



Virtual Internship Experience

PHP: Using Library

Menggunakan Library

Dalam sebuah project PHP, penggunaan library bisa memudahkan proses penulisan kode. Namun, library yang digunakan bisa saja memiliki ketergantungan (*dependency*) dengan library lain. Nah, itulah kenapa Anda perlu PHP Composer.

Dengan PHP Composer, ketergantungan antar library dapat dikelola dengan baik sehingga script library bisa dijalankan tanpa error. Asalkan, Anda menggunakannya dengan benar.

Nah, pada artikel kali ini, kami akan membahas bagaimana cara menggunakan Composer untuk mengelola *dependency library* pada project PHP Anda.

Apa itu Composer?

Composer adalah tools yang digunakan untuk mengelola dependensi php. Tools ini bertindak sebagai penghubung antara project PHP dengan package library external.

Composer mampu mengelola instalasi dan melakukan update terhadap library yang Anda gunakan. Dengan PHP Composer, penulisan kode akan lebih cepat dan terstruktur.

Sebagai contoh, dalam suatu project PHP Anda membutuhkan fitur untuk mengirim email. Anda tidak mungkin membuat fitur tersebut dari awal kan? Selain perlu waktu yang lama, kode Anda juga akan dipenuhi dengan fungsi yang tidak terstruktur.

Jika menggunakan Composer, Anda bisa dengan mudah menggunakan library PHPMailer untuk membuat fitur kirim email dengan lebih cepat.

Menariknya, Composer memungkinkan Anda untuk bisa menggunakan berbagai library yang telah tersedia dalam satu project PHP. Ataupun, menggunakan versi berbeda dari library yang sama pada project PHP yang berbeda.

Sebagai tools yang *multi-platform*, Composer bisa digunakan di berbagai sistem operasi seperti Windows, Linux maupun MacOS. Hanya saja, Composer membutuhkan PHP 5.3.2+ agar bisa berjalan dengan baik pada komputer Anda.

Cara menggunakan PHP Composer sangatlah mudah, Anda hanya perlu mengikuti 6 langkah berikut ini :

1. Menginstall PHP Composer
2. Membuat Project PHP di httdocs
3. Inisialisasi Composer
4. Menginstall Package Library
5. Menggunakan Library
6. Menghapus Library

Menginstall PHP Composer

untuk menginstall composer silahkan menuju link berikut

<https://getcomposer.org/doc/00-intro.md#installation-linux-unix-macos>

untuk menginstall nya ada option windows juga tapi kali ini kita akan mencoba nya di enviroment linux

silahkan jalankan script dibawah :

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
```

```
php -r "if (hash_file('sha384', 'composer-setup.php') ===
```

```
'e0012edf3e80b6978849f5eff0d4b4e4c79ff1609dd1e613307e16318854d24ae64f26d1
```

```
7af3ef0bf7cfb710ca74755a') { echo 'Installer verified'; } else { echo 'Installer corrupt';
```

```
unlink('composer-setup.php'); } echo PHP_EOL;"
```

```
php composer-setup.php
```

```
php -r "unlink('composer-setup.php');"
```

```
mfr@mfr@192 ~ - ssh - 80x24
Last login: Fri Dec 24 10:52:06 on console
+ ~ composer

Composer version 2.0.11 2021-02-24 14:57:23

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display this help message
  -q, --quiet               Do not output any message
  -V, --version             Display this application version
  --ansi                   Force ANSI output
  --no-ansi                Disable ANSI output
  -n, --no-interaction      Do not ask any interactive question
  --profile                Display timing and memory usage information
  --no-plugins              Whether to disable plugins.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
```


Inisialisasi Composer

Untuk melakukan inisialisasi Composer, masuklah ke folder yang baru saja Anda buat. Lalu, buka terminal pada folder tersebut. Ketik `composer init` dan tekan Enter untuk menjalankan perintah inisialisasi.

```
→ composer init

Welcome to the Composer config generator.

This command will guide you through creating your composer.json config.

Package name (<vendor>/<name>) [mfr/composer]: investree/tutor_composer
Description []: Cara menggunakan php composer untuk pemula
Author [, n to skip]: rakamin
Invalid author string. Must be in the format: John Smith <john@example.com>
Author [, n to skip]: rakamin <rakamin@gmail.com>
Minimum Stability []:
Package Type (e.g. library, project, metapackage, composer-plugin) []: project
License []: GPL

Define your dependencies.

Would you like to define your dependencies (require) interactively [yes]? no
Would you like to define your dev dependencies (require-dev) interactively [yes]? no

{
  "name": "investree/tutor_composer",
  "description": "Cara menggunakan php composer untuk pemula",
  "type": "project",
  "license": "GPL",
  "authors": [
    {
      "name": "rakamin",
      "email": "rakamin@gmail.com"
    }
  ],
  "require": {}
}

Do you confirm generation [yes]? yes
→ composer ls
composer.json
→ composer
```

Anda akan diminta untuk mengisi detail file Composer seperti pada gambar di atas. Yang perlu Anda isikan adalah:

- nama *package*
- *description*
- *author*
- *package type*
- *license*
- *define dependencies*.

Perhatikan penulisan nama author ya, formatnya harus sesuai yaitu Nama Anda <emailanda@example.com>.

Pada bagian pertanyaan *define dependencies*, jawablah dengan mengetikkan no pada setiap pertanyaan. Hal ini dilakukan karena belum ada package yang akan diinstall pada tahap inisialisasi.

Jika sudah selesai melakukan inisialisasi, konfirmasi perintah **composer init** dengan mengetikkan yes pada pertanyaan terakhir yang muncul di terminal Anda. Perintah ini akan menghasilkan file composer.json yang isinya kurang lebih seperti ini :

```
composer init
+ composer composer init

Welcome to the Composer config generator

This command will guide you through creating your composer.json config.

Package name (<vendor>/<name>) [mfr/composer]: investree/tutor_composer
Description []: Cara menggunakan php composer untuk pemula
Author [, n to skip]: rakamin
Invalid author string. Must be in the format: John Smith <john@example.com>
Author [, n to skip]: rakamin <rakamin@gmail.com>
Minimum Stability []:
Package Type (e.g. library, project, metapackage, composer-plugin) []: project
License []: GPL

Define your dependencies.

Would you like to define your dependencies (require) interactively [yes]? no
Would you like to define your dev dependencies (require-dev) interactively [yes]? no

{
  "name": "investree/tutor_composer",
  "description": "Cara menggunakan php composer untuk pemula",
  "type": "project",
  "license": "GPL",
  "authors": [
    {
      "name": "rakamin",
      "email": "rakamin@gmail.com"
    }
  ],
  "require": {}
}

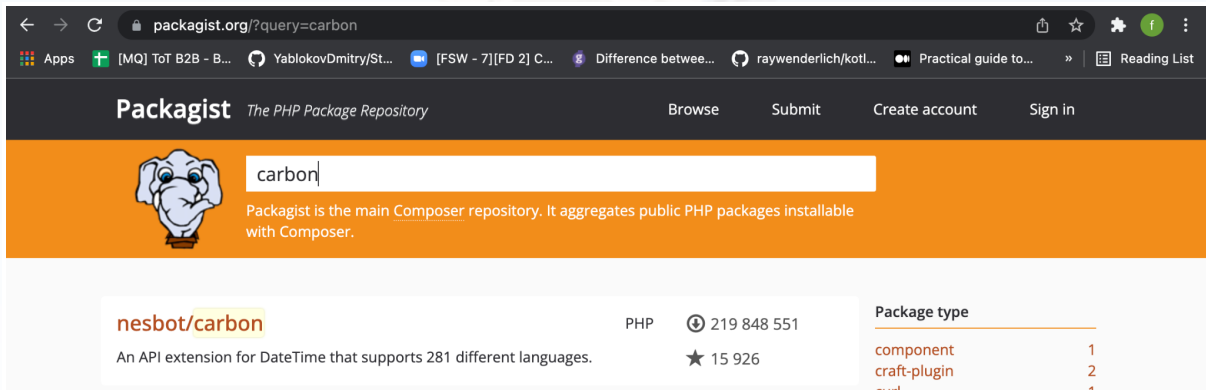
Do you confirm generation [yes]? yes
+ composer ls
composer.json
+ composer
```

Jika Anda perhatikan, bagian require masih kosong kan? Artinya, belum ada package yang terinstall pada project Anda. Anda dapat mulai mencoba menginstall package library pada langkah selanjutnya.

Menginstall Package Library

Composer memiliki repository sendiri yaitu packagist.org. Packagist merupakan kumpulan public PHP package yang tersimpan di Github dan bisa digunakan secara bebas untuk project PHP Anda.

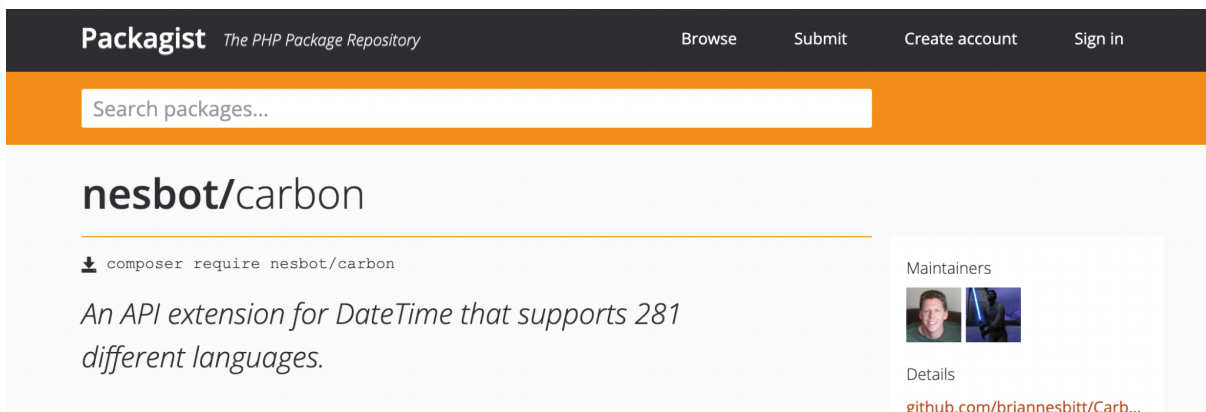
Untuk menginstall package dari packagist, carilah nama package yang akan Anda gunakan di kolom pencarian. Pada tutorial kali ini, kami akan coba menginstall package library carbon yang menyediakan fungsi pemrosesan waktu dan tanggal.



The screenshot shows the Packagist website with a search bar containing 'carbon'. Below the search bar, the results for 'nesbot/carbon' are displayed. The package is described as 'An API extension for DateTime that supports 281 different languages.' It has 219,848,551 downloads and 15,926 stars. The package type is listed as 'component'.

Jika sudah menemukan package yang sesuai, klik nama package tersebut. Pada halaman selanjutnya, Anda akan menemukan perintah Composer untuk menginstall package carbon.

composer require nesbot/carbon



The screenshot shows the details page for the 'nesbot/carbon' package. The package is described as 'An API extension for DateTime that supports 281 different languages.' The command 'composer require nesbot/carbon' is shown. The maintainers section lists 'briannesbitt' and 'Carb...'. The details section shows the GitHub repository 'github.com/briannesbitt/Carb...'. The package type is listed as 'component'.

Salin dan jalankan perintah tersebut pada terminal project PHP Anda seperti pada gambar berikut ini :

```
composer.json
[→ composer composer require nesbot/carbon
Using version ^2.55 for nesbot/carbon
./composer.json has been updated
Running composer update nesbot/carbon
Loading composer repositories with package information
Updating dependencies
Lock file operations: 6 installs, 0 updates, 0 removals
- Locking nesbot/carbon (2.55.2)
- Locking symfony/deprecation-contracts (v2.5.0)
- Locking symfony/polyfill-mbstring (v1.23.1)
- Locking symfony/polyfill-php80 (v1.23.1)
- Locking symfony/translation (v5.4.1)
- Locking symfony/translation-contracts (v2.5.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 6 installs, 0 updates, 0 removals
- Downloading symfony/translation-contracts (v2.5.0)
- Downloading symfony/deprecation-contracts (v2.5.0)
- Downloading symfony/translation (v5.4.1)
- Downloading nesbot/carbon (2.55.2)
- Installing symfony/translation-contracts (v2.5.0): Extracting archive
- Installing symfony/polyfill-php80 (v1.23.1): Extracting archive
- Installing symfony/polyfill-mbstring (v1.23.1): Extracting archive
- Installing symfony/deprecation-contracts (v2.5.0): Extracting archive
- Installing symfony/translation (v5.4.1): Extracting archive
- Installing nesbot/carbon (2.55.2): Extracting archive
3 package suggestions were added by new dependencies, use `composer suggest` to see details.
Generating autoload files
6 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
→ composer
```

Proses instalasi package library akan berlangsung beberapa saat, tergantung dengan kecepatan koneksi internet Anda.

Jika proses instalasi sudah selesai, akan ada folder baru bernama vendor yang terbuat pada direktori project Anda. Nantinya, folder ini akan berisi file dari semua library yang di download oleh Composer.

```
→ composer ls
composer.json composer.lock vendor
→ composer
```

Instalasi package library juga akan mengubah script pada composer.json. Bagian require yang semula kosong, kini akan berisi nama dan versi package yang sudah diinstall.


```
Welcome composer.json ×
Users > mfr > Documents > rakamin > composer > composer.json > ...
1
2 "name": "investree/tutor_composer",
3 "description": "Cara menggunakan php composer untuk pemula",
4 "type": "project",
5 "license": "GPL",
6 "authors": [
7     {
8         "name": "rakamin",
9         "email": "rakamin@gmail.com"
10    }
11 ],
12 "require": {
13     "nesbot/carbon": "^2.55"
14 }
15
16
```

Menggunakan Library

Untuk menggunakan library carbon yang baru saja diinstall, mari buat sebuah file sederhana dengan nama datetime.php pada direktori project Anda. Kemudian, salinlah kode berikut ini pada file tersebut :

```
1 <img alt="php icon" data-bbox="165 465 180 475"/>?php
2
3 require 'vendor/autoload.php';
4
5 use Carbon\Carbon;
6
7 $Jakarta_now = Carbon::now('Asia/Jakarta');
8
9 echo "Sekarang di Jakarta: $Jakarta_now ";
10
11 <img alt="run icon" data-bbox="165 570 180 580"/>
```

PROBLEMS DEBUG CONSOLE ... Filter (e.g. text, lexclude)

Sekarang di Jakarta: 2021-12-25 11:08:34

maka kamu akan mendapatkan hasil waktu sekarang di jakarta .

Menghapus Library

Jika Anda ingin menghapus library yang tidak digunakan lagi, jalankanlah perintah berikut ini :

```
composer remove <vendor>/<library>
```

Perintah tersebut akan menghapus semua file library yang terletak pada folder vendor. Jangan lupa untuk menjalankan perintah composer update, jika melakukan perubahan pada Composer, ya.