REST API Menggunakan Slim Framework

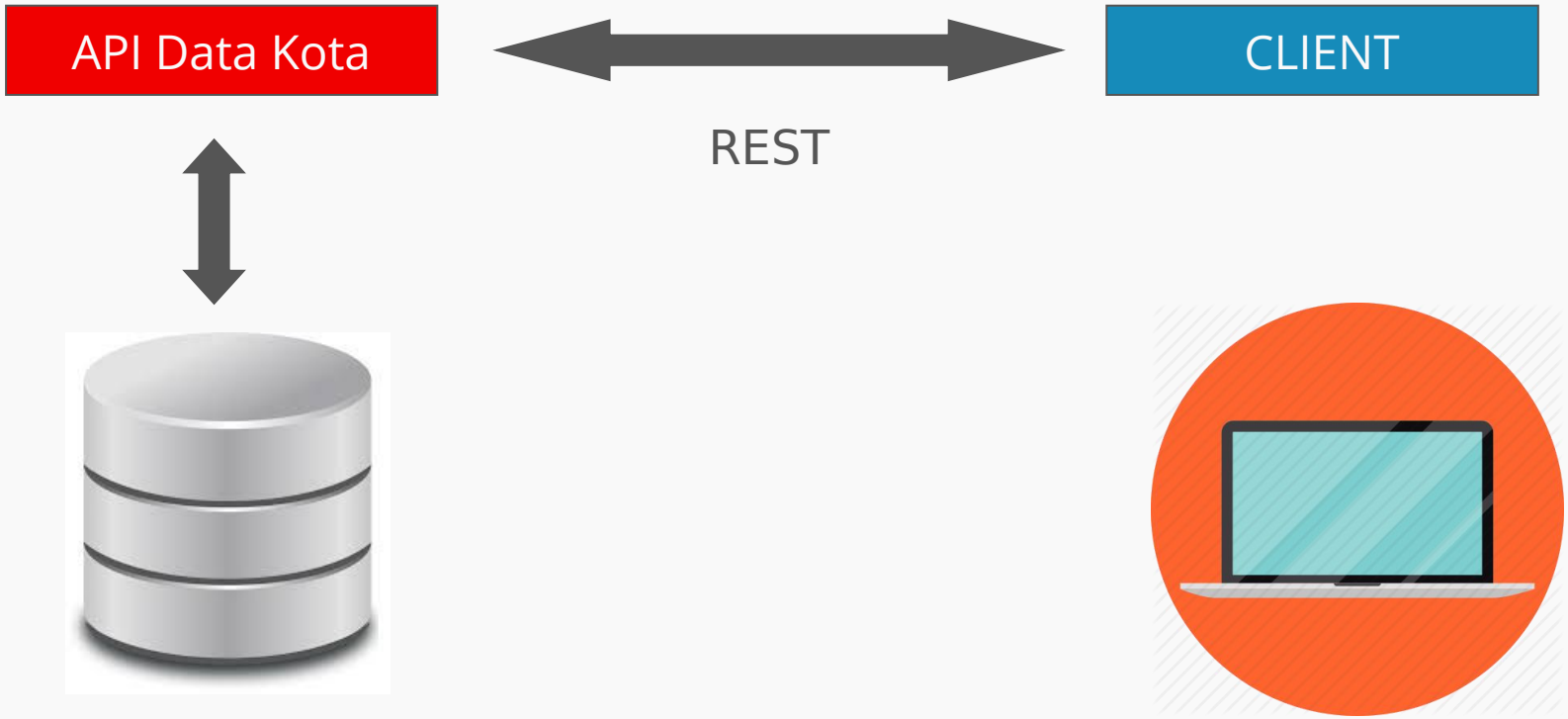
Todi Adiyatmo Wijoyo S.T.,MIT

Pendahuluan

Tujuan Tutorial Ini

- Membuat Web Service dengan arsitektur client-*server*
- Memperkenalkan Slim Framework dan Composer

Arsitektur



Data *Dummy* :
Untuk keperluan tutorial ini, telah disiapkan 100 data nama kota.

Teknologi

- Slim Framework
  - *Micro Framework*, sebuah framework yang mudah dipahami tanpa perlu pengetahuan terhadap pola MVC
  - Kita bebas menentukan sendiri struktur aplikasi yang kita buat
- Composer
  - Package Manager PHP

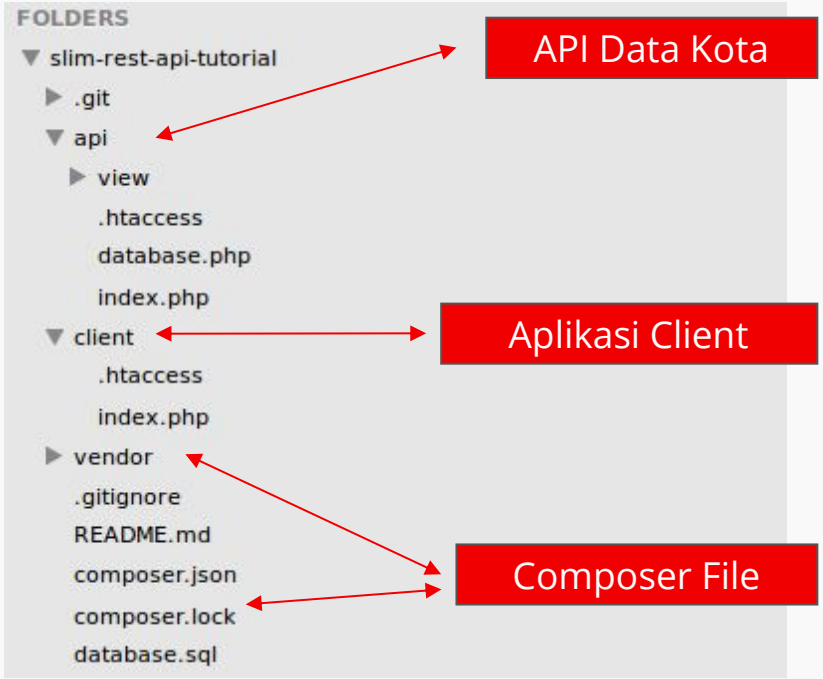
Persiapan

Persiapan - 1

- *Clone repository* untuk tutorial ini :
  - https://github.com/todiadiyatmo/slim-rest-api-tutorial

Persiapan - 2

- Struktur Folder



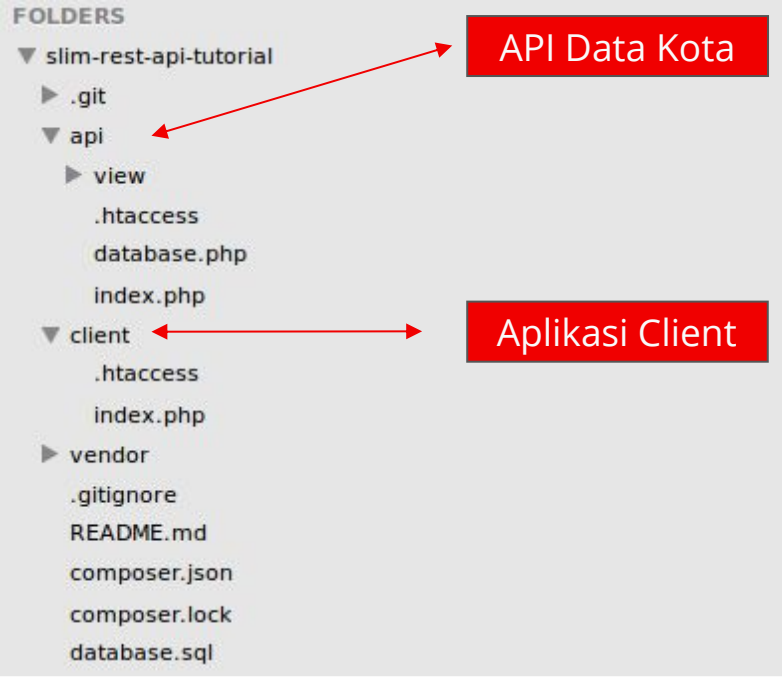
API Data Kota:
Akses ke database dan manipulasi data ke database dilakukan dari aplikasi ini

Aplikasi Client Side :
Front-end atau presentasi dari aplikasi yang kita buat. Memanfaatkan REST API untuk mengakses server

Catatan Composer :
Seharusnya repo git tidak menyertakan folder `vendor` dan file `composer.lock`. File tersebut akan di-*generate* ketika kita melakukan perintah `composer install`. Namun untuk kemudahan file tersebut disertakan dalam repo

Persiapan - 3

- Coba akses aplikasi anda pada web browser !

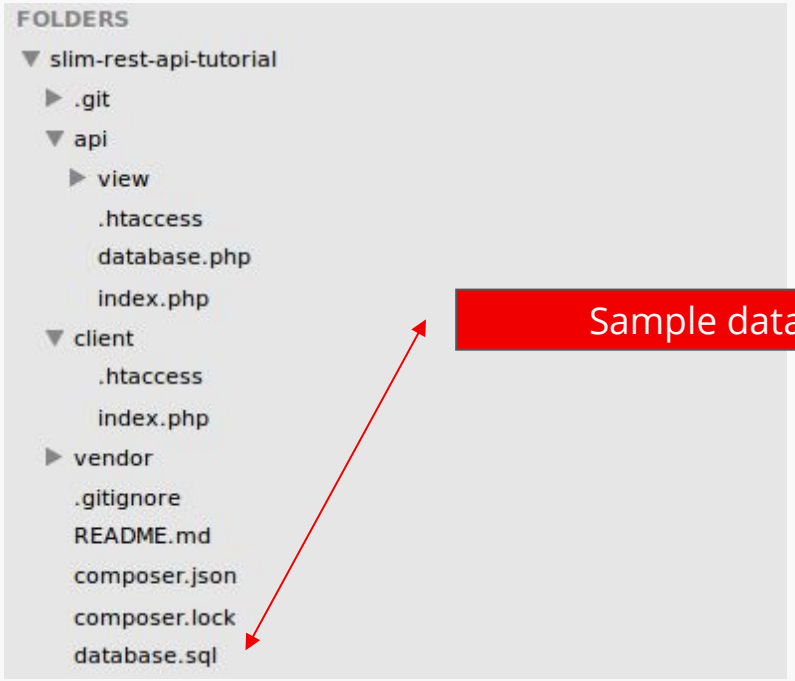


Akses Aplikasi Server Side :
http://[server-anda]/slim-rest-api-tutorial/api/index.php

Akses Aplikasi Client Side :
http://[server-anda]/slim-rest-api-tutorial/client/index.php

Persiapan - 4

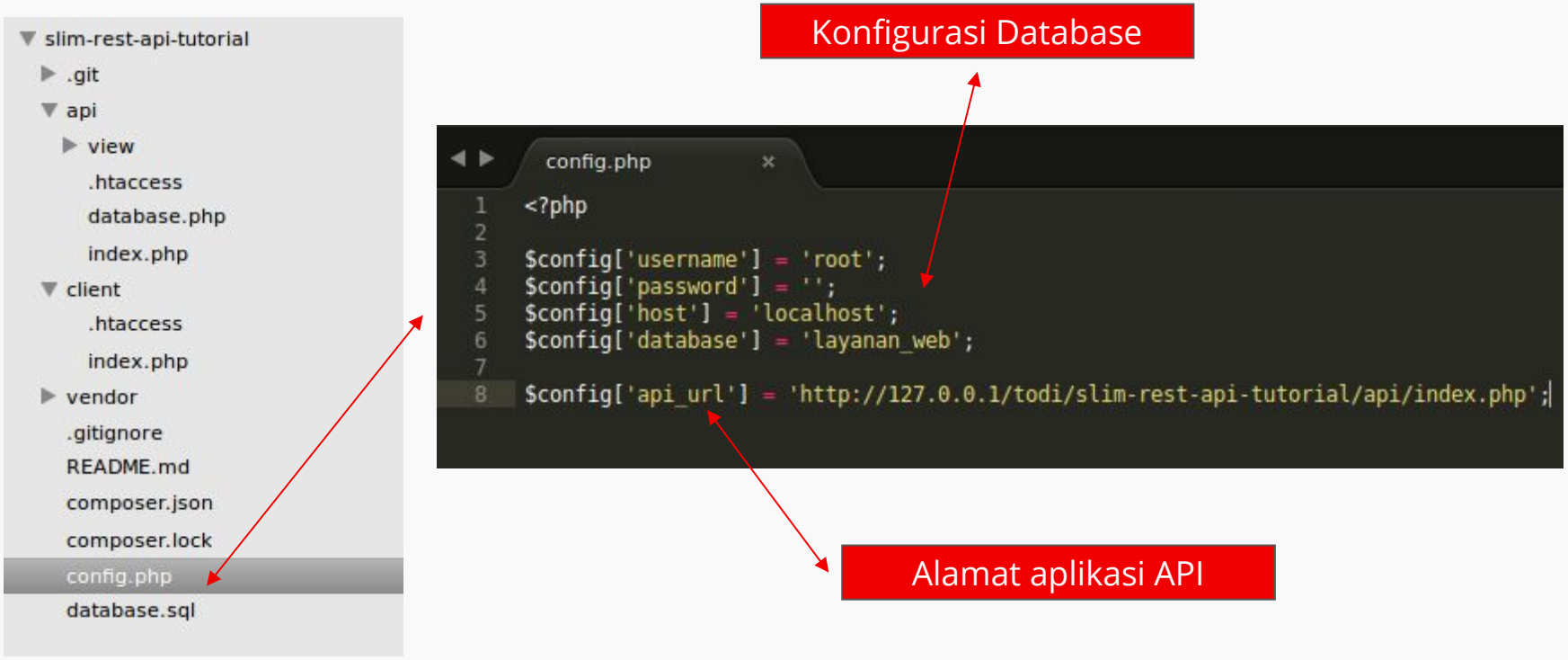
- Buat Database baru



Data *Dummy* :
Buatlah database baru di komputer anda dengan menggunakan file *dump* `database.sql`.

Persiapan - 5

- Buat file konfigurasi "config.php"



Spesifikasi Aplikasi

Route untuk API Data Kota

HTTP Verb	Path	Deskripsi	format
GET	http://[url]/cities	Melihat semua data kota	{cities:[{id:4,name:'yogya'}, {id:5,name:'paris'}]}
GET	http://[url]/cities/:id	Melihat kota dengan :id = id	{id:4,name:'yogya'}
POST	http://[url]/cities	Membuat data kota baru	{name:'solo'}
PUT	http://[url]/cities/:id	Updata data kota	{name:'karawang'}
DELETE	http://[url]/cities/:id	Menghapus kota dengan :id =id	

Route untuk Client

HTTP Verb	Path	Deskripsi
GET	http://[url]/cities	Melihat semua data kota
GET	http://[url]/cities/:id	Melihat kota dengan :id = id
GET	http://[url]/cities/:id/edit	Form untuk mengedit kota dengan :id = id
POST	http://[url]/cities/:id/update	Updata data kota
GET	http://[url]/cities/:id/delete	Menghapus kota dengan :id =id
GET	http://[url]/cities/new	Form untuk membuat kota baru
POST	http://[url]/cities/	Membuat data kota baru

Dibuat untuk kemudahan, form html tidak mendukung PUT dan DELETE

Hubungan API DAta Kota <----> Client

Aplikasi Client

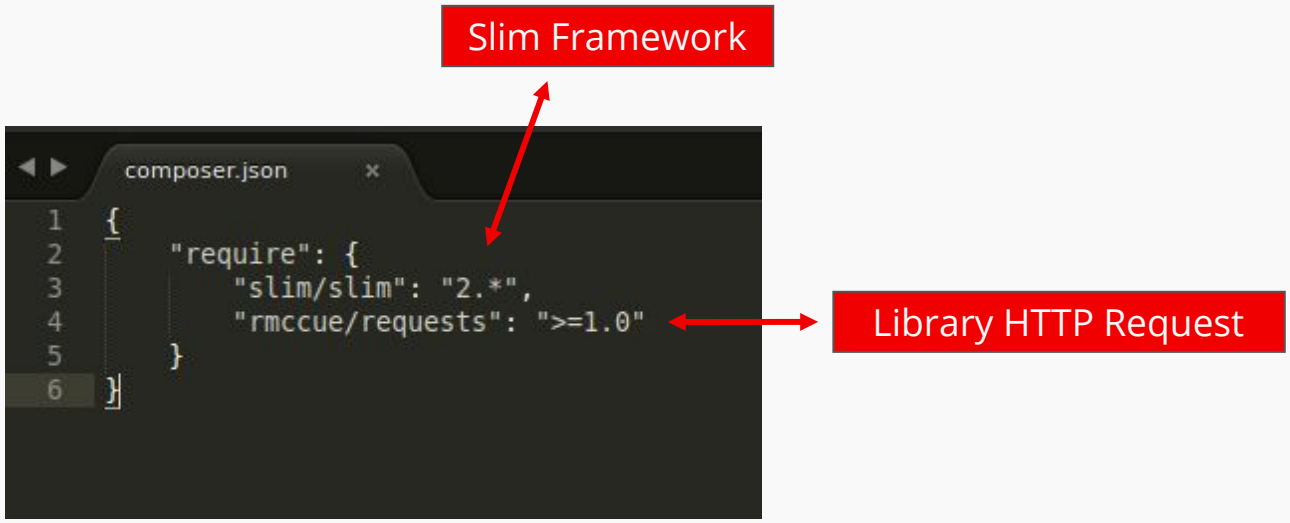
HTTP Verb	Path	
GET	http://[url]/cities	1
GET	http://[url]/cities/:id	2
GET	http://[url]/cities/:id/edit	2
POST	http://[url]/cities/:id/update	3
GET	http://[url]/cities/:id/delete	4
GET	http://[url]/cities/new	5
POST	http://[url]/cities/	

Aplikasi API Data Kota

HTTP Verb	Path	
GET	http://[url]/cities	1
GET	http://[url]/cities/:id	2
PUT	http://[url]/cities/:id	3
DELETE	http://[url]/cities/:id	4
POST	http://[url]/cities	5

Perkenalan Slim dan Composer

Composer



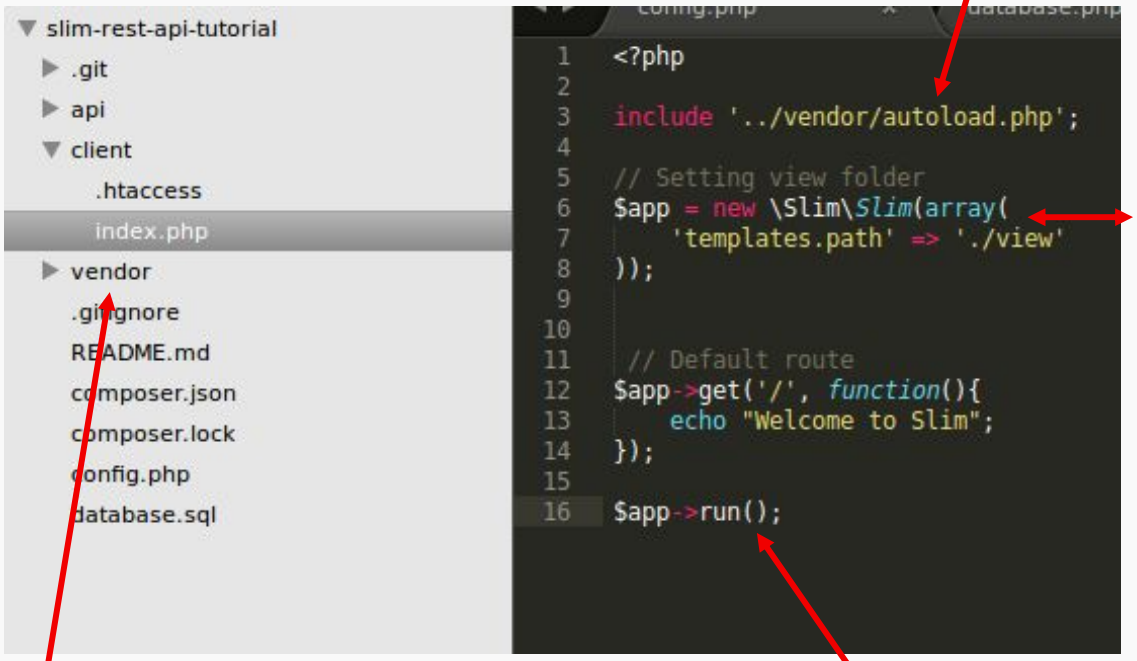
File composer.json
Dalam file ini kita menentukan komponen/*dependency* apa saja yang kita butuhkan dalam aplikasi.

Kita dapat menginstall semua *dependency* dengan perintah :

~$composer install

Proses instalasi ini akan menghasilkan folder `vendor`

SLIM : Index.php



Memanggil autoload composer

Inisiasi Slim

Folder Library Composer

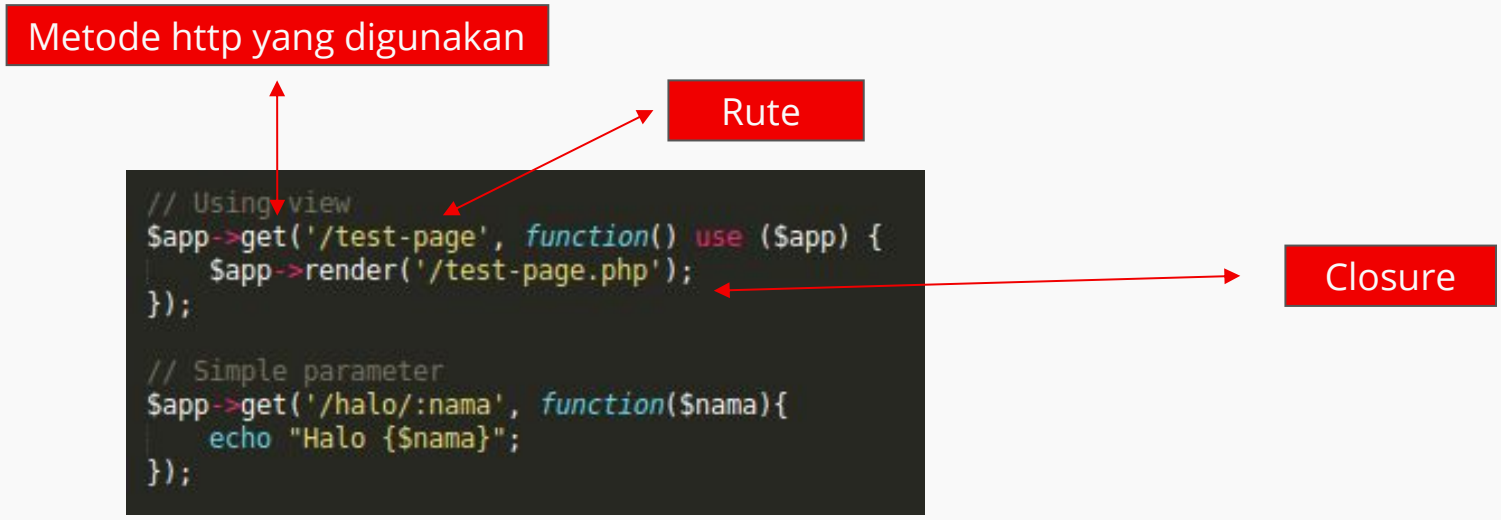
Menjalankan Slim

SLIM : Routing

PHP Closure (Anonymous Function)
Slim memungkinkan kita membuat rute dengan mudah menggunakan *closure*.

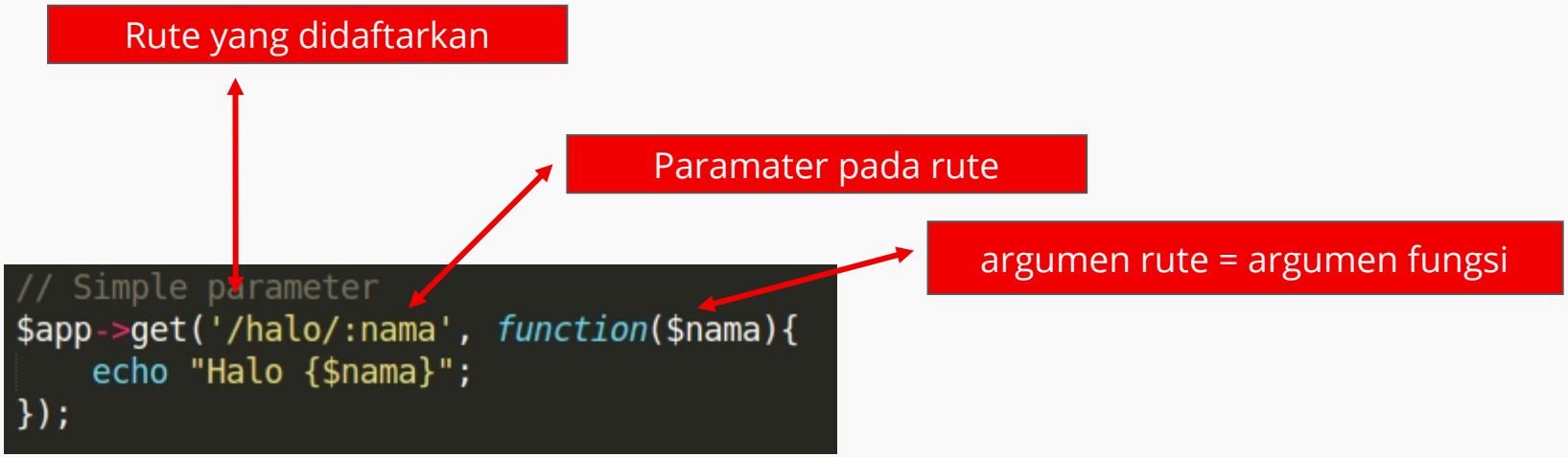
```php
memanggilClosure($parameter);


memanggilClosure(function(){
    // Parameter berupa fungsi anonymous
})
```



SLIM : Routing (2)

Argumen dalam Route
Kita juga dapat membuat sebuah bagian dari URL menjadi argumen fungsi.



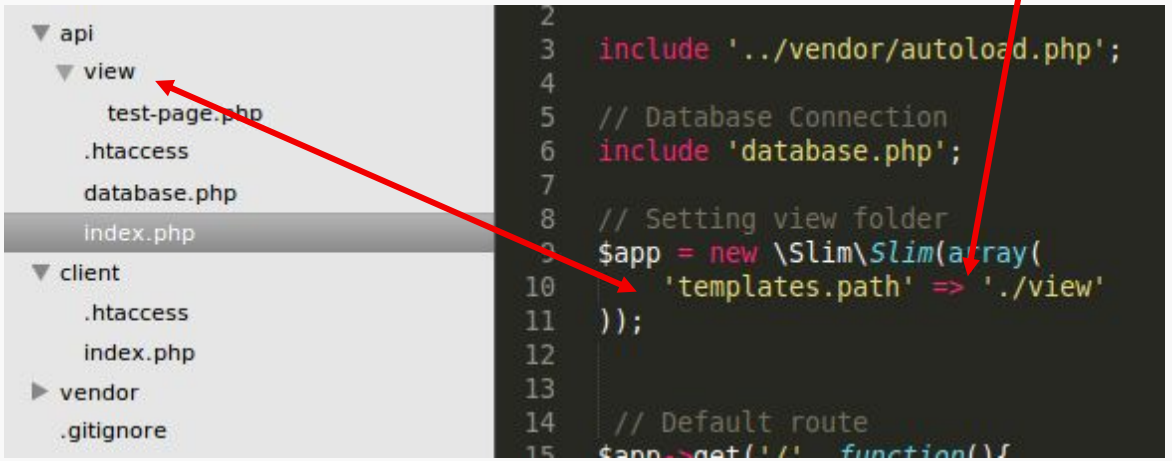
SLIM : Routing (3)

Menggunakan View
Slim juga mendukung penggunaan view dalam rute yang didaftarkan

File yang dipanggil

```
// Using view
$app->get('/test-page', function() use ($app) {
    $app->render('/test-page.php');
});
```

index.php Konfigurasi lokasi view



Mari Mulai :)

Tahap Pembuatan Aplikasi

Aplikasi Client

HTTP Verb	Path	
GET	http://[url]/cities	1
GET	http://[url]/cities/:id	2
GET	http://[url]/cities/:id/edit	2
POST	http://[url]/cities/:id/update	3
GET	http://[url]/cities/:id/delete	4
GET	http://[url]/cities/new	5
POST	http://[url]/cities/	

Aplikasi API Data Kota

HTTP Verb	Path	
GET	http://[url]/cities	1
GET	http://[url]/cities/:id	2
PUT	http://[url]/cities/:id	3
DELETE	http://[url]/cities/:id	4
POST	http://[url]/cities	5

Tahap 1 : Server Side (List city)

Aplikasi API Data Kota

HTTP Verb	Path
GET	http://[url]/cities

/api/index.php

```php
// List all cities
$app->get('/cities', function() use ($app) {

    // Set JSON Header
    $app->response()->header('Content-Type', 'application/json;charset=utf-8');

    try {
        $db = getConnection();

        $sql = "select * FROM cities ORDER BY name";

        $stmt = $db->query($sql);

        $cities = $stmt->fetchAll(PDO::FETCH_ASSOC);

        $db = null;

        echo '{"cities": ' . json_encode($cities) . '}';

    } catch(PDOException $e) {
        echo '{"error":{"text":'. $e->getMessage() .'}}';
    }
});
```

Tahap 1 : Client Side (List city)

Aplikasi Client

HTTP Verb	Path
GET	http://[url]/cities

/client/index.php

```php
// Show all cities
$app->get('/cities', function() use ($app) {

    include '../config.php';

    $headers = array('Accept' => 'application/json');

    $request = Requests::get($config['api_url'].'/cities', $headers);

    $jsonData = json_decode($request->body);

    $baseURL = $app->request->getRootUri();

    $app->render('cities.php',array('cities'=>$jsonData->cities,'baseURL'=>$baseURL));

});
```

/client/view/cities.php

```php
<h1>List of Cities</h1>
<a href="<?php echo $baseURL.'/cities/new'?>">Create an new City</a>
<table>
    <tr>
        <th>ID</th>
        <th>City</th>
        <th></th>
    </tr>
    <?php foreach($cities as $city): ?>
        <tr>
            <td>
                <?php echo $city->id?>
            </td>
            <td>
                <?php echo $city->name?>
            </td>
            <td>
                <a href="<?php echo $baseURL.'/cities/'.$city->id ?>">View</a>
                <a href="<?php echo $baseURL.'/cities/'.$city->id.'/delete' ?>">Delete</a>
            </td>
        </tr>

    <?php endforeach; ?>
</table>
```

Tahap 2 : Server Side (Show single city)

Aplikasi API Data Kota

HTTP Verb	Path
GET	http://[url]/cities/:id

/api/index.php

```php
// Update single city by :id
$app->delete('/cities/:id', function($id) use ($app) {

    // Set JSON Header
    $app->response()->header('Content-Type', 'application/json;charset=utf-8');

    $sql = "DELETE FROM cities WHERE id=:id";

    try {
        $db = getConnection();
        $stmt = $db->prepare($sql);
        $stmt->bindParam("id", $id);
        $stmt->execute();
        $db = null;
        echo json_encode(array('message'=>"Delete city with id=$id success"));
    } catch(PDOException $e) {
        echo '{"error":{"text":'. $e->getMessage() .'}}';
    }
});
```

Tahap 2 : Client Side (show city)

Aplikasi Client

HTTP Verb	Path
GET	http://[url]/cities/:id

/api/index.php

```php
// Show Single City
$app->get('/cities/:id', function($id) use ($app) {

    include '../config.php';

    $headers = array('Accept' => 'application/json');

    $request = Requests::get($config['api_url'].'/cities/'.$id, $headers);

    $city = json_decode($request->body);

    $baseURL = $app->request->getRootUri();

    $app->render('show.php',array('city'=>$city,'baseURL'=>$baseURL));

});
```

Tahap 2 : Client Side (show city form)

Aplikasi Client

HTTP Verb	Path
GET	http://[url]/cities/:id

/api/view/show.php

```
<h1>Show City</h1>

<table>
    <tr>
        <td>
            ID
        </td>
        <td>
            <?php echo $city->id ?>
        </td>
    </tr>
    <tr>
        <td>
            Name
        </td>
        <td>
            <?php echo $city->name ?>
        </td>
    </tr>

    <tr>
        <td colspan="2">
            <a href="<?php echo $baseURL.'/cities' ?>">List All</a>
            <a href="<?php echo $baseURL.'/cities/'.$city->id.'/edit' ?>">Edit</a>
            <a href="<?php echo $baseURL.'/cities/'.$city->id.'/delete' ?>">Delete</a>
        </td>
    </tr>

</table>
```

Tahap 2 : Client Side (Edit city)

Aplikasi Client

HTTP Verb	Path
GET	http://[url]/cities/:id/edit

/api/index.php

```
// Show form to edit single city
$app->get('/cities/:id/edit', function($id) use ($app) {

    include '../config.php';

    $headers = array('Accept' => 'application/json');

    $request = Requests::get($config['api_url'].'/cities/'.$id, $headers);

    $city = json_decode($request->body);

    $baseURL = $app->request->getRootUri();

    $app->render('edit.php',array('city'=>$city,'baseURL'=>$baseURL));

});
```

Tahap 2 : Client Side (Edit city Form)

Aplikasi Client

HTTP Verb	Path
GET	http://[url]/cities/:id/edit

/client/view/edit.php

```php
<h1>Edit City</h1>

<form action="<?php echo $baseURL.'/cities/'.$city->id.'/update' ?>" method='POST'>
    <table>
        <tr>
            <td>
                ID
            </td>
            <td>
                <?php echo $city->id ?>
            </td>
        </tr>
        <tr>
            <td>
                Name
            </td>
            <td>
                <input type='text' name='name' value='<?php echo $city->name ?>'>
            </td>
        </tr>

        <tr>
            <td colspan="2">
                <a href="<?php echo $baseURL.'/cities' ?>">List All</a>
                <input type='submit' name='submit' value="Update">
            </td>
        </tr>

    </table>
</form>
```

Tahap 3 : Server Side (Update city)

Aplikasi API Data Kota

HTTP Verb	Path
PUT	http://[url]/cities/:id

/api/index.php

```php
// Update single city by :id
$app->put('/cities/:id', function($id) use ($app) {

    // Set JSON Header
    $app->response()->header('Content-Type', 'application/json;charset=utf-8');

    $body = $app->request->getBody();

    $city = json_decode($body);

    $sql = "UPDATE cities SET name=:name WHERE id=:id";
    try {
        $db = getConnection();
        $stmt = $db->prepare($sql);
        $stmt->bindParam("name", $city->name);
        $stmt->bindParam("id", $id);
        $stmt->execute();
        $db = null;
        echo json_encode(array('message'=>"Update city with id=$id success"));
    } catch(PDOException $e) {
        echo '{"error":{"text":'. $e->getMessage() .'}}';
    }
});
```

Tahap 3 : Client Side (Update city)

Aplikasi Client

HTTP Verb	Path
POST	http://[url]/cities/:id/update

/client/index.php

```php
// Update Single City
$app->post('/cities/:id/update', function($id) use ($app) {

    include '../config.php';

    $headers = array('Content-Type' => 'application/json');

    $data['name'] = $_POST['name'];

    $data['id'] = $id ;

    // Remember to encode the data to json when calling the API on the other side
    $request = Requests::put($config['api_url'].'/cities/'.$id, $headers,json_encode($data));

    echo $request->body;

});
```

Tahap 4 : Server Side (Delete city)

Aplikasi API Data Kota

HTTP Verb	Path
DELETE	http://[url]/cities/:id

/api/index.php

```php
// Delete single city by :id
$app->delete('/cities/:id', function($id) use ($app) {

    // Set JSON Header
    $app->response()->header('Content-Type', 'application/json;charset=utf-8');

    $sql = "DELETE FROM cities WHERE id=:id";

    try {
        $db = getConnection();
        $stmt = $db->prepare($sql);
        $stmt->bindParam("id", $id);
        $stmt->execute();
        $db = null;
        echo json_encode(array('message'=>"Delete city with id=$id success"));
    } catch(PDOException $e) {
        echo '{"error":{"text":'. $e->getMessage() .'}}';
    }
});
```

Tahap 4 : Client Side (Delete city)

Aplikasi Client

HTTP Verb	Path
GET	http://[url]/cities/:id/delete

/client/index.php

```php
// Delete Single City
$app->get('/cities/:id/delete', function($id) use ($app) {

    include '../config.php';

    $headers = array('Content-Type' => 'application/json');

    // Remember to encode the data to json when calling the API on the other side
    $request = Requests::delete($config['api_url'].'/cities/'.$id, $headers);

    echo $request->body;

});
```

Tahap 5 : Server Side (Create city)

Aplikasi API Data Kota

HTTP Verb	Path
POST	http://[url]/cities/

/api/index.php

```php
// Create single city by :id
$app->post('/cities', function() use ($app) {

    // Set JSON Header
    $app->response()->header('Content-Type', 'application/json;charset=utf-8');

    $request = $app->request();

    $city = json_decode($request->getBody());

    $sql = "INSERT INTO cities (name) VALUES (:name)";
    try {
        $db = getConnection();
        $stmt = $db->prepare($sql);
        $stmt->bindParam("name", $city->name);
        $stmt->execute();
        $city->id = $db->lastInsertId();

        $db = null;

        echo json_encode(array('message'=>'Success create a new city','city'=>$city));

    } catch(PDOException $e) {
        echo '{"error":{"text":'. $e->getMessage() .'}}';
    }
});
```

Tahap 5 : Client Side (Create city form)

Aplikasi Client

HTTP Verb	Path
GET	http://[url]/cities/new

/client/index.php

```php
// Show Form to Create Single City
$app->get('/cities/new', function() use ($app) {

    include '../config.php';

    $baseURL = $app->request->getRootUri();

    $app->render('new.php',array('baseURL'=>$baseURL));

});
```

Tahap 5 : Client Side (Create city form)

Aplikasi Client

HTTP Verb	Path
GET	http://[url]/cities/new

/client/view/new.php

```
<h1>Create a New City</h1>

<form action="<?php echo $baseURL.'/cities' ?>" method='POST'>
    <table>
        <tr>
            <td>
                Name
            </td>
            <td>
                <input type='text' name='name' value=''>
            </td>
        </tr>

        <tr>
            <td colspan="2">
                <a href="<?php echo $baseURL.'/cities' ?>">List All</a>
                <input type='submit' name='submit' value="Create">
            </td>
        </tr>

    </table>
</form>
```

Tahap 5 : Client Side (Create city)

Aplikasi Client

HTTP Verb	Path
POST	http://[url]/cities/

/client/index.php

```php
// Create Single City
$app->post('/cities/', function() use ($app) {

    include '../config.php';

    $headers = array('Content-Type' => 'application/json');

    $data['name'] = $_POST['name'];

    // Remember to encode the data to json when calling the API on the other side
    $request = Requests::post($config['api_url'].'/cities', $headers,json_encode($data));

    echo $request->body;

});
```

Terimakasih