

# **LAPORAN PRAKTIKUM I**

## **LARAVEL**

Disusun untuk Memenuhi Mata Pelajaran Praktikum Pemrograman Web  
Dibimbing oleh Bapak Ricky Septiandy Pratama Putra Husmildin, S.Pd.

Oleh :

**Dimas Yuda Pratama (17691)**

**XI TKJ 2**



**SEKOLAH MENENGAH KEJURUAN NEGERI 2 MALANG**  
**KONSENTRASI KEAHLIAN TEKNIK KOMPUTER DAN JARINGAN**  
**MARET 2025**

## A. LATIHAN

### 1. Persiapan Lingkungan Pengembangan:

- Kode program

```
C:\Users\ADMIN> composer create-project laravel/laravel perpustakaan-sekolah  
C:\Users\ADMIN> cd perpustakaan-sekolah
```

- Keterangan
  - Install PHP dan Composer
  - Install Laravel menggunakan Composer
  - Buat proyek Laravel baru dengan nama "perpustakaan-sekolah"

### 2. Konfigurasi Database:

- Kode program

```
DB_CONNECTION=mysql  
DB_HOST=127.0.0.1  
DB_PORT=3306  
DB_DATABASE=perpustakaan_db  
DB_USERNAME=root  
DB_PASSWORD=
```

- Keterangan
  - Buat database MySQL baru dengan nama "perpustakaan\_db"
  - Konfigurasi file .env untuk menghubungkan Laravel dengan database

### 3. Membuat Model dan Migration:

- Kode program

```
C:\xampp\htdocs\perpustakaan-sekolah> php artisan make:model Buku -m
```

```
database > migrations > 2025_03_12_052302_create_bukus_table.php > ...
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12      public function up(): void
13      {
14          Schema::create(table: 'bukus', callback: function (Blueprint $table): void {
15              $table->id();
16              $table->string(column: 'judul');
17              $table->string(column: 'penulis');
18              $table->integer(column: 'tahun_terbit');
19              $table->timestamps();
20          });
21      }
22
23      /**
24       * Reverse the migrations.
25       */
26      public function down(): void
27      {
28          Schema::dropIfExists(table: 'bukus');
29      }
30  }
31  };
```

```
C:\xampp\htdocs\perpustakaan-sekolah> php artisan migrate
```

- Keterangan
  - Buat model Buku dengan perintah: **php artisan make:model Buku -m**
  - Edit file migration untuk menambahkan kolom yang diperlukan
  - Jalankan migrasi dengan perintah: **php artisan migrate**

### 4. Membuat Controller:

- Kode program

```
C:\xampp\htdocs\perpustakaan-sekolah> php artisan make:controller BukuController --resource
```

- Keterangan
  - Buat BukuController dengan perintah: **php artisan make:controller BukuController --resource**

## 5. Membuat View:

- Kode program

### Form.blade.php

```
resources > views > buku > Form.blade.php > @style > @container
1  @extends('layouts.app')
2
3  @section('content')
4  <div class="container">
5    <div class="form-card">
6      <h2@if(isset($buku) & 'Edit Buku' < 'Tambah Buku')</h2>
7      <h3>
8
9      <form action="{{(isset($buku) ? route(name: 'buku.update', parameters: $buku->id) : route(name: 'buku.store')) }}" method="POST">
10        @csrf
11        @if(isset($buku))
12          @method('PUT')
13        @endif
14
15        <div class="form-group">
16          <label for="judul">■ Judul Buku</label>
17          <input type="text" name="judul" id="judul" value="{{ old(key: 'judul', default: $buku->judul ?? '') }}" required>
18        </div>
19
20        <div class="form-group">
21          <label for="penulis">■ Penulis</label>
22          <input type="text" name="penulis" id="penulis" value="{{ old(key: 'penulis', default: $buku->penulis ?? '') }}" required>
23        </div>
24
25        <div class="form-group">
26          <label for="tahun_terbit">■ Tahun Terbit</label>
27          <input type="number" name="tahun_terbit" id="tahun_terbit" value="{{ old(key: 'tahun_terbit', default: $buku->tahun_terbit ?? '') }}" required>
28        </div>
29
30        <div class="button-group">
31          <button type="submit" class="btn-primary">■ Simpan</button>
32          <a href="{{ route(name: 'buku.index') }}" class="btn-secondary">◀ Kembali</a>
33        </div>
34      </form>
35    </div>
36  </div>
37
38  <style>
39  .container {
40    display: flex;
41    justify-content: center;
42    align-items: center;
43    min-height: 80vh;
44  }
45
46  .form-card {
47    background: #fff;
48    padding: 20px;
49    border-radius: 10px;
50    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
51
52    width: 100%;
53    max-width: 400px;
54  }
55
56  h2 {
57    text-align: center;
58    margin-bottom: 20px;
59    color: #f70000;
60  }
61
62  .form-group {
63    display: flex;
64    flex-direction: column;
65    margin-bottom: 15px;
66  }
67
68  .form-group label {
69    font-weight: bold;
70    margin-bottom: 5px;
71    color: #333;
72  }
73
74  .form-group input {
75    padding: 10px;
76    border: 1px solid #ddd;
77    border-radius: 6px;
78    font-size: 16px;
79    transition: 0.3s;
80  }
81
82  .form-group input:focus {
83    border-color: #f77000;
84    box-shadow: 0 0 5px rgba(255, 123, 0, 0.5);
85    outline: none;
86  }
87
88  .button-group {
89    display: flex;
90    justify-content: space-between;
91    margin-top: 20px;
92  }
93
94  .btn-primary, .btn-secondary {
95    text-decoration: none;
96    padding: 10px 15px;
97    border-radius: 6px;
98    font-weight: bold;
99    transition: all 0.3s;
100  }
```

```

99     text-align: center;
100     flex: 1;
101     text-align: center;
102 }
103
104 .btn-primary {
105     background: #fff7b0;
106     color: white;
107     border: none;
108 }
109
110 .btn-primary:hover {
111     background: #e6eab0;
112     box-shadow: 0 0 10px #rgba(255, 123, 0, 0.8);
113 }
114
115 .btn-secondary {
116     background: #000;
117     color: black;
118     margin-left: 10px;
119 }
120
121 .btn-secondary:hover {
122     background: #000;
123 }
124 </style>
125 @endsection
126

```

## Index.blade.php

```

resources > views > buku > Index.blade.php > @divcontainer > @br
1 @section('views: 'layouts.app')
2
3 @section(section: 'content')
4 <style>
5     .container {
6         max-width: 900px;
7         margin: auto;
8     }
9
10     h2 {
11         font-weight: bold;
12         color: #fff7b0;
13         background: #rgb(255, 255, 255); /* Warna background */
14         padding: 10px 20px; /* Ruang di dalam */
15         border-radius: 8px; /* Sudut melengkung */
16         width: fit-content; /* Lebar sesuai konten */
17         margin: 0 auto; /* Blar di tengah */
18         text-align: center; /* Teks di tengah */
19         box-shadow: 0 4px 8px #rgba(0, 0, 0, 0.2); /* Efek bayangan */
20     }
21
22     /* Tombol Tambah Buku */
23     .btn-add {
24         display: flex;
25         align-items: center;
26         justify-content: center;
27         background: #fff7b0;
28         color: white;
29         font-weight: bold;
30         border-radius: 10px;
31         padding: 10px 15px;
32         transition: 0.3s;
33         text-decoration: none;
34         width: fit-content;
35         margin: 0 auto 20px auto;
36         text-align: center;
37     }
38
39     .btn-add:hover {
40         background: #e6eab0;
41         transform: scale(1.05);
42         box-shadow: 0 4px 10px #rgba(255, 123, 0, 0.4);
43     }
44
45     /* Card Style */
46     .book-card {
47         display: flex;
48         align-items: center;
49         background: #fff;
50         padding: 10px;

```

```

51     border-radius: 10px;
52     box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
53     margin-bottom: 10px;
54     transition: 0.3s;
55 }
56
57 .book-card:hover {
58     transform: scale(1.02);
59     box-shadow: 0 6px 12px rgba(0, 0, 0, 0.2);
60 }
61
62 .book-info {
63     flex: 1;
64     display: flex;
65     justify-content: space-between;
66     align-items: center;
67     gap: 10px;
68 }
69
70 .book-info span {
71     font-size: 16px;
72     font-weight: bold;
73     color: #333;
74 }
75
76 .book-actions {
77     display: flex;
78     gap: 10px;
79 }
80
81 .btn-edit {
82     margin-left: 10px;
83     background-color: #ffc107;
84     color: white;
85     border: none;
86     padding: 8px 12px;
87     border-radius: 5px;
88     text-decoration: none;
89     transition: 0.3s;
90 }
91
92 .btn-edit:hover {
93     background-color: #ffc107;
94 }
95
96 .btn-delete {
97     background-color: #dc3545;
98     color: white;

```

```

99     border: none;
100     padding: 8px 12px;
101     border-radius: 5px;
102     transition: 0.3s;
103 }
104
105 .btn-delete:hover {
106     background-color: #dc3545;
107 }
108
109 
110
111 <div class="container">
112     <h2>Daftar Buku</h2> <br>
113     <a href="{{ route('buku.create') }}" class="btn-add"> Tambah Buku</a>
114
115     @foreach($buku as $buku)
116     <div class="book-card">
117         <div class="book-info">
118             <span>📖 {{ $buku->judul }}</span>
119             <span>✍️ {{ $buku->penulis }}</span>
120             <span>📖 {{ $buku->status_terbit }}</span>
121         </div>
122         <div class="book-actions">
123             <a href="{{ route('buku.edit', parameters: $buku->id) }}" class="btn-edit"> Edit</a>
124             <form action="{{ route('buku.destroy', parameters: $buku->id) }}" method="POST" class="d-inline">
125                 @csrf
126                 @method('DELETE')
127                 <button type="submit" class="btn-delete" onclick="return confirm('Yakin ingin menghapus?')"> Hapus</button>
128             </form>
129         </div>
130     </div>
131     @endforeach
132 </div>
133 @endsection

```

- Keterangan
  - Buat file view untuk menampilkan daftar buku (index.blade.php)
  - Buat file view untuk form tambah dan edit buku (form.blade.php)

## 6. Implementasi CRUD:

- Kode program

```
app > Http > Controllers > BukuController.php > PHP Integrease > BukuController
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\Buku;
6 use Illuminate\Http\Request;
7
8 // references: 0 implementations
9 class BukuController extends Controller
10 {
11     // references: 0 overrides
12     public function index(): View
13     {
14         $buku = Buku::all();
15         return view('buku.index', data: compact('var_name', 'buku'));
16     }
17
18     // references: 0 overrides
19     public function create(): View
20     {
21         return view('buku.form');
22     }
23
24     // references: 0 overrides
25     public function store(Request $request): RedirectResponse
26     {
27         $request->validate(rules: [
28             'judul' => 'required',
29             'penulis' => 'required',
30             'tahun_terbit' => 'required|integer',
31         ]);
32         $buku = new Buku($request->all());
33         $buku->save();
34         return redirect()->route('buku.index')->with(key: 'success', values: 'Buku berhasil ditambahkan');
35     }
36
37     // references: 0 overrides
38     public function edit(Buku $buku): View
39     {
40         return view('buku.form', data: compact('var_name', 'buku'));
41     }
42
43     // references: 0 overrides
44     public function update(Request $request, Buku $buku): RedirectResponse
45     {
46         $request->validate(rules: [
47             'judul' => 'required',
48             'penulis' => 'required',
49             'tahun_terbit' => 'required|integer',
50         ]);
51         $buku->update(attributes: $request->all());
52         return redirect()->route('buku.index')->with(key: 'success', values: 'Buku berhasil diperbarui');
53     }
54
55     // references: 0 overrides
56     public function destroy(Buku $buku): RedirectResponse
57     {
58         $buku->delete();
59         return redirect()->route('buku.index')->with(key: 'success', values: 'Buku berhasil dihapus');
60     }
61 }
```

```
routes > web.php
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\HomeController;
5 use App\Http\Controllers\BukuController;
6 use App\Http\Controllers\PeminjamanController;
7
8 // Tambahkan route untuk halaman home
9 Route::get(uri: '/', action: [HomeController::class, 'index']);
10
11 // Resource route untuk Buku & Peminjaman
12 Route::resource(name: 'buku', controller: BukuController::class);
13 Route::resource(name: 'peminjaman', controller: PeminjamanController::class);
14
15 // Route khusus untuk mengembalikan buku
16 Route::post(uri: '/peminjaman/{peminjaman}/return', action: [PeminjamanController::class, 'returnBook'])->name(name: 'peminjaman.return');
17
```

#Code **web.php** sudah include dengan konfigurasi studi kasus

- Keterangan
  - Tambahkan logika CRUD di BukuController
  - Buat route di web.php untuk menghubungkan controller dengan view

## 7. Styling dengan CSS:

- Kode program

```
resources > views > layout > app.blade.php > @html > @head > @title
1 <!DOCTYPE html>
2 <html lang="id">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Perpustakaan</title>
8     <style>
9         body {
10             margin: 0;
11             font-family: Arial, sans-serif;
12             background-color: #fff7b0;
13         }
14
15         /* Header */
16         .header {
17             background: linear-gradient(135deg, #ff7b00, #ffae12);
18             padding: 20px;
19             display: flex;
20             align-items: center;
21             justify-content: space-between;
22             color: white;
23             box-shadow: 4px 8px 0px rgba(0, 0, 0, 0.2);
24         }
25
26         .header img {
27             width: 40px;
28             height: 50px;
29             margin-right: 10px;
30             filter: drop-shadow(5px 5px 10px rgba(0, 0, 0, 0.5));
31         }
32
33         .header h1 {
34             margin: 0;
35             font-size: 20px;
36             filter: drop-shadow(5px 5px 6px rgba(0, 0, 0, 0.5));
37         }
38
39         .nav-buttons {
40             display: flex;
41             gap: 20px;
42         }
43
44         .nav-buttons a {
45             text-decoration: none;
46             padding: 10px 20px;
47             border-radius: 8px;
48             font-weight: bold;
49             transition: all 0.3s ease;
50             display: flex;
51
52             align-items: center;
53             gap: 8px;
54         }
55
56         .btn {
57             background-color: white;
58             color: #ff7b00;
59             border: 2px solid #ff7b00;
60         }
61
62         .btn:hover {
63             background: #ff7b00;
64             color: white;
65             box-shadow: 0 0 10px #ff7b00;
66             transform: scale(1.08);
67         }
68
69         /* Main Content */
70         .main-content {
71             display: flex;
72             justify-content: center;
73             align-items: center;
74             height: 70vh;
75             flex-direction: column;
76             text-align: center;
77             padding: 20px;
78         }
79
80         /* Bagian ini untuk konten dinamis dari halaman lain */
81         .content-wrapper {
82             width: 100%;
83         }
84
85         /* Footer */
86         .footer {
87             background: linear-gradient(135deg, #ffae42, #ff7b00);
88             padding: 15px;
89             text-align: center;
90             color: white;
91             margin-top: 30px;
92             width: 100%;
93         }
94     </style>
95 </head>
96 <body>
```



```

97 <!-- Header -->
98 <div class="header">
99 <div style="display: flex; align-items: center;"
100 <a href="{{ url(path: '/') }}" style="display: flex; align-items: center; text-decoration: none; color: white;"
101 
103 <div>Perpustakaan Sekolah</div>
104 </div>
105 <div class="nav-buttons">
106 <a href="{{ route(name: 'buku.index') }}" class="btn">📖 Daftar Buku</a>
107 <a href="{{ route(name: 'peminjaman.index') }}" class="btn">📌 Peminjaman</a>
108 </div>
109 </div>
110
111 <!-- Main Content -->
112 <div class="main-content">
113 <div class="content-wrapper">
114 @yield(section: 'content') <!-- Menampilkan konten dari halaman lain -->
115 </div>
116 </div>
117
118 <!-- Footer -->
119 <div class="footer">
120 <p>©copy, 2025 Perpustakaan Sekolah. SPK Negeri 2 Malang.</p>
121 </div>
122 </body>
123 </html>

```

- Keterangan
  - Tambahkan CSS untuk mempercantik tampilan pada resources/views/layouts/app.blade.php yang menjadi layout utama untuk visual seperti header, nav dan footer. Hanya saja disini saya tidak menggunakan bootstrap tetapi css custom.

## 8. Testing:

- Kode program

```
C:\xampp\htdocs\perpustakaan-sekolah> php artisan serve
```

#untuk output program dari project ini akan saya tampilkan dibagian studi kasus

- Keterangan
  - Uji coba semua fungsi CRUD untuk memastikan sistem berjalan dengan baik

## B. STUDI KASUS

### 1. Menambahkan Fitur Peminjaman Buku :

- Buat model dan migration untuk Peminjaman
- Buat relasi antara model Buku dan Peminjaman
- Buat controller untuk menangani proses peminjaman
- Tambahkan view untuk menampilkan form peminjaman dan daftar peminjaman
- Implementasikan logika untuk mencatat peminjaman dan pengembalian buku

## HASIL

### 1. Kode Program:

```
C:\xampp\htdocs\perpustakaan-sekolah> php artisan make:model Peminjaman -m
```

```
database > migrations > 2025_03_12_065311_create_peminjamans_table.php > ...
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12       public function up(): void
13       {
14           Schema::create(table: 'peminjamans', callback: function (Blueprint $table): void {
15               $table->id();
16               $table->foreignId(column: 'buku_id')->constrained()->onDelete(action: 'cascade');
17               $table->string(column: 'peminjam'); // Nama peminjam
18               $table->date(column: 'tanggal_pinjam');
19               $table->date(column: 'tanggal_kembali')->nullable();
20               $table->timestamps();
21           });
22       }
23
24       /**
25        * Reverse the migrations.
26        */
27       public function down(): void
28       {
29           Schema::dropIfExists(table: 'peminjamans');
30       }
31   };
32
33
```

```
app > Http > Controllers > PeminjamanController.php > PHP Intelephense > PeminjamanController > store
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Peminjaman;
6  use App\Models\Buku;
7  use Illuminate\Http\Request;
8
9  3 references|0 implementations
10 class PeminjamanController extends Controller
11 {
12     0 references|0 overrides
13     public function index(): View
14     {
15         $peminjamans = Peminjaman::with(relations: 'buku')->get();
16         return view(view: 'peminjaman.index', data: compact(var_name: 'peminjamans'));
17     }
18
19     0 references|0 overrides
20     public function create(): View
21     {
22         $bukus = Buku::where(column: 'status', operator: 'tersedia')->get(); // Hanya buku yang tersedia
23         return view(view: 'peminjaman.form', data: compact(var_name: 'bukus'));
24     }
25
26     0 references|0 overrides
27     public function store(Request $request): RedirectResponse
28     {
29         $request->validate(rules: [
30             'buku_id' => 'required|exists:bukus,id',
31             'peminjam' => 'required|string',
32             'tanggal_pinjam' => 'required|date',
33             'tanggal_kembali' => 'nullable|date|after:tanggal_pinjam',
34         ]);
35
36         $buku = Buku::findOrFail(id: $request->buku_id);
37
38         // Cek apakah buku sedang dipinjam
39         if ($buku->status === 'dipinjam') {
40             return redirect()->route(route: 'peminjaman.index')->with(key: 'error', value: 'Buku ini sedang dipinjam!');
41         }
42
43         // Simpan peminjaman
44         $peminjaman = new Peminjaman();
45         $peminjaman->buku_id = $request->buku_id;
46         $peminjaman->peminjam = $request->peminjam;
47         $peminjaman->tanggal_pinjam = $request->tanggal_pinjam;
48         $peminjaman->tanggal_kembali = $request->tanggal_kembali;
49         $peminjaman->save();
50     }
51 }
```

```

47 // Update status buku
48 $buku->update(['status' => 'dipinjam']);
49
50 return redirect()->route('peminjaman.index')->with(key: 'success', value: 'Peminjaman berhasil dicatat');
51
52
53 // references [0] overrides
54 public function destroy(Peminjaman $peminjaman): RedirectResponse
55 {
56     // Ambil buku terkait
57     $buku = $buku->findOrFail($id: $peminjaman->buku_id);
58
59     // Kembalikan status buku menjadi 'tersedia'
60     $buku->update(['status' => 'tersedia']);
61
62     // Hapus peminjaman
63     $peminjaman->delete();
64
65     return redirect()->route('peminjaman.index')->with(key: 'success', value: 'Peminjaman berhasil dihapus');
66 }
67
68 // references [0] overrides
69 public function returnBook(Peminjaman $peminjaman): RedirectResponse
70 {
71     // Ambil buku terkait
72     $buku = $buku->findOrFail($id: $peminjaman->buku_id);
73
74     // Update status buku menjadi 'tersedia'
75     $buku->update(['status' => 'tersedia']);
76
77     // Hapus data peminjaman
78     $peminjaman->delete();
79
80     return redirect()->route('peminjaman.index')->with(key: 'success', value: 'Buku telah dikembalikan');
81 }

```

```
buat file baru pada resources/views/peminjaman/index.blade.php
buat file baru pada resources/views/peminjaman/form.blade.php
```

```

1 <@extends('layouts.app')
2
3 @section('content')
4 <div class="container">
5   <div class="table-card">
6     <h2> Daftar Pinjaman</h2>
7     <a href="{{ route(name: 'pinjaman.create') }}" class="btn-primary"> Tambah Pinjaman</a>
8
9     <div class="table-responsive">
10       <table class="styled-table">
11         <thead>
12           <tr>
13             <th>Buku</th>
14             <th>Pinjaman</th>
15             <th>Tanggal Pinjam</th>
16             <th>Tanggal Kembali</th>
17             <th>Aksi</th>
18           </tr>
19         </thead>
20         <tbody>
21           @foreach($pinjamins as $pinjaman)
22             <tr>
23               <td>{{ $pinjaman->buku->judul }}</td>
24               <td>{{ $pinjaman->pinjaman }}</td>
25               <td>{{ $pinjaman->tanggal_pinjam }}</td>
26               <td>{{ $pinjaman->tanggal_kembali ?? '-' }}</td>
27               <td class="action-buttons">
28                 <!-- Tombol Kembali -->
29                 <form action="{{ route(name: 'pinjaman.return', parameters: [$pinjaman->id]) }}" method="POST">
30                   @csrf
31                   <button type="submit" class="btn-success" onclick="return confirm('Yakin ingin mengembalikan buku ini?')">
32                     Kembali
33                   </button>
34                 </form>
35
36                 <!-- Tombol Hapus -->
37                 <form action="{{ route(name: 'pinjaman.destroy', parameters: [$pinjaman->id]) }}" method="POST">
38                   @csrf
39                   @method("DELETE")
40                   <button type="submit" class="btn-danger" onclick="return confirm('Hapus pinjaman ini?')">
41                     Hapus
42                   </button>
43                 </form>
44               </td>
45             </tr>
46           @endforeach
47         </tbody>
48       </table>
49     </div>
50 </div>

```

```

50 | </div>
51 | </div>
52 |
53 | <style>
54 |   .container {
55 |     display: flex;
56 |     justify-content: center;
57 |     align-items: center;
58 |     min-height: 80vh;
59 |   }
60 |
61 |   .table-card {
62 |     background: #fff;
63 |     padding: 25px;
64 |     border-radius: 10px;
65 |     box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
66 |     width: 100%;
67 |     max-width: 900px;
68 |   }
69 |
70 |   h2 {
71 |     text-align: center;
72 |     margin-bottom: 20px;
73 |     color: #ff7b00;
74 |   }
75 |
76 |   .btn-primary {
77 |     display: block;
78 |     width: fit-content;
79 |     margin: 0 auto 15px;
80 |     background: #ff7b00;
81 |     color: white;
82 |     padding: 10px 15px;
83 |     border-radius: 6px;
84 |     font-weight: bold;
85 |     text-decoration: none;
86 |     text-align: center;
87 |     transition: 0.3s;
88 |   }
89 |
90 |   .btn-primary:hover {
91 |     background: #e66a00;
92 |     box-shadow: 0 0 10px rgba(255, 123, 0, 0.8);
93 |   }
94 |
95 |   .table-responsive {
96 |     overflow-x: auto;
97 |   }

```

```

99 |   .styled-table {
100 |     width: 100%;
101 |     border-collapse: collapse;
102 |     border-radius: 10px;
103 |     overflow: hidden;
104 |     box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
105 |   }
106 |
107 |   .styled-table th, .styled-table td {
108 |     padding: 12px 15px;
109 |     border-bottom: 1px solid #ddd;
110 |     text-align: left;
111 |   }
112 |
113 |   .styled-table th {
114 |     background: #ff7b00;
115 |     color: white;
116 |     text-transform: uppercase;
117 |   }
118 |
119 |   .styled-table tbody tr:nth-child(even) {
120 |     background: #f9f9f9;
121 |   }
122 |
123 |   .styled-table tbody tr:hover {
124 |     background: #f1f1f1;
125 |     transition: 0.3s;
126 |   }
127 |
128 |   .action-buttons {
129 |     display: flex;
130 |     gap: 10px;
131 |   }
132 |
133 |   .btn-success, .btn-danger {
134 |     border: none;
135 |     padding: 8px 12px;
136 |     border-radius: 6px;
137 |     font-weight: bold;
138 |     transition: 0.3s;
139 |     cursor: pointer;
140 |   }
141 |
142 |   .btn-success {
143 |     background: #28a745;
144 |     color: white;
145 |   }

```

```

147 .btn-success:hover {
148     background: #218838;
149 }
150
151 .btn-danger {
152     background: #dc3545;
153     color: white;
154 }
155
156 .btn-danger:hover {
157     background: #c82333;
158 }
159 </style>
160 @endsection

```

```

resources > views > pinjaman > form.blade.php > @style > h2
1 @extends('view:layouts.app')
2
3 @section('section:content')
4 <div class="container">
5     <div class="form-card">
6         <h2>Form Pinjaman</h2>
7         <form action="{{ route('pinjaman.store') }}" method="POST">
8             @csrf
9             <div class="form-group">
10                 <label for="buku_id"> Pilih Buku</label>
11                 <select name="buku_id" id="buku_id" class="form-control">
12                     @foreach($bukus as $buku)
13                         <option value="{{ $buku->id }}">{{ $buku->judul }}</option>
14                     @endforeach
15                 </select>
16             </div>
17
18             <div class="form-group">
19                 <label for="pinjaman"> Nama Pinjaman</label>
20                 <input type="text" name="pinjaman" id="pinjaman" class="form-control" required>
21             </div>
22
23             <div class="form-group">
24                 <label for="tanggal_pinjam"> Tanggal Pinjam</label>
25                 <input type="date" name="tanggal_pinjam" id="tanggal_pinjam" class="form-control" required>
26             </div>
27
28             <div class="form-group">
29                 <label for="tanggal_kembali"> Tanggal Kembali</label>
30                 <input type="date" name="tanggal_kembali" id="tanggal_kembali" class="form-control">
31             </div>
32
33             <button type="submit" class="btn-submit"> Simpan</button>
34         </form>
35     </div>
36 </div>
37
38 <style>
39 .container {
40     display: flex;
41     justify-content: center;
42     align-items: center;
43     min-height: 80vh;
44 }
45
46 .form-card {
47     background: #fff;
48     padding: 25px;
49     border-radius: 10px;
50     box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2);

```

```

51     width: 100%;
52     max-width: 500px;
53 }
54
55 h2 {
56     text-align: center;
57     margin-bottom: 20px;
58     color: #ff7b00;
59 }
60
61 .form-group {
62     margin-bottom: 15px;
63 }
64
65 label {
66     font-weight: bold;
67     display: block;
68     margin-bottom: 5px;
69 }
70
71 .form-control {
72     width: 100%;
73     padding: 10px;
74     border-radius: 6px;
75     border: 1px solid #ddd;
76     transition: 0.3s;
77 }
78
79 .form-control:focus {
80     border-color: #ff7b00;
81     box-shadow: 0 0 8px rgba(255, 123, 0, 0.6);
82 }
83
84 .btn-submit {
85     width: 100%;
86     padding: 10px;
87     background: #ff7b00;
88     color: white;
89     font-size: 16px;
90     border: none;
91     border-radius: 6px;
92     cursor: pointer;
93     transition: 0.3s;
94 }
95
96 .btn-submit:hover {
97     background: #e66a00;
98     box-shadow: 0 0 10px rgba(255, 123, 0, 0.8);
99 }
100 </style>
101 @endsection

```

buat file baru pada app/http/HomeController.php  
 buat file baru pada resources/views/home.blade.php

```

app > Http > Controllers > HomeController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  2 references | 0 implementations
7  class HomeController extends Controller
8  {
9      1 reference | 0 overrides
9      public function index(): View
10     {
11         return view('home');
12     }
13 }

```

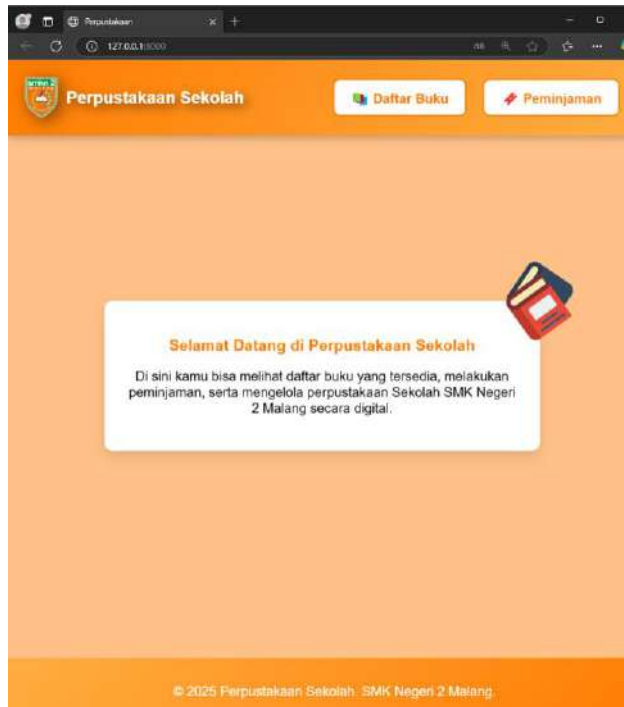
```

resources > views > home.blade.php > @style
1  @extends('layouts.app')
2
3  @section('section', 'content')
4
5  <style>
6      /* Main Content */
7      .main-content {
8          display: flex;
9          justify-content: center;
10         align-items: center;
11         height: 70vh;
12         flex-direction: column;
13         text-align: center;
14         padding: 20px;
15     }
16
17     .info-box {
18         background: white;
19         padding: 25px;
20         border-radius: 10px;
21         box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
22         max-width: 70%;
23         position: relative;
24         animation: float 3s infinite ease-in-out;
25     }
26
27     @keyframes float {
28         0%, 100% { transform: translateY(0); }
29         50% { transform: translateY(-10px); }
30     }
31
32     .info-box h3 {
33         color: #ff7b66;
34         margin-bottom: 10px;
35     }
36
37     .info-box p {
38         color: black;
39         line-height: 1.2;
40         font-weight: 300;
41     }
42
43     .info-box img {
44         width: 80px;
45         position: absolute;
46         top: -40px;
47         right: -40px;
48         animation: rotate 4s linear infinite;
49     }
50
51     @keyframes rotate {
52         0% { transform: rotate(0deg); }
53         100% { transform: rotate(360deg); }
54     }
55 </style>
56
57 <div class="main-content">
58     <div class="info-box">
59         
60         <h3>Selamat Datang di Perpustakaan Sekolah</h3>
61         <p>Di sini kamu bisa melihat daftar buku yang tersedia, melakukan peminjaman, serta mengelola perpustakaan Sekolah SMP Negeri 2 Malang secara digital.</p>
62     </div>
63 </div>

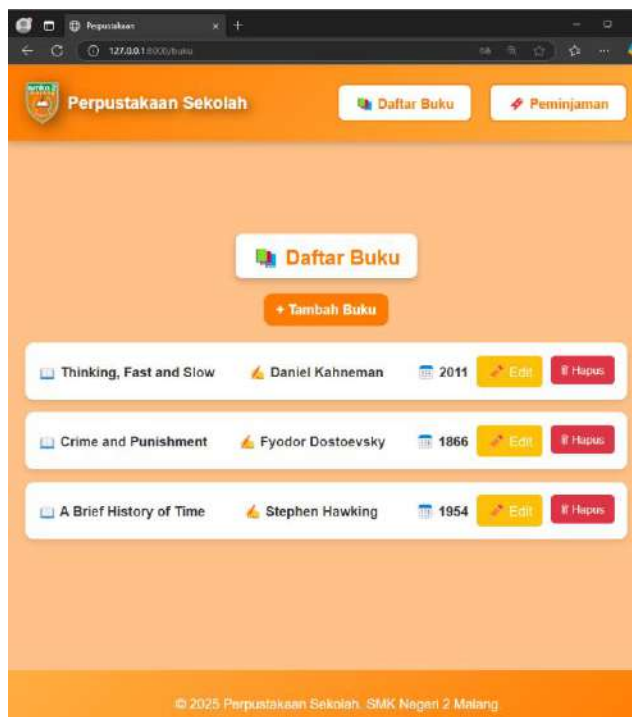
```

## 2. Output Program:

### #homepage



### #daftar\_buku





## #tambah\_buku

The screenshot shows the 'Tambah Buku' form in the 'Perpustakaan Sekolah' application. The form has three input fields: 'Judul Buku', 'Penulis', and 'Tahun Terbit'. Below the fields are two buttons: 'Simpan' (Save) and 'Kembali' (Back). The application header includes the school logo and name, and navigation buttons for 'Daftar Buku' and 'Peminjaman'.

Perpustakaan Sekolah

Daftar Buku Peminjaman

### Tambah Buku

Judul Buku

Penulis

Tahun Terbit

Simpan Kembali

© 2025 Perpustakaan Sekolah, SMK Negeri 2 Malang.

## #daftar\_pinjam

The screenshot shows the 'Daftar Peminjaman' table in the 'Perpustakaan Sekolah' application. The table has five columns: 'BUKU', 'PEMINJAM', 'TANGGAL PINJAM', 'TANGGAL KEMBALI', and 'AKSI'. There are two rows of data. Above the table is a 'Tambah Peminjaman' button. The application header includes the school logo and name, and navigation buttons for 'Daftar Buku' and 'Peminjaman'.

Perpustakaan Sekolah

Daftar Buku Peminjaman

### Daftar Peminjaman

Tambah Peminjaman

BUKU	PEMINJAM	TANGGAL PINJAM	TANGGAL KEMBALI	AKSI
Thinking, Fast and Slow	windah	2025-03-07	2025-03-17	Kembalikan Hapus
Crime and Punishment	dimas	2025-03-18	2025-05-07	Kembalikan Hapus

© 2025 Perpustakaan Sekolah, SMK Negeri 2 Malang.

## #tambah\_peminjam

Perpustakaan Sekolah

Daftar Buku Peminjaman

### Form Peminjaman

Pilih Buku  
A Brief History of Time

Nama Peminjam

Tanggal Pinjam  
mm/dd/yyyy

Tanggal Kembali  
mm/dd/yyyy

Simpan

© 2025 Perpustakaan Sekolah - SMK Negeri 2 Malang

### 3. Keterangan:

Dalam studi kasus ini, saya menambahkan fitur peminjaman buku ke dalam sistem. Pertama, saya membuat model dan migration untuk **Peminjaman** agar data peminjaman dapat tersimpan di database. Selanjutnya, saya menetapkan relasi antara model **Buku** dan **Peminjaman**, sehingga setiap peminjaman terkait dengan buku tertentu.

Kemudian, saya membuat **PeminjamanController** untuk menangani proses peminjaman dan pengembalian buku. Di dalamnya, terdapat metode **store()** untuk menyimpan data peminjaman dan **return()** untuk mencatat pengembalian buku.

Untuk tampilan, saya membuat view **peminjaman/index.blade.php** yang menampilkan daftar peminjaman beserta tombol untuk mengembalikan

atau menghapus data peminjaman. Selain itu, saya menambahkan form di `peminjaman/form.blade.php` agar pengguna dapat mencatat peminjaman baru.

Terakhir, saya menghubungkan fitur ini dengan routing di `web.php`, memastikan bahwa setiap proses peminjaman dan pengembalian dapat berjalan dengan baik.