-soap (*)
phpunit/php-code-coverage suggests installing ext-xdebug (^2.5.1)
phpunit/phpunit suggests installing phpunit/php-invoker (~1.1)
```

   Jalankan CI3 dengan perintah berikut :

```
D:\#Semester Ganjil 2020\Bahan Ajar\Praktikum RPL\latihan-php-unit\dope>php -S localhost:3000
[Fri Nov  6 15:07:02 2020] PHP 7.4.10 Development Server (http://localhost:3000) started
[Fri Nov  6 15:07:18 2020] [::1]:64367 Accepted
[Fri Nov  6 15:07:18 2020] [::1]:64367 [200]: GET /
[Fri Nov  6 15:07:18 2020] [::1]:64367 Closing
```

4. Buatlah 1 file class dan 1 file testing menggunakan method PHPUnit *assertion* di CI3 kemudian lakukan unit test!

5. Swap method `assertFalse()` menjadi `assertTrue()` pada **BasicTest.php** praktikum no.5. Kemudian run *command testing.* Jelaskan kenapa hasilnya sebagai berikut:

```
D:\#Semester Ganjil 2020\Bahan Ajar\Praktikum RPL\latihan-php-unit\phpunit-tests>.\vendor\bin\phpunit
tests\feature\BasicTest.php
PHPUnit 9.4.3 by Sebastian Bergmann and contributors.

F                                                               1 / 1 (100%)

Time: 00:00.255, Memory: 18.00 MB

There was 1 failure:

1) Tests\Feature\BasicTest::testBoxContents
Failed asserting that true is false.

D:\#Semester Ganjil 2020\Bahan Ajar\Praktikum RPL\latihan-php-unit\phpunit-tests\tests\Feature\BasicTe
st.php:31

FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
```

6. Swap method `assertEquals()` menjadi `assertNull()` pada **BasicTest.php** praktikum no.6. Kemudian run *command testing.* Jelaskan kenapa hasilnya sebagai berikut:

```
D:\#Semester Ganjil 2020\Bahan Ajar\Praktikum RPL\latihan-php-unit\phpunit-tests>.\vendor\bin\phpunit
tests\feature\BasicTest.php
PHPUnit 9.4.3 by Sebastian Bergmann and contributors.

.F                                                              2 / 2 (100%)

Time: 00:00.267, Memory: 18.00 MB

There was 1 failure:

1) Tests\Feature\BasicTest::testTakeOneFromTheBox
torch
Failed asserting that 'torch' is null.

D:\#Semester Ganjil 2020\Bahan Ajar\Praktikum RPL\latihan-php-unit\phpunit-tests\tests\Feature\BasicTe
st.php:40

FAILURES!
Tests: 2, Assertions: 3, Failures: 1.
```

7. Analisa dan jelaskan method/function yang ada di BasicTest.php berikut:

```
D:\#Semester Ganjil 2020\Bahan Ajar\Praktikum RPL\latihan-php-unit\phpunit-tests>.\vendor\bin\phpunit
--testdox tests\Feature\BasicTest.php
PHPUnit 9.4.3 by Sebastian Bergmann and contributors.

Basic (Tests\Feature\Basic)
 ☑ Box contents
 ☑ Take one from the box
 ☑ Starts with a letter

Time: 00:00.288, Memory: 18.00 MB

OK (3 tests, 9 assertions)
```

8. Buatlah 1 file class dan 1 file testing menggunakan method PHPUnit *assertion* di Laravel kemudian lakukan unit test!