

Assignment 6 PBW

Nama : Muhammad Zidan Ernandiaz

NIM: 6706223117

Kelas : D3IF – 46-04

- Struktur Tabel :

- Users

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	username	varchar(100)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	3	fullname	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	4	email	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	5	password	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	6	address	varchar(1000)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	7	birthdate	date			No	None			Change Drop More
<input type="checkbox"/>	8	phoneNumber	varchar(20)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	9	remember_token	varchar(100)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	10	created_at	timestamp			Yes	NULL			Change Drop More
<input type="checkbox"/>	11	updated_at	timestamp			Yes	NULL			Change Drop More

- Collections

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	nama	varchar(100)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	3	jenis	tinyint(4)			No	None			Change Drop More
<input type="checkbox"/>	4	created_at	datetime			No	current_timestamp()			Change Drop More
<input type="checkbox"/>	5	jumlahAwal	int(11)			No	None			Change Drop More
<input type="checkbox"/>	6	jumlahSisa	int(11)			No	None			Change Drop More
<input type="checkbox"/>	7	jumlahKeluar	int(11)			No	None			Change Drop More

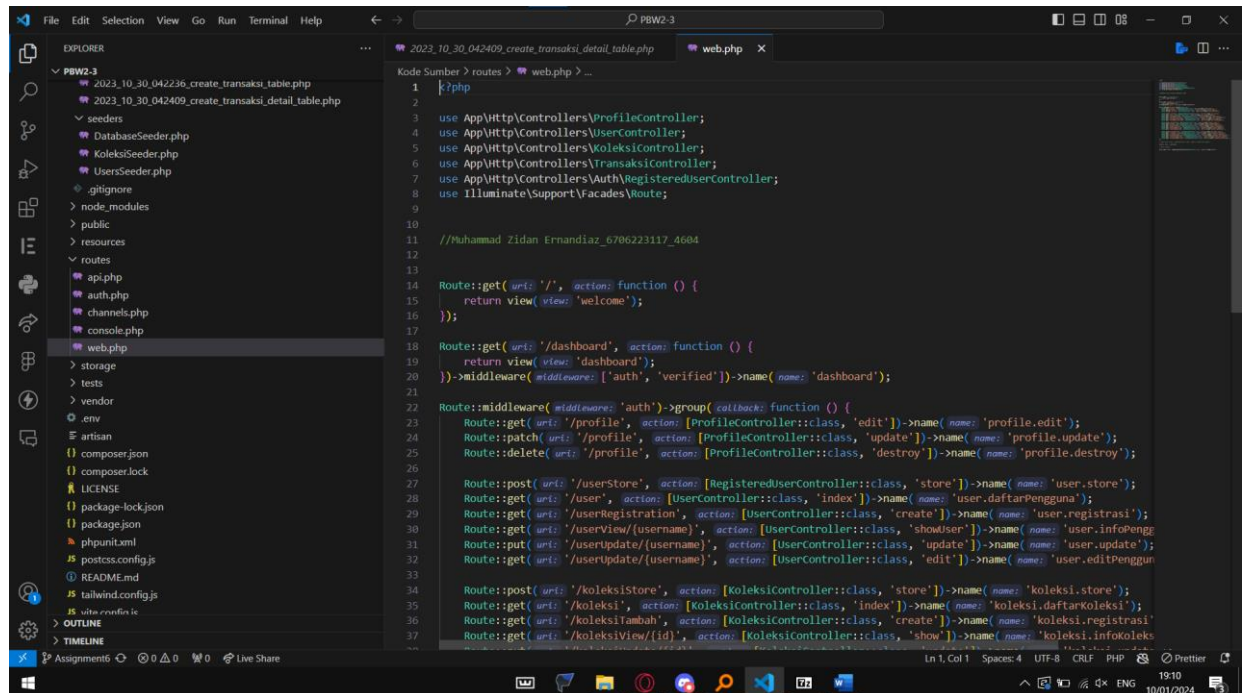
☐ Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

- Transactions

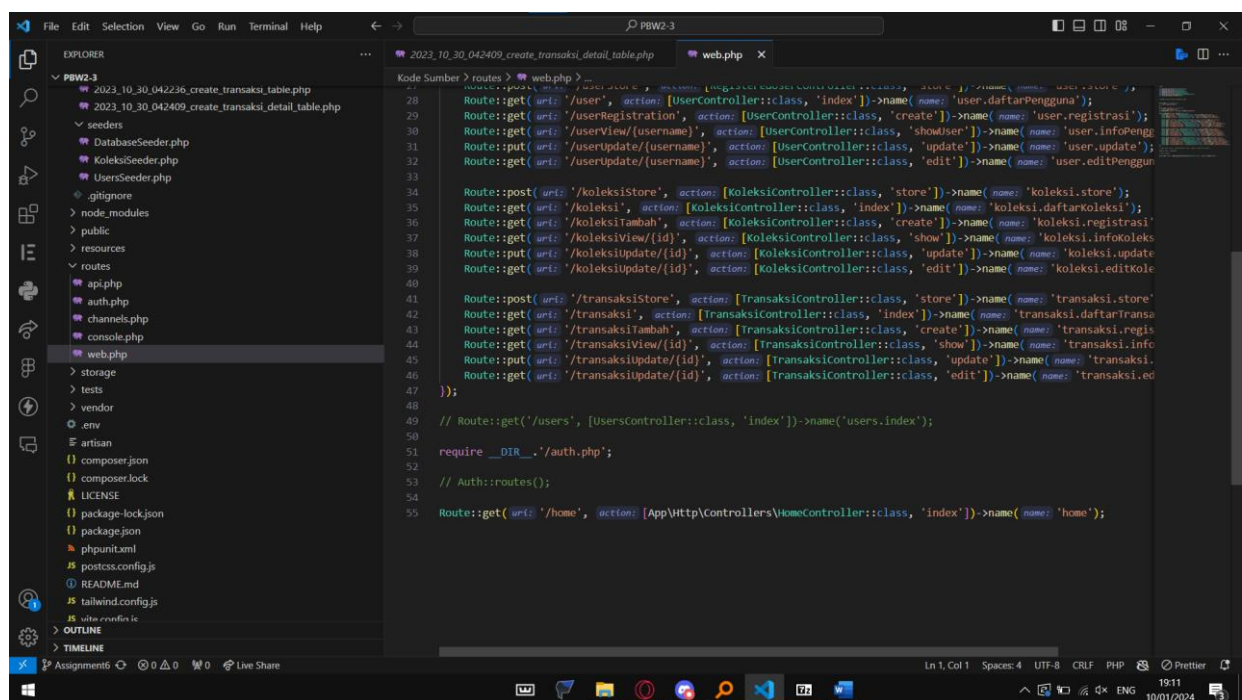
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	useridPetugas	bigint(20)		UNSIGNED	No	None			Change Drop More
<input type="checkbox"/>	3	useridPeminjam	bigint(20)		UNSIGNED	No	None			Change Drop More
<input type="checkbox"/>	4	tanggalPinjam	date			No	None			Change Drop More
<input type="checkbox"/>	5	tanggalSelesai	date			Yes	NULL			Change Drop More
<input type="checkbox"/>	6	created_at	datetime			Yes	NULL			Change Drop More
<input type="checkbox"/>	7	updated_at	datetime			Yes	NULL			Change Drop More

- Detail_transactions

1. File kelas routes/web.php



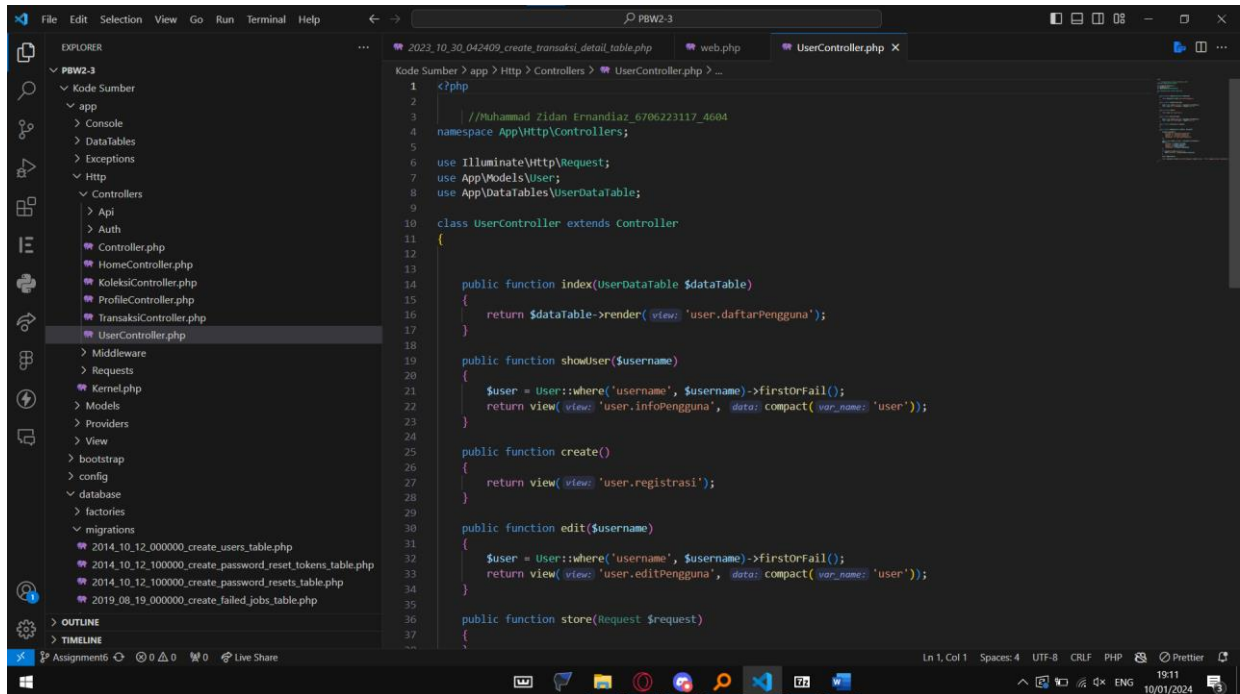
```
1 <?php
2
3 use App\Http\Controllers\ProfileController;
4 use App\Http\Controllers\UserController;
5 use App\Http\Controllers\KoleksiController;
6 use App\Http\Controllers\TransaksiController;
7 use App\Http\Controllers\Auth\RegisteredUserController;
8 use Illuminate\Support\Facades\Route;
9
10
11 //Muhammad Zidan Ernandiaz_6706223117_4604
12
13
14 Route::get('/', action: function () {
15     return view('welcome');
16 });
17
18 Route::get('/dashboard', action: function () {
19     return view('dashboard');
20 })->middleware(['auth', 'verified'])->name('dashboard');
21
22 Route::middleware(['auth'])->group(function () {
23     Route::get('/profile', action: [ProfileController::class, 'edit'])->name('profile.edit');
24     Route::patch('/profile', action: [ProfileController::class, 'update'])->name('profile.update');
25     Route::delete('/profile', action: [ProfileController::class, 'destroy'])->name('profile.destroy');
26
27     Route::post('/userStore', action: [RegisteredUserController::class, 'store'])->name('user.store');
28     Route::get('/user', action: [UserController::class, 'index'])->name('user.daftarPengguna');
29     Route::get('/userRegistration', action: [UserController::class, 'create'])->name('user.registrasi');
30     Route::get('/userView/{username}', action: [UserController::class, 'showUser'])->name('user.infoPenge');
31     Route::put('/userupdate/{username}', action: [UserController::class, 'update'])->name('user.update');
32     Route::get('/userupdate/{username}', action: [UserController::class, 'edit'])->name('user.editPenggun');
33
34     Route::post('/koleksiStore', action: [KoleksiController::class, 'store'])->name('koleksi.store');
35     Route::get('/koleksi', action: [KoleksiController::class, 'index'])->name('koleksi.daftarKoleksi');
36     Route::get('/koleksiTambah', action: [KoleksiController::class, 'create'])->name('koleksi.registrasi');
37     Route::get('/koleksiView/{id}', action: [KoleksiController::class, 'show'])->name('koleksi.infoKoleksi');
38
39     Route::post('/transaksiStore', action: [TransaksiController::class, 'store'])->name('transaksi.store');
40     Route::get('/transaksi', action: [TransaksiController::class, 'index'])->name('transaksi.daftarTransaksi');
41     Route::get('/transaksiTambah', action: [TransaksiController::class, 'create'])->name('transaksi.registrasi');
42     Route::get('/transaksiView/{id}', action: [TransaksiController::class, 'show'])->name('transaksi.infoTransaksi');
43     Route::put('/transaksiupdate/{id}', action: [TransaksiController::class, 'update'])->name('transaksi.update');
44     Route::get('/transaksiupdate/{id}', action: [TransaksiController::class, 'edit'])->name('transaksi.edit');
45
46 });
47
48 // Route::get('/users', [UserController::class, 'index'])->name('users.index');
49
50 require __DIR__.'/auth.php';
51
52 // Auth::routes();
53
54 Route::get('/home', action: [App\Http\Controllers\HomeController::class, 'index'])->name('home');
```



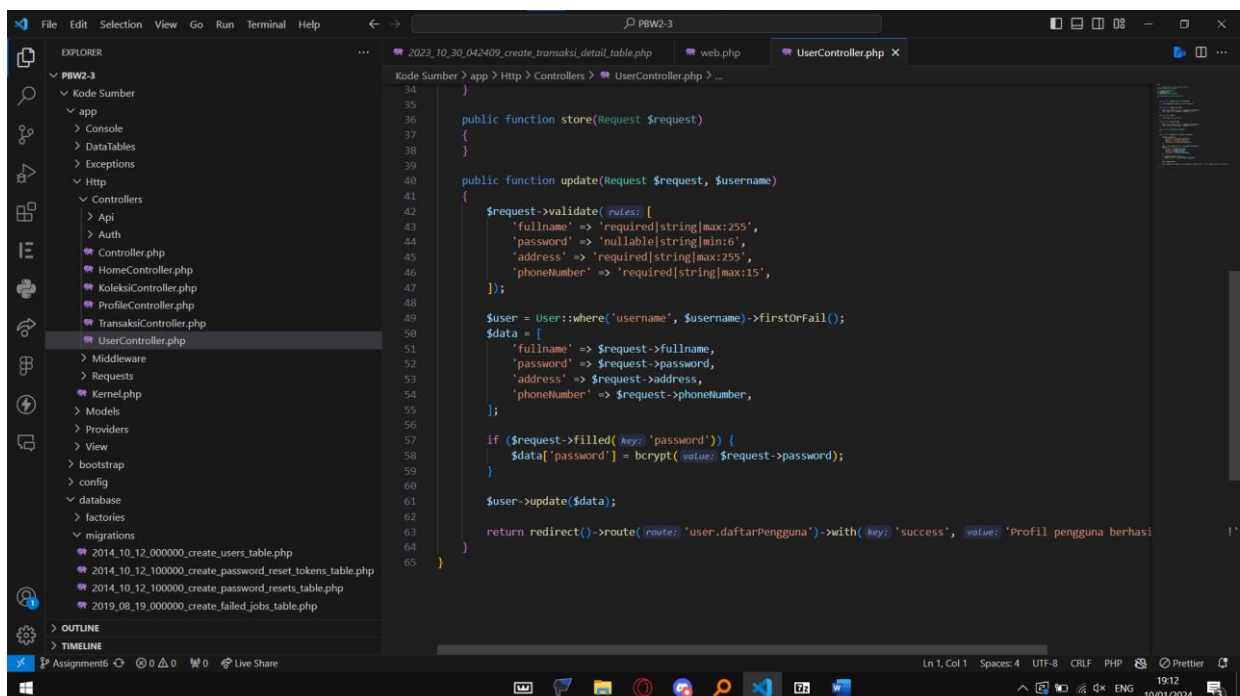
```
1 <?php
2
3 use App\Http\Controllers\ProfileController;
4 use App\Http\Controllers\UserController;
5 use App\Http\Controllers\KoleksiController;
6 use App\Http\Controllers\TransaksiController;
7 use App\Http\Controllers\Auth\RegisteredUserController;
8 use Illuminate\Support\Facades\Route;
9
10
11 //Muhammad Zidan Ernandiaz_6706223117_4604
12
13
14 Route::get('/', action: function () {
15     return view('welcome');
16 });
17
18 Route::get('/dashboard', action: function () {
19     return view('dashboard');
20 })->middleware(['auth', 'verified'])->name('dashboard');
21
22 Route::middleware(['auth'])->group(function () {
23     Route::get('/profile', action: [ProfileController::class, 'edit'])->name('profile.edit');
24     Route::patch('/profile', action: [ProfileController::class, 'update'])->name('profile.update');
25     Route::delete('/profile', action: [ProfileController::class, 'destroy'])->name('profile.destroy');
26
27     Route::post('/userStore', action: [RegisteredUserController::class, 'store'])->name('user.store');
28     Route::get('/user', action: [UserController::class, 'index'])->name('user.daftarPengguna');
29     Route::get('/userRegistration', action: [UserController::class, 'create'])->name('user.registrasi');
30     Route::get('/userView/{username}', action: [UserController::class, 'showUser'])->name('user.infoPenge');
31     Route::put('/userupdate/{username}', action: [UserController::class, 'update'])->name('user.update');
32     Route::get('/userupdate/{username}', action: [UserController::class, 'edit'])->name('user.editPenggun');
33
34     Route::post('/koleksiStore', action: [KoleksiController::class, 'store'])->name('koleksi.store');
35     Route::get('/koleksi', action: [KoleksiController::class, 'index'])->name('koleksi.daftarKoleksi');
36     Route::get('/koleksiTambah', action: [KoleksiController::class, 'create'])->name('koleksi.registrasi');
37     Route::get('/koleksiView/{id}', action: [KoleksiController::class, 'show'])->name('koleksi.infoKoleksi');
38
39     Route::post('/transaksiStore', action: [TransaksiController::class, 'store'])->name('transaksi.store');
40     Route::get('/transaksi', action: [TransaksiController::class, 'index'])->name('transaksi.daftarTransaksi');
41     Route::get('/transaksiTambah', action: [TransaksiController::class, 'create'])->name('transaksi.registrasi');
42     Route::get('/transaksiView/{id}', action: [TransaksiController::class, 'show'])->name('transaksi.infoTransaksi');
43     Route::put('/transaksiupdate/{id}', action: [TransaksiController::class, 'update'])->name('transaksi.update');
44     Route::get('/transaksiupdate/{id}', action: [TransaksiController::class, 'edit'])->name('transaksi.edit');
45
46 });
47
48 // Route::get('/users', [UserController::class, 'index'])->name('users.index');
49
50 require __DIR__.'/auth.php';
51
52 // Auth::routes();
53
54 Route::get('/home', action: [App\Http\Controllers\HomeController::class, 'index'])->name('home');
```

2. Kelas controller

- UserController



```
1 <?php
2
3 //Muhammad Zidan Ernandiaz_6706223117_4604
4 namespace App\Http\Controllers;
5
6 use Illuminate\Http\Request;
7 use App\Models\User;
8 use App\DataTables\UserDataTable;
9
10 class UserController extends Controller
11 {
12
13     public function index(UserDataTable $dataTable)
14     {
15         return $dataTable->render('user.daftarPengguna');
16     }
17
18     public function showUser($username)
19     {
20         $user = User::where('username', $username)->firstOrFail();
21         return view('user.infoPengguna', ['data' => compact('user')]);
22     }
23
24     public function create()
25     {
26         return view('user.registrasi');
27     }
28
29     public function edit($username)
30     {
31         $user = User::where('username', $username)->firstOrFail();
32         return view('user.editPengguna', ['data' => compact('user')]);
33     }
34
35     public function store(Request $request)
36     {
37
38     }
```



```
34
35
36     public function store(Request $request)
37     {
38
39
40     public function update(Request $request, $username)
41     {
42         $request->validate([
43             'fullname' => 'required|string|max:255',
44             'password' => 'nullable|string|min:6',
45             'address' => 'required|string|max:255',
46             'phoneNumber' => 'required|string|max:15',
47         ]);
48
49         $user = User::where('username', $username)->firstOrFail();
50         $data = [
51             'fullname' => $request->fullname,
52             'password' => $request->password,
53             'address' => $request->address,
54             'phoneNumber' => $request->phoneNumber,
55         ];
56
57         if ($request->filled('password')) {
58             $data['password'] = bcrypt($request->password);
59         }
60
61         $user->update($data);
62
63         return redirect()->route('user.daftarPengguna')->with('success', 'Profil pengguna berhasil');
64     }
65 }
```

- CollectionController

```

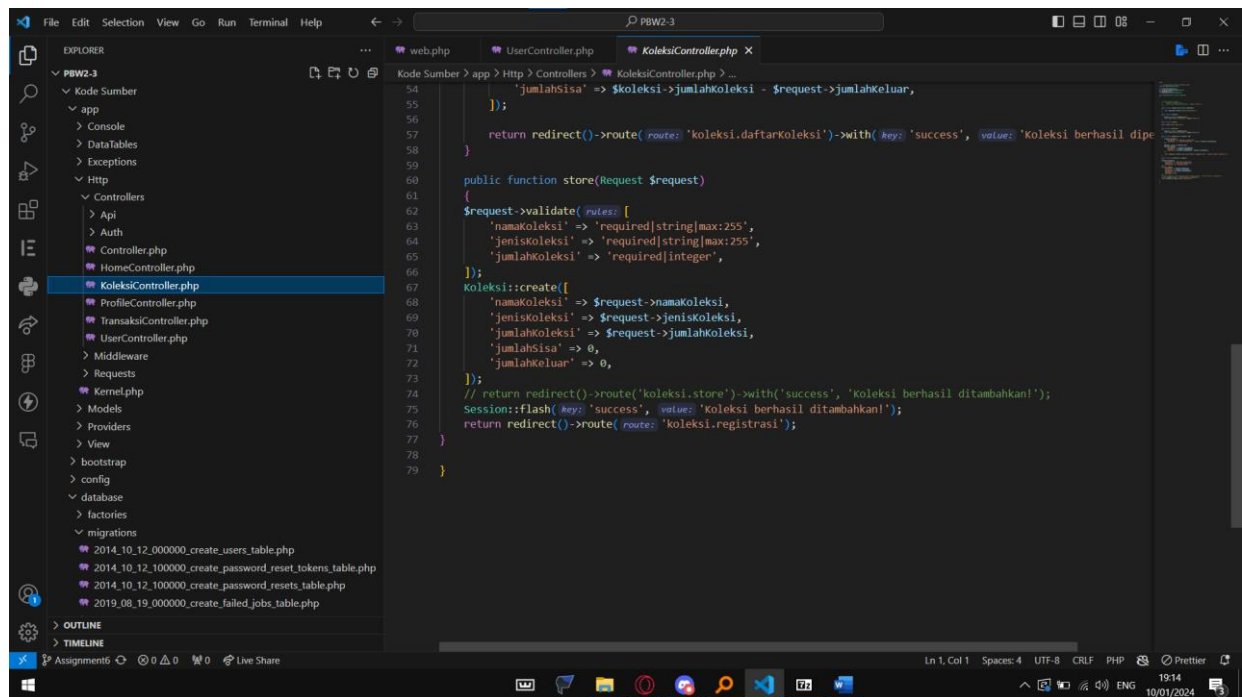
1 <?php
2
3 //Muhammad Zidan Ernandiaz_6706223117_4604
4 namespace App\Http\Controllers;
5
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\Session;
8 use App\Models\Koleksi;
9 use App\DataTables\KoleksiDataTable;
10
11 class KoleksiController extends Controller
12 {
13
14
15
16 // public function index() {
17 //     $koleksi = Koleksi::all();
18 //     return view('koleksi.daftarKoleksi', compact('koleksi'));
19 // }
20
21 public function index(KoleksiDataTable $dataTable)
22 {
23     return $dataTable->render( view: 'koleksi.daftarKoleksi');
24 }
25
26 public function show($id)
27 {
28     $koleksi = Koleksi::findOrFail($id);
29     return view( view: 'koleksi.infokoleksi', data: compact( var_name: 'koleksi'));
30 }
31
32 public function create()
33 {
34     return view( view: 'koleksi.registrasi');
35 }
36
37 public function edit($id)
38 {
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
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
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

```

25
26 public function show($id)
27 {
28     $koleksi = Koleksi::findOrFail($id);
29     return view( view: 'koleksi.infokoleksi', data: compact( var_name: 'koleksi'));
30 }
31
32 public function create()
33 {
34     return view( view: 'koleksi.registrasi');
35 }
36
37 public function edit($id)
38 {
39     $koleksi = Koleksi::findOrFail($id);
40     return view( view: 'koleksi.editkoleksi', data: compact( var_name: 'koleksi'));
41 }
42
43 public function update(Request $request, $id)
44 {
45     $request->validate( rules: [
46         'jeniskoleksi' => 'required|string|max:255',
47         'jumlahkeluar' => 'required|integer|max:'. Koleksi::find($id)->jumlahKoleksi,
48     ]);
49     $koleksi = Koleksi::findOrFail($id);
50     $koleksi->update([
51         'jeniskoleksi' => $request->jeniskoleksi,
52         'jumlahkeluar' => $request->jumlahKeluar,
53         'jumlahsiswa' => $koleksi->jumlahKoleksi - $request->jumlahKeluar,
54     ]);
55     return redirect()->route( route: 'koleksi.daftarKoleksi'->with( key: 'success', value: 'koleksi berhasil dipe
56 }
57
58 public function store(Request $request)
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
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

- TransactionController

```

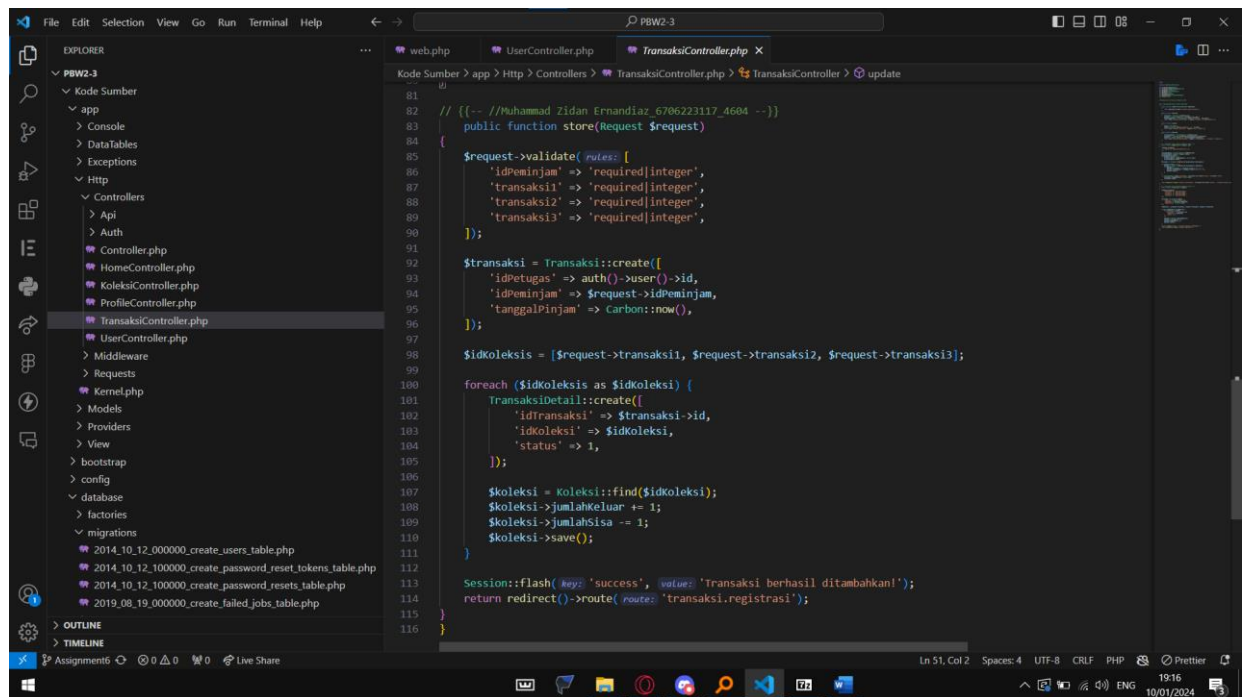
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\Session;
7 use App\Models\Transaksi;
8 use App\Models\TransaksiDetail;
9 use App\Models\User;
10 use App\Models\Koleksi;
11 use App\DataTables\TransaksiDataTable;
12 use Carbon\Carbon;
13
14 //Muhammad Zidan Ernandiaz_6706223117_4604
15
16
17 class TransaksiController extends Controller
18 {
19     public function index(TransaksiDataTable $dataTable)
20     {
21         return $dataTable->render( view: 'transaksi.daftarTransaksi');
22     }
23
24     public function show($id)
25     {
26         $transaksi = Transaksi::findOrFail($id);
27         $koleksi = Koleksi::where('id', $transaksi->id)->get();
28         $transaksiDetails = TransaksiDetail::where('idTransaksi', $id)->get();
29         return view( view: 'transaksi.infoTransaksi', data: compact( 'var_names: 'transaksi', 'var_names[0]': 'transaksiDetail', 'var_names[1]': 'transaksiDetail' ));
30     }
31
32     public function create()
33     {
34         $users = User::all();
35         $koleksi = Koleksi::where('jumlahSisa', '>=', 3)->get();
36         return view( view: 'transaksi.registrasi', data: compact( 'var_names: 'users', 'var_names[0]': 'koleksi' ));
37     }
38 }

```

```

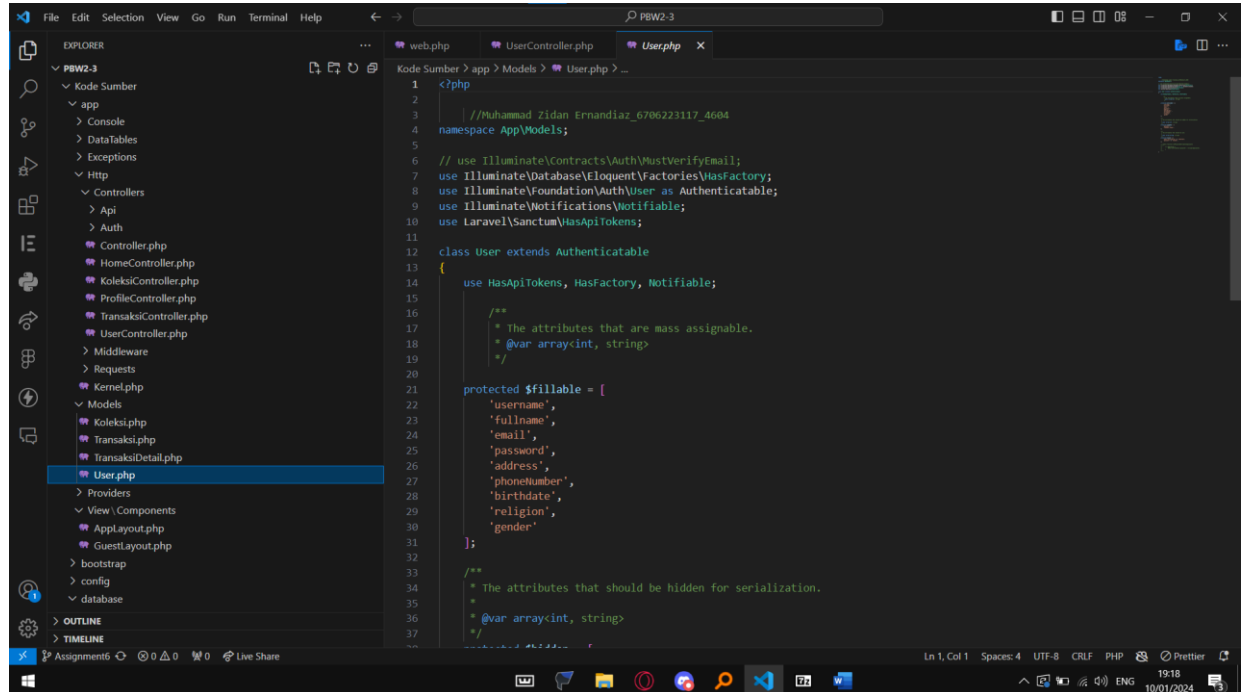
38 }
39
40 public function edit($id)
41 {
42     $transaksiDetail = TransaksiDetail::findOrFail($id);
43     $transaksi = Transaksi::findOrFail($transaksiDetail->idTransaksi);
44     $koleksi = Koleksi::where('id', $transaksiDetail->idKoleksi);
45     return view( view: 'transaksi.editTransaksi', data: compact( 'var_names: 'transaksiDetail', 'var_names[0]': 'transaksiDetail', 'var_names[1]': 'transaksiDetail' ));
46 }
47
48 // (-- //Muhammad Zidan Ernandiaz_6706223117_4604 --)
49 public function update(Request $request, $id)
50 {
51     $request->validate( rules: [
52         'status' => 'required|integer|in:1,2,3',
53     ]);
54
55     $transaksiDetail = TransaksiDetail::findOrFail($id);
56     $transaksiDetail->status = $request->status;
57     $transaksiDetail->save();
58     if ($request->status != 1) {
59         $transaksiDetail->tanggalKembali = Carbon::now();
60         $transaksiDetail->save();
61     }
62     $transaksi = Transaksi::findOrFail($transaksiDetail->idTransaksi);
63
64     if ($request->status != 1) {
65         $koleksi = Koleksi::findOrFail($transaksiDetail->idKoleksi);
66         if ($koleksi) {
67             $koleksi->jumlahSisa += ($request->status == 1) ? -1 : 1;
68             $koleksi->jumlahKembalian += ($request->status == 1) ? 1 : -1;
69             $koleksi->save();
70         }
71     }
72
73     if (TransaksiDetail::where('idTransaksi', $transaksi->id)->where('status', 1)->count() == 0) {
74

```



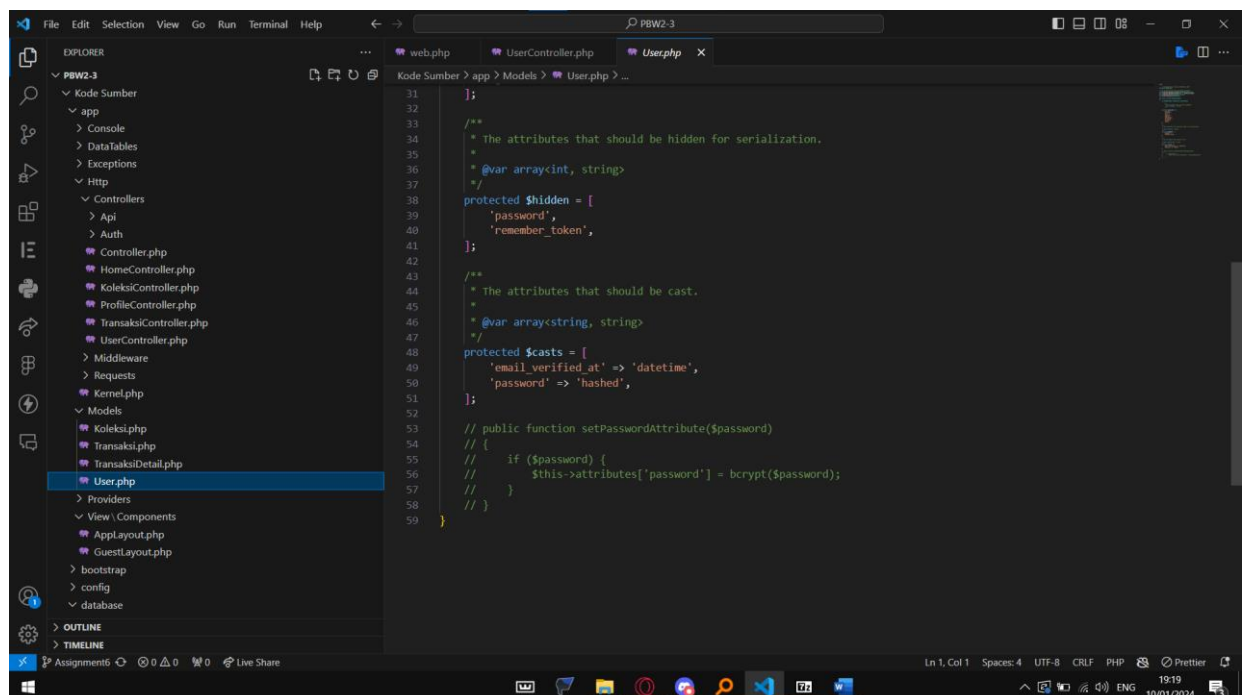
3. Kelas model

- User



The screenshot shows the Visual Studio Code editor with a project named 'PBW2-3'. The Explorer panel on the left shows the file structure, with 'User.php' selected under 'Models'. The main editor displays the code for 'User.php'.

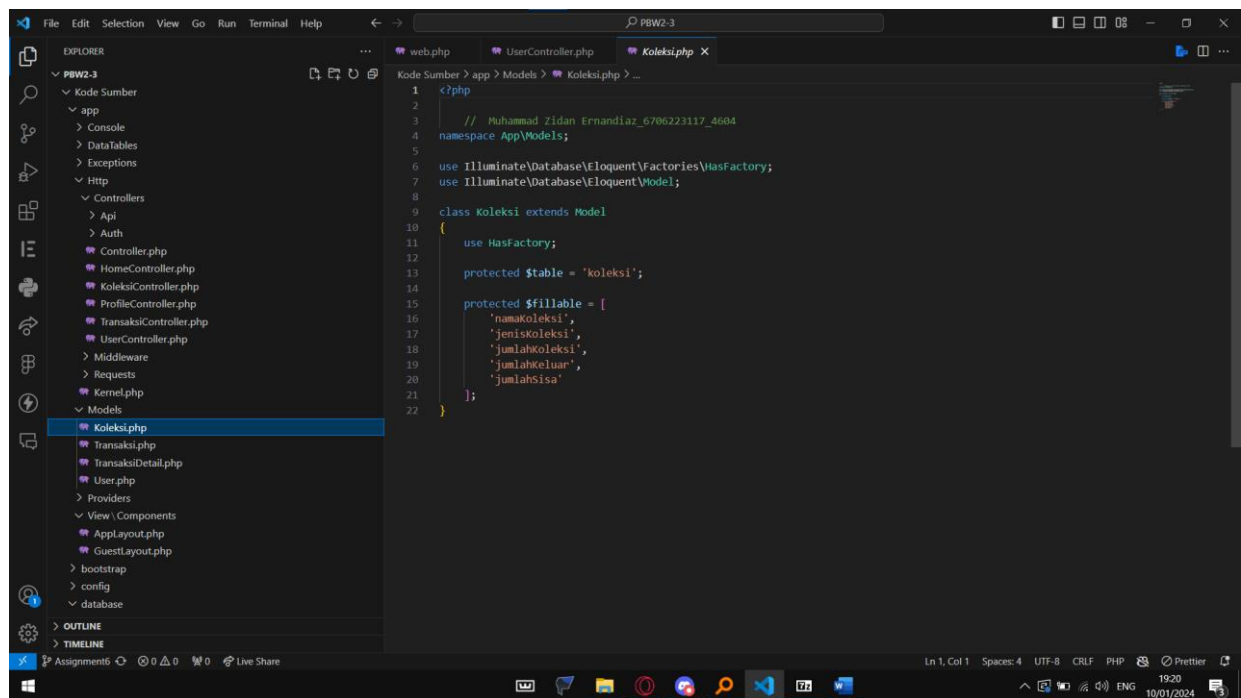
```
1 <?php
2
3 //Muhammad Zidan Ernandiaz_6706223117_4604
4 namespace App\Models;
5
6 // use Illuminate\Contracts\Auth\MustVerifyEmail;
7 use Illuminate\Database\Eloquent\Factories\HasFactory;
8 use Illuminate\Foundation\Auth\User as Authenticatable;
9 use Illuminate\Notifications\Notifiable;
10 use Laravel\Sanctum\HasApiTokens;
11
12 class User extends Authenticatable
13 {
14     use HasApiTokens, HasFactory, Notifiable;
15
16     /**
17      * The attributes that are mass assignable.
18      * @var array<int, string>
19      */
20
21     protected $fillable = [
22         'username',
23         'fullname',
24         'email',
25         'password',
26         'address',
27         'phoneNumber',
28         'birthdate',
29         'religion',
30         'gender'
31     ];
32
33     /**
34      * The attributes that should be hidden for serialization.
35      *
36      * @var array<int, string>
37      */
```



This screenshot continues the code from the previous one, showing the bottom part of the 'User.php' model.

```
31     ];
32
33     /**
34      * The attributes that should be hidden for serialization.
35      *
36      * @var array<int, string>
37      */
38     protected $hidden = [
39         'password',
40         'remember_token',
41     ];
42
43     /**
44      * The attributes that should be cast.
45      *
46      * @var array<string, string>
47      */
48     protected $casts = [
49         'email_verified_at' => 'datetime',
50         'password' => 'hashed',
51     ];
52
53     // public function setPasswordAttribute($password)
54     // {
55     //     if ($password) {
56     //         $this->attributes['password'] = bcrypt($password);
57     //     }
58     // }
59 }
```

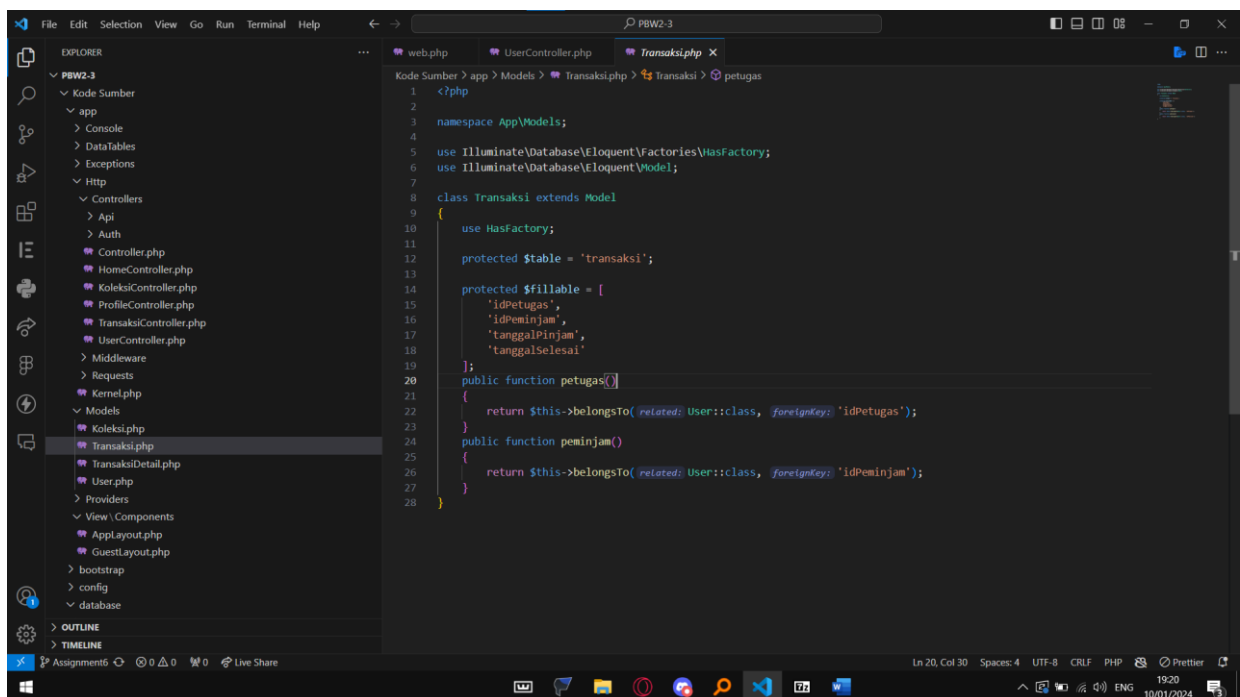
- Collection



The screenshot shows the Visual Studio Code editor with the file explorer on the left. The file explorer is expanded to show the 'Models' directory, where 'Koleksi.php' is selected. The main editor window displays the code for 'Koleksi.php'. The code is a PHP class that extends the 'Model' class from the 'Illuminate\Database\Eloquent\Factories' namespace. It defines a protected static property '\$table' with the value 'koleksi' and a protected static property '\$fillable' with an array of field names: 'namaKoleksi', 'jenisKoleksi', 'jumlahKoleksi', 'jumlahKeluar', and 'jumlahSisa'.

```
1 <?php
2
3 // Muhammad Zidan Ernandiaz_6706223117_4604
4 namespace App\Models;
5
6 use Illuminate\Database\Eloquent\Factories\HasFactory;
7 use Illuminate\Database\Eloquent\Model;
8
9 class Koleksi extends Model
10 {
11     use HasFactory;
12
13     protected $table = 'koleksi';
14
15     protected $fillable = [
16         'namaKoleksi',
17         'jenisKoleksi',
18         'jumlahKoleksi',
19         'jumlahKeluar',
20         'jumlahSisa'
21     ];
22 }
```

- Transaction



The screenshot shows the Visual Studio Code editor with the file explorer on the left. The file explorer is expanded to show the 'Models' directory, where 'Transaksi.php' is selected. The main editor window displays the code for 'Transaksi.php'. The code is a PHP class that extends the 'Model' class from the 'Illuminate\Database\Eloquent\Factories' namespace. It defines a protected static property '\$table' with the value 'transaksi' and a protected static property '\$fillable' with an array of field names: 'idPetugas', 'idPeminjam', 'tanggalPinjam', and 'tanggalSelesai'. It also defines two public methods: 'petugas()' and 'peminjam()', both of which return the result of the 'belongsTo' method call.

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Transaksi extends Model
9 {
10     use HasFactory;
11
12     protected $table = 'transaksi';
13
14     protected $fillable = [
15         'idPetugas',
16         'idPeminjam',
17         'tanggalPinjam',
18         'tanggalSelesai'
19     ];
20
21     public function petugas()
22     {
23         return $this->belongsTo('related: User::class, foreignKey: 'idPetugas');
24     }
25
26     public function peminjam()
27     {
28         return $this->belongsTo('related: User::class, foreignKey: 'idPeminjam');
29     }
30 }
```

- DetailTransaction

The screenshot shows the Visual Studio Code editor with the file explorer on the left. The file explorer shows the project structure for 'PBW2-3', including 'Kode Sumber', 'app', 'Console', 'DataTables', 'Exceptions', 'Http', 'Controllers', 'Middleware', 'Requests', 'Kernel.php', 'Models', 'TransaksiDetail.php', 'User.php', 'Providers', 'View Components', 'AppLayout.php', 'GuestLayout.php', 'bootstrap', 'config', and 'database'. The 'TransaksiDetail.php' file is selected in the 'Models' directory. The main editor window displays the PHP code for the 'TransaksiDetail' model, which extends 'Illuminate\Database\Eloquent\Model'. The code includes a 'protected \$table' property set to 'transaksi_detail', a 'protected \$fillable' array with fields 'id', 'id_transaksi', 'id_koleksi', 'tanggal_kembali', 'status', and 'keterangan', and two public methods: 'koleksi()' and 'transaksi()'. The 'koleksi()' method returns a belongsTo relationship with 'Koleksi' class and 'id_koleksi' as the foreign key. The 'transaksi()' method returns a belongsTo relationship with 'Transaksi' class and 'id_transaksi' as the foreign key.

```

1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class TransaksiDetail extends Model
9 {
10     use HasFactory;
11
12     protected $table = 'transaksi_detail';
13
14     protected $fillable = [
15         'id',
16         'id_transaksi',
17         'id_koleksi',
18         'tanggal_kembali',
19         'status',
20         'keterangan'
21     ];
22     public function koleksi()
23     {
24         return $this->belongsTo(related: Koleksi::class, foreignKey: 'idKoleksi');
25     }
26     public function transaksi()
27     {
28         return $this->belongsTo(related: Transaksi::class, foreignKey: 'idTransaksi');
29     }
30 }
31

```

4. Kelas blade View

- Transaksi
- daftarTransaksi.blade.php

The screenshot shows the Visual Studio Code editor with the file explorer on the left. The file explorer shows the project structure for 'PBW2-3', including 'node_modules', 'public', 'resources', 'css', 'js', 'sass', 'views', 'auth', 'components', 'koleksi', 'layouts', 'profile', 'transaksi', 'actionTransaksi.blade.php', 'daftarTransaksi.blade.php', 'editTransaksi.blade.php', 'infoTransaksi.blade.php', 'registrasi.blade.php', 'user', 'dashboard.blade.php', 'home.blade.php', 'welcome.blade.php', 'routes', 'api.php', 'auth.php', 'channels.php', 'console.php', 'web.php', 'storage', 'tests', 'OUTLINE', and 'TIMELINE'. The 'daftarTransaksi.blade.php' file is selected in the 'views' directory. The main editor window displays the Blade template code for the 'daftarTransaksi.blade.php' file. The code includes a header section with a title 'Daftar Transaksi', a success message alert, a table of transactions, and a script to handle the 'view-detail' action. The table has columns for 'id', 'id_transaksi', 'id_koleksi', 'tanggal_kembali', 'status', and 'keterangan'. The script uses jQuery to handle the click event on the 'view-detail' link, redirecting to the 'infoTransaksi' route with the transaction ID as a parameter.

```

1 <x-app-layout>
2     <x-slot name="header">
3         <h2 class="font-semibold text-xl text-gray-800 leading-tight">
4             {{ __('Daftar Transaksi') }}
5         </h2>
6     </x-slot>
7     @if(Session::has('success'))
8         <div class="alert alert-success">
9             {{ Session::get('success') }}
10        </div>
11    @endif
12    @if($dataTable->table())
13        <div class="card">
14            <div class="card-body">
15                {{ $dataTable->table() }}
16            </div>
17        </div>
18    @endif
19    @push('scripts')
20        {{ $dataTable->scripts(attributes: ['type' => 'module']) }}
21    @endpush
22    <script>
23        $(document).on('click', '.view-detail', function () {
24            var id = $(this).data('id');
25            // Redirect ke rute infoTransaksi dengan parameter ID
26            window.location.href = "{{ route('transaksi.infoTransaksi', '') }}" + id;
27        });
28    </script>
29 @endpush
30 </x-app-layout>
31

```

- registrasi.blade.php

```

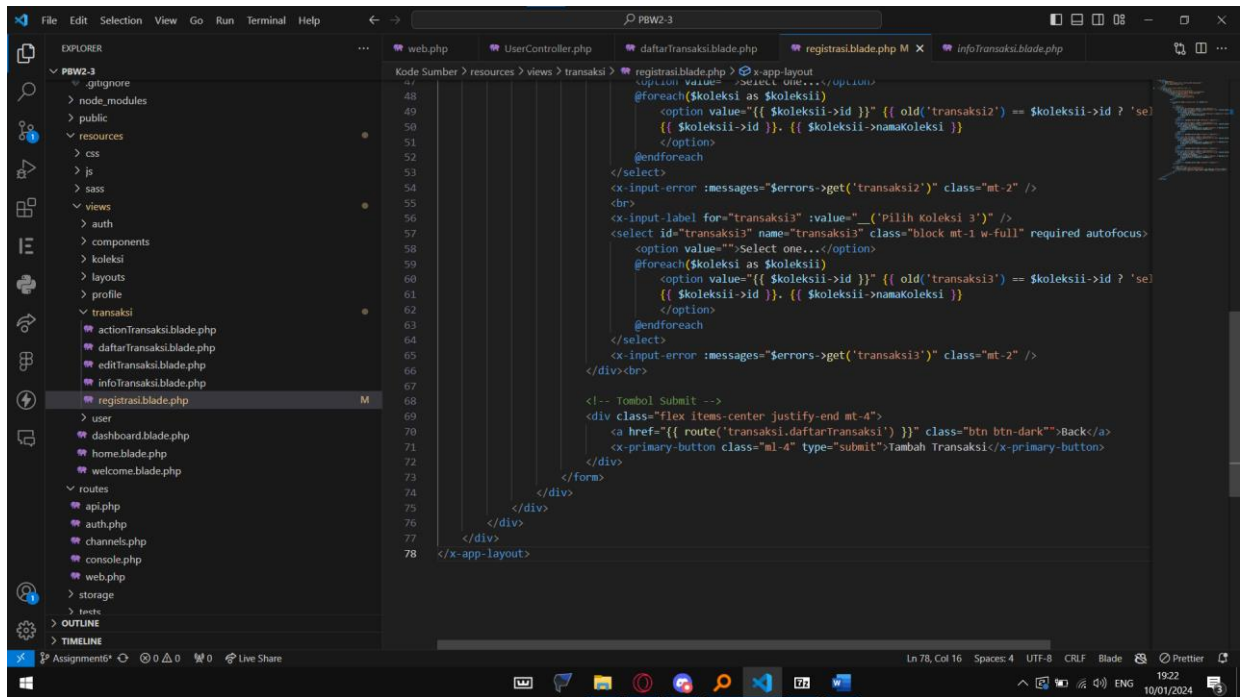
1 <@app-layout>
2 <@slot name="header">
3 <h2 class="font-semibold text-xl text-gray-800 leading-tight">
4 <@('Tambah Transaksi')>
5 </h2>
6 </@slot>
7 <!-- //Muhammad Zidan Ernandiaz_6706223117_4604 -->
8 <div class="py-12">
9 <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
10 <div class="bg-white overflow-hidden shadow-sm sm:rounded-lg">
11 <div class="p-6 bg-white border-b border-gray-200">
12 @if(Session::has('success'))
13 <div class="alert alert-success">
14 <@('Session::get('success'))>
15 </div>
16 @endif
17 <form action="{{ route('transaksi.store') }}" method="POST">
18 @csrf
19 <!-- Peminjam -->
20 <div class="mt-4">
21 <input-label for="idPeminjam" value="{{ __('Pilih Peminjam') }}" />
22 <select id="idPeminjam" name="idPeminjam" class="block mt-1 w-full required autofocus">
23 <option value="">Select one...</option>
24 @foreach($users as $user)
25 <option value="{{ $user->id }}" {{ old('idPeminjam') == $user->id ? 'selected' :
26 <(' $user->id') }} {{ $user->fullname }}>
27 </option>
28 @endforeach
29 </select>
30 <input-error :messages="$errors->get('idPeminjam')> class="mt-2" />
31 <br>
32 <input-label for="transaksi1" value="{{ __('Pilih Koleksi 1') }}" />
33 <select id="transaksi1" name="transaksi1" class="block mt-1 w-full required autofocus">
34 <option value="">Select one...</option>
35 @foreach($koleksi as $koleksi)
36 <option value="{{ $koleksi->id }}" {{ old('transaksi1') == $koleksi->id ? 'selected' :
37 <(' $koleksi->id') }} {{ $koleksi->namaKoleksi }}>
38 </option>
39 @endforeach
40 </select>
41 <input-error :messages="$errors->get('transaksi1')> class="mt-2" />
42 <br>
43 <input-label for="transaksi2" value="{{ __('Pilih Koleksi 2') }}" />
44 <select id="transaksi2" name="transaksi2" class="block mt-1 w-full required autofocus">
45 <option value="">Select one...</option>
46 @foreach($koleksi as $koleksi)
47 <option value="{{ $koleksi->id }}" {{ old('transaksi2') == $koleksi->id ? 'selected' :
48 <(' $koleksi->id') }} {{ $koleksi->namaKoleksi }}>
49 </option>
50 @endforeach
51 </select>
52 <input-error :messages="$errors->get('transaksi2')> class="mt-2" />
53 <br>
54 <input-label for="transaksi3" value="{{ __('Pilih Koleksi 3') }}" />
55 <select id="transaksi3" name="transaksi3" class="block mt-1 w-full required autofocus">
56 <option value="">Select one...</option>
57 @foreach($koleksi as $koleksi)
58 <option value="{{ $koleksi->id }}" {{ old('transaksi3') == $koleksi->id ? 'selected' :
59 <(' $koleksi->id') }} {{ $koleksi->namaKoleksi }}>
60 </option>
61 @endforeach
62 </select>
63 <input-error :messages="$errors->get('transaksi3')> class="mt-2" />
64 </div><br>
65 </div>
66 </div>
67 </div>

```

```

31 </select>
32 <input-error :messages="$errors->get('idPeminjam')> class="mt-2" />
33 <br>
34 <input-label for="transaksi1" value="{{ __('Pilih Koleksi 1') }}" />
35 <select id="transaksi1" name="transaksi1" class="block mt-1 w-full required autofocus">
36 <option value="">Select one...</option>
37 @foreach($koleksi as $koleksi)
38 <option value="{{ $koleksi->id }}" {{ old('transaksi1') == $koleksi->id ? 'selected' :
39 <(' $koleksi->id') }} {{ $koleksi->namaKoleksi }}>
40 </option>
41 @endforeach
42 </select>
43 <input-error :messages="$errors->get('transaksi1')> class="mt-2" />
44 <br>
45 <input-label for="transaksi2" value="{{ __('Pilih Koleksi 2') }}" />
46 <select id="transaksi2" name="transaksi2" class="block mt-1 w-full required autofocus">
47 <option value="">Select one...</option>
48 @foreach($koleksi as $koleksi)
49 <option value="{{ $koleksi->id }}" {{ old('transaksi2') == $koleksi->id ? 'selected' :
50 <(' $koleksi->id') }} {{ $koleksi->namaKoleksi }}>
51 </option>
52 @endforeach
53 </select>
54 <input-error :messages="$errors->get('transaksi2')> class="mt-2" />
55 <br>
56 <input-label for="transaksi3" value="{{ __('Pilih Koleksi 3') }}" />
57 <select id="transaksi3" name="transaksi3" class="block mt-1 w-full required autofocus">
58 <option value="">Select one...</option>
59 @foreach($koleksi as $koleksi)
60 <option value="{{ $koleksi->id }}" {{ old('transaksi3') == $koleksi->id ? 'selected' :
61 <(' $koleksi->id') }} {{ $koleksi->namaKoleksi }}>
62 </option>
63 @endforeach
64 </select>
65 <input-error :messages="$errors->get('transaksi3')> class="mt-2" />
66 </div><br>
67 </div>

```



- infoTransaksi.blade.php

```

1 <x-app-layout>
2   <x-slot name="header">
3     <h2 class="font-semibold text-xl text-gray-800 leading-tight">
4       {{ __('Informasi Transaksi') }}
5     </h2>
6   </x-slot>
7   @if($transaction->success)
8     <div class="container">
9       @if(Session::has('success'))
10        <div class="alert alert-success">
11          {{ Session::get('success') }}
12        </div>
13      @endif
14    </div>
15    <div class="py-12">
16      <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
17        <div class="bg-white overflow-hidden shadow-sm sm:rounded-lg">
18          <div class="p-6">
19            <a href="#" class="btn btn-dark onclick="goBack()">Back</a><br><br>
20
21            @php
22              $petugas = \App\Models\User::find($transaction->idPetugas);
23              $peminjam = \App\Models\User::find($transaction->idPeminjam);
24            @endphp
25
26            <p>Petugas: {{ $petugas->fullname }}</p>
27            <p>Peminjam: {{ $peminjam->fullname }}</p>
28            <p>Tanggal Pinjam: {{ $transaction->tanggalPinjam }}</p>
29            @if ($transaction->tanggalSelesai != null)
30              <p>Tanggal Selesai: {{ $transaction->tanggalSelesai }}</p>
31            @endif
32          </div>
33        </div>
34      </div>
35    </div>
36  </x-app-layout>

```

```

15 </div>
16 <div class="py-12">
17   <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
18     <div class="bg-white overflow-hidden shadow-sm sm:rounded-lg">
19       <div class="p-6">
20         <a href="#" class="btn btn-dark onclick="goBack()">Back</a><br><br>
21
22         @php
23           $petugas = \App\Models\User::find($transaction->idPetugas);
24           $peminjam = \App\Models\User::find($transaction->idPeminjam);
25         @endphp
26
27         <p>Petugas: {{ $petugas->fullname }}</p>
28         <p>Peminjam: {{ $peminjam->fullname }}</p>
29         <p>Tanggal Pinjam: {{ $transaction->tanggalPinjam }}</p>
30         @if ($transaction->tanggalSelesai != null)
31           <p>Tanggal Selesai: {{ $transaction->tanggalSelesai }}</p>
32         @endif
33       </div>
34     </div>
35   </div>
36 </div>
37 <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
38   <div class="bg-white overflow-hidden shadow-sm sm:rounded-lg">
39     <div class="p-6 text-gray-900">
40       <table class="min-w-full table-auto">
41         <thead>
42           <tr class="bg-gray-100">
43             <th class="border px-4 py-2">No</th>
44             <th class="border px-4 py-2">Nama Koleksi</th>
45             <th class="border px-4 py-2">Tanggal Kembali</th>
46             <th class="border px-4 py-2">Status</th>
47             @if ($transaction->tanggalSelesai == null)
48               <th class="border px-4 py-2">Action</th>
49             @endif
50           </tr>
51         </thead>

```

