

Praktikum 7: Relasi Tabel dan Query Builder

Tujuan

1. Mahasiswa mampu memahami konsep relasi antar tabel dalam database.
2. Mahasiswa mampu mengimplementasikan relasi One-to-Many.
3. Mahasiswa mampu melakukan query dengan join tabel menggunakan Query Builder.
4. Mahasiswa mampu menampilkan data dari tabel yang berelasi.

Instruksi Praktikum

1. Persiapkan text editor (VSCode).
2. Buka kembali folder proyek lab7_php_ci.
3. Ikuti langkah-langkah praktikum berikut.

Relasi Tabel dan Query Builder

Praktikum ini merupakan kelanjutan dari praktikum sebelumnya, dimana kita akan memperdalam pemahaman tentang Model, Relasi Tabel dan Joining, serta penggunaan Query Builder dalam CodeIgniter 4.

- **Model:** Dalam CodeIgniter, Model adalah bagian dari arsitektur MVC (Model-View-Controller) yang bertugas untuk berinteraksi dengan database. Model menyediakan cara untuk mengambil, menyimpan, mengubah, dan menghapus data dari database.
- **Relasi Tabel:** Relasi tabel digunakan untuk menghubungkan data antar tabel dalam database. Dalam praktikum ini, kita akan fokus pada relasi One-to-Many, di mana satu kategori dapat memiliki banyak artikel.
- **Query Builder:** Query Builder adalah fitur yang disediakan oleh CodeIgniter untuk membuat query database dengan cara yang lebih mudah dan aman. Dengan Query Builder, kita dapat melakukan join tabel, memfilter data, dan mengurutkan hasil tanpa harus menulis query SQL secara manual.

Dengan memahami konsep-konsep ini, Anda akan mampu membangun aplikasi web yang lebih kompleks dan efisien.

Langkah-langkah Praktikum

1. Persiapan

Pastikan MySQL Server sudah berjalan, dan buka database `lab_ci4`

2. Membuat Tabel Kategori

Kita akan membuat tabel baru bernama `kategori` untuk mengkategorikan artikel.

Struktur Tabel `kategori`:

Field	Type Data	Ukuran	Keterangan
id_kategori	INT	11	PRIMARY KEY, auto_increment
nama_kategori	VARCHAR	100	
slug_kategori	VARCHAR	100	

Jalankan query berikut:

```
CREATE TABLE kategori (
    id_kategori INT(11) AUTO_INCREMENT,
    nama_kategori VARCHAR(100) NOT NULL,
    slug_kategori VARCHAR(100),
    PRIMARY KEY (id_kategori)
);
```

3. Mengubah Tabel Artikel

Tambahkan *foreign key* `id_kategori` pada tabel `artikel` untuk membuat relasi dengan tabel `kategori`.

Query untuk menambahkan foreign key:

```
ALTER TABLE artikel
ADD COLUMN id_kategori INT(11),
ADD CONSTRAINT fk_kategori_artikel
FOREIGN KEY (id_kategori) REFERENCES kategori(id_kategori);
```

4. Membuat Model Kategori

Buat file model baru di `app/Models` dengan nama `KategoriModel.php`:

```
<?php

namespace App\Models;

use CodeIgniter\Model;

class KategoriModel extends Model
{
    protected $table = 'kategori';
    protected $primaryKey = 'id_kategori';
    protected $useAutoIncrement = true;
    protected $allowedFields = ['nama_kategori', 'slug_kategori'];
}
```

5. Memodifikasi Model Artikel

Modifikasi `ArtikelModel.php` untuk mendefinisikan relasi dengan `KategoriModel`:

```
<?php

namespace App\Models;

use CodeIgniter\Model;

class ArtikelModel extends Model
{
    protected $table = 'artikel';
    protected $primaryKey = 'id';
    protected $useAutoIncrement = true;
    protected $allowedFields = ['judul', 'isi', 'status', 'slug', 'gambar',
'id_kategori'];

    public function getArtikelDenganKategori()
    {
        return $this->db->table('artikel')
            ->select('artikel.*, kategori.nama_kategori')
            ->join('kategori', 'kategori.id_kategori =
artikel.id_kategori');
```

```
        ->get()  
        ->getResultArray();  
    }  
}
```

Menambahkan method `getArtikelDenganKategori()` untuk mengambil data artikel beserta nama kategorinya menggunakan join.

6. Memodifikasi Controller Artikel

Modifikasi `Artikel.php` untuk menggunakan model baru dan menampilkan data relasi:

```
<?php  
  
namespace App\Controllers;  
  
use App\Models\ArtikelModel;  
use App\Models\KategoriModel;  
  
class Artikel extends BaseController  
{  
    public function index()  
    {  
        $title = 'Daftar Artikel';  
        $model = new ArtikelModel();  
        $artikel = $model->getArtikelDenganKategori(); // Use the new method  
  
        return view('artikel/index', compact('artikel', 'title'));  
    }  
  
    public function admin_index()  
    {  
        $title = 'Daftar Artikel (Admin)';  
        $model = new ArtikelModel();  
  
        // Get search keyword  
        $q = $this->request->getVar('q') ?? '';  
        // Get category filter  
        $kategori_id = $this->request->getVar('kategori_id') ?? '';  
  
        $data = [  
            'title' => $title,  
            'q' => $q,  
            'kategori_id' => $kategori_id,  
        ];  
  
        // Building the query  
        $builder = $model->table('artikel')
```

```

        ->select('artikel.*, kategori.nama_kategori')
        ->join('kategori', 'kategori.id_kategori =
artikel.id_kategori');

    // Apply search filter if keyword is provided
    if ($q != '') {
        $builder->like('artikel.judul', $q);
    }

    // Apply category filter if category_id is provided
    if ($kategori_id != '') {
        $builder->where('artikel.id_kategori', $kategori_id);
    }

    // Apply pagination
    $data['artikel'] = $builder->paginate(10);
    $data['pager'] = $model->pager;

    // Fetch all categories for the filter dropdown
    $kategoriModel = new KategoriModel();
    $data['kategori'] = $kategoriModel->findAll();

    return view('artikel/admin_index', $data);
}

// ... (methods add, edit, delete remain largely the same, but update to
handle id_kategori)

public function add()
{
    // Validation...
    if ($this->request->getMethod() == 'post' && $this->validate([
        'judul' => 'required',
        'id_kategori' => 'required|integer' // Ensure id_kategori is
required and an integer
    ])) {
        $model = new ArtikelModel();
        $model->insert([
            'judul' => $this->request->getPost('judul'),
            'isi' => $this->request->getPost('isi'),
            'slug' => url_title($this->request->getPost('judul')),
            'id_kategori' => $this->request->getPost('id_kategori')
        ]);
        return redirect()->to('/admin/artikel');
    } else {
        $kategoriModel = new KategoriModel();
        $data['kategori'] = $kategoriModel->findAll(); // Fetch categories
for the form
        $data['title'] = "Tambah Artikel";
    }
}

```

```

        return view('artikel/form_add', $data);
    }
}

public function edit($id)
{
    $model = new ArtikelModel();
    if ($this->request->getMethod() == 'post' && $this->validate([
        'judul' => 'required',
        'id_kategori' => 'required|integer'
    ])) {
        $model->update($id, [
            'judul' => $this->request->getPost('judul'),
            'isi' => $this->request->getPost('isi'),
            'id_kategori' => $this->request->getPost('id_kategori')
        ]);
        return redirect()->to('/admin/artikel');
    } else {
        $data['artikel'] = $model->find($id);
        $kategoriModel = new KategoriModel();
        $data['kategori'] = $kategoriModel->findAll(); // Fetch categories
        for the form
        $data['title'] = "Edit Artikel";
        return view('artikel/form_edit', $data);
    }
}

public function delete($id)
{
    $model = new ArtikelModel();
    $model->delete($id);
    return redirect()->to('/admin/artikel');
}

public function view($slug)
{
    $model = new ArtikelModel();
    $data['artikel'] = $model->where('slug', $slug)->first();
    if (empty($data['artikel'])) {
        throw new \CodeIgniter\Exceptions\PageNotFoundException('Cannot
        find the article.');
```

7. Memodifikasi View

Buka folder view/artikel sesuaikan masing-masing view.

index.php

```
<?= $this->include('template/header'); ?>

<?php if ($artikel): foreach ($artikel as $row): ?>
    <article class="entry">
        <h2><a href="<?= base_url('/artikel/' . $row['slug']); ?>"><?=
$row['judul']; ?></a></h2>
        <p>Kategori: <?= $row['nama_kategori'] ?></p>
        ">
        <p><?= substr($row['isi'], 0, 200); ?></p>
    </article>
    <hr class="divider" />
<?php endforeach; else: ?>
    <article class="entry">
        <h2>Belum ada data.</h2>
    </article>
<?php endif; ?>

<?= $this->include('template/footer'); ?>
```

admin_index.php

```
<?= $this->include('template/admin_header'); ?>

<h2><?= $title; ?></h2>

<div class="row mb-3">
    <div class="col-md-6">
        <form method="get" class="form-inline">
            <input type="text" name="q" value="<?= $q; ?>" placeholder="Cari
judul artikel" class="form-control mr-2">
            <select name="kategori_id" class="form-control mr-2">
                <option value="">Semua Kategori</option>
                <?php foreach ($kategori as $k): ?>
                    <option value="<?= $k['id_kategori']; ?>" <?= ($kategori_id
== $k['id_kategori']) ? 'selected' : ''; ?>><?= $k['nama_kategori'];
?></option>
                <?php endforeach; ?>
            </select>
            <input type="submit" value="Cari" class="btn btn-primary">
        </form>
    </div>
</div>

<table class="table">
    <thead>
```

```

        <tr>
            <th>ID</th>
            <th>Judul</th>
            <th>Kategori</th>
            <th>Status</th>
            <th>Aksi</th>
        </tr>
    </thead>
    <tbody>
        <?php if (count($artikel) > 0): ?>
            <?php foreach ($artikel as $row): ?>
                <tr>
                    <td><?=$row->id; ?></td>
                    <td>
                        <b><?=$row->judul; ?></b>
                        <p><small><?=$row->isi, 0, 50); ?></small></p>
                    </td>
                    <td><?=$row->nama_kategori; ?></td>
                    <td><?=$row->status; ?></td>
                    <td>
                        <a class="btn btn-sm btn-info" href="<?=$row->id; ?>">Ubah</a>
                        <a class="btn btn-sm btn-danger" onclick="return confirm('Yakin menghapus data?');" href="<?=$row->id; ?>">Hapus</a>
                    </td>
                </tr>
            <?php endforeach; ?>
        <?php else: ?>
            <tr>
                <td colspan="5">Tidak ada data.</td>
            </tr>
        <?php endif; ?>
    </tbody>
</table>

<?=$pager->only(['q', 'kategori_id']->links(); ?>

<?=$this->include('template/admin_footer'); ?>

```

form_add.php

```

<?=$this->include('template/admin_header'); ?>

<h2><?=$title; ?></h2>

<form action="" method="post">
    <p>
        <label for="judul">Judul</label>
    </p>

```

```

        <input type="text" name="judul" id="judul" required>
    </p>
    <p>
        <label for="isi">Isi</label>
        <textarea name="isi" id="isi" cols="50" rows="10"></textarea>
    </p>
    <p>
        <label for="id_kategori">Kategori</label>
        <select name="id_kategori" id="id_kategori" required>
            <?php foreach($kategori as $k): ?>
                <option value="<?=$k['id_kategori']; ?>"><?=$k['nama_kategori']; ?></option>
            <?php endforeach; ?>
        </select>
    </p>
    <p><input type="submit" value="Kirim" class="btn btn-large"></p>
</form>

<?=$this->include('template/admin_footer'); ?>

```

form_edit.php

```

<?=$this->include('template/admin_header'); ?>

<h2><?=$title; ?></h2>

<form action="" method="post">
    <p>
        <label for="judul">Judul</label>
        <input type="text" name="judul" value="<?=$artikel['judul']; ?>"
        id="judul" required>
    </p>
    <p>
        <label for="isi">Isi</label>
        <textarea name="isi" id="isi" cols="50" rows="10"><?=$artikel['isi'];
        ?></textarea>
    </p>
    <p>
        <label for="id_kategori">Kategori</label>
        <select name="id_kategori" id="id_kategori" required>
            <?php foreach($kategori as $k): ?>
                <option value="<?=$k['id_kategori']; ?>" <?=$artikel['id_kategori'] == $k['id_kategori'] ? 'selected' : ''; ?>><?=$k['nama_kategori']; ?></option>
            <?php endforeach; ?>
        </select>
    </p>
    <p><input type="submit" value="Kirim" class="btn btn-large"></p>
</form>

```



```
<?= $this->include('template/admin_footer'); ?>
```

8. Testing

Lakukan uji coba untuk memastikan semua fungsi berjalan dengan baik:

- Menampilkan daftar artikel dengan nama kategori.
- Menambah artikel baru dengan memilih kategori.
- Mengedit artikel dan mengubah kategorinya.
- Menghapus artikel.

Pertanyaan dan Tugas

1. Selesaikan semua langkah praktikum di atas.
2. Modifikasi tampilan detail artikel (artikel/detail.php) untuk menampilkan nama kategori artikel.
3. Tambahkan fitur untuk menampilkan daftar kategori di halaman depan (opsional).
4. Buat fungsi untuk menampilkan artikel berdasarkan kategori tertentu (opsional).

Laporan Praktikum

1. Lanjutkan praktikum pada repository Lab7Web.
2. Kerjakan semua latihan sesuai urutan.
3. Screenshot setiap perubahan.
4. Update file README.md dengan penjelasan dan screenshot.
5. Commit hasil praktikum.
6. Kirim URL repository.