

MODUL 3

AUTOMATED TESTING

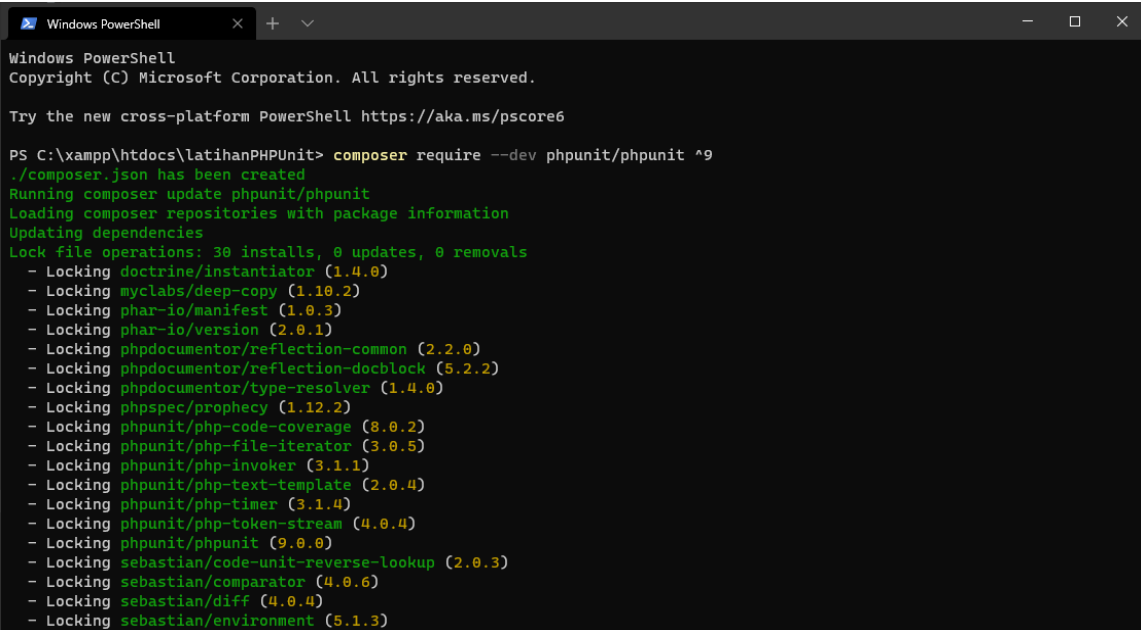
Automated Testing adalah sebuah metode pengujian proyek perangkat lunak yang menggunakan suatu *Software Testing* (terpisah dari sistem/aplikasi yang sedang diuji) dengan tujuan untuk membandingkan antara “output yang diharapkan” dengan “output yang dihasilkan” oleh sistem, sehingga dapat memastikan bahwa sistem yang dikembangkan sudah berjalan dengan baik. Salah satu jenis *automated testing* yang digunakan dalam pengujian proyek berbasis website adalah Laravel Dusk yang berjalan diatas PHPUnit. Pada modul 3 ini akan dibahas mengenai cara instalasi PHPUnit, instalasi Laravel Dusk, dan cara melakukan pengujian pada proyek yang telah dikembangkan.

1. PHPUnit

PHPUnit adalah kerangka pengujian unit untuk bahasa pemrograman PHP. Ini adalah instance dari arsitektur xUnit untuk framework pengujian unit yang berasal dari SUnit dan menjadi populer dengan JUnit. PHPUnit dibuat oleh Sebastian Bergmann dan pengembangannya dihosting di GitHub.

1.1 Instalasi PHPUnit

PHPUnit memerlukan PHP dengan versi 7.3 keatas. Instalasi PHPUnit menggunakan Composer dengan sintaks : **composer require --dev phpunit/phpunit ^9**

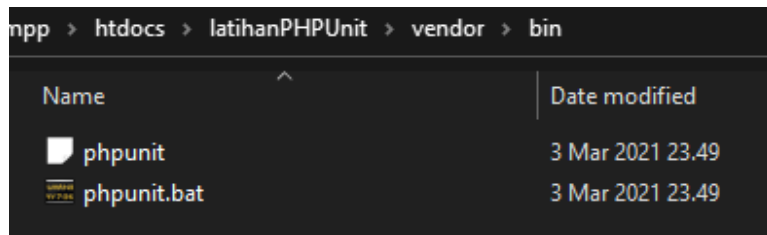


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

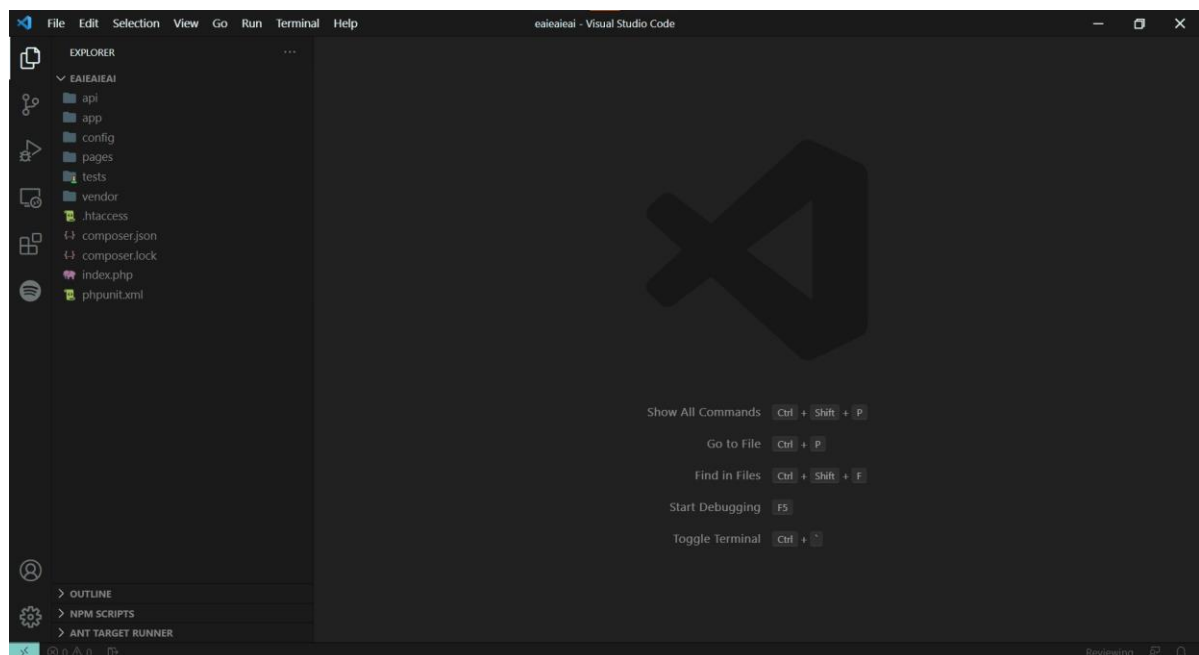
PS C:\xampp\htdocs\latihanPHPUnit> composer require --dev phpunit/phpunit ^9
./composer.json has been created
Running composer update phpunit/phpunit
Loading composer repositories with package information
Updating dependencies
Lock file operations: 30 installs, 0 updates, 0 removals
- Locking doctrine/instantiator (1.4.0)
- Locking myclabs/deep-copy (1.10.2)
- Locking phar-io/manifest (1.0.3)
- Locking phar-io/version (2.0.1)
- Locking phpdocumentor/reflection-common (2.2.0)
- Locking phpdocumentor/reflection-docblock (5.2.2)
- Locking phpdocumentor/type-resolver (1.4.0)
- Locking phpspec/prophecy (1.12.2)
- Locking phpunit/php-code-coverage (8.0.2)
- Locking phpunit/php-file-iterator (3.0.5)
- Locking phpunit/php-invoker (3.1.1)
- Locking phpunit/php-text-template (2.0.4)
- Locking phpunit/php-timer (3.1.4)
- Locking phpunit/php-token-stream (4.0.4)
- Locking phpunit/phpunit (9.0.0)
- Locking sebastian/code-unit-reverse-lookup (2.0.3)
- Locking sebastian/comparator (4.0.6)
- Locking sebastian/diff (4.0.4)
- Locking sebastian/environment (5.1.3)
```

Instalasi berhasil ketika file phpunit ada pada folder /vendor/bin

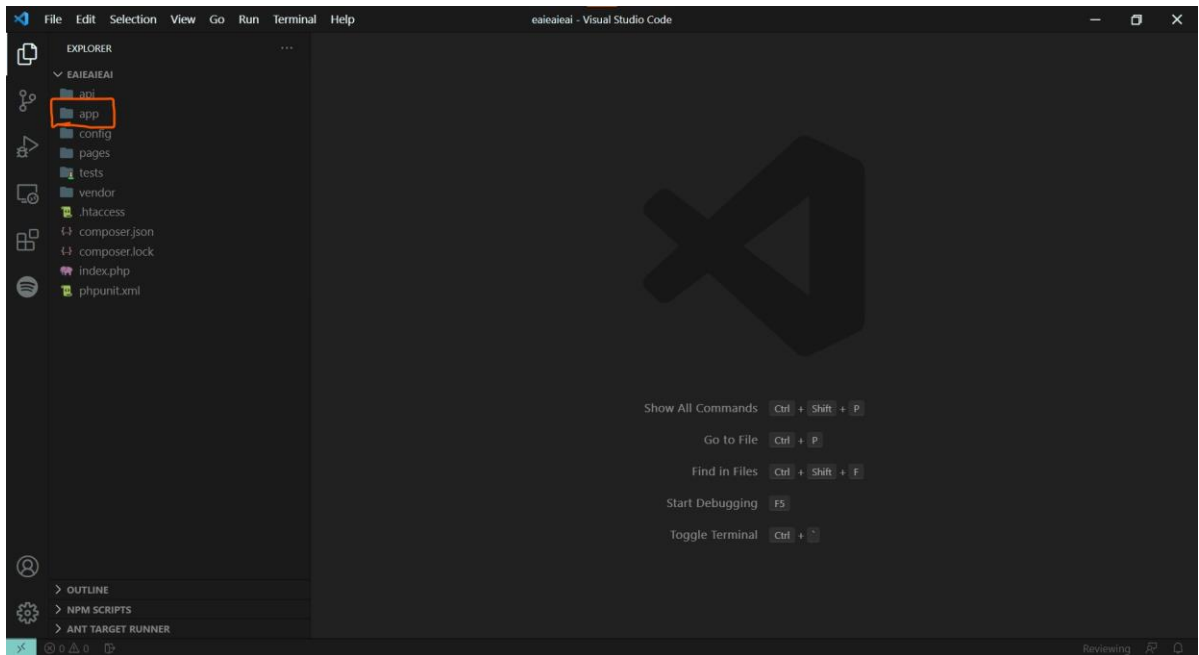


1.2 Pengujian menggunakan PHPUnit (Native)

Kali ini kita coba pengujian pertambahan menggunakan PHPUnit. Pertama, kita buka folder latihan dengan text editor (contoh disini menggunakan Visual Studio Code).



Kemudian, kita membuat folder bernama app dan tests untuk membuat fungsi-fungsi yang ingin di uji serta folder file pengujian.



Kemudian, kita mengubah isi file composer.json menjadi seperti ini.

```
{
    "require-dev": {
        "phpunit/phpunit": "9"
    },
    "autoload": {
        "psr-4": {
            "App\\": "app"
        }
    }
}
```

Selanjutnya, ketik dan jalankan perintah `composer dump-autoload -o` di Terminal.

```
PS C:\xampp\htdocs\eaieaieai> composer dump-autoload -o
Generating optimized autoload files
Class App\Login located in C:/xampp/htdocs/eaieaieai/app\1
.
Generated optimized autoload files containing 864 classes
PS C:\xampp\htdocs\eaieaieai>
```

Selanjutnya, buat sebuah file bernama `phpunit.xml` dan isikan file tersebut seperti dibawah ini.

```
<?xml version="1.0" encoding="UTF-8"?>

<phpunit      bootstrap="vendor/autoload.php"      colors="true"
verbose="true" stopOnFailure="false">
    <testsuites>
        <testsuite name="Test suite">
            <directory>tests</directory>
        </testsuite>
    </testsuites>
</phpunit>
```

Kita ingin mencoba melakukan login. Buatlah file pengujian dengan nama LoginTest.php di folder tests. Isikan file tersebut sebagai berikut.

```
<?php
use PHPUnit\Framework\TestCase;

class LoginTest extends TestCase
{
    public function test()
    {

    }
}
```

Asumsikan bahwa kita sudah memiliki akun. Kita membutuhkan email dan password serta fungsi login untuk melakukan login. Isikan file LoginTest.php sebagai berikut.

```
<?php
use PHPUnit\Framework\TestCase;

class LoginTest extends TestCase
{
    public function test()
    {
        $login = new \App\Login;
        $email = "nama@domain.co.id";
        $password = "katasandi";
        $login->setEmail($email);
        $login->setPassword($password);
        $this->assertEquals("Berhasil", $login->login());
    }
}
```

Coba kita jalankan pengujian menggunakan PHPUnit dengan cara ketikkan `.\vendor\bin\phpunit` di terminal.

```
PS C:\xampp\htdocs\eaieaieai> .\vendor\bin\phpunit
PHPUnit 9.0.0 by Sebastian Bergmann and contributors.

Runtime:       PHP 7.3.8
Configuration: C:\xampp\htdocs\eaieaieai\phpunit.xml

E                                                     1 / 1 (100%)

Time: 00:00.167, Memory: 4.00 MB

There was 1 error:

1) LoginTest::test
Error: Class 'App\Login' not found

C:\xampp\htdocs\eaieaieai\tests\LoginTest.php:8

ERRORS!
Tests: 1, Assertions: 0, Errors: 1.
PS C:\xampp\htdocs\eaieaieai>
```

Kita dapat melihat bahwa tidak ada class Login di folder App. Maka dari itu, buatlah kelas Login dan isikan file tersebut sebagai berikut.

```
<?php
namespace App;

class Login
{
    public $email;
    public $password;

    public function setEmail($email)
    {
        $this->email = $email;
    }

    public function setPassword($password)
    {
        $this->password = $password;
    }

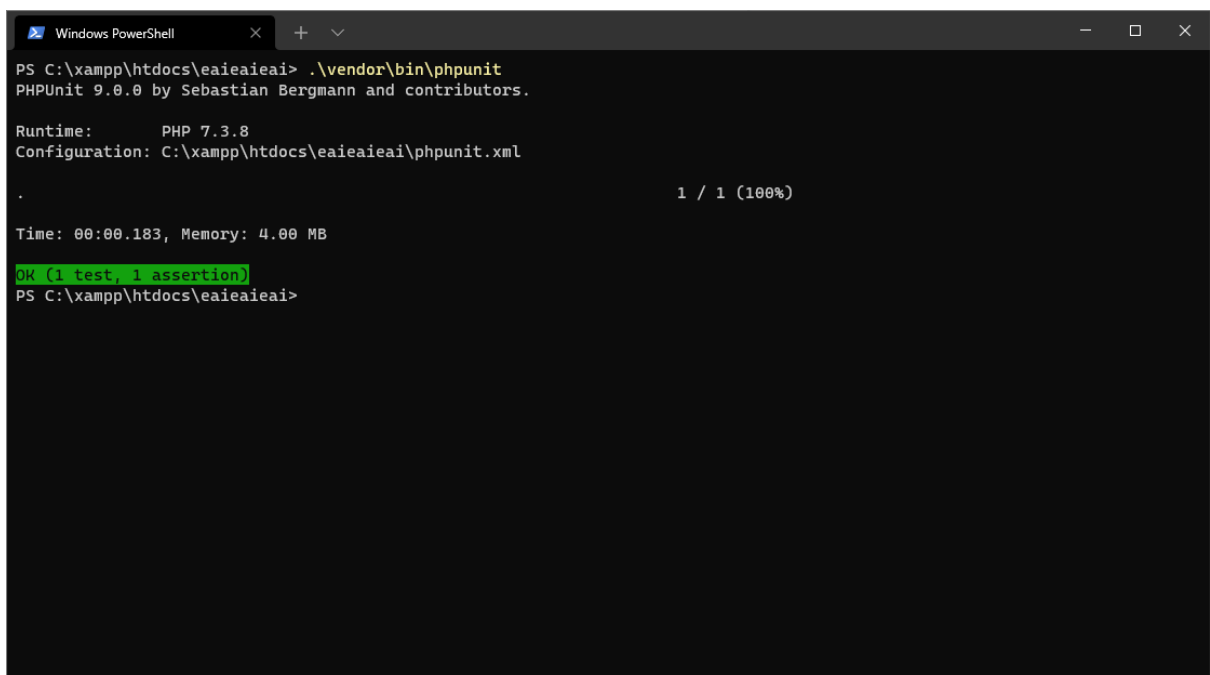
    public function login() : String
    {
        $email = $this->email;
        $password = $this->password;
```

```

    $con = mysqli_connect("localhost","root","","eai_sdm","3306");
    $sql = "SELECT * FROM `nama_tabel` WHERE email = '$email' AND
password = '$password'";
    $query = mysqli_query($con, $sql);
    $check = mysqli_num_rows($query);
    if ($check > 0) {
        return "Berhasil";
    } else {
        return "Gagal";
    }
}
}

```

Simpan file tersebut dan coba jalankan pengujian kembali.



```

Windows PowerShell
PS C:\xampp\htdocs\eaieaieai> .\vendor\bin\phpunit
PHPUnit 9.0.0 by Sebastian Bergmann and contributors.

Runtime:    PHP 7.3.8
Configuration: C:\xampp\htdocs\eaieaieai\phpunit.xml

.
1 / 1 (100%)

Time: 00:00.183, Memory: 4.00 MB

OK (1 test, 1 assertion)
PS C:\xampp\htdocs\eaieaieai>

```

Oke pengujian berhasil dilakukan. Tetapi, kalau kita lihat, source code yang telah dibuat masih terlihat kompleks, maka dari itu, kita coba untuk merestrukturisasi kembali agar source code kita terlihat lebih sederhana. Kita dapat memulai dari file Login.php, variabel yang dibutuhkan dapat ditempatkan sebagai parameter dari fungsi login. Maka file Login.php seperti berikut.

```

<?php
namespace App;

class Login

```

```

{
    public function login($email, $password) : String
    {
        $con = mysqli_connect("localhost","root","", "eai_sdm","3306");
        $sql = "SELECT * FROM `tb_karyawan` WHERE email = '$email' AND
password = '$password'";
        $query = mysqli_query($con, $sql);
        $check = mysqli_num_rows($query);
        if ($check > 0) {
            return "Berhasil";
        } else {
            return "Gagal";
        }
    }
}

```

Kemudian, kita ubah file LoginTest.php menjadi seperti berikut.

```

<?php
use PHPUnit\Framework\TestCase;

class LoginTest extends TestCase
{
    public function test()
    {
        $login = new \App\Login;
        $this->assertEquals("Berhasil", $login->login("nama@domain.co.id",
"katasandi"));
    }
}

```

Terakhir, kita coba untuk melakukan pengujian kembali.

```
Windows PowerShell
PS C:\xampp\htdocs\eaieaieai> .\vendor\bin\phpunit
PHPUnit 9.0.0 by Sebastian Bergmann and contributors.

Runtime:    PHP 7.3.8
Configuration: C:\xampp\htdocs\eaieaieai\phpunit.xml

.
1 / 1 (100%)

Time: 00:00.186, Memory: 4.00 MB

OK (1 test, 1 assertion)
PS C:\xampp\htdocs\eaieaieai> |
```

Pengujian berhasil.

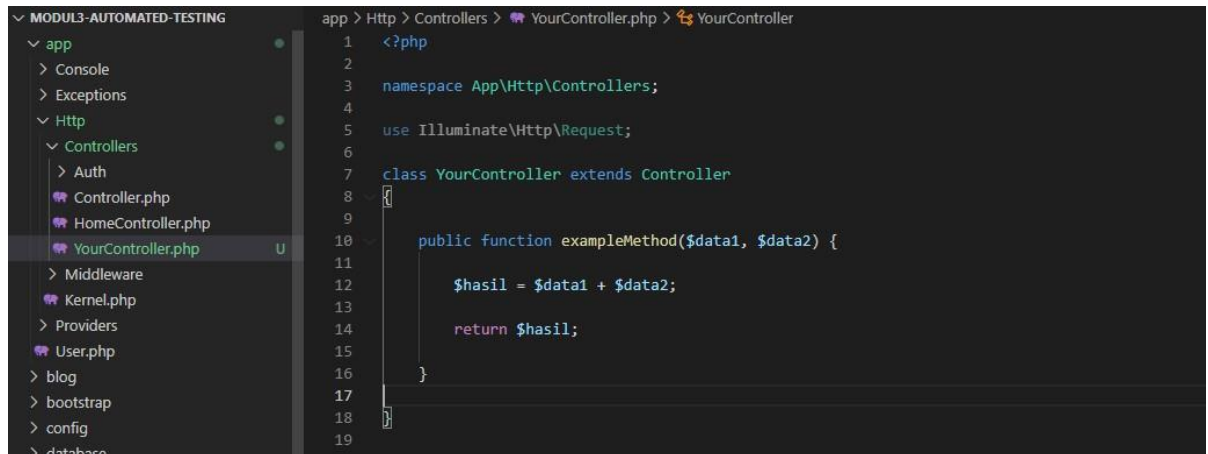
1.3 Pengujian menggunakan PHPUnit (Laravel)

Pertama, pastikan sudah membuat folder laravel atau project laravel dalam komputer lokal kalian.

```
MODUL3-AUTOMATED-TESTING
> app
> blog
> bootstrap
> config
> database
> public
> resources
> routes
> storage
> tests
> vendor
> editorconfig
> env
> env.example
> .gitattributes
> .gitignore
> phpunit.result.cache
> .styleci.yml
> artisan
> composer.json
> composer.lock
> ead_laravel.sql
> package-lock.json
> package.json
> phpunit.xml
> README.md
> server.php
> webpack.mix.js

tests > Unit > YourTest.php > testExample
1 <?php
2
3 namespace Tests\Unit;
4
5 use App\Http\Controllers\YourController;
6 use PHPUnit\Framework\TestCase;
7
8 class YourTest extends TestCase
9 {
10     /**
11      * A basic unit test example.
12      *
13      * @return void
14      */
15     public function testExample()
16     {
17         $this->assertTrue(true);
18     }
19
20     public function testMethod() {
21         $your = new YourController();
22         $hasil = $your->exampleMethod(4,4);
23         $this->assertEquals(8, $hasil);
24     }
25 }
26
```

Selanjutnya, pilihlah controller yang ingin di uji atau buatlah controller sehingga menjadi seperti berikut ini.

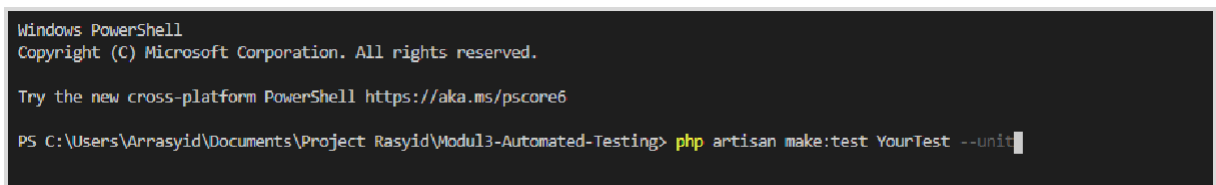


The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer is expanded to show the 'app' directory, which contains 'Http' and 'Controllers'. The 'Controllers' directory is further expanded, showing 'YourController.php' selected. The code editor displays the contents of 'YourController.php', which is a PHP class extending 'Controller' with a public method 'exampleMethod'.

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class YourController extends Controller
8 {
9
10     public function exampleMethod($data1, $data2) {
11
12         $hasil = $data1 + $data2;
13
14         return $hasil;
15     }
16 }
17
18
19
```

Pastikan kalian menghafalkan nama controller dan salah satu method yang ingin diuji di dalam project laravel kalian, contohnya dalam kasus ini adalah YourController sebagai controller dan method yang ingin diuji adalah exampleMethod dengan parameter data1 dan data2.

Kemudian buka terminal di dalam project laravel kalian dan buatlah file unit testing dengan cara mengetikkan **php artisan make:test YourTest.php --unit**



The screenshot shows a Windows PowerShell terminal window. The prompt is 'PS C:\Users\Arrasyid\Documents\Project Rasyid\Modul3-Automated-Testing>'. The command 'php artisan make:test YourTest --unit' has been entered and is highlighted.

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

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

PS C:\Users\Arrasyid\Documents\Project Rasyid\Modul3-Automated-Testing> php artisan make:test YourTest --unit
```

Kemudian buatlah sebuah pengujian dalam file tersebut dengan cara memanggil controller baru dengan contoh seperti **\$your = new YourController();**

```

<?php

namespace Tests\Unit;

use App\Http\Controllers\YourController;
use PHPUnit\Framework\TestCase;

class YourTest extends TestCase
{
    /**
     * A basic unit test example.
     *
     * @return void
     */
    public function testExample()
    {
        $this->assertTrue(true);
    }

    public function testMethod() {
        $your = new YourController();
        $hasil = $your->exampleMethod(4,4);
        $this->assertEquals(8, $hasil);
    }
}

```

Jalankan hasil method testing tersebut dengan sintax berikut ini

```

PS C:\Users\Arrasyid\Documents\Project Rasyid\Modul3-Automated-Testing> ./vendor/bin/phpunit
PHPUnit 8.5.15 by Sebastian Bergmann and contributors.

....                                                                4 / 4 (100%)

Time: 201 ms, Memory: 18.00 MB

OK (4 tests, 4 assertions)

```

Berdasarkan hasil dari gambar diatas bahwa pengujian yang ada dalam project laravel kalian berhasil semua (4/4)

2. Laravel Dusk

Laravel dusk adalah salah satu fitur testing dari laravel untuk pengujian aplikasi laravel secara otomatis. Salah satu kekurangan melakukan pengujian menggunakan PHPUnit adalah tidak bisa menguji fungsionalitas aplikasi yang ada JavaScript di sisi klien.

Dengan menggunakan Dusk bisa melakukan pengujian yang berjalan di Browser sehingga fitur JavaScript di sisi klien dapat diuji.

Dengan menggunakan Laravel Dusk kita dapat melakukan Browser Test atau *testing* terhadap aplikasi laravel dengan menggunakan *browser* dan semua *test case* yang sudah ditulis akan dijalankan oleh laravel dusk secara otomatis. Laravel Dusk memiliki beberapa method yang biasa digunakan, diantaranya sebagai berikut.

No	Method	Fungsi	Fungsi	Contoh
1	visit	Navigation	Digunakan untuk mengarahkan ke URL tertentu dalam aplikasi	<code>\$browser->visit('/login');</code>
2.	visitRoute	Navigation	Digunakan untuk mengarahkan ke route tertentu dalam aplikasi	<code>\$browser->visitRoute('login');</code>
3	type	Typing values	Digunakan untuk berinteraksi dengan elemen input.	<code>\$browser->type('email', 'taylor@laravel.com');</code>
4	select	Dropdown	Digunakan untuk memilih nilai tertentu yang tersedia pada elemen yang dipilih.	<code>\$browser->select('size', 'Large');</code>
5	check	Checkboxes	Digunakan untuk memilih (mencentang) satu atau beberapa nilai pada kotak centang yang dipilih.	<code>\$browser->check('terms');</code>
6	unchecked	Checkboxes	Digunakan untuk menghapus centang pada kotak centang tertentu.	<code>\$browser->unchecked('terms');</code>

7	radio	Radio Button	Digunakan untuk memilih opsi inputan yang disediakan.	<code>\$browser->radio('size', 'large');</code>
8	press	Pressing Buttons	Digunakan untuk menekan elemen tombol pada halaman	<code>\$browser->press('Login');</code>
9	assertPathIs	Assertions	Digunakan untuk memastikan path dari halaman yang dijalankan	<code>\$browser->assertPathIs('/home');</code>
10	assertSee	Assertions	Digunakan untuk memastikan bahwa teks yang diberikan terdapat pada halaman	<code>\$browser->assertSee(\$text);</code>

2.1 Instalasi Laravel Dusk

Penginstalan untuk laravel dusk sangat mudah dilakukan. Berikut adalah *requirement* yang harus ada untuk penginstalan Laravel Dusk.

- Composer
- Web Browser (Chrome / Firefox)
- Laravel Project

Pertama, kita menginstall laravel/dusk di dalam proyek yang ingin diuji dengan menggunakan sintaks berikut : **composer require --dev laravel/dusk**

Setelah menginstall *Dusk Package*, kita mengeksekusi dusk:install sehingga membuat otomatis direktori *tests/Browser* dan membuat *Example Test* untuk *Dusk*.

Eksekusi dusk:install dengan sintaks berikut : **php artisan dusk:install**

Selanjutnya Laravel Dusk sudah dapat digunakan, berikut adalah link untuk dokumentasi dari laravel dusk : <https://laravel.com/docs/8.x/dusk>

2.2 Pengujian menggunakan Laravel Dusk

Kita akan menguji fitur login dalam aplikasi web yang telah dikembangkan sebelumnya. Sebelum memulai pengujian pastikan bahwa server sudah dijalankan. Selanjutnya ikuti langkah - langkah dibawah ini untuk melakukan pengujian.

- a. Membuat file pengujian dengan mengetikan syntax sebagai berikut di terminal (cmd atau powershell) dalam folder proyek teman-teman : **php artisan dusk:make LoginTest**

```
PS C:\xampp\htdocs\MODUL5 ADHITYA> php artisan dusk:make LoginTest
Test created successfully.
```

- b. Membuka file yang telah dibuat dalam folder test\browser. Kemudian mencoba melakukan pengujian dengan file tersebut dengan mengetikan syntax : **php artisan dusk**

```
PS C:\xampp\htdocs\MODUL5 ADHITYA> php artisan dusk
Warning: TTY mode is not supported on Windows platform.
PHPUnit 8.5.15 by Sebastian Bergmann and contributors.

DevTools listening on ws://127.0.0.1:61937/devtools/browser/875bcf48-c835-44b2-a657-c3de988316d2
[0327/060637.802:INFO:CONSOLE(46362)] "Download the Vue Devtools extension for a better development experience:
https://github.com/vuejs/vue-devtools", source: http://127.0.0.1:8000/js/app.js (46362)
[0327/060637.803:INFO:CONSOLE(46371)] "You are running Vue in development mode.
Make sure to turn on production mode when deploying for production.
See more tips at https://vuejs.org/guide/deployment.html", source: http://127.0.0.1:8000/js/app.js (46371)
F
DevTools listening on ws://127.0.0.1:61968/devtools/browser/6f1656cf-b78c-4eb0-92fe-f4f0dd1b0f7c
[0327/060641.096:INFO:CONSOLE(46362)] "Download the Vue Devtools extension for a better development experience:
https://github.com/vuejs/vue-devtools", source: http://127.0.0.1:8000/js/app.js (46362)
[0327/060641.096:INFO:CONSOLE(46371)] "You are running Vue in development mode.
Make sure to turn on production mode when deploying for production.
See more tips at https://vuejs.org/guide/deployment.html", source: http://127.0.0.1:8000/js/app.js (46371)
F
2 / 2 (100%)

Time: 6.26 seconds, Memory: 18.00 MB

There were 2 failures:

1) Tests\Browser\ExampleTest::testBasicExample
Did not see expected text [Laravel] within element [body].
Failed asserting that false is true.

C:\xampp\htdocs\MODUL5 ADHITYA\vendor\laravel\dusk\src\Concerns\MakesAssertions.php:181
C:\xampp\htdocs\MODUL5 ADHITYA\vendor\laravel\dusk\src\Concerns\MakesAssertions.php:152
C:\xampp\htdocs\MODUL5 ADHITYA\tests\Browser\ExampleTest.php:20
C:\xampp\htdocs\MODUL5 ADHITYA\vendor\laravel\dusk\src\Concerns\ProvidesBrowser.php:68
C:\xampp\htdocs\MODUL5 ADHITYA\tests\Browser\ExampleTest.php:21

2) Tests\Browser\LoginTest::testExample
Did not see expected text [Laravel] within element [body].
Failed asserting that false is true.

C:\xampp\htdocs\MODUL5 ADHITYA\vendor\laravel\dusk\src\Concerns\MakesAssertions.php:181
C:\xampp\htdocs\MODUL5 ADHITYA\vendor\laravel\dusk\src\Concerns\MakesAssertions.php:152
C:\xampp\htdocs\MODUL5 ADHITYA\tests\Browser\LoginTest.php:20
C:\xampp\htdocs\MODUL5 ADHITYA\vendor\laravel\dusk\src\Concerns\ProvidesBrowser.php:68
C:\xampp\htdocs\MODUL5 ADHITYA\tests\Browser\LoginTest.php:21

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

Kita dapat lihat bahwa terjadi error pada kasus pengujian yang telah dibuat. Kalau dilihat, error terjadi karena tidak ditemukannya tulisan “Laravel” pada halaman yang kita uji.

- c. Kita lihat file pengujian yang dicoba oleh Laravel Dusk. Kita ambil salah satunya yaitu LoginTest.

```
<?php

namespace Tests\Browser;

use Illuminate\Foundation\Testing\DatabaseMigrations;
use Laravel\Dusk\Browser;
use Tests\DuskTestCase;

class LoginTest extends DuskTestCase
{
    /**
     * A Dusk test example.
     *
     * @return void
     */
    public function testExample()
    {
        $this->browse(function (Browser $browser) {
            $browser->visit('/')
                ->assertSee('Laravel');
        });
    }
}
```

Kita lihat bahwa yang dipastikan terdapat pada halaman yang diakses adalah tulisan “Laravel” (method `assertSee()`). Halaman yang diakses terdapat pada method `visit()`. Method tersebut mengakses halaman `(/)` yang artinya halaman utama (`http://127.0.0.1:8000/`). Pada pengujian ini halaman utama yang diakses dapat dilihat sebagai berikut.

LOGIN REGISTER



Dapat dilihat bahwa tidak ada tulisan “Laravel” pada halaman ini yang mengakibatkan error ketika menjalankan pengujian.

- d. Untuk memperbaikinya kita sesuaikan dengan halaman ini. Caranya, kita buka file pengujian `LoginTest.php`. Kemudian, kita ubah isian dari method `assertSee` menjadi “Enterprise Application Development”.

```

class LoginTest extends DuskTestCase
{
    /**
     * A Dusk test example.
     *
     * @return void
     */
    public function testExample()
    {
        $this->browse(function (Browser $browser) {
            $browser->visit('/')
                ->assertSee('Enterprise Application Development');
        });
    }
}

```

Kemudian kita coba jalankan Laravel Dusk.

```

PS C:\xampp\htdocs\MODUL5 ADHITYA> php artisan dusk
Warning: TTY mode is not supported on Windows platform.
PHPUnit 8.5.15 by Sebastian Bergmann and contributors.

DevTools listening on ws://127.0.0.1:59872/devtools/browser/5a955aab-eb32-48f7-be5a-629489d61402
F
DevTools listening on ws://127.0.0.1:59888/devtools/browser/2e0bd6ee-cedb-4613-bdcf-b8ae31328058
.
2 / 2 (100%)

Time: 5.3 seconds, Memory: 18.00 MB

There was 1 failure:

1) Tests\Browser\ExampleTest::testBasicExample
Did not see expected text [Laravel] within element [body].
Failed asserting that false is true.

C:\xampp\htdocs\MODUL5 ADHITYA\vendor\laravel\dusk\src\Concerns\MakesAssertions.php:181
C:\xampp\htdocs\MODUL5 ADHITYA\vendor\laravel\dusk\src\Concerns\MakesAssertions.php:152
C:\xampp\htdocs\MODUL5 ADHITYA\tests\Browser\ExampleTest.php:20
C:\xampp\htdocs\MODUL5 ADHITYA\vendor\laravel\dusk\src\Concerns\ProvidesBrowser.php:68
C:\xampp\htdocs\MODUL5 ADHITYA\tests\Browser\ExampleTest.php:21

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

```

Dapat dilihat bahwa terjadi error. File yang menyebabkan error adalah ExampleTest.php. Kita tidak mengubah file tersebut dan tidak menginginkan file tersebut dijalankan, maka kita menambahkan anotasi @group pada file LoginTest.php. @group berfungsi untuk menjalankan file tertentu yang terdapat dalam anotasi tersebut. Penulisan anotasi nya adalah @group [nama-group]. Jadi kita tambahkan @group login pada file LoginTest.php.

```

class LoginTest extends DuskTestCase
{
    /**
     * A Dusk test example.
     * @group Login ←
     * @return void
     */
    public function testExample()
    {
        $this->browse(function (Browser $browser) {
            $browser->visit('/')
                ->assertSee('Enterprise Application Development');
        });
    }
}

```

Jangan lupa di Save. Kemudian, kita coba jalankan Laravel Dusk dengan tambahan sintaks **--group=login**.

```

PS C:\xampp\htdocs\MODUL5 ADHITYA> php artisan dusk --group=login
Warning: TTY mode is not supported on Windows platform.
PHPUnit 8.5.15 by Sebastian Bergmann and contributors.

DevTools listening on ws://127.0.0.1:59986/devtools/browser/2581f395-5aed-4e67-a3c6-20b1ff8065ac
.
1 / 1 (100%)

Time: 2.73 seconds, Memory: 16.00 MB

OK (1 test, 1 assertion)

```

Pengujian bahwa halaman utama terdapat tulisan “Enterprise Application Development” berhasil dijalankan.

- e. Selanjutnya, kita akan coba menekan tombol Login pada halaman utama. Kita menggunakan method `click()` untuk menekan selector (element) pada halaman yang aktif. Untuk kasus ini, tombol login berupa element `<a>` maka kita perlu menambahkan sintaks pada file pengujian dengan : `click('a','login')`. Dan kita tambahkan `assertSee('Login')` untuk memastikan apakah halaman sudah berada di login atau belum. Teman-teman bisa mengganti `click()` menjadi `clickLink()`. Bedanya, `click` harus menuliskan selectornya sedangkan `clickLink` tidak perlu, `clickLink` mencari element dengan tag `<a>` dan menekannya.


```

public function testExample()
{
    $this->browse(function (Browser $browser) {
        $browser->visit('/')
            ->assertSee('Enterprise Application Development')
            ->click('a','login')
            ->assertSee('Login')
        ;
    });
}

```

Ketika dijalankan, pengujian berhasil dijalankan.

```

PS C:\xampp\htdocs\MODUL5 ADHITYA> php artisan dusk --group=login
Warning: TTY mode is not supported on windows platform.
PHPUnit 8.5.15 by Sebastian Bergmann and contributors.

DevTools listening on ws://127.0.0.1:55151/devtools/browser/eaab6c53-b851-49bd-8c98-67acf1e38199
[0327/101618.151:INFO:CONSOLE(46362)] "Download the Vue Devtools extension for a better development experience:
https://github.com/vuejs/vue-devtools", source: http://127.0.0.1:8000/js/app.js (46362)
[0327/101618.152:INFO:CONSOLE(46371)] "You are running Vue in development mode.
Make sure to turn on production mode when deploying for production.
See more tips at https://vuejs.org/guide/deployment.html", source: http://127.0.0.1:8000/js/app.js (46371)
.
1 / 1 (100%)

Time: 2.63 seconds, Memory: 16.00 MB

OK (1 test, 2 assertions)

```

- f. Selanjutnya, kita akan mengisi email dan password pada form login dan melakukan login. Caranya, menuliskan method type(name,isian). Pertama, kita mengisi emailnya dengan : type('email','a@a.com'). Kedua, kita mengisi passwordnya dengan type('password','asdadasd'). Ketiga, kita tulis method press('Login'). Method press() berfungsi untuk menekan sebuah tombol. Terakhir, kita pastikan bahwa url pada halaman ini sudah /home dengan assertPathIs('/home').

```

public function testExample()
{
    $this->browse(function (Browser $browser) {
        $browser->visit('/')
            ->assertSee('Enterprise Application Development')
            ->click('a','login')
            ->assertSee('Login')
            ->type('email', 'a@a.com')
            ->type('password', 'asdadasd')
            ->press('Login')
            ->assertPathIs('/home')
        ;
    });
}

```

- g Langkah terakhir, jalankan Laravel Dusk.

```
PS C:\xampp\htdocs\MODUL5 ADHITYA> php artisan dusk --group=login
Warning: TTY mode is not supported on Windows platform.
PHPUnit 8.5.15 by Sebastian Bergmann and contributors.

DevTools listening on ws://127.0.0.1:57472/devtools/browser/149b2214-20d0-41e1-8e5e-7c10b9ced628
[0327/130215.310:INFO:CONSOLE(46362)] "Download the Vue Devtools extension for a better development experience:
https://github.com/vuejs/vue-devtools", source: http://127.0.0.1:8000/js/app.js (46362)
[0327/130215.310:INFO:CONSOLE(46371)] "You are running Vue in development mode.
Make sure to turn on production mode when deploying for production.
See more tips at https://vuejs.org/guide/deployment.html", source: http://127.0.0.1:8000/js/app.js (46371)
[0327/130216.341:INFO:CONSOLE(46362)] "Download the Vue Devtools extension for a better development experience:
https://github.com/vuejs/vue-devtools", source: http://127.0.0.1:8000/js/app.js (46362)
[0327/130216.342:INFO:CONSOLE(46371)] "You are running Vue in development mode.
Make sure to turn on production mode when deploying for production.
See more tips at https://vuejs.org/guide/deployment.html", source: http://127.0.0.1:8000/js/app.js (46371)
.
1 / 1 (100%)

Time: 3.5 seconds, Memory: 16.00 MB

OK (1 test, 3 assertions)
```

Dapat dilihat bahwa pengujian login berhasil dijalankan.

STUDI KASUS

Dalam pengerjaan studi kasus ini, setiap individu akan melakukan pengujian satu *product backlog item* yang telah dikembangkan sebelumnya. Dalam pengujian tersebut, silahkan buat *test case* sesuai dengan *product backlog item* masing - masing.

Untuk tim yang tidak menggunakan *framework* laravel, silahkan ikuti langkah dibawah ini.

1. Buka link github berikut <https://github.com/david-william19/modul-3-automated-testing>
2. Lakukan *clone repository*
3. Buka project dan lakukan langkah - langkah yang sudah tertera pada deskripsi repository
4. Lakukan pengujian menggunakan Laravel Dusk sesuai langkah yang terdapat pada modul.