

Bootcamp Mandiri

Uji Sertifikasi Kompetensi

LSP SMK Telkom Sandhy Putra Jakarta



Kompetensi Keahlian
Pengembangan Perangkat Lunak dan Gim
SMK Telkom Jakarta
2026

Instalasi

A. Install Composer untuk Laravel

Composer digunakan sebagai dependency manager untuk mengelola library yang dibutuhkan oleh Laravel. Instalasi dilakukan dengan mengunduh file Composer-Setup.exe dari situs resmi dan mengatur path PHP ke direktori xampp/php/php.exe agar Composer dapat terhubung dengan PHP. Setelah instalasi selesai, Composer diuji melalui Command Prompt menggunakan perintah composer untuk memastikan instalasi berhasil.

1. Buka <https://getcomposer.org/download/>

Download Composer Latest: v2.9.3

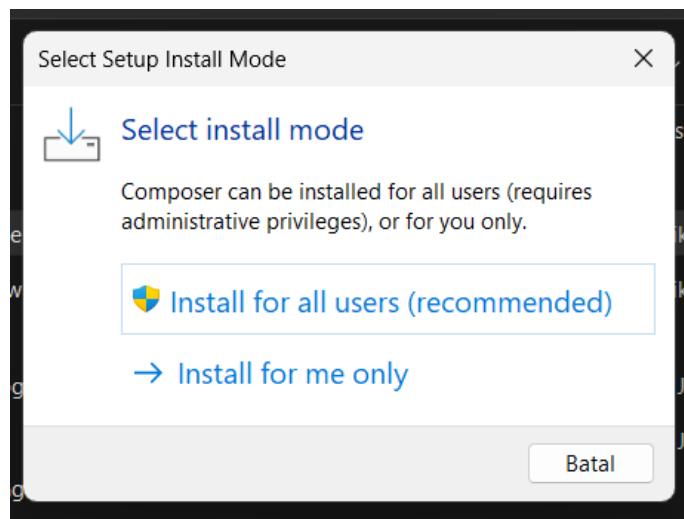
Windows Installer

The installer - which requires that you have PHP already installed - will download Composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

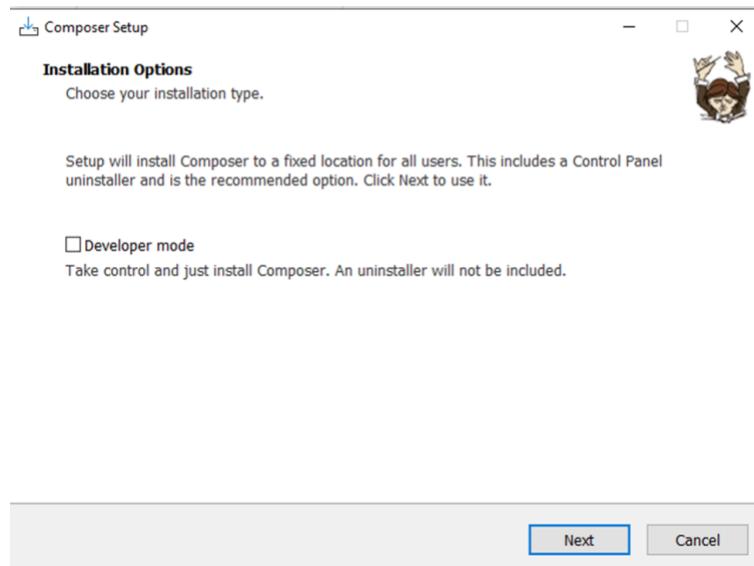
Download and run [Composer-Setup.exe](#) - it will install the latest composer version whenever it is executed.

 PRIVATE PACKAGIST
Secure & reliable package repository

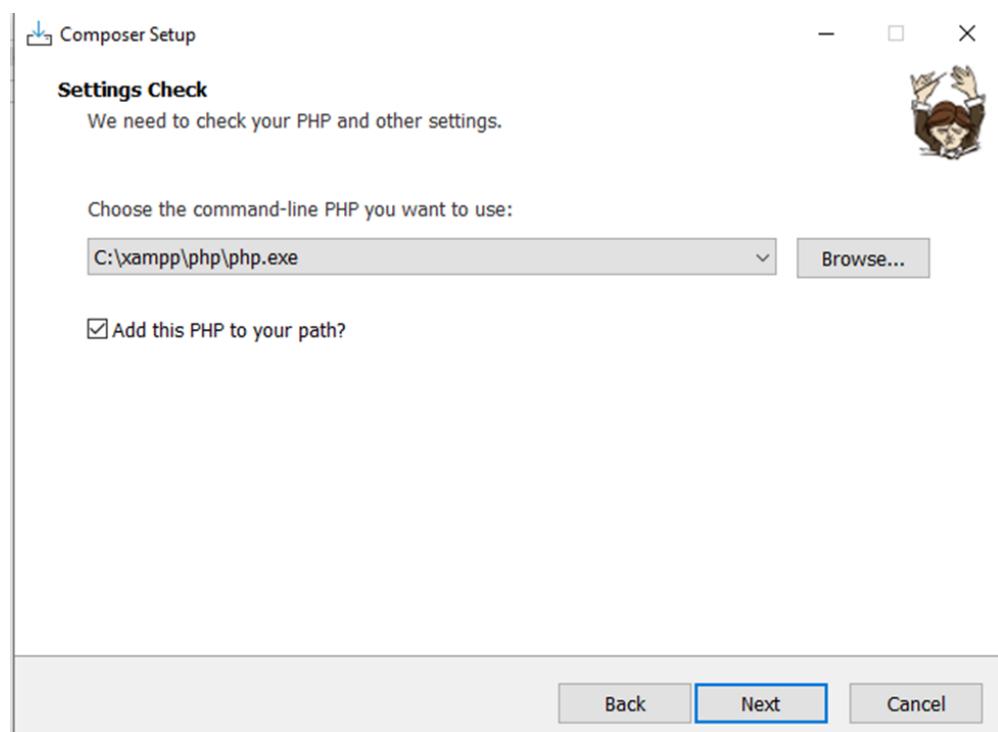
2. Buka file Composer-Setup.exe > Install for all user



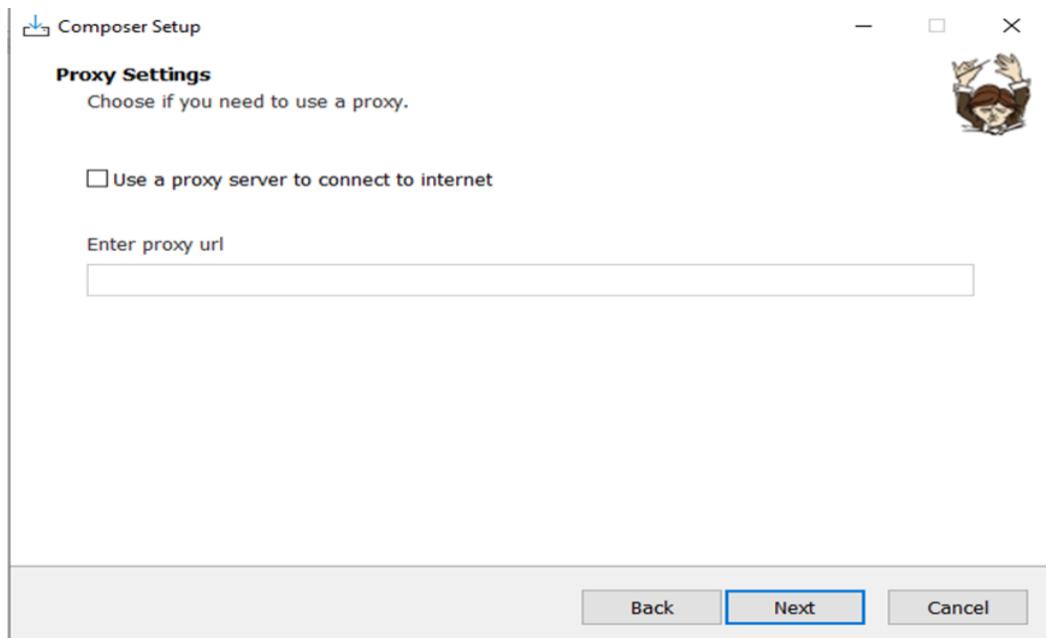
3. Pilih Next



4. Atur path menjadi di folder xampp\php\php.exe, lalu next



5. Atur proxy (jika menggunakan internet, masukkan url) jika tidak maka bisa next.



6. Lalu setelahnya tinggal install dan next sampai finish.

B. Cek Keberhasilan Install Composer

- ## 1. Buka cmd, Jalankan perintah composer

```
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

C:\Users\user>composer
```

2. Jika tampilan seperti dibawah ini, maka composer berhasil dijalankan.

```
Command Prompt
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>composer

Composer version 2.2.24 2024-06-10 22:51:52

Usage:
  command [options] [arguments]

Options:
  -h, --help                  Display this help message
  -q, --quiet                 Do not output any message
  -V, --version                Display this application version
  --ansi                      Force ANSI output
  --no-ansi                   Disable ANSI output
  -n, --no-interaction        Do not ask any interactive question
  --profile                   Display timing and memory usage information
  --no-plugins                Whether to disable plugins.
  --no-scripts                Skips the execution of all scripts defined in composer.json file.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache                  Prevent use of the cache
  -v|vv|vvv, --verbose        Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and
  3 for debug
```

Install Xampp versi 8.2 (skip karena udah pasti pada punya)

C. Membuat Projek Laravel 12

Proyek Laravel dibuat dengan menyiapkan folder kosong, kemudian menjalankan perintah composer create-project melalui Command Prompt. Proses ini bertujuan untuk menghasilkan struktur dasar aplikasi Laravel. Setelah proyek berhasil dibuat, aplikasi dijalankan menggunakan perintah php artisan serve dan diuji melalui browser.

1. Buat folder kosong
2. Buka folder tersebut menggunakan cmd
3. Ketik perintah `composer create-project laravel/laravel:^12.0 nama-project-kamu` pada cmd

```
composer create-project laravel/laravel:^12.0 nama-project-kamu
```

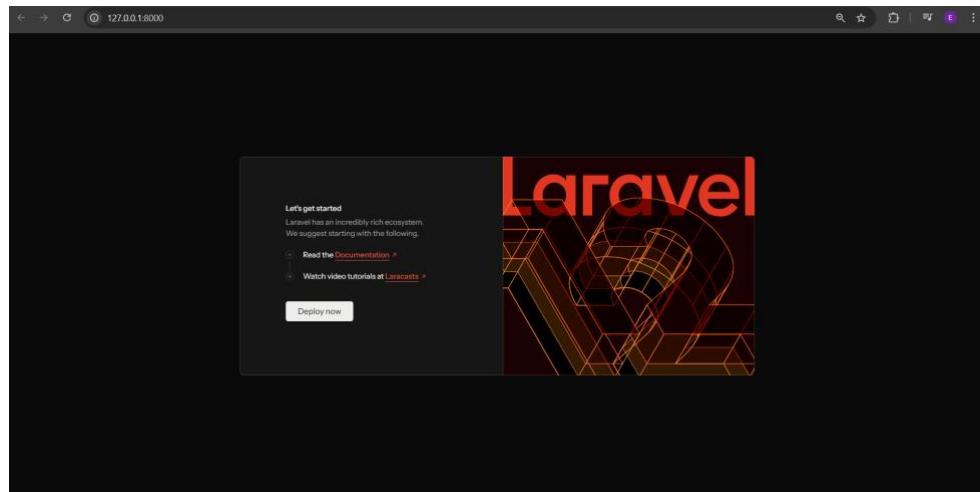
4. Setelah langkah diatas berhasil, Pada terminal ketik perintah `cd nama-project-kamu` sebagai berikut

```
0001_01_01_000001_create_books_table .....  
0001_01_01_000002_create_jobs_table .....  
PS D:\Perpustakaan> cd nama-project-kamu
```

5. Lalu ketik `php artisan serve`

```
PS D:\Perpustakaan> cd PerpustakaanDigital  
PS D:\Perpustakaan\PerpustakaanDigital> php artisan serve
```

6. Kemudian buka url dari hasil serve diatas melalui browser

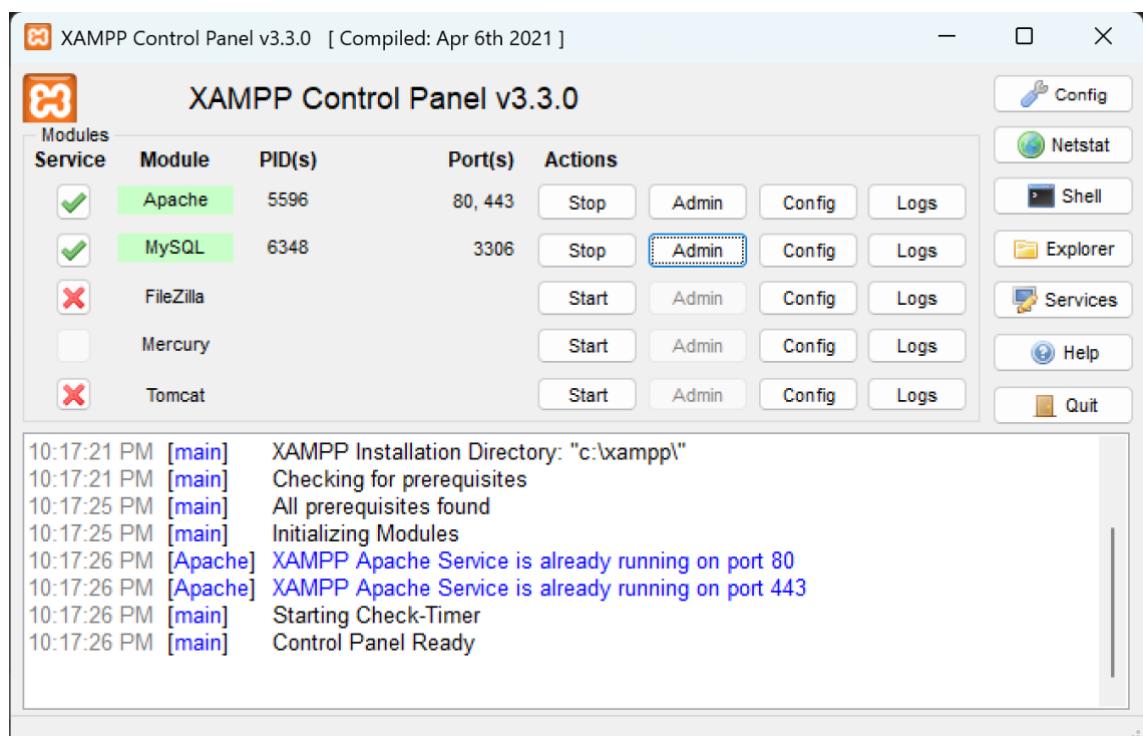


7. Setelah berhasil menampilkan logo laravel di browser, Pada terminal ketik **code .** untuk langsung membuka vscode

D. Konfigurasi database

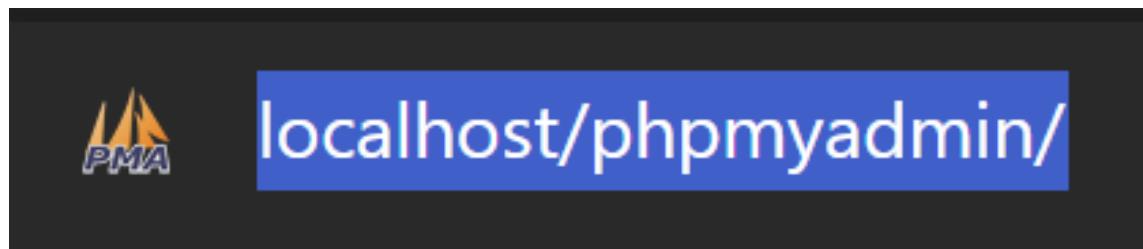
Setelah proyek berhasil dijalankan, langkah selanjutnya adalah menghubungkan ke database. Konfigurasi dilakukan pada file .env dengan mengatur nama database, username, dan password MySQL. Pengaturan ini bertujuan agar Laravel dapat terhubung dan menyimpan data ke database yang telah dibuat di phpMyAdmin. **Langkah ini wajib dilakukan sebelum migration, jika tidak maka proses migrate akan gagal.**

1. Buka phpMyAdmin menggunakan XAMPP (pilih admin MySQL) atau ketik localhost/phpmyadmin/ di browser



The screenshot shows the XAMPP Control Panel v3.3.0 interface. It lists several services: Apache (running, PID 5596), MySQL (running, PID 6348), FileZilla (stopped), Mercury (stopped), and Tomcat (stopped). The Apache and MySQL services are highlighted in green. The 'Actions' column includes buttons for Stop, Admin, Config, and Logs. On the right side, there are links for Config, Netstat, Shell, Explorer, Services, Help, and Quit. Below the service list is a log window displaying the following output:

```
10:17:21 PM [main] XAMPP Installation Directory: "c:\xampp\"  
10:17:21 PM [main] Checking for prerequisites  
10:17:25 PM [main] All prerequisites found  
10:17:25 PM [main] Initializing Modules  
10:17:26 PM [Apache] XAMPP Apache Service is already running on port 80  
10:17:26 PM [Apache] XAMPP Apache Service is already running on port 443  
10:17:26 PM [main] Starting Check-Timer  
10:17:26 PM [main] Control Panel Ready
```

The screenshot shows a web browser window with the URL localhost/phpmyadmin/ in the address bar. The page features the phpMyAdmin logo (a stylized 'PMA' with orange and yellow triangles) and the text "localhost/phpmyadmin/" in white on a blue background.

2. Buat database nya saja di phpMyAdmin, tidak perlu buat table nya



3. Buka file .env, sesuaikan bagian berikut. Pastikan nama database nya sama seperti yang telah kalian buat seperti di phpMyAdmin

```
DB_CONNECTION=mysql
DB_HOST=localhost
DB_PORT=3306
DB_DATABASE=nama-database-kamu
DB_USERNAME=root
DB_PASSWORD=
```

KONFIGURASI MODEL & MIGRATION

Migration

Migration adalah fitur Laravel yang berfungsi untuk mengatur struktur tabel dalam *database*. Dengan menggunakan migration, perubahan struktur *database* dapat dilakukan dengan mudah tanpa harus melakukannya *database* secara manual.

Model

Model adalah representasi dari tabel dari *database*. Model berfungsi untuk mengolah data dari *database* dengan lebih mudah tanpa harus menggunakan query manual.

Pada implementasi model mencakup pembuatan tabel utama yaitu:

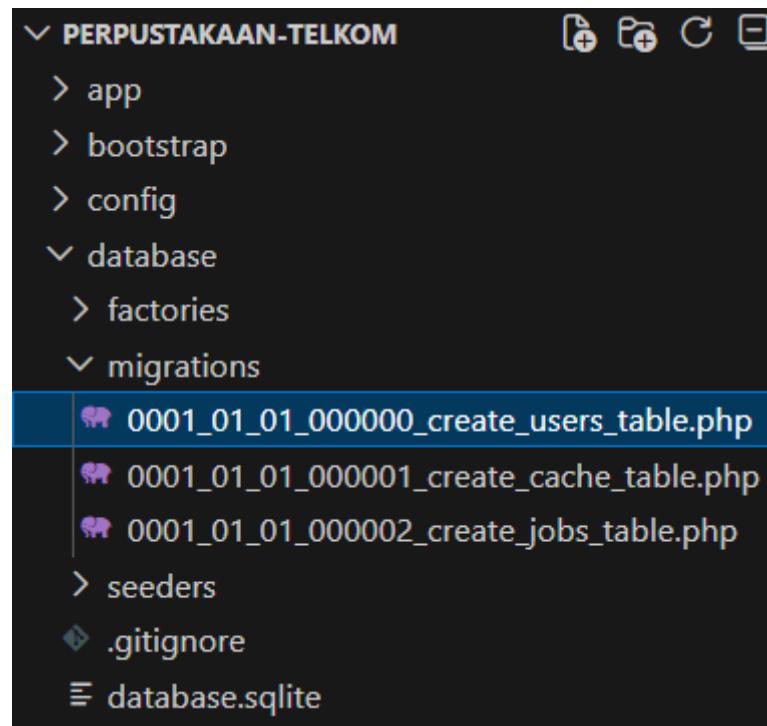
- User: Mengatur data pengguna dan autentikasi.
- Buku: Mengelola katalog buku dengan perintah "`php artisan make:model Book -mc`".

catatan: `-mc` digunakan untuk mempercepat proses pengembangan dan menciptakan file migration dan controller secara otomatis dalam satu perintah

- Transaksi: Mencatat riwayat peminjaman atau aktivitas terkait buku.
- Seeder: Setelah struktur tabel siap, dilakukan pengisian data awal (*seeding*) dan eksekusi migrasi untuk menyinkronkan struktur ke dalam database.

1.1 Migration & Model Tabel User

- a. Membuka file migrasi tabel user bawaan Laravel yang terdapat pada direktori "database/migration" dengan kata kunci "create_users_table.php".



- b. Menyesuaikan struktur kolom tabel user sesuai gambar berikut:

```
perpustakaan-telkom - 0001_01_01_000000_create_users_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('users', function (Blueprint $table) {
15             $table->id();
16             $table->integer('school_id')->unsigned();
17             $table->string('name');
18             $table->string('class');
19             $table->string('major');
20             $table->string('username')->unique();
21             $table->enum('role', ['admin', 'member'])->default('member');
22             $table->string('password');
23             $table->string('email')->unique();
24             $table->timestamp('email_verified_at')->nullable();
25             $table->rememberToken();
26             $table->timestamps();
27         });
28
29         Schema::create('password_reset_tokens', function (Blueprint $table) {
30             $table->string('email')->primary();
31             $table->string('token');
32             $table->timestamp('created_at')->nullable();
33         });
34
35         Schema::create('sessions', function (Blueprint $table) {
36             $table->string('id')->primary();
37             $table->foreignId('user_id')->nullable()->index();
38             $table->string('ip_address', 45)->nullable();
39             $table->text('user_agent')->nullable();
40             $table->longText('payload');
41             $table->integer('last_activity')->index();
42         });
43     }
44
45     /**
46      * Reverse the migrations.
47      */
48     public function down(): void
49     {
50         Schema::dropIfExists('users');
51         Schema::dropIfExists('password_reset_tokens');
52         Schema::dropIfExists('sessions');
53     }
54 };
55 }
```

- c. Membuka file migrasi pada direktori “`app/Models`” dengan kata kunci “`User.php`”.
- d. Menyesuaikan model user sesuai gambar berikut:



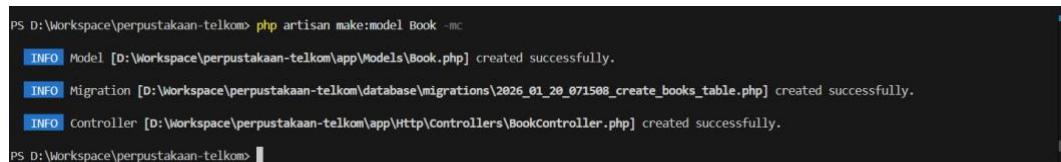
```
perpustakaan-telkom - User.php

1 <?php
2
3 namespace App\Models;
4
5 // use Illuminate\Contracts\Auth\MustVerifyEmail;
6 use Illuminate\Database\Eloquent\Factories\HasFactory;
7 use Illuminate\Foundation\Auth\User as Authenticatable;
8 use Illuminate\Notifications\Notifiable;
9
10 class User extends Authenticatable
11 {
12     /** @use HasFactory<\Database\Factories\UserFactory> */
13     use HasFactory, Notifiable;
14
15     /**
16      * The attributes that are mass assignable.
17      *
18      * @var list<string>
19      */
20     protected $fillable = [
21         'school_id',
22         'major',
23         'class',
24         'username',
25         'name',
26         'password',
27         'email',
28     ];
29
30     /**
31      * The attributes that should be hidden for serialization.
32      *
33      * @var list<string>
34      */
35     protected $hidden = [
36         'password',
37         'remember_token',
38     ];
39
```

```
40     /**
41      * Get the attributes that should be cast.
42      *
43      * @return array<string, string>
44      */
45     protected function casts(): array
46     {
47         return [
48             'email_verified_at' => 'datetime',
49             'password' => 'hashed',
50         ];
51     }
52
53     public function transactions()
54     {
55         return $this->hasMany(Transaction::class);
56     }
57 }
58
```

1.3 Migration & Model Tabel Buku

- Membuka terminal di visual studio code dan jalankan perintah
`"php artisan make:model Book -mc"` sesuai gambar berikut:



```
PS D:\Workspace\perpustakaan-telkom> php artisan make:model Book -mc
[INFO] Model [D:\Workspace\perpustakaan-telkom\app\Models\Book.php] created successfully.
[INFO] Migration [D:\Workspace\perpustakaan-telkom\database\migrations\2026_01_20_071508_create_books_table.php] created successfully.
[INFO] Controller [D:\Workspace\perpustakaan-telkom\app\Http\Controllers\BookController.php] created successfully.
```

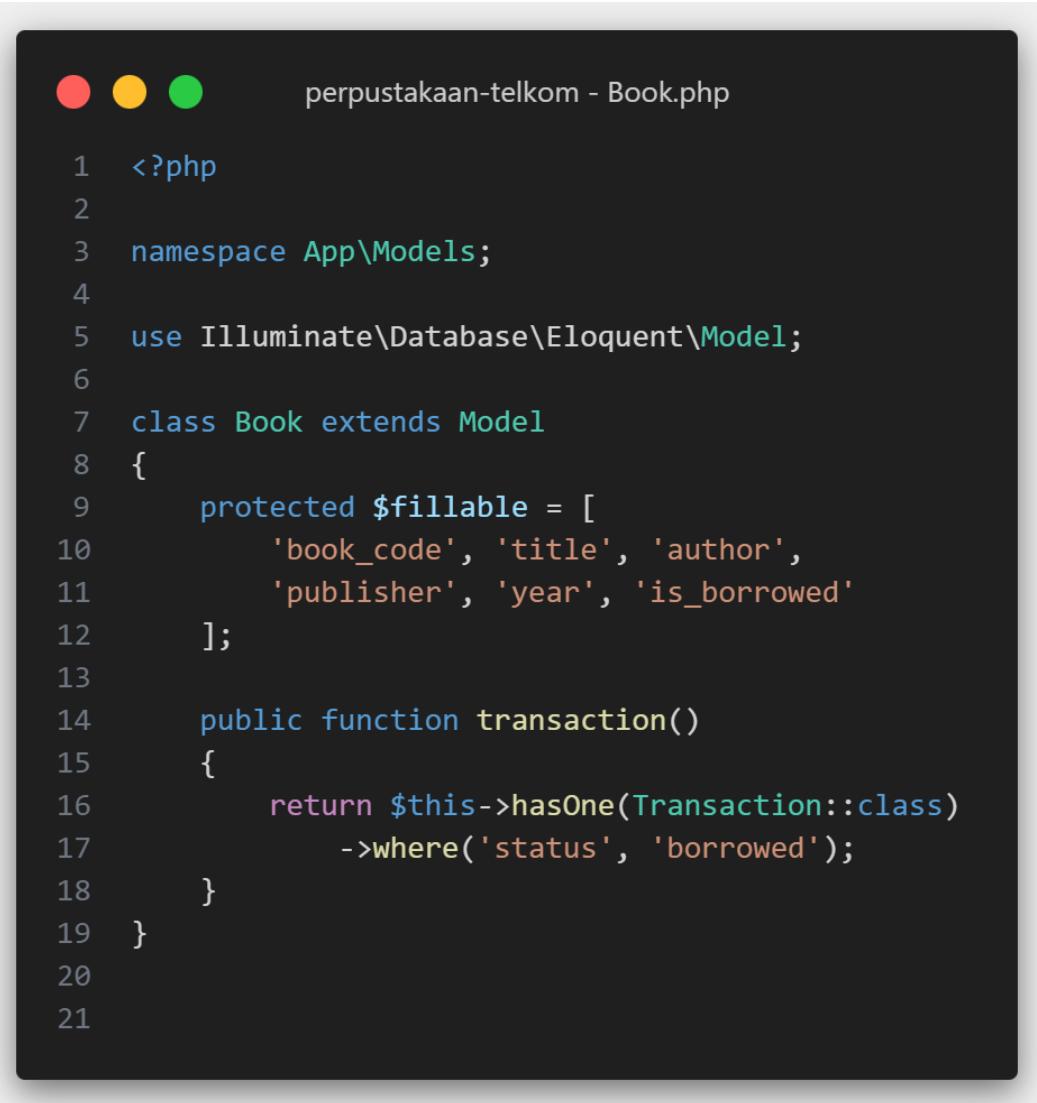
- Membuka file migrasi pada direktori "`database/migration`" dengan kata kunci "`create_books_table.php`".
- Menyesuaikan struktur kolom tabel book sesuai gambar berikut:

```
● ● ● perpustakaan-telkom - 2026_01_19_235842_create_books_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('books', function (Blueprint $table) {
15             $table->id();
16             $table->string('book_code')->unique();
17             $table->string('title');
18             $table->string('author');
19             $table->string('publisher');
20             $table->year('year');
21             $table->boolean('is_borrowed')->default(false);
22             $table->timestamps();
23         });
24     }
25
26     /**
27      * Reverse the migrations.
28      */
29     public function down(): void
30     {
31         Schema::dropIfExists('books');
32     }
33 };
34
```

- d. Membuka file migrasi pada direktori "[app/Models](#)" dengan kata kunci "[Book.php](#)".

- e. Menyesuaikan model book sesuai gambar berikut:



```

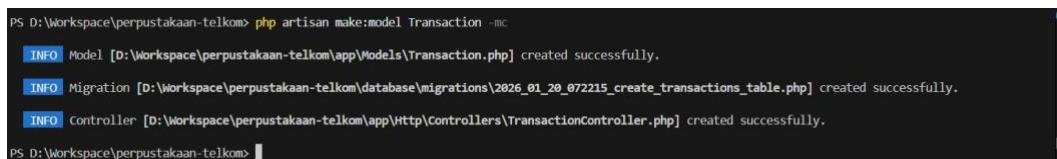
perpustakaan-telkom - Book.php

1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Book extends Model
8 {
9     protected $fillable = [
10         'book_code', 'title', 'author',
11         'publisher', 'year', 'is_borrowed'
12     ];
13
14     public function transaction()
15     {
16         return $this->hasOne(Transaction::class)
17             ->where('status', 'borrowed');
18     }
19 }
20
21

```

1.4 Migration & Model Tabel Transaksi

- a. Membuka terminal di visual studio code dan jalankan perintah "`php artisan make:model Transaction -mc`" sesuai gambar berikut:



```

PS D:\Workspace\perpustakaan-telkom> php artisan make:model Transaction -mc
[INFO] Model [D:\Workspace\perpustakaan-telkom\app\Models\Transaction.php] created successfully.
[INFO] Migration [D:\Workspace\perpustakaan-telkom\database\migrations\2026_01_20_072215_create_transactions_table.php] created successfully.
[INFO] Controller [D:\Workspace\perpustakaan-telkom\app\Http\Controllers\TransactionController.php] created successfully.

```

- b. Membuka file migrasi pada direktori "`database/migration`" dengan kata kunci "`create_transactions_table.php`".

- c. Menyesuaikan struktur kolom tabel transaksi sesuai gambar berikut:

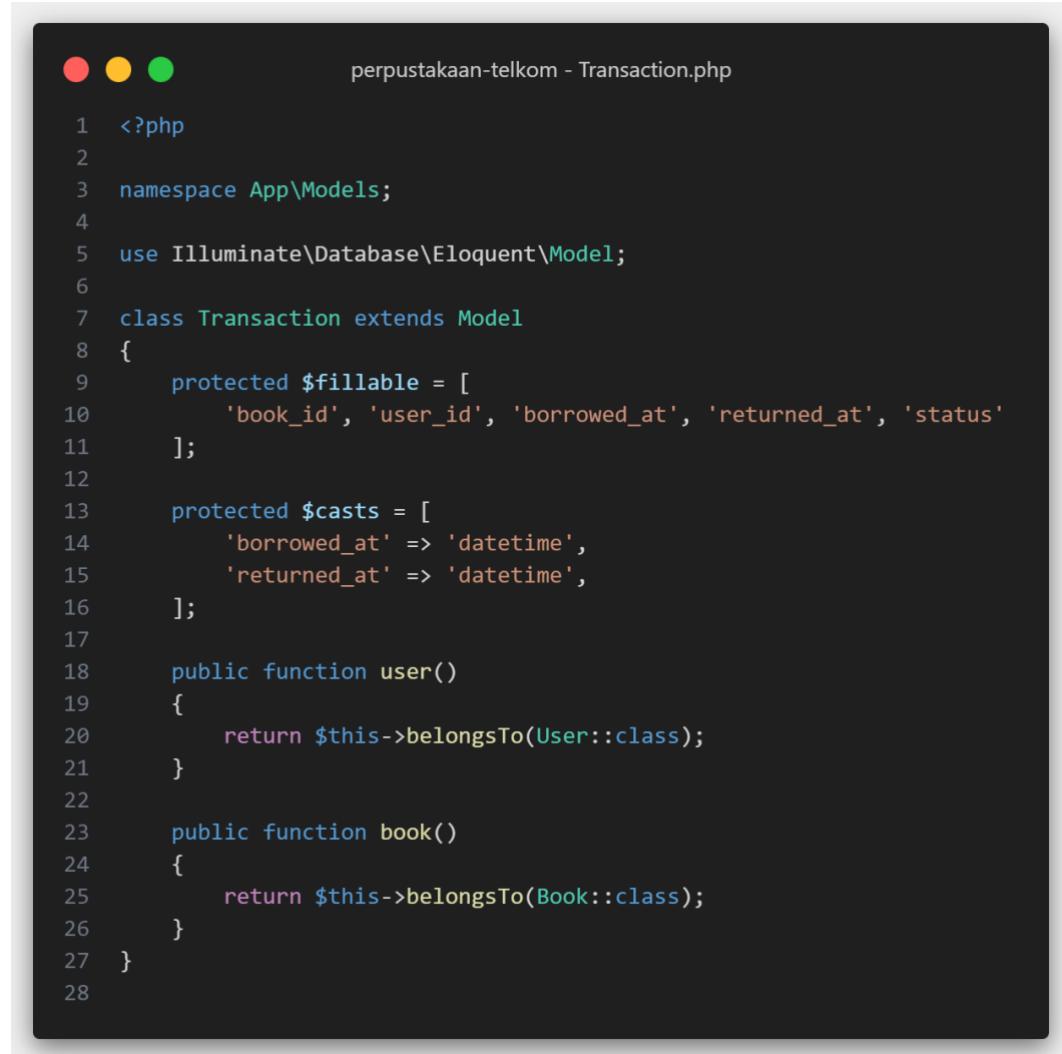


```
perpustakaan-telkom - 2026_01_20_072215_create_transactions_table.php

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('transactions', function (Blueprint $table) {
15             $table->id();
16             $table->foreignId('book_id')->constrained()->cascadeOnDelete();
17             $table->foreignId('user_id')
18                 ->constrained()
19                 ->cascadeOnDelete();
20             $table->date('borrowed_at');
21             $table->date('returned_at')->nullable();
22             $table->enum('status', ['borrowed', 'returned']);
23             $table->timestamps();
24         });
25     }
26
27     /**
28      * Reverse the migrations.
29      */
30     public function down(): void
31     {
32         Schema::dropIfExists('transactions');
33     }
34 };
35
```

- d. Membuka file migrasi pada direktori "**app/Models**" dengan kata kunci "**Transaction.php**".

- e. Menyesuaikan model transaction sesuai gambar berikut:



```
perpustakaan-telkom - Transaction.php

1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Transaction extends Model
8 {
9     protected $fillable = [
10         'book_id', 'user_id', 'borrowed_at', 'returned_at', 'status'
11     ];
12
13     protected $casts = [
14         'borrowed_at' => 'datetime',
15         'returned_at' => 'datetime',
16     ];
17
18     public function user()
19     {
20         return $this->belongsTo(User::class);
21     }
22
23     public function book()
24     {
25         return $this->belongsTo(Book::class);
26     }
27 }
28
```

1.5 Membuat Seeder dan Melakukan Laravel Migration

- a. Membuka file migrasi pada direktori "database/seeds" dengan kata kunci "**DatabaseSeeder.php**".
- b. Menyesuaikan seeder sebagai berikut:



```
perpustakaan-telkom - DatabaseSeeder.php

1 <?php
2
3 namespace Database\Seeders;
4
5 use App\Models\User;
6 use Illuminate\Database\Console\Seeds\WithoutModelEvents;
7 use Illuminate\Database\Seeder;
8 use Illuminate\Support\Facades\Hash;
9
10 use function Symfony\Component\Clock\now;
11
12 class DatabaseSeeder extends Seeder
13 {
14     use WithoutModelEvents;
15     public function run(): void
16     {
17         User::create(
18             [
19                 'id' => 0,
20                 'school_id' => 0,
21                 'name' => 'ADMIN PERPUSTAKAAN',
22                 'class' => '',
23                 'major' => '',
24                 'username' => 'admin_perpustakaan',
25                 'role' => 'admin',
26                 'password' => Hash::make('admin123'),
27                 'email' => 'perpustakaan@smktelkom-jkt.sch.id',
28                 'created_at' => now(),
29                 'updated_at' => now(),
30             ],
31         );
32     }
33 }
34 }
```

- c. Melakukan migration dengan membuka terminal di visual studio code dan jalankan perintah sesuai gambar berikut:

```
PS D:\Workspace\perpustakaan-telkom> php artisan migrate:fresh --seed
Dropping all tables ..... 6.19ms DONE
INFO Preparing database.
Creating migration table ..... 5.51ms DONE
INFO Running migrations.
0001_01_01_00000_create_users_table ..... 25.23ms DONE
0001_01_01_00001_create_cache_table ..... 4.67ms DONE
0001_01_01_00002_create_jobs_table ..... 10.67ms DONE
2026_01_20_071508_create_books_table ..... 4.69ms DONE
2026_01_20_072215_create_transactions_table ..... 7.09ms DONE
INFO Seeding database.
PS D:\Workspace\perpustakaan-telkom>
```

KONFIGURASI VIEW

Pengembangan Antarmuka (View) pada Laravel memanfaatkan *Blade Templating Engine*, yang merupakan fitur bawaan Laravel dan memungkinkan pemisahan antara logika tampilan dan logika aplikasi secara bersih. Langkah awal dalam konfigurasi view adalah menentukan struktur folder di dalam direktori "`resources\views`" agar pengelolaan file template menjadi lebih terorganisir.

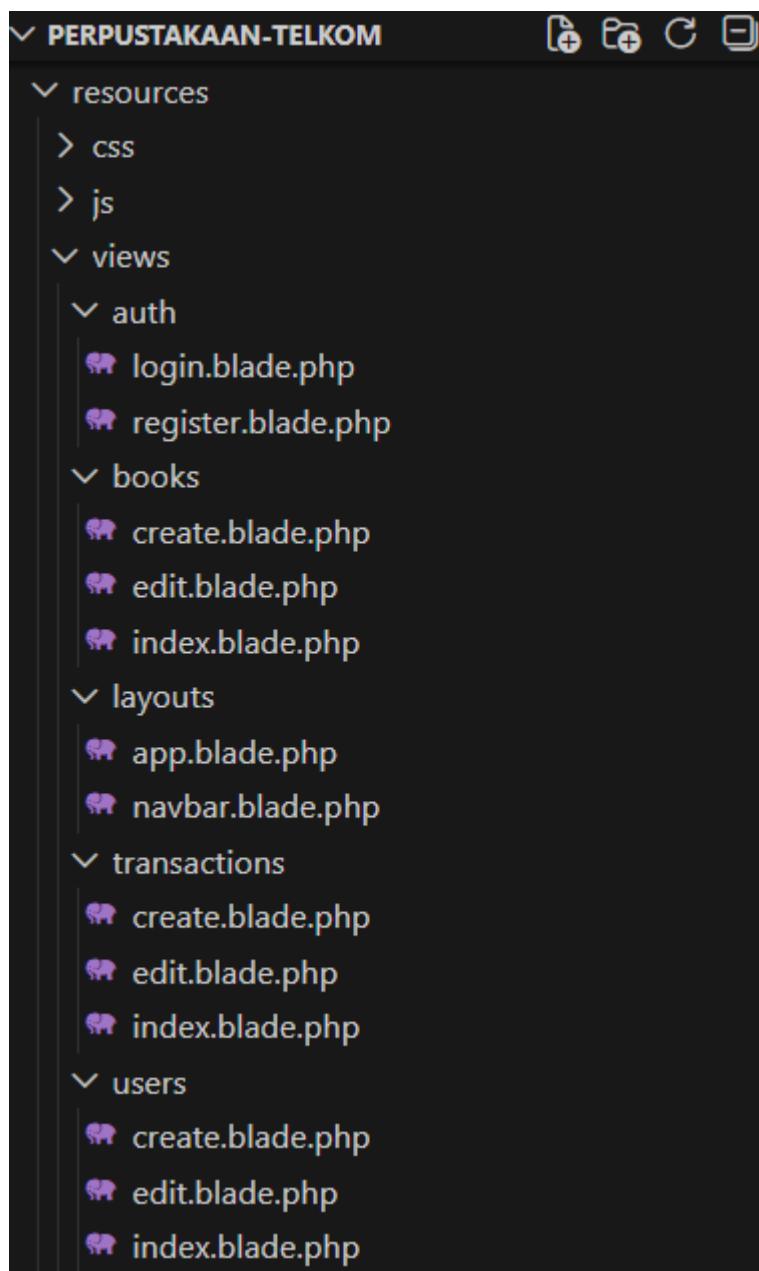
Fokus utamanya adalah menciptakan pengalaman pengguna yang konsisten melalui:

- Layout utama: Pembuatan file "`app.blade.php`" dan "`navbar.blade.php`" sebagai kerangka dasar.
- Halaman Autentikasi: Menyediakan formulir login dan registrasi.
- Manajemen Data: Membuat halaman index, tambah (*create*) dan ubah (*edit*) untuk masing-masing model seperti Buku, Transaksi dan User.

Berbeda dengan Model atau Controller yang dapat dibuat melalui Command Line Interface (CLI) Artisan, pembuatan file pada direktori ini dilakukan secara manual.

2.1 Membuat Kerangka View

- a. Membuka file view user bawaan Laravel yang terdapat pada direktori "resources\views".
- b. Membuat file view dan sesuaikan dengan cara create file untuk masing masing fitur sesuai gambar berikut:



- c. Membuka file view yang telah dibuat sebelumnya di pada direktori "resources\views\layouts".

- d. Sesuaikan file view "app.blade.php" sesuai pada gambar berikut:

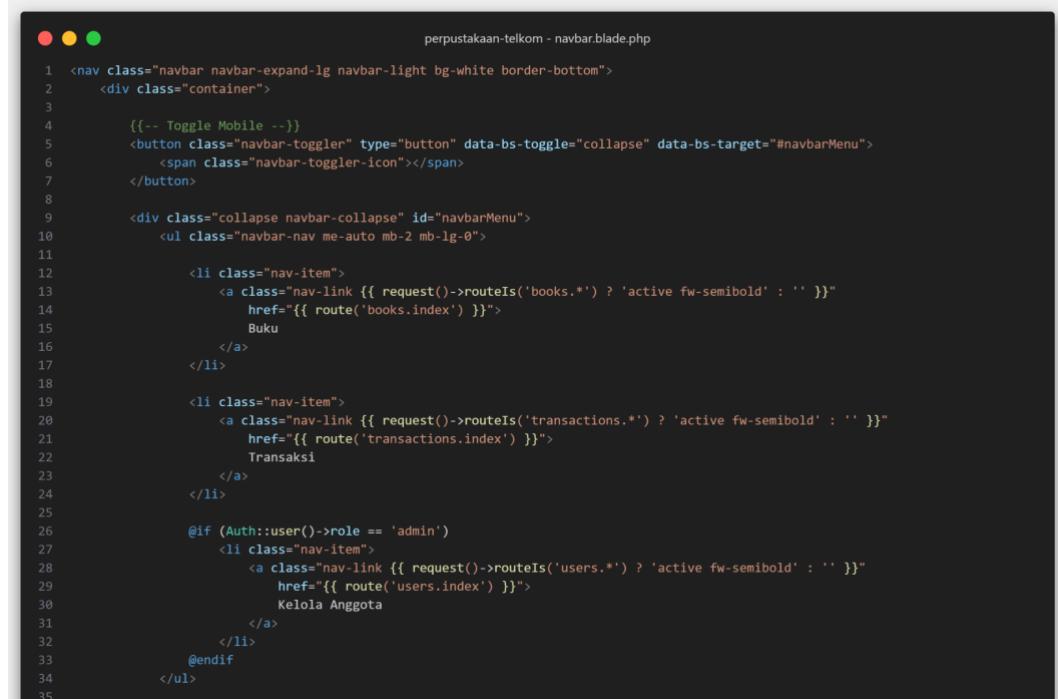


```

1 <!doctype html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <title>{{ config('app.name') }}</title>
8   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/css/bootstrap.min.css" rel="stylesheet"
9        integrity="sha384-sRl4kxILFvY47J16cr9ZwB07vP4J8+LH7qKQnuqkuIAvNWLeN8tEYBujZqJLB" crossorigin="anonymous">
10 </head>
11
12 <body class="bg-gray-50">
13   @include('layouts.navbar')
14   <main class="container">
15     @yield('content')
16   </main>
17   <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js"
18         integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/Zjvygo16FvRkX/vR+vC4jQkC+hVqc2pM80Dewa9r" crossorigin="anonymous">
19   </script>
20   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/js/bootstrap.min.js"
21         integrity="sha384-sh38-G/Ev+4j2dNv+tEPo3++6LCgdCROaejBqfUeNjuKAiuXbjrxilcCdZ6ZAVfHWe1Y" crossorigin="anonymous">
22   </script>
23 </body>
24
25 </html>

```

- e. Sesuaikan file view "navbar.blade.php" sesuai pada gambar berikut:



```

1 <nav class="navbar navbar-expand-lg navbar-light bg-white border-bottom">
2   <div class="container">
3
4     {{-- Toggle Mobile --}}
5     <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarMenu">
6       <span class="navbar-toggler-icon"></span>
7     </button>
8
9     <div class="collapse navbar-collapse" id="navbarMenu">
10       <ul class="navbar-nav me-auto mb-2 mb-lg-0">
11
12         <li class="nav-item">
13           <a class="nav-link" {{ request()->routeIs('books.*') ? 'active fw-semibold' : '' }}>
14             Buku
15           </a>
16         </li>
17
18         <li class="nav-item">
19           <a class="nav-link" {{ request()->routeIs('transactions.*') ? 'active fw-semibold' : '' }}>
20             href="{{ route('transactions.index') }}"
21             Transaksi
22           </a>
23         </li>
24
25       @if (Auth::user()->role == 'admin')
26         <li class="nav-item">
27           <a class="nav-link" {{ request()->routeIs('users.*') ? 'active fw-semibold' : '' }}>
28             href="{{ route('users.index') }}"
29             Kelola Anggota
30           </a>
31         </li>
32       @endif
33     </ul>
34   </div>
35 
```

```

36     {{-- User Info --}}
37     <div class="d-flex align-items-center gap-3">
38
39         <div class="d-none d-lg-block">
40             <div class="fw-semibold">{{ Auth::user()->name }}</div>
41             <div class="text-muted small">
42                 {{ Auth::user()->role == 'admin' ? 'Administrator' : Auth::user()->major , ' - ', Auth::user()->class }}
43             </div>
44         </div>
45
46     {{-- Logout --}}
47     <form action="{{ route('logout') }}" method="POST">
48         @csrf
49         <button type="submit" class="btn btn-dark btn-sm">
50             Logout
51         </button>
52     </form>
53     </div>
54 </div>
55 </div>
56 </nav>
57

```

2.2 Tampilan Halaman Login dan Register

- a. Membuka file view yang telah dibuat sebelumnya di pada direktori "`resources\views\auth`".
- b. Sesuaikan file view "`login.blade.php`" sesuai pada gambar berikut:

perpustakaan-telkom - login.blade.php

```

1  <!doctype html>
2  <html lang="en">
3
4  <head>
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <title>Login - {{ config('app.name') }}</title>
8      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/css/bootstrap.min.css" rel="stylesheet">
9  </head>
10 <body class="bg-light d-flex align-items-center justify-content-center vh-100">
11
12     <div class="card shadow-sm" style="width: 100%; max-width: 400px;">
13         <div class="card-body p-4">
14
15             <h3 class="text-center mb-4">Login</h3>
16
17             <form action="/login" method="POST">
18                 @csrf
19
20                     @if ($errors->any())
21                         <div class="alert alert-danger">
22                             {{ $errors->first() }}
23                         </div>
24                     @endif
25
26
27                     @if (session('success'))
28                         <div class="alert alert-success">
29                             {{ session('success') }}
30                         </div>
31                     @endif
32
33

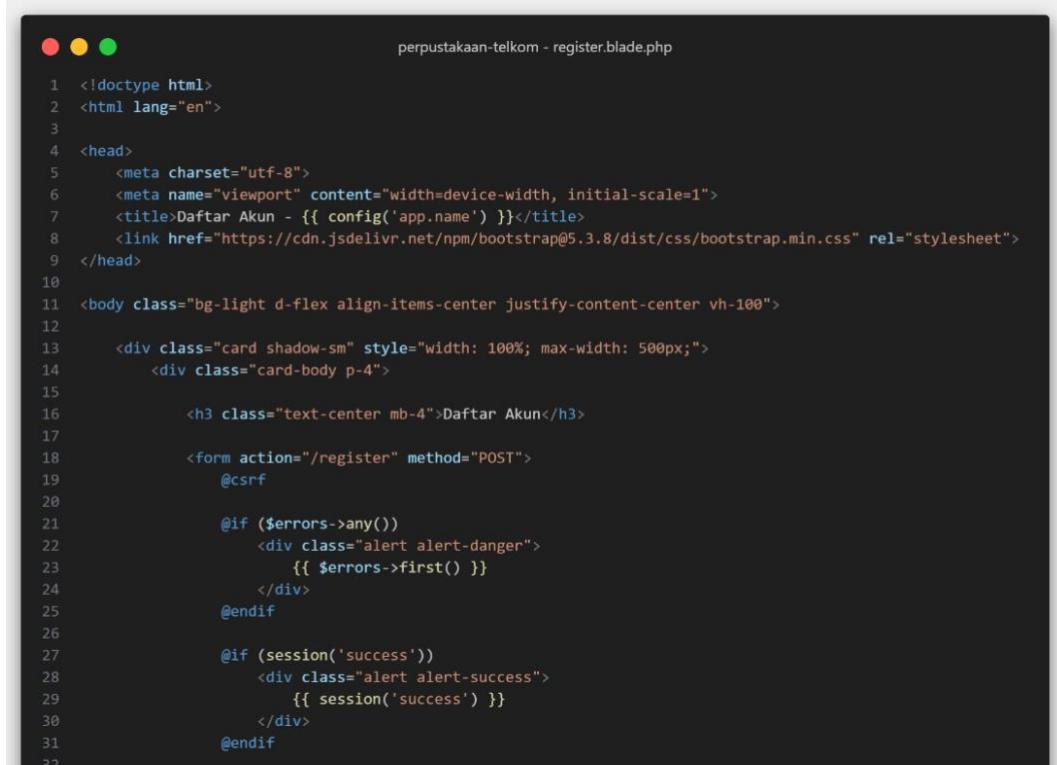
```

```

32         <div class="mb-3">
33             <label class="form-label">Username</label>
34             <input type="text" name="username" class="form-control" required>
35         </div>
36
37         <div class="mb-4">
38             <label class="form-label">Password</label>
39             <input type="password" name="password" class="form-control" required>
40         </div>
41
42         <button type="submit" class="btn btn-dark w-100">
43             Login
44         </button>
45     </form>
46
47     <p class="text-center mt-3 mb-0">
48         Belum punya akun?
49         <a href="register">Daftar</a>
50     </p>
51
52     </div>
53 </div>
54
55 </body>
56
57 </html>
58
59

```

- c. Sesuaikan file view "register.blade.php" sesuai pada gambar berikut:



```

● ○ ●
perpustakaan-telkom - register.blade.php

1  <!doctype html>
2  <html lang="en">
3
4  <head>
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <title>Daftar Akun - {{ config('app.name') }}</title>
8      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/css/bootstrap.min.css" rel="stylesheet">
9  </head>
10
11 <body class="bg-light d-flex align-items-center justify-content-center vh-100">
12
13     <div class="card shadow-sm" style="width: 100%; max-width: 500px;">
14         <div class="card-body p-4">
15
16             <h3 class="text-center mb-4">Daftar Akun</h3>
17
18             <form action="/register" method="POST">
19                 @csrf
20
21                 @if ($errors->any())
22                     <div class="alert alert-danger">
23                         {{ $errors->first() }}
24                     </div>
25                 @endif
26
27                 @if (session('success'))
28                     <div class="alert alert-success">
29                         {{ session('success') }}
30                     </div>
31                 @endif
32
33             </form>
34
35         </div>
36     </div>
37
38     <p class="text-center mt-3 mb-0">
39         Belum punya akun?
40         <a href="register">Daftar</a>
41     </p>
42
43     </div>
44 </div>
45
46 </body>
47
48 </html>
49
50
51
52
53
54
55
56
57
58
59

```

```
32      <div class="row g-3 mb-3">
33          <div class="col-md-6">
34              <label class="form-label">NIS</label>
35              <input type="number" name="school_id" class="form-control" required>
36          </div>
37
38          <div class="col-md-6">
39              <label class="form-label">Nama Lengkap</label>
40              <input type="text" name="name" class="form-control" required>
41          </div>
42
43          <div class="col-md-6">
44              <label class="form-label">Kelas</label>
45              <input type="text" name="class" class="form-control" required>
46          </div>
47
48          <div class="col-md-6">
49              <label class="form-label">Jurusan</label>
50              <input type="text" name="major" class="form-control" required>
51          </div>
52
53      </div>
54
55      <div class="mb-3">
56          <label class="form-label">Username</label>
57          <input type="text" name="username" class="form-control" required>
58      </div>
59
60      <div class="mb-3">
61          <label class="form-label">Password</label>
62          <input type="password" name="password" class="form-control" required>
63      </div>
64
65      <div class="mb-4">
66          <label class="form-label">Konfirmasi Password</label>
67          <input type="password" name="password_confirmation" class="form-control" required>
68      </div>
69
70      <button type="submit" class="btn btn-dark w-100">
71          Daftar
72      </button>
73  </form>
74
75  <p class="text-center mt-3 mb-0">
76      Sudah punya akun?
77      <a href="login">Login</a>
78  </p>
79
80  </div>
81 </div>
82
83 </body>
84
85 </html>
86
```


2.2 Tampilan Halaman Model Book

- a. Membuka file view yang telah dibuat sebelumnya di pada direktori "`resources\views\books`".
- b. Sesuaikan file view "`index.blade.php`" sesuai pada gambar berikut:

perpustakaan-telkom - index.blade.php

```

1  @extends('layouts.app')
2
3  @section('title', 'Dashboard')
4
5  @section('content')
6  <div class="container my-4">
7
8      {{-- Header --}}
9      <div class="card mb-4">
10         <div class="card-body d-flex justify-content-between align-items-center">
11             <div>
12                 <h4 class="mb-1">Dashboard Buku</h4>
13                 <small class="text-muted">Ringkasan data buku di sistem</small>
14             </div>
15
16             @if (Auth::user()->role == 'admin')
17                 <a href="{{ route('books.create') }}" class="btn btn-primary btn-sm">
18                     + Tambah Buku
19                 </a>
20             @endif
21         </div>
22     </div>
23
24     {{-- Table --}}
25     <div class="card">
26         <div class="card-body p-0">
27             <table class="table table-hover mb-0 align-middle">
28                 <thead class="table-light">
29                     <tr>
30                         <th>Judul</th>
31                         <th>Penulis</th>
32                         <th>Penerbit</th>
33                         <th>Tahun</th>
34                         <th class="text-center">Aksi</th>
35                     </tr>
36                 </thead>
37                 <tbody>
38                     @forelse ($books as $book)
39                         <tr>
40                             <td>{{ $book->title }}</td>
41                             <td>{{ $book->author }}</td>
42                             <td>{{ $book->publisher }}</td>
43                             <td>{{ $book->year }}</td>
44                             <td class="text-center">
45
46                             @if (Auth::user()->role == 'admin')
47                                 <a href="{{ route('books.edit', $book->id) }}" class="btn btn-warning btn-sm">
48                                     Edit
49                                 </a>
50
51                                 <form action="{{ route('books.destroy', $book->id) }}" method="POST" class="d-inline">
52                                     @csrf
53                                     @method('DELETE')
54                                 </form>
55                             @endif
56                         </td>
57                     @empty
58                         <tr>
59                             <td colspan="5" style="text-align: center;">Tidak ada data buku.
60                         </tr>
61                     @endempty
62                 </tbody>
63             </table>
64         </div>
65     </div>
66 
```

```
54      onsubmit="return confirm('Yakin ingin menghapus buku ini?')">>
55      @csrf
56      @method('DELETE')
57      <button class="btn btn-danger btn-sm">
58          Hapus
59      </button>
60  </form>
61  @else
62      @if (!$book->is_borrowed)
63          <form method="POST"
64              action="{{ route('transactions.borrow', $book->id) }}">
65              @csrf
66              <button class="btn btn-success btn-sm">
67                  Pinjam
68              </button>
69          </form>
70      @else
71          <span class="badge bg-secondary">
72              Tidak Tersedia
73          </span>
74      @endif
75  @endif
76
77      </td>
78  </tr>
79  @empty
80      <tr>
81          <td colspan="5" class="text-center text-muted py-4">
82              Tidak ada data buku
83          </td>
84      </tr>
85  @endforelse
86      </tbody>
87  </table>
88      </div>
89  </div>
90
91 </div>
92 @endsection
93
```

c. Sesuaikan file view "edit.blade.php" sesuai pada gambar berikut:

```
● ○ ●
perpustakaan-telkom - edit.blade.php

1 @extends('layouts.app')
2
3 @section('title', 'Edit Buku')
4
5 @section('content')
6 <div class="container my-4">
7
8     {{-- Header --}}
9     <div class="card mb-4">
10         <div class="card-body d-flex justify-content-between align-items-center">
11             <div>
12                 <h4 class="mb-1">Edit Buku</h4>
13                 <small class="text-muted">Perbarui data buku di sistem</small>
14             </div>
15
16             <button type="submit" form="form-book" class="btn btn-primary btn-sm">
17                 Update
18             </button>
19         </div>
20     </div>
21
22     {{-- Form --}}
23     <div class="card">
24         <div class="card-body">
25             <form id="form-book"
26                 action="{{ route('books.update', $book->id) }}"
27                 method="POST">
28
29                 @csrf
30                 @method('PUT')
31
32                 {{-- Kode Buku --}}
33                 <div class="mb-3">
34                     <label class="form-label">Kode Buku</label>
35                     <input type="text"
36                         name="book_code"
37                         value="{{ old('book_code', $book->book_code) }}"
38                         class="form-control @error('book_code') is-invalid @enderror">
39
40                     @error('book_code')
41                         <div class="invalid-feedback">{{ $message }}</div>
42                     @enderror
43                 </div>
44
45                 {{-- Judul --}}
46                 <div class="mb-3">
47                     <label class="form-label">Judul Buku</label>
48                     <input type="text"
49                         name="title"
50                         value="{{ old('title', $book->title) }}"
51                         class="form-control @error('title') is-invalid @enderror">
52
53                     @error('title')
54                         <div class="invalid-feedback">{{ $message }}</div>
55                     @enderror
56                 </div>
57
```

```
57
58     {{-- Penulis --}}
59     <div class="mb-3">
60         <label class="form-label">Penulis</label>
61         <input type="text"
62             name="author"
63             value="{{ old('author', $book->author) }}"
64             class="form-control @error('author') is-invalid @enderror">
65
66         @error('author')
67             <div class="invalid-feedback">{{ $message }}</div>
68         @enderror
69     </div>
70
71     {{-- Penerbit --}}
72     <div class="mb-3">
73         <label class="form-label">Penerbit</label>
74         <input type="text"
75             name="publisher"
76             value="{{ old('publisher', $book->publisher) }}"
77             class="form-control @error('publisher') is-invalid @enderror">
78
79         @error('publisher')
80             <div class="invalid-feedback">{{ $message }}</div>
81         @enderror
82     </div>
83
84     {{-- Tahun --}}
85     <div class="mb-3">
86         <label class="form-label">Tahun Terbit</label>
87         <input type="number"
88             name="year"
89             value="{{ old('year', $book->year) }}"
90             class="form-control @error('year') is-invalid @enderror">
91
92         @error('year')
93             <div class="invalid-feedback">{{ $message }}</div>
94         @enderror
95     </div>
96
97     </form>
98 </div>
99 </div>
100 </div>
101 </div>
102 @endsection
103
```

- d. Sesuaikan file view "create.blade.php" sesuai pada gambar berikut:



```
perpustakaan-telkom - create.blade.php

1 @extends('layouts.app')
2
3 @section('title', 'Tambah Buku')
4
5 @section('content')
6 <div class="container my-4">
7
8     {{-- Header --}}
9     <div class="card mb-4">
10         <div class="card-body d-flex justify-content-between align-items-center">
11             <div>
12                 <h4 class="mb-1">Tambah Buku</h4>
13                 <small class="text-muted">Masukkan data buku baru ke sistem</small>
14             </div>
15
16             <button type="submit" form="form-book" class="btn btn-primary btn-sm">
17                 Simpan
18             </button>
19         </div>
20     </div>
21
22     {{-- Form --}}
23     <div class="card">
24         <div class="card-body">
25             <form id="form-book" action="{{ route('books.store') }}" method="POST">
26                 @csrf
27
28                 {{-- Kode Buku --}}
29                 <div class="mb-3">
30                     <label class="form-label">Kode Buku</label>
31                     <input type="text"
32                         name="book_code"
33                         value="{{ old('book_code') }}"
34                         class="form-control @error('book_code') is-invalid @enderror"
35                         placeholder="AA456">
36                     @error('book_code')
37                         <div class="invalid-feedback">{{ $message }}</div>
38                     @enderror
39                 </div>
40
41                 {{-- Judul --}}
42                 <div class="mb-3">
43                     <label class="form-label">Judul Buku</label>
44                     <input type="text"
45                         name="title"
46                         value="{{ old('title') }}"
47                         class="form-control @error('title') is-invalid @enderror"
48                         placeholder="Contoh: Laravel Untuk Pemula">
49                     @error('title')
50                         <div class="invalid-feedback">{{ $message }}</div>
51                     @enderror
52                 </div>
53             </form>
54         </div>
55     </div>
56 
```

```
73
54     {{-- Penulis --}}
55     <div class="mb-3">
56         <label class="form-label">Penulis</label>
57         <input type="text"
58             name="author"
59             value="{{ old('author') }}"
60             class="form-control @error('author') is-invalid @enderror"
61             placeholder="Nama penulis">
62             @error('author')
63                 <div class="invalid-feedback">{{ $message }}</div>
64             @enderror
65         </div>
66
67     {{-- Penerbit --}}
68     <div class="mb-3">
69         <label class="form-label">Penerbit</label>
70         <input type="text"
71             name="publisher"
72             value="{{ old('publisher') }}"
73             class="form-control @error('publisher') is-invalid @enderror"
74             placeholder="Nama penerbit">
75             @error('publisher')
76                 <div class="invalid-feedback">{{ $message }}</div>
77             @enderror
78         </div>
79
80     {{-- Tahun --}}
81     <div class="mb-3">
82         <label class="form-label">Tahun Terbit</label>
83         <input type="number"
84             name="year"
85             value="{{ old('year') }}"
86             class="form-control @error('year') is-invalid @enderror"
87             placeholder="2024">
88             @error('year')
89                 <div class="invalid-feedback">{{ $message }}</div>
90             @enderror
91         </div>
92
93     </form>
94     </div>
95 </div>
96
97 </div>
98 @endsection
99
```




2.3 Tampilan Halaman Model Transaction

- a. Membuka file view yang telah dibuat sebelumnya di pada direktori "resources\views\transactions".
 - b. Sesuaikan file view "index.blade.php" sesuai pada gambar berikut:

```
61          {{-- Status --}}
62          <td class="text-center">
63              @if ($transaction->status === 'borrowed')
64                  <span class="badge bg-warning text-dark">
65                      Dipinjam
66                  </span>
67              @else
68                  <span class="badge bg-success">
69                      Dikembalikan
70                  </span>
71              @endif
72          </td>
73
74          {{-- Aksi --}}
75          <td class="text-center">
76              @if ($transaction->status === 'borrowed')
77                  <form action="{{ route('transactions.return', $transaction->id) }}">
78                      method="POST" class="d-inline">
79                          @csrf
80                          <button class="btn btn-sm btn-primary">
81                              Kembalikan
82                          </button>
83                      </form>
84                  @endif
85          </td>
86      </tr>
87      @empty
88      <tr>
89          <td colspan="7" class="text-center text-muted py-4">
90              Tidak ada data transaksi
91          </td>
92      </tr>
93      @endforelse
94      </tbody>
95      </table>
96      </div>
97  </div>
98
99
100 </div>
101 @endsection
102
```

c. Sesuaikan file view "edit.blade.php" sesuai pada gambar berikut:



```
1 @extends('layouts.app')
2
3 @section('title', 'Edit Buku')
4
5 @section('content')
6 <div class="container my-4">
7
8     {{-- Header --}}
9     <div class="card mb-4">
10         <div class="card-body d-flex justify-content-between align-items-center">
11             <div>
12                 <h4 class="mb-1">Edit Buku</h4>
13                 <small class="text-muted">Perbarui data buku di sistem</small>
14             </div>
15
16             <button type="submit" form="form-book" class="btn btn-primary btn-sm">
17                 Update
18             </button>
19         </div>
20     </div>
21
22     {{-- Form --}}
23     <div class="card">
24         <div class="card-body">
25             <form id="form-book"
26                 action="{{ route('books.update', $book->id) }}"
27                 method="POST">
28                 @csrf
29                 @method('PUT')
30
31                 {{-- Kode Buku --}}
32                 <div class="mb-3">
33                     <label class="form-label">Kode Buku</label>
34                     <input type="text"
35                         name="book_code"
36                         value="{{ old('book_code', $book->book_code) }}"
37                         class="form-control @error('book_code') is-invalid @enderror"
38                         placeholder="AA456">
39                     @error('book_code')
40                         <div class="invalid-feedback">{{ $message }}</div>
41                     @enderror
42                 </div>
43
44                 {{-- Judul --}}
45                 <div class="mb-3">
46                     <label class="form-label">Judul Buku</label>
47                     <input type="text"
48                         name="title"
49                         value="{{ old('title', $book->title) }}"
50                         class="form-control @error('title') is-invalid @enderror"
51                         @error('title')
52                             <div class="invalid-feedback">{{ $message }}</div>
53                         @enderror
54                 </div>
55             </form>
```

```
56      {{-- Penulis --}}
57      <div class="mb-3">
58          <label class="form-label">Penulis</label>
59          <input type="text"
60              name="author"
61              value="{{ old('author', $book->author) }}"
62              class="form-control @error('author') is-invalid @enderror">
63          @error('author')
64              <div class="invalid-feedback">{{ $message }}</div>
65          @enderror
66      </div>
67
68      {{-- Penerbit --}}
69      <div class="mb-3">
70          <label class="form-label">Penerbit</label>
71          <input type="text"
72              name="publisher"
73              value="{{ old('publisher', $book->publisher) }}"
74              class="form-control @error('publisher') is-invalid @enderror">
75          @error('publisher')
76              <div class="invalid-feedback">{{ $message }}</div>
77          @enderror
78      </div>
79
80      {{-- Tahun --}}
81      <div class="mb-3">
82          <label class="form-label">Tahun Terbit</label>
83          <input type="number"
84              name="year"
85              value="{{ old('year', $book->year) }}"
86              class="form-control @error('year') is-invalid @enderror">
87          @error('year')
88              <div class="invalid-feedback">{{ $message }}</div>
89          @enderror
90      </div>
91
92      </form>
93  </div>
94 </div>
95
96 </div>
97 @endsection
98
```

d. Sesuaikan file view "create.blade.php" sesuai pada gambar berikut:



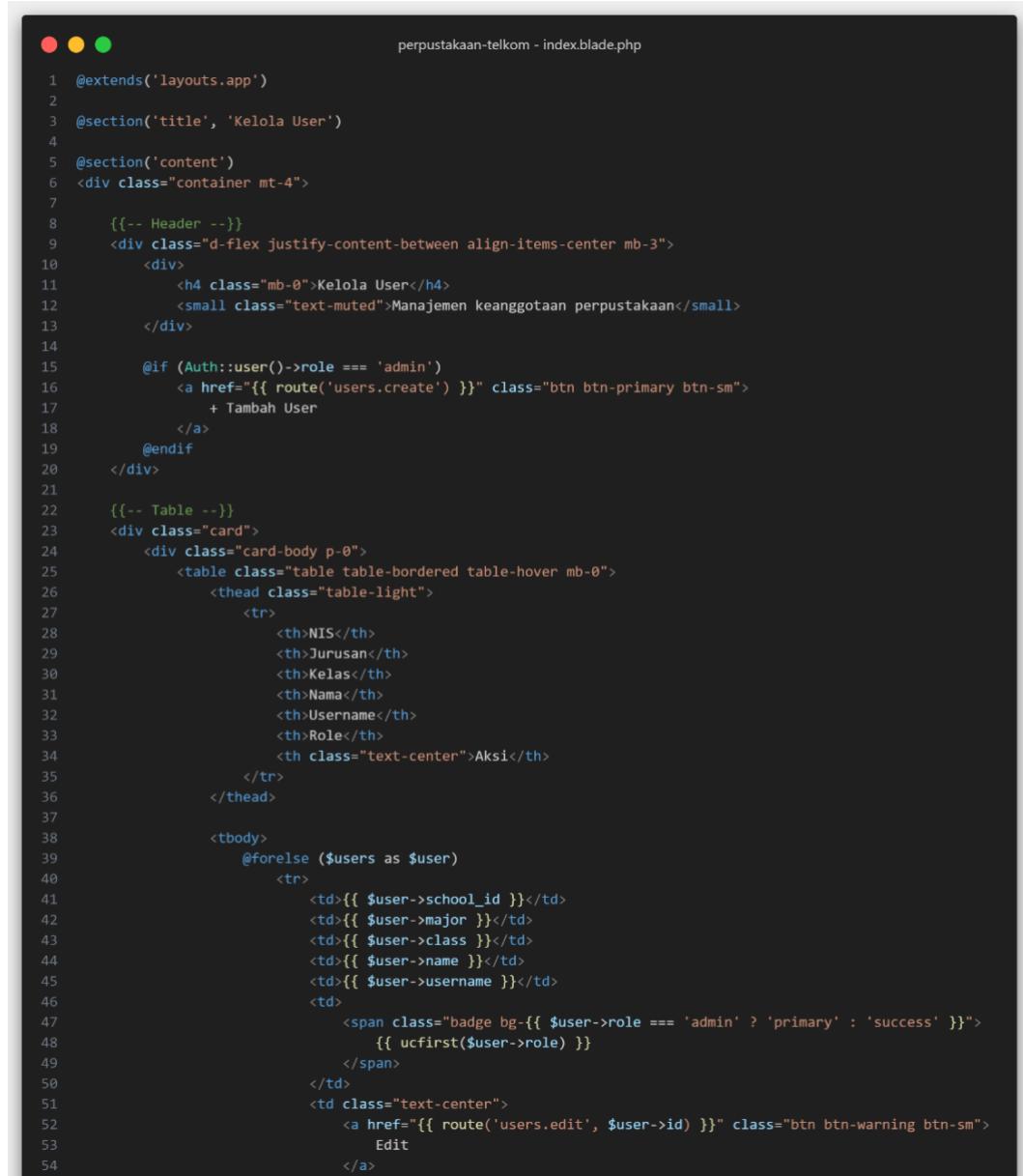
```
perpustakaan-telkom - create.blade.php

1 @extends('layouts.app')
2
3 @section('title', 'Tambah Transaksi')
4
5 @section('content')
6 <div class="container my-4">
7
8     {{-- Header --}}
9     <div class="card mb-4">
10         <div class="card-body d-flex justify-content-between align-items-center">
11             <div>
12                 <h4 class="mb-1">Tambah Transaksi</h4>
13                 <small class="text-muted">Catat peminjaman buku</small>
14             </div>
15
16             <button type="submit" form="form-transaction" class="btn btn-primary btn-sm">
17                 Simpan
18             </button>
19         </div>
20     </div>
21
22     {{-- Form --}}
23     <div class="card">
24         <div class="card-body">
25             <form id="form-transaction"
26                 action="{{ route('transactions.store') }}"
27                 method="POST">
28                 @csrf
29
30                 {{-- Buku --}}
31                 <div class="mb-3">
32                     <label class="form-label">Buku</label>
33                     <select name="book_id"
34                         class="form-select @error('book_id') is-invalid @enderror">
35                         <option value="">-- Pilih Buku --</option>
36                         @foreach ($books as $book)
37                             <option value="{{ $book->id }}"
38                                 {{ old('book_id') == $book->id ? 'selected' : '' }}>
39                                 {{ $book->title }}</option>
40                         @endforeach
41                     </select>
42                     @error('book_id')
43                         <div class="invalid-feedback">{{ $message }}</div>
44                     @enderror
45                 </div>
46             </div>
47         </div>
```

```
48     {{-- Peminjam --}}
49     <div class="mb-3">
50         <label class="form-label">Peminjam</label>
51         <select name="user_id"
52             class="form-select @error('user_id') is-invalid @enderror">
53             <option value="">-- Pilih User --</option>
54             @foreach ($users as $user)
55                 <option value="{{ $user->id }}"
56                     {{ old('user_id') == $user->id ? 'selected' : '' }}>
57                     {{ $user->name }}</option>
58             @endforeach
59         </select>
60         @error('user_id')
61             <div class="invalid-feedback">{{ $message }}</div>
62         @enderror
63     </div>
64
65     {{-- Tanggal Pinjam --}}
66     <div class="mb-3">
67         <label class="form-label">Tanggal Pinjam</label>
68         <input type="date"
69             name="borrowed_at"
70             value="{{ old('borrowed_at') }}"
71             class="form-control @error('borrowed_at') is-invalid @enderror">
72         @error('borrowed_at')
73             <div class="invalid-feedback">{{ $message }}</div>
74         @enderror
75     </div>
76
77     </form>
78     </div>
79 </div>
80 </div>
81
82 </div>
83 @endsection
84
```


2.4 Tampilan Halaman Model User

- a. Membuka file view yang telah dibuat sebelumnya di pada direktori "resources\views\users".
- b. Sesuaikan file view "index.blade.php" sesuai pada gambar berikut:



The screenshot shows a code editor with the file "index.blade.php" open. The code is written in Blade PHP template syntax. It includes sections for header, content, and a table to display user data. A conditional block checks if the user is an admin to add a new user link. The table columns include NIS, Jurusan, Kelas, Nama, Username, Role, and Aksi (Action). The Aksi column contains a badge for role and a link to edit each user.

```
1  @extends('layouts.app')
2
3  @section('title', 'Kelola User')
4
5  @section('content')
6  <div class="container mt-4">
7
8      {{-- Header --}}
9      <div class="d-flex justify-content-between align-items-center mb-3">
10         <div>
11             <h4 class="mb-0">Kelola User</h4>
12             <small class="text-muted">Manajemen keanggotaan perpustakaan</small>
13         </div>
14
15         @if (Auth::user()->role === 'admin')
16             <a href="{{ route('users.create') }}" class="btn btn-primary btn-sm">
17                 + Tambah User
18             </a>
19         @endif
20     </div>
21
22     {{-- Table --}}
23     <div class="card">
24         <div class="card-body p-0">
25             <table class="table table-bordered table-hover mb-0">
26                 <thead class="table-light">
27                     <tr>
28                         <th>NIS</th>
29                         <th>Jurusan</th>
30                         <th>Kelas</th>
31                         <th>Nama</th>
32                         <th>Username</th>
33                         <th>Role</th>
34                         <th class="text-center">Aksi</th>
35                     </tr>
36                 </thead>
37
38                 <tbody>
39                     @forelse ($users as $user)
40                         <tr>
41                             <td>{{ $user->school_id }}</td>
42                             <td>{{ $user->major }}</td>
43                             <td>{{ $user->class }}</td>
44                             <td>{{ $user->name }}</td>
45                             <td>{{ $user->username }}</td>
46                             <td>
47                                 <span class="badge bg-{{ $user->role === 'admin' ? 'primary' : 'success' }}>
48                                     {{ ucfirst($user->role) }}
49                                 </span>
50                             </td>
51                             <td class="text-center">
52                                 <a href="{{ route('users.edit', $user->id) }}" class="btn btn-warning btn-sm">
53                                     Edit
54                                 </a>
55                             </td>
56                         </tr>
57                     @empty
58                         <p>Tidak ada data user!</p>
59                     @endforelse
60                 </tbody>
61             </table>
62         </div>
63     </div>
64 
```

```
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
```

```
        @if (Auth::id() !== $user->id)
            <form action="{{ route('users.destroy', $user->id) }}" method="POST"
                  class="d-inline"
                  onsubmit="return confirm('Yakin ingin menghapus user ini?')">
                @csrf
                @method('DELETE')
                <button class="btn btn-danger btn-sm">
                  Hapus
                </button>
            </form>
        @endif
    </td>
</tr>
@empty
<tr>
    <td colspan="7" class="text-center text-muted">
        Tidak ada data user
    </td>
</tr>
@endforelse
</tbody>
</table>
</div>
</div>
</div>
@endsection
```

c. Sesuaikan file view "edit.blade.php" sesuai pada gambar berikut:

```
● ● ● perpustakaan-telkom - edit.blade.php

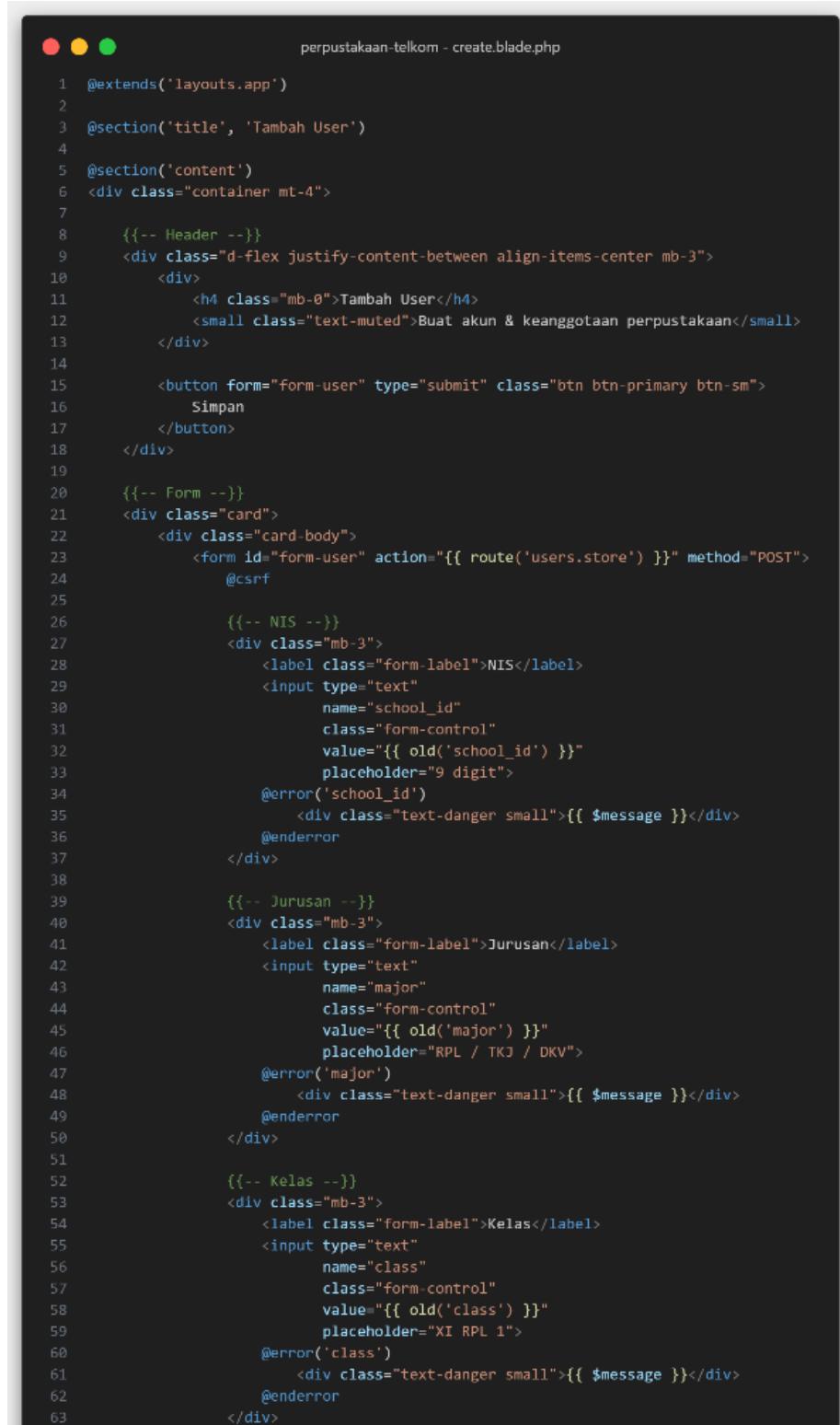
1  @extends('layouts.app')
2
3  @section('title', 'Edit User')
4
5  @section('content')
6  <div class="container mt-4">
7
8      {{-- Header --}}
9      <div class="d-flex justify-content-between align-items-center mb-3">
10         <div>
11             <h4 class="mb-0">Edit User</h4>
12             <small class="text-muted">Perbarui data pengguna</small>
13         </div>
14
15         <button form="form-user" type="submit" class="btn btn-primary btn-sm">
16             Update
17         </button>
18     </div>
19
20     {{-- Form --}}
21     <div class="card">
22         <div class="card-body">
23             <form id="form-user"
24                 action="{{ route('users.update', $user->id) }}"
25                 method="POST">
26
27                 @csrf
28                 @method('PUT')
29
30                 {{-- Username --}}
31                 <div class="mb-3">
32                     <label class="form-label">Username</label>
33                     <input type="text"
34                         name="username"
35                         class="form-control"
36                         value="{{ old('username', $user->username) }}"
37                         @error('username')
38                             <div class="text-danger small">{{ $message }}</div>
39                         @enderror
40                     </div>
41
42                 {{-- Nama --}}
43                 <div class="mb-3">
44                     <label class="form-label">Nama Lengkap</label>
45                     <input type="text"
46                         name="name"
47                         class="form-control"
48                         value="{{ old('name', $user->name) }}"
49                         @error('name')
50                             <div class="text-danger small">{{ $message }}</div>
51                         @enderror
52                     </div>
```

```

53
54     {{-- Email --}}
55     <div class="mb-3">
56         <label class="form-label">Email</label>
57         <input type="email"
58             name="email"
59             class="form-control"
60             value="{{ old('email', $user->email) }}"
61             @error('email')
62                 <div class="text-danger small">{{ $message }}</div>
63             @enderror
64     </div>
65
66     {{-- NIS --}}
67     <div class="mb-3">
68         <label class="form-label">NIS / School ID</label>
69         <input type="text"
70             name="school_id"
71             class="form-control"
72             value="{{ old('school_id', $user->school_id) }}"
73             @error('school_id')
74                 <div class="text-danger small">{{ $message }}</div>
75             @enderror
76     </div>
77
78     {{-- Jurusan --}}
79     <div class="mb-3">
80         <label class="form-label">Jurusan</label>
81         <input type="text"
82             name="major"
83             class="form-control"
84             value="{{ old('major', $user->major) }}"
85             @error('major')
86                 <div class="text-danger small">{{ $message }}</div>
87             @enderror
88     </div>
89
90     {{-- Kelas --}}
91     <div class="mb-3">
92         <label class="form-label">Kelas</label>
93         <input type="text"
94             name="class"
95             class="form-control"
96             value="{{ old('class', $user->class) }}"
97             @error('class')
98                 <div class="text-danger small">{{ $message }}</div>
99             @enderror
100    </div>
101
102    {{-- Password --}}
103    <div class="mb-3">
104        <label class="form-label">Password</label>
105        <input type="password"
106            name="password"
107            class="form-control">
108        <small class="text-muted">
109            Kosongkan jika tidak ingin mengubah password
110        </small>
111        @error('password')
112            <div class="text-danger small">{{ $message }}</div>
113        @enderror
114    </div>
115
116    </form>
117    </div>
118 </div>
119 </div>
120 @endsection
121
122

```

d. Sesuaikan file view "create.blade.php" sesuai pada gambar berikut:



```

perpustakaan-telkom - create.blade.php

1 @extends('layouts.app')
2
3 @section('title', 'Tambah User')
4
5 @section('content')
6 <div class="container mt-4">
7
8     {{-- Header --}}
9     <div class="d-flex justify-content-between align-items-center mb-3">
10        <div>
11            <h4 class="mb-0">Tambah User</h4>
12            <small class="text-muted">Buat akun & keanggotaan perpustakaan</small>
13        </div>
14
15        <button form="form-user" type="submit" class="btn btn-primary btn-sm">
16            Simpan
17        </button>
18    </div>
19
20    {{-- Form --}}
21    <div class="card">
22        <div class="card-body">
23            <form id="form-user" action="{{ route('users.store') }}" method="POST">
24                @csrf
25
26                {{-- NIS --}}
27                <div class="mb-3">
28                    <label class="form-label">NIS</label>
29                    <input type="text"
30                        name="school_id"
31                        class="form-control"
32                        value="{{ old('school_id') }}"
33                        placeholder="9 digit">
34                    @error('school_id')
35                        <div class="text-danger small">{{ $message }}</div>
36                    @enderror
37                </div>
38
39                {{-- Jurusan --}}
40                <div class="mb-3">
41                    <label class="form-label">Jurusan</label>
42                    <input type="text"
43                        name="major"
44                        class="form-control"
45                        value="{{ old('major') }}"
46                        placeholder="RPL / TKJ / DKV">
47                    @error('major')
48                        <div class="text-danger small">{{ $message }}</div>
49                    @enderror
50                </div>
51
52                {{-- Kelas --}}
53                <div class="mb-3">
54                    <label class="form-label">Kelas</label>
55                    <input type="text"
56                        name="class"
57                        class="form-control"
58                        value="{{ old('class') }}"
59                        placeholder="XI RPL 1">
60                    @error('class')
61                        <div class="text-danger small">{{ $message }}</div>
62                    @enderror
63                </div>

```

```

55      {{-- Username --}}
56      <div class="mb-3">
57          <label class="form-label">Username</label>
58          <input type="text"
59              name="username"
60              class="form-control"
61              value="{{ old('username') }}"
62              @error('username')
63                  <div class="text-danger small">{{ $message }}</div>
64              @enderror
65      </div>
66
67      {{-- Nama --}}
68      <div class="mb-3">
69          <label class="form-label">Nama Lengkap</label>
70          <input type="text"
71              name="name"
72              class="form-control"
73              value="{{ old('name') }}"
74              @error('name')
75                  <div class="text-danger small">{{ $message }}</div>
76              @enderror
77      </div>
78
79      {{-- Email --}}
80      <div class="mb-3">
81          <label class="form-label">Email</label>
82          <input type="email"
83              name="email"
84              class="form-control"
85              value="{{ old('email') }}"
86              @error('email')
87                  <div class="text-danger small">{{ $message }}</div>
88              @enderror
89      </div>
90
91      {{-- Password --}}
92      <div class="mb-3">
93          <label class="form-label">Password</label>
94          <input type="password"
95              name="password"
96              class="form-control"
97              @error('password')
98                  <div class="text-danger small">{{ $message }}</div>
99              @enderror
100         </div>
101
102     {{-- Role --}}
103     <div class="mb-3">
104         <label class="form-label">Role</label>
105         <select name="role" class="form-select">
106             <option value="member" selected>Member</option>
107             <option value="admin">Admin</option>
108         </select>
109         @error('role')
110             <div class="text-danger small">{{ $message }}</div>
111         @enderror
112     </div>
113
114         </form>
115     </div>
116 </div>
117
118 </div>
119 @endsection
120
121
122
123
124
125
126
127
128 </div>
129
130

```

KONFIGURASI MIDDLEWARE, CONTROLLER DAN ROUTES

Middleware

Middleware adalah salah satu fitur penting dalam Laravel yang berfungsi sebagai penyaring setiap permintaan yang masuk ke server sebelum mencapai controller.

Dalam modul ini, middleware "`CheckRole.php`" diimplementasikan untuk memastikan bahwa hanya pengguna dengan level akses tertentu yang dapat membuka halaman-halaman tersebut.

Controller

Controller adalah komponen yang bertanggung jawab untuk menangani logika aplikasi atau sebagai jembatan antara routing (menerima input), model (proses data) dan view (tampilan).

Pada controller "`BookController.php`", "`TransactionController.php`", "`UserController.php`". bertugas menerima input dari pengguna dan berinteraksi dengan model untuk mengambil atau menyimpan data dan juga menentukan view mana yang akan ditampilkan.

Routes

Routes adalah sebuah peta alamat URL dalam aplikasi yang digunakan untuk menentukan bagaimana aplikasi merespons permintaan dari pengguna. Sehingga aplikasi tahu apa yang harus dilakukan saat pengguna mengakses alamat tertentu.

Konfigurasi pada routes ini dilakukan di halaman "[routes/web.php](#)" .

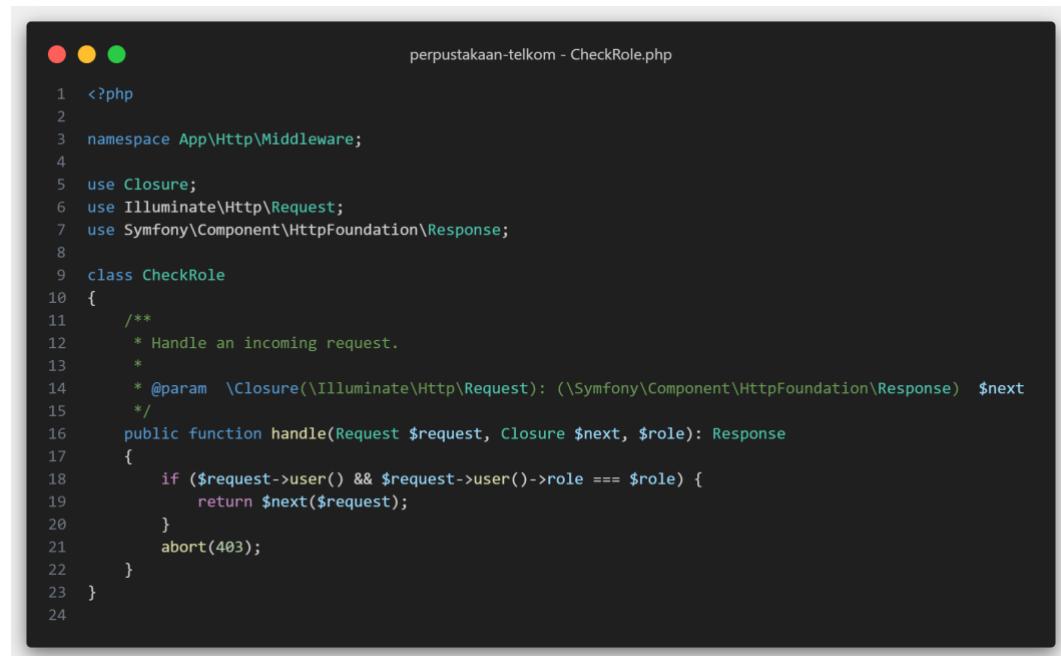
3.1 Middleware Hak Akses Admin dan Member

- a. Membuka terminal di visual studio code dan jalankan perintah "`php artisan make:middleware CheckRole`" sesuai gambar berikut:

```
PS D:\Workspace\perpustakaan-telkom> php artisan make:middleware CheckRole
[INFO] Middleware [D:\Workspace\perpustakaan-telkom\app\Http\Middleware\CheckRole.php] created successfully.

PS D:\Workspace\perpustakaan-telkom>
```

- b. Membuka file middleware yang telah dibuat sebelumnya di pada direktori "`app\http\Middleware`".
- c. Sesuaikan file middleware "`CheckRole.php`" sesuai pada gambar berikut:



```
● ● ●
perpustakaan-telkom - CheckRole.php

1 <?php
2
3 namespace App\Http\Middleware;
4
5 use Closure;
6 use Illuminate\Http\Request;
7 use Symfony\Component\HttpFoundation\Response;
8
9 class CheckRole
10 {
11     /**
12      * Handle an incoming request.
13      *
14      * @param \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response) $next
15      */
16     public function handle(Request $request, Closure $next, $role): Response
17     {
18         if ($request->user() && $request->user()->role === $role) {
19             return $next($request);
20         }
21         abort(403);
22     }
23 }
24
```

- d. Membuka file sistem laravel pada direktori "`bootstrap`".
- e. Sesuaikan file sistem laravel "`app.php`" sesuai pada gambar berikut:

```
perpustakaan-telkom - app.php

1 <?php
2
3 use Illuminate\Foundation\Application;
4 use Illuminate\Foundation\Configuration\Exceptions;
5 use Illuminate\Foundation\Configuration\Middleware;
6 use \App\Http\Middleware\CheckRole;
7
8 return Application::configure(basePath: dirname(__DIR__))
9     ->withRouting(
10         web: __DIR__.'/../routes/web.php',
11         commands: __DIR__.'/../routes/console.php',
12         health: '/up',
13     )
14     ->withMiddleware(function (Middleware $middleware) {
15         $middleware->alias([
16             'checkRole' => CheckRole::class,
17         ]);
18     })
19
20     ->withExceptions(function (Exceptions $exceptions): void {
21         /**
22     })->create();
23
```


3.2 Controller Model Book

- a. Membuka file controller di direktori "app/Http/Controllers" dengan kata kunci "BookController.php".
- b. Menyesuaikan controller book sesuai gambar berikut:



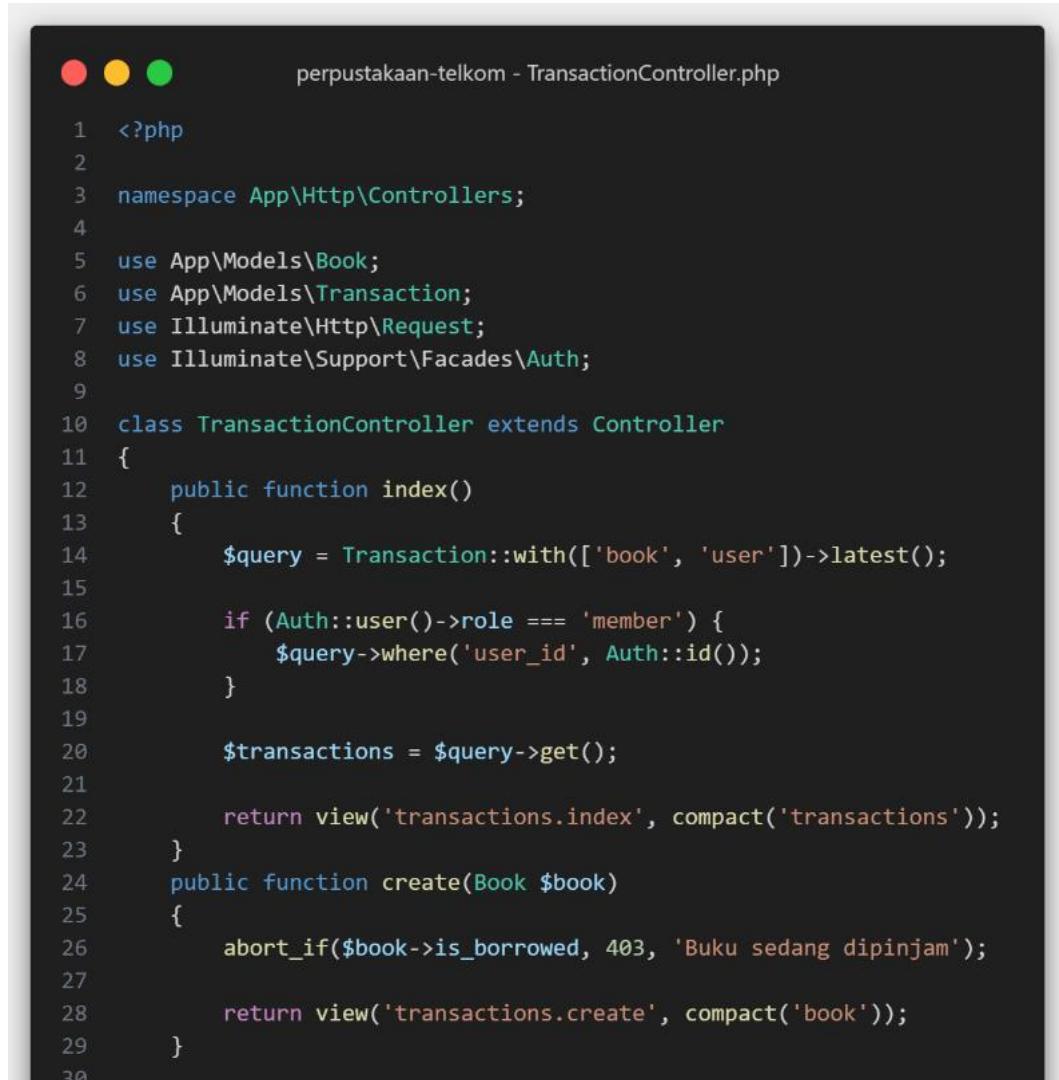
The screenshot shows a code editor window with a dark theme. At the top, there are three colored circular icons (red, yellow, green) followed by the text "perpustakaan-telkom - BookController.php". The code itself is a PHP file named BookController.php, which defines a class BookController extending the Controller class. It contains three methods: index(), create(), and store(). The store() method includes validation rules for 'title' and 'author' fields, and it uses the Book model's create() method to save the new book.

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\Book;
6 use App\Models\Transaction;
7 use Illuminate\Http\Request;
8 use Illuminate\Support\Facades\Auth;
9
10 class BookController extends Controller
11 {
12     public function index()
13     {
14         $books = Book::latest()->get();
15
16         return view('books.index', compact(
17             'books',
18         ));
19     }
20
21     public function create()
22     {
23         return view('books.create');
24     }
25
26     public function store(Request $request)
27     {
28         $request->validate([
29             'title' => 'required|string|max:255',
30             'author' => 'required|string|max:255',
31         ]);
32
33         Book::create($request->all());
34
35         return redirect()->route('books.index')
36             ->with('success', 'Book berhasil ditambahkan');
37     }
38 }
```

```
39     public function edit(Book $book)
40     {
41         return view('books.edit', compact('book'));
42     }
43
44     public function update(Request $request, Book $book)
45     {
46         $request->validate([
47             'book_code' => 'required|string|max:50',
48             'title' => 'required|string|max:255',
49             'author' => 'required|string|max:255',
50             'publisher' => 'required|string|max:255',
51             'year' => 'required|digits:4',
52         ]);
53
54         $book->update($request->only([
55             'book_code', 'title', 'author', 'publisher', 'year',
56         ]));
57
58         return redirect()->route('books.index')
59             ->with('success', 'Buku berhasil diperbarui');
60     }
61
62     public function borrow(Book $book)
63     {
64
65         Transaction::create([
66             'user_id'      => Auth::id(),
67             'book_id'      => $book->id,
68             'borrowed_at'  => now(),
69             'status'       => 'borrowed',
70         ]);
71
72
73         $book->update(['is_borrowed' => true]);
74
75         return redirect()->route('books.index')
76             ->with('success', 'Buku berhasil dipinjam');
77     }
78
79     public function destroy(Book $book)
80     {
81         $book->delete();
82
83         return redirect()->route('books.index')
84             ->with('success', 'Book berhasil dihapus');
85     }
86 }
87 }
```

3.3 Controller Model Transaction

- c. Membuka file controller di direktori "app/Http/Controllers" dengan kata kunci "**TransactionController.php**".
- d. Menyesuaikan controller transaction sesuai gambar berikut:



The screenshot shows a code editor window with a dark theme. At the top, there are three circular icons (red, yellow, green) and the text "perpustakaan-telkom - TransactionController.php". The code itself is a PHP file named TransactionController.php, which defines a class TransactionController that extends the Controller class. It contains two methods: index() and create(). The index() method queries the Transaction model, filtering by user role ('member') and returns the results via a view. The create() method checks if a book is borrowed and returns a 403 error if it is.

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\Book;
6 use App\Models\Transaction;
7 use Illuminate\Http\Request;
8 use Illuminate\Support\Facades\Auth;
9
10 class TransactionController extends Controller
11 {
12     public function index()
13     {
14         $query = Transaction::with(['book', 'user'])->latest();
15
16         if (Auth::user()->role === 'member') {
17             $query->where('user_id', Auth::id());
18         }
19
20         $transactions = $query->get();
21
22         return view('transactions.index', compact('transactions'));
23     }
24     public function create(Book $book)
25     {
26         abort_if($book->is_borrowed, 403, 'Buku sedang dipinjam');
27
28         return view('transactions.create', compact('book'));
29     }
30 }
```

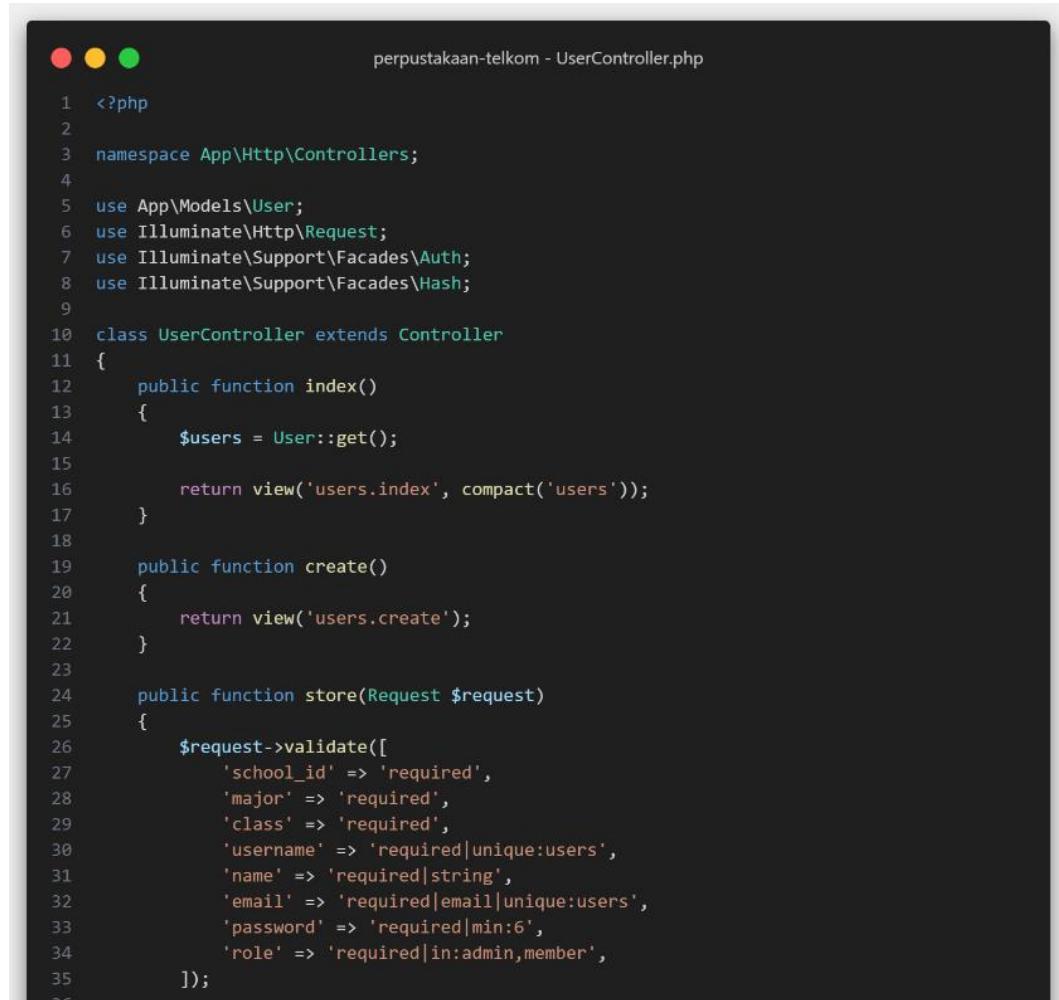
```
--  
31     public function store(Request $request, Book $book)  
32     {  
33         $request->validate([  
34             'borrower_name' => 'required|string|max:255',  
35         ]);  
36  
37         Transaction::create([  
38             'book_id'        => $book->id,  
39             'borrower_name' => $request->borrower_name,  
40             'borrowed_at'   => now(),  
41             'status'        => 'borrowed',  
42         ]);  
43  
44         $book->update(['is_borrowed' => true]);  
45  
46         return redirect()->route('books.index')  
47             ->with('success', 'Buku berhasil dipinjam');  
48     }  
49  
50     public function return(Transaction $transaction)  
51     {  
52         if (  
53             (Auth::user()->role == 'member') &&  
54             ($transaction->user_id != Auth::id())  
55         ) {  
56             abort(403);  
57         }  
58  
59         if ($transaction->status === 'returned') {  
60             return redirect()  
61                 ->route('transactions.index')  
62                 ->with('error', 'Buku ini sudah dikembalikan');  
63         }  
64  
65         $transaction->update([  
66             'returned_at' => now(),  
67             'status'       => 'returned',  
68         ]);  
69  
70         $transaction->book->update([  
71             'is_borrowed' => false,  
72         ]);  
73  
74         return redirect()  
75             ->route('transactions.index')  
76             ->with('success', 'Buku berhasil dikembalikan');  
77     }  
78 }  
79 }
```

3.4 Controller Model User

- a. Membuka terminal di visual studio code dan jalankan perintah sesuai gambar berikut:

```
PS D:\Workspace\perpustakaan-telkom> php artisan make:controller UserController
INFO Controller [D:\Workspace\perpustakaan-telkom\app\Http\Controllers\UserController.php] created successfully.
PS D:\Workspace\perpustakaan-telkom>
```

- b. Membuka file controller yang telah dibuat sebelumnya di pada direktori "**app\http\Controller**".
- c. Sesuaikan file controller "**UserController.php**" sesuai pada gambar berikut:



```
perpustakaan-telkom - UserController.php

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\User;
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\Auth;
8 use Illuminate\Support\Facades\Hash;
9
10 class UserController extends Controller
11 {
12     public function index()
13     {
14         $users = User::get();
15
16         return view('users.index', compact('users'));
17     }
18
19     public function create()
20     {
21         return view('users.create');
22     }
23
24     public function store(Request $request)
25     {
26         $request->validate([
27             'school_id' => 'required',
28             'major' => 'required',
29             'class' => 'required',
30             'username' => 'required|unique:users',
31             'name' => 'required|string',
32             'email' => 'required|email|unique:users',
33             'password' => 'required|min:6',
34             'role' => 'required|in:admin,member',
35         ]);
36     }
37 }
```

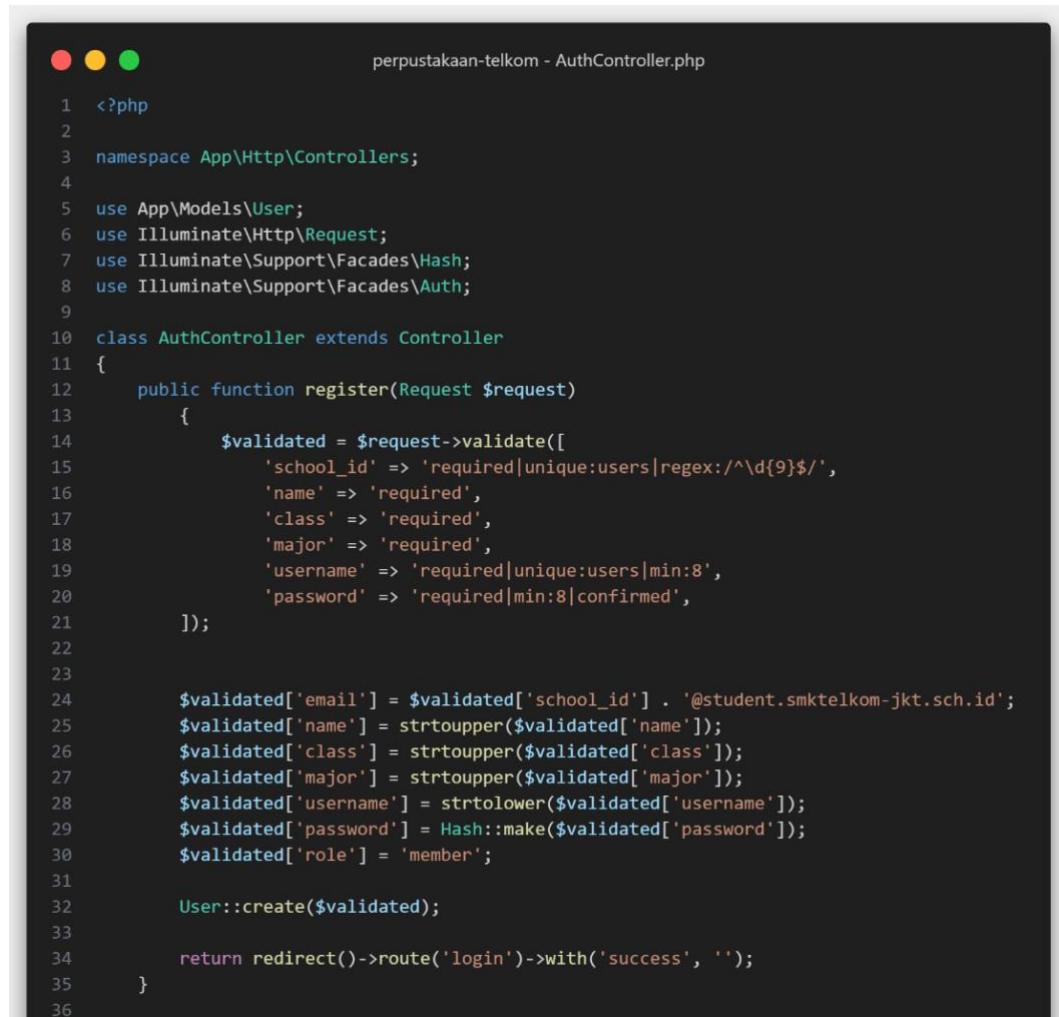
```
76
77     User::create([
78         'school_id' => $request->school_id,
79         'major' => $request->major,
80         'class' => $request->class,
81         'username' => $request->username,
82         'name' => $request->name,
83         'email' => $request->email,
84         'password' => Hash::make($request->password),
85         'role' => $request->role,
86     ]);
87
88
89     return redirect()->route('users.index')
90         ->with('success', 'User berhasil ditambahkan');
91     }
92
93     public function update(Request $request, User $user)
94     {
95         // Validasi
96         $validated = $request->validate([
97             'username' => 'required|string|max:255|unique:users,username,' . $user->id,
98             'name' => 'required|string|max:255',
99             'email' => 'required|email|unique:users,email,' . $user->id,
100            'school_id' => 'nullable|string|max:50',
101            'major' => 'nullable|string|max:100',
102            'class' => 'nullable|string|max:50',
103            'password' => 'nullable|string|min:6',
104        ]);
105
106
107        if ($request->filled('password')) {
108            $validated['password'] = Hash::make($request->password);
109        } else {
110            unset($validated['password']);
111        }
112
113        $user->update($validated);
114
115
116        return redirect()
117            ->route('users.index')
118            ->with('success', 'Data user berhasil diperbarui');
119    }
120
121    public function edit(User $user)
122    {
123        return view('users.edit', compact('user'));
124    }
125
126    public function destroy(User $user)
127    {
128        if ($user->id == User::user()->id) {
129            abort(403, '');
130        }
131
132        $user->delete();
133
134        return redirect()->route('users.index')
135            ->with('success', 'User berhasil dihapus');
136    }
137 }
```

3.5 Controller Sistem Autentikasi

- a. Membuka terminal di visual studio code dan jalankan perintah "`php artisan make:controller AuthController`" sesuai gambar berikut:

```
PS D:\Workspace\perpustakaan-telkom> php artisan make:controller AuthController
[INFO] Controller [D:\Workspace\perpustakaan-telkom\app\Http\Controllers\AuthController.php] created successfully.
PS D:\Workspace\perpustakaan-telkom>
```

- b. Membuka file controller yang telah dibuat sebelumnya di pada direktori "`app\http\Controller`".
- c. Sesuaikan file controller "`AuthController.php`" sesuai pada gambar berikut:

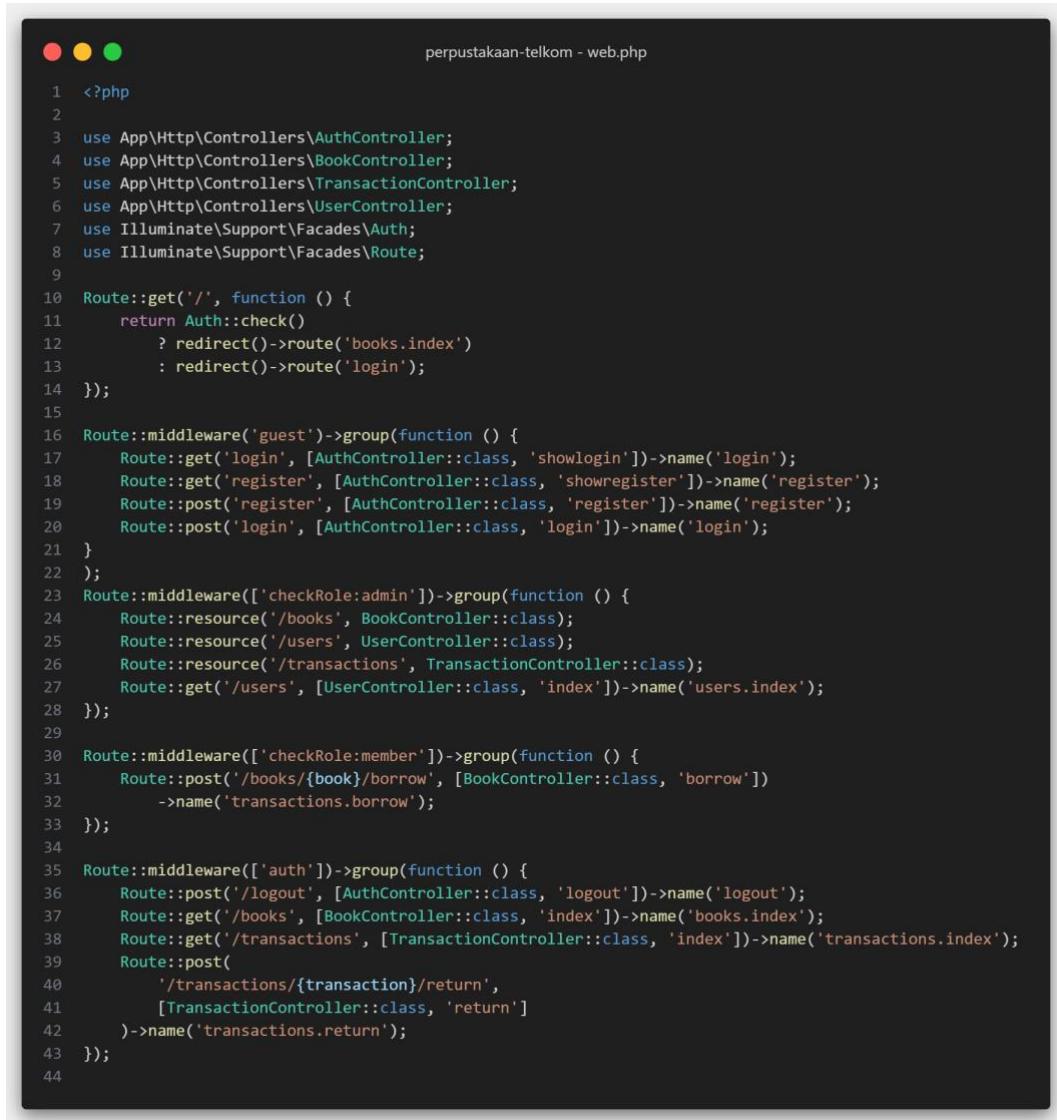


```
perpustakaan-telkom - AuthController.php

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\User;
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\Hash;
8 use Illuminate\Support\Facades\Auth;
9
10 class AuthController extends Controller
11 {
12     public function register(Request $request)
13     {
14         $validated = $request->validate([
15             'school_id' => 'required|unique:users|regex:/^\d{9}$/',
16             'name' => 'required',
17             'class' => 'required',
18             'major' => 'required',
19             'username' => 'required|unique:users|min:8',
20             'password' => 'required|min:8|confirmed',
21         ]);
22
23
24         $validated['email'] = $validated['school_id'] . '@student.smktelkom-jkt.sch.id';
25         $validated['name'] = strtoupper($validated['name']);
26         $validated['class'] = strtoupper($validated['class']);
27         $validated['major'] = strtoupper($validated['major']);
28         $validated['username'] = strtolower($validated['username']);
29         $validated['password'] = Hash::make($validated['password']);
30         $validated['role'] = 'member';
31
32         User::create($validated);
33
34         return redirect()->route('login')->with('success', '');
35     }
36 }
```


3.6 Routes

- e. Membuka file routes di direktori "routes" dengan kata kunci "web.php".
- f. Menyesuaikan routes sesuai gambar berikut:



```
perpustakaan-telkom - web.php

1 <?php
2
3 use App\Http\Controllers\AuthController;
4 use App\Http\Controllers\BookController;
5 use App\Http\Controllers\TransactionController;
6 use App\Http\Controllers\UserController;
7 use Illuminate\Support\Facades\Auth;
8 use Illuminate\Support\Facades\Route;
9
10 Route::get('/', function () {
11     return Auth::check()
12         ? redirect()->route('books.index')
13         : redirect()->route('login');
14 });
15
16 Route::middleware('guest')->group(function () {
17     Route::get('login', [AuthController::class, 'showlogin'])->name('login');
18     Route::get('register', [AuthController::class, 'showregister'])->name('register');
19     Route::post('register', [AuthController::class, 'register'])->name('register');
20     Route::post('login', [AuthController::class, 'login'])->name('login');
21 });
22
23 Route::middleware(['checkRole:admin'])->group(function () {
24     Route::resource('/books', BookController::class);
25     Route::resource('/users', UserController::class);
26     Route::resource('/transactions', TransactionController::class);
27     Route::get('/users', [UserController::class, 'index'])->name('users.index');
28 });
29
30 Route::middleware(['checkRole:member'])->group(function () {
31     Route::post('/books/{book}/borrow', [BookController::class, 'borrow'])
32         ->name('transactions.borrow');
33 });
34
35 Route::middleware(['auth'])->group(function () {
36     Route::post('/logout', [AuthController::class, 'logout'])->name('logout');
37     Route::get('/books', [BookController::class, 'index'])->name('books.index');
38     Route::get('/transactions', [TransactionController::class, 'index'])->name('transactions.index');
39     Route::post(
40         '/transactions/{transaction}/return',
41         [TransactionController::class, 'return']
42     )->name('transactions.return');
43 });
44
```

REPOSITORY

Google Drive: <http://s.id/AppPerpus>

https://drive.google.com/drive/folders/1jHu4-GEH-4KWj7V8cuC9Ahf4uEDLiUSC?usp=drive_link

1. Buka project laravel dengan VSCode, hapus foler **vendor** (**jika masih ada**).
2. Jalankan command **composer install**.
3. Pada file .env ubah value **DB_CONNECTION** menjadi **mysql** dan **DB_DATABASE** menjadi nama database yang akan digunakan.
4. Jalankan command **php artisan migrate**, jalankan juga seeder yang berisi data buku-buku dan akun admin. Atau bisa dengan menjalankan keseluruhannya dengan command **php artisan migrate --seed**
5. Jalankan project laravel.

Catatan : Front-end menggunakan Bootstrap versi online, pastikan terhubung dengan internet.

Github: <https://github.com/VickyLbr/perpustakaan-telkom>