**LAPORAN PRAKTIKUM PEMROGRAMAN**

**WEBSITE LANJUT**

**TUGAS PRAKTIKUM**

****

**Oleh :**

**M. WAHYU PRASETYO / NIM : 1841720177**

**KELAS TI 2B**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2020**

**Praktikum – Bagian 1: Instalasi library restful codeigniter**

|  |  |
| --- | --- |
| 1 | Silahkan unduh aplikasi Postman terlebih dahulu. |
| 2 | Pada praktikum kali ini kita akan menggunakan source codeigniter yang baru dan Library Rest Server. Untuk versi codeigniter yang digunakan adalah versi 3.1.9. sedangkan untuk Library Rest Server dapat di unduh pada link berikut : |
| 3 | Extract codeigniter yang sudah di unduh dan letakkan pada folder htdocs. Beri nama Code-kedelapan dan coba akses di <http://localhost/Code-kedelapan/>. Pastikan tampilan seperti gambar dibawah ini: |
| 4 | Selanjutnya lakukan konfigurasi dasar pada **config.php** seperti berikut:      Menjadi seperti berikut :      Dan file **autoload.php**      Menjadi      Dan database.php seperti berikut: |
| 5 | Selanjutnya buatlah database **rest\_server** pada phpmyadmin dan masukkan data dummy. Untuk data dummy dapat dibuat sebagai berikut:  CREATE TABLE `mahasiswa` (    `id` int(11) NOT NULL,    `nrp` char(9) NOT NULL,    `nama` varchar(250) NOT NULL,    `email` varchar(250) DEFAULT NULL,    `jurusan` varchar(64) NOT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  INSERT INTO `mahasiswa` (`id`, `nrp`, `nama`, `email`, `jurusan`) VALUES  (1, '043040001', 'Doddy Ferdiansyah', 'doy@gmail.com', 'Teknik Mesin'),  (2, '023040123', 'Erik', 'erik@gmail.com', 'Teknik Industri'),  (3, '043040321', 'Rommy Fauzi', 'rommy@gmail.com', 'Teknik Planologi'),  (4, '033040023', 'Fajar Darmawan ', 'fajar@yahoo.com', 'Teknik Informatika'),  (5, '113040321', 'Ferry Mulyanto', 'ferry@yahoo.com', 'Manajemen');  ALTER TABLE `mahasiswa`    ADD PRIMARY KEY (`id`);  ALTER TABLE `mahasiswa`    MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=33; |
| 6 | Selanjutnya extract file Library Rest Server, buka folder **application** dan duplikasi file dan folder tersebut kedalam folder **application** pada project **Code-kedelapan**. Klik **Replace** files jika ada notifikasi duplikat.          Setelah proses duplikasi file selesai, pastikan file-file yang kita sudah konfigurasi di awal tetap terkonfigurasi. |
| 7 | Selanjutnya buka project <http://localhost/Code-kedelapan/>. Jika berhasil, tampilan akan berubah menjadi seperti ini: |
| 8 | Tahap instalasi Library Rest Server selesai |

**Praktikum – Bagian 2: Membuat Aplikasi CRUD Restful Codeigniter**

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buatlah file **Mahasiswa.php** pada **controller/api**. Selanjutnya tuliskan kode berikut:  <?php  defined('BASEPATH') OR exit('No direct script access allowed');  require APPPATH . '/libraries/REST\_Controller.php';  require APPPATH . '/libraries/Format.php';  use Restserver\Libraries\REST\_Controller;  class Mahasiswa extends REST\_Controller  {      public function \_\_construct($config = 'rest')  {          parent::\_\_construct($config);          $this->load->model('Mahasiswa\_model', 'mahasiswa');      }      public function index\_get()  {          $id = $this->get('id');          if ($id === null) {              $mahasiswa = $this->mahasiswa->getMahasiswa();          } else {              $this->db->where('id', $id);              $mahasiswa = $this->mahasiswa->getMahasiswa($id);          }            if ($mahasiswa) {              $this->response([                  'status' => true,                  'data' => $mahasiswa              ], REST\_Controller::HTTP\_OK);          } else {              $this->response([                  'status' => true,                  'data' => 'id not found'              ], REST\_Controller::HTTP\_NOT\_FOUND);          }      }  }  ?> |
| 2 | Selanjutnya Buatlah file **Mahasiswa\_model.php** pada model. Selanjutnya tuliskan kode berikut:  <?php  defined('BASEPATH') OR exit('No direct script access allowed');  class Mahasiswa\_model extends CI\_Model {      public function getMahasiswa($id=null){          if ($id === null) {              return $this->db->get('mahasiswa')->result\_array();          } else {              return $this->db->get\_where('mahasiswa', ['id' => $id])->result\_array();          }      }  }  /\* End of file Mahasiswa\_model.php \*/  ?> |
| 3 | Buka aplikasi postman, masukkan alamat <http://localhost/Code-kedelapan/api/mahasiswa> dengan metode **GET**. dan jika berhasil maka tampilannya akan seperti ini:    Kita juga dapat melakukan pencarian berdasarkan id, hasilnya akan seperti ini: |
| 4 | Untuk fungsi lain **Create**, **Update** dan **Delete**. Lanjutkan kode berikut pada file **mahasiswa.php**:  // Delete      public function index\_delete()      {          $id = $this->delete('id');          if ($id === null) {              $this->response([                  'status' => false,                  'message' => 'provide an id'              ], REST\_Controller::HTTP\_BAD\_REQUEST);          } else {              if ($this->mahasiswa->deleteMahasiswa($id) > 0) {                  $this->response([                      'status' => true,                      'id' => $id,                      'message' => 'deleted'                  ], REST\_Controller::HTTP\_OK);              } else {                  $this->response([                      'status' => false,                      'message' => 'id not found!'                  ], REST\_Controller::HTTP\_BAD\_REQUEST);              }          }      }  // post      public function index\_post()      {          $data = [              'nrp' => $this->post('nrp'),              'nama' => $this->post('nama'),              'email' => $this->post('email'),              'jurusan' => $this->post('jurusan'),          ];          if ($this->mahasiswa->createMahasiswa($data) > 0) {              $this->response([                  'status' => true,                  'message' => 'new mahasiswa has been created'              ], REST\_Controller::HTTP\_CREATED);          } else {              $this->response([                  'status' => false,                  'message' => 'failed to create new data'              ], REST\_Controller::HTTP\_BAD\_REQUEST);          }      }  // Put      public function index\_put()      {          $id = $this->put('id');          $data = [              'nrp' => $this->put('nrp'),              'nama' => $this->put('nama'),              'email' => $this->put('email'),              'jurusan' => $this->put('jurusan'),          ];          if ($this->mahasiswa->updateMahasiswa($data, $id) > 0) {              $this->response([                  'status' => true,                  'message' => 'new mahasiswa has been updated'              ], REST\_Controller::HTTP\_OK);          } else {              $this->response([                  'status' => false,                  'message' => 'failed to update data'              ], REST\_Controller::HTTP\_BAD\_REQUEST);          }      } |
| 5 | Dan lanjutkan kode pada file **Mahasiswa\_model.php**:  public function deleteMahasiswa($id){          $this->db->delete('mahasiswa', ['id' => $id]);          return $this->db->affected\_rows();        }      public function createMahasiswa($data){          $this->db->insert('mahasiswa', $data);          return $this->db->affected\_rows();        }      public function updateMahasiswa($data, $id){          $this->db->update('mahasiswa', $data, ['id' => $id]);          return $this->db->affected\_rows();        } |
| 6 | Selanjutnya lakukan operasi **Create** data pada aplikasi postman, untuk operasi **Create**, **Update** dan **Delete**, arahkan parameter ke **Body** dan **x-www-form-urlencoded** seperti berikut: |
| 7 | Lakukan juga operasi **Update** data pada aplikasi postman, masukkan parameter id seperti berikut: |
| 8 | Lakukan juga operasi **Delete** data pada aplikasi postman, parameter yang dimasukkan cukup id saja, seperti berikut: |