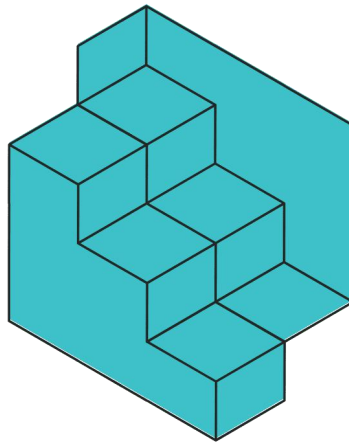


# **MANUAL BOOK OFI PHP Framework**

**Bahasa Indonesia**



**OFI PHP**  
Framework

**Version 1.1**

**2020**

## **KATA PENGANTAR**

Puji syukur kami sampaikan kehadiran Allah swt, karena atas berkah dan rahmatNya akhirnya telah dapat diselesaikan buku panduan (manual book) ini dengan judul Manual Book OFI PHP Framework.

Framework adalah sebuah software untuk memudahkan para programmer untuk membuat sebuah aplikasi web yang di dalam nya ada berbagai fungsi diantaranya plugin, dan konsep untuk membentuk suatu sistem tertentu agar tersusun dan terstruktur dengan rapi.

Dengan menggunakan framework bukan berarti kita akan terbebas dengan pengkodean. Karena kita sebagai seorang pengguna framework haruslah menggunakan fungsi – fungsi dan variable yang ada di dalam sebuah framework yang kita gunakan. Untuk saat ini framework terbagi dua yaitu framework PHP dan framework CSS.

Kami menyadari masih banyak kekurangan yang terdapat dalam aplikasi ini, untuk itu kritik dan saran sangat kami terima demi penyempurnaan aplikasi ini. Harapan kami, OFI PHP Framework dapat memberikan kontribusi dalam upaya kemajuan teknologi saat ini.

Purwokerto, Maret 2020

Penyusun

## DAFTAR ISI

HALAMAN JUDUL .....	1
KATA PENGANTAR.....	2
DAFTAR ISI.....	3
PENDAHULUAN.....	4
1.1 Pengenalan Framework.....	4
1.2. OFI PHP Framework.....	4
1.1. Membuat Navigasi.....	6
1.2. Membuat Collapse.....	8
1.3. Menjalankan Framework pada development mode.....	9
1.4. Membuat Form Group Secara Cepat.....	11
1.5. Routing.....	12
1.6. Block IP oleh OFI PHP Framework.....	16
1.7. Hashing password.....	17
1.8. Membuka / Load View Melalui Controller.....	18
1.9. Mengubah spasi menjadi tanda - pada judul artikel atau blog.....	20
1.10. Membuat flash notifikasi.....	20
1.11 Github OFI PHP Framework.....	25

## **PENDAHULUAN**

### **1.1 Pengenalan Framework**

Kata framework pastilah tidak asing lagi bagi para programmer, karena untuk membuat aplikasi atau pemrograman web akan membutuhkan sebuah framework agar dalam proses koding lebih mudah.

Framework adalah sebuah software untuk memudahkan para programmer untuk membuat sebuah aplikasi web yang di dalamnya ada berbagai fungsi diantaranya plugin, dan konsep untuk membentuk suatu sistem tertentu agar tersusun dan terstruktur dengan rapi.

Dengan menggunakan framework bukan berarti kita akan terbebas dengan pengkodean. Karena kita sebagai seorang pengguna framework haruslah menggunakan fungsi – fungsi dan variable yang ada di dalam sebuah framework yang kita gunakan. Untuk saat ini framework terbagi dua yaitu framework PHP dan framework CSS.

### **1.2. OFI PHP Framework**

OFI PHP Framework merupakan sebuah Framework yang ditulis dalam Bahasa Pemrograman PHP atau kepanjangannya Hypertext Preprocessor. Jika kamu yang sudah terbiasa menggunakan Framework Bootstrap, maka dengan mudah kamu bisa menggunakan framework php ini. OFI PHP Framework lebih mengunggulkan untuk tidak membuat bentuk desain website berulang ulang, misalnya jika kamu ingin membuat sebuah header navigasi namun dengan bentuk yang berbeda disetiap halamannya.

Maka dengan OFI PHP Framework kamu cukup menuliskan kode PHP saja, maka header navigasi yang kamu inginkan akan dengan sendirinya muncul. Untuk lebih lengkapnya silahkan baca dokumentasi dibawah ini

## 1.2 Instalasi projek

Pastikan kamu telah menginstal composer, selanjutnya buka cmd atau terminalmu silahkan ketik perintah ini

```
composer create-project ofitech/ofit-tech-framework
```

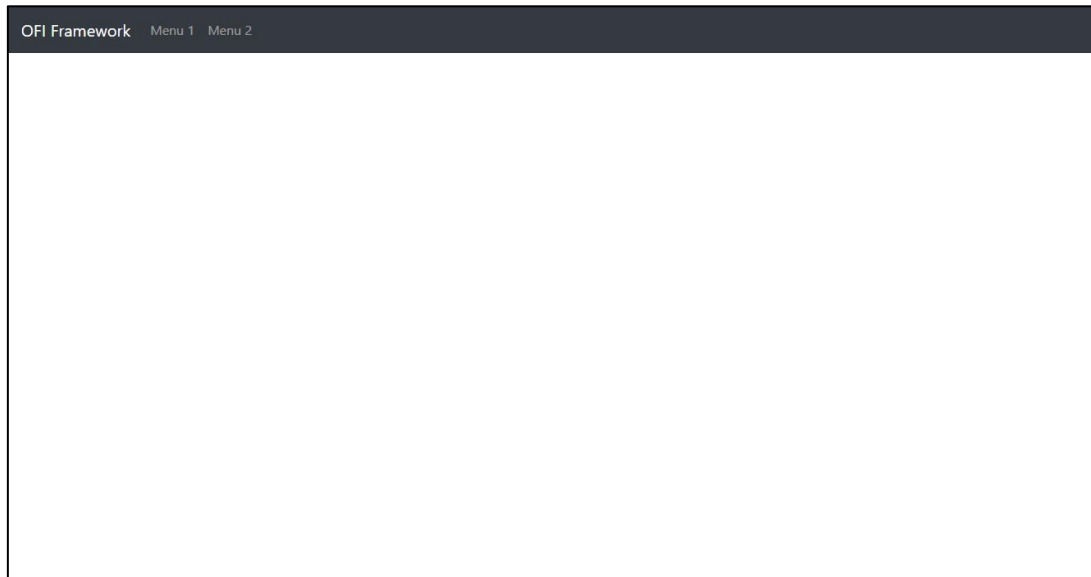
## CARA PENGGUNAAN

### 1.1. Membuat Navigasi

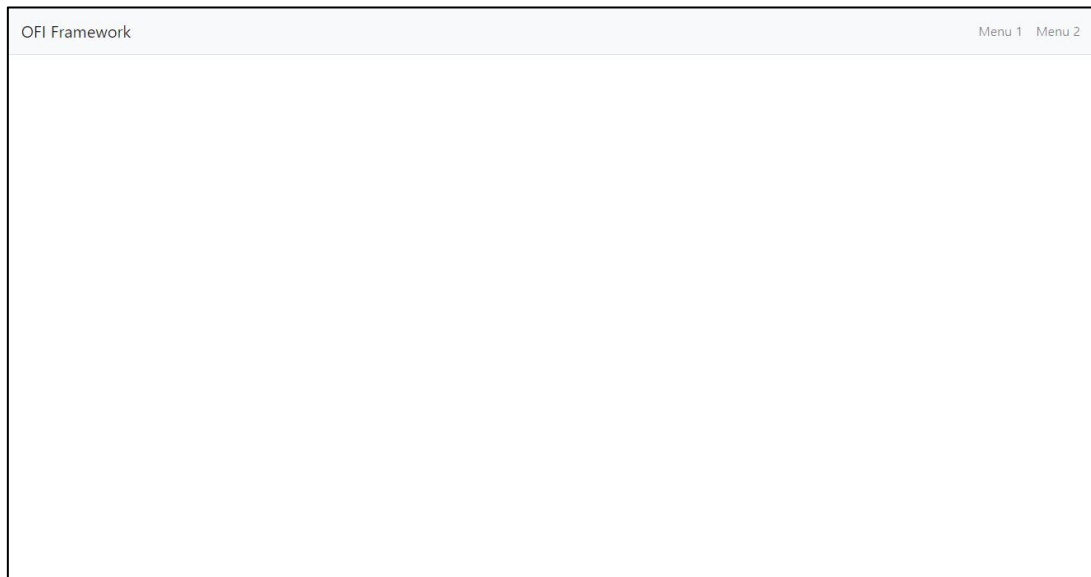
Untuk membuat navigasi kamu bisa menuliskan kode dibawah ini dimanapun kamu inginkan

```
<?php
$design→header(array(
    'title' ⇒ true,
    'theme' ⇒ 'dark', // white or dark
    'float' ⇒ 'left', // left, right, center
    'menu' ⇒ [
        [
            'name' ⇒ "Menu 1",
            'url' ⇒ "http://google.com"
        ],
        [
            'name' ⇒ "Menu 2",
            'url' ⇒ "http://google.com"
        ]
    ],
));
?>
```

Hasilnya akan seperti ini



Jika float right dan themenya white



## 1.2. Membuat Collapse

Untuk membuat collapse kamu perlu membuat trigger untuk memicu supaya collapse terbuka, dan membuat collapse itu sendiri. Trigger disini adalah berupa button. Perhatikan penulisan type pada kode dibawah ini. Perhatikan juga value name (lebih jelasnya lihat kode dibawah)

- a) Type btn untuk membuat trigger tombol
- b) Type Link untuk membuat trigger tag anchor <a></a>
- c) Dan type collapse untuk membuat collapse itu sendiri

Kode untuk membuat trigger collapse :



```
<?php
$design->collapse([
    'name' => "Collapseku", // must same
    'type' => 'link', // btn, link, collapse
    'value' => "Toggle Collapse"
]);

?>
```

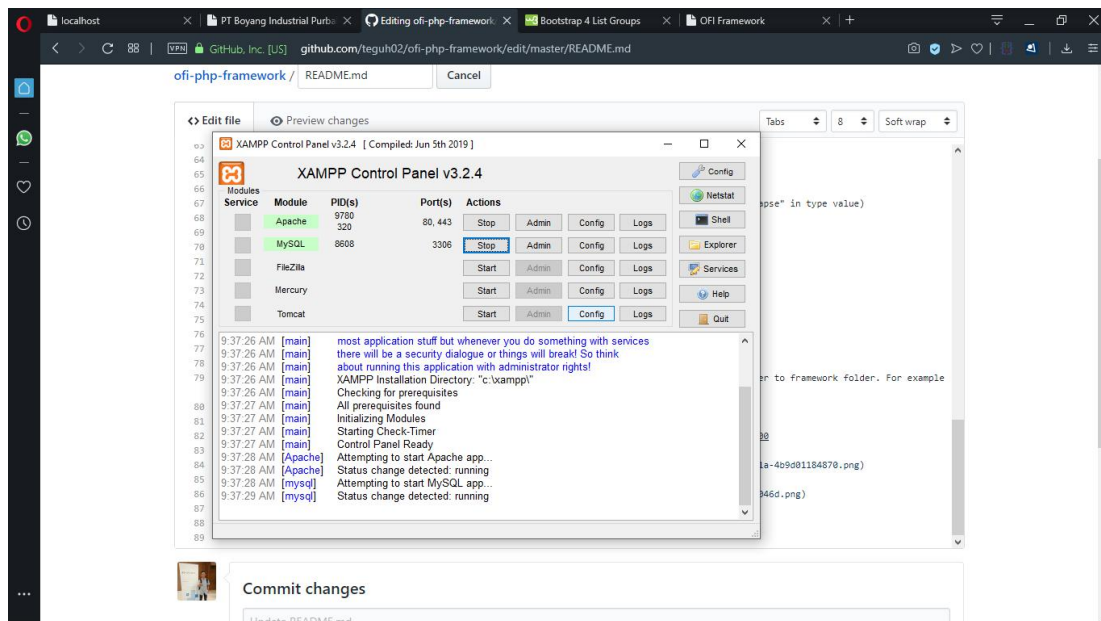
Kode untuk membuat konten collapse itu sendiri:



```
<?php
// You can put this script where do you want
$design->collapse([
    'name' => "Collapseku", // must same
    'type' => 'collapse',
    'value' => "My first collapse with php function"
]);
?>
```

### 1.3. Menjalankan Framework pada development mode

Untuk menjalankan framework ini pada development mode pertama kamu perlu menyalakan aplikasi local servermu misalnya xampp lalu kedua masuk ke direktorimu.

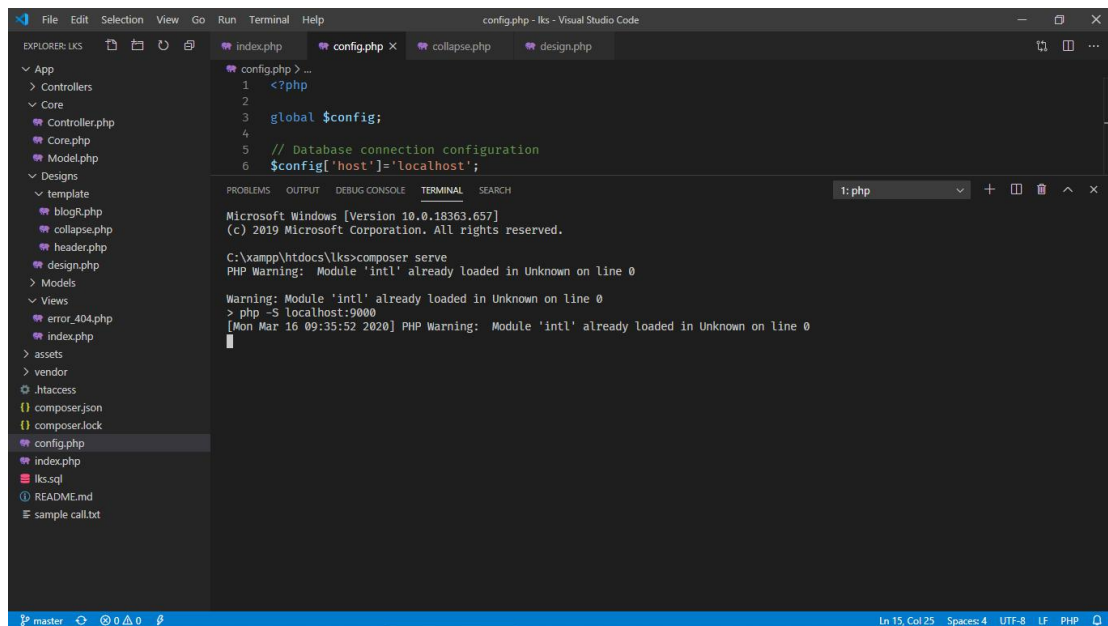


Pada kesempatan kali ini direktori framework saya bernama **lks**, maka saya perlu mengetikkan perintah dibawah ini pada aplikasi cmd bawaan windows untuk masuk ke folder proyek saya.

```
cd C:\xampp\htdocs\lks
```

Setelah kamu masuk kedalam direktori menggunakan cmd, silahkan ketik perintah **composer serve** maka projek akan berjalan pada alamat <http://localhost:9000>

Setelah development mode berjalan maka akan muncul kalimat yang sama persis seperti dibawah ini.



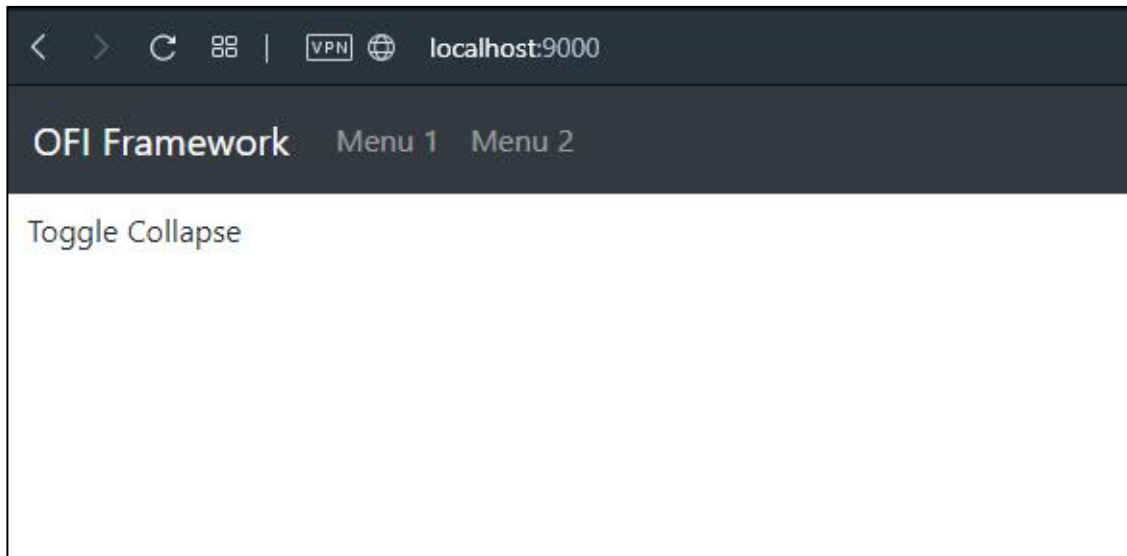
```
config.php > ...
1 <?php
2
3 global $config;
4
5 // Database connection configuration
6 $config['host']='localhost';

Microsoft Windows [Version 10.0.18363.657]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\xampp\htdocs\lks>composer serve
PHP Warning: Module 'intl' already loaded in Unknown on line 0

Warning: Module 'intl' already loaded in Unknown on line 0
> php -S localhost:9000
[Mon Mar 16 09:35:52 2020] PHP Warning: Module 'intl' already loaded in Unknown on line 0
```

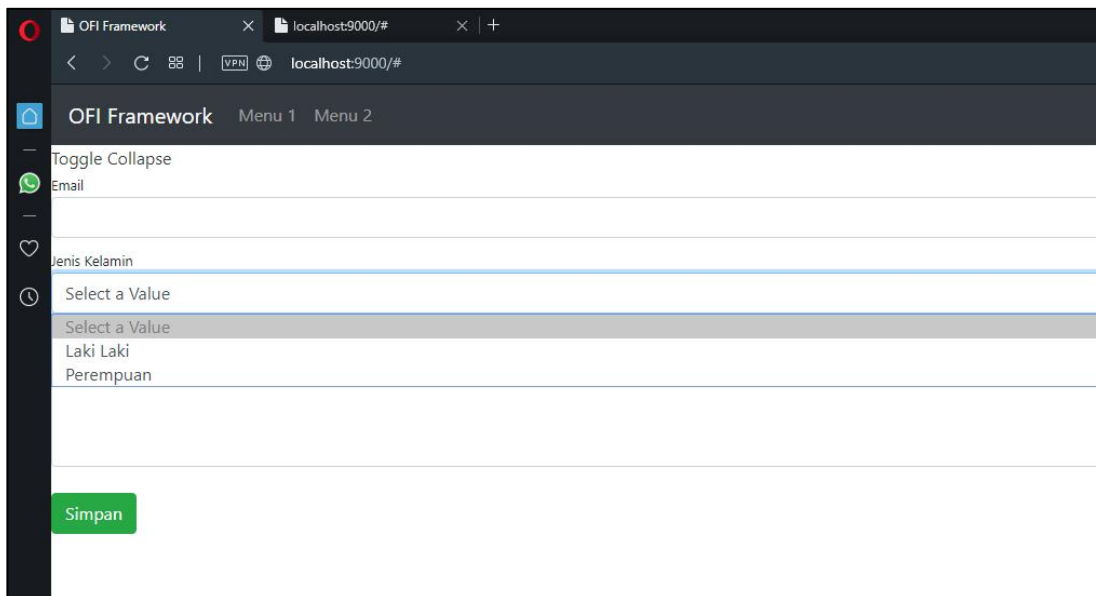
Dan buka aplikasi browser kesayanganmu lalu ketik alamat url tadi, maka hasilnya akan sama seperti dibawah ini.



#### 1.4. Membuat Form Group Secara Cepat

Kamu bisa menulis kode seperti dibawah ini jika kamu ingin membuat form group dengan cepat dan mudah tanpa harus menulis koding html. Semua input tersebut bersifat required, dan kamu bisa membuat inputan tersebut tidak required dengan cara memberi value pada atribut required menjadi no. Silahkan simak kode dibawah ini.

Hasil setelah kita menuliskan kode



Potongan kode beserta penjelasannya

```
<?php
// Default all required input
$design→form('post', [
    'url' ⇒ '#', // Up to you Where you want to post data
    'submit' ⇒ 'Simpan', // value of submit button
    'input' ⇒ [ // contains ofi php framework input script
        [
            'name' ⇒ 'Email',
            'type' ⇒ 'email', // null, text, password, email and etc
            'required' ⇒ 'no', // optional, you can remove this line
        ],
        [
            'name' ⇒ 'Jenis Kelamin', // jenis kelamin (gender)
            'input' ⇒ 'select',
            'required' ⇒ 'yes',
            'option' ⇒ [ // option can be use in select input only
                [
                    'value' ⇒ 'l', // value of option tag
                    'text' ⇒ "laki laki" // text in option tag
                ],
                [
                    'value' ⇒ 'p',
                    'text' ⇒ "Perempuan",
                ]
            ]
        ],
        [
            'name' ⇒ 'alamat',
            'input' ⇒ 'textarea', // input, textarea, select
            'required' ⇒ 'no', // yes or no
        ]
    ],
]);
?>
```

### 1.5. Routing

Jika kamu ingin mendeklarasikan URL pada OFI PHP Framework, kamu bisa mencari file web.php pada folder route. Pada file web.php ini memuat semua pendeklarasian route website kamu. Struktur penulisannya seperti pada gambar.

```

// other route

[
  'url' => "z",
  'type' => 'url',
  'method' => 'get', // GET AND POST only
  'to' => 'www.google.com'
],
[
  'url' => "print",
  'type' => 'print',
  // 'method' => "post", // if you need, you can remove this line
  'to' => 'Testing Only'
],

// other route

```

Keterangan :

1. URL berfungsi sebagai alamat URL mana yang ingin kamu gunakan
2. Type adalah tipe url tersebut (akan dijelaskan di bawah)
3. Method adalah untuk menentukan apa jenis HTTP Method route tersebut, HTTP Method yang didukung adalah GET dan POST, kamu juga boleh untuk tidak menulis value method ini, karena secara default methodnya GET.
4. To disini berfungsi sebagai alamat, artinya url kamu itu nantinya mau kemana.

Jenis Type :

- A. controller, jenis ini akan mengarahkan urlmu ke sebuah controller. Gunakan tanda @ untuk memilih method yang lebih spesifik. Contohnya disini penulis memilih method password\_hash pada controller yang dipilih.

```
[
  'url' => "test-controller-pass-hash",
  'type' => 'controller',
  'to' => 'indexController@password_hash'
],
```

- B. URL, tipe jenis ini akan mengarahkan url yang telah kamu deklarasikan sebelumnya ke url baru yang kamu pilih, misalnya ketika kamu membuka url /z maka akan otomatis mengarahkan ke halaman google Contohnya :

```
[
  'url' => "z",
  'type' => 'url',
  'method' => 'get', // GET AND POST only
  'to' => 'www.google.com'
],
```

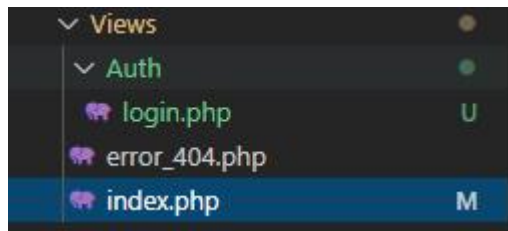
- C. Print, tipe jenis ini akan menampilkan atau mencetak karakter yang kamu masukan, contohnya dibawah ini akan mencetak value “Testing Only”

```
[  
  'url' => "print",  
  'type' => 'print',  
  // 'method' => "post", // if you need, you can remove this line  
  'to' => 'Testing Only'  
],
```

- D. View, tipe jenis ini akan meredirect url yang kamu deklarasikan ke sebuah file view, misalnya saja jika kamu mendeklarasikan url "" (kosong) maka kamu akan diarahkan ke halaman index.php utamamu. Contoh penulisannya

```
[  
  'url' => "", // Home (main index.php file)  
  'type' => 'view',  
  'to' => 'index'  
],
```

url tadi meredirect menuju file index.php seperti pada gambar



- E. JSON, tipe jenis ini akan mengambil data dari sebuah Server API, HTTP Method yang didukung hanyalah GET. Jadi kamu tidak bisa mengambil data jika server api bukan menggunakan metode GET, contoh penulisannya

```
[
  'url' => "json-sample",
  'type' => 'json',
  // Where do you want to fetch API Data
  'from' => 'https://jsonplaceholder.typicode.com/todos/1',
],
```

Hasilnya akan seperti ini

```
{
  "userId": 1,
  "id": 1,
  "title": "delectus aut autem",
  "completed": false
}
```

## 1.6. Block IP oleh OFI PHP Framework

Jika kamu ingin website kamu tidak dapat diakses melalui alamat IP (internet protocol) tertentu, kamu bisa membuka file blockip.php pada folder middleware. Kamu cukup menuliskan alamat IPnya saja sesuai contoh. Berikut adalah struktur penulisannya





### 1.7. Hashing password

Tidak mungkin jika sebuah password tidak memiliki keamanan tersendiri ketika disimpan dalam database, Teknik pengamanan ini dinamakan hashing, nah pada OFI PHP Framework juga memiliki bantuan hashing tersendiri, berikut adalah struktur penulisan hash pada sebuah method. Saya ingin mengamankan kata teguh2001 ini. Dan hasilnya pun bisa berbeda beda walaupun satu kata “teguh2001”, itulah keamanan password yang diterapkan pada framework ini

```
// Jangan lupa tuliskan teks dibawah ini paling atas (dibawah teks namespace)
// sebuah controller untuk mengimport library helper tadi
use App\Core\helper;

// Struktur penulisannya
helper::hash('teguh2001');

// hasil hash
$2y$10$YSL7SW7.iVs7IHpwLLQB.OLrYhh0bVA7Xa.Tv/R2kfb3HbKR5CcVK
```

### Mengambil request ketika metode POST

Jika kamu ingin mengambil sebuah value ketika metode POST terjadi pada framework ini, kamu bisa menuliskan seperti ini, saya contohkan saya ingin mengambil value bernama email maka penulisannya seperti ini. Bantuan ini hanya bisa dipakai pada metode POST, kamu tidak bisa mengambil value parameter pada metode GET dengan bantuan ini

```
// Jangan lupa letakan skrip ini diatas sebuah controller
// dan dibawah teks namespace
use App\Core\helper;

// Penulisannya seperti ini
$email = helper::request('email');

// proses mencetak datanya
echo $email;
```

### 1.8. Membuka / Load View Melalui Controller

Jika kamu ingin membuka sebuah view melalui controller kamu bisa menggunakan fungsi dibawah ini

```

// Jangan lupa import class ini
use App\Core\Controller;

// dan menambahkan extends controller pada class yang digunakan

class indexController extends Controller
{
    public function crud()
    {
        // Untuk meload suatu view, kamu bisa menggunakan fungsi
        // loadTemplate dan diikuti dengan data array yang ingin kamu
        // bawa ketika meload file view

        $this->loadTemplate('home', []);
    }
}

```

Skrip diatas akan merujuk pada file home.php yang berada di folder view, jika kamu ingin membawa suatu data array ketika meload view tadi, kamu bisa menuliskannya seperti ini (fungsi database akan dijelaskan pada halaman selanjutnya)

```

class indexController extends Controller
{
    public function crud()
    {
        // Fungsi database akan dijelaskan pada halaman
        // selanjutnya
        $data = DB::all('blog');
        $this->loadTemplate('blog/crud', ['data' => $data]);
    }
}

```

### 1.9. Mengubah spasi menjadi tanda - pada judul artikel atau blog

Mengubah spasi menjadi tanda - pada judul artikel atau blog teknik ini sering dibilang dengan slug, kamu juga bisa Mengubah spasi menjadi tanda - pada judul atau blog dengan menggunakan bantuan slug dari OFI PHP Framework, cara penulisan dan petunjuk penggunaanya dijelaskan pada gambar



```
// Jangan lupa untuk mengimport class berikut
use App\Core\Helper;

public function slug
{
    // Kamu bisa memanggil fungsi tadi dengan menuliskan
    // helper::slug('') misalnya saya ingin mengubah kata
    // OFI PHP Framework

    echo helper::slug('OFI PHP Framework');

    // hasilnya seperti ini
    // ofi-php-framework
}
```

### 1.10. Membuat flash notifikasi

Pada kesempatan kali ini saya ingin memberi tips bagaimana cara membuat flash notifikasi pada OFI PHP Framework, sebelum itu saya ingin memperkenalkan jenis notifikasi yang dapat dibuat, apa sajakah itu?

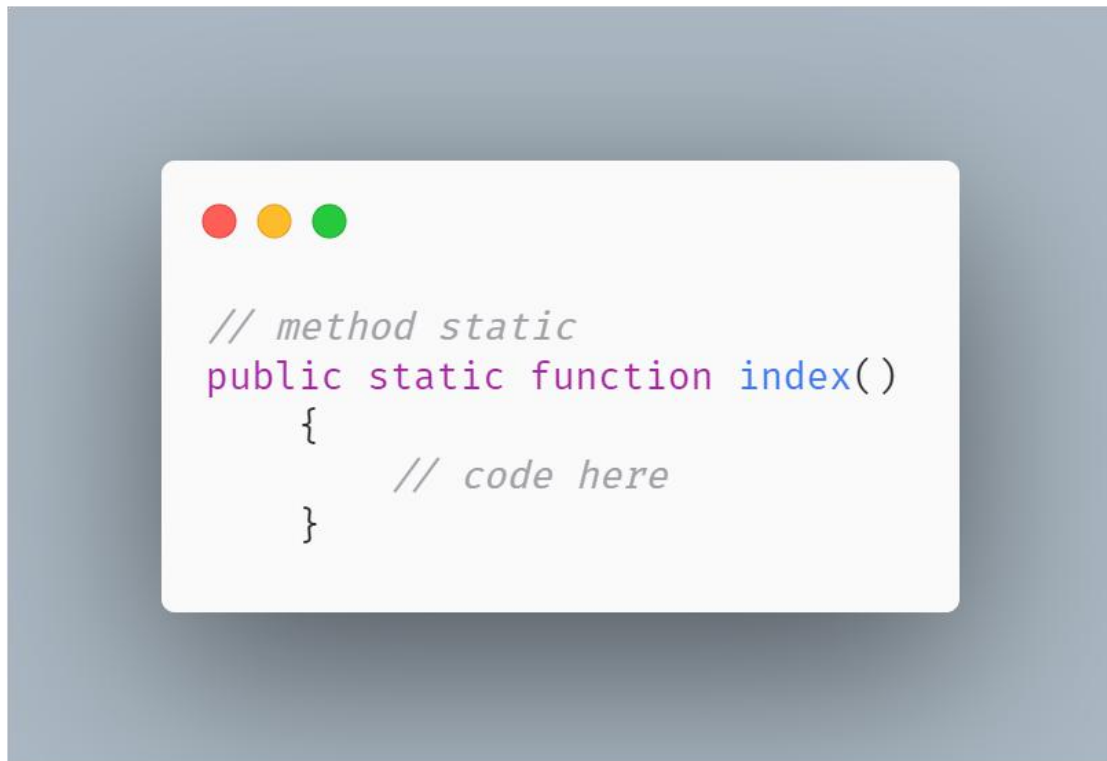
- Success notifikasi berwarna hijau, gunannya untuk menandakan berhasilnya suatu proses
- Info notifikasi berwarna biru muda, gunannya untuk memberi informasi kepada pengguna
- Warning notifikasi berwarna kuning, gunanya untuk memberi peringatan (ibarat lampu kuning pada lampu lalu lintas)

- d. Error notifikasi berwarna merah, gunanya untuk memberi tahu bahwa terjadi kegagalan dalam suatu proses

A. Membuat kode pada controller

- 1. Membuat flash notifikasi pada method static

Dibawah ini adalah contoh method static



Untuk membuat flash notifikasi pada method static kita perlu membuat class notifikasi baru lalu kita pilih jenis notifikasi yang ingin dibuat, contoh penulisannya seperti ini.



```
// Contohnya membuat pesan notifikasi error
public static function index()
{
    // Buat class notifikasi baru
    $flash = new \Plasticbrain\FlashMessages\FlashMessages();

    // Pilih jenis pesan error diikuti dengan isi pesan
    // dan alamat URL (otomatis kamu akan diarahkan ke alamat URL
    // yang diatur beserta membawa isi pesannya)
    $flash->error('Failed to login, try again', '/login');
}
```

Selanjutnya setelah kita buat kode pada controller kita buat kode pada view untuk menampilkan notifikasinya, contoh penulisannya seperti

ini

```

    // dan letakan kode ini dimanapun kamu ingin menampilkan notifikasi
    // flash tadi
    <?php
        $flash→display();
    ?>

    // Misalnya saja

    <div class="container col-md-4">
        <?php
            $flash→display();
        ?>
        <div class="card">
            <div class="card-body">
                <h5 class="mb-3">Login Page</h5>

                <?php
                    $design→form('post', [
                        'url' ⇒ '/login-detect',
                        'submit' ⇒ 'Login',
                        'input' ⇒ [
                            [
                                'name' ⇒ 'username or email',
                            ],
                            [
                                'name' ⇒ 'password',
                                'type' ⇒ 'password',
                            ],
                        ],
                    ]);
                ?>
            </div>
        </div>
    </div>

    // kamu bisa menemukan kode ini pada file login di folder auth

```

## 2. Membuat flash notifikasi pada method non static



```
// Contoh penulisan method non static  
public function index()  
{  
  
}
```

Untuk menulis kode notifikasi flash pada method non static kamu bisa mengikuti contoh dibawah ini untuk penulisannya.





```
// Sebelum mulai menulis pastikan kamu telah mengimport class controller
// dan mengextends kan pada controller yang sedang kamu gunakan

// kode import
use App\Core\Controller;

// kode untuk extends class controller
class indexController extends Controller {

}

// lanjut tahap selanjutnya 📌

// Kamu bisa memanggil dengan fungsi $this->flash
// diikuti dengan jenis notifikasi yang kamu inginkan,
// untuk membuat notifikasi flash pada method non static
// Dan jangan lupa untuk mengisi url yang kamu inginkan yaah hehe
// saya mencontohkan ingin mengembalikannya pada halaman utama
// dengan menuliskan tanda /

public function index()
{
    // ..kode lainnya disini

    $this->flash->success('Sukses menghapus data artikel', '/');
}
```

Untuk lebih lengkapnya teman-teman bisa membuka kodenya pada controller yang sudah ada misalnya controller auth.

### 1.11 Github OFI PHP Framework

Untuk tutorial lainnya silahkan kunjungi repository github melalui link dibawah ini. Kedepannya akan kami lakukan update dan perbaikan bug pada framework ini. Mohon dukungannya teman-teman :)

<https://github.com/teguh02/ofl-php-framework/>

### 1.12 Mengirim Email

Untuk mengirim email pada OFI PHP Framework pertama kamu harus ubah file config menjadi seperti ini.



```
// Other code

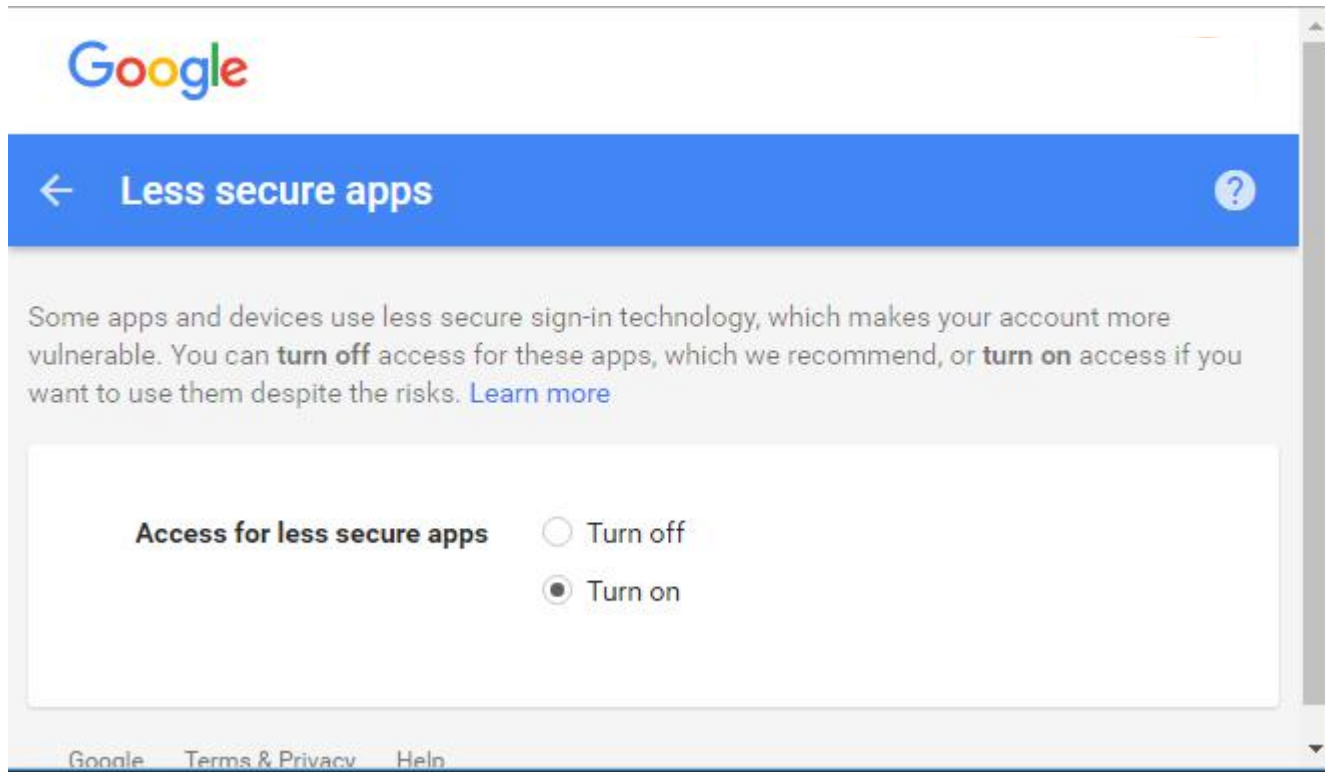
define('SMTPSecure', 'tls');
define('Host', 'smtp.gmail.com');
define('Port', 587);
define('GmailUsername', 'hisaon4@gmail.com'); // your gmail email
define('GmailPassword', 'nan*****'); // Your gmail password (censored)
define('SMTPDebug', 0);
define('senderEmail', 'system@ofiFramework.com');
define('senderName', 'OFI Framework Mailing System');

// Other code
```

Selanjutnya Buka halaman untuk settingan Less secure apps akun google yang kamu gunakan pada config tadi. Alamat URLnya adalah

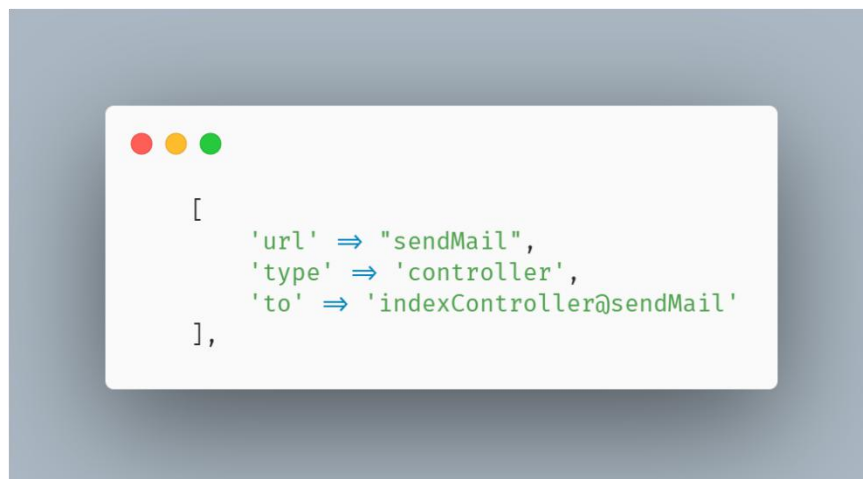
<https://myaccount.google.com/lesssecureapps>

Pada Access for less secure apps pilih Turn On.



Selanjutnya kamu bisa ikuti penulisan kode dibawah ini

Pada Route :



Pada controller

```

// Contoh penulisan kode untuk mengirim email
// pada OFI PHP Framework

public function sendMail()
{
    $mail = [
        'to' => 'teguhrijanandi02@gmail.com',
        'receiverName' => 'Hai teguh Rijnandi',
        'subject' => 'First Testing', // subjcet message
        'body' => 'wkwkwwkw', // in html

        // attachment is optional
        'attachment' => [
            'name' => 'attachment.jpg',
            'type' => 'url',
            'value' => '', // url of image here
        ]
    ];

    // Call the helper
    $sendMail = helper::sendEmail($mail);

    // If fail to send
    if (!$sendMail) {
        echo "Mailer Error: " . $sendMail->ErrorInfo;
    } else {
        // If success
        echo "Message sent!";
    }
}

```

Source code ini bisa kamu dapatkan di

[https://github.com/teguh02/ofl-php-framework/releases/tag/Example\\_OFI\\_PHP\\_Send\\_Email](https://github.com/teguh02/ofl-php-framework/releases/tag/Example_OFI_PHP_Send_Email)