

Script ini dibuat oleh :

Syahroni Wahyu Iriananda, S.Kom, MT

Teknik Informatika, Universitas Widya Gama

081331711385

Untuk bisa mendapatkan tampilan seperti ini dari deepseek :

- delete element yang tidak diperlukan
- lalu masuk ke clearlyreader
- intip tampilan
- jika sudah sesuai copy html
- masuk ke dillinger.io dan paste clipboard
- Download MD atau PDF lebih bagus

Kode CRUDL Form Builder and Generator Versi 2

DeepSeek

<https://chat.deepseek.com/a/chat/s/3a5a90e4-4831-4217-8b3e-106f881fd660>

=====

Config/Routes.php

```
$routes->group('forms', ['namespace' => 'App\Controllers'], function ($routes) {  
    $routes->get('/', 'FormController::index');  
    $routes->get('configure/(:any)', 'FormController::configure/$1');  
    $routes->get('configureAjax/(:num)', 'FormController::configureAjax/$1');  
    $routes->post('save', 'FormController::save');  
    $routes->get('preview/(:num)', 'FormController::preview/$1');  
    $routes->get('edit/(:num)', 'FormController::edit/$1');  
    $routes->get('delete/(:num)', 'FormController::delete/$1');  
    $routes->get('regenerate/(:num)', 'FormController::regenerate/$1');  
    $routes->post('update', 'FormController::update'); // Update form  
    $routes->get('generate/(:num)', 'FormController::generate/$1');
```

```
$routes->post('submit/(:num)', 'SubmissionController::submit/$1');
$routes->get('submissions/(:num)', 'SubmissionController::view/$1');
$routes->get('addtime/(:any)', 'FormController::addTimestamp/$1');
$routes->post('getfields', 'FormController::getTableFields');

});
```

Controller/FormController.php

```
<?php
// https://chat.deepseek.com/a/chat/s/3a5a90e4-4831-4217-8b3e-106f881fd660

namespace App\Controllers;

use App\Controllers\BaseController;
use CodeIgniter\HTTP\ResponseInterface;
use CodeIgniter\HTTP\RequestInterface;
use App\Models\FormModel;
use App\Models\FormSubmissionModel;
use CodeIgniter\Database\BaseConnection;

class FormController extends BaseController
{
    protected $db;

    public function __construct()
    {
        $this->db = \Config\Database::connect();
    }

    // Step 1: Fetch MySQL tables
    public function index()
    {
        $formModel = new FormModel();
        $data['tables'] = $this->db->listTables();
        $data['forms'] = $formModel->findAll(); // Fetch all forms from the database
        return view('formbuilder/index', $data);
    }

    public function configureAjax($id = null)
```

```

{
    $data = [];

    if ($id) {
        // Mode Edit: Ambil data form yang sudah ada
        $formModel = new FormModel();
        $form = $formModel->find($id);

        if (!$form) {
            return $this->response->setJSON(['error' => 'Form not found']);
        }

        $data['form'] = $form;
        $data['fields'] = json_decode($form['structure'], true);
    } else {
        // Mode Configure: Ambil daftar tabel
        $db = \Config\Database::connect();
        $data['tables'] = $db->listTables();
    }

    return view('formbuilder/configure_edit', $data);
}

// Step 2: Configure form fields
public function configure($table,$id=null)
{
    $formModel = new FormModel();
    $fields = $this->db->getFieldData($table);
    // Jika mode edit, ambil data form yang sudah ada
    if ($id) {
        $formConfig = $formModel->find($id);
        if (!$formConfig) {
            return redirect()->to('/forms')->with('error', 'Form not found.');
```

```

        // Validasi apakah tabel memiliki kolom created_at, updated_at, dan delete
d_at

        $requiredFields = ['created_at', 'updated_at', 'deleted_at'];
        $missingFields = [];

        foreach ($requiredFields as $field) {
            if (!in_array($field, array_column($fields, 'name'))) {
                $missingFields[] = $field;
            }
        }

        // Jika ada kolom yang hilang, tampilkan pesan error dan tombol untuk mena
mbahkan kolom

        if (!empty($missingFields)) {
            $errorMessage = "Tabel $table tidak memiliki kolom berikut: " . implode(', ', $missingFields) . ". Silakan tambahkan kolom tersebut.";
            session()->setFlashdata('error', $errorMessage);

            // Tambahkan tombol untuk menambahkan kolom timestamp
            $data['missingFields'] = $missingFields;
            // $data['table'] = $table;
            // $data['fields'] = $fields;
            // return view('formbuilder/configure', $data);
        }
    }

    $data['tables'] = $this->db->listTables(); // Daftar tabel untuk chained selec
t

    // Jika semua kolom ada, tampilkan form konfigurasi
    $data['table'] = $table;
    $data['fields'] = $fields;
    return view('formbuilder/configure', $data);
}

// Fungsi untuk menambahkan kolom timestamp
public function addTimestamp($table)
{
    $db = \Config\Database::connect();

```

```

try {
    // Tambahkan kolom created_at, updated_at, dan deleted_at jika belum ada
    if (!$db->fieldExists('created_at', $table)) {
        $db->query("ALTER TABLE $table ADD created_at TIMESTAMP DEFAULT CURREN
T_TIMESTAMP");
    }
    if (!$db->fieldExists('updated_at', $table)) {
        $db->query("ALTER TABLE $table ADD updated_at TIMESTAMP DEFAULT CURREN
T_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP");
    }
    if (!$db->fieldExists('deleted_at', $table)) {
        $db->query("ALTER TABLE $table ADD deleted_at TIMESTAMP NULL");
    }

    return redirect()->to("/forms/configure/$table")->with('success', 'Kolom t
imestamp berhasil ditambahkan.');
```

```

    } catch (\Exception $e) {
        return redirect()->to("/forms/configure/$table")->with('error', 'Gagal men
ambahkan kolom timestamp: ' . $e->getMessage());
    }
}

public function saveAjax()
{
    $formModel = new FormModel();

    $data = [
        'title' => $this->request->getPost('title'),
        'description' => $this->request->getPost('description'),
        'structure' => json_encode($this->request->getPost('fields')),
        'routes' => $this->request->getPost('routes'),
        'status' => 'draft',
        'table_name' => $this->request->getPost('table'),
    ];

    if ($this->request->getPost('id')) {
        // Mode Edit: Update form yang sudah ada

```

```

        $formModel->update($this->request->getPost('id'), $data);
    } else {
        // Mode Configure: Buat form baru
        $formModel->insert($data);
    }

    return $this->response->setJSON(['success' => true]);
}

// Step 3: Save form structure
public function save() {
    $formModel = new FormModel();
    $fields = $this->request->getPost('fields');

    // Validasi tipe input baru
    foreach ($fields as &$field) {
        if (in_array($field['type'], ['datepicker', 'daterangepicker', 'file', 'email', 'password', 'switch'])) {
            $field['type'] = strtolower($field['type']); // Pastikan lowercase
        }
    }

    $validationRules = [
        'title' => 'required|min_length[3]|max_length[255]',
        'table' => 'required',
        'routes' => 'required',
    ];

    if (!$this->validate($validationRules)) {
        return redirect()->back()->withInput()->with('errors', $this->validator->getErrors());
    }

    // Ambil data dari POST request
    $data = [
        'title' => $this->request->getPost('title'),
        'description' => $this->request->getPost('description'),
        'structure' => json_encode($fields),
        'routes' => $this->request->getPost('routes'),
    ];
}

```

```

        'status' => 'draft',
        'table_name' => $this->request->getPost('table'),
    ];

    if ($this->request->getPost('id')) {
        $formModel->update($this->request->getPost('id'), $data);
    } else {
        $formModel->insert($data);
    }

    return redirect()->to('/forms')->with('message', 'Form berhasil disimpan.');
```

}

```

public function submit($id)
{
    $formSubmissionModel = new FormSubmissionModel();

    $data = [
        'form_id' => $id,
        'data' => json_encode($this->request->getPost()),
    ];

    $formSubmissionModel->insert($data);
    return redirect()->to('/thank-you');
}

// Step 4: Preview form
public function preview($id)
{
    $formModel = new FormModel();
    $form = $formModel->find($id);

    if (!$form) {
        return redirect()->to('/forms')->with('error', 'Form tidak ditemukan.');
```

}

```

    return view('formbuilder/preview', ['form' => $form]);
}

```

```

// Step 5: Edit form
// Edit a form
public function edit($id)
{
    $formModel = new FormModel();
    $formConfig = $formModel->find($id);
    // echo var_dump($formConfig);

    // if (!$formConfig['form']) {
    //     return redirect()->to('/forms')->with('error', 'Form not found.');
```

m configuration

```

    // }
    $table = $formConfig['table_name']; // Assuming 'table' is a field in your form configuration

    // $data['tables'] =
    $data['forms'] = $formModel->findAll();
    // $fields = $this->db->getFieldData($form['table_name']);

    // Pass the data to the view
    $data = [
        'formConfig' => $formConfig,
        'table' => $table,
        'fields' => $this->db->getFieldData($table), // Example function to get fields
```

elds

```

        'tables' => $this->db->listTables(), // Example function to get all tables
    ];
    return view('formbuilder/edit', $data);
}

// Update a form
public function update()
{
    $formModel = new FormModel();

    $data = [
        'title' => $this->request->getPost('title'),
        'description' => $this->request->getPost('description'),
```



```

        'structure' => json_encode($this->request->getPost('fields')),
        'routes' => $this->request->getPost('routes'),
        'status' => 'draft',
        'table_name' => $this->request->getPost('table'),
    ];

    $formModel->update($this->request->getPost('id'), $data);
    return redirect()->to('/forms')->with('message', 'Form updated successfull
y.');
```

}

// Step 6: Delete form

```

public function delete($id)
{
    $formModel = new FormModel();
    $formModel->delete($id);

    return redirect()->to('/forms');
}
```

// Step 7: Regenerate form

```

public function regenerate($id)
{
    $formModel = new FormModel();
    $form = $formModel->find($id);

    if (!$form) {
        return redirect()->to('/forms')->with('error', 'Form tidak ditemukan.');
```

}

// Fetch updated table structure

```

$fields = $this->db->getFieldData($form['table_name']);

// Update form structure
$form['structure'] = json_encode($fields);
$formModel->update($id, $form);

return redirect()->to('/forms')->with('message', 'Form berhasil diregeneras
```

```

i. ');
    }

// app/Controllers/FormController.php
public function generate($id)
{
    $formModel = new FormModel();
    // try {
    // Ambil data form dari database
    $form = $formModel->find($id);

    if (!$form) {
        throw new \RuntimeException("Form not found");
    }
    // Validasi struktur tabel
    $this->validateTableStructure($form['table_name']);

    // Inisialisasi CrudGenerator
    $crudGenerator = service('CrudGenerator');

    // Generate CRUD untuk form ini saja
    $crudGenerator->generateFromForm($form);
    // Update status form menjadi 'published'
    $formModel->update($id, ['status' => 'published']);

    // return redirect()->back()->with('success', "CRUD untuk tabel {$form['table_
name']] berhasil digenerate!");
    // } catch (\Exception $e) {
    // return redirect()->back()->with('error', $e->getMessage());
    // }
}
// Get primary key of a table
public function getPrimaryKey($tableName)
{
    $fields = $this->db->getFieldData($tableName);

```

```

        foreach ($fields as $field) {
            if ($field->primary_key) {
                return $field->name; // Mengembalikan nama Primary Key
            }
        }

        return 'id'; // Default jika tidak ada Primary Key
    }

    // Validate table structure
    protected function validateTableStructure($tableName)
    {
        $fields = $this->db->getFieldData($tableName);

        $requiredFields = ['created_at', 'updated_at', 'deleted_at'];
        $missingFields = [];

        foreach ($requiredFields as $field) {
            if (!in_array($field, array_column($fields, 'name'))) {
                $missingFields[] = $field;
            }
        }

        if (!empty($missingFields)) {
            throw new \RuntimeException("Tabel $tableName tidak memiliki kolom berikut: " . implode(', ', $missingFields) . ". Silakan tambahkan kolom tersebut.");
        }
    }

    public function getTableFields() {
        $table = $this->request->getPost('table');
        if (empty($table)) {
            return $this->response->setJSON(['error' => 'Nama tabel tidak ditemukan']);
        }

        try {
            $fields = $this->db->getFieldData($table);

```

```

        if (empty($fields)) {
            return $this->response->setJSON(['error' => 'Tidak ada field yang dite
mukan untuk tabel ini']);
        }

        // Convert fields to array of objects
        $fieldArray = [];
        foreach ($fields as $field) {
            $fieldArray[] = [
                'name' => $field->name,
                'type' => $field->type
            ];
        }

        // Log response untuk debugging
        // log_message('debug', "Response JSON: " . json_encode($fieldArray));

        return $this->response->setJSON($fieldArray);
    } catch (\Exception $e) {
        log_message('error', "Failed to fetch fields: " . $e->getMessage());
        return $this->response->setJSON(['error' => 'Gagal mengambil field dari ta
bel: ' . $e->getMessage()]);
    }
}
}

```

Models/FormModel.php

```

<?php

namespace App\Models;

use CodeIgniter\Model;

class FormModel extends Model
{

```

```
protected $table          = 'forms';
protected $primaryKey     = 'id';
protected $useAutoIncrement = true;
protected $returnType     = 'array';
protected $useSoftDeletes = false;
protected $protectFields  = true;
protected $allowedFields = ['title', 'description', 'structure', 'status', 'table_name', 'routes'];

protected bool $allowEmptyInserts = false;
protected bool $updateOnlyChanged = true;

protected array $casts = [];
protected array $castHandlers = [];

// Dates
protected $useTimestamps = true;
protected $dateFormat    = 'datetime';
protected $createdField   = 'created_at';
protected $updatedField   = 'updated_at';
protected $deletedField   = 'deleted_at';

// Validation
protected $skipValidation = false;
protected $cleanValidationRules = true;

// Callbacks
protected $allowCallbacks = true;
protected $beforeInsert   = [];
protected $afterInsert    = [];
protected $beforeUpdate   = [];
protected $afterUpdate    = [];
protected $beforeFind     = [];
protected $afterFind      = [];
protected $beforeDelete   = [];
protected $afterDelete    = [];

protected $validationRules = [
```

```

        'title' => 'required|min_length[3]|max_length[255]',
        'structure' => 'required',
        'status' => 'required|in_list[draft,published]'
    ];

    protected $validationMessages = [
        'title' => [
            'required' => 'Form title is required',
            'min_length' => 'Form title must be at least 3 characters long',
            'max_length' => 'Form title cannot exceed 255 characters'
        ],
        'structure' => [
            'required' => 'Form structure is required'
        ],
        'status' => [
            'required' => 'Form status is required',
            'in_list' => 'Invalid form status'
        ]
    ];

    public function getFormWithSubmissions($id)
    {
        $form = $this->find($id);
        if ($form) {
            $submissionModel = new \App\Models\FormSubmissionModel();
            $form['submissions'] = $submissionModel->where('form_id', $id)->findAll();
        }
        return $form;
    }
}

```

Models/FormSubmissionModel.php

```

<?php

namespace App\Models;

```

```
use CodeIgniter\Model;

class FormSubmissionModel extends Model
{
    protected $table          = 'form_submissions';
    protected $primaryKey     = 'id';
    protected $useAutoIncrement = true;
    protected $returnType     = 'array';
    protected $useSoftDeletes  = false;
    protected $protectFields   = true;
    protected $allowedFields   = ['form_id', 'data'];

    protected bool $allowEmptyInserts = false;
    protected bool $updateOnlyChanged = true;

    protected array $casts = [];
    protected array $castHandlers = [];

    // Dates
    protected $useTimestamps = false;
    protected $dateFormat    = 'datetime';
    protected $createdField   = 'created_at';
    protected $updatedField   = 'updated_at';
    protected $deletedField   = 'deleted_at';

    // Validation
    protected $validationMessages = [];
    protected $skipValidation      = false;
    protected $cleanValidationRules = true;

    // Callbacks
    protected $allowCallbacks = true;
    protected $afterInsert    = [];
    protected $beforeUpdate   = [];
    protected $afterUpdate    = [];
    protected $beforeFind     = [];
```

```
protected $afterFind      = [];
protected $beforeDelete   = [];
protected $afterDelete    = [];

protected $validationRules = [
    'form_id' => 'required|numeric|is_natural_no_zero',
    'data' => 'required'
];

protected $beforeInsert = ['setSubmittedAt'];

protected function setSubmittedAt(array $data)
{
    $data['data']['created_at'] = date('Y-m-d H:i:s');
    return $data;
}
```

Views/formbuilder/index.php

```
<?= $this->extend('themes/default/layouts/main') ?>

<?= $this->section('styles') ?>

<?= $this->endSection() ?>

<?= $this->section('content') ?>

<div class="container mt-5">
    <div class="row">
        <div class="col-md-3">

            <h1>Form Builder</h1>

            <!-- Step 1: List of Tables -->
            <h5>Select a Table to Configure</h5>
            <ul class="list-group">
                <?php foreach ($tables as $table): ?>
                    <li class="list-group-item">
```



```

                <a class="btn btn-primary btn-sm" href="/forms/configure/<?=$table ?>"><?=$table ?></a>
            </li>
        <?php endforeach; ?>
    </ul>
</div>
<div class="col-md-9">

    <!-- List of Existing Forms -->
    <h3 class="mt-5">Existing Forms</h3>
    <table class="table">
        <thead>
            <tr>
                <th>Title</th>
                <th>Description</th>
                <th>Actions</th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($forms as $form): ?>
                <tr>
                    <td><?=$form['title'] ?></td>
                    <td><?=$form['description'] ?></td>
                    <td>
                        <!-- Tambahkan tombol Generate -->
                        <a href="/forms/generate/<?=$form['id'] ?>" class="btn btn-sm btn-success" onclick="return confirm('Generate CRUD untuk <?=$form['title'] ?>?')">
                            <i class="fa fa-exclamation-triangle"></i>Generate
CRUD
                        </a>
                        <a href="/forms/preview/<?=$form['id'] ?>" class="btn btn-sm btn-info">Preview</a>
                        <a href="/forms/edit/<?=$form['id'] ?>" class="btn btn-sm btn-warning">Edit</a>

```

```

                <a href="/forms/delete/<?= $form['id'] ?>" class="btn
btn-sm btn-danger">Delete</a>

            </td>

        </tr>

    <?php endforeach; ?>

</tbody>

</table>

</div>

</div>

</div>

<?= $this->endSection() ?>

<?= $this->section('scripts') ?>

<script>
    $(document).ready(function() {

        <?php if (session()->getFlashdata('success')): ?>
            toastr.success('<?= session()->getFlashdata('success') ?>');
        <?php endif; ?>

        <?php if (session()->getFlashdata('error')): ?>
            toastr.error('<?= session()->getFlashdata('error') ?>');
        <?php endif; ?>

    });
</script>

<?= $this->endSection() ?>

```

Views/formbuilder/configure.php

```

<?= $this->extend('themes/default/layouts/main') ?>

<?= $this->section('styles') ?>

<link href="https://cdnjs.cloudflare.com/ajax/libs/select2/4.0.9/css/select2.min.css"
rel="stylesheet">

<style>

    .select2-container {

        width: 100% !important;

    }

```

```
</style>
<?= $this->endSection() ?>

<?= $this->section('content') ?>
<div class="container mt-5">
    <h1>Configure Form for Table: <?= esc($table) ?></h1>

    <?php if (session()->getFlashdata('error')): ?>
        <div class="alert alert-danger"><?= session()->getFlashdata('error') ?></div>
    <?php endif; ?>

    <?php if (session()->getFlashdata('success')): ?>
        <div class="alert alert-success"><?= session()->getFlashdata('success') ?></div>
    <?php endif; ?>

    <form action="/forms/save" method="post">
        <input type="hidden" name="table" value="<?= esc($table) ?>">

        <!-- Form Title and Description -->
        <div class="row">
            <div class="col-md-6">
                <div class="form-group">
                    <label>Form Title</label>
                    <input type="text" name="title" class="form-control" required>
                </div>
            </div>
            <div class="col-md-6">
                <div class="form-group">
                    <label>Form Description</label>
                    <textarea name="description" class="form-control"></textarea>
                </div>
            </div>
        </div>

        <!-- User-Friendly URL -->
        <div class="row">
```

```

        <div class="col-md-6">
            <div class="form-group">
                <label>User-Friendly URL</label>
                <input type="text" name="routes" class="form-control" placeholder
="Contoh: pegawai" required>
                <small class="text-muted">URL ini akan digunakan untuk routing CRU
DL (contoh: /pegawai).</small>
            </div>
        </div>
    </div>

<!-- Field Configuration -->
<div id="fieldsContainer">
    <?php foreach ($fields as $field): ?>
        <?php if (in_array($field->name, ['created_at', 'updated_at', 'deleted
_at'])) continue; ?>
        <div class="card mb-3 field-item">
            <div class="card-body">
                <h5>Field: <?= esc($field->name) ?></h5>
                <div class="row">
                    <div class="col-md-3">
                        <div class="form-group">
                            <label>Field Label</label>
                            <input type="text" name="fields[<?= esc($field->na
me) ?>][label]" class="form-control" value="<?= ucfirst($field->name) ?>" required>
                            <input type="hidden" name="fields[<?= esc($field->
name) ?>][name]" value="<?= esc($field->name) ?>">
                        </div>
                    </div>

                    <div class="col-md-3">
                        <div class="form-group">
                            <label>Field Type</label>
                            <select name="fields[<?= esc($field->name) ?>][typ
e]" class="form-control select2 field-type">
                                <option value="text">Text</option>
                                <option value="textarea">Textarea</option>

```

```

        <option value="select">Select</option>
        <option value="checkbox">Checkbox</option>
        <option value="radio">Radio</option>
        <option value="date">Date</option>
        <option value="file">File Upload</option>
    </select>
</div>
</div>

<!-- Chained Select Options -->
<div class="col-md-3 chained-select" style="display: none;">

    <div class="form-group">
        <label>Select Option Source</label>
        <select name="fields[<?= esc($field->name) ?>][select_source]" class="form-control select-source">
            <option value="manual">Manual</option>
            <option value="table">From Table</option>
        </select>
    </div>
</div>

<!-- Manual Options -->
<div class="col-md-3 manual-options" style="display: none;">

    <div class="form-group">
        <label>Manual Options</label>
        <textarea name="fields[<?= esc($field->name) ?>][manual_options]" class="form-control" placeholder="Contoh: Male,Male\nFemale,Female"></textarea>
    </div>
</div>

<!-- Table Options -->
<div class="col-md-3 table-options" style="display: none;">

    <div class="form-group">

```

```

        <label>Select Table</label>
        <select name="fields[<?= esc($field->name) ?>][table]" class="form-control select-table">
            <option value="">Pilih Tabel</option>
            <?php foreach ($tables as $tableOption): ?>
                <option value="<?= esc($tableOption) ?>">
<?= esc($tableOption) ?></option>
            <?php endforeach; ?>
        </select>
    </div>
    <div class="form-group">
        <label>Value Field</label>
        <select name="fields[<?= esc($field->name) ?>][value_field]" class="form-control value-field" disabled>
            <option value="">Pilih Field</option>
        </select>
    </div>
    <div class="form-group">
        <label>Label Field</label>
        <select name="fields[<?= esc($field->name) ?>][label_field]" class="form-control label-field" disabled>
            <option value="">Pilih Field</option>
        </select>
    </div>
</div>

<div class="col-md-3">
    <div class="form-group">
        <label>Show On</label>
        <div>
            <input type="checkbox" name="fields[<?= esc($field->name) ?>][show_on][]" value="create" checked> Create
            <input type="checkbox" name="fields[<?= esc($field->name) ?>][show_on][]" value="edit" checked> Edit
            <input type="checkbox" name="fields[<?= esc($field->name) ?>][show_on][]" value="list" checked> List
            <input type="checkbox" name="fields[<?= esc($f

```

```

field->name) ?>][show_on][]" value="read"> Read
                <input type="checkbox" name="fields[<?= esc($f
field->name) ?>][show_on][]" value="delete"> Delete
                </div>
            </div>
        </div>

        <div class="col-md-3">
            <div class="form-group">
                <label>Validation Rules</label>
                <input type="text" name="fields[<?= esc($field->na
me) ?>][validation_rule]" class="form-control" placeholder="e.g., required|min_length
[5]">
            </div>
        </div>
    </div>
<?php endforeach; ?>
</div>

    <button type="submit" class="btn btn-primary">Save Form</button>
</form>
</div>
<?= $this->endSection() ?>

<?= $this->section('scripts') ?>
<script src="https://code.jquery.com/jquery-3.7.0.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/select2/4.0.9/js/select2.min.js">
</script>
<script>
    $(document).ready(function() {
        // Initialize Select2
        $('.select2').select2();

        // Show/Hide Chained Select
        $('select[name*="[type]")').change(function() {

```

```

const fieldType = $(this).val();
const chainedSelect = $(this).closest('.field-item').find('.chained-select');

if (fieldType === 'select') {
    chainedSelect.show();
} else {
    chainedSelect.hide();
}
});

// Show/Hide Manual/Table Options
$('.select-source').change(function() {
    const selectSource = $(this).val();
    const manualOptions = $(this).closest('.field-item').find('.manual-options');

    const tableOptions = $(this).closest('.field-item').find('.table-options');

    if (selectSource === 'manual') {
        manualOptions.show();
        tableOptions.hide();
    } else if (selectSource === 'table') {
        manualOptions.hide();
        tableOptions.show();
    } else {
        manualOptions.hide();
        tableOptions.hide();
    }
});

// Ambil field dari tabel yang dipilih
$('.select-table').change(function() {
    const table = $(this).val();
    const valueField = $(this).closest('.table-options').find('.value-field');
    const labelField = $(this).closest('.table-options').find('.label-field');

    if (table) {

```



```

$.ajax({
  url: '/forms/getfields',
  method: 'POST',
  data: {
    table: table
  },
  success: function(response) {
    try {
      // const fields = JSON.parse(response);
      // **Pastikan response bukan string JSON**
      const fields = (typeof response === "string") ? JSON.parse
(response) : response;

      if (fields.error) {
        alert(fields.error); // Tampilkan pesan error jika ada
        return;
      }

      let valueOptions = '<option value="">Pilih Field</option
>';

      let labelOptions = '<option value="">Pilih Field</option
>';

      fields.forEach(field => {
        valueOptions += `<option value="${field.name}">${fiel
d.name}</option>`;

        labelOptions += `<option value="${field.name}">${fiel
d.name}</option>`;

      });

      valueField.html(valueOptions).prop('disabled', false);
      labelField.html(labelOptions).prop('disabled', false);
    } catch (error) {
      console.error('Error parsing JSON: ', error);
      alert('Gagal memproses data dari server.');
```

```

        },
        error: function(xhr, status, error) {
            console.error('AJAX Error:', xhr.responseText, status, error);
            alert('Gagal mengambil field dari tabel.');
```

Views/formbuilder/edit.php

```

<?= $this->extend('themes/default/layouts/main') ?>

<?= $this->section('styles') ?>
<link href="https://cdnjs.cloudflare.com/ajax/libs/highlight.js/11.11.0/styles/default.min.css" rel="stylesheet">
<link href="https://cdnjs.cloudflare.com/ajax/libs/select2/4.0.9/css/select2.min.css"
rel="stylesheet" integrity="sha512-nMnlpuaDPPrqlEls3IX/Q56H36qvBASwb3ipuo3MxeWbsQB1881o
x0cRv7UPTgBlriqoynt35KjEwgGUeUXIPnw==" crossorigin="anonymous" referrerpolicy="no-refe
rrer" />
<style>
    .select2-container .select2-selection--single {
        height: 38px;
    }

    .select2-container--default .select2-selection--single .select2-selection__rendered {
        line-height: 38px;
    }

```

```

        .select2-container--default .select2-selection--single .select2-selection__arrow {
            height: 38px;
        }
</style>
<?= $this->endSection() ?>

<?= $this->section('content') ?>
<div class="container mt-5">
    <h1>Edit Form Configuration for Table: <?= esc($table) ?></h1>
    <div class='card row ' style='margin: 10px;'>
        <div class='card-header col-md-12'>
            <h5>JSON Array</h5>
        </div>
        <div class='card-body col-md-12'>
            <code class="php"><?= json_encode($tables) ?></code>
        </div>
    </div>
    <?php if (session()->getFlashdata('error')): ?>
        <div class="alert alert-danger">
            <?= session()->getFlashdata('error') ?>
            <?php if (isset($missingFields)): ?>
                <br>
                <a href="/forms/addtime/<?= esc($table) ?>" class="btn btn-warning mt-2">Tambahkan Kolom Timestamp</a>
            <?php endif; ?>
        </div>
    <?php endif; ?>

    <?php if (session()->getFlashdata('success')): ?>
        <div class="alert alert-success">
            <?= session()->getFlashdata('success') ?>
        </div>
    <?php endif; ?>

    <form action="/forms/update" method="post">
        <input type="hidden" name="<?= csrf_token() ?>" value="<?= csrf_hash() ?>">

```

```

<input type="hidden" name="id" value="<?= esc($formConfig['id'] ?? '') ?>">
<input type="hidden" name="table" value="<?= esc($table) ?>">
<div class="row">
    <div class="mb-3 col-md-4">
        <label>Form Title</label>
        <input type="text" name="title" class="form-control" value="<?= esc($f
ormConfig['title'] ?? '') ?>" required>
    </div>
    <div class="mb-3 col-md-4">
        <label>Form Description</label>
        <textarea name="description" class="form-control"><?= esc($formConfig
['description'] ?? '') ?></textarea>
    </div>
    <div class="mb-3 col-md-4">
        <label>User-Friendly URL</label>
        <input type="text" name="routes" class="form-control" value="<?= esc
($formConfig['routes'] ?? '') ?>" placeholder="Contoh: pegawai" required>
        <small class="text-muted">URL ini akan digunakan untuk routing CRUDL
(contoh: /pegawai).</small>
    </div>
</div>
<!-- Field Configuration -->
<div id="fieldsContainer">
    <?php foreach ($fields as $field): ?>
        <?php
        // Skip field created_at, updated_at, dan deleted_at
        if (in_array($field->name, ['created_at', 'updated_at', 'deleted_a
t'])) {
            continue;
        }
        ?>
        <div class="mb-3 border p-3 row">
            <h5>Field: <?= esc($field->name) ?></h5>
            <div class="mb-3 col-md-3">
                <label>Field Label</label>
                <input type="text" name="fields[<?= esc($field->name) ?>][labe
l]" class="form-control" value="<?= esc($formConfig['fields'][$field->name]['label']

```

```

?? ucfirst($field->name)) ?>" required>

        <input type="hidden" name="fields[<?= esc($field->name) ?>][na
me]" value="<?= esc($field->name) ?>">

    </div>

    <div class="row fields col-md-3">

        <div class="mb-3 field-type">

            <label>Field Type</label>

            <select name="fields[<?= esc($field->name) ?>][type]" clas
s="form-control select2">

                <option value="text" <?= (isset($formConfig['fields']
[$field->name]['type']) && $formConfig['fields'][$field->name]['type'] === 'text') ?
'selected' : '' ?>>Text</option>

                <option value="textarea" <?= (isset($formConfig['field
s'][$field->name]['type']) && $formConfig['fields'][$field->name]['type'] === 'textare
a') ? 'selected' : '' ?>>Textarea</option>

                <option value="select" <?= (isset($formConfig['field
s'][$field->name]['type']) && $formConfig['fields'][$field->name]['type'] === 'selec
t') ? 'selected' : '' ?>>Select</option>

                <option value="checkbox" <?= (isset($formConfig['field
s'][$field->name]['type']) && $formConfig['fields'][$field->name]['type'] === 'checkbo
x') ? 'selected' : '' ?>>Checkbox</option>

                <option value="radio" <?= (isset($formConfig['fields']
[$field->name]['type']) && $formConfig['fields'][$field->name]['type'] === 'radio') ?
'selected' : '' ?>>Radio</option>

            </select>

        </div>

        <!-- Chained Select2 untuk Select -->

        <div class="mb-3 chained-select" style="display: <?= (isset($f
ormConfig['fields'][$field->name]['type']) && $formConfig['fields'][$field->name]['typ
e'] === 'select') ? 'block' : 'none'; ?>;">

            <label>Select Option Source</label>

            <select name="fields[<?= esc($field->name) ?>][select_sour
ce]" class="form-control select-source">

                <option value="manual" <?= (isset($formConfig['field
s'][$field->name]['select_source']) && $formConfig['fields'][$field->name]['select_sou
rce'] === 'manual' ? 'selected' : '') ?>>Manual</option>

                <option value="table" <?= (isset($formConfig['fields']

```

```

[$field->name]['select_source']) && $formConfig['fields'][$field->name]['select_source'] === 'table' ? 'selected' : '') ?>>From Table</option>

</select>

<!-- Manual Options -->
<div class="mb-3 manual-options" style="display: <?=(isset($formConfig['fields'][$field->name]['select_source']) && $formConfig['fields'][$field->name]['select_source'] === 'manual' ? 'block' : 'none') ?>;">
    <label>Manual Options</label>
    <textarea name="fields[<?= esc($field->name) ?>][manual_options]" class="form-control" placeholder="Contoh: Male,Male\nFemale,Female"><?= esc($formConfig['fields'][$field->name]['manual_options']) ?? '' ?></textarea>
</div>

<!-- Table Options -->
<div class="mb-3 table-options" style="display: <?=(isset($formConfig['fields'][$field->name]['select_source']) && $formConfig['fields'][$field->name]['select_source'] === 'table' ? 'block' : 'none') ?>;">
    <label>Select Table</label>
    <select name="fields[<?= esc($field->name) ?>][table]"
class="form-control select-table">
        <option value="">Pilih Tabel</option>
        <?php if (isset($tables)): ?>
            <?php foreach ($tables as $table): ?>
                <option value="<?= esc($table) ?>" <?=(isset($formConfig['fields'][$field->name]['table']) && $formConfig['fields'][$field->name]['table'] === $table ? 'selected' : '') ?>><?= esc($table) ?></option>
            <?php endforeach; ?>
        <?php endif; ?>
    </select>

    <div class="mb-3">
        <label>Value Field</label>
        <select name="fields[<?= esc($field->name) ?>][value_field]" class="form-control value-field" <?=(isset($formConfig['fields'][$field->name]['table']) && $formConfig['fields'][$field->name]['table']) ? '' : 'disabled' ?>>
            <option value="">Pilih Field</option>
            <?php if (isset($formConfig['fields'][$field->

```

```

name]['value_field'])): ?>

                                <option value="<?= esc($formConfig['field
s'][$field->name]['value_field']) ?>" selected><?= esc($formConfig['fields'][$field->n
ame]['value_field']) ?></option>

                                <?php endif; ?>
                            </select>
                        </div>

                    <div class="mb-3">
                        <label>Label Field</label>
                        <select name="fields[<?= esc($field->name) ?>][lab
el_field]" class="form-control label-field" <?= (isset($formConfig['fields'][$field->n
ame]['table']) && $formConfig['fields'][$field->name]['table']) ? '' : 'disabled' ?>>
                            <option value="">Pilih Field</option>
                            <?php if (isset($formConfig['fields'][$field->
name]['label_field'])): ?>
                                <option value="<?= esc($formConfig['field
s'][$field->name]['label_field']) ?>" selected><?= esc($formConfig['fields'][$field->n
ame]['label_field']) ?></option>

                                <?php endif; ?>
                            </select>
                        </div>
                    </div>
                </div>

                <div class="mb-3 col-md-3">
                    <label>Show On</label>
                    <div>
                        <input type="checkbox" name="fields[<?= esc($field->na
me) ?>][show_on][]" value="create" <?= (isset($formConfig['fields'][$field->name]['sho
w_on']) && in_array('create', $formConfig['fields'][$field->name]['show_on'])) ? 'check
ed' : '' ?>> Create

                        <input type="checkbox" name="fields[<?= esc($field->na
me) ?>][show_on][]" value="edit" <?= (isset($formConfig['fields'][$field->name]['show_
on']) && in_array('edit', $formConfig['fields'][$field->name]['show_on'])) ? 'checked'
: '' ?>> Edit

                        <input type="checkbox" name="fields[<?= esc($field->na

```

```

me) ?>][show_on][]" value="list" <?=(isset($formConfig['fields'][$field->name]['show_
on']) && in_array('list', $formConfig['fields'][$field->name]['show_on']) ? 'checked'
: '') ?>> List

<input type="checkbox" name="fields[<?=(isset($formConfig['fields'][$field->name]['show_
on']) && in_array('read', $formConfig['fields'][$field->name]['show_on']) ? 'checked'
: '') ?>> Read

<input type="checkbox" name="fields[<?=(isset($formConfig['fields'][$field->name]['sho
w_on']) && in_array('delete', $formConfig['fields'][$field->name]['show_on']) ? 'check
ed' : '') ?>> Delete

</div>
</div>
<div class="mb-3 col-md-3">
<label>Validation Rules</label>
<input type="text" name="fields[<?=(isset($formConfig['fields'][$field->name) ?>]
[validation_rule]" class="form-control" value="<?=(isset($formConfig['fields'][$field->n
ame]['validation_rule'] ?? '') ?>" placeholder="Contoh: required|numeric|min_length
[5]">

</div>
<div class="mb-3">
<input type="checkbox" name="fields[<?=(isset($formConfig['fields'][$field->name)
?>][required]" value="1" <?=(isset($formConfig['fields'][$field->name]['required']) &
& $formConfig['fields'][$field->name]['required'] == 1 ? 'checked' : '') ?>> Required
</div>
</div>
<?php endforeach; ?>
</div>

<button type="submit" class="btn btn-primary">Update Form</button>

</form>
</div>
<?=$this->endSection() ?>

<?=$this->section('scripts') ?>
<script src="https://cdnjs.cloudflare.com/ajax/libs/highlight.js/11.11.0/highlight.mi

```



```

n.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/highlight.js/11.11.1/languages/php.min.js" integrity="sha512-uSKj9vayq7XKzfzflBQdmCuLIzKtsmsv7jjfr85Z0GQxNyID1anc0GMYHsNMo93A0oaro6696CQ5Q00xvCpoBQ==" crossorigin="anonymous" referrerpolicy="no-referrer">
</script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/highlight.js/11.11.1/languages/json.min.js" integrity="sha512-f2/ljYb/tG4fTHu6672tyNdoyhTIpt4N1bGrBE8ZjwIgrjDCd+r1jLpWCZ2Vym9PBWQy2Tl9022Pp2rMOMvH4g==" crossorigin="anonymous" referrerpolicy="no-referrer">
</script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/select2/4.0.9/js/select2.min.js" integrity="sha512-9p/L4acAjbjIaaGXmZf0Q2bV42HetlCLbv8EP0z3rLbQED2TAFUldVAezy7kumYqg5T8jHtDdlm1fgIsr5QzKg==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
<script>
    hljs.highlightAll();
    $(document).ready(function() {
        <?php if (session()->getFlashdata('success')): ?>
            toastr.success('<?= session()->getFlashdata('success') ?>');
        <?php endif; ?>

        <?php if (session()->getFlashdata('error')): ?>
            toastr.error('<?= session()->getFlashdata('error') ?>');
        <?php endif; ?>

        $('.select2').select2();
        $('.select-table').select2();
        $('.value-field').select2();
        $('.label-field').select2();

        $('select[name*="[type]"').change(function() {
            const fieldType = $(this).val();
            const chainedSelect = $(this).closest('.field-type').next('.chained-select');

            if (fieldType === 'select') {
                chainedSelect.show();
            } else {

```

```
        chainedSelect.hide();
    }
});

$('.select-source').change(function() {
    const selectSource = $(this).val();
    const manualOptions = $(this).closest('.chained-select').next('.manual-opt
ions');
    const tableOptions = $(this).closest('.chained-select').nextAll('.table-op
tions').first();

    if (selectSource === 'manual') {
        manualOptions.show();
        tableOptions.hide();
    } else if (selectSource === 'table') {
        manualOptions.hide();
        tableOptions.show();
    } else {
        manualOptions.hide();
        tableOptions.hide();
    }
});

$('.select-table').change(function() {
    const table = $(this).val();
    const valueField = $(this).closest('.table-options').find('.value-field');
    const labelField = $(this).closest('.table-options').find('.label-field');

    if (table) {
        $.ajax({
            url: '/forms/getfields',
            method: 'POST',
            data: {
                table: table
            },
            success: function(response) {
                try {
```

```

const fields = JSON.parse(response);
if (fields.error) {
    alert(fields.error);
    return;
}
let valueOptions = '<option value="">Pilih Field</option>';

let labelOptions = '<option value="">Pilih Field</option>';

fields.forEach(field => {
    valueOptions += `<option value="${field.name}">${field.name}</option>`;
    labelOptions += `<option value="${field.name}">${field.name}</option>`;
});

valueField.html(valueOptions).prop('disabled', false);
labelField.html(labelOptions).prop('disabled', false);
} catch (error) {
    console.error('Error parsing JSON:', error);
    alert('Gagal memproses data dari server.');
```

```
</script>
<?= $this->endSection() ?>
```

Views/formbuilder/preview.php

```
<!-- form_builder/preview.php -->
<!-- https://chat.deepseek.com/a/chat/s/3a5a90e4-4831-4217-8b3e-106f881fd660 -->
<?= $this->extend('themes/default/layouts/main') ?>
<?= $this->section('content') ?>
<form action="/forms/submit/<?= $form['id'] ?>" method="post">
    <?php foreach (json_decode($form['structure'], true) as $field): ?>
        <div class="mb-3">
            <label><?= $field['label'] ?></label>
            <?php if ($field['type'] === 'text'): ?>
                <input type="text" name="<?= $field['name'] ?? $field['label'] ?>" cla
ss="form-control" <?= isset($field['required']) && $field['required'] ? 'required' :
'' ?>>

                <?php elseif ($field['type'] === 'textarea'): ?>
                    <textarea name="<?= $field['name'] ?? $field['label'] ?>" class="form-
control" <?= isset($field['required']) && $field['required'] ? 'required' : '' ?>></te
xtarea>

                    <?php elseif ($field['type'] === 'select'): ?>
                        <select name="<?= $field['name'] ?? $field['label'] ?>" class="form-co
ntrol" <?= isset($field['required']) && $field['required'] ? 'required' : '' ?>>
                            <?php if (!empty($field['options'])): ?>
                                <?php foreach (explode(',', $field['options'])) as $option): ?>
                                    <option value="<?= trim($option) ?>"><?= trim($option) ?>
</option>

                                <?php endforeach; ?>
                            <?php endif; ?>
                        </select>
                    <?php elseif ($field['type'] === 'checkbox'): ?>
                        <?php if (!empty($field['options'])): ?>
                            <?php foreach (explode(',', $field['options'])) as $option): ?>
                                <div class="form-check">
                                    <input type="checkbox" name="<?= $field['name'] ?? $field
['label'] ?>[]" value="<?= trim($option) ?>" class="form-check-input">
```

```

        <label class="form-check-label"><?= trim($option) ?></label>
    l>

        </div>

        <?php endforeach; ?>

        <?php endif; ?>

        <?php elseif ($field['type'] === 'radio'): ?>
            <?php if (!empty($field['options'])): ?>
                <?php foreach (explode(',', $field['options'])) as $option): ?>
                    <div class="form-check">
                        <input type="radio" name="<?= $field['name'] ?? $field['label'] ?>" value="<?= trim($option) ?>" class="form-check-input">
                            <label class="form-check-label"><?= trim($option) ?></label>
                        l>

                            </div>

                        <?php endforeach; ?>

                    <?php endif; ?>

                <?php endif; ?>
            </div>

        <?php endforeach; ?>

        <button type="submit" class="btn btn-primary">Submit</button>
    </form>

    <?= $this->endSection() ?>

    <?= $this->section('scripts') ?>

    <script>
    $(document).ready(function() {
        <?php if(session()->getFlashdata('success')): ?>
            toastr.success('<?= session()->getFlashdata('success') ?>');
        <?php endif; ?>

        <?php if(session()->getFlashdata('error')): ?>
            toastr.error('<?= session()->getFlashdata('error') ?>');
        <?php endif; ?>
    });
    </script>

    <?= $this->endSection() ?>

```

Services/CrudGenerator.php

```
<?php

namespace App\Services;

use App\Template\ModelTemplate;
use App\Template\ControllerTemplate;
use App\Template\Views\IndexTemplate;
use App\Template\Views\CreateTemplate;
use App\Template\Views>EditTemplate;
use App\Template\Views\AjaxIndexTemplate;
use App\Template\Views\AjaxCreateTemplate;
use App\Template\Views\AjaxEditTemplate;
use App\Template\RoutesTemplate;
use App\Models\FormModel;
use CodeIgniter\Log\Logger;
use CodeIgniter\Files\Exceptions\FileNotFoundException;

class CrudGenerator
{
    protected $formModel;
    protected $logger;

    public function __construct(FormModel $formModel, Logger $logger)
    {
        $this->formModel = $formModel;
        $this->logger = $logger;
    }
    // app/Services/CrudGenerator.php
    public function generateFromForm(array $form)
    {
        try {
            $tableName = $form['table_name'];
            $fields = json_decode($form['structure'], true);
            $routes = $form['routes']; // Ambil routes dari form
```

```

        if (json_last_error() !== JSON_ERROR_NONE) {
            throw new \RuntimeException("Invalid JSON structure for table: $tableName");
        }

        $primaryKey = $this->getPrimaryKey($tableName);

        $this->generateModel($tableName, $fields, $primaryKey);
        $this->generateController($tableName,$primaryKey,$routes);
        $this->generateViewsAjax($tableName, $fields, $primaryKey,$routes);
        $this->generateRoutes($tableName, $routes);

        $this->logger->info("CRUD generated for table: $tableName");
        return true;
    } catch (\RuntimeException | FileNotFoundException $e) {
        $this->logger->error($e->getMessage());
        throw $e;
    }
}

public function generateFromFormsTable()
{
    try {
        $forms = $this->formModel->findAll();

        if (empty($forms)) {
            throw new \RuntimeException("No forms found in the database.");
        }

        foreach ($forms as $form) {
            $this->generateFromForm($form); // Reuse generateFromForm for each form
        }

        $this->logger->info("CRUDL generated successfully for all tables.");
        return true;
    } catch (\RuntimeException | FileNotFoundException $e) {
        $this->logger->error($e->getMessage());
    }
}

```

```

        return false;
    }
}

```

```

protected function generateModel($tableName, $fields, $primaryKey)
{
    helper('filesystem'); // Load helper filesystem

    // Siapkan aturan validasi
    $validationRules = [];
    foreach ($fields as $fieldName => $fieldConfig) {
        if (!empty($fieldConfig['validation_rule'])) {
            $validationRules[$fieldName] = $fieldConfig['validation_rule'];
        }
    }

    // Generate kode model menggunakan ModelTemplate
    $modelCode = ModelTemplate::generate($tableName, $fields, $primaryKey, $validationRules);
    $modelPath = APPPATH . "Models/{$tableName}Model.php";

    if (!write_file($modelPath, $modelCode)) {
        throw new FileNotFoundException("Failed to create model: $tableName");
    }
}

```

```

protected function generateController($tableName, $primaryKey, $routes)
{
    helper('filesystem'); // Load helper filesystem

    // Generate kode controller menggunakan ControllerTemplate
    $controllerCode = ControllerTemplate::generate($tableName, $primaryKey, $routes);
    $controllerPath = APPPATH . "Controllers/{$tableName}Controller.php";

    if (!write_file($controllerPath, $controllerCode)) {

```



```

        throw new FileNotFoundException("Failed to create controller: $tableName
e");
    }
}

protected function generateViewsAjax($tableName, $fields, $primaryKey,$routes)
{
    helper('filesystem'); // Load helper filesystem
    $indexView = AjaxIndexTemplate::generate($tableName, $fields, $primaryKey,$rou
tes);
    $createView = AjaxCreateTemplate::generate($tableName, $fields,$routes);
    $editView = AjaxEditTemplate::generate($tableName, $fields, $primaryKey,$route
s);

    $viewDir = APPPATH . "Views/{$tableName}";
    if (!is_dir($viewDir) && !mkdir($viewDir, 0777, true)) {
        throw new FileNotFoundException("Failed to create view directory: $tableNa
me");
    }

    if (!write_file("$viewDir/index.php", $indexView)) {
        throw new FileNotFoundException("Failed to create index view: $tableName
e");
    }

    if (!write_file("$viewDir/create.php", $createView)) {
        throw new FileNotFoundException("Failed to create create view: $tableName
e");
    }

    if (!write_file("$viewDir/edit.php", $editView)) {
        throw new FileNotFoundException("Failed to create edit view: $tableName");
    }
}

protected function generateViews($tableName, $fields, $primaryKey)
{
    helper('filesystem'); // Load helper filesystem

```

```

$indexView = IndexTemplate::generate($tableName, $fields, $primaryKey);
$createView = CreateTemplate::generate($tableName, $fields);
$editView = EditTemplate::generate($tableName, $fields, $primaryKey);

$viewDir = APPPATH . "Views/{$tableName}";
if (!is_dir($viewDir) && !mkdir($viewDir, 0777, true)) {
    throw new FileNotFoundException("Failed to create view directory: $tableName");
}

if (!write_file("$viewDir/index.php", $indexView)) {
    throw new FileNotFoundException("Failed to create index view: $tableName");
}

if (!write_file("$viewDir/create.php", $createView)) {
    throw new FileNotFoundException("Failed to create create view: $tableName");
}

if (!write_file("$viewDir/edit.php", $editView)) {
    throw new FileNotFoundException("Failed to create edit view: $tableName");
}
}

protected function generateRoutes($tableName,$routes)
{
    helper('filesystem'); // Load helper filesystem

    // Generate routes code
    $routesCode = RoutesTemplate::generate($tableName,$routes);
    // Define generated routes file path
    // Pastikan direktori Generated ada
    $generatedDir = APPPATH . "Config/Generated";
    if (!is_dir($generatedDir)) {
        mkdir($generatedDir, 0777, true);
    }
}

```

```

    }

    $generatedRoutesPath = APPPATH . "Config/Generated/Routes_Generated.php";

    // Debug routes code
    log_message('debug', 'Routes Code: ' . $routesCode);
    // Debug generated routes path
    log_message('debug', 'Generated Routes Path: ' . $generatedRoutesPath);

    // Ensure the directory is writable
    if (!is_writable(dirname($generatedRoutesPath))) {
        throw new FileNotFoundException("Directory is not writable: " . dirname($g
eneratedRoutesPath));
    }

    // Write the routes code to the generated file
    if (!write_file($generatedRoutesPath, $routesCode)) {
        throw new FileNotFoundException("Failed to write routes for: $tableName");
    }
}

public function getPrimaryKey($tableName)
{
    $db = \Config\Database::connect();
    $fields = $db->getFieldData($tableName);

    foreach ($fields as $field) {
        if ($field->primary_key) {
            return $field->name; // Mengembalikan nama Primary Key
        }
    }

    return 'id'; // Default jika tidak ada Primary Key
}
}

```

Template/ControllerTemplate.php

```
<?php
```

```
namespace App\Template;
```

```
class ControllerTemplate {
```

```
    public static function generate($tableName,$primaryKey,$routes) {
```

```
        return <<<EOD
```

```
        <?php
```

```
        namespace App\Controllers;
```

```
        use App\Controllers\BaseController;
```

```
        use CodeIgniter\HTTP\ResponseInterface;
```

```
        use CodeIgniter\HTTP\RequestInterface;
```

```
        use App\Models\{$tableName}Model;
```

```
        use CodeIgniter\Database\BaseConnection;
```

```
        class {$tableName}Controller extends BaseController {
```

```
            protected \{$tableName}Model;
```

```
            public function __construct() {
```

```
                \{$this->{$tableName}Model = new {$tableName}Model();
```

```
            }
```

```
            public function index() {
```

```
                \{$data['{$tableName}'] = \{$this->{$tableName}Model->findAll();
```

```
                return view('{$tableName}/index', \{$data);
```

```
            }
```

```
            public function create() {
```

```
                return view('{$tableName}/create');
```

```
            }
```

```
            public function store() {
```

```
                \{$data = \{$this->request->getPost();
```

```
                \{$this->{$tableName}Model->insert(\{$data);
```

```
                return redirect()->to('/{$routes}');
```

```

    }

    public function storeAjax() {
        \ $data = \ $this->request->getPost();
        \ $this->{ $tableName }Model->insert(\ $data);
        return \ $this->response->setJSON(['success' => true]);
    }

    public function edit(\ $id) {
        \ $data[' $tableName '] = \ $this->{ $tableName }Model->find(\ $id);
        return view(' $tableName /edit', \ $data);
    }

    public function update(\ $id) {
        \ $data = \ $this->request->getPost();
        if (!\ $this->validate(\ $this->{ $tableName }Model->validationRules)) {
            return redirect()->back()->withInput()->with('errors', \ $this->validator->getErrors());
        }
        \ $this->{ $tableName }Model->update(\ $id, \ $data);
        return redirect()->to('/ $routes ');
    }

    public function updateAjax(\ $id) {
        \ $data = \ $this->request->getPost(); // Ambil data dari form
        if (!\ $this->validate(\ $this->{ $tableName }Model->validationRules)) {
            return redirect()->back()->withInput()->with('errors', \ $this->validator->getErrors());
        }
        if (\ $this->{ $tableName }Model->update(\ $id, \ $data)) {
            return \ $this->response->setJSON(['success' => true]);
        } else {
            return \ $this->response->setJSON(['success' => false]);
        }
    }

    public function delete(\ $id) {

```

```

        \$this->{$tableName}Model->delete(\$id);
        return redirect()->to('/$routes');
    }
    public function deleteAjax(\$id) {
        \$this->{$tableName}Model->delete(\$id);
        return \$this->response->setJSON(['success' => true]);
    }
    public function data() {
        \$model = new {$tableName}Model();
        \$data = \$model->findAll();
        return \$this->response->setJSON(\$data);
    }
}
EOD;
}
}

```

Template/ModelTemplate.php

```

<?php

namespace App\Template;

class ModelTemplate {
    public static function generate($tableName, $fields, $primaryKey, $validationRules) {
        $allowedFields = implode("'", "'", array_keys($fields));
        $validationRulesJson = json_encode($validationRules, JSON_PRETTY_PRINT);

        return <<<EOD
<?php

namespace App\Models;

use CodeIgniter\Model;

class {$tableName}Model extends Model {

```

```

protected \$table = '$tableName';
protected \$primaryKey = '$primaryKey';
protected \$allowedFields = ['$allowedFields'];
protected \$useSoftDeletes = true;
protected bool \$allowEmptyInserts = false;
protected bool \$updateOnlyChanged = true;
protected \$protectFields = true;
protected array \$casts = [];
protected array \$castHandlers = [];

// Dates

protected \$useTimestamps = true; //or true use timestamps
protected \$dateFormat = 'datetime';
protected \$createdField = 'created_at';
protected \$updatedField = 'updated_at';
protected \$deletedField = 'deleted_at';

// Validation
protected \$validationRules = $validationRulesJson;
protected \$validationMessages = [];
protected \$skipValidation = false;
protected \$cleanValidationRules = true;

// Callbacks
protected \$allowCallbacks = true;
protected \$beforeInsert = [];
protected \$afterInsert = [];
protected \$beforeUpdate = [];
protected \$afterUpdate = [];
protected \$beforeFind = [];
protected \$afterFind = [];
protected \$beforeDelete = [];
protected \$afterDelete = [];
}
EOD;
}
}

```

Template/RoutesTemplate.php

```
<?php

namespace App\Template;

class RoutesTemplate {
    public static function generate($tableName,$routes) {
        return <<<EOD

        \ $routes->group('/$routes', function (\ $routes) {
            \ $routes->get('/', '{$tableName}Controller::index'); // List
            \ $routes->get('create', '{$tableName}Controller::create'); // Create form
            \ $routes->post('store', '{$tableName}Controller::store'); // Save new record

            \ $routes->post('storeAjax', '{$tableName}Controller::storeAjax'); // Save new record

            \ $routes->get('edit/(:num)', '{$tableName}Controller::edit/\$1'); // Edit form

            \ $routes->post('update/(:num)', '{$tableName}Controller::update/\$1'); // Update record

            \ $routes->post('updateAjax/(:num)', '{$tableName}Controller::updateAjax/\$1'); // Update record

            \ $routes->delete('delete/(:num)', '{$tableName}Controller::deleteAjax/\$1'); // Delete record

            \ $routes->get('data', '{$tableName}Controller::data');
        });
        EOD;
    }
}
```

Template/views/CreateTemplate.php

```
<?php
```



```

namespace App\Template\Views;

class CreateTemplate
{
    public static function generate($table, $fields)
    {
        $formFields = '';

        foreach ($fields as $key => $field) {
            $formFields .= "<div class=\"mb-3\">\n";
            $formFields .= "    <label>{$field['label']}</label>\n";
            if ($field['type'] === 'text') {
                $formFields .= "    <input type=\"text\" name=\"$key\" class=\"form-control\" required>\n";
            } elseif ($field['type'] === 'select') {
                $formFields .= "    <select name=\"$key\" class=\"form-control\" required>\n";

                $formFields .= "        <option value=\"Male\">Male</option>\n";
                $formFields .= "        <option value=\"Female\">Female</option>\n";
                $formFields .= "    </select>\n";
            } elseif ($field['type'] === 'checkbox') {
                $formFields .= "    <input type=\"checkbox\" name=\"$key\" class=\"form-check-input\">\n";
            } elseif ($field['type'] === 'radio') {
                $formFields .= "    <input type=\"radio\" name=\"$key\" class=\"form-check-input\">\n";
            }
            $formFields .= "</div>\n";
        }

        return <<<EOD
        <!DOCTYPE html>
        <html lang="en">
        <head>
            <meta charset="UTF-8">
            <meta name="viewport" content="width=device-width, initial-scale=1.0">
            <title>Create $table</title>
    
```

```

        <link href="https://cdn.jsdelivr.net/npm/[email protected]/dist/css/bootstrap.min.css" rel="stylesheet">
    </head>
    <body>
        <div class="container mt-5">
            <h1>Create $table</h1>
            <form action="/$table/store" method="post">
                $formFields
                <button type="submit" class="btn btn-primary">Save</button>
            </form>
        </div>
    </body>
</html>
EOD;
    }
}

```

Template/views/IndexTemplate.php

```

<?php

namespace App\Template\Views;

class IndexTemplate {
    public static function generate($tableName, $fields,$primaryKey) {
        $headers = '';
        $columns = '';

        foreach ($fields as $key => $field) {
            $headers .= "<th>{$field['label']}</th>\n";
            $columns .= "<td><?= \${row['$key']} ?></td>\n";
        }

        return <<<EOD
        <!DOCTYPE html>
        <html lang="en">
    
```

```

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>{$tableName} List</title>
    <link href="https://cdn.jsdelivr.net/npm/[email protected]/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
    <div class="container mt-5">
        <h1>{$tableName} List</h1>
        <a href="/{$tableName}/create" class="btn btn-primary mb-3">Create New
</a>

        <table id="dataTable" class="table table-striped table-responsive" style="width:100%">
            <thead>
                <tr>
                    <th>ID</th>
                    {$headers}
                    <th>Actions</th>
                </tr>
            </thead>
            <tbody>
                <?php foreach (\{$tableName} as $row): ?>
                    <tr>
                        <td><?= $row['$primaryKey'] ?></td>
                        {$columns}
                        <td>
                            <a href="/{$tableName}/edit/<?= $row['$primaryKey'] ?>" class="btn btn-sm btn-warning">Edit</a>
                            <a href="/{$tableName}/delete/<?= $row['$primaryKey'] ?>" class="btn btn-sm btn-danger" onclick="return confirm('Are you sure?')">Delete</a>
                        </td>
                    </tr>
                <?php endforeach; ?>
            </tbody>
        </table>

```

```

        </div>
    </body>
</html>
EOD;
}
}

```

Template/views/EditTemplate.php

```

<?php

namespace App\Template\Views;

class EditTemplate {
    public static function generate($table, $fields, $primaryKey) {
        $formFields = '';

        foreach ($fields as $key => $field) {
            $formFields .= "<div class=\"mb-3\">\n";
            $formFields .= "    <label>{$field['label']}</label>\n";
            if ($field['type'] === 'text') {
                $formFields .= "        <input type=\"text\" name=\"\$key\" class=\"form-control\" value=\"<? = \${$table}['$key'] ?>\" required>\n";
            } elseif ($field['type'] === 'select') {
                $formFields .= "        <select name=\"\$key\" class=\"form-control\" required>\n";

                $formFields .= "            <option value=\"Male\" <? = \${$table}['$key'] === 'Male' ? 'selected' : '' ?>>Male</option>\n";
                $formFields .= "            <option value=\"Female\" <? = \${$table}['$key'] === 'Female' ? 'selected' : '' ?>>Female</option>\n";
                $formFields .= "        </select>\n";
            } elseif ($field['type'] === 'checkbox') {
                $formFields .= "        <input type=\"checkbox\" name=\"\$key\" class=\"form-check-input\" <? = \${$table}['$key'] ? 'checked' : '' ?>>\n";
            } elseif ($field['type'] === 'radio') {
                $formFields .= "        <input type=\"radio\" name=\"\$key\" class=\"form-check-input\" <? = \${$table}['$key'] ? 'checked' : '' ?>>\n";
            }
        }
    }
}

```

```

    }

    $formFields .= "</div>\n";
}

return <<<EOD
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Edit $table</title>
    <link href="https://cdn.jsdelivr.net/npm/[email protected]/dist/css/bootstrap.min.
css" rel="stylesheet">
</head>
<body>
    <div class="container mt-5">
        <h1>Edit $table</h1>
        <form action="/$table/update/<?= \${$table}['$primaryKey'] ?>" method="post">
            $formFields
            <button type="submit" class="btn btn-primary">Update</button>
        </form>
    </div>
</body>
</html>
EOD;
    }
}

```

Template/views/AjaxEditTemplate.php

```

<?php

namespace App\Template\Views;

class AjaxEditTemplate
{
    public static function generate($table, $fields, $primaryKey, $routes)

```

```

{
    $formFields = '';

    foreach ($fields as $key => $field) {
        $formFields .= "<div class=\"mb-3\">\n";
        $formFields .= "    <label>{$field['label']}</label>\n";
        if ($field['type'] === 'text') {
            $formFields .= "        <input type=\"text\" name=\"\$key\" class=\"form-control\" value=\"<?=\${$table}['$key'] ?>\" required>\n";
        } elseif ($field['type'] === 'select') {
            $formFields .= "        <select name=\"\$key\" class=\"form-control\" required>\n";

            $formFields .= "            <option value=\"Male\" <?=\${$table}['$key'] === 'Male' ? 'selected' : '' ?>>Male</option>\n";
            $formFields .= "            <option value=\"Female\" <?=\${$table}['$key'] === 'Female' ? 'selected' : '' ?>>Female</option>\n";
            $formFields .= "        </select>\n";
        } elseif ($field['type'] === 'checkbox') {
            $formFields .= "        <input type=\"checkbox\" name=\"\$key\" class=\"form-check-input\" <?=\${$table}['$key'] ? 'checked' : '' ?>>\n";
        } elseif ($field['type'] === 'radio') {
            $formFields .= "        <input type=\"radio\" name=\"\$key\" class=\"form-check-input\" <?=\${$table}['$key'] ? 'checked' : '' ?>>\n";
        }
        $formFields .= "</div>\n";
    }

    return <<<EOD
    <?=\$this->extend('themes/default/layouts/main') ?>
    <?=\$this->section('content') ?>
        <div class="container mt-5">
            <h1>Edit $table</h1>
            <form id="editForm">
                $formFields
                <button type="submit" class="btn btn-primary">Update</button>
            </form>
        </div>

```

```

<?= \$this->endSection() ?>
<?= \$this->section('scripts') ?>
    <!-- SweetAlert2 -->
    <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
    <!-- Toastr -->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/toast
r.min.js"></script>
    <!-- jQuery (required for AJAX) -->
    <script src="https://code.jquery.com/jquery-3.7.0.min.js"></script>
    <!-- Custom Script -->
    <script>
        $(document).ready(function() {
            const baseUrl = '/{{$routes}}';

            $('#editForm').on('submit', function(e) {
                e.preventDefault(); // Mencegah form submit default

                // Kirim data via AJAX
                $.ajax({
                    url: baseUrl + '/update/' + <?= \${{$table}}['$primaryKey']
?>,

                    method: 'POST',
                    data: $(this).serialize(), // Serialize form data
                    success: function(response) {
                        // Tampilkan notifikasi sukses
                        toastr.success('Data updated successfully!');
                        // Redirect ke halaman list setelah 2 detik
                        setTimeout(function() {
                            window.location.href = baseUrl;
                        }, 2000);
                    },
                    error: function(xhr) {
                        // Tampilkan notifikasi error
                        toastr.error('Failed to update data.');
```

```

        });
    </script>
    <?= \$this->endSection() ?>
EOD;
    }
}

```

Template/views/AjaxCreateTemplate.php

```

<?php

namespace App\Template\Views;

class AjaxCreateTemplate {
    public static function generate($table, $fields,$routes) {
        $formFields = '';

        foreach ($fields as $key => $field) {
            $formFields .= "<div class=\"mb-3\">\n";
            $formFields .= "    <label>{$field['label']}</label>\n";
            if ($field['type'] === 'text') {
                $formFields .= "        <input type=\"text\" name=\"$key\" class=\"form-control\" required>\n";
            } elseif ($field['type'] === 'select') {
                $formFields .= "        <select name=\"$key\" class=\"form-control\" required>\n";

                $formFields .= "            <option value=\"Male\">Male</option>\n";
                $formFields .= "            <option value=\"Female\">Female</option>\n";
                $formFields .= "        </select>\n";
            } elseif ($field['type'] === 'checkbox') {
                $formFields .= "        <input type=\"checkbox\" name=\"$key\" class=\"form-check-input\">\n";
            } elseif ($field['type'] === 'radio') {
                $formFields .= "        <input type=\"radio\" name=\"$key\" class=\"form-check-input\">\n";
            }
        }
    }
}

```



```

        $formFields .= "</div>\n";
    }

    return <<<EOD
<?= \$this->extend('themes/default/layouts/main') ?>

<?= \$this->section('content') ?>
    <div class="container mt-5">
        <h1>Create $table</h1>
        <form id="createForm">
            $formFields
            <button type="submit" class="btn btn-primary">Save</button>
        </form>
    </div>
<?= \$this->endSection() ?>
<?= \$this->section('scripts') ?>
    <!-- Bootstrap 5 JS -->
    <script src="https://cdn.jsdelivr.net/npm/[email protected]/dist/js/bootstrap.bundle.min.js"></script>
    <!-- jQuery -->
    <script src="https://code.jquery.com/jquery-3.7.0.min.js"></script>
    <!-- SweetAlert2 JS -->
    <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
    <!-- Custom Script -->
    <script>
        $(document).ready(function() {
            const baseUrl = '/{$routes}';

            $('#createForm').submit(function(e) {
                e.preventDefault();
                const formData = $(this).serialize();

                $.ajax({
                    url: baseUrl + '/store',
                    method: 'POST',
                    data: formData,
                    success: function(response) {

```

```

        Swal.fire({
            icon: 'success',
            title: 'Success!',
            text: 'Data has been created.',
        }).then(() => {
            window.location.href = baseUrl;
        });
    },
    error: function() {
        Swal.fire({
            icon: 'error',
            title: 'Error!',
            text: 'Failed to create data.',
        });
    }
});
});
});
</script>
<?= \$this->endSection() ?>
EOD;
}
}

```

Template/views/AjaxIndexTemplate.php

```

<?php

namespace App\Template\Views;

class AjaxIndexTemplate {
    public static function generate($tableName, $fields,$primaryKey,$routes) {
        $headers = '';
        $columns = '';

        foreach ($fields as $key => $field) {
            $headers .= "<th>{$field['label']}</th>\n";

```

```

        $columns .= "{ data: '{$key}' },\n";
    }

    return <<<EOD
<?= \$this->extend('themes/default/layouts/main') ?>

<?= \$this->section('content') ?>
<div class="container mt-5">
    <h1>{$tableName} List</h1>
    <button id="btn-create" class="btn btn-primary mb-3">Create New</button>
    <table id="dataTable" class="table table-striped" style="width:100%">
        <thead>
            <tr>
                <th>ID</th>
                {$headers}
                <th>Actions</th>
            </tr>
        </thead>
        <tbody>
            <!-- Data akan diisi oleh AJAX -->
        </tbody>
    </table>
</div>
<?= \$this->endSection() ?>
<?= \$this->section('scripts') ?>

<script>
    $(document).ready(function() {
        const baseUrl = '/{$routes}';
        // Inisialisasi DataTables
        if ($.fn.DataTable.isDataTable('#dataTable')) {
            // Hancurkan instance DataTables yang sudah ada
            $('#dataTable').DataTable().destroy();
        }
        const table = $('#dataTable').DataTable({
            ajax: {
                url: baseUrl + '/data',

```

```

        method: 'GET',
        dataSrc: ''
    },
    columns: [
        { data: '$primaryKey' },
        {$columns}
        {
            data: null,
            render: function(data) {
                return `
                    <button class="btn btn-sm btn-warning btn-edit" data-i
d="\${data.$primaryKey}">Edit</button>
                    <button class="btn btn-sm btn-danger btn-delete" data-
id="\${data.$primaryKey}">Delete</button>
                `;
            }
        }
    ]
});

// Tombol Create
$('#btn-create').click(function() {
    window.location.href = baseUrl + '/create';
});

// Tombol Edit
$('#dataTable').on('click', '.btn-edit', function() {
    const id = $(this).data('id');
    window.location.href = baseUrl + `/edit/${id}`;
});

// Tombol Delete
$('#dataTable').on('click', '.btn-delete', function() {
    const id = $(this).data('id');
    Swal.fire({
        title: 'Are you sure?',
        text: "You won't be able to revert this!",

```

```

        icon: 'warning',
        showCancelButton: true,
        confirmButtonColor: '#d33',
        cancelButtonColor: '#3085d6',
        confirmButtonText: 'Yes, delete it!'
    }).then((result) => {
        if (result.isConfirmed) {
            $.ajax({
                url: baseUrl + `/delete/${id}`,
                method: 'DELETE',
                success: function(response) {
                    table.ajax.reload();
                    Swal.fire('Deleted!', 'Your data has been deleted.',
'success');

                },
                error: function() {
                    Swal.fire('Error!', 'Failed to delete data.', 'erro
r');

                }
            });
        }
    });
});
});
});
</script>
<?= \$this->endSection() ?>
EOD;
}
}

```