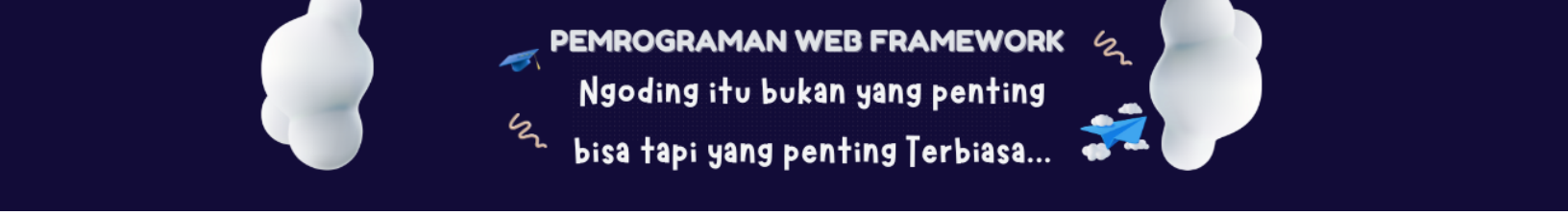
PENGEMBANGAN DAN PENYELENGGARAAN
PEMBELAJARAN DIGITAL (P3D)

Modul Pembelajaran

SISTEM INFORMASI

Pemrgoraman Web Framework

Disusun oleh : Wicaksono Yuli Sulistyo



Modul Ajar: CRUD (Create, Read, Update, Delete) pada CodeIgniter 3

Pendahuluan

Salah satu fitur inti dari aplikasi berbasis database adalah kemampuan untuk melakukan operasi **CRUD**: **Create (Insert)**, **Read (Select)**, **Update**, dan **Delete**. Operasi CRUD memungkinkan aplikasi untuk menambahkan, menampilkan, mengubah, dan menghapus data dari database. Pada framework **CodeIgniter 3**, operasi CRUD dapat dilakukan dengan mudah berkat bantuan library database dan Active Record yang sudah terintegrasi dalam framework ini.

Pada modul ini, kita akan membahas **teori CRUD secara mendalam**, dengan penjelasan tentang cara mengimplementasikan masing-masing operasi dalam **CodeIgniter 3**. Selain itu, setiap bagian akan dilengkapi dengan contoh kode dan penjelasan cara kerjanya.

---

1. Teori CRUD dalam Pemrograman Web

CRUD adalah singkatan dari:

- **Create (Insert)**: Operasi untuk menambahkan data baru ke dalam database.
- **Read (Select)**: Operasi untuk mengambil dan menampilkan data dari database.
- **Update**: Operasi untuk memperbarui data yang sudah ada di dalam database.
- **Delete**: Operasi untuk menghapus data dari database.

Setiap operasi ini sangat umum dalam aplikasi web berbasis database, seperti aplikasi manajemen, e-commerce, forum, dan sebagainya. CRUD memberikan interaksi dasar antara aplikasi dan database yang memungkinkan manipulasi data.

---

2. Struktur Database untuk CRUD

Untuk melakukan operasi CRUD, Anda memerlukan **struktur database** yang mencerminkan data yang ingin dikelola. Sebagai contoh, kita akan menggunakan tabel **produk** yang memiliki atribut seperti **id_produk**, **nama_produk**, **harga_produk**, dan **stok_produk**.

```
sql

CREATE TABLE produk (
    id_produk INT AUTO_INCREMENT PRIMARY KEY,
    nama_produk VARCHAR(100),
    harga_produk DECIMAL(10, 2),
    stok_produk INT
```

```
);
```

---

3. Operasi CREATE (INSERT)

Teori INSERT:

Operasi **INSERT** adalah proses untuk menambahkan data baru ke dalam database. Ketika sebuah entitas baru seperti produk, user, atau kategori ingin disimpan, kita menggunakan perintah **INSERT INTO**. Di CodeIgniter 3, operasi insert dilakukan dengan metode **insert()** dari library database bawaan CodeIgniter.

Contoh Penggunaan:

Misalkan kita ingin menambahkan produk baru ke dalam tabel **produk**.

```
php

class Produk_model extends CI_Model {
    public function tambahProduk($data) {
        // Insert data ke dalam tabel 'produk'
        return $this->db->insert('produk', $data);
    }
}
```

Pada model di atas, fungsi `tambahProduk()` menerima parameter berupa array `$data` yang berisi informasi produk, kemudian fungsi tersebut memanggil metode **insert()** untuk menambahkan data ke tabel **produk**.

Contoh Penggunaan di Controller:

```
php

class Produk extends CI_Controller {
    public function tambah() {
        // Data produk yang akan diinsert
        $data = [
            'nama_produk' => $this->input->post('nama_produk'),
            'harga_produk' => $this->input->post('harga_produk'),
            'stok_produk' => $this->input->post('stok_produk')
        ];

        // Memanggil model untuk insert data
        $this->Produk_model->tambahProduk($data);

        // Menampilkan pesan sukses dan redirect ke halaman produk
        $this->session->set_flashdata('message', 'Produk berhasil
ditambahkan!');
        redirect('produk');
    }
}
```

Penjelasan Teori:

- **$this->input->post()**: Digunakan untuk menangkap input dari form HTML.
- **$this->db->insert()**: Fungsi bawaan CodeIgniter yang digunakan untuk melakukan **INSERT** ke dalam tabel database.

---

4. Operasi READ (SELECT)

Teori SELECT:

Operasi **SELECT** adalah proses untuk mengambil data dari database. Di CodeIgniter 3, kita bisa menggunakan metode **get()**, **get_where()**, dan **query()** untuk melakukan pengambilan data dari tabel.

Contoh Penggunaan:

Misalkan kita ingin menampilkan semua produk yang ada di database.

```
php

class Produk_model extends CI_Model {
    public function getAllProduk() {
        // Mengambil semua data produk
        return $this->db->get('produk')->result_array();
    }

    public function getProdukById($id) {
        // Mengambil data produk berdasarkan ID
        return $this->db->get_where('produk', ['id_produk' => $id])-
>row_array();
    }
}
```

Pada kode di atas:

- **getAllProduk()**: Mengambil semua data dari tabel **produk** dan mengembalikannya sebagai array.
- **getProdukById($id)**: Mengambil data produk berdasarkan **id_produk** yang diberikan.

Contoh Penggunaan di Controller:

```
php

class Produk extends CI_Controller {
    public function index() {
        $data['produk'] = $this->Produk_model->getAllProduk();
        $this->load->view('produk/index', $data);
    }
```

```
    public function detail($id) {
        $data['produk'] = $this->Produk_model->getProdukById($id);
        $this->load->view('produk/detail', $data);
    }
}
```

Penjelasan Teori:

- **$this->db->get()**: Metode ini digunakan untuk mengambil data dari tabel.
- **$this->db->get_where()**: Digunakan untuk mengambil data dengan kondisi tertentu (misalnya berdasarkan ID).

---

5. Operasi UPDATE

Teori UPDATE:

Operasi **UPDATE** adalah proses untuk memperbarui data yang sudah ada di dalam database. Misalnya, jika Anda ingin mengubah harga atau stok produk, Anda bisa menggunakan perintah **UPDATE**. Di CodeIgniter, metode **update()** digunakan untuk memperbarui data.

Contoh Penggunaan:

```
php

class Produk_model extends CI_Model {
    public function updateProduk($data, $id) {
        $this->db->where('id_produk', $id);
        return $this->db->update('produk', $data);
    }
}
```

Pada contoh di atas, fungsi `updateProduk()` digunakan untuk memperbarui data produk berdasarkan **id_produk**.

Contoh Penggunaan di Controller:

```
php

class Produk extends CI_Controller {
    public function edit($id) {
        $data['produk'] = $this->Produk_model->getProdukById($id);

        $this->form_validation->set_rules('nama_produk', 'Nama Produk',
'required');
        $this->form_validation->set_rules('harga_produk', 'Harga Produk',
'required');
        $this->form_validation->set_rules('stok_produk', 'Stok Produk',
'required');
```

```
        if ($this->form_validation->run() == false) {
            $this->load->view('produk/edit', $data);
        } else {
            $dataUpdate = [
                'nama_produk' => $this->input->post('nama_produk'),
                'harga_produk' => $this->input->post('harga_produk'),
                'stok_produk' => $this->input->post('stok_produk')
            ];

            $this->Produk_model->updateProduk($dataUpdate, $id);
            $this->session->set_flashdata('message', 'Produk berhasil
diupdate!');
            redirect('produk');
        }
    }
}
```

Penjelasan Teori:

- **$this->db->where()**: Digunakan untuk menentukan baris yang akan diupdate.
- **$this->db->update()**: Digunakan untuk melakukan perintah update terhadap tabel.

---

6. Operasi DELETE

Teori DELETE:

Operasi **DELETE** adalah proses untuk menghapus data dari database. Di CodeIgniter, metode **delete()** digunakan untuk menghapus data berdasarkan kondisi yang ditentukan.

Contoh Penggunaan:

```
php

class Produk_model extends CI_Model {
    public function deleteProduk($id) {
        $this->db->where('id_produk', $id);
        return $this->db->delete('produk');
    }
}
```

Fungsi `deleteProduk()` digunakan untuk menghapus data produk berdasarkan **id_produk**.

Contoh Penggunaan di Controller:

```
php

class Produk extends CI_Controller {
    public function delete($id) {
```

```
        $this->Produk_model->deleteProduk($id);
        $this->session->set_flashdata('message', 'Produk berhasil dihapus!');
        redirect('produk');
    }
}
```

Penjelasan Teori:

- **$this->db->delete()**: Digunakan untuk menghapus data dari tabel berdasarkan kondisi yang ditentukan di dalam **where**.

---

7. Kesimpulan

Operasi CRUD adalah dasar dari hampir semua aplikasi berbasis database. Dengan menggunakan CodeIgniter 3, operasi CRUD dapat dilakukan dengan mudah berkat metode-metode bawaan seperti **insert()**, **get()**, **update()**, dan **delete()**. Dalam modul ini, kita telah membahas:

- Cara menambahkan data baru ke database dengan operasi **INSERT**.
- Cara menampilkan data dari database dengan operasi **SELECT**.
- Cara memperbarui data yang sudah ada dengan operasi **UPDATE**.
- Cara menghapus data dari database dengan operasi **DELETE**.

Dengan pemahaman yang baik tentang operasi CRUD, pengembang dapat membangun aplikasi berbasis database yang fungsional, dinamis, dan efisien.