

Pengumpulan Tugas Akhir PWBS Sesi 3 (UAS)

Kelompok : PROLA

Ketua : Ryan Satria – 19312045 (Front End)

Anggota :

- 1) Imam Asyrofi Alfarisi -19312141 (Back End)
- 2) Bagas Aditama – 19312094 (Back End)

Secrenshoot dan Penjelasan Program (Back End)

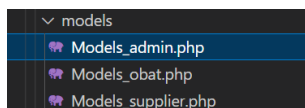
1. Pada laporan kali ini, kami melanjutkan terkait pembuatan JWT, yang mana kami menambahkan source code seperti pada gambar dibawah ini.

```
14 class Token extends Server
15 {
16     var $privateKey = <<<EOD
17     -----BEGIN RSA PRIVATE KEY-----
18     MIICBAIBAAKggQChka1p5jbsYZvbtTRBLx8zSH412p/11CrEhtaSkau/Rn
19     vuE4M8oH3+3luYw4VQaZyqFwuzJkOI+17t5t8tyazyZ8JXw+KjXTx1dMPEL9
20     5+qVhgXwrt1hXC1cSoGbRIEDvDF6Sa53rcFvayJ4ehde/zUxo6Uv57UrBQIDAQAB
21     AoGAB/XOV46XCFRvNU8BLyAtMLDgi/xRnTALJHJ5ACdddkykem8//Beztw9Fzrz
22     bWZ/1/doQQUHBYZU8aDzzj59FZ78dyZNFoF91hVZKkg+6wdy/LNGV5+Xre0
23     N1L00wH2HONZUYvHvHY7P1Q+8BCHz1LQsg*28k6a0DF8CQ0zbpjYx1Bqg1
24     pFS9UHIjP2YCk1cSa1z7+uMnt3QhwaQ2DC11Qm2NuybwwbIDkxw55agM85
25     StNoQDIFaEaxkyfFUsythgg/H81xe1I2z680bF85b1z0/177Det3n4VsjYIDdk
26     ZkcV35K2IRL4c9yY9h1Ihs+K9WKTQWw7BA099sk148m07woPR9UD22jDpN8ue
27     k980HepTj25v1hjbfuPH1IdhqvS2TDychWm1k177LQZg08Fv90CFZ/gqMCQG0b
28     qe0wHmU6x2pUbbacnYrIM6nL5kXg0aw7XXCoJvY6147LTK31X1LBOxPu5Eu13k
29     eUz9sHyD6vkgZzjtXXECQAp4XerFSTGFQXhXIX52yHhN2LtuJcdkQZjXasGdm
30     B2ZhzvrlgRng8nk1NtrHYgn1NPcw+RLGcwGz2PTVWt+
31     -----END RSA PRIVATE KEY-----
32     EOD;
33
34     var $publicKey = <<<EOD
35     -----BEGIN PUBLIC KEY-----
36     MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKgQChka1p5jbsYZvbtTRBLx8zSH
37     412p/11CrEhtaSkau/RnvuE4M8oH3+3luYw4VQaZyqFwuzJkOI+17t5t
38     8tyazyZ8JXw+KjXTx1dMPEL95+qVhgXwrt1hXC1cSoGbRIEDvDF6Sa53rcFvayJ4
39     ehde/zUxo6Uv57UrBQIDAQAB
40     -----END PUBLIC KEY-----
41     EOD;
```

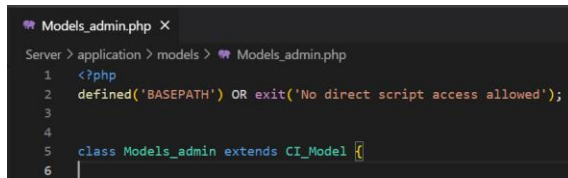
2. Lalu kami membuat class service_post, yang mana class ini berguna untuk mengecek apakah token benar atau salah.

```
67 //membuat fungsi "POST"
68 function service_post()
69 {
70     date_default_timezone_set("Asia/Jakarta");
71     $exp = $this->configToken()['exp']+time();
72     $token = array(
73         "exp" => $exp,
74     );
75
76     // $jwt = JWT::encode($token, $this->configToken()['key'],'HS256');
77     $jwt = JWT::encode($token, $this->privateKey,'RS256');
78
79
80     $data = array('token' => $jwt, 'exp' => date("d/m/Y H:i",$exp));
81     $this->response($data, 200);
82 }
83
84 }
```

3. Setelah itu kami membuat models baru yang kami beri nama Models_admin.php

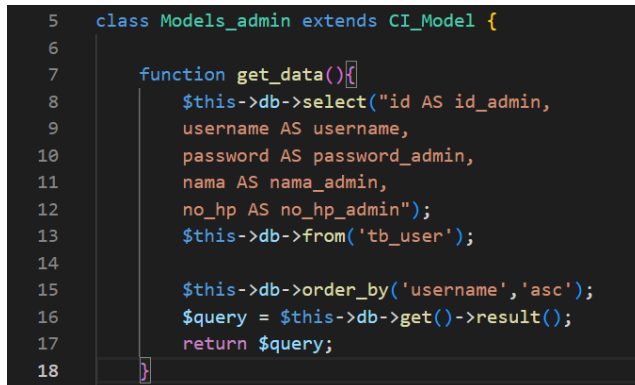


4. Kemudian kami ganti nama class nya menjadi Models_admin seperti pada gambar dibawah ini.



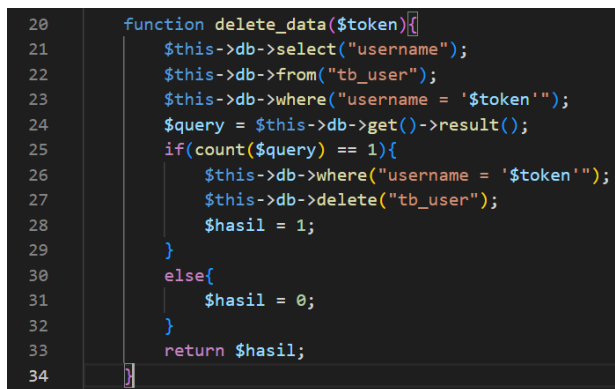
```
Models_admin.php X
Server > application > models > Models_admin.php
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4
5 class Models_admin extends CI_Model {
6
```

5. Kemudian kami membuat method untuk menampilkan data yang kami beri nama get_data, data tersebut kami urutkan secara ascending berdasarkan username.



```
5 class Models_admin extends CI_Model {
6
7     function get_data(){
8         $this->db->select("id AS id_admin,
9             username AS username,
10            password AS password_admin,
11            nama AS nama_admin,
12            no_hp AS no_hp_admin");
13         $this->db->from('tb_user');
14
15         $this->db->order_by('username','asc');
16         $query = $this->db->get()->result();
17         return $query;
18     }
```

6. Lalu kami membuat method untuk menghapus data yang kami beri nama delete_data, proses penghapusan data dilakukan berdasarkan username.



```
20 function delete_data($token){
21     $this->db->select("username");
22     $this->db->from("tb_user");
23     $this->db->where("username = '$token'");
24     $query = $this->db->get()->result();
25     if(count($query) == 1){
26         $this->db->where("username = '$token'");
27         $this->db->delete("tb_user");
28         $hasil = 1;
29     }
30     else{
31         $hasil = 0;
32     }
33     return $hasil;
34 }
```

7. Setelah itu kami membuat method untuk menyimpan atau save data yang kami beri nama save_data, proses menyimpan data dilakukan dengan cara mengisi nilai di semua field yang ada di dalam tabel, kecuali id, yang mana id akan terisi otomatis, dikarenakan ketika membuat tabel, dia bersifat auto increment atau isi otomatis.

```

36     function save_data($username,$password,$nama,$no_hp,$token){
37         $this->db->select("username");
38         $this->db->from("tb_user");
39         $this->db->where("TO_BASE64(username) = '$token'");
40         $query = $this->db->get()->result();
41         if(count($query) == 0){
42             $data = array(
43                 "username" => $username,
44                 "password" => $password,
45                 "nama" => $nama,
46                 "no_hp" => $no_hp,
47             );
48             $this->db->insert("tb_user",$data);
49             $hasil = 0;
50         }
51         else{
52             $hasil = 1;
53         }
54         return $hasil;
55     }

```

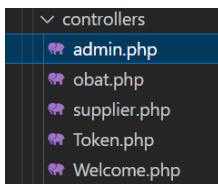
8. Kemudian kami membuat method untuk mengubah atau update data yang kami beri nama `update_data`. Proses mengubah atau update data dilakukan berdasarkan username.

```

57     function update_data($username,$password,$nama,$no_hp,$token){
58         $this->db->select("username");
59         $this->db->from("tb_user");
60         $this->db->where("TO_BASE64(username) != '$token' AND username = '$username'");
61         $query = $this->db->get()->result();
62         if(count($query) == 0){
63             $data = array(
64                 "username" => $username,
65                 "password" => $password,
66                 "nama" => $nama,
67                 "no_hp" => $no_hp,
68             );
69
70             $this->db->where("TO_BASE64(username) != '$token'");
71             $this->db->update("tb_user",$data);
72             $hasil = 0;
73         }
74         else{
75             $hasil = 1;
76         }
77
78         return $hasil;
79     }

```

9. Dibawah ini merupakan proses dari pembuatan file controllers baru di dalam folder controllers, yang mana controllers tersebut kami beri nama `admin.php`



10. Kemudian class nya kami ubah sesuai dengan nama models yaitu `admin`.

```

admin.php X
Server > application > controllers > admin.php
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  require APPPATH."libraries/Server.php";
5  class admin extends Server {

```

11. Lalu pada fungsi construct nya kami ubah nama models menjadi Models_admin, seperti pada gambar dibawah ini.

```
4  require APPPATH."libraries/Server.php";
5  class admin extends Server {
6
7      public function __construct()
8      {
9          parent::__construct();
10
11          $this->load->model("Models_admin","model",TRUE);
12      }
13  }
```

12. Kemudian kami membuat fungsi get yang kami beri nama service_get, yang mana fungsi tersebut untuk mengambil data. Di dalam fungsi tersebut, kita panggil file models yaitu Models_admin, kemudian kita panggil function get_data yang ada di dalam Models_admin. Di dalam fungsi tersebut sudah tersedia autentifikasi JWT yang menggunakan token untuk autentifikasi.

```
15  function service_get(){
16      if ($this->token_login() == 0) {
17          return $this->response(array("result" => 0, "error" => "Kode Signature Tidak Sesuai !"), 200);
18      }
19      else{
20          $this->load->model("Models_admin","mdl",TRUE);
21
22          $hasil = $this->mdl->get_data();
23
24          $this->response(array("admin" => $hasil),200);
25      }
26  }
27  }
```

13. Lalu kami membuat fungsi put yang kami beri nama service_put, yang mana fungsi tersebut untuk mengubah atau mengupdate data. Di dalam fungsi tersebut, kita panggil file models yaitu Models_admin, kemudian kami mengambil parameter yang akan di isi. Di dalam fungsi tersebut sudah tersedia autentifikasi JWT yang menggunakan token untuk autentifikasi.

```
30  function service_put(){
31      if ($this->token_login() == 0) {
32          return $this->response(array("result" => 0, "error" => "Kode Signature Tidak Sesuai !"), 200);
33      }
34      else{
35          $data = array(
36              "username" => $this->put("username"),
37              "password" => $this->put("password"),
38              "nama" => $this->put("nama"),
39              "no_hp" => $this->put("no_hp"),
40              "token" => base64_encode($this->put("token")),
41          );
42
43          $hasil = $this->model->update_data($data["username"],$data["password"],$data["nama"],
44              $data["no_hp"],$data["token"]);
45
46          if($hasil == 0){
47              $this->response(array("status"=>"Data admin Berhasil Diubah"),200);
48          }
49          else{
50              $this->response(array("status"=>"Data admin Gagal Diubah!"),200);
51          }
52      }
53  }
54  }
55  }
```

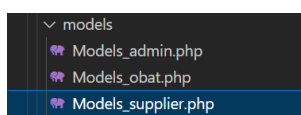
14. Setelah itu kami membuat fungsi post yang kami beri nama `service_post`, yang mana fungsi tersebut untuk menyimpan atau save data. Di dalam fungsi tersebut, kita panggil file models yaitu `Models_admin`, kemudian kami mengambil parameter yang akan di isi. Di dalam fungsi tersebut sudah tersedia autentifikasi JWT yang menggunakan token untuk autentifikasi.

```
57 function service_post(){
58     if ($this->token_login() == 0) {
59         return $this->response(array("result" => 0, "error" => "Kode Signature Tidak Sesuai !"), 200);
60     }
61     else{$data = array(
62         "username" => $this->post("username"),
63         "password" => $this->post("password"),
64         "nama" => $this->post("nama"),
65         "no_hp" => $this->post("no_hp"),
66         "token" => base64_encode($this->put("token")),
67     );
68
69     $hasil = $this->model->save_data($data["username"],$data["password"],$data["nama"],
70     $data["no_hp"],$data["token"]);
71
72     if($hasil == 0){
73         $this->response(array("status"=>"Data admin Berhasil Disimpan"),200);
74     }
75
76     else{
77         $this->response(array("status"=>"Data admin Gagal Disimpan!"),200);
78     }
79
80 }
81 }
```

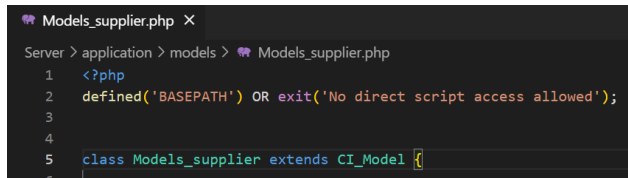
15. Setelah itu kami membuat fungsi delete yang kami beri nama `service_delete`, yang mana fungsi tersebut untuk menghapus atau delete data. Di dalam fungsi tersebut, kita panggil file models yaitu `Models_admin`, proses penghapusan data dilakukan berdasarkan username. Di dalam fungsi tersebut sudah tersedia autentifikasi JWT yang menggunakan token untuk autentifikasi.

```
83 function service_delete(){
84     if ($this->token_login() == 0) {
85         return $this->response(array("result" => 0, "error" => "Kode Signature Tidak Sesuai !"), 200);
86     }
87     else{$token = $this->delete("username");
88
89     $hasil = $this->model->delete_data($token);
90
91     if($hasil == 1)
92     {
93         $this->response(array("status"=>"Data admin berhasil dihapus"),200);
94     }
95     else
96     {
97         $this->response(array("status" => "Data admin gagal dihapus !"), 200);
98     }
99
100 }
101 }
102 }
```

16. Setelah itu kami membuat models baru yang kami beri nama `Models_supplier.php`

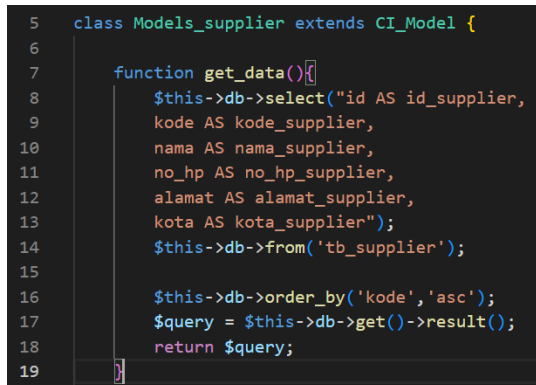


17. Kemudian kami ganti nama class nya menjadi Models_supplier seperti pada gambar dibawah ini.



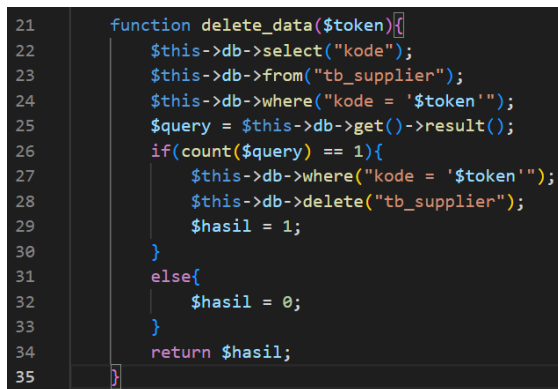
```
Models_supplier.php X
Server > application > models > Models_supplier.php
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4
5  class Models_supplier extends CI_Model {
```

18. Kemudian kami membuat method untuk menampilkan data yang kami beri nama get_data, data tersebut kami urutkan secara ascending berdasarkan kode supplier.



```
5  class Models_supplier extends CI_Model {
6
7      function get_data(){
8          $this->db->select("id AS id_supplier,
9              kode AS kode_supplier,
10             nama AS nama_supplier,
11             no_hp AS no_hp_supplier,
12             alamat AS alamat_supplier,
13             kota AS kota_supplier");
14          $this->db->from('tb_supplier');
15
16          $this->db->order_by('kode', 'asc');
17          $query = $this->db->get()->result();
18          return $query;
19      }
```

19. Lalu kami membuat method untuk menghapus data yang kami beri nama delete_data, proses penghapusan data dilakukan berdasarkan kode supplier.



```
21  function delete_data($token){
22      $this->db->select("kode");
23      $this->db->from("tb_supplier");
24      $this->db->where("kode = '$token'");
25      $query = $this->db->get()->result();
26      if(count($query) == 1){
27          $this->db->where("kode = '$token'");
28          $this->db->delete("tb_supplier");
29          $hasil = 1;
30      }
31      else{
32          $hasil = 0;
33      }
34      return $hasil;
35  }
```

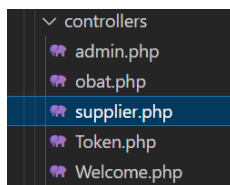
20. Setelah itu kami membuat method untuk menyimpan atau save data yang kami beri nama `save_data`, proses menyimpan data dilakukan dengan cara mengisi nilai di semua field yang ada di dalam tabel, kecuali id, yang mana id akan terisi otomatis, dikarenakan ketika membuat tabel, dia bersifat auto increment atau isi otomatis.

```
37 function save_data($kode,$nama,$no_hp,$alamat,$kota,$token){
38     $this->db->select("kode");
39     $this->db->from("tb_supplier");
40     $this->db->where("TO_BASE64(kode) = '$token'");
41     $query = $this->db->get()->result();
42     if(count($query) == 0){
43         $data = array(
44             "kode" => $kode,
45             "nama" => $nama,
46             "no_hp" => $jenis,
47             "alamat" => $alamat,
48             "kota" => $kota,
49         );
50         $this->db->insert("tb_supplier",$data);
51         $hasil = 0;
52     }
53     else{
54         $hasil = 1;
55     }
56     return $hasil;
57 }
```

21. Kemudian kami membuat method untuk mengubah atau update data yang kami beri nama `update_data`. Proses mengubah atau update data dilakukan berdasarkan kode supplier.

```
59 function update_data($kode,$nama,$no_hp,$alamat,$kota,$token){
60     $this->db->select("kode");
61     $this->db->from("tb_supplier");
62     $this->db->where("TO_BASE64(kode) != '$token' AND kode = '$kode'");
63     $query = $this->db->get()->result();
64     if(count($query) == 0){
65         $data = array(
66             "kode" => $kode,
67             "nama" => $nama,
68             "no_hp" => $jenis,
69             "alamat" => $alamat,
70             "kota" => $kota,
71         );
72     }
73     $this->db->where("TO_BASE64(kode) != '$token'");
74     $this->db->update("tb_supplier",$data);
75     $hasil = 0;
76 }
77 else{
78     $hasil = 1;
79 }
80 return $hasil;
81 }
82 }
83 }
84 }
```

22. Dibawah ini merupakan proses dari pembuatan file controllers baru di dalam folder controllers, yang mana controllers tersebut kami beri nama `supplier.php`



23. Kemudian class nya kami ubah sesuai dengan nama models yaitu supplier.

```
supplier.php X
Server > application > controllers > supplier.php
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 require APPPATH.'libraries/Server.php';
5 class supplier extends Server {
```

24. Lalu pada fungsi construct nya kami ubah nama models menjadi Models_supplier, seperti pada gambar dibawah ini.

```
4 require APPPATH.'libraries/Server.php';
5 class supplier extends Server {
6
7     public function __construct()
8     {
9         parent::__construct();
10
11         $this->load->model("Models_supplier", "model", TRUE);
12     }
13 }
```

25. Kemudian kami membuat fungsi get yang kami beri nama service_get, yang mana fungsi tersebut untuk mengambil data. Di dalam fungsi tersebut, kita panggil file models yaitu Models_supplier, kemudian kita panggil function get_data yang ada di dalam Models_supplier. Di dalam fungsi tersebut sudah tersedia autentifikasi JWT yang menggunakan token untuk autentifikasi.

```
15 function service_get(){
16     if ($this->token_login() == 0) {
17         return $this->response(array("result" => 0, "error" => "Kode Signature Tidak Sesuai !"), 200);
18     }
19     else{
20         $this->load->model("Models_supplier", "mdl", TRUE);
21
22         $hasil = $this->mdl->get_data();
23
24         $this->response(array("supplier" => $hasil), 200);
25     }
26 }
27 }
```

26. Lalu kami membuat fungsi put yang kami beri nama service_put, yang mana fungsi tersebut untuk mengubah atau mengupdate data. Di dalam fungsi tersebut, kita panggil file models yaitu Models_supplier, kemudian kami mengambil parameter yang akan di isi. Di dalam fungsi tersebut sudah tersedia autentifikasi JWT yang menggunakan token untuk autentifikasi.


```

30 function service_put() {
31     if ($this->token_login() == 0) {
32         return $this->response(array("result" => 0, "error" => "Kode Signature Tidak Sesuai !"), 200);
33     }
34     else {
35         $data = array(
36             "kode" => $this->put("kode"),
37             "nama" => $this->put("nama"),
38             "no_hp" => $this->put("no_hp"),
39             "alamat" => $this->put("alamat"),
40             "kota" => $this->put("kota"),
41             "token" => base64_encode($this->put("token")),
42         );
43
44         $hasil = $this->model->update_data($data["kode"], $data["nama"], $data["no_hp"],
45             $data["alamat"], $data["kota"], $data["token"]);
46
47         if ($hasil == 0) {
48             $this->response(array("status" => "Data supplier Berhasil Diubah"), 200);
49         }
50         else {
51             $this->response(array("status" => "Data supplier Gagal Diubah!"), 200);
52         }
53     }
54 }
55
56

```

27. Setelah itu kami membuat fungsi post yang kami beri nama `service_post`, yang mana fungsi tersebut untuk menyimpan atau save data. Di dalam fungsi tersebut, kita panggil file models yaitu `Models_supplier`, kemudian kami mengambil parameter yang akan di isi. Di dalam fungsi tersebut sudah tersedia autentifikasi JWT yang menggunakan token untuk autentifikasi.

```

58 function service_post() {
59     if ($this->token_login() == 0) {
60         return $this->response(array("result" => 0, "error" => "Kode Signature Tidak Sesuai !"), 200);
61     }
62     else {
63         $data = array(
64             "kode" => $this->post("kode"),
65             "nama" => $this->post("nama"),
66             "no_hp" => $this->post("no_hp"),
67             "alamat" => $this->post("alamat"),
68             "kota" => $this->post("kota"),
69             "token" => base64_encode($this->post("token")),
70         );
71
72         $hasil = $this->model->update_data($data["kode"], $data["nama"],
73             $data["no_hp"], $data["alamat"], $data["kota"], $data["token"]);
74
75         if ($hasil == 0) {
76             $this->response(array("status" => "Data supplier Berhasil Disimpan"), 200);
77         }
78         else {
79             $this->response(array("status" => "Data supplier Gagal Disimpan!"), 200);
80         }
81     }
82 }
83

```

28. Setelah itu kami membuat fungsi delete yang kami beri nama `service_delete`, yang mana fungsi tersebut untuk menghapus atau delete data. Di dalam fungsi tersebut, kita panggil file models yaitu `Models_supplier`, proses penghapusan data dilakukan berdasarkan kode supplier. Di dalam fungsi tersebut sudah tersedia autentifikasi JWT yang menggunakan token untuk autentifikasi.

```

85 function service_delete() {
86     if ($this->token_login() == 0) {
87         return $this->response(array("result" => 0, "error" => "Kode Signature Tidak Sesuai !"), 200);
88     }
89     else { $token = $this->delete("kode");
90
91         $hasil = $this->model->delete_data($token);
92
93         if ($hasil == 1)
94         {
95             $this->response(array("status" => "Data supplier berhasil dihapus"), 200);
96         }
97         else
98         {
99             $this->response(array("status" => "Data supplier gagal dihapus !"), 200);
100         }
101     }
102 }
103
104 }

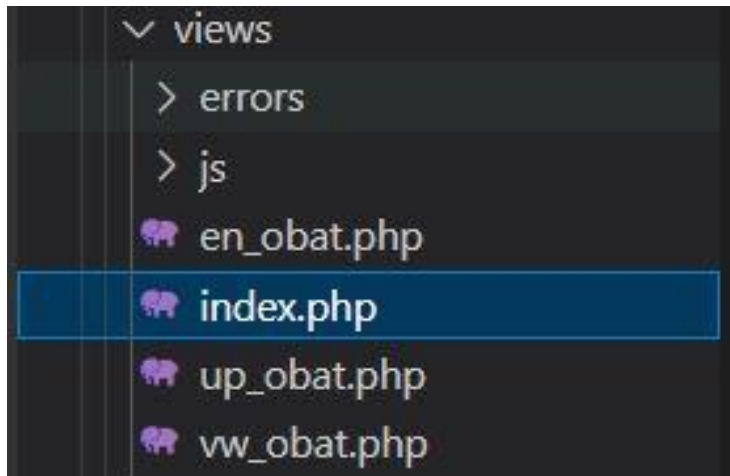
```

29. Dan yang terakhir yaitu kami update data obat, yang mana ada beberapa obat yang kami inputkan berdasarkan studi kasus kami yaitu pengelolaan data obat.

1 Messages 2 Table Data 3 Info						
<input type="checkbox"/>	id	kode	nama	jenis	harga	stok
<input type="checkbox"/>	18	B0001	Fatigon	Vitamin	20000	34
<input type="checkbox"/>	19	B0002	Paramex	Obat	12000	23
<input type="checkbox"/>	20	B0003	Bodrex	Obat	5000	45
<input type="checkbox"/>	21	B0004	OBH Combi	Obat	25000	24
<input type="checkbox"/>	22	B0005	OBH Anak	Obat	20000	12
<input type="checkbox"/>	23	B0006	Panadol	Obat	30000	34
<input type="checkbox"/>	24	B0007	Diapet	Obat	30000	43
<input type="checkbox"/>	25	B0008	Konidin	Obat	23000	23
<input type="checkbox"/>	26	B0009	Stimuno	Vitamin	30000	12
<input type="checkbox"/>	27	B0010	Hemaviton	Vitamin	25000	25
*	(Auto)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Secrenshoot dan Penjelasan Program (Front End)

1. Membuat file index.php di dalam Client lalu ke folder view.

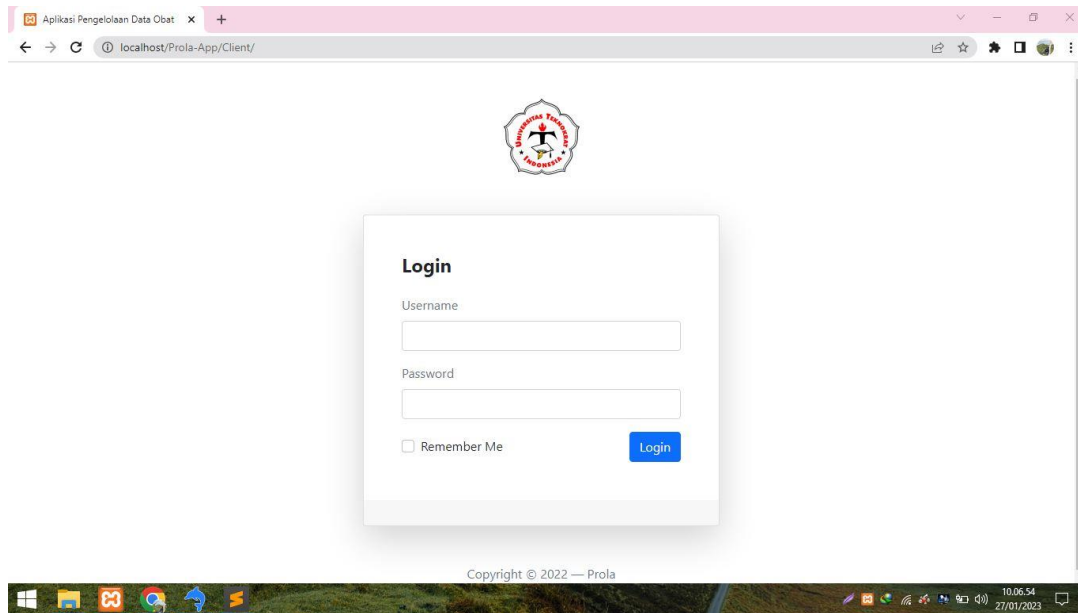


2. Isi kode tampilan index yaitu tampilan login dari aplikasi.

```
index.php x
Client > application > views > index.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <meta name="author" content="-">
6   <meta name="viewport" content="width=device-width,initial-scale=1">
7   <meta name="description" content="This is a login page template based on Bootstrap 5">
8   <title>Aplikasi Pengelolaan Data Obat</title>
9   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-...>
10 </head>
11
12 <body>
13   <section class="h-100">
14     <div class="container h-100">
15       <div class="row justify-content-sm-center h-100">
16         <div class="col-xxl-4 col-xl-5 col-lg-5 col-md-7 col-sm-9">
17           <div class="text-center my-5">
18             
19           </div>
20           <div class="card shadow-lg">
21             <div class="card-body p-5">
22               <h1 class="fs-4 card-title fw-bold mb-4">Login</h1>
23               <form method="POST" class="needs-validation" novalidate="" autocomplete="off">
24                 <div class="mb-3">
25                   <label class="mb-2 text-muted" for="">Username</label>
26                   <input id="" type="text" class="form-control" name="" value="" required autofocus>

```

3. Tampilan halaman login website



4. Lalu pada bagian controller Obat.php, memanggil index untuk ditampilkan di awal website, lalu menambahkan variable KEY, serta menambahkan code untuk key_bearer agar dapat diakses fungsi dari index pada controller.

```

Obat.php M X
Client > application > controllers > Obat.php
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class Obat extends CI_Controller {
5
6      // buat variabel global
7      var $key_name = 'KEY-API';
8      var $key_value = 'RESTAPI';
9      var $key_bearer = 'Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJleHAiOiJlE2NzI3Njc4NzB9.sR9o91Cm73t
10
11     public function index()
12     {
13         $this->client->http_header($this->key_bearer);
14
15         $data["tampil"] = json_decode($this->client->simple_get(APIOBAT, [$this->key_name => $this->key_value]));
16
17         if ($data["tampil"]->result == 0) {
18             echo $data["tampil"]->error;
19         } else {
20             $this->load->view('index', $data);
21         }
22     }
23
24

```

5. Lalu penambahan code untuk key_bearer dan key_name agar dapat diakses fungsi dari delete pada controller.

```
function setDelete()
{
    $this->client->http_header($this->key_bearer);

    //buat variable
    $json = file_get_contents("php://input");
    $hasil = json_decode($json);

    $delete = json_decode($this->client->simple_delete(APIOBAT,array("kode" => $hasil->kodenya, $this->key_name =>

if ($delete->result == 0) {
    echo json_encode(array("statusnya" => $delete->error));
} else {
    echo json_encode(array("statusnya" => $delete->status));
}
}
```

6. Lalu penambahan code untuk key_bearer dan key_name agar dapat diakses fungsi dari setSave pada controller.

```
Obat.php M
Client > application > controllers > Obat.php
29
30 function setSave()
31 {
32     $this->client->http_header($this->key_bearer);
33
34     //baca nilai dari fetch
35     $data = array(
36         "kode" => $this->input->post("kodenya"),
37         "nama" => $this->input->post("namanya"),
38         "jenis" => $this->input->post("jenisnya"),
39         "harga" => $this->input->post("harganya"),
40         "stok" => $this->input->post("stoknya"),
41         "token" => $this->input->post("tokennya"),
42         $this->key_name => $this->key_value
43     );
44
45     $save = json_decode($this->client->simple_post(APIOBAT,$data));
46
47     if ($save->result == 0) {
48         echo json_encode(array("statusnya" => $save->error));
49     } else {
50         echo json_encode(array("statusnya" => $save->status));
51     }
52
53 }
54
```

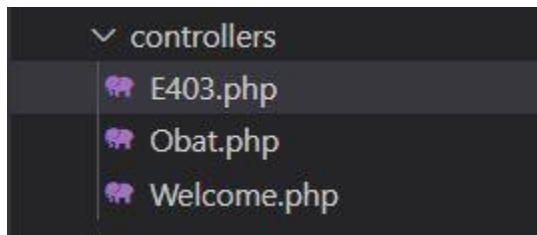
7. Lalu penambahan code untuk key_bearer dan key_name agar dapat diakses fungsi dari updateObat pada controller.

```
Obat.php M X
Client > application > controllers > Obat.php
74 //fungsi untuk update data
75 function updateObat()
76 {
77     $this->client->http_header($this->key_bearer);
78     //ambil nilai kode
79     $token = $this->uri->segment(3);
80     $stampil = json_decode($this->client->simple_get(APIOBAT, array("kode" => $token, $this->key_name => $this->key_value)));
81
82     if ($stampil->result == 0) {
83         echo $stampil->error;
84     } else {
85         foreach($stampil->obat as $result)
86         {
87             //echo $result->nama_mhs."<br>";
88             $data = array(
89                 "kode" => $result->kode_obat,
90                 "nama" => $result->nama_obat,
91                 "jenis" => $result->jenis_obat,
92                 "harga" => $result->harga_obat,
93                 "stok" => $result->stok_obat,
94                 "token" => $token,
95             );
96         }
97         $this->load->view('up_obat', $data);
98     }
99 }
```

8. Lalu penambahan code untuk key_bearer dan key_name agar dapat diakses fungsi dari setUpdate pada controller.

```
Obat.php M X
Client > application > controllers > Obat.php
103 //buat fungsi untuk simpan data mahasiswa
104 function setUpdate()
105 {
106     $this->client->http_header($this->key_bearer);
107     //baca nilai dari fetch
108     $data = array(
109         "kode" => $this->input->post("kodenya"),
110         "nama" => $this->input->post("namanya"),
111         "jenis" => $this->input->post("jenisnya"),
112         "harga" => $this->input->post("harganya"),
113         "stok" => $this->input->post("stoknya"),
114         "token" => $this->input->post("tokennya"),
115         $this->key_name => $this->key_value
116     );
117
118     $update = json_decode($this->client->simple_put(APIOBAT,$data));
119     // kirim hasil ke "up_obat"
120     if ($update->result == 0) {
121         echo json_encode(array("statusnya" => $update->error));
122     } else {
123         echo json_encode(array("statusnya" => $update->status));
124     }
125 }
126 }
```

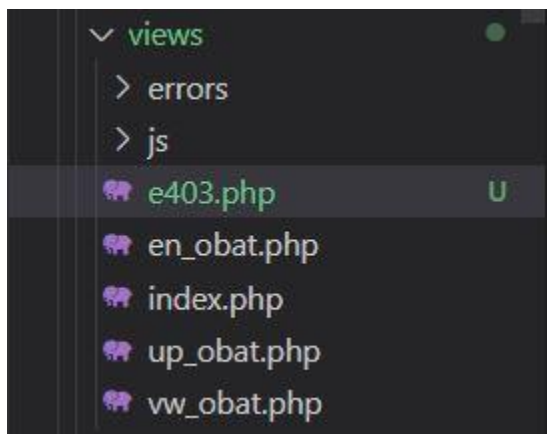

9. Menambah controller error pada folder controllers dengan file E403.php



10. Code pada file E403.php mengarahkan ke bagian tampilan yaitu mengarah ke view dan akan dibuat file untuk tampilannya pada nomor berikut



11. Menambahkan file untuk menampilkan error dengan nama file e403.php



12. Isi code tampilan error, menampilkan tidak dapat diakses

```
e403.php U X
Client > application > views > e403.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>403</title>
8  </head>
9  <body>
10     <p>Error 403 - Forbidden Access</p>
11 </body>
12 </html>
```