

# **LAPORAN PRAKTIKUM PENGENALAN APLIKASI BERBASIS INTERNET**

## **CRUD LARAVEL**



**Agus Pranata Marpaung**

**13323033**

**DIII TEKNOLOGI KOMPUTER**

**INSTITUT TEKNOLOGI DEL  
FAKULTAS VOKASI**

## Judul Praktikum

---

<b>Minggu/Sesi</b>	:	XII/3
<b>Kode Mata Kuliah</b>	:	1331204
<b>Nama Mata Kuliah</b>	:	Pengenalan Aplikasi Berbasis Internet
<b>Setoran</b>	:	Softcopy
<b>Batas Waktu Setoran</b>	:	23 April 2024 jam 18:00
<b>Tujuan</b>	:	1. Mahasiswa dapat melakukan praktikum mengenai CRUD pada laravel 11

## Framework Laravel

Laravel merupakan salah satu framework Laravel yang bersifat opensource. Laravel dirancang agar proses pengembangan aplikasi web menjadi lebih mudah dan cepat karena berbagai fitur bawaan didalamnya. Fitur-fitur tersebut merupakan salah satu alasan mengapa banyak pengembangan memanfaatkan kerangka kerja Laravel yang menggunakan konsep Model- ViewController (MVC). Laravel berada di bawah lisensi MIT, dengan menggunakan GitHub sebagai tempat berbagi kode. Pada saat ini pengembangan Laravel berada di versi 11 dan baru dirilis pada 12 Maret 2024 dan hanya mendukung versi php dengan minimum 8.2 ke atas.

Dengan menggunakan pattern MVC ini, struktur kode yang dihasilkan menjadi lebih terstruktur dan memiliki standar yang jelas dengan memisahkan bisnis logic (alur pikir), data logic (penyimpanan data) dan presentation logic (antarmuka aplikasi) atau secara sederhana adalah memisahkan antara desain, data dan proses dimana komponen dari MVC antara lain adalah:

### 1. Model

Model berhubungan dengan data dan interaksi ke database atau webservice. Model juga merepresentasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk file teks, file XML maupun webservice. Biasanya di dalam model akan berisi class dan fungsi untuk mengambil, melakukan update dan menghapus data website. Sebuah aplikasi web biasanya menggunakan basis data dalam menyimpan data, maka pada bagian Model biasanya akan berhubungan dengan perintah-perintah query SQL

### 2. View

View berhubungan dengan segala sesuatu yang akan ditampilkan ke end-user. Bisa berupa halaman web, rss, javascript dan lain-lain. Kita harus menghindari adanya logika atau pemrosesan data di view. Di dalam view hanya berisi variabel-variabel yang berisi data yang siap ditampilkan. View dapat dikatakan sebagai halaman website yang dibuat dengan menggunakan HTML dan bantuan CSS atau JavaScript. Di dalam view jangan pernah ada kode untuk melakukan koneksi ke basisdata. View hanya dikhususkan untuk menampilkan data-data hasil dari model dan controller.

### 3. Controller

Controller bertindak sebagai penghubung data dan view. Di dalam Controller inilah terdapat class- class dan fungsi-fungsi yang memproses permintaan dari View ke dalam struktur data di dalam Model. Controller juga tidak boleh berisi kode untuk mengakses basis data karena tugas mengakses data telah diserahkan kepada model. Tugas controller adalah menyediakan berbagai variabel yang akan ditampilkan di view, memanggil model untuk melakukan akses ke basis data, menyediakan penanganan kesalahan/error, mengerjakan proses logika dari aplikasi serta melakukan validasi atau cek terhadap input.

## 1. Membuat CRUD dengan Framework Laravel

1. Buatlah folder baru dengan nama **Laravel**, Kemudian lakukan instalasi laravel dengan nama project **Barang** dengan menggunakan composer. Jika instalasi telah selesai, maka folder yang akan dihasilkan akan seperti berikut:

Name	Date modified	Type	Size
app	09/04/2024 9:13 PM	File folder	
bootstrap	09/04/2024 9:13 PM	File folder	
config	09/04/2024 9:13 PM	File folder	
database	19/04/2024 8:28 AM	File folder	
public	09/04/2024 9:13 PM	File folder	
resources	09/04/2024 9:13 PM	File folder	
routes	09/04/2024 9:13 PM	File folder	
storage	09/04/2024 9:13 PM	File folder	
tests	09/04/2024 9:13 PM	File folder	
vendor	19/04/2024 8:28 AM	File folder	
.editorconfig	09/04/2024 9:13 PM	Editor Config Sour...	1 KB
.env	19/04/2024 8:30 AM	ENV File	2 KB
.env.example	09/04/2024 9:13 PM	EXAMPLE File	2 KB
.gitattributes	09/04/2024 9:13 PM	Git Attributes Sour...	1 KB
.gitignore	09/04/2024 9:13 PM	Git Ignore Source ...	1 KB
artisan	09/04/2024 9:13 PM	File	1 KB
composer	09/04/2024 9:13 PM	JSON Source File	2 KB
composer.lock	19/04/2024 8:27 AM	LOCK File	292 KB
package	09/04/2024 9:13 PM	JSON Source File	1 KB

2. Kemudian aktifkan **Apache** dan **MySQL** melalui XAMPP.
3. Lakukan konfigurasi koneksi antara **MySQL** dengan laravel 11. Silahkan Anda membuka file **.env** lalu cari code seperti dibawah ini.

```
DB_CONNECTION=sqlite
# DB_HOST=127.0.0.1
# DB_PORT=3306
# DB_DATABASE=laravel
# DB_USERNAME=root
# DB_PASSWORD=
```

Setelah Anda menemukan codenya, silahkan Anda ubah code menjadi seperti code dibawah ini

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

Pada code diatas kita pertama kali mengubah konfigurasi DB\_CONNECTION yang semula menggunakan sqlite menjadi mysql. Lalu kita hapus semua tanda # pada konfigurasi key agar variable nya diaktifkan. Lalu untuk DB\_DATABASE kita atur sesuai dengan database yang kita buat, disini kita buat nama databasenya yaitu db\_barang\_itdel.

4. Selanjutnya buat model dan migration pada project dengan menjalankan perintah dibawah di dalam terminal/CMD Anda.

```
C:\Laravel\Barang>php artisan make:model Barang -m
```

Jika berhasil maka pada project Anda akan bertambah 2 file baru yang berada di dalam folder yaitu:

- app/Models/Barang.php
- database/migrations/2024\_04\_17\_025054\_create\_barangs\_table.php

```
INFO Model [C:\Laravel\Barang\app\Models\Barang.php] created successfully.
INFO Migration [C:\Laravel\Barang\database\migrations\2024_04_19_014325_create_barangs_table.php] created successfully.
```

5. Tambahkan field/kolom di dalam migration dengan membuka file migration lalu ketikkan code dibawah.

```
2024_04_19_014325_create_barangs_table.php X
database > migrations > 2024_04_19_014325_create_barangs_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('barang', function (Blueprint $table) {
15             $table->id();
16             $table->string('nama_barang');
17             $table->integer('stok')->default(0);
18             $table->string('status');
19             $table->string('image');
20             $table->timestamps();
21         });
22     }
23
24     /**
25      * Reverse the migrations.
26      */
27     public function down(): void
28     {
29         Schema::dropIfExists('barang');
30     }
31 };
32
```

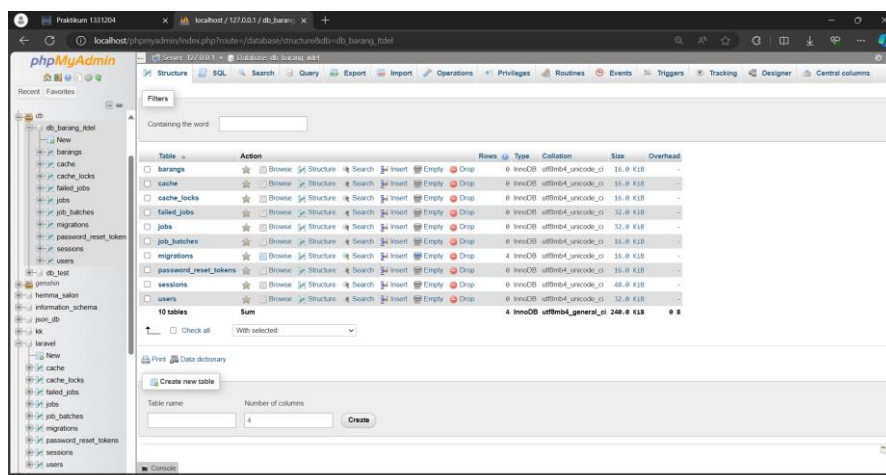
6. Tambahkan Mass Assignment agar field yang telah ditambahkan di dalam migration dapat Melakukan manipulasi ke dalam database. Silahkan buka file Barang.php yang ada pada folder app/Models/Barang.php lalu ketikkan code berikut.

```
Barang.php X
app > Models > Barang.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Barang extends Model
9  {
10     use HasFactory;
11     /**
12      * fillable
13      *
14      * @var array
15      */
16     protected $fillable = [
17         'namabarang',
18         'stok',
19         'status',
20         'image',
21     ];
22 }
23
```

7. Jalankan proses migrate dalam project laravel, proses ini akan men-generate database dan table beserta field/kolom di dalam database. Silahkan jalankan perintah berikut di terminal/CMD Anda.

```
C:\Laravel\Barang>php artisan migrate
```

Jika keluar pertanyaan, silahkan pilih YES lalu enter dan tunggu prosesnya sampai selesai. Pastikan jaringan Anda tetap terhubung. Jika proses migrate telah selesai, silahkan untuk cek database pada PhpMyAdmin di <http://localhost/phpmyadmin>. Table-table serta field-field di dalamnya telah berhasil di generate seperti pada gambar.



8. Setelah berhasil migrate, Selanjutnya membuat Controller Barang. Controller ini bertujuan untuk mengatur semua aksi-aksi di dalam project dan tentunya akan menghubungkan antara model, view dan juga route. Silahkan jalankan perintah di bawah pada terminal/CMD di dalam project Laravel.

```
C:\Laravel\Barang>php artisan make:controller BarangController
```

Jika perintah diatas berhasil dijalankan, maka pada project Anda akan bertambah 1 file controller baru yang berada di folder app/Http/Controllers/BarangController.php. Lalu silahkan Anda membuka BarangController yang baru saja dibuat dan ubah codenya seperti pada code dibawah.

```
BarangController.php X
app > Http > Controllers > BarangController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\View\View;
6  use App\Models\Barang;
7
8  class BarangController extends Controller
9  {
10     /**
11      * index
12      *
13      * @return \Illuminate\View\View
14      */
15     public function index(): view
16     {
17         //get all barangs
18         $barangs = Barang::latest()->paginate(10);
19
20         //render view with barangs
21         return view('barangs.index', compact('barangs'));
22     }
23 }
24
```

Dari perubahan code diatas, kita Melakukan import Model **Barang** (use App\Models\Barang;) serta import return type **View**(use Illuminate\View\View;) dan menambahkan method baru yaitu index. Dalam method tersebut dilakukan pemanggilan data Barangs dari database melalui Model Barang (\$barangs = Barang::latest()->paginate(10);). Jika data berhasil didapatkan dan ditampung di dalam variable \$barangs lalu melakukan render view serta mengirim data Barang yang ada di dalam variable \$barangs (return view('barangs.index', compact('barangs')))).

9. Selanjutnya membuat route untuk menghubungkan antara user dengan Aplikasi. Silahkan Anda buka file routes/web.php lalu ubah codenya menjadi code seperti dibawah.

```
web.php X
routes > web.php
1  <?php
2
3  namespace App\Http\Controllers;
4  use Illuminate\Support\Facades\Route;
5
6  Route::resource('/barangs', \App\Http\Controllers\BarangController::class);
7
```

Mengapa route yang kita buat jenisnya resource? Agar laravel akan secara otomatis menghasilkan route-route untuk berbagai operasi CRUD seperti GET/ POST/ PUT/ PATCH/ DELETE. Untuk memastikan apakah route-route diatas telah di generate oleh laravel, silahkan jalankan perintah di bawah pada terminal/CMD project Anda dan tunggu sampai proses generate selesai.

```
C:\Laravel\Barang>php artisan route:list
```

Dan menghasilkan berikut:

```
POST      _ignition/execute-solution ignition.executeSolution > S...
GET|HEAD  _ignition/health-check  ignition.healthCheck > Spatie\La...
POST      _ignition/update-config  ignition.updateConfig > Spatie\...
GET|HEAD  barangs ..... barangs.index > BarangController@index
POST      barangs ..... barangs.store > BarangController@store
GET|HEAD  barangs/create barangs.create > BarangController@create
GET|HEAD  barangs/{barang} .. barangs.show > BarangController@show
PUT|PATCH barangs/{barang} barangs.update > BarangController@upda...
DELETE    barangs/{barang} barangs.destroy > BarangController@des...
GET|HEAD  barangs/{barang}/edit barangs.edit > BarangController@e...
GET|HEAD  up .....

Showing [11] routes
```



10. Setelah berhasil di generate, sekarang Anda membuat view untuk menampilkan Data. Silahkan Anda membuat folder baru dengan nama `barangs` di dalam folder `resources/views`. Dalam folder `barangs` silahkan buat file baru dengan nama `index.blade.php`, kemudian silahkan Anda ketikkan code dibawah ke dalam file `index`.

```
index.blade.php X
resources > views > barangs > index.blade.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="X-UA-Compatible" content="ie=edge">
6 <title>Data Fasilitas Barang</title>
7 </head>
8 <body>
9 <div class="container">
10 <h3>Data Fasilitas Barang di Institut Teknologi Del</h3>
11 <a href="{{ route('barangs.create') }}" class="btn btn-success mb-3">+Tambah</a>
12 <table border="1">
13 <thead>
14 <tr>
15 <th>Nama barang</th>
16 <th>Stok</th>
17 <th>Status</th>
18 <th>Gambar</th>
19 <th>Aksi</th>
20 </tr>
21 </thead>
22 <tbody>
23 @foreach ($barangs as $barang)
24 <tr>
25 <td>{{ $barang->namabarang }}</td>
26 <td>{{ $barang->stok }}</td>
27 <td>{{ $barang->status }}</td>
28 <td></td>
29 <td>
30 <form onsubmit="return confirm('Apakah Anda Yakin ?');" action="{{ route('barangs.destroy', $barang->id) }}"
31 method="POST" style="display: inline;">
32 @csrf
33 @method('DELETE')
34 <button type="submit" class="btn btn-danger">Hapus</button>
35 </form>
36 <form action="{{ route('barangs.edit', $barang->id) }}" method="POST" style="display: inline;">
37 @csrf
38 @method('edit')
39 <button type="submit" class="btn btn-danger">Edit</button>
40 </form>
41 </td>
42 </tr>
43 @empty
44 <tr>
45 <td colspan="5">Data Barang belum Tersedia.</td>
46 </tr>
47 @endforeach
48 </tbody>
49 </table>
50 {{ $barangs->links() }}
51 </div>
52 </body>
53 </html>
```

Selanjutnya coba jalankan project laravel yang telah Anda buat pada browser dengan mengakses <http://127.0.0.1:8000/barangs> maka tampilannya akan seperti di bawah.



11. Setelah Anda dapat menampilkan Data pada project laravel Anda, selanjutnya kita akan membuat insert data ke dalam database. Langkah awal yang perlu dilakukan adalah menambahkan method `create` dan `store` ke dalam Controller. Method `create` ini digunakan untuk menampilkan halaman form tambah data sedangkan method `store` digunakan untuk insert data ke dalam database dan Melakukan upload gambar. Silahkan Anda buka kembali file `BarangController.php` lalu ubah codenya seperti pada code berikut.

```
BarangController.php X
app > Http > Controllers > BarangController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\View\View;
6  use App\Models\Barang;
7
8  use Illuminate\Http\RedirectResponse;
9  use Illuminate\Http\Request;
10
11 class BarangController extends Controller
12 {
13     /**
14      * index
15      *
16      * @return \Illuminate\View\View
17      */
18     public function index(): View
19     {
20         //get all barangs
21         $barangs = Barang::latest()->paginate(10);
22
23         //render view with barangs
24         return view('barangs.index', compact('barangs'));
25     }
26
27     /**
28      * create
29      *
30      * @return View
31      */
32     public function create(): View
33     {
34         return view('barangs.create');
35     }
36
37     /**
38      * store
39      *
40      * @param mixed $request
41      * @return RedirectResponse
42      */
43     public function store(Request $request): RedirectResponse
44     {
45         //validate form
46
47         $request->validate([
48             'namabarang' => 'required|min: 5',
49             'stok'       => 'required|numeric',
50             'status'     => 'required|min:5',
51             'image'      => 'required|image|mimes:jpeg,jpg,png|max:2048'
52         ]);
53
54         //upload image
55         $image = $request->file('image');
56         $image->storeAs('public/barangs', $image->hashName());
57
58         //create barang
59         Barang::create([
60             'namabarang' => $request->namabarang,
61             'stok'       => $request->stok,
62             'status'     => $request->status,
63             'image'      => $image->hashName()
64         ]);
65
66         //redirect to index
67         return redirect()->route('barangs.index')->with(['success' => 'Data berhasil Disimpan!']);
68     }
69 }
```

Dari code diatas, pertama kita import return type RedirectRespon dan Http Request pada line 8 dan 9. Dalam BarangController kita menambahkan 2 method baru yaitu function create(line 3235) dan function store(line 43-67).

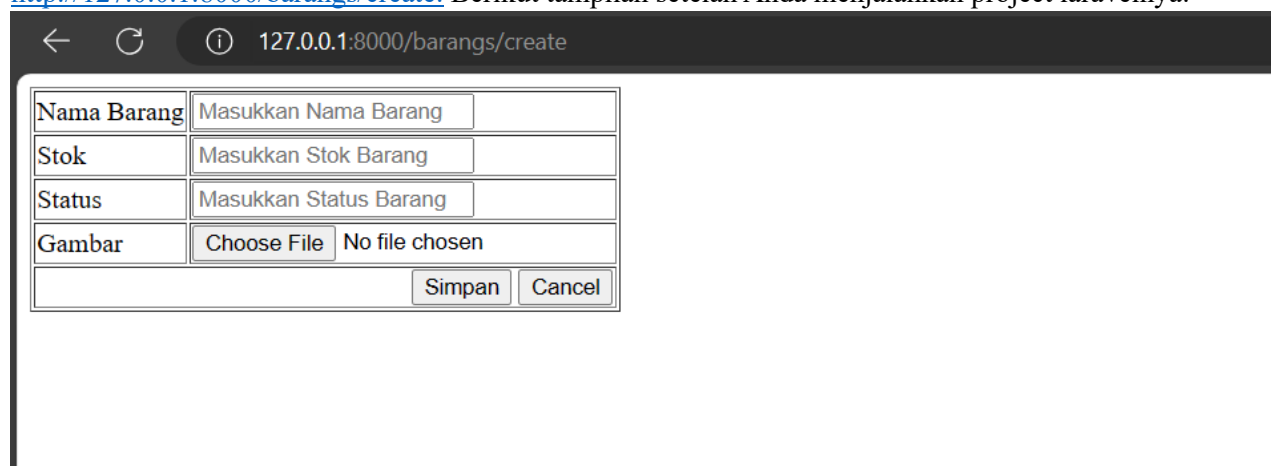
12. Silahkan mengetikkan code berikut untuk membuat view form create Barang. Silahkan buat terlebih dahulu file baru dengan penamaan create.blade.php di dalam folder resources/views/products.

```

create.blade.php X
resources > views > barangs > create.blade.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <meta http-equiv="X-UA-Compatible" content="ie=edge">
7 <title>Tambahkan Barang </title>
8 </head>
9 <body>
10
11 <div class="container">
12 <form action="{{ route('barangs.store') }}" method="POST" enctype="multipart/form-data">
13 @csrf
14 <table border="1">
15 <tr>
16 <td>Nama Barang</td>
17 <td><input type="text" name="namabarang" value="{{ old('namabarang') }}" placeholder="Masukkan Nama Barang"></td>
18 </tr>
19 <tr>
20 <td>Stok</td>
21 <td><input type="number" name="stok" value="{{ old('stok') }}" placeholder="Masukkan Stok Barang"></td>
22 </tr>
23 <tr>
24 <td>Status</td>
25 <td><input type="text" name="status" value="{{ old('status') }}" placeholder="Masukkan Status Barang"></td>
26 </tr>
27 <tr>
28 <td>Gambar</td>
29 <td><input type="file" name="image"></td>
30 </tr>
31 <tr>
32 <td colspan="2" style="text-align: right;">
33 <button type="submit" class="btn btn-md btn-primary me-3">Simpan</button>
34 <button type="reset" class="btn btn-md btn-warning">Cancel</button>
35 </td>
36 </tr>
37 </table>
38 </form>
39 </div>
40 </body>
41 </html>

```

13. Setelah selesai Anda ketik, silahkan jalankan project laravel Anda. Anda bisaklik button +Tambah pada halaman barangs index atau bisa juga mengakses link berikut pada browser Anda <http://127.0.0.1:8000/barangs/create>. Berikut tampilan setelah Anda menjalankan project laravelnya.




127.0.0.1:8000/barangs/create

Nama Barang	Kursi
Stok	200
Status	Tersedia
Gambar	Choose File OIP.jpg
<div> Simpan Cancel </div>	

127.0.0.1:8000/barangs

### Data Fasilitas Barang di Institut Teknologi Del

[+Tambah](#)

Nama barang	Stok	Status	Gambar	Aksi
Kursi	200	Tersedia		<div> Hapus Edit </div>

Tugas:

- Silahkan Anda buatkan untuk Update dan Delete pada project laravel diatas.


Jawab:

- Update**

127.0.0.1:8000/barangs

### Data Fasilitas Barang di Institut Teknologi Del

[+ Tambah](#)

Nama Barang	Stok	Status	Gambar	Aksi
Kursi	100	Tersedia		<div> Hapus Edit </div>

← ↻ 127.0.0.1:8000/barangs/7/edit

### Edit Barang

Nama Barang

Stok

Status

Gambar


Choose File No file chosen

Simpan Batal

← ↻ 127.0.0.1:8000/barangs

### Data Fasilitas Barang di Institut Teknologi Del

+ Tambah

Nama Barang	Stok	Status	Gambar	Aksi
Kursi	50	Tersedia		<span>Hapus</span> <span>Edit</span>

- **Delete**

← → ↻ 127.0.0.1:8000/barangs


127.0.0.1:8000 says

Apakah Anda Yakin ?

OK Cancel

### Data Fasilitas Barang di Institut Teknologi Del

+ Tambah

Nama Barang	Stok	Status	Gambar	Aksi
Kursi	50	Tersedia		<span>Hapus</span> <span>Edit</span>

← ↻ 127.0.0.1:8000/barangs

### Data Fasilitas Barang di Institut Teknologi Del

+ Tambah

Nama Barang	Stok	Status	Gambar	Aksi
Data Barang belum Tersedia.				

2. Silahkan Anda tambahkan script css pada project laravel Anda.

**Jawab:**

Data Fasilitas Barang di Institut Teknologi Del

+ Tambah

Nama Barang	Stok	Status	Gambar	Aksi
-------------	------	--------	--------	------

Data Barang belum Tersedia.

## Tambahkan Barang

Nama Barang

Stok

Status

Gambar

Choose File No file chosen

Simpan

Cancel

Data Fasilitas Barang di Institut Teknologi Del

+ Tambah

Nama Barang	Stok	Status	Gambar	Aksi
-------------	------	--------	--------	------

Kursi

200

Tersedia



Aksi

Hapus

Edit

Praktikum sesi 3 dibuatkan ke dalam Laporan dalam bentuk pdf dengan penamaan CRUD\_Laravel\_NimLengkap.pdf. Tugas akan di submit ke ecourse dengan batas waktu Minggu, 21 April 2024 Pukul 22:00 WIB.