LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK LANJUT

2023

Prepared By:

Nama : Wahyudi

NIM : 210511076

Kelas : TIF21B (R2)

1. Mahasiswa.php

```
<?php
//Simpanlah dengan nama file : Mahasiswa.php
require_once 'database.php';
class Mahasiswa
{
   private $db;
   private $table = 'mahasiswa';
   public $nim = "";
   public $nama = "";
   public $jk = "";
   public $prodi = "";
   public function __construct(MySQLDatabase $db)
   {
      $this->db = $db;
   }
   public function get_all()
   {
      $query = "SELECT * FROM $this->table";
      $result_set = $this->db->query($query);
      return $result_set;
   }
   public function get_by_id(int $id)
   {
      $query = "SELECT * FROM $this->table WHERE id = $id";
      $result_set = $this->db->query($query);
      return $result_set;
   }
   public function get_by_nim(int $nim)
   {
      $query = "SELECT * FROM $this->table WHERE nim = $nim";
      $result_set = $this->db->query($query);
      return $result_set;
   }
   public function insert(): int
   {
      $query = "INSERT INTO $this->table (`nim`,`nama`,`jk`,`prodi`) VALUES
('$this->nim','$this->nama','$this->jk','$this->prodi')";
      $this->db->query($query);
      return $this->db->insert_id();
   }
   public function update(int $id): int
   {
      $query = "UPDATE $this->table SET nim = '$this->nim', nama = '$this-
>nama', jk = '$this->jk', prodi = '$this->prodi'
      WHERE id = $id";
      $this->db->query($query);
      return $this->db->affected_rows();
   }
   public function update_by_nim($nim): int
   {
      $query = "UPDATE $this->table SET nim = '$this->nim', nama = '$this-
>nama', jk = '$this->jk', prodi = '$this->prodi'
```

```
            WHERE nim = $nim";
            $this->db->query($query);
            return $this->db->affected_rows();
        }
        public function delete(int $id): int
        {
            $query = "DELETE FROM $this->table WHERE id = $id";
            $this->db->query($query);
            return $this->db->affected_rows();
        }
        public function delete_by_nim($nim): int
        {
            $query = "DELETE FROM $this->table WHERE nim = $nim";
            $this->db->query($query);
            return $this->db->affected_rows();
        }
    }
?>
```

2. Mahasiswa_api.php

```
    <?php
    require_once 'database.php';
    require_once 'Mahasiswa.php';
    $db = new MySQLDatabase();
    $mahasiswa = new Mahasiswa($db);
    $id=0;
    $nim=0;
    // Check the HTTP request method
    $method = $_SERVER['REQUEST_METHOD'];
    // Handle the different HTTP methods
    switch ($method) {
        case 'GET':
            if(isset($_GET['id'])){
                $id = $_GET['id'];
            }
            if(isset($_GET['nim'])){
                $nim = $_GET['nim'];
            }
            if($id>0){
                $result = $mahasiswa->get_by_id($id);
            }elseif($nim>0){
                $result = $mahasiswa->get_by_nim($nim);
            } else {
                $result = $mahasiswa->get_all();
            }

            $val = array();
            while ($row = $result->fetch_assoc()) {
                $val[] = $row;
            }

            header('Content-Type: application/json');
            echo json_encode($val);
            break;
```

```
case 'POST':
    // Add a new mahasiswa
    $mahasiswa->nim = $_POST['nim'];
    $mahasiswa->nama = $_POST['nama'];
    $mahasiswa->jk = $_POST['jk'];
    $mahasiswa->prodi = $_POST['prodi'];

    $mahasiswa->insert();
    $a = $db->affected_rows();
    if($a>0){
        $data['status']='success';
        $data['message']='Data Mahasiswa created successfully.';
    } else {
        $data['status']='failed';
        $data['message']='Data Mahasiswa not created.';
    }
    header('Content-Type: application/json');
    echo json_encode($data);
    break;
case 'PUT':
    // Update an existing data
    $_PUT = [];
    if(isset($_GET['id'])){
        $id = $_GET['id'];
    }
    if(isset($_GET['nim'])){
        $nim = $_GET['nim'];
    }
    parse_str(file_get_contents("php://input"), $_PUT);
    $mahasiswa->nim = $_PUT['nim'];
    $mahasiswa->nama = $_PUT['nama'];
    $mahasiswa->jk = $_PUT['jk'];
    $mahasiswa->prodi = $_PUT['prodi'];
    if($id>0){
        $mahasiswa->update($id);
    }elseif($nim<>""){
        $mahasiswa->update_by_nim($nim);
    } else {

    }

    $a = $db->affected_rows();
    if($a>0){
        $data['status']='success';
        $data['message']='Data Mahasiswa updated successfully.';
    } else {
        $data['status']='failed';
        $data['message']='Data Mahasiswa update failed.';
    }
    header('Content-Type: application/json');
    echo json_encode($data);
    break;
case 'DELETE':
    // Delete a user
```

```
            if(isset($_GET['id'])){
                $id = $_GET['id'];
            }
            if(isset($_GET['nim'])){
                $nim = $_GET['nim'];
            }
            if($id>0){
                $mahasiswa->delete($id);
            }elseif($nim>0){
                $mahasiswa->delete_by_nim($nim);
            } else {

            }

            $a = $db->affected_rows();
            if($a>0){
                $data['status']='success';
                $data['message']='Data Mahasiswa deleted successfully.';
            } else {
                $data['status']='failed';
                $data['message']='Data Mahasiswa delete failed.';
            }
            header('Content-Type: application/json');
            echo json_encode($data);
            break;
        default:
            header("HTTP/1.0 405 Method Not Allowed");
            break;
        }
    $db->close()
?>
```

3. Mahasiswa.py

```
    import requests
    import json
    class Mahasiswa:
        def __init__(self):
            self.__id=None
            self.__nim = None
            self.__nama = None
            self.__jk = None
            self.__prodi = None
            self.__url = "http://localhost/myakademik/mahasiswa_api.php"

        @property
        def id(self):
            return self.__id
        @property
        def nim(self):
            return self.__nim

        @nim.setter
        def nim(self, value):
```

```
        self.__nim = value
    @property
    def nama(self):
        return self.__nama

    @nama.setter
    def nama(self, value):
        self.__nama = value
    @property
    def jk(self):
        return self.__jk

    @jk.setter
    def jk(self, value):
        self.__jk = value
    @property
    def prodi(self):
        return self.__prodi

    @prodi.setter
    def prodi(self, value):
        self.__prodi = value
    def get_all(self):
        payload ={}
        headers = {'Content-Type': 'application/json'}
        response = requests.get(self.__url, json=payload, headers=headers)
        return response.text
    def get_by_nim(self, nim):
        url = self.__url+"?nim="+nim
        payload = {}
        headers = {'Content-Type': 'application/json'}
        response = requests.get(url, json=payload, headers=headers)
        data = json.loads(response.text)
        for item in data:
            self.__id = item['id']
            self.__nim = item['nim']
            self.__nama = item['nama']
            self.__jk = item['jk']
            self.__prodi = item['prodi']
        return data
    def simpan(self):
        payload = {
            "nim":self.__nim,
            "nama":self.__nama,
            "jk":self.__jk,
            "prodi":self.__prodi
            }
        headers = {'Content-Type': 'application/x-www-form-urlencoded'}
        response = requests.post(self.__url, data=payload, headers=headers)
        return response.text
    def update_by_nim(self, nim):
        url = self.__url+"?nim="+nim
        payload = {
            "nim":self.__nim,
            "nama":self.__nama,
```

```
                "jk":self.__jk,
                "prodi":self.__prodi
                }
            headers = {'Content-Type': 'application/x-www-form-urlencoded'}
            response = requests.put(url, data=payload, headers=headers)
            return response.text
        def delete_by_nim(self,nim):
            url = self.__url+"?nim="+nim
            headers = {'Content-Type': 'application/json'}
            payload={}
            response = requests.delete(url, json=payload, headers=headers)
    return response.text
```

4. FrmMahasiswa.py

```
import tkinter as tk
import json
from tkinter import Frame,Label,Entry,Button,Radiobutton,ttk,VERTICAL,YES,BOTH,END,Tk,W,StringVar,messagebox
from Mahasiswa import *
class FrmMahasiswa:

    def __init__(self, parent, title):
        self.parent = parent
        self.parent.geometry("450x450")
        self.parent.title(title)
        self.parent.protocol("WM_DELETE_WINDOW", self.onKeluar)
        self.ditemukan = None
        self.aturKomponen()
        self.onReload()

    def aturKomponen(self):
        mainFrame = Frame(self.parent, bd=10)
        mainFrame.pack(fill=BOTH, expand=YES)
        Label(mainFrame, text='NIM:').grid(row=0, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text='NAMA:').grid(row=1, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text='JK:').grid(row=2, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text='PRODI:').grid(row=3, column=0,
            sticky=W, padx=5, pady=5)
        # Textbox
        self.txtNim = Entry(mainFrame)
        self.txtNim.grid(row=0, column=1, padx=5, pady=5)
        self.txtNim.bind("<Return>",self.onCari) # menambahkan event Enter key
        # Textbox
        self.txtNama = Entry(mainFrame)
        self.txtNama.grid(row=1, column=1, padx=5, pady=5)
        # Combo Box
        self.txtJk = StringVar()
        Cbo_jk = ttk.Combobox(mainFrame, width = 17, textvariable = self.txtJk)
        Cbo_jk.grid(row=2, column=1, padx=5, pady=5)
        # Adding jk combobox drop down list
```

```
        Cbo_jk['values'] = ('L','P')
        Cbo_jk.current()
        # Combo Box
        self.txtProdi = StringVar()
        Cbo_prodi = ttk.Combobox(mainFrame, width = 17, textvariable =
self.txtProdi)
        Cbo_prodi.grid(row=3, column=1, padx=5, pady=5)
        # Adding prodi combobox drop down list
        Cbo_prodi['values'] = ('IND','TIF','PET')
        Cbo_prodi.current()
        # Button
        self.btnSimpan = Button(mainFrame, text='Simpan', command=self.onSimpan,
width=10)
        self.btnSimpan.grid(row=0, column=3, padx=5, pady=5)
        self.btnClear = Button(mainFrame, text='Clear', command=self.onClear,
width=10)
        self.btnClear.grid(row=1, column=3, padx=5, pady=5)
        self.btnHapus = Button(mainFrame, text='Hapus', command=self.onDelete,
width=10)
        self.btnHapus.grid(row=2, column=3, padx=5, pady=5)
        # define columns
        columns = ('id','nim','nama','jk','prodi')
        self.tree = ttk.Treeview(mainFrame, columns=columns, show='headings')
        # define headings
        self.tree.heading('id', text='ID')
        self.tree.column('id', width="30")
        self.tree.heading('nim', text='NIM')
        self.tree.column('nim', width="80")
        self.tree.heading('nama', text='NAMA')
        self.tree.column('nama', width="200")
        self.tree.heading('jk', text='JK')
        self.tree.column('jk', width="30")
        self.tree.heading('prodi', text='PRODI')
        self.tree.column('prodi', width="60")
        # set tree position
        self.tree.place(x=0, y=200)

    def onClear(self, event=None):
        self.txtNim.delete(0,END)
        self.txtNim.insert(END,"")
        self.txtNama.delete(0,END)
        self.txtNama.insert(END,"")
        self.txtJk.set("")
        self.txtProdi.set("")
        self.btnSimpan.config(text="Simpan")
        self.onReload()
        self.ditemukan = False

    def onReload(self, event=None):
        # get data mahasiswa
        obj = Mahasiswa()
        result = obj.get_all()
        parsed_data = json.loads(result)
        for item in self.tree.get_children():
            self.tree.delete(item)
```

```
            for i, d in enumerate(parsed_data):
                self.tree.insert("", i, text="Item {}".format(i),
values=(d["id"],d["nim"],d["nama"],d["jk"],d["prodi"]))
        def onCari(self, event=None):
            nim = self.txtNim.get()
            obj = Mahasiswa()
            a = obj.get_by_nim(nim)
            if(len(a)>0):
                self.TampilkanData()
                self.ditemukan = True
            else:
                self.ditemukan = False
                messagebox.showinfo("showinfo", "Data Tidak Ditemukan")
        def TampilkanData(self, event=None):
            nim = self.txtNim.get()
            obj = Mahasiswa()
            res = obj.get_by_nim(nim)
            self.txtNim.delete(0,END)
            self.txtNim.insert(END,obj.nim)
            self.txtNama.delete(0,END)
            self.txtNama.insert(END,obj.nama)
            self.txtJk.set(obj.jk)
            self.txtProdi.set(obj.prodi)
            self.btnSimpan.config(text="Update")

        def onSimpan(self, event=None):
            # get the data from input
            nim = self.txtNim.get()
            nama = self.txtNama.get()
            jk = self.txtJk.get()
            prodi = self.txtProdi.get()
            # create new Object
            obj = Mahasiswa()
            obj.nim = nim
            obj.nama = nama
            obj.jk = jk
            obj.prodi = prodi
            if(self.ditemukan==False):
                # save the record
                res = obj.simpan()
            else:
                # update the record
                res = obj.update_by_nim(nim)
            # read data in json format
            data = json.loads(res)
            status = data["status"]
            msg = data["message"]
            # display json data into messagebox
            messagebox.showinfo("showinfo", status+', '+msg)
            #clear the form input
            self.onClear()
        def onDelete(self, event=None):
            nim = self.txtNim.get()
            obj = Mahasiswa()
```

```
            obj.nim = nim
            if(self.ditemukan==True):
                res = obj.delete_by_nim(nim)
            else:
                messagebox.showinfo("showinfo", "Data harus ditemukan dulu sebelum
        dihapus")

            # read data in json format
            data = json.loads(res)
            status = data["status"]
            msg = data["message"]

            # display json data into messagebox
            messagebox.showinfo("showinfo", status+', '+msg)

            self.onClear()

        def onKeluar(self, event=None):
            # memberikan perintah menutup aplikasi
            self.parent.destroy()
    if __name__ == '__main__':
        root2 = tk.Tk()
        aplikasi = FrmMahasiswa(root2, "Aplikasi Data Mahasiswa")
root2.mainloop()
```

5. SS Tampilan Form