

**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('bukus', function (Blueprint $table): void {
15             $table->id();
16             $table->string('judul');
17             $table->string('penulis');
18             $table->integer('tahun_terbit');
19             $table->timestamps();
20         });
21     }
22
23
24     /**
25     * Reverse the migrations.
26     */
27     public function down(): void
28     {
29         Schema::dropIfExists('bukus');
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 > form.css
1  <xmp><!--view: 'layouts.app' -->
2
3  @section('content')
4  <div class="container">
5      <div class="form-card">
6          <form action="{{ isset($buku) ? route('buku.update', parameters: $buku->id) : route('name: 'buku.store') }}" method="POST">
7              @csrf
8              @if(isset($buku))
9                  @method('PUT')
10              @endif
11
12              <div class="form-group">
13                  <label for="juluk">■ Judul Buku</label>
14                  <input type="text" name="buku[id]" id="juluk" value="{{ old(key: 'juluk', default: $buku->juluk ?? '') }}" required>
15              </div>
16
17              <div class="form-group">
18                  <label for="penulis">■ Penulis</label>
19                  <input type="text" name="penulis" id="penulis" value="{{ old(key: 'penulis', default: $buku->penulis ?? '') }}" required>
20              </div>
21
22              <div class="form-group">
23                  <label for="tahun_terbit">■ Tahun Terbit</label>
24                  <input type="number" name="tahun_terbit" id="tahun_terbit" value="{{ old(key: 'tahun_terbit', default: $buku->tahun_terbit ?? '') }}" required>
25              </div>
26
27              <div class="button-group">
28                  <button type="submit" class="btn-primary" █ Simpan</button>
29                  <a href="{{ route('name: buku.index') }}" class="btn-secondary" █ Kembali</a>
30              </div>
31          </form>
32      </div>
33  </div>
34
35  <style>
36      .container {
37          display: flex;
38          justify-content: center;
39          align-items: center;
40          min-height: 80vh;
41      }
42
43      .form-card {
44          background: #fff;
45          padding: 25px;
46          border-radius: 10px;
47          box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
48
49      }
50
51      width: 100%;
52      max-width: 400px;
53  }
54
55  h2 {
56      text-align: center;
57      margin-bottom: 20px;
58      color: █ #FF7043;
59  }
56
60  .form-group {
61      display: flex;
62      flex-direction: column;
63      margin-bottom: 15px;
64  }
65
66  .form-group label {
67      font-weight: bold;
68      margin-bottom: 5px;
69      color: #333;
70  }
71
72
73  .form-group input {
74      padding: 10px;
75      border: 1px solid #ccc;
76      border-radius: 6px;
77      font-size: 16px;
78      transition: 0.3s;
79  }
80
81  .form-group input:focus {
82      border-color: █ #FF7043;
83      box-shadow: 0 0 5px rgba(255, 123, 0, 0.5);
84      outline: none;
85  }
86
87  .button-group {
88      display: flex;
89      justify-content: space-between;
90      margin-top: 20px;
91  }
92
93  .btn-primary, .btn-secondary {
94      text-decoration: none;
95      padding: 10px 25px;
96      border-radius: 6px;
97      font-weight: bold;
98      transition: all 0.3s;
99  }
```

```

95     text-align: center;
96     flex: 1;
97     text-align: center;
98   }
99
100  .btn-primary {
101    background: #ff7b00;
102    color: white;
103    border: none;
104  }
105
106  .btn-primary:hover {
107    background: #e6a000;
108    box-shadow: 0 0 10px rgba(255, 123, 0, 0.8);
109  }
110
111  .btn-secondary {
112    background: #ddd;
113    color: black;
114    margin-left: 10px;
115  }
116
117  .btn-secondary:hover {
118    background: #bbb;
119  }
120
121  </style>
122  @endsection
123
124
125
126

```

Index.blade.php

```

resources > views > buku > Indexblade.php > div.container > br
1  @extends('views.layouts.app')
2
3  @section('section', 'content')
4  <style>
5    .container {
6      max-width: 900px;
7      margin: auto;
8    }
8
9
10  h2 {
11    font-weight: bold;
12    color: #ff7b00;
13    background: linear-gradient(255, 250, 250); /* 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; /* Blitar di tengah */
18    text-align: center; /* Teks di tengah */
19    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2); /* Efek bayangan */
20  }
20
21
22  /* Tombol Tambah Buku */
23  .btn-add {
24    display: flex;
25    align-items: center;
26    justify-content: center;
27    background: #ff7b00;
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  }
37
38
39  .btn-add:hover {
40    background: #e6a000;
41    transform: scale(1.05);
42    box-shadow: 0 4px 18px rgba(255, 123, 0, 0.4);
43  }
43
44
45  /* Card Style */
46  .book-card {
47    display: flex;
48    align-items: center;
49    background: #fff;
50    padding: 15px;
50

```

```

51    border-radius: 10px;
52    box-shadow: 8px 8px 0px rgba(0, 0, 0, 0.1);
53    margin-bottom: 15px;
54    transition: 0.3s;
55  }
56
57  .book-card:hover {
58    transform: scale(1.02);
59    box-shadow: 8px 12px 0px 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: #ffcc00;
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: #e6a800;
94  }
95
96  .btn-delete {
97    background: #dc3545;
98    color: white;
99
100   border: none;
101   padding: 8px 12px;
102   border-radius: 5px;
103   transition: 0.3s;
104 }
105
106 .btn-delete:hover {
107   background: #e82333;
108 }
109
110 </style>
111
112 <div class="container">
113   <h2>Daftar Buku</h2> <br>
114   <a href="{{ route('buku.create') }}" class="btn-add">Tambah Buku</a>
115
116   @foreach($bukus as $buku)
117   <div class="book-card">
118     <div class="book-info">
119       <span>{{ $buku->judul }}</span>
120       <span>{{ $buku->penulis }}</span>
121       <span>{{ $buku->tahun_terbit }}</span>
122     </div>
123     <div class="book-actions">
124       <a href="{{ route('buku.edit', ['id' => $buku->id]) }}" class="btn-edit"> Edit </a>
125       <form action="{{ route('buku.destroy', ['id' => $buku->id]) }}" method="POST" class="d-inline">
126         @csrf
127         @method('DELETE')
128         <button type="submit" class="btn-delete" onclick="return confirm('Yakin ingin menghapus?')"> Hapus </button>
129       </form>
130     </div>
131   </div>
132   @endforeach
133 </div>
134 </div>

```

- 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 Implementation > BukuController
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\Buku;
6 use Illuminate\Http\Request;
7
8 references|0 overrides
9 class BukuController extends Controller
10 {
11     references|0 overrides
12     public function index(): View
13     {
14         $bukus = Buku::all();
15         return view('view/buku/index', compact('var_name', 'bukus'));
16     }
17
18     references|0 overrides
19     public function create(): View
20     {
21         return view('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
33         Buku::create(attributes: $request->all());
34
35         return redirect()->route(route: 'buku.index')->with(key: 'success', value: 'Buku berhasil ditambahkan');
36     }
37
38     references|0 overrides
39     public function edit(Buku $buku): View
40     {
41         return view('view/buku.form', compact('var_name', 'buku'));
42     }
43
44     references|0 overrides
45     public function update(Request $request, Buku $buku): RedirectResponse
46     {
47         $buku->update(attributes: $request->all());
48
49         return redirect()->route(route: 'buku.index')->with(key: 'success', value: 'Buku berhasil diperbarui');
50     }
51
52     references|0 overrides
53     public function destroy(Buku $buku): RedirectResponse
54     {
55         $buku->delete();
56
57         return redirect()->route(route: 'buku.index')->with(key: 'success', value: 'Buku berhasil dihapus');
58     }
}
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(url: '/', 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(url: '/peminjaman/{peminjaman}/return', action: [PeminjamanController::class, 'returnBook'])->name(name: 'peminjaman.return');
```

#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 > layouts > 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
9      <style>
10         body {
11             margin: 0;
12             font-family: Arial, sans-serif;
13             background-color: #fff7b077;
14         }
15
16         /* Header */
17         .header {
18             background: linear-gradient(135deg, #ff7b00, #ff7b00);
19             padding: 20px;
20             display: flex;
21             align-items: center;
22             justify-content: space-between;
23             color: white;
24             box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
25         }
26
27         .header-img {
28             width: 40px;
29             height: 50px;
30             margin-right: 10px;
31             filter: drop-shadow(5px 5px 10px rgba(0, 0, 0, 0.5));
32         }
33
34         .header h1 {
35             margin: 0;
36             font-size: 20px;
37             filter: drop-shadow(5px 5px 6px rgba(0, 0, 0, 0.5));
38         }
39
40         .nav-buttons {
41             display: flex;
42             gap: 20px;
43         }
44
45         .nav-buttons a {
46             text-decoration: none;
47             padding: 10px 20px;
48             border-radius: 5px;
49             font-weight: bold;
50             transition: all 0.3s ease;
51             display: flex;
52
53             align-items: center;
54             gap: 8px;
55         }
56
57         .btn {
58             background-color: white;
59             color: #ff7b00;
60             border: 2px solid #fff7b00;
61         }
62
63         .btn:hover {
64             background: #ff7b00;
65             color: white;
66             box-shadow: 0 0 10px rgba(255, 123, 0, 0.8);
67             transform: scale(1.08);
68         }
69
70         /* Main Content */
71         .main-content {
72             display: flex;
73             justify-content: center;
74             align-items: center;
75             height: 70vh;
76             flex-direction: column;
77             text-align: center;
78             padding: 20px;
79         }
80
81         /* Bagian ini untuk konten dinamis dari halaman lain */
82         .content-wrapper {
83             width: 100%;
84         }
85
86         /* Footer */
87         .footer {
88             background: linear-gradient(135deg, #ffac42, #ff7b00);
89             padding: 10px;
90             text-align: center;
91             color: white;
92             margin-top: 20px;
93             width: 100%;
94         }
95     </style>
96  </head>
97
98  <body>
```

```

97     <!-- Header -->
98     <div class="header">
99       <div style="display: flex; align-items: center;">
100         <a href="{{ url('') }}" style="display: flex; align-items: center; text-decoration: none; color: white;">
101           
103           <div>Perpustakaan Sekolah</div>
104         </a>
105       </div>
106       <div class="nav-buttons">
107         <a href="{{ route('buku.index') }}" class="btn"> Daftar Buku</a>
108         <a href="{{ route('peminjaman.index') }}" class="btn"> Peminjaman</a>
109       </div>
110     </div>
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     <!-- Footer -->
118     <div class="footer">
119       <p>© 2025 Perpustakaan Sekolah, SMK Negeri 2 Malang.</p>
120     </div>
121   </body>
122 </html>
123
124

```

- 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    /**
11     * Run the migrations.
12     */
13    public function up(): void
14    {
15        Schema::create('peminjamans', function (Blueprint $table): void {
16            $table->id();
17            $table->foreignId(column: 'buku_id')->constrained()->onDelete(action: 'cascade');
18            $table->string(column: 'peminjam'); // Nama peminjam
19            $table->date(column: 'tanggal_pinjam');
20            $table->date(column: 'tanggal_kembali')->nullable();
21            $table->timestamps();
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  class PeminjamanController extends Controller
10 {
11
12     /**
13      * Index
14      */
15     public function index(): View
16     {
17         $peminjamans = Peminjaman::with(relations: 'buku')->get();
18         return view(view: 'peminjaman.index', data: compact(var_name: 'peminjamans'));
19     }
20
21     /**
22      * Create
23      */
24     public function create(): View
25     {
26         $bukus = Buku::where(column: 'status', operator: 'tersedia')->get(); // Hanya buku yang tersedia
27         return view(view: 'peminjaman.form', data: compact(var_name: 'bukus'));
28     }
29
30     /**
31      * Store
32      */
33     public function store(Request $request): RedirectResponse
34     {
35         $request->validate(rules: [
36             'buku_id' => 'required|exists:bukus,id',
37             'peminjam' => 'required|string',
38             'tanggal_pinjam' => 'required|date',
39             'tanggal_kembali' => 'nullable|date|after:tanggal_pinjam',
40         ]);
41
42         $buku = Buku::findOrFail(id: $request->buku_id);
43
44         // Cek apakah buku sedang dipinjam
45         if ($buku->status === 'dipinjam') {
46             return redirect()->route(route: 'peminjaman.index')->with(key: 'error', value: 'Buku ini sedang dipinjam!');
47         }
48
49         // Simpan peminjaman
50         $peminjaman = new Peminjaman();
51         $peminjaman->buku_id = $request->buku_id;
52         $peminjaman->peminjam = $request->peminjam;
53         $peminjaman->tanggal_pinjam = $request->tanggal_pinjam;
54         $peminjaman->tanggal_kembali = $request->tanggal_kembali;
55         $peminjaman->save();
56     }
57 }
```

```

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 
```

References [3] (minim)

```

53 public function destroy(Peminjaman $peminjaman): RedirectResponse
54 {
55     // Ambil buku terkait
56     $buku = Buku::findOrFail($peminjaman->buku_id);
57
58     // Kembalikan status buku menjadi 'tersedia'
59     $buku->update(['status' => 'tersedia']);
60
61     // Hapus peminjaman
62     $peminjaman->delete();
63
64     return redirect()->route('peminjaman.index')->with(key:'success', value: 'Peminjaman berhasil dihapus');
65 }
66
67 
```

References [0] (minim)

```

68 public function returnbook(Peminjaman $peminjaman): RedirectResponse
69 {
70     // Ambil buku terkait
71     $buku = Buku::findOrFail($peminjaman->buku_id);
72
73     // Update status buku menjadi 'tersedia'
74     $buku->update(['status' => 'tersedia']);
75
76     // Hapus data peminjaman
77     $peminjaman->delete();
78
79     return redirect()->route('peminjaman.index')->with(key:'success', value: 'Buku telah dikembalikan');
80 }
81

```

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

```

resources > views > peminjaman > index.blade.php > div.container > div.table-card > h2
1 <extends view='layouts.app'>
2
3 <section section='content'>
4 <div class='container'>
5     <div class='table-card'>
6         <h2>Data Peminjaman</h2>
7         <a href="{{ route('peminjaman.create') }}" class='btn-primary'>Tambah Peminjaman</a>
8
9         <div class='table-responsive'>
10            <table class='styled-table'>
11                <thead>
12                    <tr>
13                        <th>Buku</th>
14                        <th>Peminjaman</th>
15                        <th>Tanggal Pinjam</th>
16                        <th>Tanggal Kembali</th>
17                        <th>aksi</th>
18                    </tr>
19                </thead>
20                <tbody>
21                    @foreach($peminjamans as $peminjaman)
22                    <tr>
23                        <td>{{ $peminjaman->buku->judul }}</td>
24                        <td>{{ $peminjaman->peminjaman }}</td>
25                        <td>{{ $peminjaman->tanggal_pinjam }}</td>
26                        <td>{{ $peminjaman->tanggal_kembali ?? '-' }}</td>
27                        <td class='action-buttons'>
28                            <!-- Tombol Kembalikan -->
29                            <form action="{{ route('peminjaman.return', parameters: $peminjaman->id) }}" method="POST">
30                                @csrf
31                                <button type="submit" class="btn-success" onclick="return confirm('Yakin ingin mengembalikan buku ini?')">
32                                    Kembalikan
33                                </button>
34                            </form>
35
36                            <!-- Tombol Hapus -->
37                            <form action="{{ route('peminjaman.destroy', parameters: $peminjaman->id) }}" method="POST">
38                                @csrf
39                                @method('DELETE')
40                                <button type="submit" class="btn-danger" onclick="return confirm('Hapus peminjaman ini?')">
41                                    X 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    }
59
60    .table-card {
61      background: #fff;
62      padding: 25px;
63      border-radius: 10px;
64      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
65      width: 180px;
66      max-width: 900px;
67    }
67
68    h2 {
69      text-align: center;
70      margin-bottom: 20px;
71      color: #f77b00;
72    }
72
73    .btn-primary {
74      display: block;
75      width: fit-content;
76      margin: 0 auto 15px;
77      background: #fff77b00;
78      color: white;
79      padding: 10px 15px;
80      border-radius: 6px;
81      font-weight: bold;
82      text-decoration: none;
83      text-align: center;
84      transition: 0.3s;
84
85    }
85
86    .btn-primary:hover {
87      background: #e66a00;
88      box-shadow: 0 0 10px rgba(255, 123, 0, 0.8);
89    }
89
90    .table-responsive {
91      overflow-x: auto;
92    }
92
92
93    .styled-table {
94      width: 100%;
95      border-collapse: collapse;
96      border-radius: 10px;
97      overflow: hidden;
98      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
99    }
99
100   .styled-table th, .styled-table td {
101     padding: 12px 15px;
102     border-bottom: 1px solid #add;
103     text-align: left;
104   }
104
105   .styled-table th {
106     background: #fff77b00;
107     color: white;
108     text-transform: uppercase;
109   }
109
110   .styled-table tbody tr:nth-child(even) {
111     background: #f9f9f9;
112   }
112
113   .styled-table tbody tr:hover {
114     background: #f1f1f1;
115     transition: 0.3s;
116   }
116
117   .action-buttons {
118     display: flex;
119     gap: 10px;
120   }
120
121   .btn-success, .btn-danger {
122     border: none;
123     padding: 8px 12px;
124     border-radius: 6px;
125     font-weight: bold;
126     transition: 0.3s;
127     cursor: pointer;
128   }
128
129   .btn-success {
130     background: #28a745;
131     color: white;
132   }

```

```

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 
```

```

160 </style>
161 @endsection

```

```

resources > views > pinjamank > form.blade.php > style > h2
1   @extends('view.layouts.app')
2
3   @section('content')
4   <div class="container">
5       <div class="form-card">
6           <h2>Form Pinjamank</h2>
7           <form action="{{ route('pinjamank.store') }}" method="POST">
8               <button type="submit" class="btn-submit">Simpan</button>
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="peminjam"> Nama Peminjam</label>
20                  <input type="text" name="peminjam" id="peminjam" 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 16px rgba(0, 0, 0, 0.2);
51     }

```

```

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  class HomeController extends Controller
8  {
9      public function index(): View
10     {
11         return view('home');
12     }
13 }

```

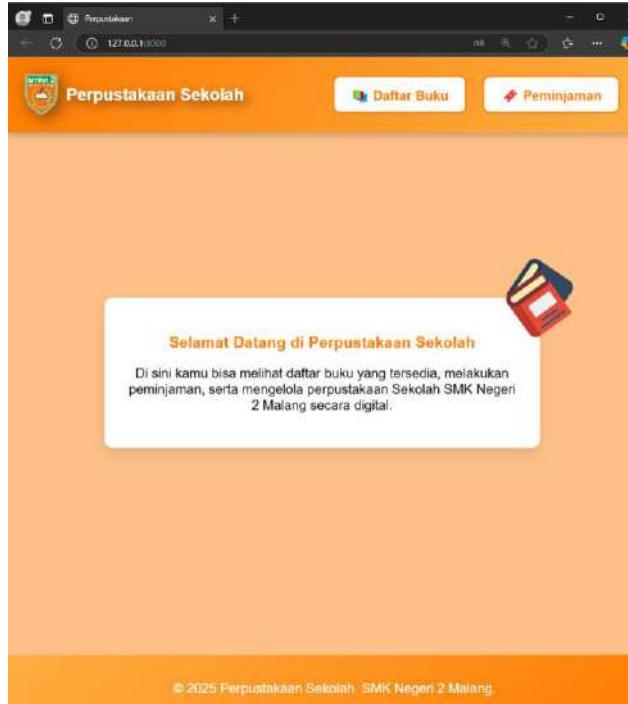
```

resources > views >  home.blade.php >  style.css
1  @extends('views.layouts.app')
2
3  @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: 30px;
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: 700px;
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: #FF7043;
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 SMK Negeri 2 Malang secara digital.</p>
62     </div>
63 </div>

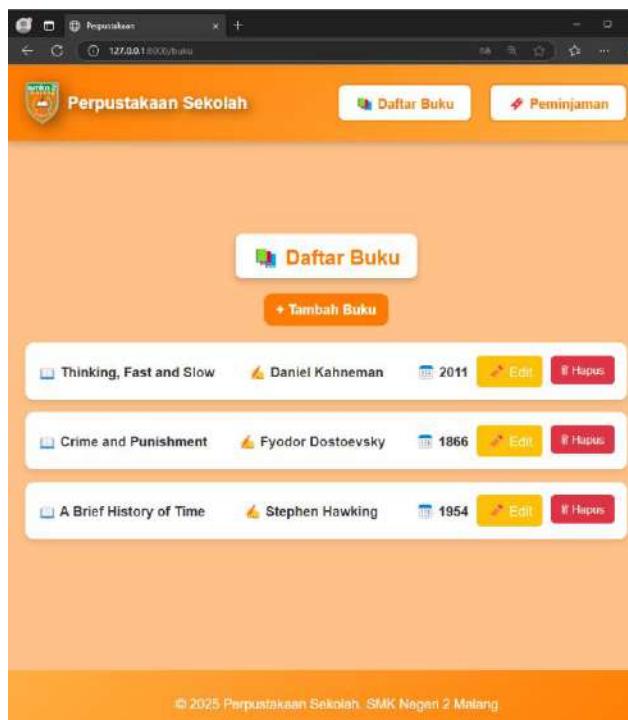
```

2. Output Program:

#homepage



#daftar_buku



#tambah_buku

Tambah Buku

Judul Buku

Penulis

Tahun Terbit

Simpan Kembali

#daftar_pinjam

Daftar Peminjaman

Tambah Peminjaman

BUKU	PEMINJAM	TANGGAL PINJAM	TANGGAL KEMBALI	AKSI
Thinking, Fast and Slow	windah	2025-03-07	2025-03-17	<button>5 Kembalikan</button> <button>Hapus</button>
Crime and Punishment	dimas	2025-03-18	2025-05-07	<button>6 Kembalikan</button> <button>Hapus</button>

© 2025 Perpustakaan Sekolah, SMK Negeri 2 Malang.

#tambah_peminjam

The screenshot shows a web-based library management system. The main page has a header with the school library logo and navigation links for 'Daftar Buku' and 'Peminjaman'. A central modal window titled 'Form Peminjaman' is displayed, containing fields for selecting a book, entering the borrower's name, and specifying the loan and return dates. An orange 'Simpan' button at the bottom right of the modal is used to save the loan record.

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.