



WEB DEVELOPMENT

JUNIOR WEB PROGRAMMER

Universitas Komputer Indonesia

Hello!

Lilis Ayumsari, S.Kom, MTA, PSM-I

Instructure (web)

Developer @Refrada

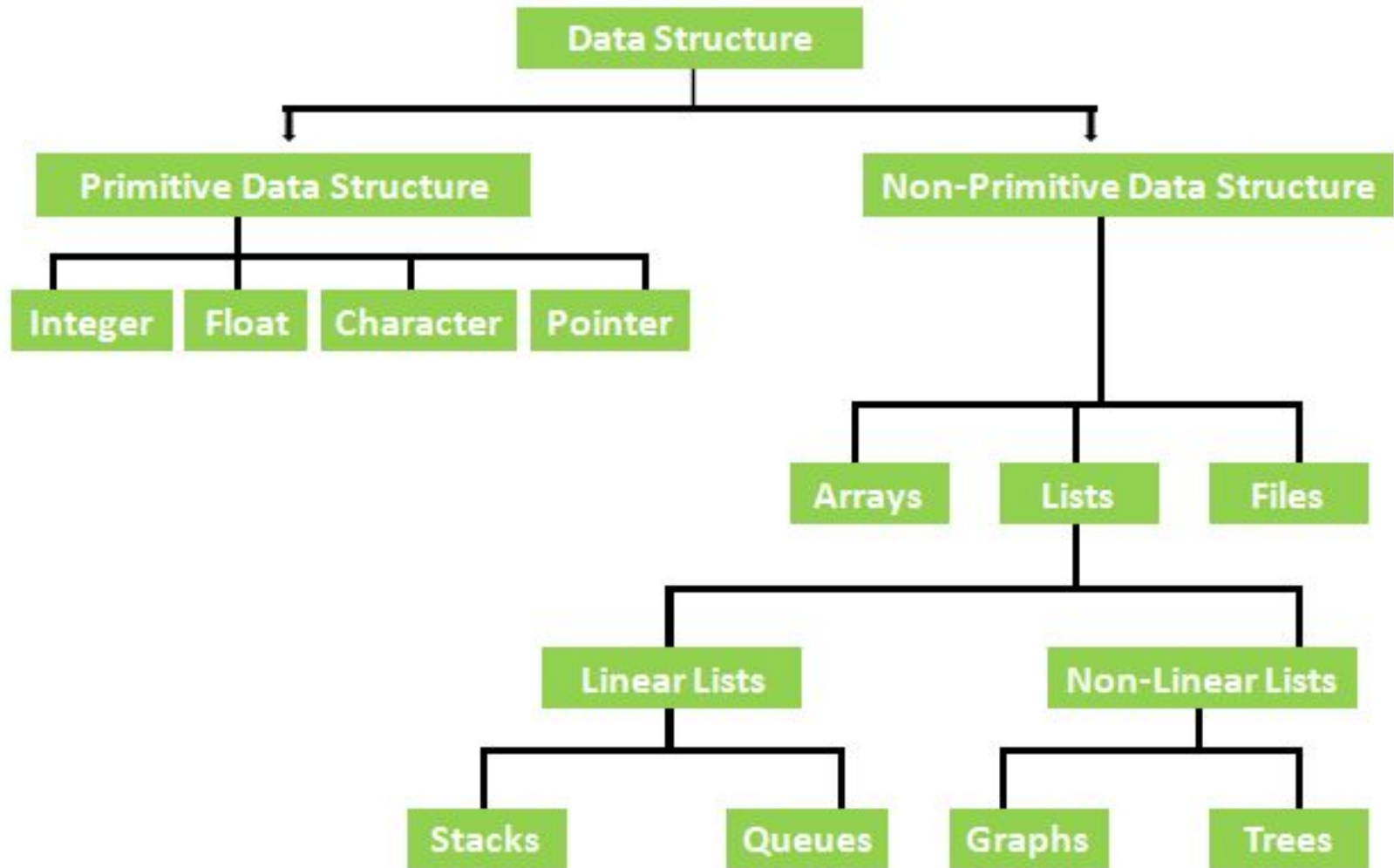
PMO @BIGIO



- Menggunakan struktur data
- Melakukan instalasi software tools pemrograman
- Menulis kode dengan prinsip sesuai guidelines dan best practices
- Membuat dokumen kode program
- Mengimplementasikan User Interface
- Menggunakan library atau komponen pre-existing
- Melakukan debugging



Tipe Struktur Data

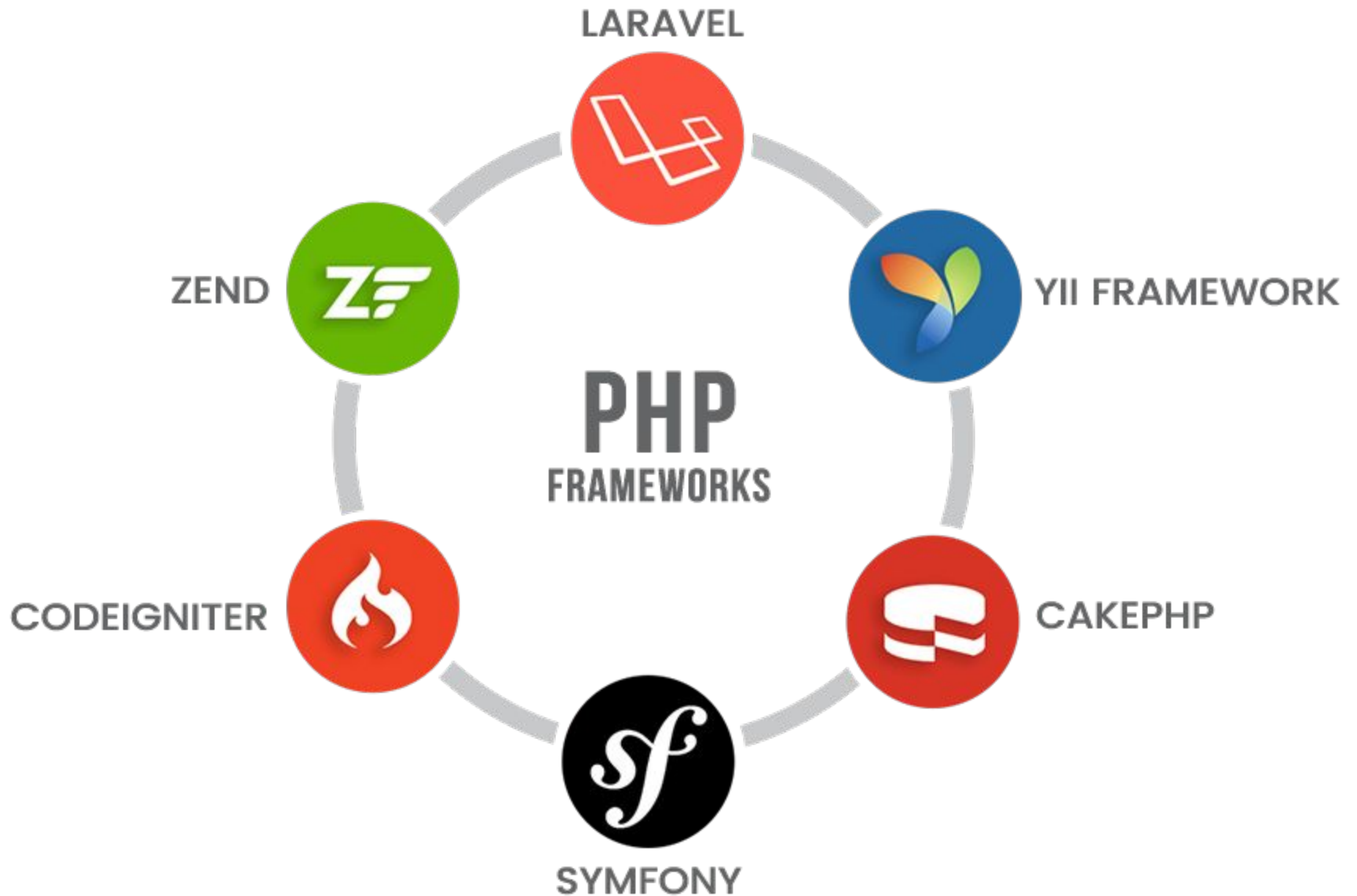


Software & Library

- XAMPP 7.0.30
(<https://www.apachefriends.org/index.html>)
- Postman (<https://www.getpostman.com>)
- Heidi (<https://www.heidisql.com/download.php>)
- Codeigniter versi 3.19 (<https://codeigniter.com>)
- Library Codeigniter-rest server
(<https://github.com/chriskacerguis/codeigniter-restserver>) Versi 3.0.0 (sebelum update juni)



Framework PHP (Tech Stack)



- Codeigniter merupakan framework PHP yang menerapkan konsep MVC (Model View Controller).
- Waktu eksekusi cepat
- Ukuran kecil
- Konfigurasi minimal, tidak perlu install cukup dengan dicopy
- Dokumentasi yang lengkap, sehingga mempermudah pengguna
- Tersedia banyak library pendukung

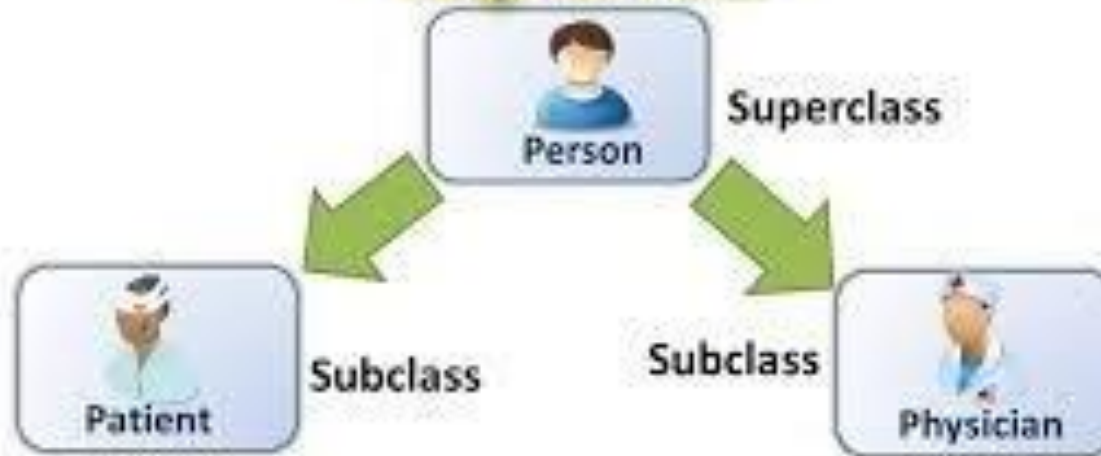


Dasar Penulisan Kode

- Mengomentari & Mendokumentasikan (Komentar berlebihan)
- Konsisten Indentasi
- Pengelompokan Kode
- Konsisten Skema Penamaan (Case sensitive)
- Sarang yang dalam (Deepnest)
- Batas panjang Baris
- Organisasi File Folder
- OOP, prosedural, dll



Object Oriented Programming Explained



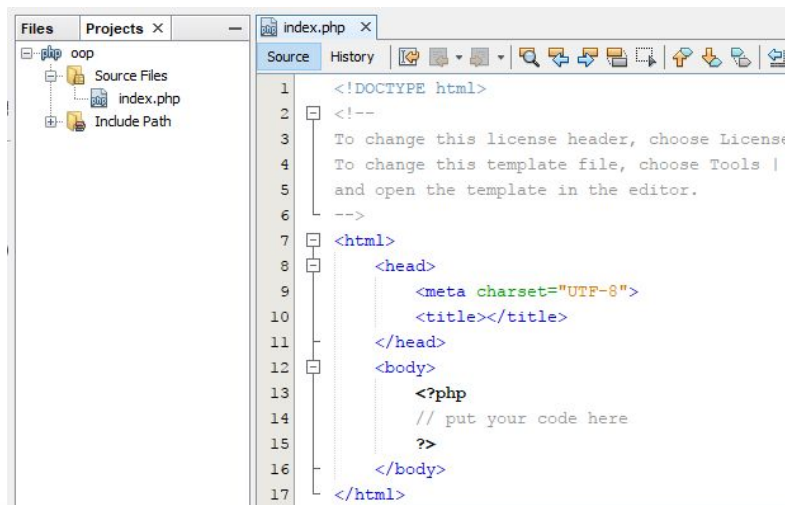
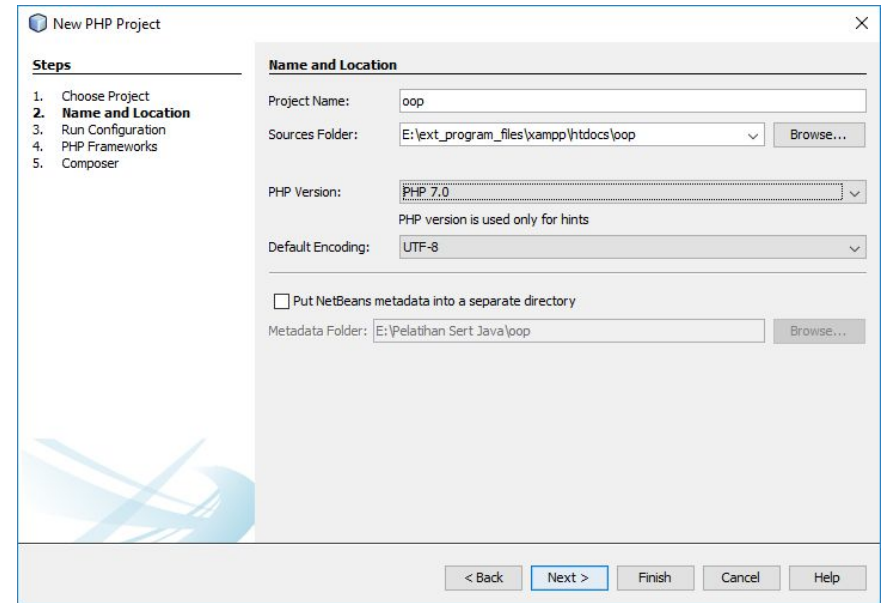
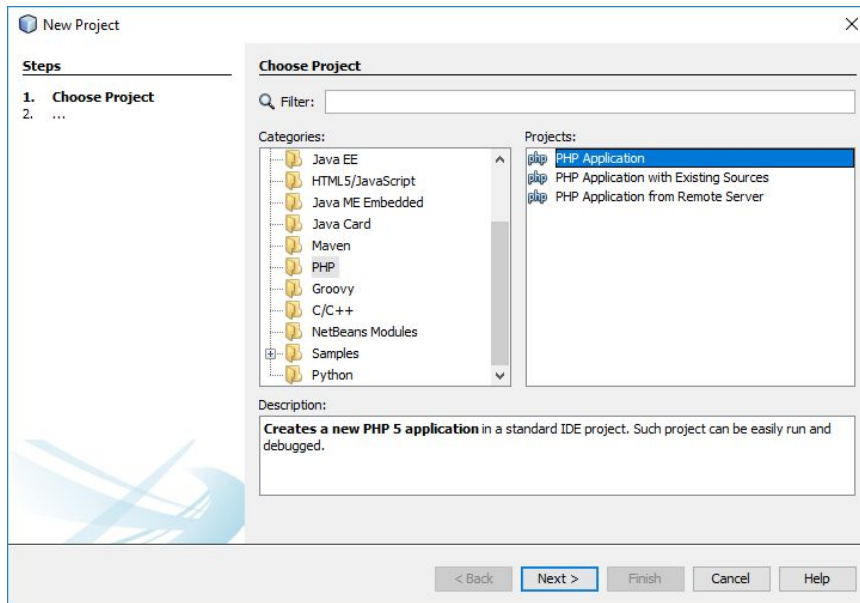
Object Oriented Programming

OOP

- OOP adalah suatu metode pemrograman yang berorientasi kepada objek, memodelkan objek di dunia nyata ke perangkat lunak
- Class merupakan blue print untuk membuat object
- Class terdiri dari
 - Property/attribute/field
 - Function/method
- Object merupakan hasil cetakan dari class tersebut



New Project



Contoh

person.php x test_person.php test_mahasiswa.php mahasiswa.php

```
1  <?php
2  class Person{
3      var $nama;
4      var $alamat;
5
6      function __construct($nama,$alamat) {
7          $this->nama = $nama;
8          $this->alamat = $alamat;
9      }
10
11     function tampil(){
12         echo 'nama = ' . $this->nama . ',alamat = ' . $this->alamat;
13     }
14 }
15 ?>
```

```
1  <?php
2  require 'person.php';
3
4  $person = new Person('denny chandra','bandung');
5  $person->tampil();
6  ?>
```

← → ↻ ⓘ localhost/oop/test_person.php

nama = denny chandra,alamat = bandung

Contoh

```
1  <?php
2  require 'Person.php';
3
4  class Mahasiswa extends Person{
5
6      function __construct($nama,$alamat) {
7          parent::__construct($nama,$alamat);
8      }
9
10 }
```

```
1  <?php
2  require 'mahasiswa.php';
3
4  $mhs = new Mahasiswa('budi','bandung');
5  $mhs->tampil();
6  ?>
```

← → ↻ ⓘ localhost/oop/test_mahasiswa.php

nama = budi,alamat = bandung

Access Modifier

Modifier	Class	Subclass	World
Private	Y	N	N
Protected	Y	Y	N
Public	Y	Y	Y

Public:

Public method or variable can be accessible from anywhere. we can access them inside the class, outside the class and in child class also.

Private

Private: Method or property with private visibility can only be accessible inside the class. we can not access private method or variable outside the class.

Protected:

Method or variable with protected visibility can only be accessed within a class and in the derived class. We use Protected in the process of inheritance.

Access Modifier

```
1  <?php
2  class Person{
3      var $nama;
4      var $alamat;
5      protected $ktp;
6
7      function __construct($nama,$alamat) {
8          $this->nama = $nama;
9          $this->alamat = $alamat;
10     }
11
12     function set_ktp($ktp){
13         $this->ktp = $ktp;
14     }
15
16     function tampil(){
17         echo 'nama = ' . $this->nama . ',alamat = ' . $this->alamat;
18         echo '<br>';
19         echo ',ktp = ' . $this->ktp;
20     }
21 }
22 ?>
```

Access Modifier

person.php

test_person.php x

test_mahasiswa.php

```
1  <?php
2  require 'person.php';
3
4  $person = new Person('denny chandra','bandung');
5  $person->tampil();
6  echo '<hr>';
7  $person->set_ktp('123456');
8  $person->tampil();
9  ?>
```


Access Modifier

```
1  <?php
2  require 'Person.php';
3
4  class Mahasiswa extends Person{
5
6      private $nim;
7      public $prodi;
8
9      function __construct($nama,$alamat,$nim='') {
10         parent::__construct($nama,$alamat);
11         $this->nim = $nim;
12     }
13
14     function tampil(){
15         echo 'nama = ' . $this->nama . ',alamat = ' . $this->alamat . ',nim = ' . $this->nim;
16     }
17
18     function get_ktp(){
19         return $this->ktp;
20     }
21 }
22
23
24 ?>
```

Access Modifier

```
1  <?php
2  require 'mahasiswa.php';
3
4  $mhs = new Mahasiswa('budi','bandung','007');
5  $mhs->tampil();
6  $mhs->set_ktp('888');
7  $mhs->prodi = 'Teknik Elektro';
8  echo ',ktp = ' . $mhs->get_ktp();
9  echo ',prodi = ' . $mhs->prodi;
10 ?>
```



CodeIgniter Rocks

CodeIgniter is a powerful PHP framework with a very small footprint, built for developers who need a simple and elegant toolkit to create full-featured web applications.

**Star**

16,343

**Fork**

7,462

**Follow**

22.4K followers

**Download**The latest is **Version 3.1.9****Read the Manual**

Clear documentation

**View the Forums**

Get Support & Discuss Things

**On GitHub**

Fix Bugs or Add Features

CodeIgniter

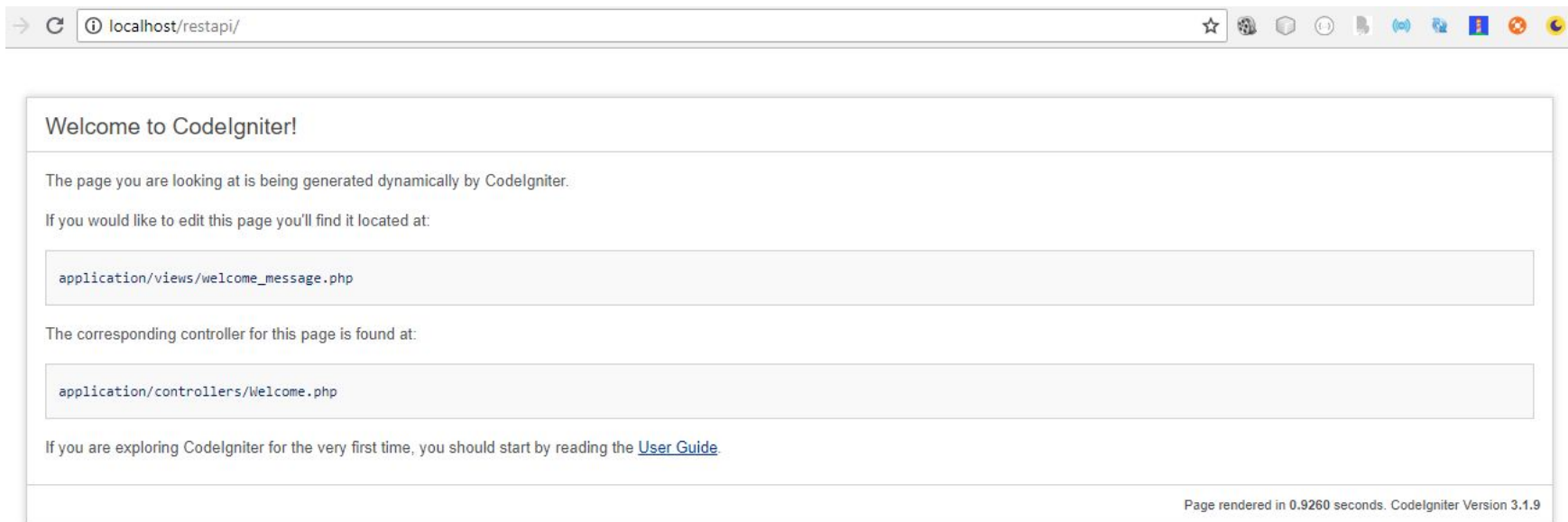
Codeigniter

- Codeigniter merupakan framework PHP yang menerapkan konsep MVC (Model View Controller).
- Waktu eksekusi cepat
- Ukuran kecil
- Konfigurasi minimal, tidak perlu install cukup dengan dicopy
- Dokumentasi yang lengkap, sehingga mempermudah pengguna
- Tersedia banyak library pendukung

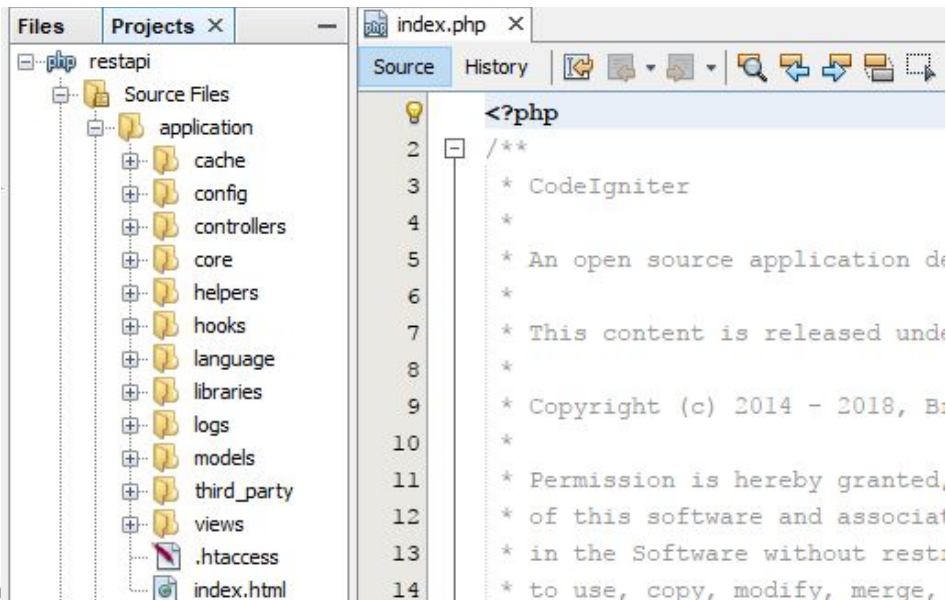
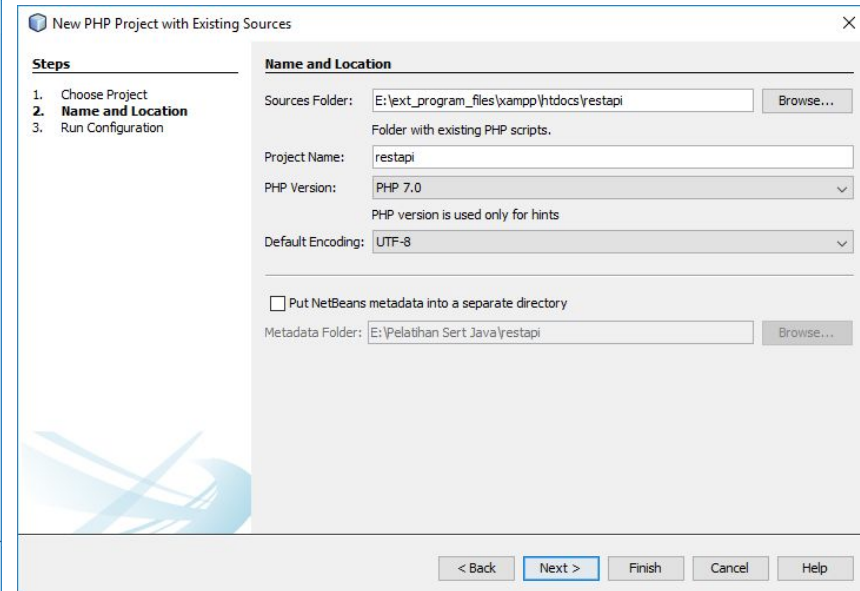
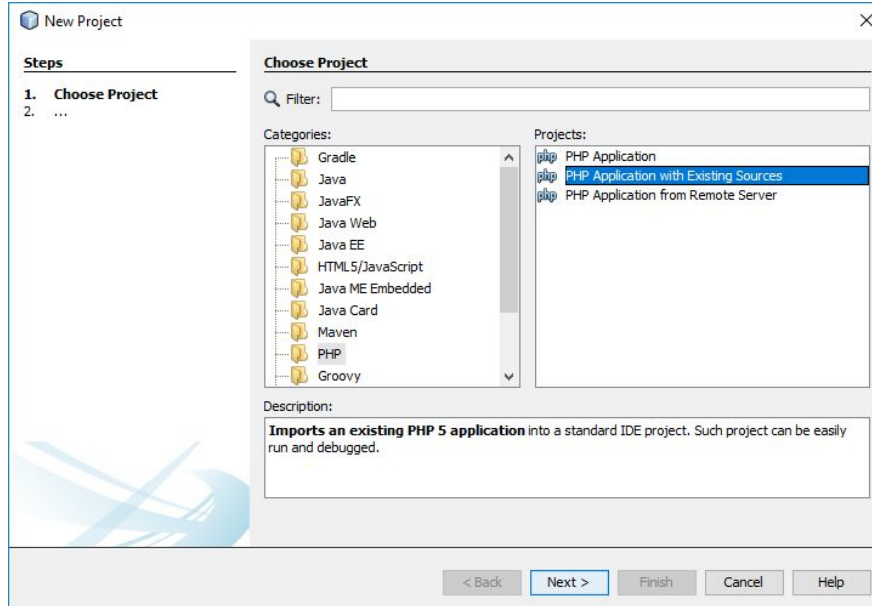


Instalasi

- Download <https://codeigniter.com> versi 3.19
- Unzip di htdocs di xampp, rename folder : restapi
- Akses di browser : <http://localhost/restapi/>



Netbeans Project



User Guide



The screenshot shows a web browser window with the address bar displaying 'localhost/restapi/user_guide/'. The page has a dark orange header with the CodeIgniter logo and a search bar. A dark sidebar on the left contains a list of navigation links. The main content area has a breadcrumb trail 'Docs » CodeIgniter User Guide' and a large heading 'CodeIgniter User Guide'. Below the heading is a list of links organized into two columns.

← → ↻ ⓘ localhost/restapi/user_guide/

 **CodeIgniter**

Search docs


















Welcome to CodeIgniter
Installation Instructions
CodeIgniter Overview
Tutorial
Contributing to CodeIgniter
General Topics
Libraries
Database Reference
Helpers

Docs » CodeIgniter User Guide

CodeIgniter User Guide

- [License Agreement](#)
- [Change Log](#)
- [Welcome](#)
- [Basic Info](#)
- [Installation](#)
- [Introduction](#)
- [Tutorial](#)
- [Contributing to CodeIgniter](#)
- [General Topics](#)
- [Library Reference](#)
- [Database Reference](#)
- [Helper Reference](#)

Struktur Direktori

- ▼  restapi
 - ▼  application
 -  cache
 -  config
 - >  controllers
 -  core
 -  helpers
 -  hooks
 - >  language
 -  libraries
 -  logs
 -  models
 -  third_party
 - >  views
 - >  system
 - >  tests
 - >  user_guide

- Application
 - Config
 - Autoload
 - Database
 - Controller
 - Libraries
 - Model



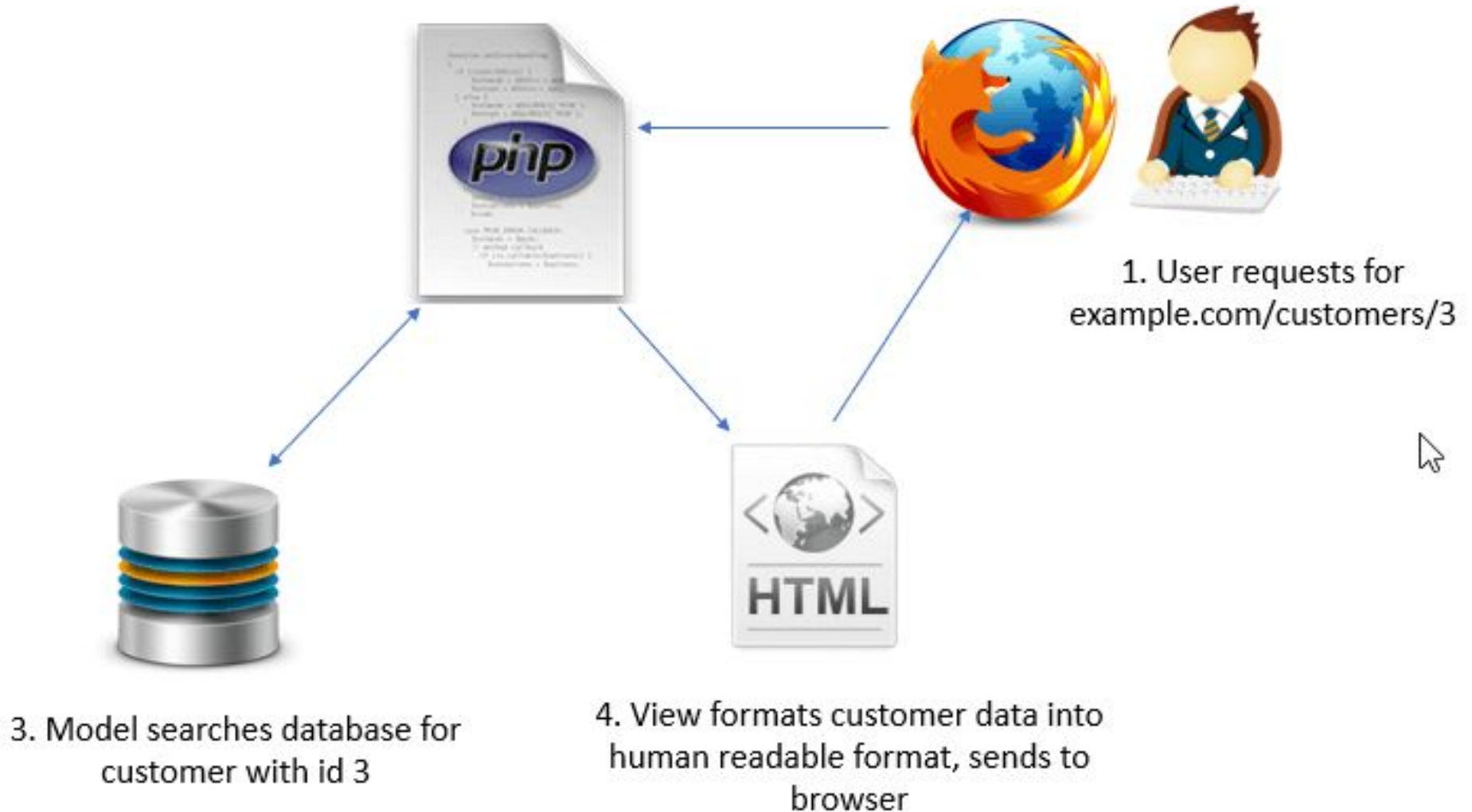
Model, View, Controller

- Model, merupakan folder untuk meletakkan semua file yang berhubungan dengan cara pengambilan data dari basis data
- View, merupakan folder untuk meletakkan semua file view, yaitu file yang berhubungan dengan tampilan aplikasi
- Controller, merupakan folder yang digunakan untuk meletakkan file-file controller.



Model, View, Controller

2. Controller requests for customer with id 3 from model



Codeigniter URL

URL in Codeigniter is Segment Based.

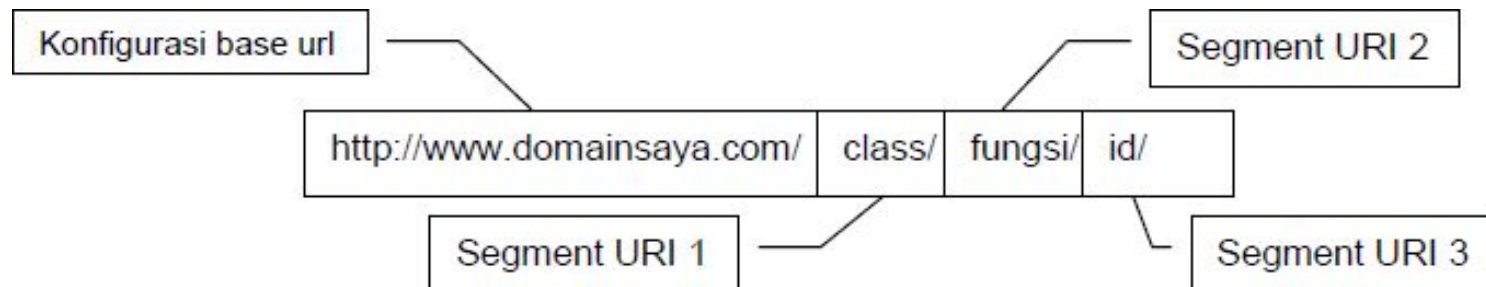
```
www.your-site.com/news/article/my_article
```

Segments in a URI

```
www.your-site.com/class/function/ID
```

Codeigniter Optionally Supports Query String URL

```
www.your-site.com/index.php?c=news&m=article&ID=345
```



Welcome controller

```
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class Welcome extends CI_Controller {
5
6      public function index()
7      {
8          $this->load->view('welcome_message');
9      }
10
11     public function name(){
12         //load class helper
13         //letakkan helper di autoload supaya tidak perlu memanggil berulang-ulang
14         $this->load->helper('url');
15         //fungsi php native gunakan spt biasa
16         echo "hai " . urldecode($this->uri->segment(3));
17     }
18 }
```

← → ↻ ⓘ localhost/ci/index.php/welcome/name/budi

hai budi

Config/autoload.php

Welcome.php

autoload.php x

```
85 | -----
86 |   Auto-load Helper Files
87 | -----
88 | Prototype:
89 |
90 |   $autoload['helper'] = array('url', 'file');
91 | */
92 | $autoload['helper'] = array('url');

11 |     public function name(){
12 |         echo "hai " . urldecode($this->uri->segment(3));
13 |     }
14 | }
```

Menghilangkan /index.php/

```
1 Options +FollowSymLinks
2 IndexIgnore */*
3 RewriteEngine on
4 # if a directory or a file exists, use it directly
5 RewriteCond %{REQUEST_FILENAME} !-f
6 RewriteCond %{REQUEST_FILENAME} !-d
7 # otherwise forward it to index.php
8 #RewriteRule . index.php/$1 [L,QSA]
9 RewriteRule ^(.*)$ index.php/$1 [L]
```

ext_program_files > xampp > htdocs > ci >

Name

- application
- system
- tests
- user_guide
- .editorconfig
- .htaccess
- composer
- contributing.md
- index.php
- license
- readme.rst

← → ↻ ⓘ localhost/ci/welcome/name/budi

hai budi

JSON (API Service)

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999.

JSON

<https://www.json.org/json-id.html>



JSON

```
[0] => Array (  
  [id] => 1  
  [isbn] => 111222333  
  [judul] => Dasar Pemrograman  
  [pengarang] => A Jogianto  
)  
[1] => Array (  
  [id] => 2  
  [isbn] => 223344  
  [judul] => Dasar Jaringan  
  [pengarang] => A Sunarto  
)
```

```
[  
  {  
    "id": "1",  
    "isbn": "111222333",  
    "judul": "Dasar Pemrograman",  
    "pengarang": "A Jogianto"  
  },  
  {  
    "id": "2",  
    "isbn": "223344",  
    "judul": "Dasar Jaringan",  
    "pengarang": "A Sunarto"  
  }  
]
```



RESTful API

DELETE POST PUT GET

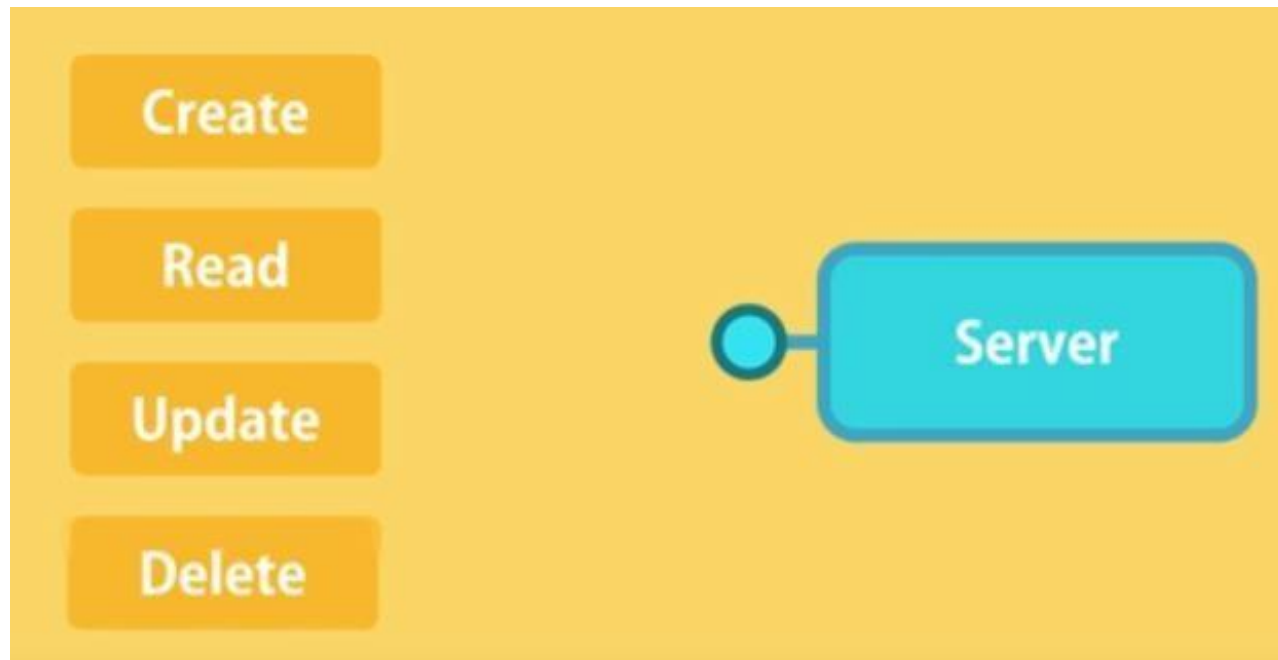
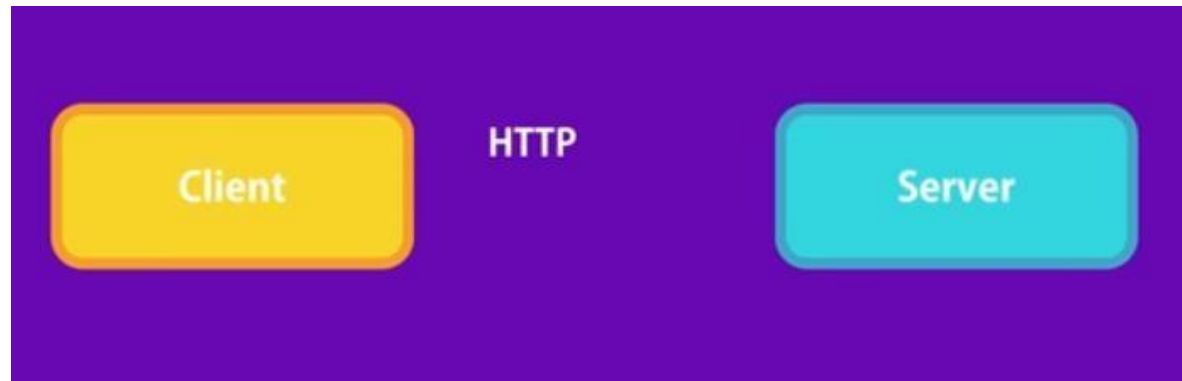
REST

REST

- (REpresentational State Transfer) : *Transferring, accessing, and manipulating textual data representations in a stateless manner.*
- Menggunakan HTTP URIs untuk mengidentifikasi resources
 - Dapat menggunakan path segment
 - Dapat menggunakan string query parameter
- Menggunakan HTTP verbs (methods)
- Menggunakan MIME types
- Menggunakan HTTP status/error codes



Client-Server



- HTTP

Verb	Objective	Usage	Multiple requests	Cache/Bookmark
GET	Retrieve items from resource	links	yes	yes
POST	Create new item in resource	forms	no	no
PUT	Replace existing item in resource	forms	yes	no
PATCH	Update existing item in resource	forms	no/yes	no
DELETE	Delete existing item in resource	forms/ links	yes	no

MIME Types

Extension	Kind of Document	MIME Type
.txt	Textual file, human-readable and must not contain binary data	text/plain
.json	JSON format	application/json
.json	JSON format structured using Hypertext Application Language (HAL)	application/hal+json
.xml	XML	application/xml
.htm .html	HyperText Markup Language (HTML)	text/html
.jpeg .jpg	JPEG images	image/jpeg
.gif	Graphics Interchange Format (GIF)	image/gif
.png	Portable Network Graphics	image/png
.pdf	Adobe Portable Document Format (PDF)	application/pdf
	Form data sent as a long query string	application/x-www-form-urlencoded
	Form data used for uploading files to the server	multipart/form-data

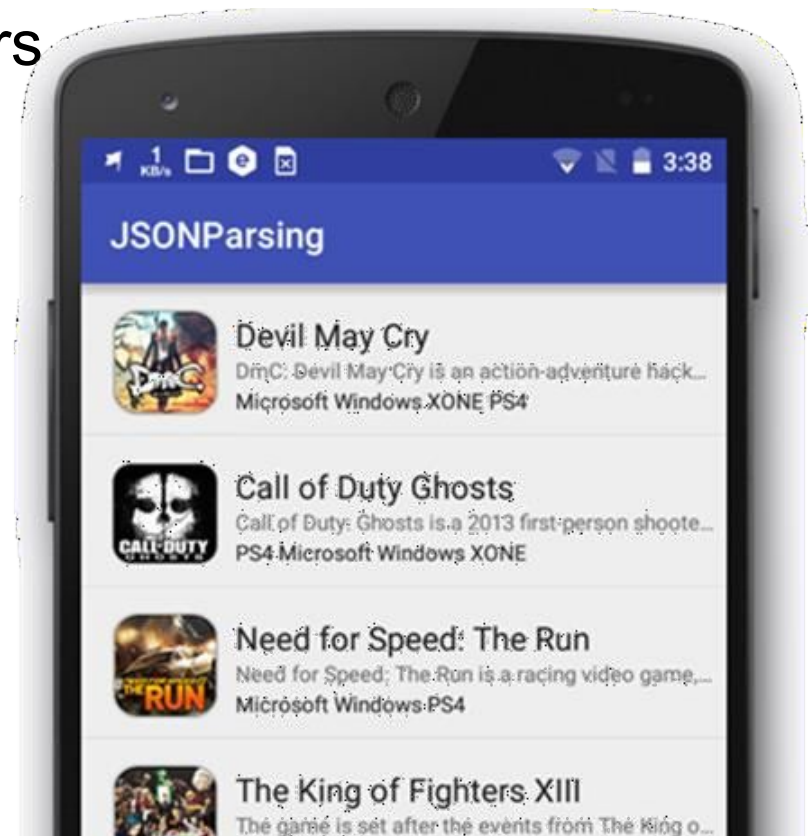
Common HTTP Status Codes

Code	Definition	Returned by
200	OK – The request succeeded.	GET, DELETE, PATCH, PUT
201	Created – A new resource or object in a collection.	POST
304	Not modified – Nothing was modified by the request.	PATCH, PUT
400	Bad request – The request could not be performed by the server due to bad syntax or other reason in request.	All
401	Unauthorized – Authorization credentials are required or user does not have access to the resource/method they are requesting.	All
404	Resource not found – The URI is not recognized by the server.	All
500	Server error – Generic something went wrong on the server side.	All

JSON (JavaScript Object Notation)

- human- and machine- readable and writable
- natively supported by browsers

```
{
  "id": "1",
  "title": "Devil May Cry",
  "desc": "DMC: Devil May Cry is an action-adventure hack and slash video game developed by Ninja Theory and published by Capcom for Playstation 3, Xbox 360 and Microsoft Windows. It is the fifth instalment of the Devil May Cry series, serving as a reboot. Announced in late 2010 during the Tokyo Game Show, the game is set in an alternate reality in the Devil May Cry series",
  "platforms": "Microsoft Windows XONE PS4",
  "image": "devilmaycry.png"
},
{
  "id": "2",
  "title": "Call of Duty Ghosts",
  "desc": "Call of Duty: Ghosts is a 2013 first-person shooter video game developed by Infinity Ward, with assistance from Raven Software, Neversoft and Certain Affinity. Published by Activision, it is the tenth primary installment in the Call of Duty series and the sixth developed by Infinity Ward.",
  "platforms": "PS4 Microsoft Windows XONE",
  "image": "ghost.png"
}
```



PHP dan JSON

```
$info = [  
    'version' => '0.1-dev',  
    'app' => 'rest api'  
];
```

```
{  
    "version": "0.1-dev",  
    "app": "rest api"  
}
```

```
$info = new stdClass();  
$info->version = '0.1-dev';  
$info->app = 'rest api';
```

```
{  
    "version": "0.1-dev",  
    "app": "rest api"  
}
```

```
$info = [  
    'version' => '0.1-dev',  
    'app' => [  
        'name' => 'Rest API',  
        'creator' => 'Me'  
    ]  
];
```

```
{  
    "version": "0.1-dev",  
    "app": {  
        "name": "Rest API",  
        "creator": "Me"  
    }  
}
```

PHP dan JSON

```
$info = [  
    'version' => '0.1-dev',  
    'app' => [  
        'name' => 'Rest API',  
        'creator' => [  
            ['name' => 'denny', 'id' => '123456'],  
            ['name' => 'budi', 'id' => '123457']  
        ]  
    ]  
];
```

```
{  
    "version": "0.1-dev",  
    "app": {  
        "name": "Rest API",  
        "creator": [  
            {  
                "name": "denny",  
                "id": "123456"  
            },  
            {  
                "name": "budi",  
                "id": "123457"  
            }  
        ]  
    }  
}
```

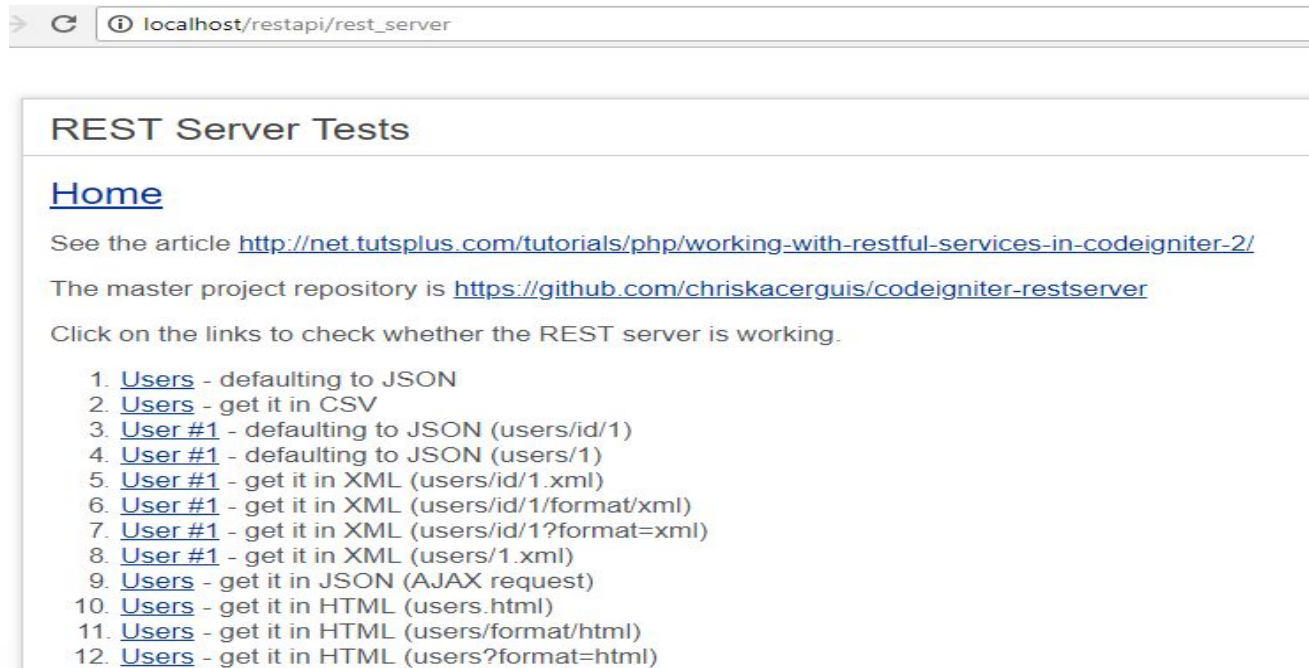


RESTful API
DELETE POST PUT GET

REST
menggunakan
Codeigniter

Library Codeigniter-rest server

- <https://github.com/chriskacerguis/codeigniter-restserver>
- Versi 3.0.0 (sebelum update juni)
- Instalasi:
 - Copy folder application (overwrite)
- Run

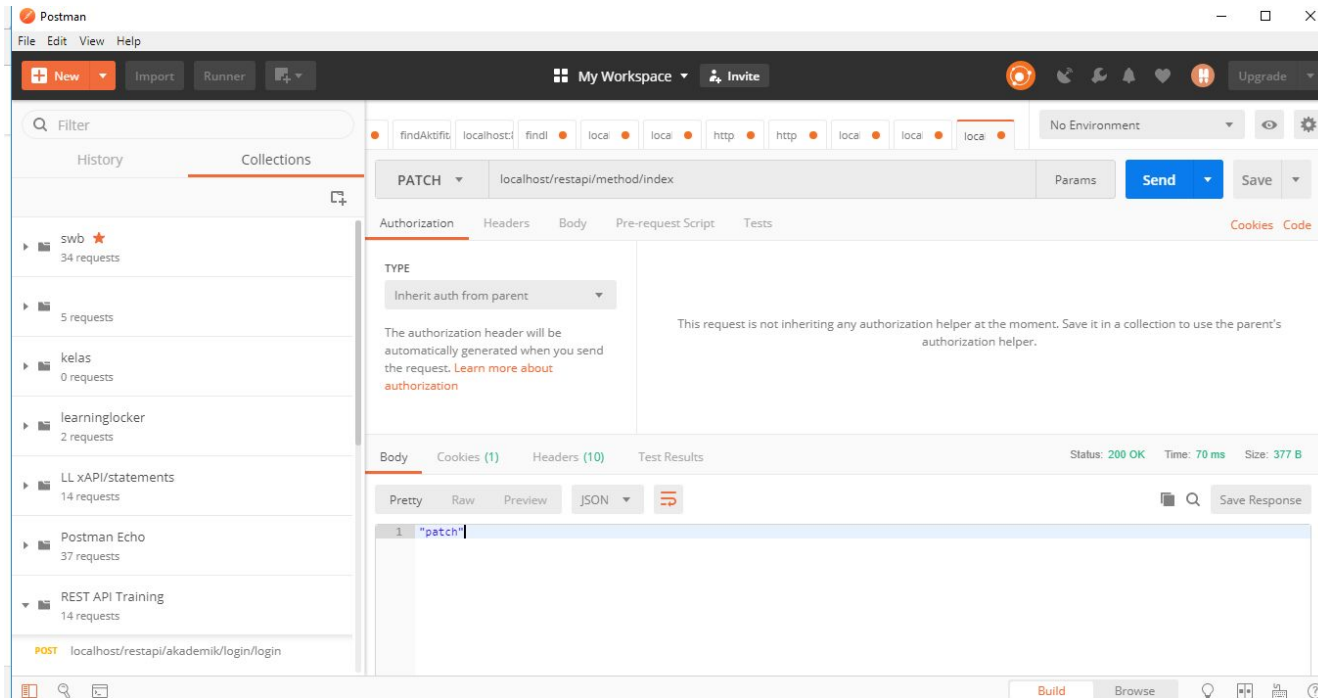


Membuat Rest Controller

```
1  <?php
2
3  require APPPATH . '/libraries/REST_Controller.php';
4
5  class Api extends REST_Controller {
6
7      function __construct($config = 'rest') {
8          parent::__construct($config);
9      }
10
11     public function index_get(){
12         $info = [
13             'version' => '0.1-dev',
14             'app' => 'rest api'
15         ];
16         $this->response($info, REST_Controller::HTTP_OK);
17     }
18 }
```

Postman

- Tools untuk menguji REST yang dibuat sebelum digunakan di aplikasi yang sebenarnya
- Download di : <https://www.getpostman.com>



Menggunakan Postman

```
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3  require APPPATH . '/libraries/REST_Controller.php';
4  class Method extends REST_Controller {
5
6      function __construct($config = 'rest') {
7          parent::__construct($config);
8      }
9
10     function index_get(){
11         $this->response('get');
12     }
13
14     function index_post(){
15         $this->response('post');
16     }
17
18     function index_put(){
19         $this->response('put');
20     }
21
22     function index_patch(){
23         $this->response('patch');
24     }
25
26     function index_delete(){
27         $this->response('delete');
28     }
29 }
```

GET

```
function index_get(){  
    $nama = $this->get('nama');  
    $this->response($nama);  
}
```

GET ▼

localhost/restapi/method/index?nama=Agus

GET ▼

localhost/restapi/method/index/nama/Agus



POST

```
function index_post(){  
    $nama = $this->post('nama');  
    $this->response($nama);  
}
```

POST ▾

localhost/restapi/method/index

Params

Send ▾

Save ▾

AuthorizationHeadersBody ●Pre-request ScriptTests

form-data●x-www-form-urlencoded●raw●binary

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	nama	Agus			
	Key	Value	Description		

BodyCookies (1)Headers (10)Test Results

Status: 200 OKTime: 192 msSize: 377 B

PrettyRawPreviewJSON ▾

Save Response

1"Agus"



PUT

```
function index_put(){  
    $nama = $this->put('nama');  
    $this->response($nama);  
}
```

PUT

localhost/restapi/method/index

Params

Send

Save

Authorization

Headers (1)

Body

Pre-request Script

Tests

Cookies

Code

☐ form-data

☒ x-www-form-urlencoded

☐ raw

☐ binary

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	nama	agus			
	Key	Value	Description		

Body

Cookies (1)

Headers (10)

Test Results

Status: 200 OK

Time: 84 ms

Size: 377 B

Pretty

Raw

Preview

JSON

Save Response

1

"agus"



DELETE

```
function index_delete(){  
    $nama = $this->delete('nama');  
    $this->response('delete');  
}
```

DELETE

localhost/restapi/method/index

Params

Send

Save

Authorization

Headers (1)

Body

Pre-request Script

Tests

Cookies

Code

☐ form-data

☒ x-www-form-urlencoded

☐ raw

☐ binary

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	nama	agus			
	Key	Value	Description		

Body

Cookies (1)

Headers (11)

Test Results

Status: 200 OK

Time: 89 ms

Size: 507 B

Pretty

Raw

Preview

JSON

Save Response

1

"delete"



Membuat API Dari Database Perpustakaan

- Setting Database
- DDL dan DML
- Membuat Model
- Membuat Controller
- Testing API melalui Postman
- Dokumentasi API



Setting Database


```
$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => '',
    'database' => 'perpustakaan',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);
```

Setting library (autoload)



DDL Tabel

```
CREATE TABLE `pengarang` (  
    `id` int(11) NOT NULL AUTO_INCREMENT,  
    `nama` varchar(50) DEFAULT NULL,  
    `email` varchar(25) DEFAULT NULL,  
    PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
CREATE TABLE `buku` (  
    `id` int(11) NOT NULL AUTO_INCREMENT,  
    `isbn` varchar(15) DEFAULT NULL,  
    `judul` varchar(50) DEFAULT NULL,  
    `idpengarang` int(50) DEFAULT NULL, PRIMARY  
    KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



DML Tabel (untuk sample via SQLYOG)

lokal

Filter tables in perpustakaan

Filter (Ctrl+Shift+B)

root@localhost

- akademik
- buku
- information_schema
- mysql
- performance_schema
- perpustakaan**
 - Tables
 - buku
 - pengarang
 - Views
 - Stored Procs
 - Functions
 - Triggers
 - Events

Query x pengarang buku +

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List All Tags. [Ctrl+Enter]->List Matching Tags. [Ctrl+Shift]

```
1 INSERT INTO `pengarang` (nama,email) VALUES ('Denny Chandra','denny@gmail.com');
2 INSERT INTO `pengarang` (nama,email) VALUES ('Wendi Gunawan','wendi@gmail.com');
```

1 Messages 2 Table Data 3 Info

	id	nama	email
	1	Denny Chandra	denny@gmail.com
	2	Wendi Gunawan	wendi@gmail.com
*	(Auto)	(NULL)	(NULL)

Membuat Model

```
<?php
2  /**
3   * Model untuk pengarang
4   *
5   * @author lenovo
6   */
7  class Pengarang_model extends CI_Model {
8      //put your code here
9      public $table = 'pengarang';
10
11     public function __construct() {
12         parent::__construct();
13     }
14
15     /**
16     *
17     * @param array $data
18     * @return boolean
19     */
20     public function insert($data){
21         return $this->db->insert($this->table, $data);
22     }
23
24     public function update($id, $data){
25         $this->db->where('id', $id);
26         return $this->db->update($this->table, $data);
27     }
```

Membuat Model

```
28
29 public function findAll() {
30     return $this->db->get($this->table)->result_array();
31 }
32
33 public function findById($id) {
34     return $this->db->get_where($this->table, ['id' => $id])->result_array();
35 }
36
37 public function delete($id) {
38     $this->db->where('id', $id);
39     return $this->db->delete($this->table);
40 }
41 }
```

Membuat Controller

```
1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5  require APPPATH . 'libraries/REST_Controller.php';
6
7  /**
8   * controller untuk pengarang
9   *
10  * @author lenovo
11  */
12  class Pengarang extends REST_Controller {
13
14  @ function __construct() {
15      parent::__construct();
16      $this->load->model('pengarang_model', 'pengarang');
17  }
18
19  function insert_post() {
20      $data = [
21          'nama' => $this->post('nama'),
22          'email' => $this->post('email')
23      ];
24      $res = $this->pengarang->insert($data);
25      if ($res) {
26          $this->response('data berhasil disimpan', REST_Controller::HTTP_CREATED);
27      }
28  }
```

Membuat Controller

```
27     } else {
28         $this->response('gagal menyimpan data', REST_Controller::HTTP_INTERNAL_SERVER_ERROR);
29     }
30 }
31
32 function update_post() {
33     $id = $this->post('id');
34     $data = [
35         'nama' => $this->post('nama'),
36         'email' => $this->post('email')
37     ];
38     $res = $this->pengarang->update($id, $data);
39     if ($res) {
40         $this->response('data berhasil diubah', REST_Controller::HTTP_OK);
41     } else {
42         $this->response('gagal mengubah data', REST_Controller::HTTP_INTERNAL_SERVER_ERROR);
43     }
44 }
45
46 function delete_post() {
47     $id = $this->post('id');
48     $res = $this->pengarang->delete($id);
```

Membuat Controller

```
49     if ($res) {
50         $this->response('data berhasil dihapus', REST_Controller::HTTP_OK);
51     } else {
52         $this->response('gagal menghapus data', REST_Controller::HTTP_INTERNAL_SERVER_ERROR);
53     }
54 }
55
56 function all_get() {
57     $res = $this->pengarang->findAll();
58     if ($res) {
59         $this->response($res);
60     } else {
61         $this->response('gagal mengambil data', REST_Controller::HTTP_INTERNAL_SERVER_ERROR);
62     }
63 }
64
65 function byid_get() {
66     $id = $this->get('id');
67     $res = $this->pengarang->findById($id);
68     if ($res) {
69         $this->response($res[0]);
70     } else {
71         $this->response('gagal mengambil data', REST_Controller::HTTP_INTERNAL_SERVER_ERROR);
72     }
73 }
74 }
```

Pengujian method all, semua data

localhost/resta localhost/resta localhost/resta localhost/resta localhost/resta + ... No Environment

GET localhost/restapi/pengarang/all Params Send

Authorization Headers Body Pre-request Script Tests

TYPE

Inherit auth from parent

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the authorization helper.

Body Cookies Headers (7) Test Results Status: 200 OK Time: 311 ms

Pretty Raw Preview JSON

```
1 [
2   {
3     "id": "1",
4     "nama": "Denny Chandra",
5     "email": "denny@gmail.com"
6   },
7   {
8     "id": "2",
9     "nama": "Wendi Gunawan",
10    "email": "wendi@gmail.com"
11  }
```

Pengujian get by id

localhost/resta

localhost/resta

localhost/resta

localhost/resta

localhost/resta

+

...

No Environment

GET

localhost/restapi/pengarang/byid?id=1

Params

Send

	KEY	VALUE	DESCRIPTION	...
<input checked="" type="checkbox"/>	id	1		
	Key	Value	Description	

Authorization

Headers

Body

Pre-request Script

Tests

TYPE

Inherit auth from parent

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the authorization helper.

Body

Cookies

Headers (7)

Test Results

Status: 200 OK

Time: 446 ms

Pretty

Raw

Preview

JSON

1

{

2

"id": "1",

3

"nama": "Denny Chandra",

4

"email": "denny@gmail.com"

5

}



Pengujian insert

POST

localhost/restapi/pengarang/insert

Params

Send

Authorization

Headers

Body

Pre-request Script

Tests

☒ form-data

☐ x-www-form-urlencoded

☐ raw

☐ binary

	KEY	VALUE	DESCRIPTION	..
<input checked="" type="checkbox"/>	nama	Agus Haryono		
<input checked="" type="checkbox"/>	email	agus@gmail.com		
	Key	Value	Description	

Body

Cookies

Headers (7)

Test Results

Status: 201 Created Time: 368 ms

Pretty

Raw

Preview

JSON

1 "data berhasil disimpan"



Pengujian update

POST ▾

localhost/restapi/pengarang/update

Params

Send ▾

AuthorizationHeadersBody ●Pre-request ScriptTests

☒ form-data

☐ x-www-form-urlencoded

☐ raw

☐ binary

	KEY	VALUE	DESCRIPTION	⋮
<input checked="" type="checkbox"/>	id	1		
<input checked="" type="checkbox"/>	nama	Denny Chandra		
<input checked="" type="checkbox"/>	email	chandra@gmail.com		
	Key	Value	Description	

BodyCookiesHeaders (7)Test Results

Status: 200 OKTime: 394 ms

PrettyRawPreview

JSON ▾

1

"data berhasil diubah"

Pengujian delete

POST ▾

localhost/restapi/pengarang/delete

Params

Send ▾

AuthorizationHeadersBody ●Pre-request ScriptTests

☒ form-data

☐ x-www-form-urlencoded

☐ raw

☐ binary

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	id	3	
	Key	Value	Description

BodyCookiesHeaders (7)Test Results

Status: 200 OKTime: 478 ms

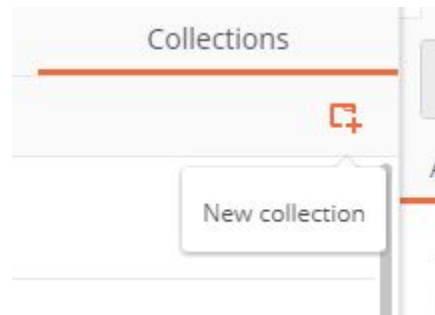
PrettyRawPreview

JSON ▾

1"data berhasil dihapus"



Mendokumentasikan API



CREATE A NEW COLLECTION

Name

API Aplikasi Perpustakaan

Description

Authorization

Pre-request Scripts

Tests

Variables

This description will show in your collection's documentation, along with the descriptions of its folders and requests.

API CRUD Pengarang dan Buku

Descriptions support Markdown

Cancel

Create

Simpan setiap pengujian

SAVE REQUEST

localhost/restapi/pengarang/all

Request description (Optional)

Adding a description makes your docs better

Descriptions support Markdown

Select a collection or folder to save to:

Q perp

Search Results

+ Create Collection "perp"

API Aplikasi Perpustakaan
API Aplikasi Perpustakaan

Cancel

Save

API Aplikasi Perpustakaan	
5 requests	
GET	localhost/restapi/pengarang/all
GET	localhost/restapi/pengarang/byid?id=1
POST	localhost/restapi/pengarang/insert
POST	localhost/restapi/pengarang/update
POST	localhost/restapi/pengarang/delete

Membuat dokumentasi

1. Select requests to document

2. Configure documentation

3. Next steps

Create a new API Use collection from this workspace

Enter the requests you want to document. Add headers and sample responses to these requests by clicking on the (***) icon.

learninglocker
2 requests

VidyanusaAPITMDG11
3 requests

gpsid
5 requests

LL xAPI/statements
14 requests

swb
34 requests

Postman Echo
37 requests

kelas
0 requests

REST API Training
30 requests

akad
6 requests

API Aplikasi Perpustakaan
5 requests

BackNext

✓ Select requests to document

2. Configure documentation

3. Next steps

Name

API Aplikasi Perpustakaan

Enter a title to describe your requests. This will help you identify your documentation and API collection in Postman.

Edit the description

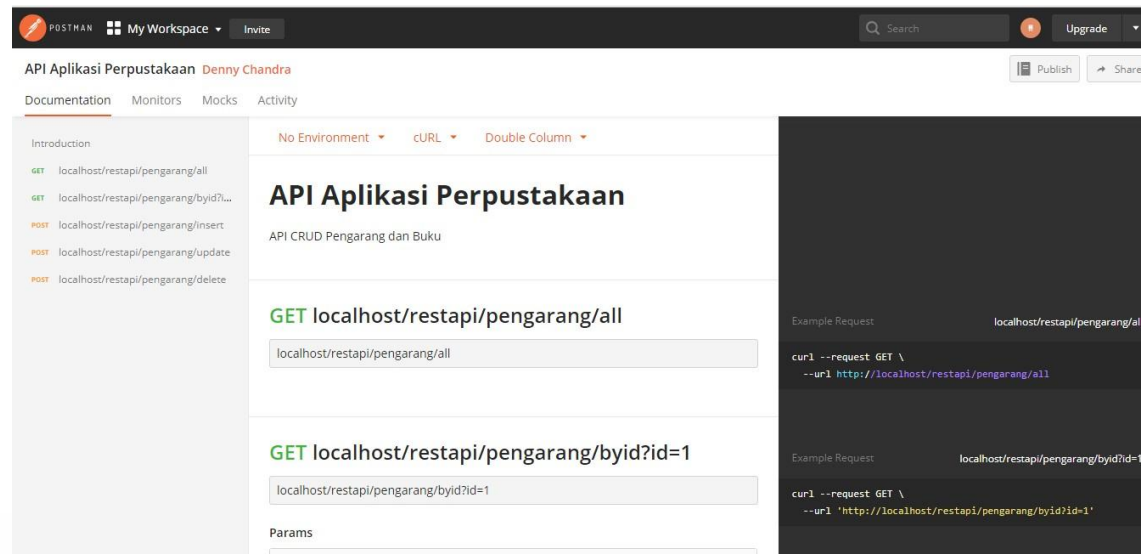
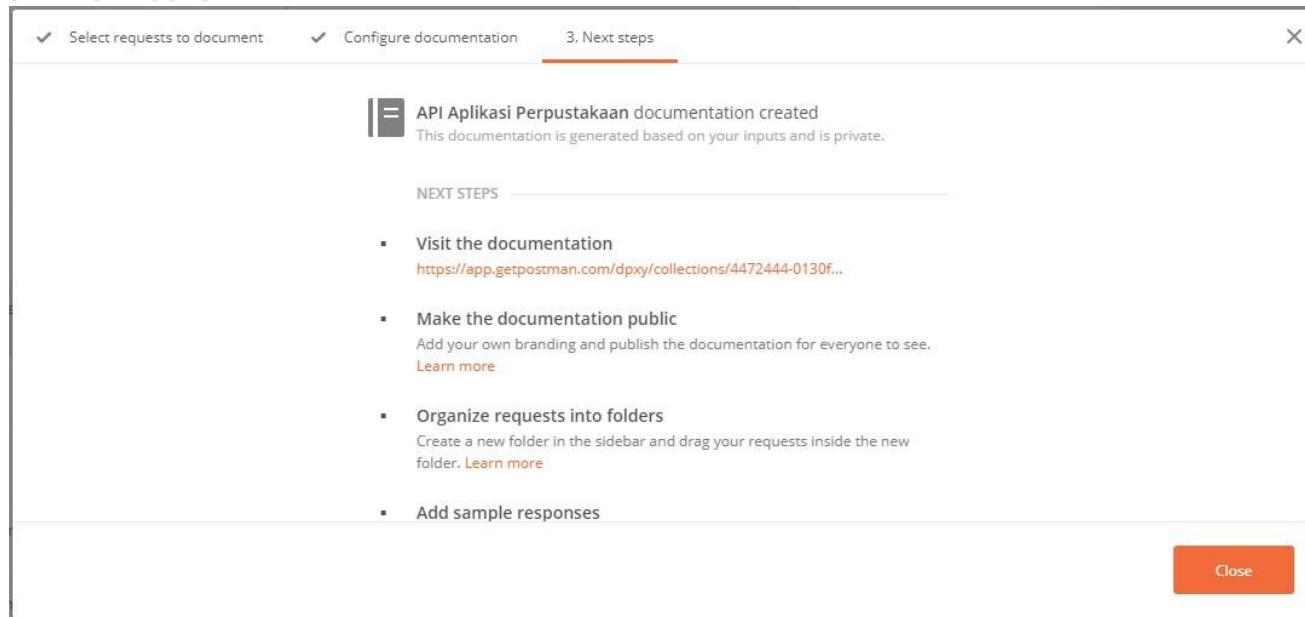
API CRUD Pengarang dan Buku

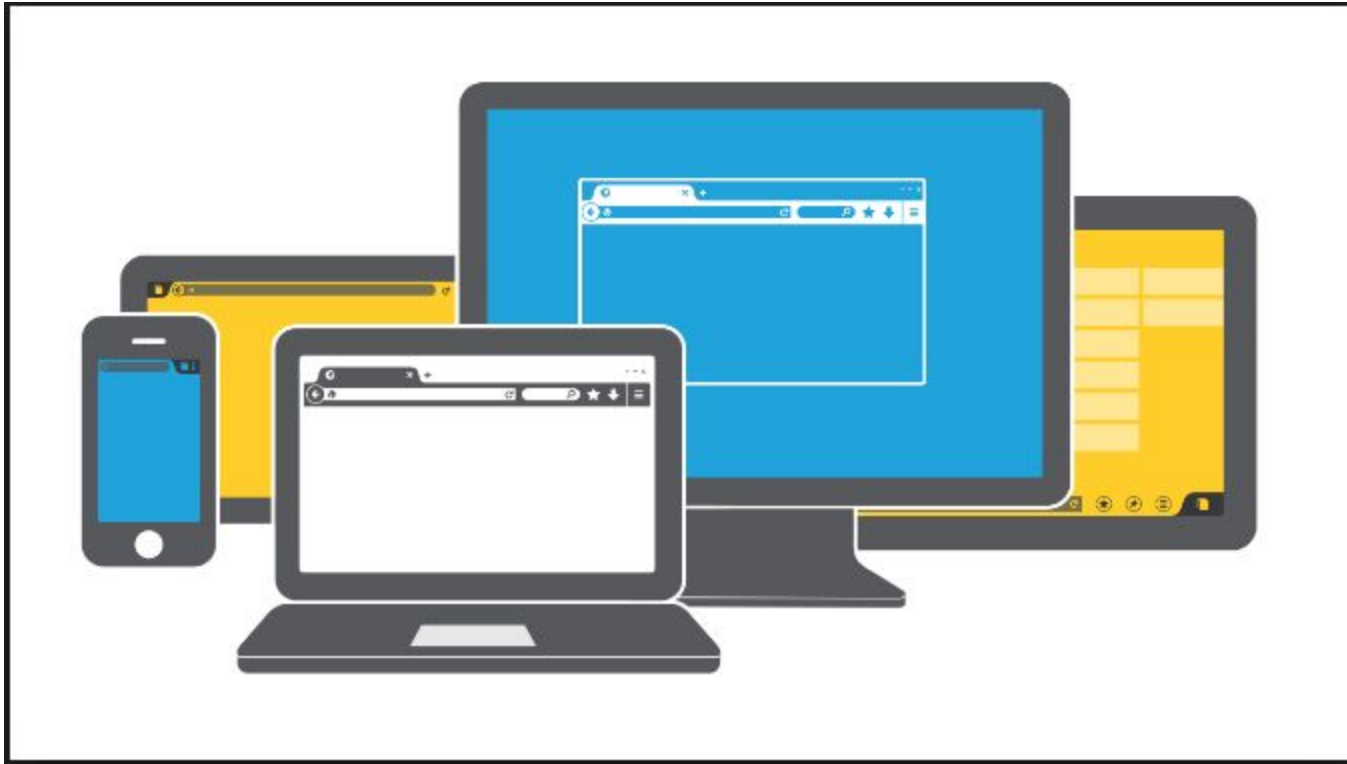
Add a general description for your requests, eg. overview and authentication details.

You can use **markdown** for adding headings, lists, code snippets etc. in your description.

BackCreate

Membuat dokumentasi





Akses REST melalui User Interface

Cross Origin Resources Sharing



Setting CORS

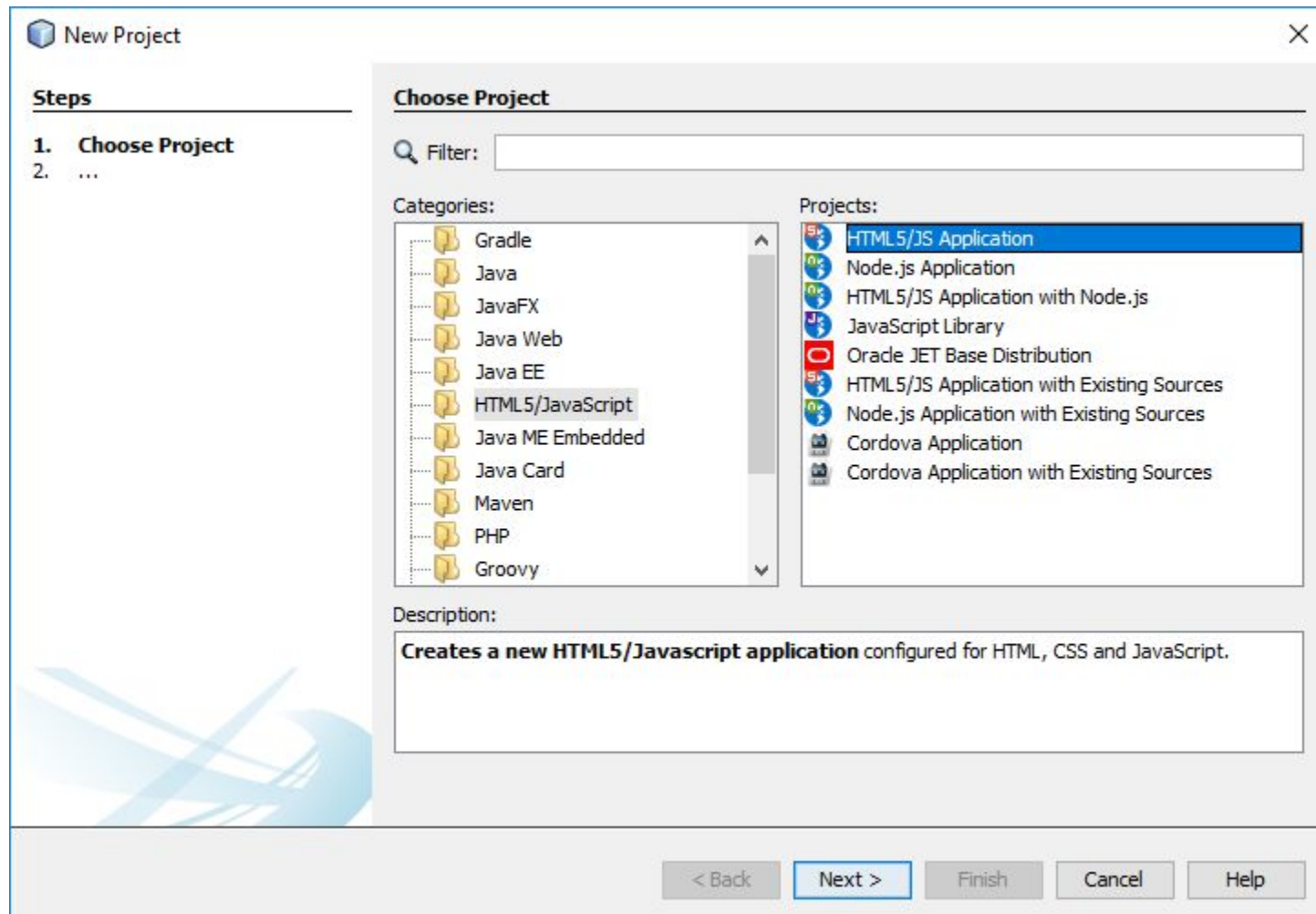
`access-control-allow-methods` → POST,GET

`access-control-allow-origin` → *

`connection` → Keep-Alive

```
class Pengarang extends REST_Controller {  
  
    function __construct() {  
        header('Access-Control-Allow-Origin: *');  
        header("Access-Control-Allow-Methods: POST,GET");  
        parent::__construct();  
        $this->load->model('pengarang_model', 'pengarang');  
    }  
}
```

Membuat project untuk front end



Membuat project untuk front end

New HTML5/JS Application

Steps

1. Choose Project
- 2. Name and Location**
3. Site Template
4. Tools

Name and Location

Project Name:

Project Location:

Project Folder:

< Back **Next >** Finish Cancel Help

Membuat project untuk front end

New HTML5/JS Application

Steps

1. Choose Project

2. Name and Location

3. **Site Template**

4. Tools

Site Template

Optionally initialize your project structure with a site template.

☒ No Site Template

☐ Select Template:

Template:

Browse...

Local archive file or a URL pointing to remote archive file can be specified.

☐ Download Online Template:

AngularJS Seed

Initializr: Bootstrap

Initializr: Classic

Initializr: Responsive

HTML5 Boilerplate 5.3.0

Twitter Bootstrap 3.3.6

Mobile Boilerplate 4.1.2

Grunt Hello World

Description:

[Clear local cache](#)

Site template from html5boilerplate.com.

< Back

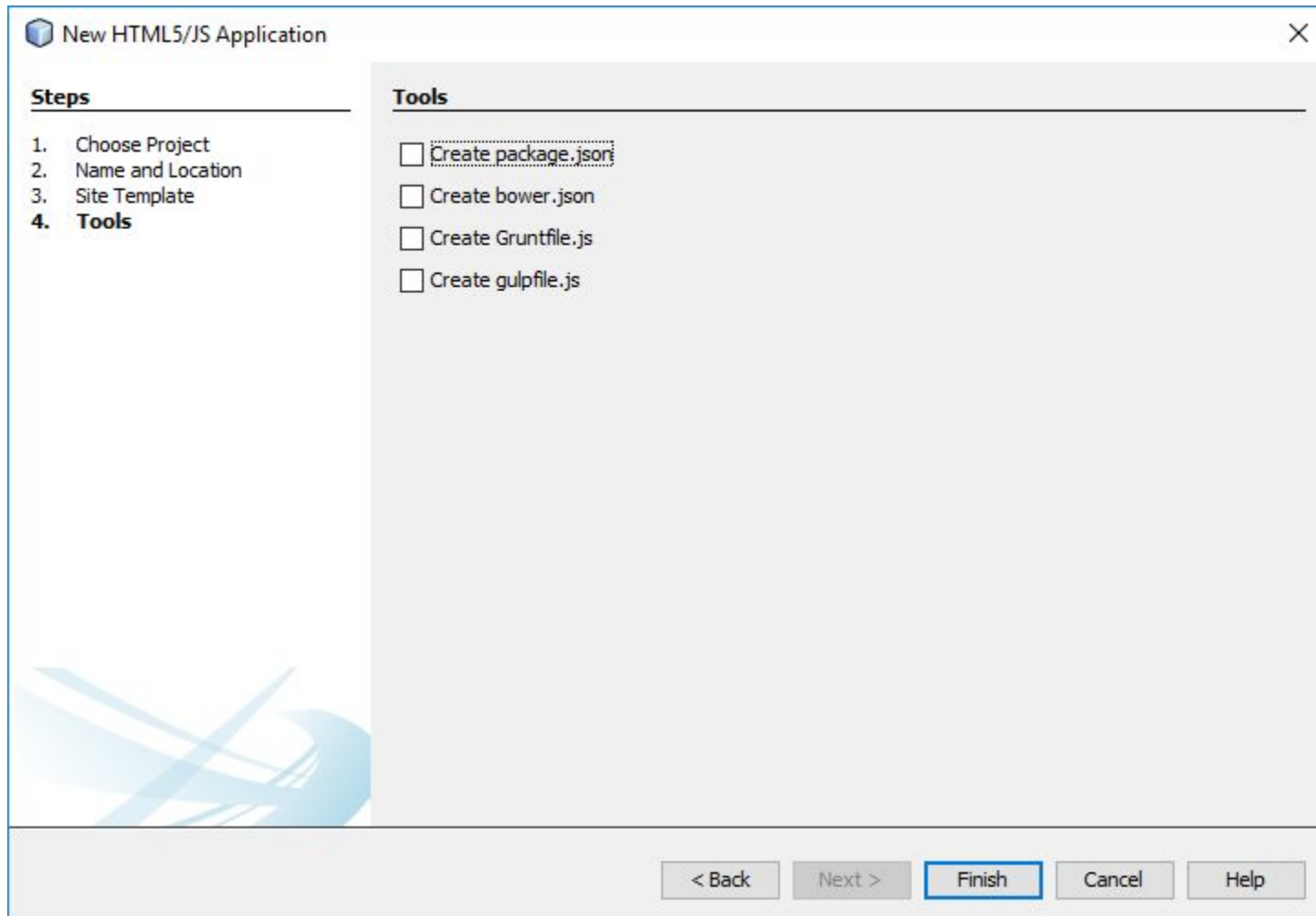
Next >

Finish

Cancel

Help

Membuat project untuk front end



The image shows a Windows-style dialog box titled "New HTML5/JS Application". It has a close button (X) in the top right corner. The dialog is divided into two main sections: "Steps" on the left and "Tools" on the right. The "Steps" section lists four steps: 1. Choose Project, 2. Name and Location, 3. Site Template, and 4. Tools (which is currently selected and bolded). The "Tools" section contains four checkboxes, each followed by a label: "Create package.json" (checked), "Create bower.json", "Create Gruntfile.js", and "Create gulpfile.js". At the bottom of the dialog, there are five buttons: "< Back", "Next >", "Finish" (highlighted with a blue border), "Cancel", and "Help".

New HTML5/JS Application

Steps

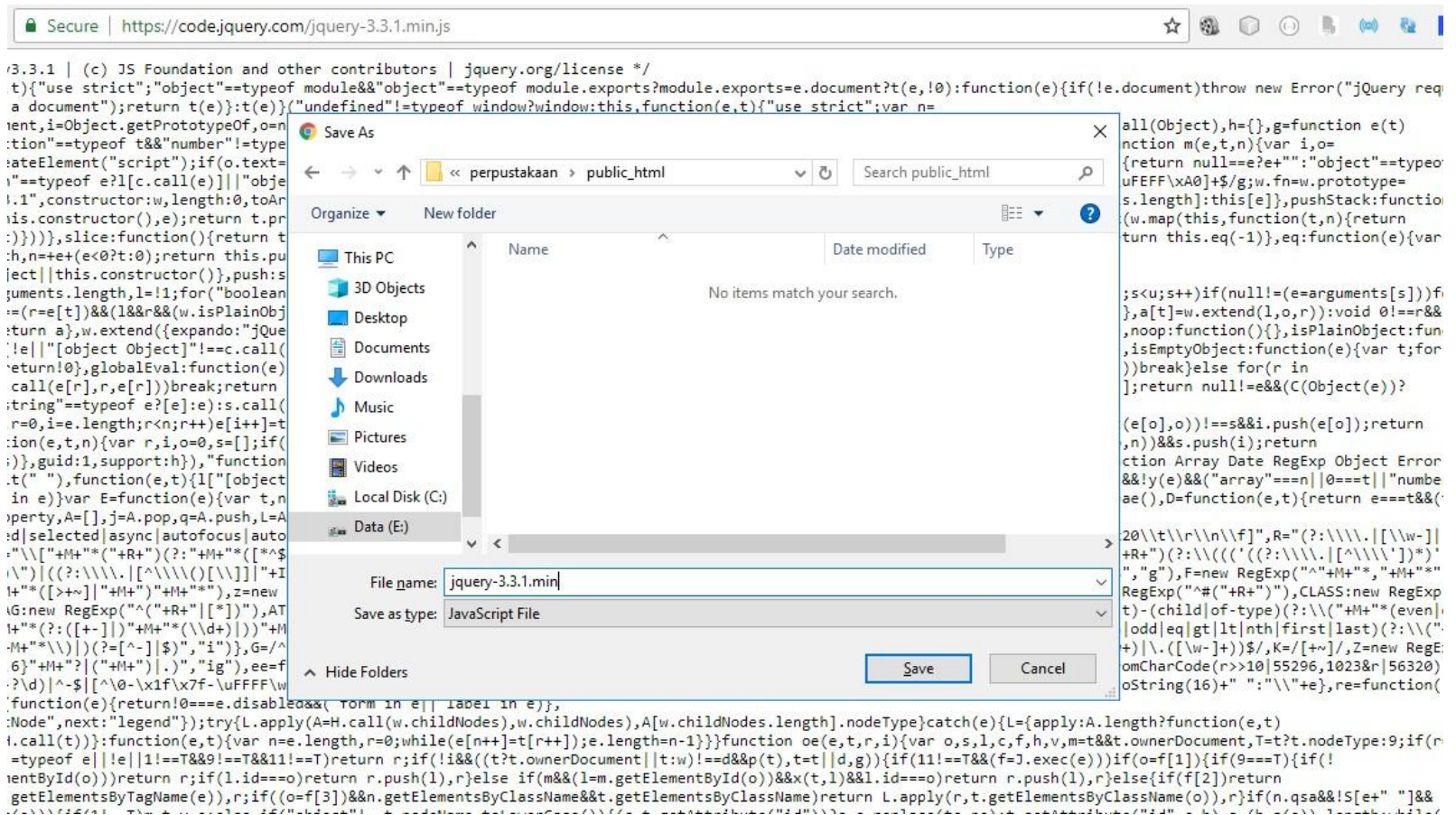
1. Choose Project
2. Name and Location
3. Site Template
- 4. Tools**

Tools

- ☒ Create package.json
- ☐ Create bower.json
- ☐ Create Gruntfile.js
- ☐ Create gulpfile.js

< Back Next > **Finish** Cancel Help

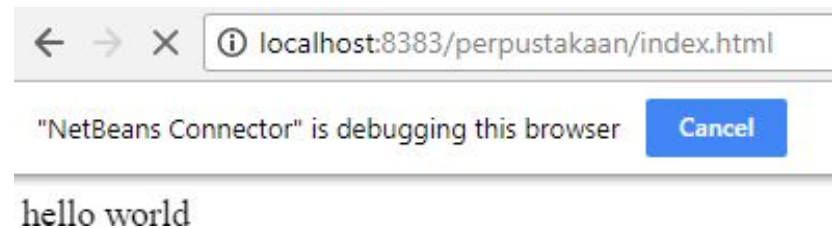
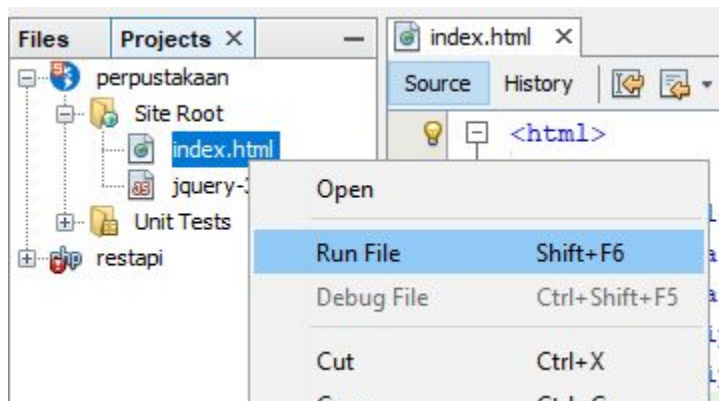
Download jquery



Test jquery loaded



```
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script type="text/javascript" src="jquery-3.3.1.min.js"></script>
    <script type="text/javascript">
      $(function () {
        document.write("hello world");
      });
    </script>
  </head>
  <body>
    <div>TODO write content</div>
  </body>
</html>
```



Membuat fungsi all

```
1 <html>
2   <head>
3     <title>TODO supply a title</title>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <script type="text/javascript" src="jquery-3.3.1.min.js"></script>
7     <script type="text/javascript">
8       $(function () {
9         all();
10      });
11      function all() {
12        $.ajax({
13          url: "http://localhost/restapi/pengarang/all"
14        }).then(function (data) {
15          console.log(data);
16        });
17      }
18    </script>
19  </head>
20  <body>
21    <div>TODO write content</div>
22  </body>
23 </html>
```


Debugging melalui DevTools



Safari

Apple



Firefox

Mozilla



Chrome

Google



Edge new

Microsoft



Opera

Opera Software

Testing melalui developer console

← → ↻ ⓘ localhost:8383/perpustakaan/index.html

"NetBeans Connector" is debugging this browser Cancel

TODO write content

Elements Console Network Performance Sources Applicati

top Filter Default levels ▼

Initialized content script!

GET http://localhost:8383/favicon.ico 0 ()

▼ (2) [{...}, {...}] ⓘ

▼ 0:

- email: "chandra@gmail.com"
- id: "1"
- nama: "Denny Chandra"
- ▶ __proto__: Object

▼ 1:

- email: "wendi@gmail.com"
- id: "2"
- nama: "Wendi Gunawan"
- ▶ __proto__: Object

length: 2

▶ __proto__: Array(0)

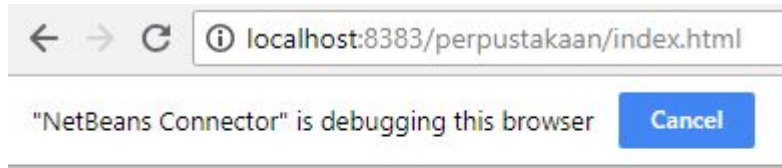
>

Menampilkan ke tabel

```
7 <script type="text/javascript">
8   $(function () {
9       all();
10    });
11    function all() {
12        $.ajax({
13            url: "http://localhost/restapi/pengarang/all"
14        }).then(function (data) {
15            var row = '';
16            $.each(data, function (idx, pengarang) {
17                row += '<tr>';
18                row += '<td>'+pengarang.id+'</td>';
19                row += '<td>'+pengarang.nama+'</td>';
20                row += '<td>'+pengarang.email+'</td>';
21                row += '</tr>';
22            });
23            $('#row').append(row);
24        });
25    }
26 </script>
27 </head>
```

Menampilkan ke tabel

```
28 <body>
29   <table>
30     <tr>
31       <th>Id</th>
32       <th>Nama</th>
33       <th>Email</th>
34     </tr>
35     <tbody id="row">
36     </tbody>
37   </table>
38 </body>
39 </html>
```



Id	Nama	Email
1	Denny Chandra	chandra@gmail.com
2	Wendi Gunawan	wendi@gmail.com

insert

```
<form id="form">
  <input type="hidden" name="id" id="id"/><br>
  Nama : <input type="text" name="nama" id="nama"/><br>
  Email : <input type="text" name="email" id="email"/><br>
  <button onclick="insertOrUpdate()" type="button">Simpan</button>
  <button onclick="hapus()" type="button">Hapus</button>
  <input id="reset" type="reset" value="Reset"/>
</form>
```

```
function insert() {
  $.ajax({
    url: "http://localhost/restapi/pengarang/insert",
    type: 'POST',
    data: 'nama=' + $('#nama').val() + '&email=' + $('#email').val()
  }).then(function (data) {
    console.log(data);
    all();
    $("#reset").trigger('click');
  });
}
```

insert

Nama :

Email :

<u>Id</u>	<u>Nama</u>	<u>Email</u>
<u>1</u>	Denny Chandra	chandra@gmail.com
<u>2</u>	Wendi Gunawan Arjuna	wendi@gmail.com

Nama :

Email :

<u>Id</u>	<u>Nama</u>	<u>Email</u>
<u>1</u>	Denny Chandra	chandra@gmail.com
<u>2</u>	Wendi Gunawan Arjuna	wendi@gmail.com
<u>9</u>	ihsan	ihsan@gmail.com

update

```
function all() {  
    $.ajax({  
        url: "http://localhost/restapi/pengarang/all"  
    }).then(function (data) {  
        var row = '';  
        $.each(data, function (idx, pengarang) {  
            row += '<tr>';  
            row += '<td onclick="prepareUpdate(' + pengarang.id + ') "><a href="#">' + pengarang.id + '</a></td>';  
        });  
    });  
}
```

```
function prepareUpdate(id) {  
    isNew = false;  
    $.ajax({  
        url: "http://localhost/restapi/pengarang/byid?id=" + id  
    }).then(function (data) {  
        $('#id').val(data.id);  
        $('#nama').val(data.nama);  
        $('#email').val(data.email);  
    });  
}
```

update

Nama :

Email :

Id	Nama	Email
<u>1</u>	Denny Chandra	chandra@gmail.com
<u>2</u>	Wendi Gunawan Arjuna	wendi@gmail.com
<u>9</u>	ihsan	ihsan@gmail.com

Nama :

Email :

Id	Nama	Email
<u>1</u>	Denny Chandra	chandra@gmail.com
<u>2</u>	Wendi Gunawan Arjuna	wendi@gmail.com
<u>9</u>	ihsan Maulana	ihsan@gmail.com

Insert menjadi insertOrUpdate

```
<script type="text/javascript">
```

```
    var isNew = true;
```

```
    $(function () {
```

```
function insertOrUpdate() {
```

```
    if (isNew) {
```

```
        $.ajax({
```

```
            url: "http://localhost/restapi/pengarang/insert",
```

```
            type: 'POST',
```

```
            data: 'nama=' + $('#nama').val() + '&email=' + $('#email').val()
```

```
        }).then(function (data) {
```

```
            console.log(data);
```

```
            all();
```

```
            $("#reset").trigger('click');
```

```
        });
```

```
    } else {
```

```
        $.ajax({
```

```
            url: "http://localhost/restapi/pengarang/update",
```

```
            type: 'POST',
```

```
            data: 'nama=' + $('#nama').val() + '&email=' + $('#email').val() + '&id=' + $('#id').val()
```

```
        }).then(function (data) {
```

```
            console.log(data);
```

```
            all();
```

```
        });
```

```
    }
```

```
    isNew = true;
```

```
    $("#reset").trigger('click');
```

```
}
```

hapus

```
function hapus() {  
    if (confirm('menghapus data?')) {  
        $.ajax({  
            url: "http://localhost/restapi/pengarang/delete",  
            type: 'POST',  
            data: 'id=' + $('#id').val()  
        }).then(function (data) {  
            console.log(data);  
            all();  
            $("#reset").trigger('click');  
            isNew = true;  
        });  
    }  
}
```

← → ↻ ⓘ localhost:8383/perpustakaan/index.html#

"NetBeans Connector" is debugging this browser [Cancel](#)

localhost:8383 says
menghapus data?
[OK](#) [Cancel](#)

Nama :
Email :
[Simpan](#) [Hapus](#) [Reset](#)

	Id	Nama	Email
1	Denny Chandra	chandra@gmail.com	
2	Wendi Gunawan Arjuna	wendi@gmail.com	
8	budi	budi@gmail.com	

Wassalamualaikum

“SEMOGA SUKSES”



- Lilis A. | ayumsari.lilis@gmail.com