



Mata Kuliah : Pemrograman Mobile
Program Studi : D4 – Sistem Informasi Bisnis
Semester : 5

Kelas : SIB 3D
NIM : 2241760066
Nama : Adelia Syaharani Hermawan
Jobsheet Ke- : API laravel breeze

Laporan Jobsheet

Praktikum Ke-1

Langkah	Jawaban/Deskripsi
1	<p>Step 1: Install Laravel</p> <p>First, create a new Laravel project using the Laravel installer or Composer.</p> <p>laravel new api-breeze</p> <p># Or via Composer</p> <p>composer create-project laravel/laravel api-breeze</p> <p>cd api-breeze</p> <pre>Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows PS D:\Adelia SIB 3D\Mobile> cd api-breeze PS D:\Adelia SIB 3D\Mobile\api-breeze> composer require laravel/breeze --dev Cannot use laravel/breeze's latest version v2.2.5 as it requires php ^8.2.0 which is not satisfied by your platform. ./composer.json has been updated Running composer update laravel/breeze Loading composer repositories with package information Updating dependencies Lock file operations: 112 installs, 0 updates, 0 removals - Locking brick/math (0.12.1) - Locking carbonphp/carbon-doctrine-types (2.1.0) - Locking dflydev/dot-access-data (v3.0.3)</pre>
2	<p>Step 2: Install Laravel Breeze</p> <p>Next, install Laravel Breeze and its dependencies.</p> <p>composer require laravel/breeze --dev</p> <p>php artisan breeze:install api</p> <pre>PS D:\Adelia SIB 3D\Mobile\api-breeze> php artisan breeze:install api INFO Breeze scaffolding installed successfully.</pre>



3	<p>Step 3: Configure the Database and Run Migrations</p> <p>1. Update your .env file with your database credentials:</p> <pre>12 DB_CONNECTION=mysql 13 DB_HOST=127.0.0.1 14 DB_PORT=3306 15 DB_DATABASE=laravel11_api 16 DB_USERNAME=root 17 DB_PASSWORD=</pre>
	<p>2. Run the migrations to set up your database tables:</p> <p>php artisan migrate</p> <pre>PS D:\Adelia SIB 3D\Mobile\api-breeze> php artisan migrate INFO: Preparing database. Creating migration table 16ms DONE INFO: Running migrations. 2014_10_12_000000_create_users_table 27ms DONE 2014_10_12_100000_create_password_reset_tokens_table 7ms DONE 2019_08_19_000000_create_failed_jobs_table 27ms DONE</pre>
4	<p>Step 4: Create Authentication Endpoints</p> <p>Laravel Breeze provides the necessary endpoints for registration, login, and logout. The routes are defined in routes/api.php.</p> <pre>19 Route::post('/register', [RegisteredUserController::class, 'store']); 20 Route::post('/login', [AuthenticatedSessionController::class, 'store']); 21 Route::post('/logout', [AuthenticatedSessionController::class, 'destroy'])->middleware('auth:sanctum'); 22 23 Route::middleware(['auth:sanctum'])->group(function () { 24 Route::get('/user', function (Request \$request) { 25 return \$request->user(); 26 }); 27 });</pre>
5	<p>Step 5: Update Controllers</p> <p>Modify the RegisteredUserController and AuthenticatedSessionController</p> <pre>app > Http > Controllers > Auth > AuthenticatedSessionController.php > ... 44 use Illuminate\Http\Request; 45 use Illuminate\Support\Facades\Auth; 46 use App\Http\Controllers\Controller; 47 48 class AuthenticatedSessionController extends Controller 49 { 50 public function store(Request \$request) 51 { 52 \$request->validate([53 'email' => ['required', 'string', 'email'], 54 'password' => ['required', 'string'], 55]); 56 57 if (!Auth::attempt(\$request->only('email', 'password'))) { 58 return response()->json(['message' => 'Invalid login credentials'], 401); 59 } 60 61 \$user = Auth::user(); 62 \$token = \$user->createToken('auth_token')->plainTextToken; 63 64 return response()->json([65 'access_token' => \$token, 66 'token_type' => 'Bearer', 67 'user' => \$user, 68 'status' => 'Login successful', 69]); 70 }</pre>



6	Running artisan serve 
7	Check Following API Test Registrasi di postman 
	Test login 



Pembuatan Aplikasi Mobile Flutter

Langkah	Jawaban/Deskripsi
1	Buat project flutter baru 
2	Tambahkan dependencies, lalu jalankan flutter pub get 
3	Atur struktur folder proyek 
4	Buat model user_model.dart 



5	Membuat file auth_service untuk controller authenticate api saat login <pre>pubspec.yaml x user_model.dart U auth_services U x lib > services > auth_services 1 import 'dart:convert'; 2 import 'package:http/http.dart' as http; 3 import 'package:flutter_secure_storage/flutter_secure_storage.dart'; 4 import '../models/user_model.dart'; 5 6 class AuthService { 7 // URL API 8 final String apiUrl = 'http://your-laravel-api-url.com/api'; 9 10 // Penyimpanan aman untuk token 11 final FlutterSecureStorage storage = FlutterSecureStorage(); 12 13 /// Login dengan email dan password 14 Future<bool> login(String email, String password) async { 15 try { 16 final response = await http.post(17 Uri.parse('\$apiUrl/login'), 18 headers: {'Content-Type': 'application/json'}, 19 body: jsonEncode({'email': email, 'password': password}), 20); 21 22 if (response.statusCode == 200) { 23 final data = jsonDecode(response.body); 24 await storage.write(key: 'token', value: data['token']); 25 return true; 26 } else { 27 print('Login failed: \${response.body}'); 28 return false; 29 } 30 } catch (e) {</pre>
6	Membuat auth providers <pre>lib > providers > auth_provider.dart > AuthProvider > logout 1 import 'package:flutter/material.dart'; 2 import '../models/user_model.dart'; 3 import '../services/auth_service.dart'; 4 5 class AuthProvider with ChangeNotifier { 6 final AuthService _authService = AuthService(); 7 User? _user; 8 9 // Getter untuk mendapatkan user saat ini 10 User? get user => _user; 11 12 /// Login dengan email dan password 13 Future<bool> login(String email, String password) async { 14 bool success = await _authService.login(email, password); 15 if (success) { 16 _user = await _authService.getProfile(); 17 notifyListeners(); 18 } 19 return success; 20 }</pre>
7	Membuat login_screen <pre>lib > screens > login_screen.dart > ... 1 import 'package:flutter/material.dart'; 2 import 'package:provider/provider.dart'; 3 import '../providers/auth_provider.dart'; 4 5 class LoginScreen extends StatelessWidget { 6 // Controller untuk input email dan password 7 final TextEditingController emailController = TextEditingController(); 8 final TextEditingController passwordController = TextEditingController(); 9 10 @override 11 Widget build(BuildContext context) { 12 final authProvider = Provider.of<AuthProvider>(context); 13 14 return Scaffold(15 appBar: AppBar(title: const Text('Login')), 16 body: Padding(17 padding: const EdgeInsets.all(16.0), 18 child: Column(</pre>



8

Membuat profile_screen

```
lib > screens > profile_screen.dart > ...
1  import 'package:flutter/material.dart';
2  import 'package:provider/provider.dart';
3  import '../providers/auth_provider.dart';
4
5  class ProfileScreen extends StatelessWidget {
6    @override
7    Widget build(BuildContext context) {
8      final authProvider = Provider.of<AuthProvider>(context);
9      final user = authProvider.user;
10
11      return Scaffold(
12        appBar: AppBar(
13          title: const Text('Profile'),
14          actions: [
15            // Tombol Logout
16            IconButton(
17              icon: const Icon(Icons.logout),
```

9

Hasil running

