

LAPORAN PRAKTIKUM
PEMROGRAMAN BERBASIS WEB
“CRUD”



Disusun Oleh :

Nama : I Gusti Gede Ngurah Adryan Pratama Putra Bueka

Nim : 1808561045

Kelas : B

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS UDAYANA
BADUNG
2020

KATA PENGANTAR

Puji dan syukur ke hadirat Tuhan Yang Maha Kuasa atas segala rahmat yang diberikan-Nya sehingga tugas Laporan Praktikum 8 yang berjudul "CRUD" ini dapat saya selesaikan. Laporan ini saya buat sebagai kewajiban untuk memenuhi tugas. Dalam kesempatan ini, penulis menghaturkan terimakasih yang dalam kepada semua pihak yang telah membantu menyumbangkan ide dan pikiran mereka demi terwujudnya makalah ini. Akhirnya saran dan kritik pembaca yang dimaksud untuk mewujudkan kesempurnaan makalah ini penulis sangat hargai.

Gianyar, 8 Desember 2020

Edo Krishnanda Aditya

DAFTAR ISI

Kata Pengantar	ii
Daftar Isi.....	iii
BAB I Pendahuluan.....	1
1.1 Latar Belakang	1
1.2 Tujuan	1
1.3 Manfaat	1
BAB II Kajian Pustaka.....	2
BAB III Pembahasan.....	3
BAB IV Penutup	8
4.1 Kesimpulan	8
4.2 Saran.....	8
Daftar Pustaka	9

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam pemrograman komputer, buat, baca, update, dan hapus (sebagai singkatan CRUD) adalah empat fungsi dasar penyimpanan persisten. Kata-kata alternatif kadang-kadang digunakan saat menentukan empat fungsi dasar CRUD, seperti mengambil alih-alih membaca, memodifikasi, bukan memperbarui, atau menghancurkan alih-alih menghapus. CRUD juga terkadang digunakan untuk menggambarkan konvensi antarmuka pengguna yang memudahkan melihat, mencari, dan mengubah informasi; sering menggunakan formulir berbasis komputer dan laporan. Istilah ini kemungkinan dipopulerkan pertama kali oleh James Martin dalam bukunya 1983 the Managing Data-base Environment. Akronimnya dapat diperluas ke CRUD untuk mencakup daftar kumpulan data besar yang membawa kompleksitas tambahan seperti pagination saat kumpulan data terlalu besar untuk disimpan dengan mudah di memori.

1.2 Tujuan

Tujuan dari makalah ini adalah :

- Memahami CRUD

1.3 Manfaat

Manfaat dari makalah ini antara lain :

- Pembaca mengerti dan memahami CRUD.

BAB II

KAJIAN PUSTAKA

Sistem informasi yang baik akan menyimpan dengan baik setiap proses transaksi yang terjadi ke dalam database. Hal ini menjadi hal yang krusial, karena ini menjadi titik vital dari perusahaan, karena setiap transaksi perusahaan tercatat dengan baik. Untuk menunjang itu diperlukannya penggunaan database yang baik.

Dalam proses database saat ini sering menggunakan database relational atau RDBMS. Proses database ini menggunakan query untuk memanggil data dari database. Sebuah sistem informasi akan sering melakukan proses query ke dalam database, hal ini untuk mendapatkan data yang sesuai dan dalam keadaan real. Untuk itu diperlukan sebuah proses query database yang baik.

Proses CRUD ini akan menjadi sangat penting karena, akan memberatkan sistem jika proses berlangsung secara serampangan. Salah satu prosesnya adalah teknik normalisasi dari database, teknik ini sudah sering digunakan dan menjadi bagian penting. Salah satu teknik penting di dalam proses CRUD ke database, kita bisa menggunakan teknik ORM atau Object Relational Mapping, terlebih teknik ini akan sangat bekerja optimal dibawah pengaruh Object Oriented.

CRUD merupakan empat fungsi dasar dari penyimpanan persisten.

- C (Create): berarti membuat sebuah data baru, contoh kita sedang melakukan registrasi disebuah web itu sudah merupakan Create dari CRUD karena kita membuat dan menyimpan data registrasi ke database.
- R (Read): Membaca atau menampilkan suatu data yang tadinya berada didatabase MySQL misalnya, kemudian ditampilkan di WEB menggunakan bahasa pemrograman Php.
- U (Update): adalah mengedit sebuah data dari database yang kemudian di edit menggunakan bahasa pemrograman Php berupa WEB. Contoh edit profil facebook.
- D (Delete): Fungsinya hampir sama dengan Update akan tetapi proses ini adalah untuk melakukan penghapusan data di database melalui bahasa Php. Contoh pada sebuah blog terkadang ada komentar,

kemudian kita hapus komentar tersebut, nah itu sudah termasuk proses delete dalam CRUD.

User Interface

CRUD juga relevan di tingkat user interface dari sebagian besar aplikasi. Misalnya, dalam buku alamat software, unit penyimpanan dasar adalah seorang individu kontak entri. Sebagai minimal, perangkat lunak harus memungkinkan pengguna untuk

- Membuat atau menambah entri baru.
- Baca, mengambil, pencarian, atau melihat entri yang ada.
- Memperbarui atau mengedit entri yang ada.
- Hapus / menonaktifkan / menghapus entri yang ada.

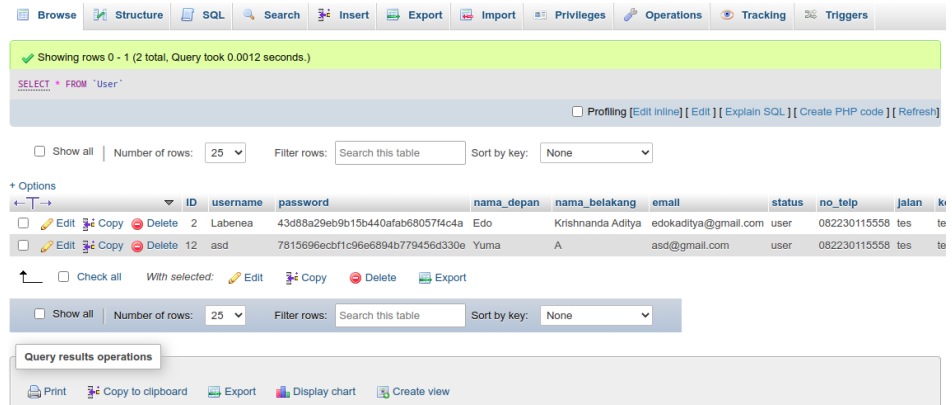
Tanpa empat operasi setidaknya ini, perangkat lunak tidak dapat dianggap lengkap. Karena operasi ini sangat mendasar, mereka sering didokumentasikan dan dijelaskan di bawah satu judul yang luas, seperti "manajemen kontak", "manajemen konten" atau "pemeliharaan kontak" (atau "manajemen dokumen" secara umum, tergantung pada unit penyimpanan dasar untuk aplikasi tertentu).

BAB III

PEMBAHASAN

1. Persiapan

Siapkan Database dan tabel yang akan digunakan



2. Penjelasan Code

a. Koneksi Database

Disini kami menggunakan PDO untuk koneksi PHP terhadap mysql karena kami menggunakan konsep OOP PHP

```
class Database {  
    private $host = 'localhost';  
    private $dbname = 'petanikita';  
    private $username = 'root';  
    private $password = '';  
    private $error;  
    private $pdo;  
    private $stmt;  
  
    public function __construct(){  
        $dsn = 'mysql:host=' . $this->host . ';dbname=' . $this->dbname;  
        try{  
            $this->pdo = new PDO($dsn,$this->username,$this->password);  
            $this->pdo->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE,PDO::FETCH_OBJ);  
        }catch (PDOException $e){  
            $this->error = $e->getMessage();  
            return $this->error;  
        }  
    }  
}
```

Pertama-tama kita deklarasikan host, database name, username dan password. Lalu kita akan menggunakan try/catch untuk mencoba connect terhadap database.

b. Create

Pada proses create kami akan menggunakan proses registasi user sebagai contoh .

Disini
adalah
code
untuk

```
<?php
require(APPROOT . "/Views/include/header.php");
$error = Message();
?>

<?php if ($error != null) : ?>
<div class="container-fluid" style="max-width:90%;>
<div class="alert alert-danger" role="alert">
<php echo $error; ?>
</div>
</div>

<?php endif ?>

<div class="container-fluid regist" style="max-width: 90%;>
<div class="row mt-5 mb-5 justify-content-center">
<div class="col-10">
<div class="card">
<div class="card-body p-5">
<div class="row justify-content-center">
<a href="#<?php echo BASEURL ?>">
<h1 class="title col-12 mt-1">
<span class="green">Petani</span><span class="orange">Kita</span>
</h1>
</a>
</div>
<div class="row">
<div class="col-7">
<form class="row justify-content-center" action="#<?php echo BASEURL ?>register" method="post">
<div class="col-12">
<div class="row mb-3">
<label for="name" class="form-label">Name</label>
<div class="col-6">
<input type="text" name="nama_depan" id="nama_depan" class="form-control form-control-sm" />
<div id="" class="form-text">Nama Depan</div>
</div>
<div class="col-6">
<input type="text" name="nama_belakang" id="nama_belakang" class="form-control form-control-sm" />
<div id="" class="form-text">Nama Belakang</div>
</div>
</div>
<div class="row mb-3">
<div class="col-6">
<label for="username" class="form-label">Username</label>
<input type="text" name="username" id="username" class="form-control form-control-sm" />
</div>
<div class="col-6">
<label for="email" class="form-label">Email</label>
<input type="email" name="email" id="email" class="form-control form-control-sm" />
</div>
</div>
<div class="row mb-3">
<div class="col-6">
<label for="password" class="form-label">Password</label>
<input type="password" name="password" id="password" class="form-control form-control-sm" />
</div>
<div class="col-6">
<label for="password" class="form-label">Confirm Password</label>
<input type="password" name="cpassword" id="cpassword" class="form-control form-control-sm" />
</div>
</div>
<div class="row mb-3">
<div class="col-12">
<label for="no_telp" class="form-label">No Telp</label>
<input type="text" name="no_telp" id="no_telp" class="form-control form-control-sm" />
</div>
</div>
<div class="row mb-3">
<div class="col-12">
<label for="jalan" class="form-label">Jalan</label>
<input type="text" name="jalan" id="jalan" class="form-control form-control-sm" />
</div>
</div>
<div class="row mb-3">
<div class="col-6">
<label for="kota" class="form-label">Kota</label>
<input type="text" name="kota" id="kota" class="form-control form-control-sm" />
</div>
<div class="col-6">
<label for="kabupaten" class="form-label">Kabupaten</label>
<input type="text" name="kabupaten" id="kabupaten" class="form-control form-control-sm" />
</div>
</div>
<div class="row mb-3 justify-content-end">
<div class="col-12">
<div class="d-grid gap-2 col-12 mx-auto mt-4">
<button class="btn btn-primary" id="registerBtn" type="submit">
Sign Up
</button>
</div>
</div>
</div>
</div>
<div class="col-5 bg-card"></div>
</div>
</div>
</div>
</div>
</div>

<?php
require(APPROOT . "/Views/include/footer.php");
?>
```

tampilan registernya dimana akan ada form dengan method POST dan Action ke register dan setelah disumbit akan diproses.

```
$_POST = filter_input_array(INPUT_POST, FILTER_SANITIZE_STRING);
if($_SERVER['REQUEST_METHOD'] == 'POST'){

    $data = [
        'username' => trim($_POST['username']),
        'password' => trim($_POST['password']),
        'nama_depan' => trim($_POST['nama_depan']),
        'nama_belakang' => trim($_POST['nama_belakang']),
        'email' => trim($_POST['email']),
        'status' => 'user',
        'notelp' => trim($_POST['no_telp']),
        'jalan' => trim($_POST['jalan']),
        'kota' => trim($_POST['kota']),
        'kabupaten' => trim($_POST['kabupaten']),
    ];

    foreach($data as $dat){
        if(empty($dat)){
            $SESSION['message'] = 'All Field Must Be Filled';
            return var_dump($dat);
            return header('location: '.URLROOT.'/register');
        }
    }

    $data['password'] = md5($data['password']);
    if($this->userModel->register($data){
        $loggedInUser = $this->userModel->login($data);
        if($loggedInUser){
            $this->createUserSession($loggedInUser);
        }
        $SESSION['message'] = 'Register Successful';
        return header('location: '.URLROOT.'/');
    }
}

else{
    $data = [
        'title' => 'Register',
    ];
    return $this->createView('Register', $data);
}
```


Disini data akan diterima dengan `$_POST['nama data']` dan akan dimasukan kedalam array data dan akan memanggil fungsi register dari userModel

```
public function register($data){
    $this->db->query('INSERT INTO User (username,password,nama_depan,nama_belakang,email,status,no_telp,jalan,kota,kabupaten) VALUES(:username, :password, :nama_depan, :nama_belakang, :email, :status, :no_telp, :jalan, :kota, :kabupaten)');

    $this->db->bind(':username',$data['username']);
    $this->db->bind(':password',$data['password']);
    $this->db->bind(':nama_depan',$data['nama_depan']);
    $this->db->bind(':nama_belakang',$data['nama_belakang']);
    $this->db->bind(':email',$data['email']);
    $this->db->bind(':status',$data['status']);
    $this->db->bind(':no_telp',$data['notelp']);
    $this->db->bind(':jalan',$data['jalan']);
    $this->db->bind(':kota',$data['kota']);
    $this->db->bind(':kabupaten',$data['kabupaten']);

    if($this->db->execute()){
        return true;
    }else{
        return false;
    }
}
```

Disini `db->query` adalah untuk memasukan sql kita yaitu disini adalah insert dan akan di execute oleh `db->execute`.

c. Read

Untuk contoh read disini adalah untuk melihat profil.

```
public function profile(){
    $this->db->query('SELECT * FROM User WHERE ID = :id');
    $this->db->bind(':id',$SESSION['user_id']);
    $row = $this->db->singleRow();
    if($row != null){
        return $row;
    }else{
        return false;
    }
}
```

d. Update

Untuk contoh update disini adalah mengganti nama profile

```
public function editNama($data){
    $this->db->query('UPDATE User SET nama_depan = :nama_depan , nama_belakang = :nama_belakang WHERE ID = :id');
    $this->db->bind(':id',$SESSION['user_id']);
    $this->db->bind(':nama_depan',$data['nama_depan']);
    $this->db->bind(':nama_belakang',$data['nama_belakang']);
    if($this->db->execute()){
        return true;
    }else{
        return false;
    }
}
```

e. Delete

Dan untuk delete adalah melakukan delete User

```
public function deleteUser(){
    $this->db->query('DELETE FROM User WHERE ID = :id');
    $this->db->bind(':id',$SESSION['user_id']);
    if($this->db->execute()){
        return true;
    }else{
        return false;
    }
}
```

BAB IV

PENUTUP

4.1. Simpulan

Dalam pemrograman komputer, buat, baca, update, dan hapus (sebagai singkatan CRUD) adalah empat fungsi dasar penyimpanan persisten. Kata-kata alternatif kadang-kadang digunakan saat menentukan empat fungsi dasar CRUD, seperti mengambil alih-alih membaca, memodifikasi, bukan memperbarui, atau menghancurkan alih-alih menghapus. CRUD juga terkadang digunakan untuk menggambarkan konvensi antarmuka pengguna yang memudahkan melihat, mencari, dan mengubah informasi.

4.2. Saran

Saya sadar makalah ini masih banyak menemui kesalahan atau kurang lengkapnya informasi mengenai CRUD dalam makalah ini. Maka dari itu, saya memohon maaf. Besar harapan saya untuk mengembangkan kembali pengetahuan saya. Saya juga mengharapkan saran-saran yang membangun dari pembaca.

DAFTAR PUSTAKA

Indah, 2015. *Indah Uminaroh's Blog*. [Online]

Available at: <http://iuminaroh.blogspot.com/2017/10/pengertian-crud.html>

[Accessed 8 12 2020].

J, F., 2017. *Teknologi Informasi*. [Online]

Available at: <http://fadilajunitan.blogspot.com/2017/01/pengenalan-crud-create-read-update.html>

[Accessed 8 12 2020].

Rahayu, V., 2017. *VIVINRP*. [Online]

Available at: <http://vivinrahayu24.blogspot.com/2017/10/pengenalan-tentang-crud.html>

[Accessed 8 12 2020].