



IDLaravel

Belajar Laravel Ala Indonesia

- [Download](#)

Mengenal Namespace: Menjelajah Milyaran Galaksi

The Milky Way

Jika Anda pernah belajar Java atau C#, maka namespace bukan merupakan hal baru. Tapi jika Anda seorang programmer fanatik PHP (saking fanatiknya tidak tertarik belajar bahasa pemrograman yang lain) maka [namespace](#), yang baru muncul di [PHP versi 5.3](#), merupakan mainan baru. Dan sebagaimana layaknya mainan baru, kita pasti penasaran apa sih yang bisa dilakukan mainan baru bernama namespace ini. So, mari kita jelajahi satu persatu.

Apa Itu Namespace?

Apakah Anda pernah mengalami tidak bisa membuat sebuah `class` hanya karena namanya sudah dipakai di tempat lain? Apakah Anda pernah menulis kode seperti di bawah ini hanya untuk memastikan bahwa nama `function` yang Anda buat tidak bentrok dengan `function` di tempat lain?

```
if ( ! function_exists('format_price'))
{
    function format_price($number)
    {
        // code here
    }
}
```

Atau Anda pernah menemukan nama `class` yang sangat panjang dan kelihatan seperti jemuran pakaian seperti ini:

```
class Zend_Cache_Backend_Apc extends Zend_Cache_Backend implements Zend_Cache_Backend_ExtendedInterface
{
    ...
}
```

Jika Anda pernah mengalami salah satu kejadian di atas, maka Anda patut bersyukur sekarang karena bahasa yang kita cintai ini akhirnya mengizinkan kita untuk menjadi programmer yang lebih berkelas. Dengan namespace, Anda tidak perlu takut `class` yang Anda buat bentrok dengan `class` yang lain. Dengan namespace, Anda tidak perlu takut `function` yang Anda buat memiliki nama yang sama dengan `function` yang lain. Dengan namespace, Anda bisa memberi nama `class` dengan lebih singkat dan jelas, tidak perlu lagi menambahkan prefix **Embel_embel_ga_penting_** yang seperti jemuran. Dengan namespace, Anda bisa mengorganisasi kode dengan lebih rapi dan terstruktur, sehingga lebih mudah di-*maintenance*.

Cara kerja namespace mirip dengan cara kerja folder dan file. Dalam satu folder kita tidak bisa membuat 2 buah file dengan nama yang sama. Tetapi hal tersebut bisa dilakukan jika foldernya berbeda. Misal Anda mempunyai folder C:\Users\tono dan C:\Users\guest, maka di masing-masing folder tersebut Anda bisa membuat file dengan nama yang sama, misalnya suratcinta.txt. Selanjutnya Anda bisa merefer kedua file tersebut dengan C:\Users\tono\suratcinta.txt dan C:\Users\guest\suratcinta.txt. Jika tidak percaya silakan dicoba sendiri :) Namespace menjadi semacam pembungkus (encapsulation) untuk kode-kode di dalamnya, sehingga masing-masing kode memiliki identifier yang unik.

Bagaimana Membuat Namespace?

Untuk mempelajari bagaimana membuat namespace, mari kita praktekkan dengan contoh sederhana. Buat beberapa file php dengan struktur seperti berikut:

```
Australia
    Person.php
Indonesia
    Person.php
index.php
```

File Australia/Person.php:

```
<?php

class Person {

    function talk()
    {
        echo 'Welcome, Bro';
    }
}
```

```
}  
}
```

File Indonesia/Person.php:

```
<?php  
  
class Person {  
  
    function talk()  
    {  
        echo 'Selamat Datang, Bro';  
    }  
}
```

File index.php:

```
<?php  
  
include 'Australia/Person.php';  
include 'Indonesia/Person.php';  
  
//tono, orang Indonesia  
$tono = new Person;  
$tono->talk();  
  
echo '<br />';  
  
//alex, orang Australia  
$alex = new Person;  
$alex->talk();
```

Jika kita jalankan file index.php, maka akan muncul error seperti berikut:

```
Fatal error: Cannot redeclare class Person in /Users/uyab/Sites/snippet/namespace/Indonesia/Person.php on line 3
```

Yup, [malaikat juga tahu](#) kalau kita tidak boleh membuat class dengan nama yang sama. Eits, tapi itu dulu. Sekarang hal tersebut mungkin dilakukan dengan menggunakan namespace. Saya tidak tahu apakah malaikat mengikuti perkembangan PHP dan sudah mengetahui hal ini.

Untuk mengatasi error di atas, mari kita tambahkan namespace ke masing-masing class.

File Australia/Person.php:

```
<?php  
  
namespace Australia; // deklarasi namespace  
  
class Person {  
  
    function talk()  
    {  
        echo 'Welcome, Bro';  
    }  
}
```

File Indonesia/Person.php:

```
<?php  
  
namespace Indonesia;  
  
class Person {  
  
    function talk()  
    {  
        echo 'Selamat Datang, Bro';  
    }  
}
```

Refresh browser Anda, masih error bukan? Hehe, sama seperti file suratcinta.txt tadi, ketika Anda menggunakan namespace (membungkus file dengan folder), maka untuk mengakses semua resource dalam namespace tersebut Anda harus menulisnya dengan full path. Pada contoh di atas, jika hanya ditulis `Person` saja maka akan terjadi kebingungan, yang dimaksud `Person` yang mana. Apakah `Person` yang ada di Australia atau `Person` yang ada di Indonesia?

Untuk itu kita modif sedikit file index.php menjadi seperti ini:

```
<?php  
  
include 'Australia/Person.php';
```

```
include 'Indonesia/Person.php';

//tono, orang Indonesia
$tono = new \Indonesia\Person;
$tono->talk();

echo '<br />';

//alex, orang Australia
$alex = new \Australia\Person;
$alex->talk();
```

Refresh kembali browser Anda dan perhatikan hasilnya.

```
Selamat Datang, Bro
Welcome, Bro
```

Selamat, Anda telah berhasil membuat namespace pertama Anda di PHP. Jika Anda penasaran (sama seperti saya) kenapa menggunakan karakter backslash '\' sebagai karakter penanda namespace, silakan baca [penjelasan resmi dari PHP](#).

Namespace Resolution

Ok, untuk membuat penjelajahan kali ini lebih menarik, sekarang kita buat satu file lagi dengan nama zombie.php (di folder yang sama dengan index.php).

```
<?php

class Person {

    function talk()
    {
        echo 'Huarrrrr, grrhhhaahhh';
    }
}
```

Lalu tambahkan namespace dan include file zombie tadi di index.php:

```
<?php
namespace Universe;

include 'Australia/Person.php';
include 'Indonesia/Person.php';
include 'zombie.php';
```

Yup, sekarang semua kode sudah memiliki namespace masing-masing, kecuali zombie.php, karena memang zombie itu tidak jelas dan tidak ada namespace manapun yang mau menerimanya.

Selanjutnya mari kita tambahkan kode berikut di index.php:

```
//zombie, tidak tahu harus masuk ke namespace apa
$zombie = new Person;
$zombie->talk();

echo '<br />';
```

Refresh browser Anda, dan sekali lagi... error. Benar kan apa kata saya, zombie memang tidak jelas dan selalu hanya membuat susah.

```
Fatal error: Class 'Universe\Person' not found in /Users/uyab/Sites/snippet/namespace/index.php on line 9
```

Ada 2 cara yang bisa dilakukan untuk membuat kode berjalan dengan benar. Pertama, buang namespace yang ada di index.php:

```
// namespace Universe;
```

Atau cara kedua, biarkan namespace Universe tetap ada tetapi tambahkan backslash ketika membuat obyek Person dari zombie.php:

```
$zombie = new \Person;
```

Begini penjelasannya. Pada dasarnya, tanpa mendeklarasikan namespace apapun, PHP sudah mengenal yang namanya **global namespace**. Semua kode yang kita tulis akan masuk ke global namespace. Kemudian ketika kita mendeklarasikan namespace secara eksplisit (dengan menyetikkan keyword namespace), maka kode selanjutnya yang kita tulis akan otomatis pindah dari global namespace menuju namespace buatan kita sendiri. Ibarat file, maka sudah beda folder, jadi untuk bisa saling mengakses harus menyetikkan full path yang sesuai.

Pada contoh di atas, kita tahu bahwa zombie tidak memiliki namespace, maka zombie secara otomatis masuk ke **global namespace**. Nah, jika index.php kita tambahkan namespace Universe maka otomatis antara index.php dan zombie.php sudah beda namespace, yang satu Universe yang satu global. Oleh karena itu, agar index.php bisa mengakses class Person dari zombie.php, maka harus ditambahkan 'backslash' di depan, yang artinya class Person yang ingin dipanggil adalah class Person dari global namespace.

Sebenarnya ada cara ketiga untuk membuat kode yang error tadi berjalan benar. Ketika kita menuliskan `$zombie = new Person`, maka PHP akan mencari `class Person` di namespace yang sama dengan tempat kita memanggilnya. Karena kita memanggilnya dari namespace `Universe`, maka kita harus tambahkan `class Person` di namespace `Universe` tersebut.

Edit kembali file `index.php` dan tambahkan deklarasi `class Person` di bagian bawah seperti berikut:

```
...  
  
class Person  
{  
    function talk()  
    {  
        echo 'Pidipippp pip pip, saya person dari universe';  
    }  
}
```

Selanjutnya kita perlu instansiasi `class Person` tersebut, tambahkan kode berikut dimanapun di `index.php`:

```
$alien = new Person; // perhatikan, tanpa backslash didepan  
$alien->talk();
```

Coba lihat hasilnya di browser. Selamat, Anda sudah berhasil berbicara dalam 4 bahasa: Indonesia, Australia, zombie, dan alien ;)

Bonus

Buat sebuah file dengan nama `anotherUniverse.php` di folder yang sama dengan `index.php`, lalu pindahkan deklarasi `class Person` yang ada di `index.php` ke file tersebut.

```
<?php  
  
namespace Universe;  
  
class Person  
{  
    function talk()  
    {  
        echo 'Pidipippp pip pip, saya person dari universe';  
    }  
}
```

Lalu edit `index.php` agar meng-*include* file tersebut:

```
<?php  
  
namespace Universe;  
  
include 'Australia/Person.php';  
include 'Indonesia/Person.php';  
include 'zombie.php';  
include 'anotherUniverse.php';
```

Refresh kembali browser Anda, apa yang terjadi?

3 Komentar Laravel Indonesia**Masuk** **Rekomendasikan**  **Tweet**  **Bagikan****Urut dari yang terbaik**



Ikut berdiskusi...

MASUK DENGAN**ATAU DAFTAR DISQUS** 



Nama

**rudi** • 3 tahun yang lalu





wah terimakasih mas.. atas tulisannya.. sagat bermanfaat.. bahasanya bagus..baget.. (y) :)

  • **Balas** • **Bagikan** ›**nadiar429** • 3 tahun yang lalu

Done, asik banget tulisannya, mau dong kenal sama penulisnya

  • **Balas** • **Bagikan** ›**Bayu Hendra Winata** **Mod** ➔ **nadiar429** • 3 tahun yang lalu

Halo, salam kenal....

  • **Balas** • **Bagikan** › **Langganan**  **Pasang Disqus di websitemu** **Tambahkan Disqus** **Tambahkan****Laravel Indonesia**Berbagi tutorial, tips, *best practice*, dan cerita tentang PHP dan [Laravel](#).**Yuk Kontribusi**Semua boleh nyumbang ilmu, semua boleh sharing link yang menarik tentang Laravel. [Cara berkontribusi](#).**Sponsor**Terima kasih kepada [Javan Cipta Solusi](#) dan [Niagahoster](#) karena telah men-sponsori website ini.Kritik, saran, dan tawaran kerja sama atau kolaborasi bisa dikirimkan ke alamat email **id.laravel@gmail.com**.**Sumber Ilmu**

- [Laravel.com](#) ~ Website resmi
- [Laravel News](#) ~ Artikel pilihan
- [Laracasts](#) ~ Screencast
- [Grup Facebook](#)

Mitra Hosting



Beli Hosting Diskon

50%

+ Gratis
Domain

.com
powered by VPSBISIDN

BELI SEKARANG

Website ini ditenagai oleh [Github Pages](#) dan [Jekyll](#).

