Jurusan Teknologi Informasi Politeknik Negeri Malang

Jobsheet-2: Routing dan Controller

Mata Kuliah Pemrograman Web Lanjut

September 2021

Pendahuluan

Pada praktikum kali ini anda akan mempelajari tentang routing dan controller pada web framework.

Tujuan Pembelajaran

1. Mahasiswa memahami konsep routing Web Framework
2. Mahasiswa menerapkan routing pada Web Framework
3. Mahasiswa memahami konsep controller
4. Mahasiswa menerapkan controller pada Web Framework

KAJIAN TEORI

Routing

Jika kita diterjemahkan secara bebas, route berarti jalur. Dan hal itu bisa kita bayangkan sebagai konsep routing pada Laravel, yaitu jalur URL yang bisa diakses oleh pengguna aplikasi dan ke mana jalur itu diproses. Contoh ada routing sederhana /hello, route ini bisa diakses pada http://localhost:8000/hello dan akan menampilkan string 'Hello World'. Sintak dasar dari routing adalah sebagai berikut:

```
Route::verb("/path", callback);
```

Untuk membuat route digunakan Facade Route diikuti dengan verb yang merupakan HTTP verb, umumnya terdiri dari get, post, put, delete, options, patch. Selain itu dibutuhkan path yang berupa URL setelah nama domain aplikasi yang diakses oleh pengguna. Dan pada bagian akhir terdapat callback yang dapat berupa callback function atau controller action yang menjalankan logika ketika path diakses oleh pengguna.

Berikut adalah contoh routing menggunakan callback function yang akan menampilkan pesan 'Hello World'.

```
Route::get('/hello', function () {
    return 'Hello World';
});
```

Route di atas dapat diterjemahkan ketika pengguna mengakses URL pada http://localhost:8000/hello akan mengeksekusi callback function yang menampilkan pesan 'Hello World'. Pengguna akan mendapatkan pesan tersebut yang akan ditampilkan pada browser mereka. Akan tetapi penggunaan callback function jarang sekali dipakai dalam pembuatan aplikasi sesungguhnya, karena untuk logika yang kompleks menjadikan kode susah di-*maintenance*. Sebagai solusi diperkenalkan konsep *controller*. Jika route di atas dikonversi ke *controller* menjadi sebagai berikut:

```
Route::get('/hello', [WelcomeController::class,'hello']);
```

Penggunaan controller menyederhanakan pendefinisian route, karena cukup dituliskan nama controller diikuti action pada callback. Untuk implementasi teknis ada dalam class WelcomeController.

Pendefinisian route sebaiknya dituliskan sesuai kegunaannya. Secara umum laravel membagi menjadi empat tempat, yaitu:

1. `routes/web.php` digunakan untuk web standard
2. `routes/api.php` digunakan untuk web service/API
3. `routes/console.php` digunakan untuk command line
4. `routes/channel.php` digunakan untuk broadcast channel melalui websocket

Secara umum aplikasi yang dibuat cukup dengan `routes/web.php` dan `routes/api.php`. Bahkan jika aplikasi tidak perlu menyediakan API hanya menggunakan `routes/web.php` saja. Untuk dokumentasi lebih lanjut anda dapat membaca pada https://laravel.com/docs/8.x/routing dokumentasi official Laravel.

Basic Routing

Pada dasarnya Routing di Laravel membutuhkan informasi mengenai http verb kemudian input berupa url dan apa yang harus dilakukan ketika menerima url tersebut. Untuk membuat sebuah route anda dapat menggunakan callback function atau menggunakan sebuah controller. Berikut ini contoh rencana pembuatan route dan implementasinya :

No	Http Verb	Url	Fungsi
1	get	/hello	Tampilkan String Hello ke browser.
2	get	/world	Tampilkan String World ke browser

Berikut ini contoh routing untuk kasus nomor 1 dengan menggunakan callback function

```
use Illuminate\Support\Facades\Route;
Route::get('/hello', function () {
    return 'Hello';
});
```

Perhatikan dengan baik apa jenis route yang digunakan kemudian input url yang dimasukkan dan bagaimana cara menambahkan callback function. Sedangkan untuk kasus kedua jika ingin diselesaikan dengan menggunakan controller function anda harus membuat sebuah controller yang berisi method yang anda sebutkan di router. Berikut ini kode program untuk membuat route menggunakan controller.

```
use Illuminate\Support\Facades\Route;
Route::get('/world', [WelcomeController::class,'world']);
```

Perhatikan pada route dengan controller perbedaannya hanya ada pada penggantian callback function dengan parameter controller dan pemberian nama function. Pahamilah pada Laravel anda harus mendaftarkan semua url yang dapat diakses pada file file routes ini.

Router Method

Pada Laravel anda dapat menggunakan semua http verb untuk dipasang sebagai method router yang ingin anda gunakan, sudah dijelaskan sebelumnya bahwa semua http verb dapat dilayani dengan Router pada laravel. Endpoint / url router sebaiknya mengikuti best practice berikut ini dimana sebuah resource dapat dilayani dengan fungsi berbeda pada setiap http verb nya.

Resource	POST	GET	PUT	DELETE
/mahasiswa	Membuat record mahasiswa baru	Mengambil Daftar Mahasiswa	Update banyak data mahasiswa	Delete banyak data mahasiswa
/mahasiswa/{id}	Error	Tampilkan Data Satu Mahasiswa sesuai id yang dikirim	Update data mahasiswa jika ada data sesuai id yang dikirim	Delete satu data mahasiswa sesuai id yang dikirim

Perhatikan tabel diatas, dengan membuat tabel route seperti ini lebih baik daripada membuat tabel routing yang ada sebelumnya. Dengan cara seperti ini lebih jelas sebuah resource / url (/mahasiswa) memiliki http verb apa saja dan apa yang dilakukan pada masing masing http verb tersebut. Perlu diketahui laravel dapat mendukung satu route yang memiliki lebih dari satu http verb atau memiliki semua http verb. Berikut ini kode program routing untuk tabel di atas

```
Route::get('mahasiswa', function ($id) {
});

Route::post('mahasiswa', function ($id) {
});

Route::put('mahasiswa', function ($id) {
});

Route::delete('mahasiswa', function ($id) {
});

Route::get('mahasiswa/{id}', function ($id) {
});

Route::put('mahasiswa/{id}', function ($id) {
});

Route::delete('mahasiswa/{id}', function ($id) {
});
```

Anda dapat memeriksa dan memvalidasi apakah route yang dibuat sudah benar dengan menggunakan perintah berikut.

```
php artisan route:list
```

Output dari perintah tersebut jika routing yang anda buat benar akan menjadi seperti ini

```
+--------+----------+----------------+------+---------+------------+
| Domain | Method   | URI            | Name | Action  | Middleware |
+--------+----------+----------------+------+---------+------------+
|        | GET|HEAD | /              |      | Closure | web        |
|        | GET|HEAD | api/user       |      | Closure | api        |
|        |          |                |      |         | auth:api   |
|        | GET|HEAD | mahasiswa      |      | Closure | web        |
|        | POST     | mahasiswa      |      | Closure | web        |
|        | PUT      | mahasiswa      |      | Closure | web        |
|        | DELETE   | mahasiswa      |      | Closure | web        |
|        | GET|HEAD | mahasiswa/{id} |      | Closure | web        |
|        | PUT      | mahasiswa/{id} |      | Closure | web        |
|        | DELETE   | mahasiswa/{id} |      | Closure | web        |
+--------+----------+----------------+------+---------+------------+
```

Jika anda membutuhkan route yang dapat memiliki lebih dari satu http method routing nya dapat dibuat dengan cara seperti ini.

```
Route::match(['get', 'post'], '/specialUrl', function () {
});
Route::any('/specialMahasiswa', function ($id) {
});
```

```
+--------+-----------------------------------------+-----------------+------+---------+------------+
| Domain | Method                                  | URI             | Name | Action  | Middleware |
+--------+-----------------------------------------+-----------------+------+---------+------------+
|        | GET|HEAD                                | /               |      | Closure | web        |
|        | GET|HEAD                                | api/user        |      | Closure | api        |
|        |                                         |                 |      |         | auth:api   |
|        | GET|HEAD|POST|PUT|PATCH|DELETE|OPTIONS | specialMahasiswa |      | Closure | web        |
|        | GET|POST|HEAD                           | specialUrl      |      | Closure | web        |
+--------+-----------------------------------------+-----------------+------+---------+------------+
```

Dependency Injection

Route pada laravel dapat dilakukan dependency injection dimana sebuah route dapat dititipkan dependency yang dibutuhkan, Laravel akan meresolve dependency ini secara otomatis dan ikut dikirim di callback function yang dimiliki route. Contoh nya pada route dibawah ini kita menitipkan dependency Request pada route /users

```
use Illuminate\Http\Request;

Route::get('/users', function (Request $request) {
    // ...
});
```

CSRF Protection

Semua routes yang didefinisikan di web.php harus menyertakan CSRF Token untuk pemrosesan http verb POST,PUT,PATCH atau DELETE jika tidak memiliki token CSRF request nya akan ditolak. Berikut ini cara menambahkan csrf ke form yang ada di template html

```
<form method="POST" action="/profile">
    @csrf
    ...
</form>
```

Redirect Routes

Untuk melakukan redirect pada laravel dapat dilakukan dengan menggunakan Route::redirect cara penggunaannya dapat dilihat pada kode program dibawah ini.

```
Route::redirect('/here', '/there');
```

Anda akan sering menggunakan redirect ini pada kasus kasus CRUD atau kasus lain yang membutuhkan redirect.

View Routes

Laravel juga menyediakan sebuah route khusus yang memudahkan anda membuat sebuah routes tanpa menggunakan controller atau callback function routes ini langsung menerima input berupa url dan mengembalikan view / tampilan. Berikut ini cara membuat view routes.

```
Route::view('/welcome', 'welcome');
Route::view('/welcome', 'welcome', ['name' => 'Taylor']);
```

Perhatikan pada view routes diatas /welcome akan menampilkan view welcome dan pada route kedua /welcome akan menampilkan view welcome dengan tambahan data berupa variabel name.

Route Parameters

Pada kasus tertentu anda akan memerlukan salah satu segment dari url sebagai parameter **wajib** untuk controller atau callback function pada routing. Contohnya anda membutuhkan id user yang dikirim melalui sebuah URL. Untuk membuat routingnya, dapat dilakukan dengan cara berikut ini

```
Route::get('/user/{id}', function ($id) {
    return 'User '.$id;
});

Route::get('/posts/{post}/comments/{comment}', function ($postId,
$commentId) {
    //
});
```

Perhatikan cara penulisan parameter dan cara penggunaan parameter di callback function / controller. Parameter ditulis menggunakan kurung kurawal {} dan variabel yang bersesuaian disusun sebagai callback param di callback function perhatikan urutannya, urutan ini juga berlaku untuk route yang menggunakan controller function.

Jika sebuah route membutuhkan parameter wajib namun juga membutuhkan dependency dari dependency injection semua parameter ditulis setelah penulisan dependency, berikut ini contoh routenya.

```
Route::get('/user/{id}', function (Request $request, $id) {
    return 'User '.$id;
});
```

Sedangkan untuk parameter yang opsional atau parameter yang tidak selalu ada di route maka anda dapat menggunakan optional parameter di route yang anda buat. Berikut ini cara membuat optional param di routing.

```
Route::get('/user/{name?}', function ($name = null) {
    return $name;
});

Route::get('/user/{name?}', function ($name = 'John') {
    return $name;
});
```

Perhatikan pada function di atas dapat anda lihat sebuah parameter opsional diberikan tanda ? dan dapat dibuat nilai awal nya pada callback function atau controller function nya.

Route Name

Sebuah route bisa diberi nama custom untuk mempermudah pembuatan url pada saat melakukan coding. Contoh penamaan pada routing dapat dilihat pada kode program berikut.

```
Route::get('/user/profile', function () {
    //
})->name('profile');

Route::get(
    '/user/profile',
    [UserProfileController::class, 'show']
)->name('profile');

// Generating URLs...
$url = route('profile');

// Generating Redirects...
return redirect()->route('profile');
```

Route Group

Beberapa route yang memiliki atribut yang sama seperti middleware yang sama dapat dikelompokkan menjadi satu kelompok untuk mempermudah penulisan route selain digunakan untuk middleware masih ada lagi penggunaan route group untuk route yang berada dibawah satu subdomain. Contoh penggunaan route group adalah sebagai berikut :

```
Route::middleware(['first', 'second'])->group(function () {
    Route::get('/', function () {
        // Uses first & second middleware...
    });
    Route::get('/user/profile', function () {
        // Uses first & second middleware...
    });
});
Route::domain('{account}.example.com')->group(function () {
    Route::get('user/{id}', function ($account, $id) {
```

```
        //
    });
});
```

Route Prefix

Pengelompokan route juga dapat dilakukan untuk route yang memiliki prefix (awalan) yang sama contoh pembuatan route dengan prefix dapat dilihat pada kode program dibawah ini

```
Route::prefix('admin')->group(function () {
    Route::get('/users', function () {
        // Matches The "/admin/users" URL
    });
});
```

Controller

Controller digunakan untuk mengorganisasi logika aplikasi menjadi lebih terstruktur. Logika action aplikasi yang masih ada kaitan dapat dikumpulkan dalam satu kelas Controller. Atau sebuah Controller dapat juga hanya berisi satu buah action. Controller pada Laravel disimpan dalam folder `app/Http/Controllers`.

Untuk membuat controller pada Laravel telah disediakan perintah untuk men-generate struktur dasarnya. Anda dapat menggunakan perintah artisan diikuti dengan definisi nama controller yang akan dibuat. Silahkan perhatikan contoh perintah berikut:

```
php artisan make:controller WelcomeController
```

Silahkan buka file pada `app/Http/Controllers/WelcomeController.php`. Struktur pada controller dapat digambarkan sebagai berikut:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class WelcomeController extends Controller
{
      //
}
```

Untuk mendefinisikan action, silahkan tambahkan function dengan access public. Sehingga controller di atas menjadi sebagai berikut:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
```

```
class WelcomeController extends Controller
{
      public function hello() {
          return 'Hello World';
      }
}
```

Setelah sebuah controller telah didefinisikan action, kemudian anda dapat menambahkan controller tersebut pada route. Sehingga untuk route menjadi berikut:

Contoh route lama dan baru

```
Route::get('/hello', function () {
    return 'Hello World';
});
```

```
use App\Http\Controllers\WelcomeController;

Route::get('/hello', [WelcomeController::class,'hello']);
```

Untuk dokumentasi lebih lengkap terkait controller, anda dapat membaca dokumentasi official dari Laravel pada tautan https://laravel.com/docs/8.x/controllers.

Resource Controller

Khusus untuk controller yang terhubung dengan Eloquent model dan dapat dilakukan operasi CRUD terhadap model Eloquent tersebut dapat dibuatkan sebuah controller yang bertipe Resource Controller dimana dengan membuat sebuah resource controller anda akan di buatkan sebuah controller yang lengkap dengan method method yang mendukung proses CRUD dan sebuah route resource yang menampung route untuk controller tersebut. Untuk membuatnya dilakukan dengan menjalankan perintah berikut ini di terminal.

```
php artisan make:controller PhotoController --resource
```

Perintah ini akan men generate sebuah controller dengan nama PhotoController yang berisi method method standar untuk proses CRUD.

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class PhotoController extends Controller
{
      /**
      * Display a listing of the resource.
      *
      * @return \Illuminate\Http\Response
      */
      public function index()
```

```
    {
    //
    }

    /**
    * Show the form for creating a new resource.
    *
    * @return \Illuminate\Http\Response
    */
    public function create()
    {
    //
    }

    /**
    * Store a newly created resource in storage.
    *
    * @param  \Illuminate\Http\Request  $request
    * @return \Illuminate\Http\Response
    */
    public function store(Request $request)
    {
    //
    }

    /**
    * Display the specified resource.
    *
    * @param  int  $id
    * @return \Illuminate\Http\Response
    */
    public function show($id)
    {
    //
    }

    /**
    * Show the form for editing the specified resource.
    *
    * @param  int  $id
    * @return \Illuminate\Http\Response
    */
    public function edit($id)
    {
    //
    }

    /**
    * Update the specified resource in storage.
    *
    * @param  \Illuminate\Http\Request  $request
    * @param  int  $id
    * @return \Illuminate\Http\Response
    */
    public function update(Request $request, $id)
```

```
        {
        //
        }

        /**
        * Remove the specified resource from storage.
        *
        * @param  int  $id
        * @return \Illuminate\Http\Response
        */
        public function destroy($id)
        {
        //
        }
}
```

Setelah controller berhasil di generate selanjutnya harus dibuatkan route agar dapat terhubung dengan frontend tambahkan kode program berikut pada file web.php.

```
Route::resource('photos', PhotoController::class);
```

Jika sekarang dijalankan cek list route akan dihasilkan route berikut ini

Domain	Method	URI	Name	Action	Middleware
	GET|HEAD	/		Closure	web
	GET|HEAD	api/user		Closure	api auth:api
	GET|HEAD	photos	photos.index	App\Http\Controllers\PhotoController@index	web
	POST	photos	photos.store	App\Http\Controllers\PhotoController@store	web
	GET|HEAD	photos/create	photos.create	App\Http\Controllers\PhotoController@create	web
	GET|HEAD	photos/{photo}	photos.show	App\Http\Controllers\PhotoController@show	web
	PUT|PATCH	photos/{photo}	photos.update	App\Http\Controllers\PhotoController@update	web
	DELETE	photos/{photo}	photos.destroy	App\Http\Controllers\PhotoController@destroy	web
	GET|HEAD	photos/{photo}/edit	photos.edit	App\Http\Controllers\PhotoController@edit	web
	GET|HEAD|POST|PUT|PATCH|DELETE|OPTIONS	specialMahasiswa		Closure	web
	GET|POST|HEAD	specialUrl		Closure	web

Perhatikan pada route list semua route yang berhubungan untuk crud photo sudah di generate oleh laravel. Jika tidak semua route pada resource controller dibutuhkan dapat dikurangi dengan mengupdate route pada web.php menjadi seperti berikut ini.

```
Route::resource('photos', PhotoController::class)->only([
    'index', 'show'
]);

Route::resource('photos', PhotoController::class)->except([
    'create', 'store', 'update', 'destroy'
]);
```

Perhatikan kita dapat menggunakan keyword only atau except sesuai dengan kebutuhan controller yang dibuat.

PRAKTIKUM

Praktikum 1 - Routing Web Framework Laravel

1. Buatlah project laravel baru beri nama minggu2_prak1.
2. Buatlah routing untuk url dengan requirement sebagai berikut, gunakanlah callback function berupa anonymous function untuk mengeluarkan output.

```
Route::get('/', function () {
      echo "Selamat Datang";
});
```

URL	Output
/	Menampilkan pesan 'Selamat Datang'
/about	Menampilkan NIM dan nama anda
/articles/{id}	Menampilkan Output "Halaman Artikel dengan ID {id}" ganti id sesuai dengan input dari url

3. Gunakan konsep route parameter sehingga dapat menangani request pada URL dengan pola `/articles/1`, `/articles/2`. Tampilkan id parameter sehingga ketika URL tersebut diakses menampilkan pesan 'Halaman artikel dengan id 1' jika yang dimasukkan adalah 1.
4. Simpan praktikum ini dalam project anda menggunakan Git publish ke github dengan nama minggu2_prak1 ke repository github anda.

Praktikum 2 - Controller Web Framework Laravel

1. Buatlah project laravel baru beri nama minggu2_prak2.
2. Modifikasi Praktikum 1 dengan konsep controller. Pindahkan logika eksekusi ke dalam controller dengan nama `PageController`. Sesuaikan juga routing yang digunakan. Simpan perubahan ini dengan menggunakan Git.

Resource	GET
/	Tampilkan Pesan 'Selamat Datang' → `PageController : index`
/about	Tampilkan Nama dan NIM → `PageController : about`
/articles/{id}	Tampilkan halaman dinamis 'Halaman Artikel dengan Id {id}' id diganti sesuai input dari url → `PageController : articles`

3. Modifikasi kembali implementasi sebelumnya dengan konsep Single Action Controller. Sehingga untuk hasil akhir yang didapatkan anda mempunyai `HomeController`, `AboutController` dan `ArticleController`. Hapus route ke PageController pada project anda. Simpan perubahan ini dengan menggunakan Git.
4. Simpan praktikum ini dalam project anda menggunakan Git publish ke github dengan nama minggu2_prak2 ke repository github anda.

Praktikum 3 - Desain Routing Web Company Profile

Jurusan Teknologi Informasi Politeknik Negeri Malang meminta anda untuk membuatkan company profile menggunakan Laravel, web ini membutuhkan fitur sebagai berikut :

1	Halaman Home Menampilkan halaman awal website
2	Halaman Program Studi Menampilkan daftar program studi (route prefix) /prodi/manajemen-informatika /prodi/teknik-informatika
3	Halaman News Menampilkan Daftar berita (route param) /news/1 /news/2 /news/3
4	Halaman Sarana Menampilkan Daftar Sarana (route prefix) /sarana/perkantoran /sarana/laboratorium /sarana/kelas /sarana/lainnya
5	Halaman About Us Menampilkan About Us (route biasa)
6	Halaman Comment Menampilkan isi dari parameter nama dan pesan (route param)

1. Buatlah project laravel baru beri nama minggu2_prak3.
2. Buatlah routing untuk website ini menggunakan konsep routing pada laravel serta buatlah controller yang dipanggil oleh route tersebut.
3. Simpan perubahan pada code dengan menggunakan Git.
4. Simpan praktikum ini dalam project anda menggunakan Git publish ke github dengan nama minggu2_prak3 ke repository github anda.