

LAPORAN

“Tugas Final Project Pemrograman Framework”

Praktik Study Kasus Perpustakaan STMIK Palangkaraya

Dosen Pengampu: Herkules S.Kom.,M.Cs



Disusun oleh :

Kelompok Framework tim 5

Citra Apri Herlina (C2055201011)

Crisnawati (C2055201013)

Dieni Isa Lawineini (C2055201012)

Normalia (C2055201008)

Sepia Dori (C2055201007)

KELAS A

SEMESTER III

JURUSAN TEKNIK INFORMATIKA

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER

(STMIK) PALANGKARAYA

2021

KATA PENGANTAR

Puji Syukur kami panjatkan kepada Allah SWT atas segala rahmat dan karunia-Nya sehingga laporan final yang berjudul “Tugas Final Project Pemrograman Framework” dengan Study Kasus Perpustakaan STMIK Palangkaraya dapat kami selesaikan dengan baik.

Pada kesempatan ini kami berterimakasih kepada semua pihak yang telah memberi semangat maupun pengetahuan baru dalam pembuatan makalah ini. Kepada dosen pengampu yang telah memberikan banyak ilmu dan pengetahuan dalam matakuliah Pemrograman Framework dan kepada teman-teman seperjuangan yang memantu kami dalam berbagai hal.

Demikian laporan ini kami buat, apabila terdapat kesalahan dalam penulisan, atau pun adanya ketidaksesuaian materi yang kami angkat pada laporan ini, kami mohon maaf. Kami menerima kritik dan saran seluas-luasnya dari pembaca agar bisa membuat karya makalah yang lebih baik pada kesempatan berikutnya.

Palangka Raya, 31 Desember 2021

Penulis

DAFTAR ISI

LAPORAN..... 1

KATA PENGANTAR..... 2

DAFTAR ISI..... 3

BAB I PENDAHULUAN..... 4

BAB II ISI..... 5

BAB III PENUTUP..... 80

BAB I PENDAHULUAN

A. Latar Belakang

Framework adalah kerangka kerja untuk mengembangkan aplikasi berbasis website maupun desktop. Kerangka kerja disini sangat membantu developer dalam menuliskan sebuah kode program dengan lebih terstruktur dan tersusun rapi. Kerangka kerja diciptakan untuk mempermudah kinerja dari programmer. Sehingga, seorang programmer tidak perlu untuk menuliskan kode secara berulang – ulang. Karena didalamnya sendiri kita hanya perlu menyusun komponen – komponen pemrograman saja

CodeIgniter merupakan framework PHP yang menggunakan arsitektur berbasis MVC. Lebih spesifiknya, pada CI menggunakan komponen yang berbeda untuk mengelola tugas pengembangan website. Keunggulan dari CI terletak pada performanya yang ringan dan dapat diandalkan.

B. Tujuan Praktikum

Tujuan dari praktikum yang kami buat studi kasus perpustakaan STMIK dapat dikatakan paling utama adalah terbentuknya pengetahuan baru dalam tugas mata kuliah pemograman framework bagi kami yang berdasarkan prior knowledge (pengetahuan sebelumnya) dan eksperimen yang dimiliki dan dilakukan oleh mahasiswa.

C. Manfaat Praktikum

Pembelajaran dengan praktikum sangat efektif untuk mencapai seluruh ranah pengetahuan secara bersamaan, antara lain melatih agar teori dapat diterapkan pada permasalahan yang nyata (kognitif), melatih perencanaan kegiatan secara mandiri (afektif), dan melatih penggunaan instrumen tertentu (psikomotor). Salah satu kelebihan pembelajaran praktikum (Web Aplication Framework) adalah mahasiswa dapat berlatih secara trial and error, dapat mengulang-ulang kegiatan atau tindakan yang sama sampai benar-benar terampil . Hasil dari pratikum ini diharapkan dapat bermanfaat:

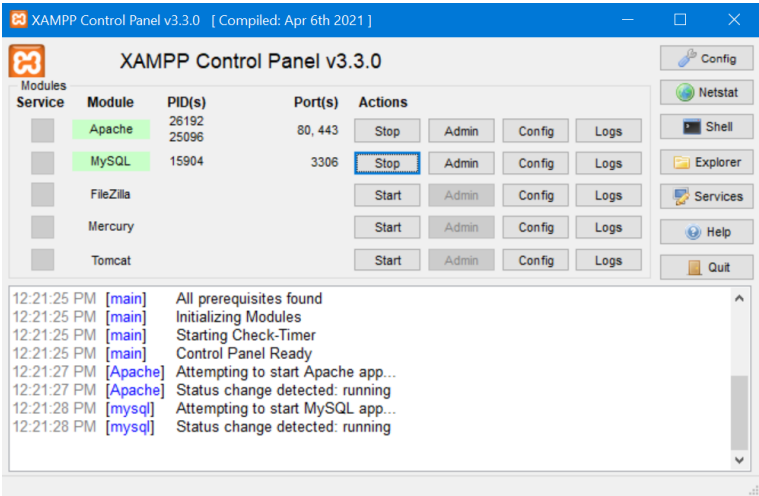
1. Bagi mahasiswa selanjutnya, dapat menjadi sumber referensi guna memperbaiki kelemahan dari mahasiswa ini. Atau dapat dijadikan sumber referensi untuk pratikum-pratikum baru.
2. Bagi pembaca, dapat menambah informasi mengenai praktikum pemograman framework perpustakaan dan memodifikasi dan berkreasi lagi

BAB II ISI

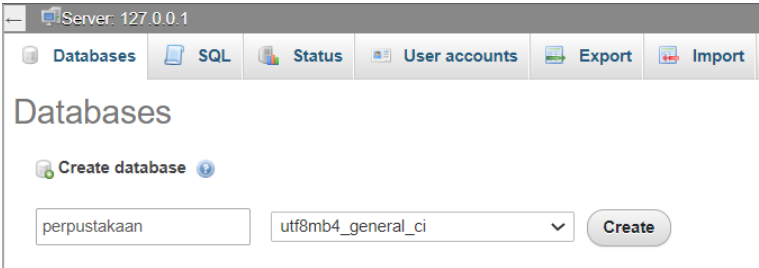
A. Langkah Membuat CRUD Table Buku

- Membuat dan mengkonfigurasi database

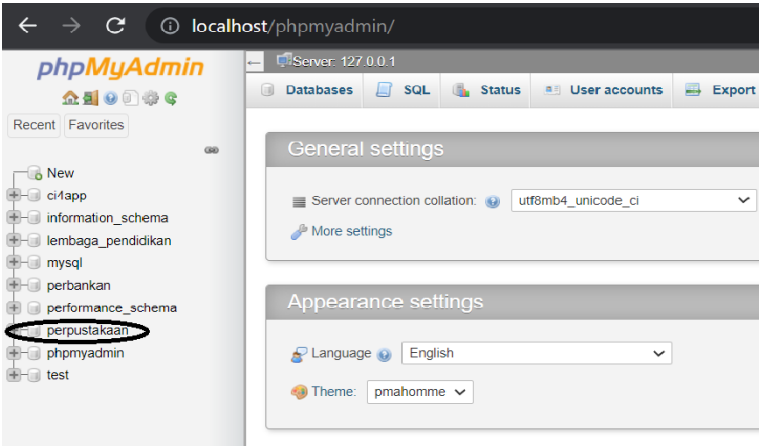
Aktifkan apache web server di control panel xampp terlebih dahulu seperti gambar dibawah ini :



Kemudian buat database di localhost/phpmyadmin pada browser dan buat database dengan nama perpustakaan.

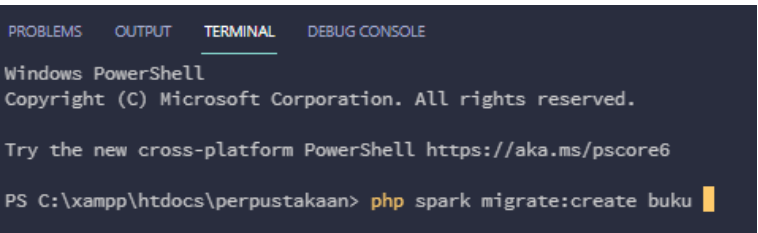


Hingga database yang berhasil dibuat muncul di daftar database local kita. Kemudian buat table product dengan cara ketik command berikut pada terminal yang ada di bawah ini :

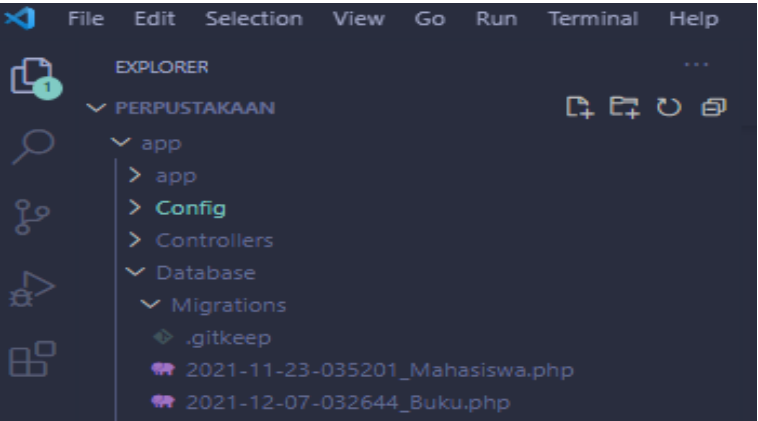


Kemudian buat table buku dengan cara ketik command berikut pada terminal VScode kita:

`php spark migrate:create buku`



Sekarang akan muncul file baru di folder app/Database/Migrations.



Lalu akan terbuat sebuah file didalam folder app/Database/Migrations, nama file tersebut akan diakhiri dengan nama Buku, buka file tersebut, dan isi dengan codenya seperti berikut ini :

```
<?php

namespace App\Database\Migrations;
use CodeIgniter\Database\Migration;
class Mahasiswa extends Migration
{
    public function up()
    {
        //
        $this->forge->addField([
            'id_buku' => [
                'type' => 'VARCHAR',
```

```
'constraint' => 11,  
  
],  
'judul' => [  
  'type' => 'VARCHAR',  
  'constraint' => '255',  
],  
'pengarang' => [  
  'type' => 'VARCHAR',  
  'constraint' => '255',  
],  
'penerbit' => [  
  'type' => 'VARCHAR',  
  'constraint' => '255',  
],  
'tahun' => [  
  'type' => 'CHAR',  
  'constraint' => '4',  
],  
'sampul' => [  
  'type' => 'VARCHAR',  
  'constraint' => '500',  
],  
'qty' => [  
  'type' => 'INT',  
  'constraint' => '11',  
],  
'keluar' => [  
  'type' => 'INT',  
  'constraint' => '11',  
],  
'created_at' => [  
  'type' => 'DATETIME',  
  'null' => true,  
],
```

```
'updated_at' => [
    'type' => 'DATETIME',
    'null' => true,
]
]);
$this->forge->addPrimaryKey('id_buku');
$this->forge->createTable('buku');

}

public function down()
{
    //
    $this->forge->dropTable('buku');
}
}
```

Silahkan di cek database pada phpmyadmin untuk melihat struktur table yang telah dibuat melalui fitur migration yang ada pada codeigniter 4

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id_buku	varchar(11)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	2 judul	varchar(255)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	3 pengarang	varchar(255)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	4 penerbit	varchar(255)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	5 tahun	char(4)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	6 sampul	varchar(500)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	7 qty	int(11)			No	None			Change Drop More
<input type="checkbox"/>	8 keluar	int(11)			No	None			Change Drop More
<input type="checkbox"/>	9 created_at	datetime			Yes	NULL			Change Drop More
<input type="checkbox"/>	10 updated_at	datetime			Yes	NULL			Change Drop More

Silahkan masuk kedalam project codeigniter 4 anda, melalui terminal, lalu jalankan perintah dibawah ini :

```
php spark make:seeder BukuSeeder
```

lalu tekan enter, jika berhasil maka akan terbuat sebuah file dengan nama **BukuSeeder.php** didalam folder **app\Database\Seeds** buka file tersebut, lalu tuliskan code didalam file BukuSeeder.php


```

<?php
namespace App\Database\Seeds;

use CodeIgniter\Database\Seeder;
use CodeIgniter\I18n\Time;

class BukuSeeder extends Seeder
{
    public function run()
    {
        //
        $data = [
            [
                'id_buku' => '01',
                'judul' => 'Teknologi Informasi',
                'pengarang' => 'Irvan Gunawan',
                'penerbit' => 'Elazabet',
                'tahun' => '2011',
                'created_at' => Time::now()
            ],
            [
                'id_buku' => '02',
                'judul' => 'Sistem Operasi',
                'pengarang' => 'Sisy',
                'penerbit' => 'ajis',
                'tahun' => '2012',
                'created_at' => Time::now()
            ],
            [
                'id_buku' => '03',
                'judul' => 'Sistem Informasi',
                'pengarang' => 'Ahmad Baihaqi',
                'penerbit' => 'Bibil',
                'tahun' => '2015',
                'created_at' => Time::now()
            ]
        ];
    }
}

```

```
    ]
    ];
    $this->db->table('buku')->insertBatch($data);

}
}
```

silahkan masuk ke folder project codeigniter melalui terminal, lalu jalankan perintah sebagai berikut ini :

```
php spark db:seed BukuSeeder
```

Lalu Silahkan cek tabel Buku pada phpmyadmin

☐ Show all

Number of rows: 25

Filter rows: Search this table

Sort by key: None

+ Options

	id_buku	judul	pengarang	penerbit	tahun	sampul
<input type="checkbox"/>  Edit  Copy  Delete 01		Pemograman Web Dengan Menggunakan PHP dan Framework...	Supono dan Vidiandry Putratama	Deepublish	2018	1641204273_b966e02dda29748424;
<input type="checkbox"/>  Edit  Copy  Delete 02		Jaringan Komputer	Edy Victor Haryanto	Andi	2012	1641204338_1cebef125d099d0725f6
<input type="checkbox"/>  Edit  Copy  Delete 03		Komputer Forensik	Feri Sulianta	Elex Media Komputindo	2013	1641204393_6972a9d920433e7c3ac

• **Membuat Model Buku**

Buat file model dengan nama **BukuModel.php**, simpan file tersebut didalam folder app/Models lalu isi file **BukuModel.php** dengan code sebagai berikut :

```
<?php

namespace App\Models;
use CodeIgniter\Model;
class BukuModel extends Model
{
    protected $table = "buku";
    protected $primaryKey = "id_buku";
    protected $returnType = "object";
    protected $useTimestamps = true;
```

```
protected $allowedFields = [
    'id_buku', 'judul', 'pengarang', 'penerbit',
    'tahun', 'sampul'
];
}
```

- **Membuat Controller Buku**

Buatlah sebuah controller dengan nama file Buku.php didalam folder app/Controllers, dan isi dengan code seperti berikut ini :

```
<?php

namespace App\Controllers;
use App\Models\MahasiswaModel;
class Buku extends BaseController
{
    protected $buku;
    function __construct()
    {
        $this->buku = new MahasiswaModel();
    }
    public function index()

    {
        $data['pageTitle'] = 'Daftar Buku';
        $data['buku'] = $this->buku->findAll();
        return view('dashboard/buku', $data);
    }
}
```

Kemudian edit kembali file routes.php kita yang ada pada folder apps/config/routes.php dengan menambahkan :

```
$routes->get('/buku', 'Buku::index',['filter' => 'auth']);
```

- **Membuat Halaman View Buku pada folder app/views/dashboard/**

Buku.php

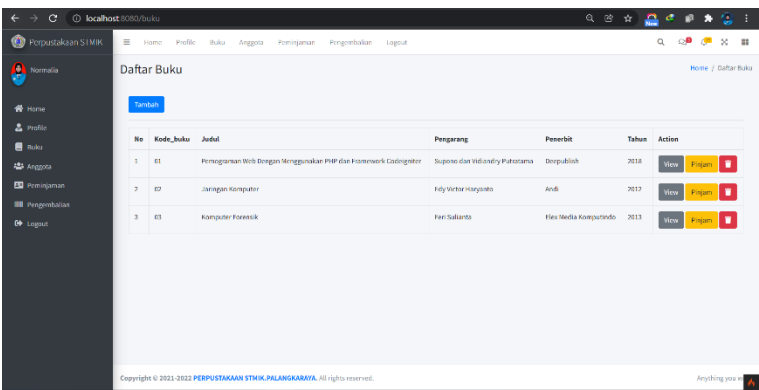
```
<?= $this->extend('layout/dashboard-layout'); ?>
<?= $this->section('content'); ?>
<!-- /.card-header -->
<div class="card-body">
    <?php if (!empty(session()->getFlashdata('message'))) : ?>
        <div class="alert alert-success alert-dismissible fade show"
role="alert">
            <?php echo session()->getFlashdata('message'); ?>
            <button type="button" class="close" data-dismiss="alert"
aria-label="Close">
                <span aria-hidden="true">&times;</span>
            </button>
        </div>
    <?php endif; ?>
    <a href="<?= base_url('/buku/create'); ?>" class="btn btn-
primary">Tambah</a>
    <hr />
    <table class="table table-bordered">
        <thead>
            <tr>
                <th>No</th>
                <th>Kode_Buku</th>
                <th>Judul Buku</th>
                <th>Pengarang</th>
                <th>Penerbit</th>
                <th>Tahun</th>
                <th>Action</th>
            </tr>
        </thead>
        <tbody>
            <?php
            $no = 1;
            foreach ($buku as $row) {
```

```

?>
    <tr>
        <td><?= $no++; ?></td>
        <td><?= $row->kode_buku; ?></td>
        <td><?= $row->judul; ?></td>
        <td><?= $row->pengarang; ?></td>
        <td><?= $row->penerbit; ?></td>
        <td><?= $row->tahun; ?></td>
        <td>
            <a title="View" href="<?= base_url('buku/view/$row-
>id_buku'); ?>" class="btn btn-info">view</a>
            <a title="Edit" href="<?= base_url('buku/edit/$row-
>id_buku'); ?>" class="btn btn-info">Edit</a>
            <a
                title="Delete"
                href="<?=
base_url('buku/delete/$row->id__buku') ?>"
                class="btn btn-
danger"
                onclick="return confirm('Apakah Anda yakin ingin
menghapus data ?')">Delete</a>
        </td>
    </tr>
<?php
}
?>
</tbody>
</table>
</div>
<!-- /.card-body -->
<?= $this->endSection('content'); ?>

```

Kemudian lihat hasil pada browser dengan mengetik <http://localhost:8080/buku>



• **Membuat Fitur Create**

Setelah dalam bagian sebelumnya kita telah menampilkan data mahasiswa, berikutnya kita akan membuat fitur untuk menambahkan data mahasiswa, langkah-langkahnya adalah sebagai berikut :

pada step awal kita bisa menambahkan routing untuk kebutuhan menampilkan form tambah data baru, silahkan buka file dengan nama Routes.php didalam folder **app/Config**.tambahkan code routing seperti berikut :

```
$routes->get('/buku/create', 'Buku::create');
$routes->get('/buku/store', 'Buku::store');
```

Kemudian buatlah form input data pada folder **app/views/dassboard/buku_create.php** untuk isi kode programnya dapat dilihat seperti dibawah ini :

buku_create.php

```
<?= $this->extend('layout/dashboard-layout'); ?>
<?= $this->section('content'); ?>
<div class="col-md-10">
    <!-- general form elements -->
    <div class="card card-primary">
        <div class="card-header">
            <h3 class="card-title">Tambah Data Buku</h3>
        </div>
        <!-- /.card-header -->
        <!-- form start -->
        <?php if (!empty(session()->getFlashdata('error'))): ?>
            <div class="alert alert-light alert-dismissible fade show"
role="alert">
```

```

<h4 class="alert-heading">Periksa Entrian Form</h4>
</hr />
<?php echo session()->getFlashdata('error'); ?>
<button type="button" class="close" data-dismiss="alert"
aria-label="Close">
    <span aria-hidden="true">&times;</span>
</button>
</div>
<?php endif; ?>
<form class="" method="post" action="<?=
base_url('buku/store') ?>" enctype="multipart/form-data">
    <?= csrf_field(); ?>
    <div class="card-body">
        <div class="form-group row">
            <label for="nim" class="col-sm-2 col-form-
label">Kode_Buku</label>
            <div class="col-sm-6">
                <input type="text" class="form-control" id="nim"
name="nim" placeholder="id buku" value="<?= old('kode buku');
?>">
            </div>
        </div>
        <div class="form-group row">
            <label for="nama" class="col-sm-2 col-form-
label">id_buku</label>
            <div class="col-sm-6">
                <input type="text" class="form-control" id="nama"
name="nama" placeholder="Nama Mahasiswa" value="<?=
old('nama'); ?>">
            </div>
        </div>
        <div class="form-group row">

```

```

        <label for="jurusan" class="col-sm-2 col-form-
label">Jurusan</label>

        <div class="col-sm-6">
            <select name="jurusan" class="form-control"
id="jurusan">
                <option value="pilih">Pilih Jurusan</option>
                <option value="Teknik Informatika">Teknik
Informatika</option>
                <option value="Sistem Informasi">Sistem
Informasi</option>
                <option value="Manajemen
Informatika">Manajemen Informatika</option>
            </select>
        </div>
    </div>

    <div class="form-group row">
        <label for="jenis_kelamin" class="col-sm-2 col-form-
label">Jenis Kelamin</label>
        <div class="col-sm-6">
            <select name="jenis_kelamin" class="form-control"
id="jenis_kelamin">
                <option value="pilih">Pilih</option>
                <option value="pria">Pria</option>
                <option value="wanita">Wanita</option>
            </select>
        </div>
    </div>

    <div class="form-group row">
        <label for="no_telp" class="col-sm-2 col-form-label">No
Telp</label>
        <div class="col-sm-6">

```



```

        <input type="text" class="form-control" id="no_telp"
name="no_telp" placeholder="No.Telp Mahasiswa" value="<?=
old('no_telp') ?>" />
    </div>

</div>

<div class="form-group row">

        <label for="email" class="col-sm-2 col-form-
label">Email</label>

        <div class="col-sm-6">
            <input type="text" class="form-control" id="email"
name="email" placeholder="Email Mahasiswa" value="<?=
old('email') ?>" />
        </div>
    </div>

    <div class="form-group row">

        <label for="alamat" class="col-sm-2 col-form-
label">Alamat</label>

        <div class="col-sm-10">
            <textarea class="form-control" name="alamat"
id="alamat"><?= old('alamat') ?></textarea>
        </div>
    </div>

    <div class="form-group row">
        <label for="photo" class="col-sm-2 col-form-
label">Photo</label>
        <div class="custom-file col-sm-6">
            <input type="file" class="custom-file-input" <?=
($validation->hasError('photo')) ? 'is-invalid' : ''; ?> id="foto"
name="photo">

```

```
<div class="invalid-feedback">
    <?= $validation->getError('photo'); ?>
</div>

<label class="custom-file-label" for="photo">Pilih
Photo </label>

</div>

</div>

</div>

<!-- /.card-body -->
<div class="card-footer">

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

</div>

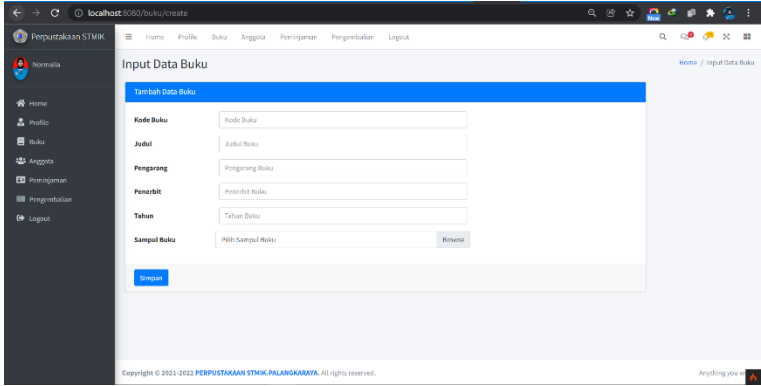
<!-- /.card-footer -->

</form>

</div>

<!-- /.card -->
</div>
<?= $this->endSection('content'); ?>
```

selanjutnya kita lihat hasilnya pada browser dengan mengetikkan <http://localhost:8080/buku/create>



- Membuat Fitur Pinjam

Kemudian buatlah form edit data pada folder **app/views/dassboard/buku_edit.php** untuk isi kode programnya dapat dilihat seperti dibawah ini :

```
<?= $this->extend('layout/dashboard-layout'); ?>
<?= $this->section('content'); ?>
<div class="col-md-10">
    <!-- general form elements -->
    <div class="card card-primary">
        <div class="card-header">
            <h3 class="card-title">Tambah Data Peminjaman</h3>
        </div>
        <!-- /.card-header -->
        <!-- form start -->
        <?php if (!empty(session()->getFlashdata('error'))): ?>
            <div class="alert alert-light alert-dismissible fade show"
role="alert">
                <h4 class="alert-heading">Periksa Entrian Form</h4>
                </hr />
                <?php echo session()->getFlashdata('error'); ?>
                <button type="button" class="close" data-dismiss="alert"
aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
        <?php endif; ?>
        <form class="" method="post" action="<?=
base_url('peminjaman/store') ?>" enctype="multipart/form-data">
            <?= csrf_field(); ?>
            <div class="card-body">
                <div class="form-group row">
                    <label for="kode_pinjam" class="col-sm-2 col-form-
label">Kode Pinjam</label>
                    <div class="col-sm-6">
```

```

        <input      type="text"      class="form-control"
id="kode_pinjam"  name="kode_pinjam"  placeholder="Kode
Pinjam" value="<?= old('kode_pinjam'); ?>">
    </div>
</div>

<div class="form-group row">
    <label for="tanggal_peminjaman" class="col-sm-2 col-
form-label">Tanggal Peminjaman</label>
    <div class="col-sm-6">
        <input      type="text"      class="form-control"
id="tanggal_peminjaman"      name="tanggal_peminjaman"
placeholder="Tanggal      Peminjaman"      value="<?=
old('tanggal_peminjaman'); ?>">
    </div>
</div>

<div class="form-group row">
    <label for="total_pinjam" class="col-sm-2 col-form-
label">Total Pinjam</label>
    <div class="col-sm-6">
        <input      type="text"      class="form-control"
id="total_pinjam"  name="total_pinjam"  placeholder="Total
Pinjam" value="<?= old('total_pinjam'); ?>">
    </div>
</div>

<div class="form-group row">
    <label for="nim_mahasiswa" class="col-sm-2 col-form-
label">Nim</label>
    <div class="col-sm-6">
        <input      type="text"      class="form-control"
id="nim_mahasiswa"  name="nim_mahasiswa"  placeholder="Nim
Mahasiswa" value="<?= old('nim_mahasiswa'); ?>">
    </div>

```

```

</div>

<div class="form-group row">
  <label for="nama_mahasiswa" class="col-sm-2 col-form-
label">Nama</label>
  <div class="col-sm-6">
    <input      type="text"      class="form-control"
id="nama_mahasiswa"      name="nama_mahasiswa"
placeholder="Nama      Mahasiswa"      value="<?=
old('nama_mahasiswa'); ?>">
  </div>
</div>

<div class="form-group row">
  <label  for="alamat"  class="col-sm-2  col-form-
label">Alamat</label>
  <div class="col-sm-6">
    <input type="text" class="form-control" id="alamat"
name="alamat"  placeholder="Alamat  Mahasiswa"  value="<?=
old('alamat'); ?>">
  </div>
</div>

<div class="form-group row">
  <label  for="id_buku"  class="col-sm-2  col-form-
label">Kode Buku</label>
  <div class="col-sm-6">
    <input      type="text"      class="form-control"
id="id_buku"  name="id_buku"  placeholder="Kode  Buku"
value="<?= old('id_buku'); ?>">
  </div>
</div>

<div class="form-group row">

```

```

        <label    for="judul"    class="col-sm-2    col-form-
label">Judul Buku</label>

        <div class="col-sm-6">

            <input type="text" class="form-control" id="judul"
name="judul" placeholder="Judul Buku" value="<?= old('judul');
?>">

            </div>

        </div>

        <div class="form-group row">

            <label    for="pengarang"    class="col-sm-2    col-form-
label">Pengarang</label>

            <div class="col-sm-6">

                <input        type="text"        class="form-control"
id="pengarang"    name="pengarang"    placeholder="Pengarang
Buku" value="<?= old('pengarang'); ?>">

                </div>

            </div>

            <div class="form-group row">

                <label    for="penerbit"    class="col-sm-2    col-form-
label">Penerbit</label>

                <div class="col-sm-6">

                    <input        type="text"        class="form-control"
id="penerbit"    name="penerbit"    placeholder="Penerbit  Buku"
value="<?= old('penerbit'); ?>">

                    </div>

                </div>

                <div class="form-group row">

                    <label    for="jatuh_tempo"    class="col-sm-2    col-form-
label">Jatuh Tempo</label>

                    <div class="col-sm-6">

                        <input        type="text"        class="form-control"
id="jatuh_tempo"    name="jatuh_tempo"    placeholder="Jatuh
Tempo" value="<?= old('jatuh_tempo'); ?>">

```

```

        </div>
    </div>
</div>
<!-- /.card-body -->
<div class="card-footer">

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

    </div>
    <!-- /.card-footer -->
</form>
</div>
<!-- /.card -->
</div>
<?= $this->endSection('content'); ?>

```

- **Membuat Fitur View Detail**

Mahasiswa_view.php

```

<?= $this->extend('layout/dashboard-layout'); ?>
<?= $this->section('content'); ?>
<div class="col-md-12">
    <!-- general form elements -->
    <section class="content">
        <div class="container-fluid">
            <div class="row">
                <!-- /.card-header -->
                <!-- form start -->
                <div class="col-md-3">
                    <!-- Profile Image -->
                    <div class="card card-primary card-outline">
                        <div class="card-body box-profile">
                            <div class="text-center">

```

```

        </div>

        <h3 class="profile-username text-center"><?=
$buku->judul; ?></h3>

        <p class="text-muted text-center"><?= $buku-
>id_buku; ?><br><?= $buku->pengarang; ?></p>

        </div>

        <!-- /.card-body -->

    </div>

    <!-- /.card -->

</div>

<div class="col-md-9">

    <div class="card card-primary">

        <div class="card-header">

            <h3 class="card-title">Book Description</h3>

        </div>

        <!-- /.card-header -->

        <div class="card-body">

            <strong><i class="fas fa-book mr-1"></i>
                Education</strong>

            <p class="text-muted">
                Pengarang <?= $buku->pengarang; ?> Pada
                Sekolah Tinggi Manajemen dan Informatika
                (STMIK) Palangkaraya
            </p>

            <hr>

            <strong><i class="fas fa-mobile-alt mr-1"></i>
                Tahun

                Buku</strong>

            <p class="text-muted"><?= $buku->tahun; ?></p>

        </div>
    </div>

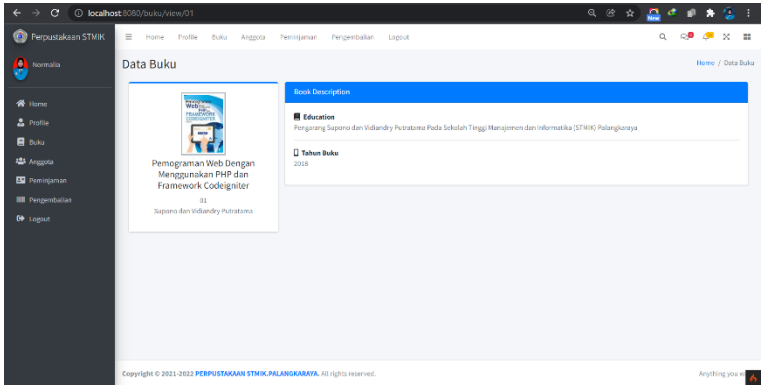
```



```

        <!-- /.card-body -->
    </div>
    <!-- /.card -->
</div>
</div>
    <!-- /.card -->
</div>
</div>
<?= $this->endSection('content'); ?>

```



B. Fasilitas Peminjaman dan Pengembalian Buku.

- Pertama kita akan membuat table daftar peminjaman. dengan cara ketik command berikut pada terminal VScode kita:

```
php spark migrate:create peminjaman
```

Setelah melakukan pembuatan table, maka akan muncul file baru di folder app/Database/Migrations.

Lalu akan terbuat sebuah file didalam folder app/Database/Migrations, nama file tersebut akan diakhiri dengan nama Peminjaman, buka file tersebut, dan isi dengan codenya seperti berikut ini:

```
<?php
namespace App\Database\Migrations;
use CodeIgniter\Database\Migration;
class Mahasiswa extends Migration
{
    public function up()
    {
        //
    }
    public function down()
    {
        //
    }
}
```

Selanjutnya, kita akan modifikasi kode program di atas untuk function menjadi :

```
<?php
namespace App\Database\Migrations;
use CodeIgniter\Database\Migration;
class Peminjaman extends Migration
```

```
{  
  
    public function up()  
  
    {  
  
        //  
  
        $this->forge->addField([  
  
            'kode_pinjam' => [  
  
                'type' => 'VARCHAR',  
  
                'constraint' => 11,  
  
            ],  
  
            'tanggal_peminjaman' => [  
  
                'type' => 'VARCHAR',  
  
                'constraint' => '255',  
  
            ],  
  
            'total_pinjam' => [  
  
                'type' => 'VARCHAR',  
  
                'constraint' => '255',  
  
            ],  
  
            'nim_mahasiswa' => [  
  
                'type' => 'VARCHAR',  
  
                'constraint' => '255',  
  
            ],  
  
            'nama_mahasiswa' => [  
  
                'type' => 'VARCHAR',  
  
                'constraint' => '255',  
  
            ],  
  
        ],  
  
    }  
  
}
```

```
'alamat' => [  
  
  'type' => 'VARCHAR',  
  
  'constraint' => '255',  
  
],  
  
'id_buku' => [  
  
  'type' => 'VARCHAR',  
  
  'constraint' => '255',  
  
],  
  
'judul' => [  
  
  'type' => 'VARCHAR',  
  
  'constraint' => '255',  
  
],  
  
'pengarang' => [  
  
  'type' => 'VARCHAR',  
  
  'constraint' => '255',  
  
],  
  
'penerbit' => [  
  
  'type' => 'VARCHAR',  
  
  'constraint' => '255',  
  
],  
  
'jatuh_tempo' => [  
  
  'type' => 'VARCHAR',  
  
  'constraint' => '255',  
  
],  
  
'qty' => [  
  
  'type' => 'INT',
```

```

        'constraint' => '11',

    ],

    'keluar' => [

        'type' => 'INT',

        'constraint' => '11',

    ],

    'created_at' => [

        'type' => 'DATETIME',

        'null' => true,

    ],

    'updated_at' => [

        'type' => 'DATETIME',

        'null' => true,

    ]

    ]);

    $this->forge->addPrimaryKey('kode_pinjam');

    $this->forge->createTable('peminjaman');

}

public function down()

{

    //

    $this->forge->dropTable('peminjaman');

}

}

```

Perintah diatas digunakan untuk membuat sebuah tabel dengan nama mahasiswa, yang berisi kolom antara lain :

Kode pinjam (VARCHAR 11)

Tanggal peminjaman (VARCHAR 255)

Total pinjam (VARCHAR 255)

Nim mahasiswa (VARCHAR 255)

Nama mahasiswa (VARCHAR 255)

Alamat (VARCHAR 255)

Id buku (VARCHAR 255)

Judul (VARCHAR 255)

Pengarang (VARCHAR 255)

Penerbit (VARCHAR 255)

Jatuh tempo (VARCHAR 255)

Qty (INT 11)

created_at (Date Time)

updated_at (Date Time)

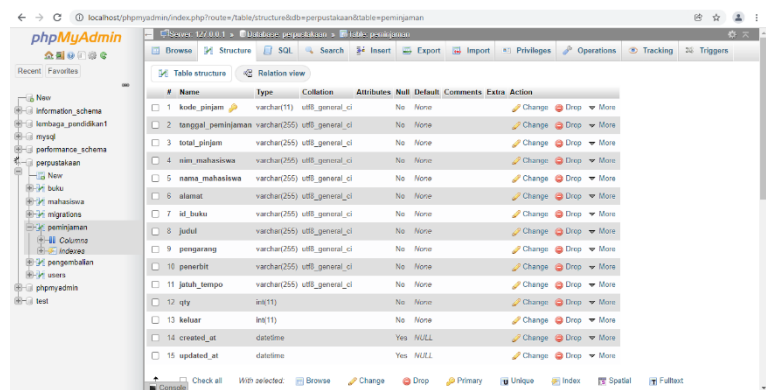
Didalam Codeigniter 4, kita dapat membuat table secara manual atau melalui migration. Selanjutnya kita akan menjalankan file migration untuk membuat tabel didatabase, silahkan masuk ke project codeigniter anda melalui terminal, lalu jalankan perintah : `php spark migrate` Seperti pada tampilan dibawah ini :

Php spark migrate

Seperti pada tampilan dibawah ini :

```
P5 C:\xampp\htdocs\perpustakaan> php spark migrate
```

Setalah melakukan perintah diatas dan berhasil maka, Silahkan kita cek database pada phpmyadmin untuk melihat struktur table yang telah dibuat melalui fitur migration yang ada pada codeigniter 4.



Sampai tahap ini kita sudah berhasil untuk membuat database ci4app dan table peminjaman untuk menyimpan data - data mahasiswa. Untuk mengisi data peminjamannya bisa secara manual melalui phpmyadmin atau kita dapat membuat Insert Dummy Data menggunakan Seeding yang telah disediakan oleh Codeigniter 4. Kita coba akan menambahkan dummy data didalam tabel mahasiswa menggunakan fitur seeding, sehingga nanti saat kita belajar untuk menampilkan data mahasiswa ke dalam bentuk tabel kita sudah memiliki beberapa data didalam tabel tersebut, langkah – langkahnya adalah sebagai berikut : Silahkan masuk kedalam project codeigniter 4 anda, melalui terminal, lalu jalankan perintah dibawah ini :

```
php spark make:seeder PeminjamanSeeder
```

lalu tekan enter, jika berhasil maka akan terbuat sebuah file dengan nama PeminjamanSeeder.php didalam folder app\Database\Seeds\ buka file tersebut, lalu tuliskan code didalam file PeminjamanSeeder.php

```
<?php

namespace App\Database\Seeds;

use CodeIgniter\Database\Seeder;

class PeminjamanSeeder extends Seeder
{
    public function run()
    {
```

```
//
$data = [
    [
        'kode_pinjam' => 'PJ003',
        'tanggal_peminjaman' => '28/12/2021',
        'total_pinjam' => '4',
        'nim_mahasiswa' => 'C2055201008',
        'nama_mahasiswa' => 'Normalia',
        'alamat' => 'Jalan Kalibata 2A',
        'id_buku' => '01',
        'judul' => 'Pemograman Web Dengan Menggunakan PHP dan Framework
Codeigniter',
        'pengarang' => 'Supono dan Vidiandry Putratama',
        'penerbit' => 'Deepublish',
        'jatuh_tempo' => '03/01/2022',
        'created_at' => Time::now()
    ],
    [
        'kode_pinjam' => 'PJ002',
        'tanggal_peminjaman' => '29/12/2021',
        'total_pinjam' => '3',
        'nim_mahasiswa' => 'C2055201013',
        'nama_mahasiswa' => 'Crisnawati',
        'alamat' => 'Jalan Kalbiata 1',
        'id_buku' => '03',
        'judul' => 'Komputer Forensik',
        'pengarang' => 'Feri Sulianta',
        'penerbit' => 'Elex Media Komputindo',
    ]
];
```



```
'jatuh_tempo' => '02/01/2022',

'created_at' => Time::now()

],

[

'kode_pinjam' => 'PJ001',

'tanggal_peminjaman' => '31/12/2021',

'total_pinjam' => '4',

'nim_mahasiswa' => 'C2055201014',

'nama_mahasiswa' => 'Rahmaniah',

'alamat' => 'Jalan Kecipir',

'id_buku' => '02',

'judul' => 'Jaringan Komputer',

'pengarang' => 'Edy Victor Haryanto',

'penerbit' => 'Andi',

'jatuh_tempo' => '03/01/2022',

'created_at' => Time::now()

]

];

$this->db->table('peminjaman')->insertBatch($data);

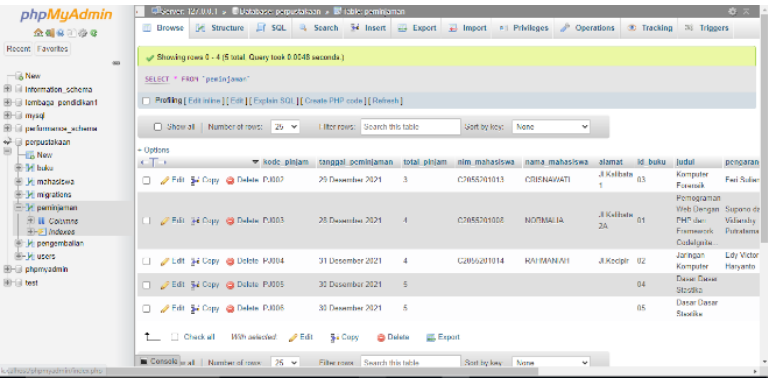
}

}
```

Keterangan : dalam perintah diatas, kita menginsert 3 data mahasiswa yang telah meminjam buku diperpustakaan dengan nama Normalia, Crisnawati, dan Rahmaniah. berikutnya kita akan jalankan perintah diatas untuk insert data mahasiswa tersebut ke dalam table peminjaman, silahkan masuk ke folder project codeigniter melalui terminal, lalu jalankan perintah sebagai berikut ini:

Php spark db:seed PeminjamanSedeer

yang perlu kita perhatikan adalah dibagian nama file yaitu PeminjamanSeeder silahkan sesuaikan untuk bagian ini sesuaikan dengan nama file Seeder, lalu tekan enter untuk menjalankan perintah tersebut. Lalu silahkan cek untuk tabel mahasiswa, harusnya sudah berisi 3 data mahasiswa yang kita insert melalui fitur seeder ini.



- **Membuat model peminjaman**

Sebelum kita memulai proses membuat CRUD, kita persiapkan dulu model yang akan kita gunakan untuk berkomunikasi dengan tabel peminjaman di database, untuk langkah awal buat file model dengan nama PeminjamanModel.php, simpan file tersebut didalam folder app/Models lalu isi file PeminjamanModel.php dengan code sebagai berikut :

```
<?php

namespace App\Models;

use CodeIgniter\Model;

class PeminjamanModel extends Model
{
    protected $table = "peminjaman";

    protected $primaryKey = "kode_pinjam";

    protected $returnType = "object";

    protected $useTimestamps = true;

    protected $allowedFields = [
```

```
        'kode_pinjam',    'tanggal_peminjaman',    'total_pinjam',  
'nim_mahasiswa','nama_mahasiswa','alamat',  
  
        'id_buku','judul','pengarang','penerbit','jatuh_tempo'  
  
    ];  
  
}
```

Keterangan :

untuk classnya kita beri nama yang sama dengan nama filenya yaitu MahasiswaModel

C. Membuat controller peminjaman

Berikutnya kita akan membuat menu untuk menampilkan data peminjaman, langkah – langkahnya adalah sebagai berikut :

Buatlah sebuah controller dengan nama file Peminjaman.php didalam folder app/Controllers, dan isi dengan code seperti berikut ini :

```
<?php  
  
  
namespace App\Controllers;  
  
use App\Models\PeminjamanModel;  
  
class Peminjaman extends BaseController  
{  
  
    protected $peminjaman;  
  
    function __construct()  
    {  
  
        $this->peminjaman = new PeminjamanModel();  
  
    }  
  
    public function index()  
    {  
  
        $data['pageTitle'] = 'Daftar Peminjaman';  
  
        $data['peminjaman'] = $this->peminjaman->findAll();  
  
        return view('dashboard/peminjaman', $data);  
  
    }  
}
```

```
}  
  
}
```

Kemudian edit kembali file routes.php kita yang ada pada folder apps/config/routes.php dengan menambahkan :

```
$routes->get('peminjaman', 'Peminjaman::index', ['as' => 'user.peminjaman']);
```

• **Membuat halaman view**

Membuat Halaman View Peminjaman pada folder app/views/dashboard/ Pada pertemuan sebelumnya kita telah membuat layout dashboard-layout.php pada pertemuan integrasi codeigniter 4 dan admin LTE, layout yang kita buat sekarang merupakan section content untuk halaman dinamis daftar peminjaman. Berikut isi dari file view peminjaman yang kita buat :

```
<?= $this->extend('layout/dashboard-layout'); ?>  
  
<?= $this->section('content'); ?>  
  
<!-- /.card-header -->  
  
<div class="card-body">  
  
    <?php if (!empty(session()->getFlashdata('message'))): ?>  
  
        <div class="alert alert-success alert-dismissible fade show" role="alert">  
  
            <?php echo session()->getFlashdata('message'); ?>  
  
            <button type="button" class="close" data-dismiss="alert" aria-label="Close">  
  
                <span aria-hidden="true">&times;</span>  
  
            </button>  
  
        </div>  
  
    <?php endif; ?>  
  
    <a href="<?= base_url('/peminjaman/create'); ?>" class="btn btn-primary">Tambah</a>  
  
    <hr />  
  
    <table class="table table-bordered"></a>
```

```
<thead>

<tr>

<th>No</th>

<th>Kode Pinjam</th>

<th>Tanggal Peminjaman</th>

<th>Total Pinjam</th>

<th>Kode Buku</th>

<th>Judul Buku</th>

<th>Jatuh Tempo</th>

<th>Action</th>

</tr>

</thead>

<tbody>

<?php

$no = 1;

foreach ($peminjaman as $row) {

?>

<tr>

<td><?= $no++; ?></td>

<td><?= $row->kode_pinjam; ?></td>

<td><?= $row->tanggal_peminjaman; ?></td>

<td><?= $row->total_pinjam; ?></td>

<td><?= $row->id_buku; ?></td>

<td><?= $row->judul; ?></td>

<td><?= $row->jatuh_tempo; ?></td>
```

```
<td>

        <a title="View" href="<?= base_url("peminjaman/view/$row->kode_pinjam"); ?>" class="btn btn-info"><i class="nav-icon fas fa-tasks"></i></a>

        <a title="Return" href="<?= base_url("peminjaman/edit/$row->kode_pinjam"); ?>" class="btn btn-warning"><i class="nav-icon fas fa-sign-in-alt"></i></a>

        <a title="Delete" href="<?= base_url("peminjaman/delete/$row->kode_pinjam") ?>" class="btn btn-danger" onclick="return confirm('Apakah Anda yakin ingin menghapus data ?')"><i class="nav-icon fas fa-trash"></i></a>

    </td>

</tr>

<?php
}

?>

</tbody>

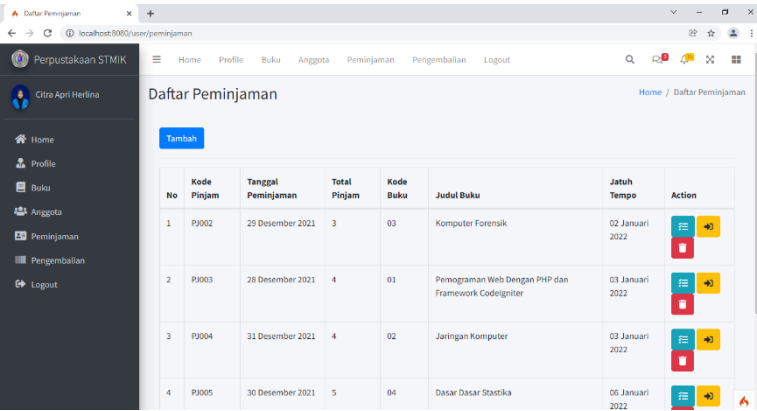
</table>

</div>

<!-- /.card-body -->

<?= $this->endSection('content'); ?>
```

Kemudian selanjutnya kita lihat hasilnya pada browser dengan mengetikkan `http://localhost:8080/peminjaman` jangan lupa untuk mengaktifkan server development local kita dan database server kita di xampp



D. Membuat fitur create

Setelah dalam bagian sebelumnya kita telah menampilkan data mahasiswa, berikutnya kita akan membuat fitur untuk menambahkan data mahasiswa, langkah – langkahnya adalah sebagai berikut :

pada step awal kita bisa menambahkan routing untuk kebutuhan menampilkan form tambah data baru, silahkan buka file dengan nama Routes.php didalam folder app/Config tambahkan code routing seperti berikut :

```
$routes->get('/peminjaman/create', 'Peminjaman::create');

$routes->get('/peminjaman/store', 'Peminjaman::store');
```

Kemudian buatlah form input data pada folder app/views/dassboard/peminjaman_ create.php untuk isi kode programnya dapat dilihat seperti dibawah ini :

Peminjaman_create.php

```
<?= $this->extend('layout/dashboard-layout'); ?>

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

<div class="col-md-10">

    <!-- general form elements -->

    <div class="card card-primary">

        <div class="card-header">

            <h3 class="card-title">Tambah Data Peminjaman</h3>

        </div>

        <!-- /.card-header -->

        <!-- form start -->

        <?php if (!empty(session()->getFlashdata('error'))): ?>

            <div class="alert alert-light alert-dismissible fade show" role="alert">

                <h4 class="alert-heading">Periksa Entrian Form</h4>

                </hr />

                <?php echo session()->getFlashdata('error'); ?>

            </div>

        </?php if (!empty(session()->getFlashdata('error'))): ?>
```

```

        <button type="button" class="close" data-dismiss="alert" aria-
label="Close">

        <span aria-hidden="true">&times;</span>

    </button>

</div>

<?php endif; ?>

<form class="" method="post" action="<?= base_url('peminjaman/store') ?>"
enctype="multipart/form-data">

    <?= csrf_field(); ?>

    <div class="card-body">

        <div class="form-group row">

            <label for="kode_pinjam" class="col-sm-2 col-form-label">Kode
Pinjam</label>

            <div class="col-sm-6">

                <input type="text" class="form-control" id="kode_pinjam"
name="kode_pinjam" placeholder="Kode Pinjam" value="<?= old('kode_pinjam');
?>">

            </div>

        </div>

        <div class="form-group row">

            <label for="tanggal_peminjaman" class="col-sm-2 col-form-
label">Tanggal Peminjaman</label>

            <div class="col-sm-6">

                <input type="text" class="form-control" id="tanggal_peminjaman"
name="tanggal_peminjaman" placeholder="Tanggal Peminjaman" value="<?=
old('tanggal_peminjaman'); ?>">

            </div>

        </div>

    </div>

```



```
<div class="form-group row">

    <label for="total_pinjam" class="col-sm-2 col-form-label">Total
Pinjam</label>

    <div class="col-sm-6">

        <input type="text" class="form-control" id="total_pinjam"
name="total_pinjam" placeholder="Total Pinjam" value="<?= old('total_pinjam');
?>">

    </div>

</div>

<div class="form-group row">

    <label for="nim_mahasiswa" class="col-sm-2 col-form-
label">Nim</label>

    <div class="col-sm-6">

        <input type="text" class="form-control" id="nim_mahasiswa"
name="nim_mahasiswa" placeholder="Nim Mahasiswa" value="<?=
old('nim_mahasiswa'); ?>">

    </div>

</div>

<div class="form-group row">

    <label for="nama_mahasiswa" class="col-sm-2 col-form-
label">Nama</label>

    <div class="col-sm-6">

        <input type="text" class="form-control" id="nama_mahasiswa"
name="nama_mahasiswa" placeholder="Nama Mahasiswa" value="<?=
old('nama_mahasiswa'); ?>">

    </div>

</div>
```

```
<div class="form-group row">

    <label for="alamat" class="col-sm-2 col-form-label">Alamat</label>

    <div class="col-sm-6">

        <input type="text" class="form-control" id="alamat"
name="alamat" placeholder="Alamat Mahasiswa" value="<?= old('alamat'); ?>">

    </div>

</div>

<div class="form-group row">

    <label for="id_buku" class="col-sm-2 col-form-label">Kode
Buku</label>

    <div class="col-sm-6">

        <input type="text" class="form-control" id="id_buku"
name="id_buku" placeholder="Kode Buku" value="<?= old('id_buku'); ?>">

    </div>

</div>

<div class="form-group row">

    <label for="judul" class="col-sm-2 col-form-label">Judul
Buku</label>

    <div class="col-sm-6">

        <input type="text" class="form-control" id="judul" name="judul"
placeholder="Judul Buku" value="<?= old('judul'); ?>">

    </div>

</div>

<div class="form-group row">

    <label for="pengarang" class="col-sm-2 col-form-
label">Pengarang</label>
```

```
<div class="col-sm-6">

    <input type="text" class="form-control" id="pengarang"
name="pengarang" placeholder="Pengarang Buku" value="<?= old('pengarang');
?>">

</div>

</div>

<div class="form-group row">

<label for="penerbit" class="col-sm-2 col-form-label">Penerbit</label>

    <div class="col-sm-6">

        <input type="text" class="form-control" id="penerbit"
name="penerbit" placeholder="Penerbit Buku" value="<?= old('penerbit'); ?>">

        </div>

    </div>

    <div class="form-group row">

        <label for="jatuh_tempo" class="col-sm-2 col-form-label">Jatuh
Tempo</label>

        <div class="col-sm-6">

            <input type="text" class="form-control" id="jatuh_tempo"
name="jatuh_tempo" placeholder="Jatuh Tempo" value="<?= old('jatuh_tempo');
?>">

            </div>

        </div>

    </div>

<!-- /.card-body -->

<div class="card-footer">

    <button type="submit" class="btn btn-primary">Simpan</button>
```

```

        </div>

        <!-- /.card-footer -->

    </form>

</div>

<!-- /.card -->

</div>

<?= $this->endSection('content'); ?>

```

Selanjutnya kita modifikasi controller Peminjaman.php yang ada pada folder app/controllers/Peminjaman.php

Tambahkan method baru create() dan store() di bawah ini pada file controller Mahasiswa.php yang sebelumnya sudah pernah kita buat

```

public function create()
{

    $data = [

        'pageTitle' => 'Input Data Peminjaman',

        'validation' => \Config\Services::validation()

    ];

    return view('dashboard/peminjaman_create', $data);
}

public function store()
{
    if (!$this->validate([

        'kode_pinjam' => [

            'rules' => 'required|is_unique[peminjaman.kode_pinjam]',

            'errors' => [

```

```
'required' => '{field} Harus diisi',

'is_unique' => 'Kode_pinjam sudah terdaftar'

]

],

'tanggal_peminjaman' => [

    'rules' => 'required',

    'errors' => [

        'required' => '{field} Harus diisi'

    ]

],

'total_pinjam' => [

    'rules' => 'required',

    'errors' => [

        'required' => '{field} Harus diisi'

    ]

],

'nim_mahasiswa' => [

    'rules' => 'required',

    'errors' => [

        'required' => '{field} Harus diisi'

    ]

],

'nama_mahasiswa' => [

    'rules' => 'required',

    'errors' => [

        'required' => '{field} Harus diisi'

    ]

]
```

```

],

'alamat' => [

    'rules' => 'required',

    'errors' => [

        'required' => '{field} Harus diisi'

    ]

],

'id_buku' => [

    'rules' => 'required',

    'errors' => [

        'required' => '{field} Harus diisi'

    ]

],

'judul' => [

    'rules' => 'required',

    'errors' => [

        'required' => '{field} Harus diisi'

    ]

],

'pengarang' => [

    'rules' => 'required',

    'errors' => [

        'required' => '{field} Harus diisi'

    ]

],

'penerbit' => [

    'rules' => 'required',

```

```

        'errors' => [

            'required' => '{field} Harus diisi'

        ]

    ],

    'jatuh_tempo' => [

        'rules' => 'required',

        'errors' => [

            'required' => '{field} Harus diisi'

        ]

    ],

    )) {

        session()->setFlashdata('error', $this->validator->listErrors());

        return redirect()->back()->withInput();

    }

    $this->peminjaman->insert([

        'kode_pinjam' => $this->request->getVar('kode_pinjam'),

        'tanggal_peminjaman' => $this->request->getVar('tanggal_peminjaman'),

        'total_pinjam' => $this->request->getVar('total_pinjam'),

        'id_buku' => $this->request->getVar('id_buku'),

        'judul' => $this->request->getVar('judul'),

        'jatuh_tempo' => $this->request->getVar('jatuh_tempo')

    ]);

    session()->setFlashdata('message', 'Tambah Data Peminjaman Berhasil');

    return redirect()->to('/peminjaman');

}

public function view($kode_pinjam)

```

```

{

$dataPeminjaman = $this->peminjaman->find($kode_pinjam);

if (empty($dataPeminjaman)) {

throw new

\CodeIgniter\Exceptions\PageNotFoundException('Data Peminjaman

Tidak ditemukan !');

}

$data['pageTitle'] = 'Data Peminjaman';

$data['peminjaman'] = $dataPeminjaman;

return view('dashboard/peminjaman_view', $data);

}

function edit($kode_pinjam)

{

$dataPeminjaman = $this->peminjaman->find($kode_pinjam);

if (empty($dataPeminjaman)) {

throw new

\CodeIgniter\Exceptions\PageNotFoundException('Data Peminjaman

Tidak ditemukan !');

}

$data['pageTitle'] = 'Tambah Data Pengembalian';

$data['peminjaman'] = $dataPeminjaman;

return view('dashboard/peminjaman_edit', $data);

}

public function update($kode_pinjam)

{

if (!$this->validate([

'kode_pinjam' => [

```



```
'rules' => 'required',

'errors' => [

'required' => '{field} Harus diisi'

]

],

'tanggal_peminjaman' => [

'rules' => 'required',

'errors' => [

'required' => '{field} Harus diisi'

]

],

'total_pinjam' => [

'rules' => 'required',

'errors' => [

'required' => '{field} Harus diisi'

]

],

'id_buku' => [

'rules' => 'required',

'errors' => [

'required' => '{field} Harus diisi'

]

],

'judul' => [

'rules' => 'required',

'errors' => [

'required' => '{field} Harus diisi'
```

```

    ]
    ],
    'jatuh_tempo' => [
        'rules' => 'required',
        'errors' => [
            'required' => '{field} Harus diisi'
        ]
    ],
    ))) {
        session()->setFlashdata('error', $this->validator->listErrors());

        return redirect()->back();
    }

    $this->peminjaman->update($kode_pinjam, [
        'kode_pinjam' => $this->request->getVar('kode_pinjam'),
        'tanggal_peminjaman' => $this->request->getVar('tanggal_peminjaman'),
        'total_pinjam' => $this->request->getVar('total_pinjam'),
        'id_buku' => $this->request->getVar('id_buku'),
        'judul' => $this->request->getVar('judul'),
        'jatuh_tempo' => $this->request->getVar('jatuh_tempo'),

    ]);

    session()->setFlashdata('message', 'Update Data Peminjaman
    Berhasil');

    return redirect()->to('/peminjaman');
}

function delete($kode_pinjam)
{

```

```
$dataPeminjaman = $this->peminjaman->find($kode_pinjam);

if (empty($dataPeminjaman)) {

    throw new

    \CodeIgniter\Exceptions\PageNotFoundException('Data Peminjaman

    Tidak ditemukan !');

}

$this->peminjaman->delete($kode_pinjam);

session()->setFlashdata('message', 'Delete Data peminjaman

    Berhasil');

return redirect()->to('/peminjaman');

}

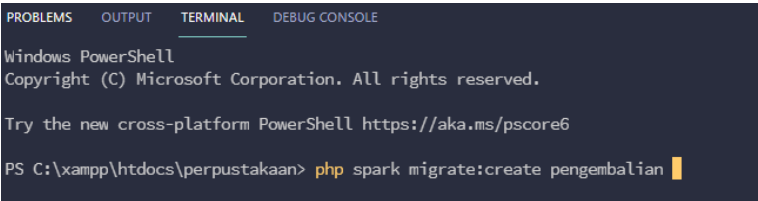
}
```

selanjutnya kita lihat hasilnya pada browser dengan mengetikkan `http://localhost:8080/peminjaman/create` jangan lupa untuk mengaktifkan server development local kita dan database server kita di xampp agar dapat diakses melalui browser.

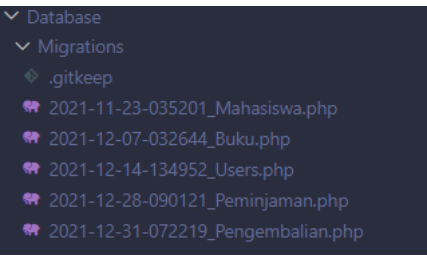
Lakukan dengan hal yang seperti crud buku,crud peminjaman,disini kami akan langsung masuk ke crud pengembalian .

Kemudian buat table pengembalian dengan cara ketik command berikut pada terminal VScode kita:

```
php spark migrate:create pengembalian
```



Setelah melakukan pembuatan table, maka akan muncul file baru di folder `app/Database/Migrations`.



Lalu akan terbuat sebuah file didalam folder app/Database/Migrations, nama file tersebut akan diakhiri dengan nama Pengembalian, buka file tersebut, dan isi dengan codenya seperti berikut ini:

```
<?php

namespace App\Database\Migrations;

use CodeIgniter\Database\Migration;

class Pengembalian extends Migration
{
    public function up()
    {
        //

        $this->forge->addField([

            'kode_pinjam' => [

                'type' => 'VARCHAR',

                'constraint' => 11,

            ],

            'nim' => [

                'type' => 'VARCHAR',

                'constraint' => '255',

            ],

            'nama' => [

                'type' => 'VARCHAR',
```

```
'constraint' => '255',

],

'id_buku' => [

'type' => 'VARCHAR',

'constraint' => '255',

],

'judul' => [

'type' => 'VARCHAR',

'constraint' => '255',

],

'tanggal_peminjaman' => [

'type' => 'VARCHAR',

'constraint' => '255',

],

'jatuh_tempo' => [

'type' => 'VARCHAR',

'constraint' => '255',

],

'tanggal_kembali' => [

'type' => 'VARCHAR',

'constraint' => '255',

],

'telat' => [

'type' => 'VARCHAR',

'constraint' => '255',

],

'denda' => [
```

```
'type' => 'VARCHAR',

'constraint' => '255',

],

'bayar' => [

'type' => 'VARCHAR',

'constraint' => '255',

],

'created_at' => [

'type' => 'DATETIME',

'null' => true,

],

'updated_at' => [

'type' => 'DATETIME',

'null' => true,

]

]);

$this->forge->addPrimaryKey('kode_pinjam');

$this->forge->createTable('pengembalian');

}

public function down()

{

    //

    $this->forge->dropTable('pengembalian');

}

}
```

Didalam Codeigniter 4, kita dapat membuat table secara manual atau melalui migration. Selanjutnya kita akan menjalankan file migration untuk membuat

tabel didatabase, silahkan masuk ke project codeigniter anda melalui terminal, lalu jalankan perintah : `php spark migrate` Seperti pada tampilan dibawah ini :

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\xampp\htdocs\perpustakaan> php spark migrate
```

Silahkan di cek database pada phpmyadmin untuk melihat struktur table yang telah dibuat melalui fitur migration yang ada pada codeigniter 4

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 kode_pinjam	varchar(11)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	2 nim	varchar(255)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	3 nama	varchar(255)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	4 id_buku	varchar(255)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	5 judul	varchar(255)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	6 tanggal_peminjaman	varchar(255)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	7 jatuh_tempo	varchar(255)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	8 tanggal_kembali	varchar(255)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	9 telat	varchar(255)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	10 denda	varchar(255)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	11 bayar	varchar(255)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	12 created_at	datetime			Yes	NULL			Change Drop More
<input type="checkbox"/>	13 updated_at	datetime			Yes	NULL			Change Drop More

Sampai tahap ini kita sudah berhasil untuk membuat database perpustakaan dan table pengembalian untuk menyimpan data data pengembalian. Untuk mengisi data pengembaliannya bisa secara manual melalui phpmyadmin atau kita dapat membuat Insert Dummy Data menggunakan Seeding yang telah disediakan oleh Codeigniter 4.Kita coba akan tambahkan dummy data didalam tabel pengembalian menggunakan fitur seeding, sehingga nanti saat kita belajar untuk menampilkan data pengembalian ke dalam bentuk tabel kita sudah

memiliki beberapa data didalam tabel tersebut, langkah – langkahnya adalah sebagai berikut :

Silahkan masuk kedalam project codeigniter 4 anda, melalui terminal, lalu jalankan perintah dibawah ini :

```
PS C:\xampp\htdocs\perpustakaan> php spark make:seeder PengembalianSeeder
```

lalu tekan enter, jika berhasil maka akan terbuat sebuah file dengan nama PengembalianSeeder.php didalam folder app\Database\Seeds\ buka file tersebut, lalu tuliskan code didalam file PengembalianSeeder.php

```
<?php
namespace App\Database\Seeds;
use CodeIgniter\Database\Seeder;

class PengembalianSeeder extends Seeder
{
    public function run()
    {
        //
        $data = [
            [
                'kode_pinjam' => 'PJ001',
                'nim' => 'C2055201014',
                'nama' => 'Rahmaniah',
                'id_buku' => '02',
                'judul' => 'Jaringan Komputer',
                'tanggal_peminjaman' => '31-12-2021',
                'jatuh_tempo' => '06-01-2022',
                'tanggal_kembali' => '08-01-2022',
```



```

'telat' => '2 hari',

'denda' => '2000',

'bayar' => '2000',

'created_at' => Time::now()

],

[

'kode_pinjam' => 'PJ002',

'nim' => 'C2055201013',

'nama' => 'Crisnawati',

'id_buku' => '03',

'judul' => 'Komputer Forensik',

'tanggal_peminjaman' => '31-12-2021',

'jatuh_tempo' => '06-01-2022',

'tanggal_kembali' => '08-01-2022',

'telat' => '2 hari',

'denda' => '2000',

'bayar' => '2000',

'created_at' => Time::now()

],

[

'kode_pinjam' => 'PJ003',

'nim' => 'C2055201008',

'nama' => 'Normalia',

'id_buku' => '01',

'judul' => 'Pemograman Web Dengan Menggunakan PHP dan Framework
Codeigniter',

'tanggal_peminjaman' => '31-12-2021',

'jatuh_tempo' => '06-01-2022',

```

```
'tanggal_kembali' => '08-01-2022',

'telat' => '2 hari',

'denda' => '2000',

'bayar' => '2000',

'created_at' => Time::now()

]

];

$this->db->table('pengembalian')->insertBatch($data);

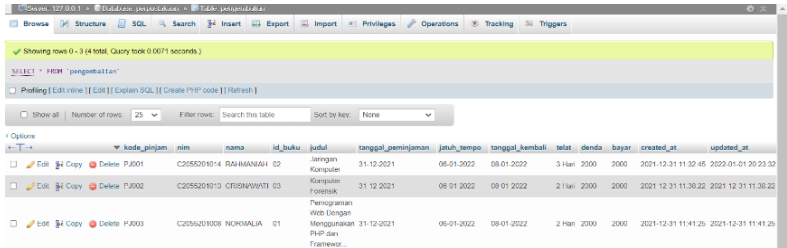
}

}
```

Keterangan : dalam perintah diatas, kita menginsert 3 data mahasiswa yang telah kembalikan buku diperpustakaan dengan nama Normalia, Crisnawati, dan Rahmaniah. berikutnya kita akan jalankan perintah diatas untuk insert data mahasiswa tersebut ke dalam table pengembalian, silahkan masuk ke folder project codeigniter melalui terminal, lalu jalankan perintah sebagai berikut ini:

Php spark db:seed PeminjamanSedeer

yang perlu kita perhatikan adalah dibagian nama file yaitu PengembalianSeeder silahkan sesuaikan untuk bagian ini sesuaikan dengan nama file Seeder, lalu tekan enter untuk menjalankan perintah tersebut. Lalu silahkan cek untuk tabel



code_peminj	nim	nama	id_buku	judul	tanggal_peminjaman	jatuh_tempo	tanggal_kembali	telat	denda	bayar	created_at	updated_at
PJ001	C2055201014	RAHMANIAH	02	lingkaran Kupu-kupu	31-12-2021	06-01-2022	08-01-2022	3 Hari	2000	2000	2021-12-31 11:32:45	2022-01-01 20:23:59
PJ002	C2055201013	CRISNAWATI	03	Kompas Forensik	31-12-2021	06-01-2022	08-01-2022	2 Hari	2000	2000	2021-12-31 11:30:22	2021-12-31 11:30:22
PJ003	C2056201008	NORMALIA	01	Peminjaman Web Loggan Menggunakan PHP dan Framework...	31-12-2021	06-01-2022	08-01-2022	2 Hari	2000	2000	2021-12-31 11:41:25	2021-12-31 11:41:25

melalui fitur seeder ini.

mahasiswa, harusnya sudah berisi 3 data mahasiswa yang kita insert

- **Membuat model pengembalian**

Sebelum kita memulai proses membuat CRUD, kita persiapkan dulu model yang akan kita gunakan untuk berkomunikasi dengan tabel pengembalian di database, untuk langkah awal buat file model dengan nama PengembalianModel.php, simpan file tersebut didalam folder app/Models lalu isi file PengembalianModel.php dengan code sebagai berikut :

```
<?php

namespace App\Models;

use CodeIgniter\Model;

class PengembalianModel extends Model
{
    protected $table = "pengembalian";

    protected $primaryKey = "kode_pinjam";

    protected $returnType = "object";

    protected $useTimestamps = true;

    protected $allowedFields = [

        'kode_pinjam', 'nim', 'nama', 'id_buku','judul','tanggal_peminjaman',

        'jatuh_tempo','tanggal_kembali','telat','denda','bayar'

    ];
}
```

Keterangan :

untuk classnya kita beri nama yang sama dengan nama filenya yaitu PengembalianModel.

C. Membuat controller pengembalian

Berikutnya kita akan membuat menu untuk menampilkan data peminjaman, langkah – langkahnya adalah sebagai berikut :

Buatlah sebuah controller dengan nama file Pengembalian.php didalam folder app/Controllers, dan isi dengan code seperti berikut ini :

```
<?php
namespace App\Controllers;
use App\Models\PengembalianModel;
class Pengembalian extends BaseController
{
    protected $pengembalian;

    function __construct()
    {
        $this->pengembalian = new PengembalianModel();
    }

    public function index()
    {
        $data['pageTitle'] = 'Daftar Pengembalian';

        $data['pengembalian'] = $this->pengembalian->findAll();

        return view('dashboard/pengembalian', $data);
    }
}
```

Kemudian edit kembali file routes.php kita yang ada pada folder apps/config/routes.php dengan menambahkan :

```
$routes->get('/pengembalian', 'Pengembalian::index');
```

- **Membuat halaman view**

Membuat Halaman View Pengembalian pada folder app/views/dashboard/ Pada pertemuan sebelumnya kita telah membuat layout dashboard-layout.php pada pertemuan integrasi codeigniter 4 dan admin LTE, layout yang kita buat sekarang merupakan section content untuk halaman dinamis daftar pengembalian. Berikut isi dari file view pengembalian yang kita buat :

```
<?= $this->extend('layout/dashboard-layout'); ?>

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

<!-- /.card-header -->

<div class="card-body">

    <?php if (!empty(session()->getFlashdata('message'))) : ?>

        <div class="alert alert-success alert-dismissible fade show" role="alert">

            <?php echo session()->getFlashdata('message'); ?>

            <button type="button" class="close" data-dismiss="alert" aria-label="Close">

                <span aria-hidden="true">&times;</span>

            </button>

        </div>

    <?php endif; ?>

    <a href="<?= base_url('/pengembalian/create'); ?>" class="btn btn-
primary">Tambah</a>

    <hr />

    <table class="table table-bordered">

        <thead>

            <tr>

                <th>No</th>

                <th>Kode Pinjam</th>

                <th>Kode Buku</th>
```

```
<th>Judul Buku</th>

<th>Tgl. Peminjaman</th>

<th>Jatuh Tempo</th>

<th>Tgl. Kembali</th>

<th>Telat</th>

<th>Action</th>

</tr>

</thead>

<tbody>

<?php

$no = 1;

foreach ($pengembalian as $row) {

?>

<tr>

<td><?= $no++; ?></td>

<td><?= $row->kode_pinjam; ?></td>

<td><?= $row->id_buku; ?></td>

<td><?= $row->judul; ?></td>

<td><?= $row->tanggal_peminjaman; ?></td>

<td><?= $row->jatuh_tempo; ?></td>

<td><?= $row->tanggal_kembali; ?></td>

<td><?= $row->telat; ?></td>

<td>
```

```
<a title="View" href="<?= base_url("pengembalian/view/$row-
>kode_pinjam"); ?>" class="btn btn-dark"><i class="nav-icon fas fa-tasks"></i></a>

<a title="Edit" href="<?= base_url("pengembalian/edit/$row-
>kode_pinjam"); ?>" class="btn btn-warning"><i class="nav-icon fas fa-
edit"></i></a>

<a title="Delete" href="<?= base_url("pengembalian/delete/$row-
>kode_pinjam") ?>" class="btn btn-danger" onclick="return confirm('Apakah Anda
yakin ingin menghapus data ?')"><i class="nav-icon fas fa-trash"></i></a>

</td>

</tr>

<?php
}
?>

</tbody>

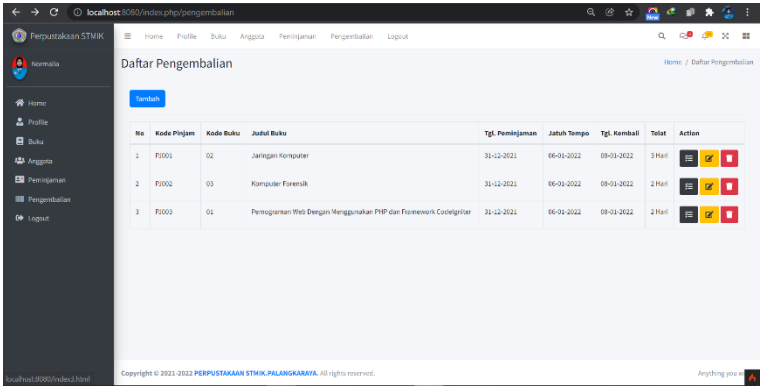
</table>

</div>

<!-- /.card-body -->

<?= $this->endSection('content'); ?>
```

Kemudian selanjutnya kita lihat hasilnya pada browser dengan mengetikkan <http://localhost:8080/pengembalian> jangan lupa untuk mengaktifkan server development local kita dan database server kita di xampp ☺.



D. Membuat fitur create

Setelah dalam bagian sebelumnya kita telah menampilkan data pengembalian, berikutnya kita akan membuat fitur untuk menambahkan data pengembalian, langkah – langkahnya adalah sebagai berikut :

pada step awal kita bisa menambahkan routing untuk kebutuhan menampilkan form tambah data baru, silahkan buka file dengan nama Routes.php didalam folder app/Config tambahkan code routing seperti berikut :

```
$routes->get('/pengembalian/create', 'Pengembalian::create');

$routes->get('/pengembalian/store', 'Pengembalian::store');
```

Kemudian buatlah form input data pada folder app/views/dassboard/pengembalian_create.php untuk isi kode programnya dapat dilihat seperti dibawah ini :

Pengembalian_create.php

```
<?= $this->extend('layout/dashboard-layout'); ?>

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

<div class="col-md-10">

    <!-- general form elements -->

    <div class="card card-primary">

        <div class="card-header">

            <h3 class="card-title">Tambah Data Pengembalian</h3>

        </div>

        <!-- /.card-header -->
```



```
<!-- form start -->

<?php if (!empty(session()->getFlashdata('error'))) : ?>

    <div class="alert alert-light alert-dismissible fade show" role="alert">

        <h4 class="alert-heading">Periksa Entrian Form</h4>

        <hr />

        <?php echo session()->getFlashdata('error'); ?>

        <button type="button" class="close" data-dismiss="alert" aria-
label="Close">

            <span aria-hidden="true">&times;</span>

        </button>

    </div>

<?php endif; ?>

<form class="" method="post" action="<?= base_url('pengembalian/store') ?>"
enctype="multipart/form-data">

    <?= csrf_field(); ?>

    <div class="card-body">

        <div class="form-group row">

            <label for="kode_pinjam" class="col-sm-2 col-form-label">Kode
Pinjam</label>

            <div class="col-sm-6">

                <input type="text" class="form-control" id="kode_pinjam"
name="kode_pinjam" placeholder="Kode Pinjam" value="<?= old('kode_pinjam');
?>">

            </div>

        </div>

    </div>

    <div class="form-group row">

        <label for="nim" class="col-sm-2 col-form-label">Nim</label>

        <div class="col-sm-6">
```

```

        <input type="text" class="form-control" id="nim" name="nim"
placeholder="Nim Mahasiswa" value="<?= old('nim'); ?>">

        </div>

    </div>

    <div class="form-group row">

        <label for="nama" class="col-sm-2 col-form-label">Nama</label>

        <div class="col-sm-6">

            <input type="text" class="form-control" id="nama" name="nama"
placeholder="Nama Mahasiswa" value="<?= old('nama'); ?>">

            </div>

        </div>

    <div class="form-group row">

        <label for="id_buku" class="col-sm-2 col-form-label">Kode
Buku</label>

        <div class="col-sm-6">

            <input type="text" class="form-control" id="id_buku"
name="id_buku" placeholder="Kode Buku" value="<?= old('id_buku'); ?>">

            </div>

        </div>

    <div class="form-group row">

        <label for="judul" class="col-sm-2 col-form-label">Judul Buku</label>

        <div class="col-sm-6">

            <input type="text" class="form-control" id="judul" name="judul"
placeholder="Judul Buku" value="<?= old('judul'); ?>">

            </div>

        </div>
    </div>

```

```
<div class="form-group row">

    <label for="tanggal_peminjaman" class="col-sm-2 col-form-
label">Tanggal Peminjaman</label>

    <div class="col-sm-6">

        <input type="text" class="form-control" id="tanggal_peminjaman"
name="tanggal_peminjaman" placeholder="Tanggal Peminjaman" value="<?=
old('tanggal_peminjaman'); ?>">

    </div>

</div>

<div class="form-group row">

    <label for="jatuh_tempo" class="col-sm-2 col-form-label">Jatuh
Tempo</label>

    <div class="col-sm-6">

        <input type="text" class="form-control" id="jatuh_tempo"
name="jatuh_tempo" placeholder="Jatuh Tempo" value="<?= old('jatuh_tempo');
?>">

    </div>

</div>

<div class="form-group row">

    <label for="tanggal_kembali" class="col-sm-2 col-form-label">Tanggal
Kembali</label>

    <div class="col-sm-6">

        <input type="text" class="form-control" id="tanggal_kembali"
name="tanggal_kembali" placeholder="Tanggal Kembali" value="<?=
old('tanggal_kembali'); ?>">

    </div>

</div>
```

```
<div class="form-group row">

  <label for="telat" class="col-sm-2 col-form-label">Telat
Pengembalian</label>

  <div class="col-sm-6">

    <input type="text" class="form-control" id="telat" name="telat"
placeholder="Telat Pengembalian" value="<?= old('telat'); ?>">

  </div>

</div>

<div class="form-group row">

  <label for="denda" class="col-sm-2 col-form-label">Denda</label>

  <div class="col-sm-6">

    <input type="text" class="form-control" id="denda" name="denda"
placeholder="Denda" value="<?= old('denda'); ?>">

  </div>

</div>

<div class="form-group row">

  <label for="bayar" class="col-sm-2 col-form-label">Bayar Denda</label>

  <div class="col-sm-6">

    <input type="text" class="form-control" id="bayar" name="bayar"
placeholder="Bayar Denda" value="<?= old('bayar'); ?>">

  </div>

</div>

</div>

<!-- /.card-body -->

<div class="card-footer">
```

```

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

        <button type="submit" class="btn btn-warning">Batal</button>

    </div>

    <!-- /.card-footer -->

</form>

</div>

<!-- /.card -->

</div>

<?= $this->endSection('content'); ?>

```

Selanjutnya kita modifikasi controller Pengembalian.php yang ada pada folder app/controllers/Pengembalian.php

Tambahkan method baru create(),store(),view(),edit(),dan delete() di bawah ini pada file controller Pengembalian.php yang sebelumnya sudah pernah kita buat

```

public function create()
{

    $data = [

        'pageTitle' => 'Input Data Pengembalian',

        'validation' => \Config\Services::validation()

    ];

    return view('dashboard/pengembalian_create', $data);

}

public function store()
{

    if (!$this->validate([

```

```
'kode_pinjam' => [  
  'rules' => 'required|is_unique[pengembalian.kode_pinjam]',  
  'errors' => [  
    'required' => '{field} Harus diisi',  
    'is_unique' => 'Kode_pinjam sudah terdaftar'  
  ]  
,  
'nim' => [  
  'rules' => 'required',  
  'errors' => [  
    'required' => '{field} Harus diisi'  
  ]  
,  
'nama' => [  
  'rules' => 'required',  
  'errors' => [  
    'required' => '{field} Harus diisi'  
  ]  
,  
'nama' => [  
  'rules' => 'required',  
  'errors' => [  
    'required' => '{field} Harus diisi'  
  ]  
,  
'id_buku' => [  
  'rules' => 'required',
```

```

      'errors' => [
        'required' => '{field} Harus diisi'
      ]
    ],
    'judul' => [
      'rules' => 'required',
      'errors' => [
        'required' => '{field} Harus diisi'
      ]
    ],
    'tanggal_peminjaman' => [
      'rules' => 'required',
      'errors' => [
        'required' => '{field} Harus diisi'
      ]
    ],
    'jatuh_tempo' => [
      'rules' => 'required',
      'errors' => [
        'required' => '{field} Harus diisi'
      ]
    ],
    'tanggal_kembali' => [
      'rules' => 'required',
      'errors' => [
        'required' => '{field} Harus diisi'
      ]
    ]
  ]
}

```

```

    ],

    'telat' => [

        'rules' => 'required',

        'errors' => [

            'required' => '{field} Harus diisi'

        ]

    ],

    'denda' => [

        'rules' => 'required',

        'errors' => [

            'required' => '{field} Harus diisi'

        ]

    ],

    'bayar' => [

        'rules' => 'required',

        'errors' => [

            'required' => '{field} Harus diisi'

        ]

    ],

))) {

    session()->setFlashdata('error', $this->validator->listErrors());

    return redirect()->back()->withInput();

}

$this->pengembalian->insert([

    'kode_pinjam' => $this->request->getVar('kode_pinjam'),

    'id_buku' => $this->request->getVar('id_buku'),

```



```

        'judul' => $this->request->getVar('judul'),

        'tanggal_peminjaman' => $this->request->getVar('tanggal_peminjaman'),

        'jatuh_tempo' => $this->request->getVar('jatuh_tempo'),

        'tanggal_kembali' => $this->request->getVar('tanggal_kembali'),

        'telat' => $this->request->getVar('telat'),

        ]);

        session()->setFlashdata('message', 'Tambah Data Pengembalian Berhasil');

        return redirect()->to('/pengembalian');
    }

    public function view($kode_pinjam)
    {
        $dataPengembalian = $this->pengembalian->find($kode_pinjam);

        if (empty($dataPengembalian)) {

            throw new

            \CodeIgniter\Exceptions\PageNotFoundException('Data Pengembalian

            Tidak ditemukan !');

        }

        $data['pageTitle'] = 'Data Pengembalian';

        $data['pengembalian'] = $dataPengembalian;

        return view('dashboard/pengembalian_view', $data);

    }

    function edit($kode_pinjam)
    {

        $dataPengembalian = $this->pengembalian->find($kode_pinjam);

        if (empty($dataPengembalian)) {

            throw new

```

```

\CodeIgniter\Exceptions\PageNotFoundException('Data Pengembalian
Tidak ditemukan !');

}

$data['pageTitle'] = 'Edit Data Pengembalian';

$data['pengembalian'] = $dataPengembalian;

return view('dashboard/pengembalian_edit', $data);

}

public function update($kode_pinjam)

{

if (!$this->validate([

'kode_pinjam' => [

'rules' => 'required',

'errors' => [

'required' => '{field} Harus diisi'

]

],

'id_buku' => [

'rules' => 'required',

'errors' => [

'required' => '{field} Harus diisi'

]

],

'judul' => [

'rules' => 'required',

'errors' => [

'required' => '{field} Harus diisi'

]

]

```

```

    ],

    'tanggal_peminjaman' => [

        'rules' => 'required',

        'errors' => [

            'required' => '{field} Harus diisi'

        ]

    ],

    'jatuh_tempo' => [

        'rules' => 'required',

        'errors' => [

            'required' => '{field} Harus diisi'

        ]

    ],

    'tanggal_kembali' => [

        'rules' => 'required',

        'errors' => [

            'required' => '{field} Harus diisi'

        ]

    ],

    'telat' => [

        'rules' => 'required',

        'errors' => [

            'required' => '{field} Harus diisi'

        ]

    ],

    ])) {

    session()->setFlashdata('error', $this->validator->listErrors());

```

```

return redirect()->back();

}

$this->pengembalian->update($kode_pinjam, [

'kode_pinjam' => $this->request->getVar('kode_pinjam'),

'id_buku' => $this->request->getVar('id_buku'),

'judul' => $this->request->getVar('judul'),

'tanggal_peminjaman' => $this->request->getVar('tanggal_peminjaman'),

'jatuh_tempo' => $this->request->getVar('jatuh_tempo'),

'tanggal_kembali' => $this->request->getVar('tanggal_kembali'),

'telat' => $this->request->getVar('telat'),

]);

session()->setFlashdata('message', 'Update Data Pengembalian

Berhasil');

return redirect()->to('/pengembalian');

}

function delete($kode_pinjam)

{

    $dataPengembalian = $this->pengembalian->find($kode_pinjam);

    if (empty($dataPengembalian)) {

        throw new

        \CodeIgniter\Exceptions\PageNotFoundException('Data Pengembalian

        Tidak ditemukan !');

    }

    $this->pengembalian->delete($kode_pinjam);

    session()->setFlashdata('message', 'Delete Data Pengembalian

    Berhasil');

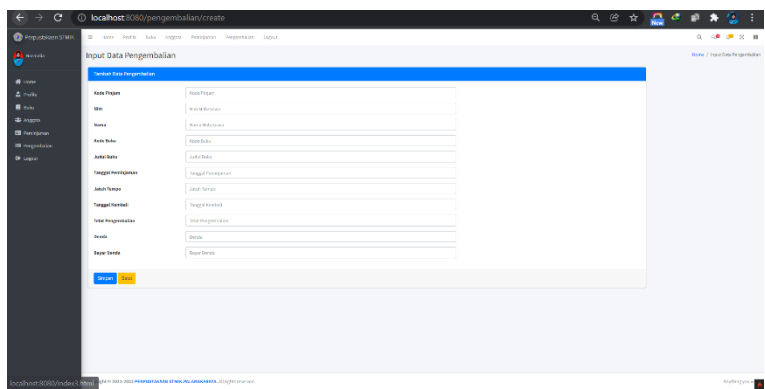
```

```
return redirect()->to('/pengembalian');

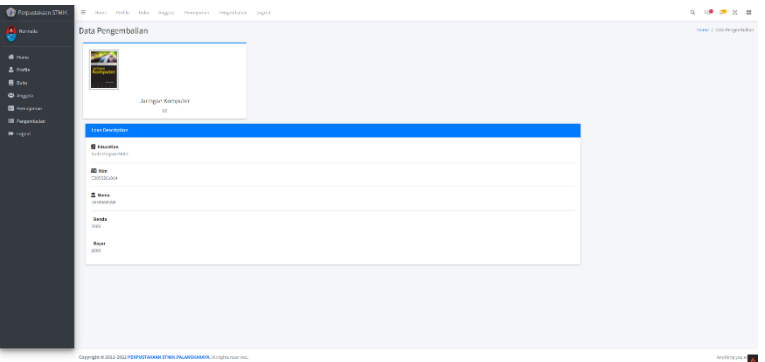
}

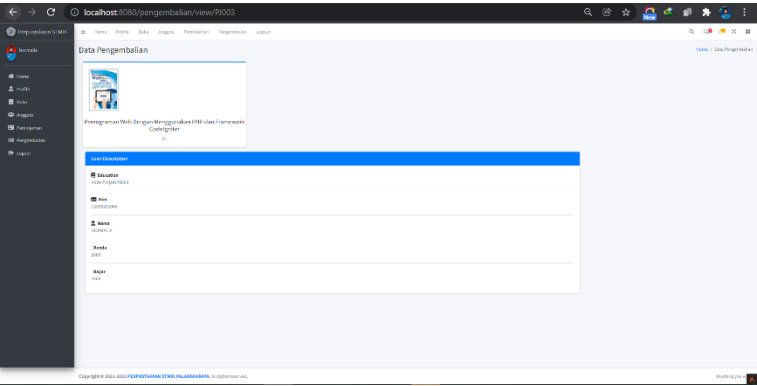
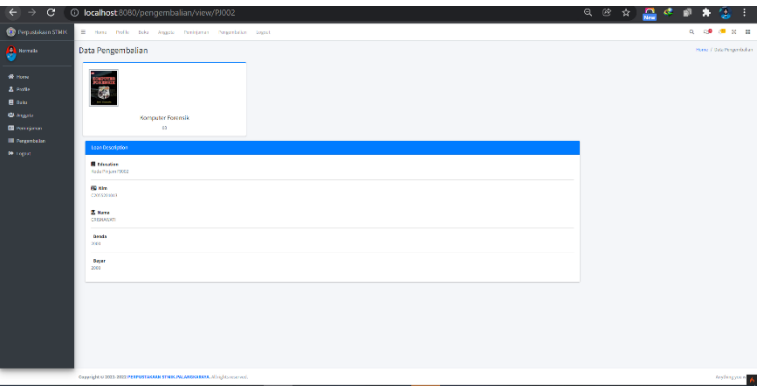
}
```

selanjutnya kita lihat hasilnya pada browser dengan mengetikkan <http://localhost:8080/pengembalian/create> jangan lupa untuk mengaktifkan server development local kita dan database server kita di xampp agar dapat diakses melalui browser.

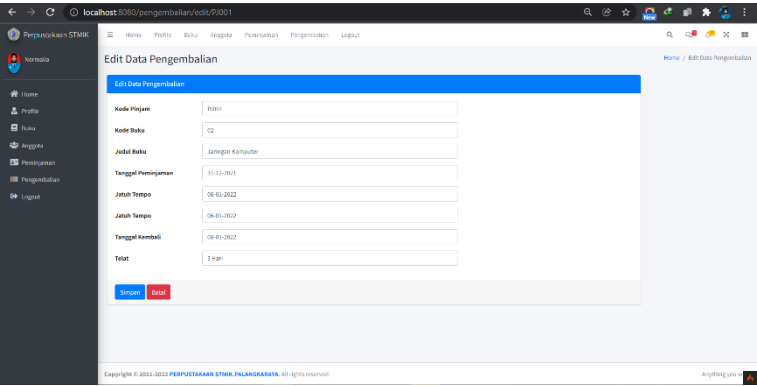


Tampilan view pengembalian

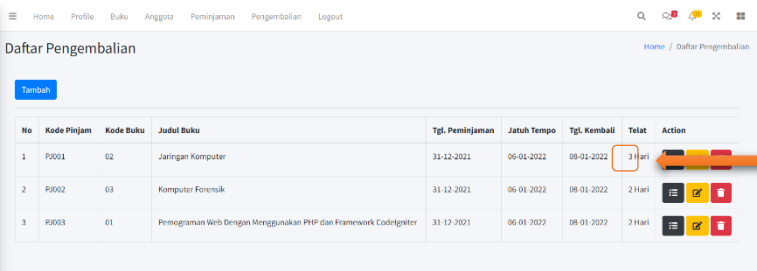




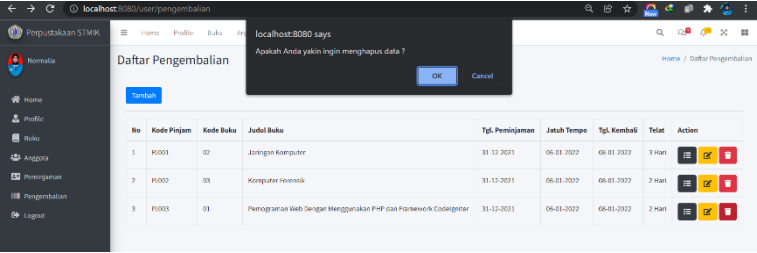
Tampilan Edit Pengembalian



Tampilan Delete Pengembalian



Pada tanda panah orange itu fitur delete jika ditekan akan muncul perintah seperti pada tampilan dibawah ini:



Jika kita klik ok data pengembalian tersebut akan terdelete atau terhapus.

BAB III PENUTUP

A. KESIMPULAN

Berdasarkan hasil pembuatan dan pembahasan yang telah dilakukan, dapat kami simpulkan bahwa:

- Dengan menggunakan framework CodeIgniter PHP dapat dibuat sebuah perangkat lunak aplikasi web sistem informasi untuk membantu pengelolaan data. Dalam penelitian ini, perangkat lunak dikembangkan untuk membantu mengelola data perpustakaan stmik palangkaraya.
- Framework CodeIgniter PHP dapat melakukan kombinasi dokumen, minifikasi data, dan melakukan kompresi GZIP. Selain itu, Framework CodeIgniter PHP juga digunakan untuk melakukan validasi data input, sehingga setiap pengguna memberi masukan data, sistem akan memberi peringatan jika data yang dimasukkan tidak sesuai. Dengan adanya peringatan ini, dapat membantu pengguna untuk memperbaikinya kembali. Halaman-halaman web yang dikembangkan juga terlihat konsisten secara bentuk, warna, tata letak, dsb.
- Terakhir, framework CodeIgniter PHP dengan pendekatan Model-View-Controller membuat proses penulisan kode program saat pengembangan aplikasi berbasis web menjadi lebih mudah. Jadi kualitas perangkat lunak yang dikembangkan dari sisi maintainability sudah baik f. Kualitas perangkat lunak yang dikembangkan dari sisi portability sudah memiliki tampilan yang baik, dapat dilihat dari keseluruhan halaman aplikasi web dapat diakses tanpa error dengan menggunakan berbagai web browser baik desktop maupun mobile.