

Tugas Akhir PWBS

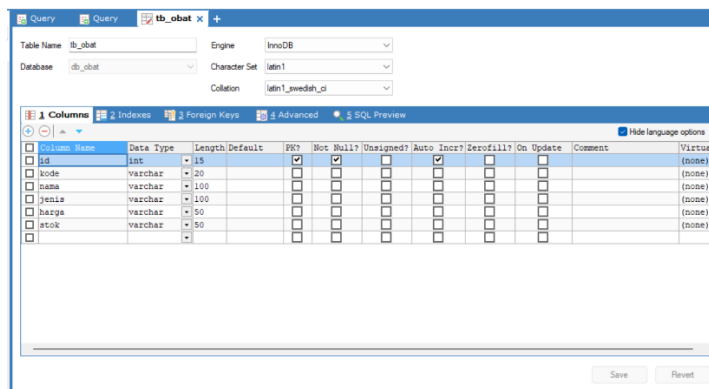
Penjelasan Backend Sesi 1

Bagian Backend :

- 1) Imam Asyrofi Alfarisi -19312141
- 2) Bagas Aditama - 19312094

Secrenshoot dan Penjelasan Program

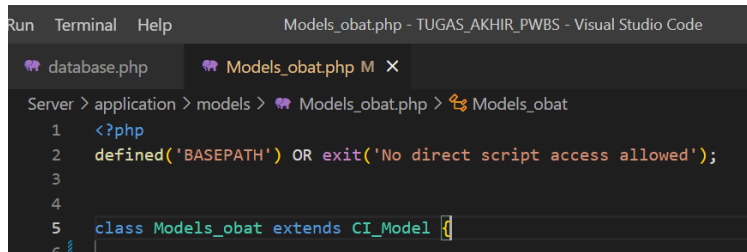
1. Pembuatan Database, dibawah ini merupakan proses pembuatan database beserta tabel, kami membuat database obat dan tabel obat sebagai database nya. Terdapat 6 field di dalam tabel kami, yaitu id, kode, nama, jenis, harga, dan stok.



2. Kemudian kami membuat terlebih dahulu bagian back end, yang mana langkah pertama yaitu konfigurasi database terlebih dahulu dengan cara mengubah nama database sesuai dengan yang telah kami buat, yaitu db_obat, dan mengisi username dengan root seperti gambar dibawah ini.

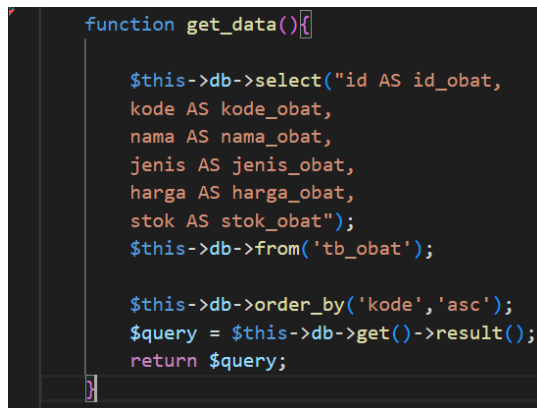
```
$db['default'] = array(  
    'dsn' => '',  
    'hostname' => 'localhost',  
    'username' => 'root',  
    'password' => '',  
    'database' => 'db_obat',  
    'dbdriver' => 'mysqli',  
    'dbprefix' => '',  
    'pconnect' => FALSE,  
    'db_debug' => (ENVIRONMENT !== 'production'),  
  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

3. Dibawah ini merupakan proses dari pembuatan file models baru di dalam folder models, yang mana models tersebut kami beri nama Models_obat.php dan class nya kami ubah sesuai dengan nama models yaitu Models_obat.



```
Run Terminal Help Models_obat.php - TUGAS_AKHIR_PWBS - Visual Studio Code
database.php Models_obat.php M X
Server > application > models > Models_obat.php > Models_obat
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4
5 class Models_obat extends CI_Model {
6 }
```

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



```
function get_data(){
    $this->db->select("id AS id_obat,
    kode AS kode_obat,
    nama AS nama_obat,
    jenis AS jenis_obat,
    harga AS harga_obat,
    stok AS stok_obat");
    $this->db->from('tb_obat');

    $this->db->order_by('kode','asc');
    $query = $this->db->get()->result();
    return $query;
}
```

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



```
function delete_data($token){
    $this->db->select("kode");
    $this->db->from("tb_obat");
    $this->db->where("kode = '$token'");
    $query = $this->db->get()->result();
    if(count($query) == 1){
        $this->db->where("kode = '$token'");
        $this->db->delete("tb_obat");
        $hasil = 1;
    }
    else{
        $hasil = 0;
    }
    return $hasil;
}
```

6. 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.

```
function save_data($kode,$nama,$jenis,$harga,$stok,$token){
    $this->db->select("kode");
    $this->db->from("tb_obat");
    $this->db->where("TO_BASE64(kode) = '$token'");
    $query = $this->db->get()->result();
    if(count($query) == 0){
        $data = array(
            "kode" => $kode,
            "nama" => $nama,
            "jenis" => $jenis,
            "harga" => $harga,
            "stok" => $stok,
        );
        $this->db->insert("tb_obat",$data);
        $hasil = 0;
    }
    else{
        $hasil = 1;
    }
    return $hasil;
}
```

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

```
function update_data($kode,$nama,$jenis,$harga,$stok,$token){
    $this->db->select("kode");
    $this->db->from("tb_obat");
    $this->db->where("TO_BASE64(kode) != '$token' AND kode = '$kode'");
    $query = $this->db->get()->result();
    if(count($query) == 0){
        $data = array(
            "kode" => $kode,
            "nama" => $nama,
            "jenis" => $jenis,
            "harga" => $harga,
            "stok" => $stok,
        );
        $this->db->where("TO_BASE64(kode) != '$token'");
        $this->db->update("tb_obat",$data);
        $hasil = 0;
    }
    else{
        $hasil = 1;
    }
    return $hasil;
}
```

8. Dibawah ini merupakan proses dari pembuatan file controllers baru di dalam folder controllers, yang mana controllers tersebut kami beri nama `obat.php` dan class nya kami ubah sesuai dengan nama models yaitu `obat`.

```
database.php  obat.php  Models_obat.php  Server.php
Server > application > controllers > obat.php > obat
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  require APPPATH."libraries/Server.php";
5  class obat extends Server {
6
```

9. 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_obat`, kemudian kita panggil function `get_data` yang ada di dalam `Models_obat`.

```
15 function service_get(){
16
17     $this->load->model("Models_obat","mdl",TRUE);
18     $hasil = $this->mdl->get_data();
19     $this->response(array("obat" => $hasil),200);
20 }
21
```

10. 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_obat`, kemudian kami mengambil parameter yang akan di isi seperti kode, nama, jenis, harga, stok.

```
23 function service_put(){
24
25     $this->load->model("Models_obat", "model", TRUE);
26
27     $data = array(
28         "kode" => $this->put("kode"),
29         "nama" => $this->put("nama"),
30         "jenis" => $this->put("jenis"),
31         "harga" => $this->put("harga"),
32         "stok" => $this->put("stok"),
33         "token" => base64_encode($this->put("token")),
34     );
35
36     $hasil = $this->model->update_data($data["kode"],$data["nama"],$data["jenis"],$data["harga"],$data["stok"],
37     $data["token"]);
38
39
40     if($hasil == 0){
41         $this->response(array("status"=>"Data obat Berhasil Diubah"),200);
42     }
43
44     else{
45         $this->response(array("status"=>"Data obat Gagal Diubah!"),200);
46     }
47 }
48
```

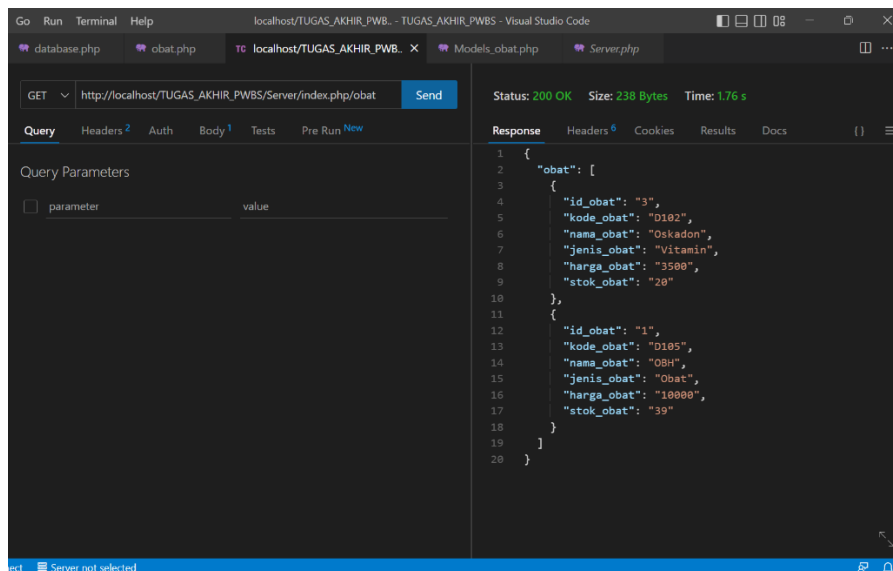
11. Setelah itu kami membuat fungsi post yang kami beri nama `service_post`, yang mana fungsi tersebut untuk menyimpan atau save data. Di dalam fungsi tersebut, kita panggil file models yaitu `Models_obat`, kemudian kami mengambil parameter yang akan di isi seperti kode, nama, jenis, harga, stok.

```
50 function service_post(){
51
52     $this->load->model("Models_obat", "model", TRUE);
53
54     $data = array(
55         "kode" => $this->post("kode"),
56         "nama" => $this->post("nama"),
57         "jenis" => $this->post("jenis"),
58         "harga" => $this->post("harga"),
59         "stok" => $this->post("stok"),
60         "token" => base64_encode($this->post("token")),
61     );
62
63     $hasil = $this->model->save_data($data["kode"],$data["nama"],$data["jenis"],$data["harga"],$data["stok"],
64     $data["token"]);
65
66     if($hasil == 0){
67         $this->response(array("status"=>"Data obat Berhasil Disimpan"),200);
68     }
69
70     else{
71         $this->response(array("status"=>"Data obat Gagal Disimpan!"),200);
72     }
73 }
74
```

12. 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_obat, proses penghapusan data dilakukan berdasarkan kode obat.

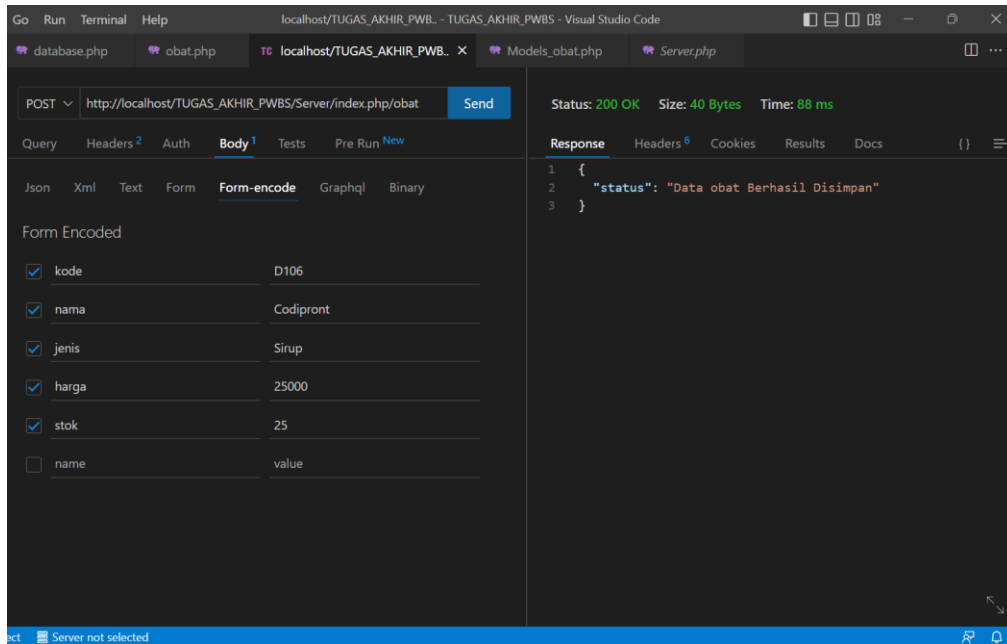
```
76 function service_delete()
77 {
78     $this->load->model("Models_obat", "model", TRUE);
79
80     $token = $this->delete("kode");
81
82     $hasil = $this->model->delete_data($token);
83
84     if($hasil == 1)
85     {
86
87         $this->response(array("status"=>"Data obat berhasil dihapus"),200);
88     }
89     else
90     {
91         $this->response(array("status" => "Data obat gagal dihapus !"), 200);
92     }
93 }
94
95 }
```

13. Kemudian kami melakukan pengecekan menggunakan Thunder Client, yang mana kami membuat request baru dengan menetikkan alamat “http://localhost/TUGAS_AKHIR_PWBS/Server/index.php/obat” dengan menggunakan fungsi GET untuk melihat data yang sebelumnya sudah di inputkan, seperti pada gambar dibawah ini.

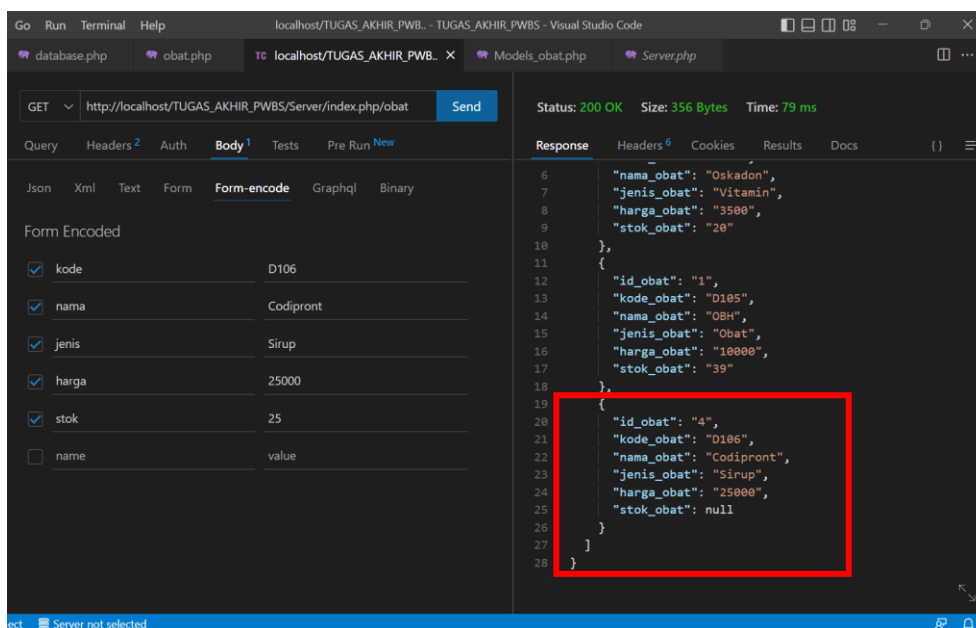


14. Setelah itu kami melakukan penambahan data menggunakan Thunder Client, yang mana kami membuat request baru dengan menetikkan alamat

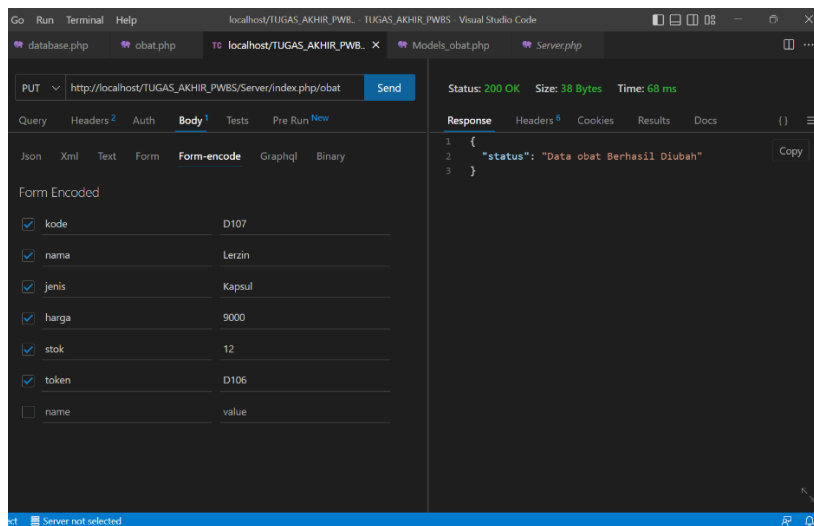
“http://localhost/TUGAS_AKHIR_PWBS/Server/index.php/obat” dengan menggunakan fungsi POST untuk menambahkan data, dengan cara mengisi field sesuai dengan tabel yang sudah dibuat seperti, kode, nama, jenis, harga, stok. Jika berhasil, maka keterangan nya yaitu “Data obat berhasil disimpan”.



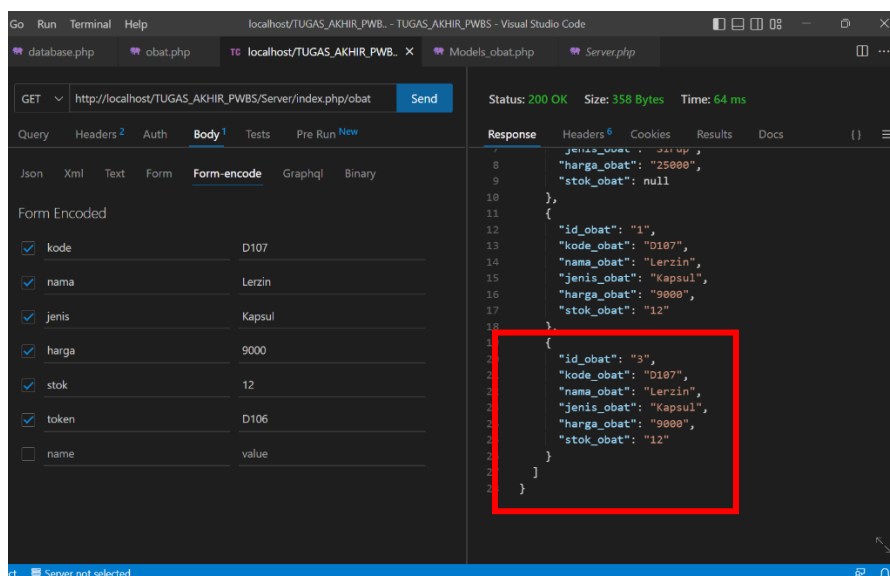
15. Lalu kami cek kembali ke bagian GET dengan mengetikkan alamat “http://localhost/TUGAS_AKHIR_PWBS/Server/index.php/obat” untuk mengecek apakah data sudah berhasil disimpan atau belum.



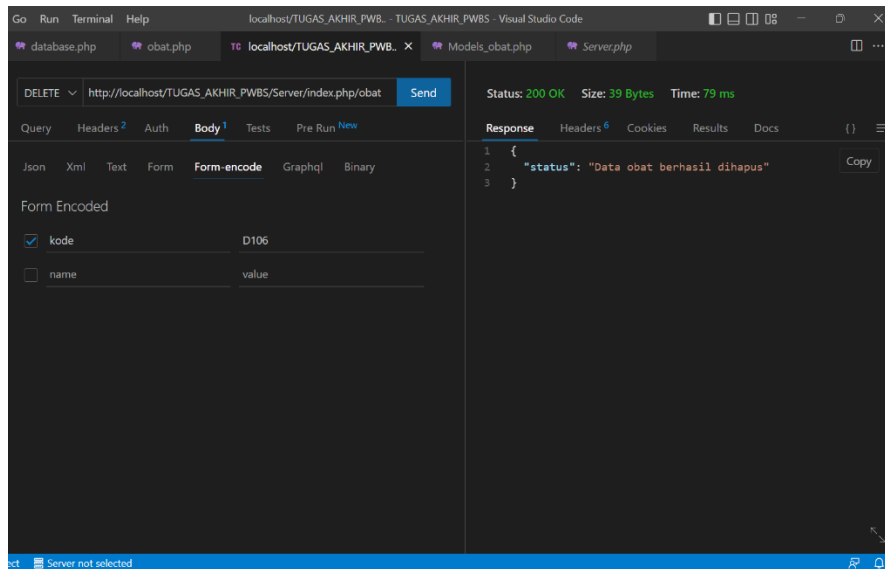
16. Kemudian kami melakukan perubahan data atau update data menggunakan Thunder Client, yang mana kami membuat request baru dengan mengetikkan alamat “http://localhost/TUGAS_AKHIR_PWBS/Server/index.php/obat” dengan menggunakan fungsi PUT untuk update data, dengan cara mengisi field sesuai dengan tabel yang sudah dibuat seperti, kode, nama, jenis, harga, stok. Lalu di field paling bawah kami tambahkan field token yang mana kami isi kode obat sebelumnya atau sebelum di update. Jika berhasil, maka keterangan nya yaitu “Data obat berhasil diubah”.



17. Setelah itu kami cek kembali ke bagian GET dengan mengetikkan alamat “http://localhost/TUGAS_AKHIR_PWBS/Server/index.php/obat” untuk mengecek apakah data sudah berhasil diubah atau belum.



18. Kemudian kami melakukan penghapusan data atau hapus data menggunakan Thunder Client, yang mana kami membuat request baru dengan mengetikkan alamat “http://localhost/TUGAS_AKHIR_PWBS/Server/index.php/obat” dengan menggunakan fungsi DELETE untuk hapus data, dengan cara mengisi field kode obat. Jika berhasil, maka keterangan nya yaitu “Data obat berhasil dihapus”.



19. Setelah itu kami cek kembali ke bagian GET dengan mengetikkan alamat “http://localhost/TUGAS_AKHIR_PWBS/Server/index.php/obat” untuk mengecek apakah data sudah berhasil dihapus atau belum.

