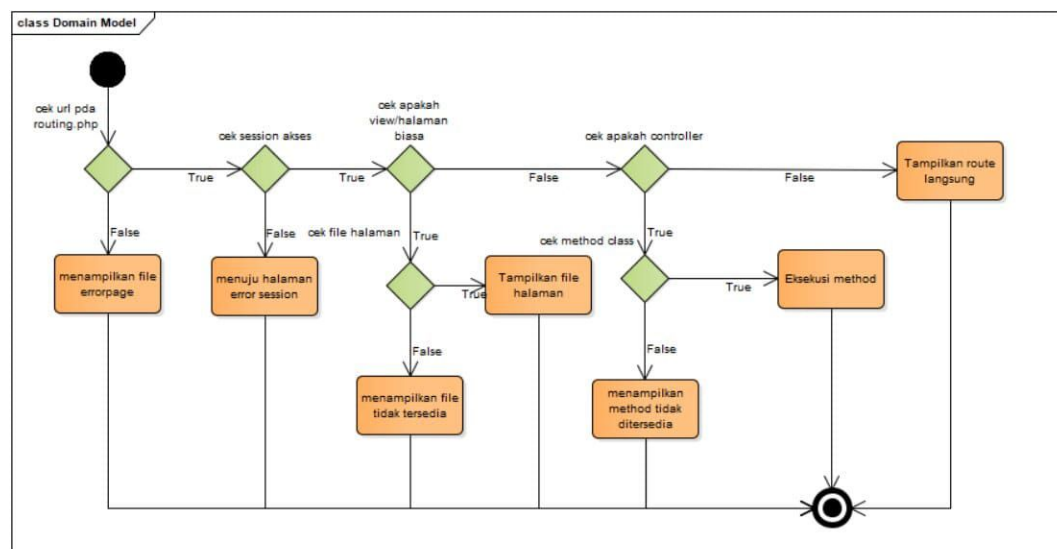


WEB FRAMEWORK NANO-ALPHA

BAGIAN ROUTING DAN INIT OBJEK

Routing merupakan salah satu fitur yang digunakan untuk pengelolaan halaman yang dapat diakses oleh user melalui browser, dalam pengaksesan disetiap halaman website seperti yang diketahui harus mengikuti path file halaman tersebut. Namun pada Nano-alpha tidak seperti itu meskipun path file tersebut ada dalam direktori jika tidak didaftarkan halamannya ke dalam routing maka tidak akan tampil pula halamannya. Jadi routing bisa dikatakan sebagai lalu lintas untuk halaman website. Bukan hanya mengatur halaman website, routing juga bisa mengatur hak akses (session) dan juga halaman yang tidak tersedia didalam website. bentuk dari halaman website bukan hanya file halaman seperti .php atau .html saja namun bisa juga dalam bentuk fungsi class controller.



Gambar diatas merupakan alur dari proses routing yang memeriksa mulai dari halaman url sampai ke class controller, istilah dari routing bisa dikenal dengan url mapping yang memiliki kunci sebagai halaman url dan nilai ditiap datanya menjadi path file direktori. Jadi bisa dikatakan bahwa pengguna bisa membuat halaman url dengan unik tanpa mempengaruhi path file nya.

ATURAN PENULISAN HALAMAN URL

Tidak banyak aturan khusus dalam menuliskan halaman url, penulisannya masih sama seperti pada umumnya. Hanya saja terdapat beberapa perbedaan sedikit pada bagian penambahan variabel di halaman url. Konsep routing ini kurang lebih sama seperti yang ada pada laravel tapi saya membuatnya versi sendiri, routing yang saya buat ini ditampung oleh sebuah file yang bernama web.php pada folder public (bisa dilihat dipenjelasan tiap folder pada bagian 1), jadi semua atribut yang berkaitan dengan web bisa dimanipulasi pada file ini termasuk beberapa fungsi yang sudah saya atur demikian. Dibawah ini ada 2 contoh penulisan url pada halaman website yang sudah dihosting dan halaman pada localhost.

$$\text{Tutorial.} \frac{\text{com}}{1} \text{/} \frac{\text{belajar}}{2} \text{/} \text{(id)} \frac{\text{11}}{2} \text{11}$$

A

$$\text{localhost/tutorial/} \frac{\text{11}}{1} \text{/} \frac{\text{belajar}}{2} \text{/} \text{(id)} \frac{\text{11}}{2} \text{11}$$

B

1. Berisi domain dari website yang nilainya tidak berubah - ubah.
2. Tidak ada batasan dalam mengisi nilai dari sub-domainnya, wajib menutup sub-

domain dengan garis miring(/). Dalam sub-domain bisa di isi juga dengan nilai variabel, seperti contoh diatas terdapat variabel id.

ATURAN PENULISAN SOURCE KODE

Dalam penulisan source kode routing ini terdapat fungsi dari 2 objek class yang sangat berpengaruh, yaitu objek route dan juga response (fungsi view). File yang digunakan untuk pengelolaan routing ini berada pada file routing.php pada folder view.

objek route mempunyai 3 fungsi yang menjadi bagian penting dan sering digunakan nantinya dalam mengembangkan website dengan framework ini. Berikut penjelasan dari setiap fungsinya :

- get : memeriksa dan mengakses halaman url.
- post : memeriksa dan mengakses halaman url khusus untuk aksi pada menggunakan form.
- checkRoute : mengakumulasikan route lalu memeriksa tiap route tersebut apakah nilai dari url tersedia.

objek response mempunyai empat fungsi yang menjadi bagian penting dan sering digunakan nantinya dalam mengembangkan website dengan framework ini. Berikut penjelasan dari setiap fungsinya :

- Redirect : pindah halaman berdasarkan pada parameternya.
- View : menampilkan halaman sesuai dengan lokasi dari file didalam direktori.
- Url : mencetak url halaman yang digunakan untuk akses ke halaman lain.
- Back : pindah ke halaman sebelumnya.

Dari fungsi kedua objek tersebut tidak digunakan semua hanya fungsi get dan post pada objek route dan fungsi view pada objek response. Berikut ini penulisan source kodenya beserta penjelasan untuk routing.

```
route->get(url, view, acces)

Url = halaman url untuk diakses.
View = tampilan halaman yang ingin ditampilkan, maka digunakan
fungsi
Anonymus dengan fungsi view didalamnya. Penulisannya
view('path file
Url pada folder public atau class controller dan
fungsinya(controller@fungsi)', variabel(alternatif)).
Acces = hak akses yang diperbolehkan untuk mengakses halaman
website
Aturan penulisannya ['nama session' => ['value sessionnya']],
acuan
hak akses berada pada file view/Acces.php.

Route tanpa hak akses dan variabel
$route->get('', function (){
    $this->response->view('index');
});
```

```

Route yg memiliki variabel dan hak akses
$route->get('Belajar/(id)/', function ($id){
    $this->response->view('Belajar/index',$id);
},['jabatan' => ['guru','siswa']]);

Route dengan kelas controller sebagai halaman(controller
dijelaskan pada bagian MVC)
$route->get('Belajar/', function (){
    $this->response->view('SiswaController&create');
},['jabatan' => ['guru','siswa']]);

route->post(url, view, button, acces)
Button = nama tombol submit dari form

Route dengan kelas controller
$route->post('perbaruiMahasiswa/(id)/', function ($id){
    $this->response->view('MahasiswaController&update',
$id);
},'perbarui');

$route->post('tambahMahasiswa/', function (){
    $this->response->view('MahasiswaController&remove');
},'tambah');

```

Berikut ini gambar isi dari file acces.php yang digunakan untuk menyimpan hak akses tersedia dapat proyek.

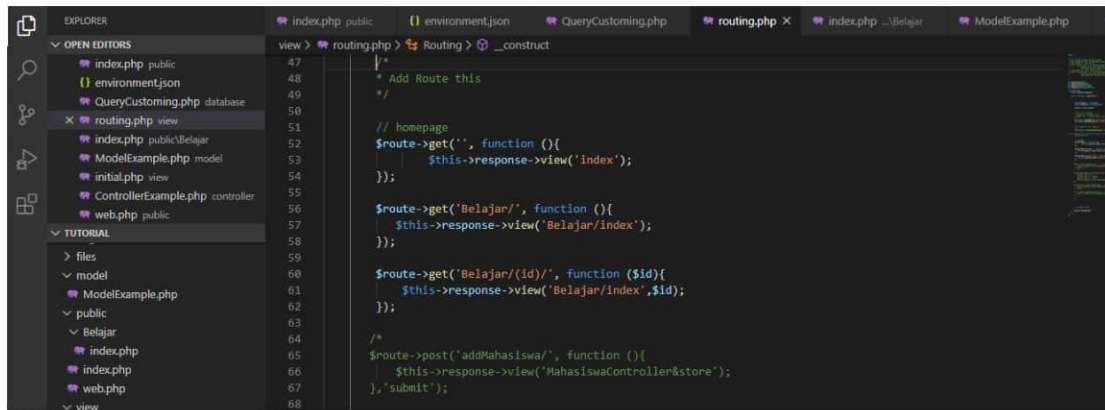
```

10  * Description of Acces
11  *
12  * @author Irwansyah
13  */
14  class Acces {
15
16      /*
17       * semua value yang tersedia dalam hak akses tanpa perlu memasukan nama sessionnya
18       */
19      public static $accesRight = [
20          'all'
21      ];
22
23      public static $accesCookie = [
24
25      ];
26
27
28  }
29

```

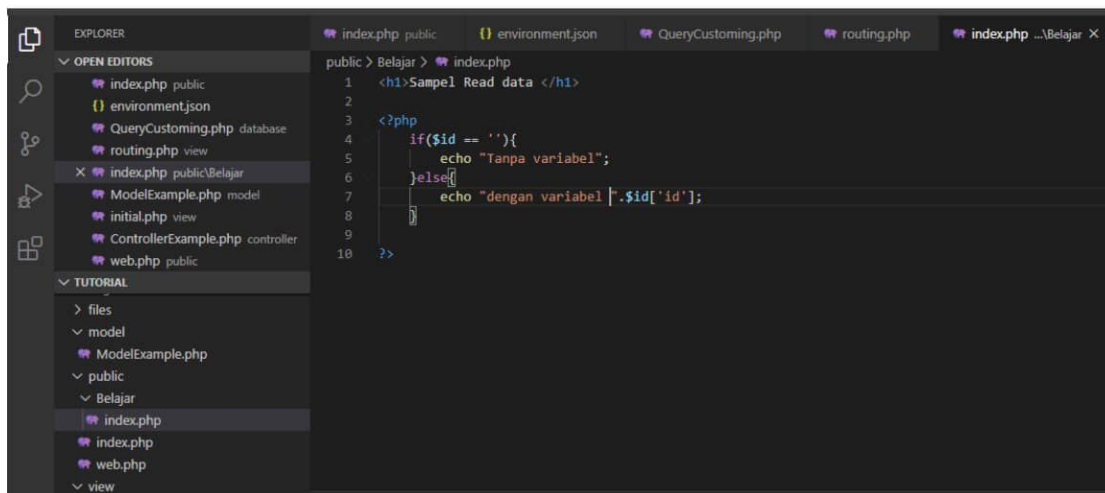
Sampel contoh

Berikut ini saya memberikan contoh routing tanpa variabel dan dengan variabel, berikut potongan source kodenya pada routing



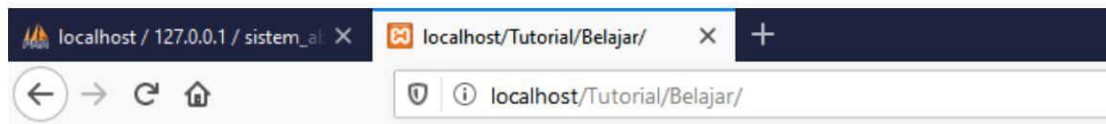
```
47  /*
48  * Add Route this
49  */
50
51  // homepage
52  $route->get('/', function () {
53      $this->response->view('index');
54  });
55
56  $route->get('Belajar/', function () {
57      $this->response->view('Belajar/index');
58  });
59
60  $route->get('Belajar/{id}/', function ($id) {
61      $this->response->view('Belajar/index', $id);
62  });
63
64  /*
65  $route->post('addMahasiswa/', function () {
66      $this->response->view('MahasiswaController@store');
67      $submit;
68  });
```

Dari gambar diatas terdapat 3 data routing namun kita berfokus pada 2 routing paling akhir, yakni Belajar dan Belajar/{id}/, bisa diperiksa juga path file pada response view. Nama variabel dari function bisa selain id, yang penting variabel function dan view mesti sama.



```
1  <h1>Sample Read data </h1>
2
3  <?php
4      if($id == ''){
5          echo "Tanpa variabel";
6      }else{
7          echo "dengan variabel ". $id['id'];
8      }
9
10 ?>
```

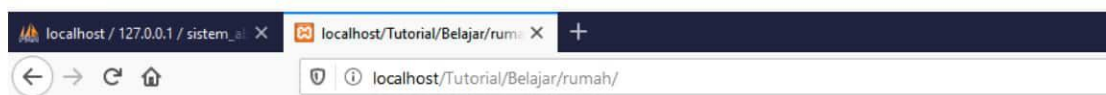
Gambar diatas merupakan source code untuk file index.php pada folder Belajar. Tidak akan terjadi kesalahan meskipun file tersebut digunakan 2 kali pada routing yang berbeda, perlu diingatkan bahwa variabel \$id itu parameter yang tidak bisa dirubah nilainya, \$id itu bertipe data array jika terdapat variabel dalam halaman selain itu hanya var dengan nilai null, untuk menggunakannya hanya perlu mengetikkan kunci array dengan param urlnya (id), bisa dilihat pada gambar routing sebelumnya.



Sampel Read data

Tanpa variabel

Gambar diatas dan dibawah ini tampilan dari routing yang diakses melalui web browser

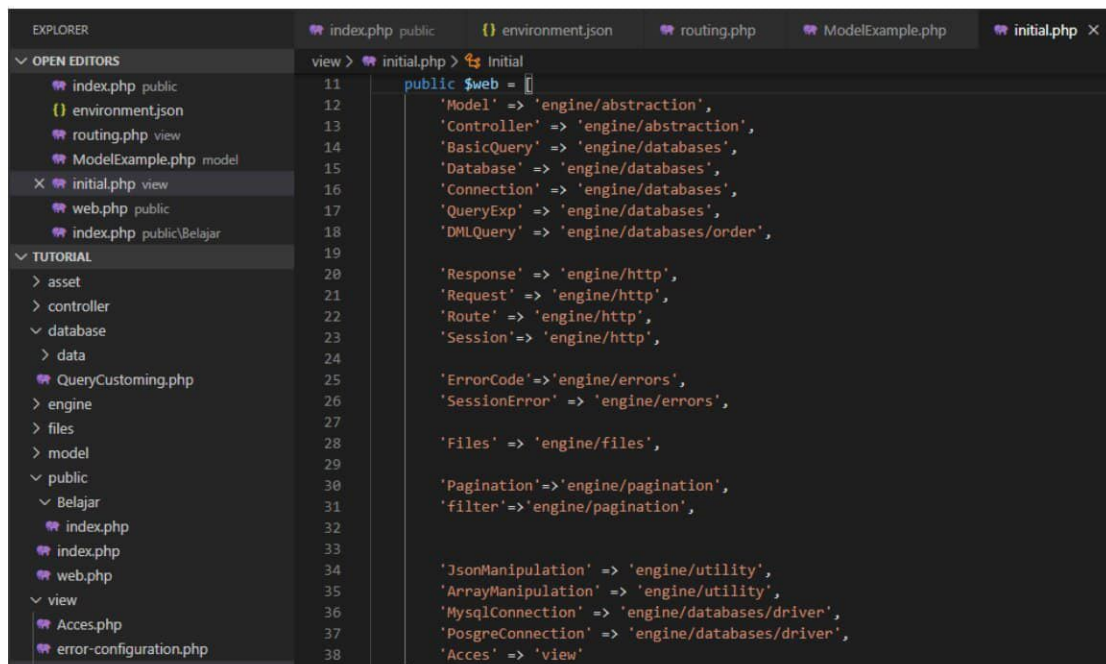


Sampel Read data

dengan variabel rumah

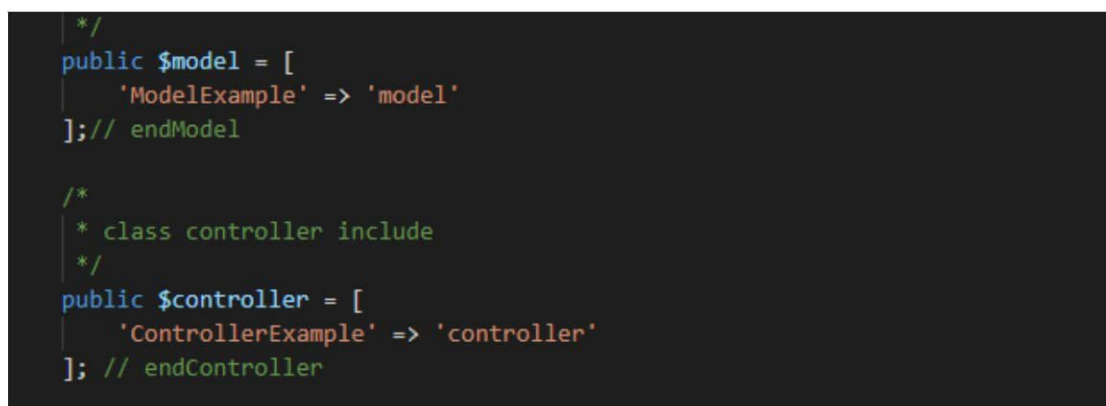
INIT OBJEK

Objek yang digunakan untuk menampung semua objek yang tersedia didalam Nano-alpha seperti file engine web, model dan controller yang disimpan pada file initial.php pada folder view. Jadi, di file inilah letak untuk inisialisasi dari objek disimpan, khusus untuk variabel web tidak boleh ada yang dihapus. Namun jika ingin ditambahkan bisa dengan pola penulisan 'nama kelas' => 'path file atau namespace' pathfile dengan namespace harus sesuai polanya, saya tidak menyarankan untuk merubahnya atau menambahkan jika anda masih pemula.



```
11 public $web = []
12
13 'Model' => 'engine/abstraction',
14 'Controller' => 'engine/abstraction',
15 'BasicQuery' => 'engine/databases',
16 'Database' => 'engine/databases',
17 'Connection' => 'engine/databases',
18 'QueryExp' => 'engine/databases',
19 'DMLQuery' => 'engine/databases/order',
20
21 'Response' => 'engine/http',
22 'Request' => 'engine/http',
23 'Route' => 'engine/http',
24 'Session' => 'engine/http',
25
26 'ErrorCode' => 'engine/errors',
27 'SessionError' => 'engine/errors',
28
29 'Files' => 'engine/files',
30
31 'Pagination' => 'engine/pagination',
32 'filter' => 'engine/pagination',
33
34 'JsonManipulation' => 'engine/utility',
35 'ArrayManipulation' => 'engine/utility',
36 'MysqlConnection' => 'engine/databases/driver',
37 'PosgreConnection' => 'engine/databases/driver',
38 'Acces' => 'view'
```

Setelah variabel web, selanjutnya ialah variabel model dan controller. Tidak perlu ada perubahan apapun disini kecuali menghapus kelas model yang menghapus file dari folder model secara langsung, karena ketika membuat file menggunakan console get, otomatis akan diinit bagian yang seperti model dan controller. apabila tidak dihapus maka akan terjadi error file tidak tersedia.



```
1 /*
2 public $model = [
3     'ModelExample' => 'model'
4 ]; // endModel
5
6 /*
7  * class controller include
8  */
9 public $controller = [
10     'ControllerExample' => 'controller'
11 ]; // endController
```