

**SISTEM PENDUKUNG KEPUTUSAN
JOBSHEET 6 - TEMPLATE FORM (ADMINLTE),
SERVER VALIDATION, CLIENT VALIDATION,
CRUD**



Disusun oleh:

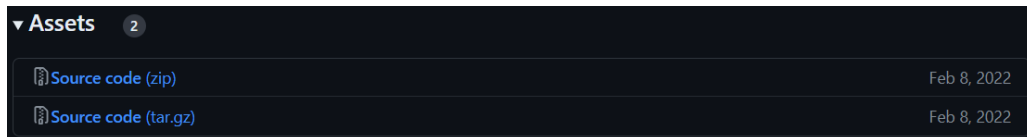
Stefanus Ageng Budi Utomo

(2241720126)

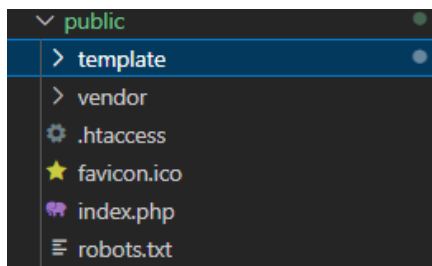
**JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
TAHUN AJARAN 2023/2024**

A. Template Form (AdminLTE)

1. Penggunaan template AdminLTE Bootstrap Admin Dashboard Template dengan langkah pertama akses <https://adminlte.io/> , lalu klik download pada source code (zip)



2. Lakukan extract All dan ubah nama folder AdminLTE-3.2.0 menjadi template, folder template pindahkan ke laragon/www/PWL_2024/public. Buka folder public pada VSCode maka akan tampil folder baru bernama template



3. Copy isi public/template/index2.html pada welcome.blade.php lakukan edit agar dapat diload, edit href="{{ asset('template/.....') }}" dan src="{{ asset('template/.....') }}" sesuai gambar berikut



4. Scroll ke bagian paling bawah dan sesuaikan dengan kode berikut

```

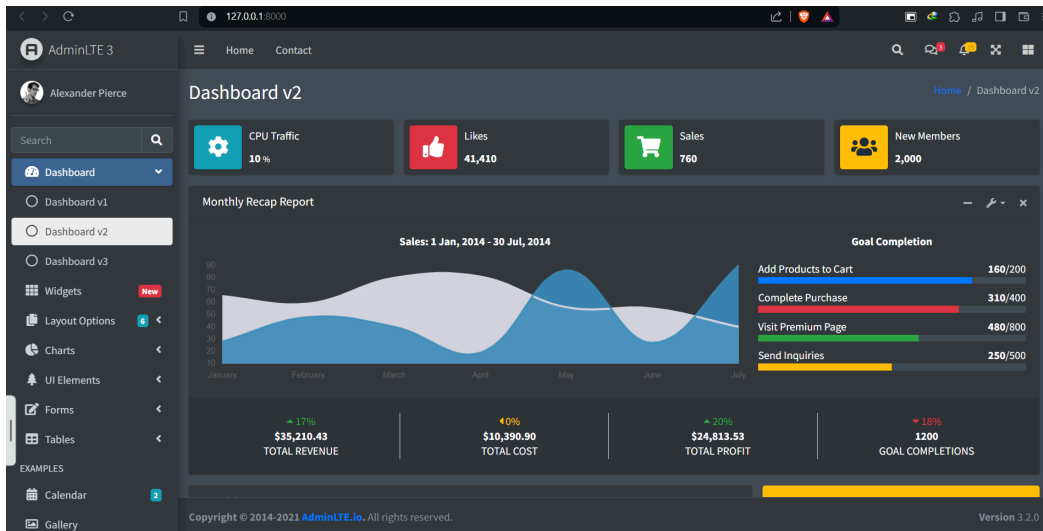
<!-- REQUIRED SCRIPTS -->
<!-- jQuery -->
<script src="{{ asset('template/plugins/jquery/jquery.min.js') }}"></script>
<!-- Bootstrap -->
<script src="{{ asset('template/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
<!-- overlayScrollbars -->
<script src="{{ asset('template/plugins/overlayScrollbars/js/jquery.overlayScrollbars.min.js') }}"></script>
<!-- AdminLTE App -->
<script src="{{ asset('template/dist/js/adminlte.js') }}"></script>

<!-- PAGE PLUGINS -->
<!-- jQuery Mapael -->
<script src="{{ asset('template/plugins/jquery-mousewheel/jquery.mousewheel.js') }}"></script>
<script src="{{ asset('template/plugins/raphael/raphael.min.js') }}"></script>
<script src="{{ asset('template/plugins/jquery-mapael/jquery.mapael.min.js') }}"></script>
<script src="{{ asset('template/plugins/jquery-mapael/maps/usa_states.min.js') }}"></script>
<!-- ChartJS -->
<script src="{{ asset('template/plugins/chart.js/Chart.min.js') }}"></script>

<!-- AdminLTE for demo purposes -->
<script src="{{ asset('template/dist/js/demo.js') }}"></script>
<!-- AdminLTE dashboard demo (This is only for demo purposes) -->
<script src="{{ asset('template/dist/js/pages/dashboard2.js') }}"></script>

```

5. s. Jalankan pada browser



6. Pada bagian menu terdapat pilihan Form kita khususkan dulu pada Form yang disediakan oleh Admin LTE
- General Elements
 - Advanced Elements
 - Editor
 - Validation

Cara menggunakan template bisa dilihat nama formnya contoh judulnya 'General Elements', agar dapat digunakan dapat melakukan akses pada folder yang terdapat template adminLTE contoh : C:\laragon\www\IPWL_2024\public\template\pages\forms

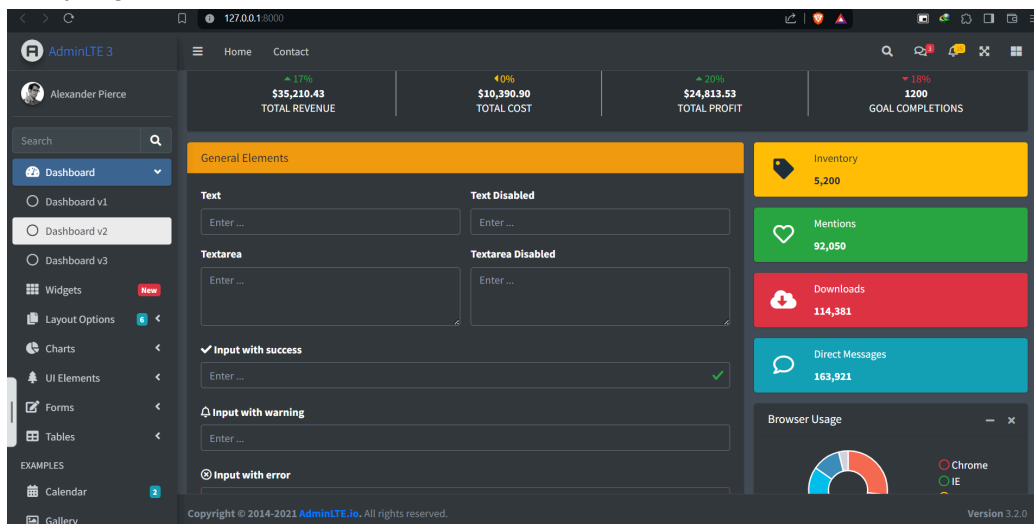
7. Buka file pada form> general cari keyword dengan 'general elements' dan coba modifikasi pada welcome.blade.php

```

<div class="card card-warning">
  <div class="card-header">
    <h3 class="card-title">General Elements</h3>
  </div>
  <!-- /.card-header -->
  <div class="card-body">
    <form>
      <div class="row">
        <div class="col-sm-6">
          <!-- text input -->
          <div class="form-group">
            <label>Text</label>
            <input type="text" class="form-control" placeholder="Enter ...">
          </div>
        </div>
        <div class="col-sm-6">
          <div class="form-group">
            <label>Text Disabled</label>
            <input type="text" class="form-control" placeholder="Enter ..." disabled>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

8. Apa yang tampil?



9. Eksplorasi jenis form pada adminLTE dan coba terapkan yang sesuai untuk studi kasus POS PWL, buatlah form untuk tabel m_user dan m_level

User > Create

Form Tambah Data User

Username

Masukan Username

Nama

Masukan nama

Password

Password

Level_id

Level ID

Submit

Level > Create

Form Tambah Data Level

Level Kode

Masukan level kode

Level Nama

Masukan level kode

Submit

B. Validasi Pada Server

1. Buka kembali file tugas di jobsheet 05, klik dan baca documentation ini Validation - Laravel 10.x - The PHP Framework For Web Artisans
2. Pastikan sudah terdefiniskan route pada web.php :

```
Route::post('/kategori', [KategoriController::class, 'store']);
Route::get('/kategori/create', [KategoriController::class, 'create'])->name('kategori.create');
```

3. Edit pada KategoriKontroller dengan kode:

```
class KategoriController extends Controller
{
    public function store(Request $request)
    {
        // KategoriModel::create([
        //     'kategori_kode' => $request->kodeKategori,
        //     'kategori_nama' => $request->namaKategori,
        // ]);
        $validated = $request->validate([
            'kategori_kode' => 'required',
            'kategori_nama' => 'required',
        ]);
        // the post is valid
        return redirect('/kategori');
    }
}
```

4. Berikut ini sebagai alternatif contoh penulisan validasi dan bisa Anda sesuaikan katakata dengan apa yang sedang Anda kerjakan,

Alternatively, validation rules may be specified as arrays of rules instead of a single | delimited string:

```
$validatedData = $request->validate([
    'title' => ['required', 'unique:posts', 'max:255'],
    'body' => ['required'],
]);
```

In addition, you may use the `validateWithBag` method to validate a request and store any error messages within a named error bag:

```
$validatedData = $request->validateWithBag('post', [
    'title' => ['required', 'unique:posts', 'max:255'],
    'body' => ['required'],
]);
```

5. Tulis perbedaan penggunaan `validate` dengan `validateWithBag`!
 - penggunaan `validateWithBag('post')` memberikan kontrol lebih besar atas penanganan pesan error validasi dengan memungkinkan Anda untuk menentukan "bag" khusus untuk menyimpan pesan error. Sedangkan `validate()` akan menyimpan pesan error validasi secara default tanpa perlu menentukan "bag" khusus. Pilihan antara keduanya tergantung pada kebutuhan dan preferensi Anda dalam penanganan pesan error validasi.
6. Terkadang Anda mungkin ingin berhenti menjalankan aturan validasi pada suatu atribut setelah kegagalan validasi pertama. Untuk melakukannya, tetapkan `bail` aturan ke atribut: coba sesuaikan dengan field pada `m_kategori`, yuk bisa.. apa yang terjadi?

```

public function store(Request $request)
{
    $validated = $request->validate([
        'kategori_kode' => 'bail|required|unique:m_kategori, kategori_kode',
        'kategori_nama' => 'required',
    ]);
    KategoriModel::create([
        'kategori_kode' => $request->kodeKategori,
        'kategori_nama' => $request->namaKategori,
    ]);

    // the post is valid

    return redirect('/kategori');
}

```

- bail akan berhenti melakukan validasi pada atribut jika validasi gagal pada aturan tersebut.

7. Pada view/create.blade.php tambahkan code berikut agar ketika validasi gagal, kita dapat menampilkan pesan kesalahan dalam tampilan:

```

resources > views > kategori > create.blade.php > div.container > div.card.card-primary > form
8  <div class="container">
9  <div class="card card-primary">
14 <form action=".." />
25
26 <div class="card-footer">
27 <button type="submit" class="btn btn-primary">Submit</button>
28 </div>
29 @if ($errors->any())
30 <div class="alert alert-danger">
31 <ul>
32 @foreach ($errors->all() as $error)
33 <li>{{ $error }}</li>
34 @endforeach
35 </ul>
36 </div>
37 @endif
38 </form>
39 </div>
40 </div>
41 @endsection

```

8. Screenshot hasilnya

9. Catatan : Menyesuaikan Pesan Kesalahan

Aturan validasi bawaan Laravel masing-masing memiliki pesan kesalahan yang terletak di lang/en/validation.php file aplikasi Anda. Jika aplikasi Anda tidak memiliki langdirektori, Anda dapat menginstruksikan Laravel untuk membuatnya menggunakan lang:publish perintah Artisan. Di dalam lang/en/validation.php file tersebut, Anda akan menemukan entri terjemahan untuk setiap aturan validasi. Anda bebas mengubah atau memodifikasi pesan-pesan ini berdasarkan kebutuhan aplikasi Anda

10. Pada view/create.blade.php tambahkan dan coba running code berikut :

```
<label for="kategori_kode">Kode kategori</label>

<input id="kategori_kode"
      type="text"
      name="kategori_kode"
      class="@error('kategori_kode') is-invalid @enderror">

@error('kategori_kode')
    <div class="alert alert-danger">{{ $message }}</div>
@enderror
```


C. Form Request Validation

1. Membuat permintaan form dengan menuliskan pada terminal

```
php artisan make:request StorePostRequest
```

kelas permintaan formulir yang dihasilkan akan ditempatkan di `app/Http/Requests` direktori. Jika direktori ini tidak ada, maka akan dibuat saat Anda menjalankan `make:request` perintah. Setiap permintaan formulir yang dihasilkan oleh Laravel memiliki dua metode: `authorize` dan `rules`.

2. Ketik kode berikut pada `Http/request/StorePostRequest`

```
class StorePostRequest extends FormRequest
{
    public function rules(): array
    {
        return [
            //
            'kategori_kode' => 'required',
            'kategori_nama' => 'required',
        ];
    }
}
```

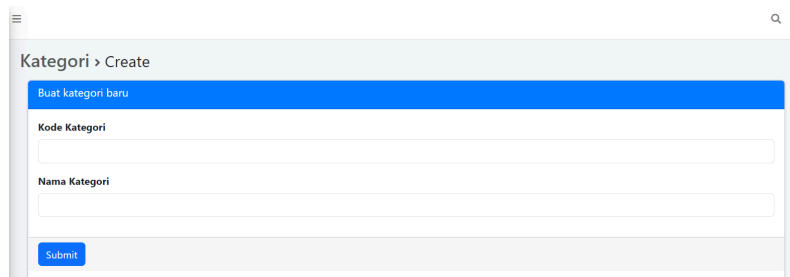
```
class KategoriController extends Controller
{
    public function store(StorePostRequest $request) // type before Request
    {
        // Retrieve the validated input data
        $validated = $request->validated();

        // Retrieve a portion of the validated input data...
        $validated = $request->safe()->only(['kategori_kode', 'kategori_nama']);
        $validated = $request->safe()->except(['kategori_kode', 'kategori_nama']);

        // KategoriModel::create([
        //     'kategori_kode' => $request->kodeKategori,
        //     'kategori_nama' => $request->namaKategori,
        // ]);

        // the post is valid
        return redirect('/kategori');
    }
}
```

Note : Jika validasi gagal, respons pengalihan akan dihasilkan untuk mengirim pengguna kembali ke lokasi sebelumnya. Kesalahan juga akan ditampilkan ke sesi sehingga tersedia untuk ditampilkan. Jika permintaannya adalah permintaan XHR, respons HTTP dengan kode status 422 akan dikembalikan ke pengguna.



Kategori > Create

Buat kategori baru

Kode Kategori

Nama Kategori

Submit

- The kategori kode field is required.
- The kategori nama field is required.

3. Terapkan validasi juga pada tabel m_user dan m_level

```

public function tambah_simpan(StorePostRequest $request) //Request
{
    $validated = $request->validate([
        'username' => 'bail|required|unique:m_user, username',
        'nama' => 'required',
        'password' => 'required',
        'level_id' => 'required',
    ]);

    try {
        //code...

        // Retrieve the validated input data
        $validated = $request->validated();

        // Retrieve a portion of the validated input data...
        $validated = $request->safe()->only(['username', 'nama', 'password', 'level_id']);
        $validated = $request->safe()->except(['username', 'nama', 'password', 'level_id']);

        UserModel::create([
            'username' => $request->username,
            'nama' => $request->nama,
            'password' => Hash::make($request->password),
            'level_id' => $request->level_id,
        ]);
        return redirect('/user')->with('success', 'Data user berhasil disimpan');
    } catch (\Throwable $th) {
        //throw $th;
        return redirect()->back()->with('error', 'Data user gagal disimpan, silahkan cek dan coba lagi');
    }
}

```

m_level

```
public function store(StorePostRequest $request) // type before Request
{
    $validated = $request->validate([
        'level_kode' => 'bail|required|unique:m_level, level_kode',
        'level_nama' => 'required',
    ]);

    try {
        //code...
        // The incoming request is valid...

        // Retrieve the validated input data
        $validated = $request->validated();

        // Retrieve a portion of the validated input data...
        $validated = $request->safe()->only(['level_kode', 'level_nama']);
        $validated = $request->safe()->except(['level_kode', 'level_nama']);

        LevelModel::create([
            'level_kode' => $request->kodeLevel,
            'level_nama' => $request->namaLevel,
        ]);

        // the post is valid

        return redirect('/level')->with('success', 'Data level berhasil disimpan');
    } catch (\Throwable $th) {
        //throw $th;
        return redirect()->back()->with('error', 'Data level gagal disimpan, silahkan cek dan coba lagi');
    }
}
```

D. CRUD(create, read,update, delete)

1. Buat POSController lengkap dengan resourcenya, Membuat Resource Controller dan Route yang berfungsi untuk route CRUD sehingga tidak perlu repot-repot membuat masing-masing route seperti post, get, delete dan update.
2. Berikut perintahnya: sekarang tambahkan kode pada route/web.php

```
// resource
Route::resource('m_user', POSController::class);
```

3. Atur pada Models/m_user

```
class UserModel extends Model
{
    use HasFactory;

    protected $table = 'm_user'; // Mendefinisiakan nama tabel yang digunakan oleh model ini
    protected $primaryKey = 'user_id'; // Mendefinisikan primary key dari tabel yang digunakan
    public $timestamps = false;

    protected $fillable = ['user_id', 'level_id', 'username', 'nama', 'password'];

    public function level(): BelongsTo
    {
        return $this->belongsTo(LevelModel::class, 'level_id', 'level_id');
    }
}
```

4. Atur pada Migration/m_user_table

```
public function up(): void
{
    Schema::create('m_user', function (Blueprint $table) {
        $table->id('user_id');
        $table->unsignedBigInteger('level_id')->index(); //indexing untuk ForeignKey
        $table->string('username', 20)->unique();
        $table->string('nama', 100);
        $table->string('password');
        $table->timestamps();

        // mendefinisikan Foreign Key pada kolom level_id mengacu pada kolom level_id di tabel m_level
        $table->foreign('level_id')->references('level_id')->on('m_level');
    });
}
```

5. Pada file Controller yang kita dapat secara otomatis akan terdapat 7 fungsi berikut yang dapat kita gunakan untuk membuat operasi CRUD.
 - a. index()
 - b. create()
 - c. store()
 - d. show()
 - e. edit()
 - f. update()
 - g. destroy()
6. Selanjutnya bukalah app/Http/Controllers/POSController.php kemudian ketikkan, kodenya seperti berikut ini:

```

class POSController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        //fungsi eloquent menampilkan data menggunakan pagination
        $m_user = UserModel::all(); // Mengambil semua isi tabel
        return view('user.index', compact('m_user'))->with('i');
    }

    /**
     * Show the form for creating a new resource.
     */
    public function create()
    {
        //
        return view('user.index');
    }
}

```

```

public function store(Request $request)
{
    // melakukan validasi data
    $request->validate([
        'user_id' => 'max 20',
        'username' => 'required',
        'nama' => 'required',
    ]);

    UserModel::create($request->all());

    return redirect()->route('user.index')->with('success', 'user berhasil ditambahkan');
}

/**
 * Display the specified resource.
 */
public function show(string $id)
{
    //
    $m_user = UserModel::find($id);
    return view('user.show', compact('m_user'));
}

```

```

public function edit(string $id)
{
    //
    $m_user = UserModel::find($id);
    return view('user.edit', compact('m_user'));
}

/**
 * Update the specified resource in storage.
 */
public function update(Request $request, string $id)
{
    //
    $request->validate([
        'username' => 'required',
        'nama' => 'required',
        'password' => 'required',
    ]);
    // fungsi eloquent untuk mengupdate data inputan kita
    UserModel::find($id)->update($request->all());
    return redirect()->route('user.index')->with('success', 'Data berhasil diupdate');
}

```

```

/**
 * Remove the specified resource from storage.
 */
public function destroy(string $id)
{
    //
    $m_user = UserModel::findOrFail($id)->delete();
    return \redirect()->route('user.index')->with('success', 'Data berhasil dihapus');
}

```

7. Buatlah folder di Resources/Views/m_user dengan beberapa blade dan isian kode berikut:

a. template.blade.php

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>CRUD Laravel</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJ1SA8A" crossorigin="anonymous">
    <style>|...
</style>
</head>

<body>
    <div class="container">
        @yield('content')
    </div>
</body>

</html>
```

b. index.blade.php

```
@extends('user/template')

@section('content')
    <div class="container mt-5">
        <div class="row">
            <div class="col-lg-12">
                <div class="card">
                    <div class="card-header d-flex justify-content-between align-items-center">
                        <h2 class="m-0">CRUD User</h2>
                        <a href="{{ route('user.create') }}" class="btn btn-success">Input User</a>
                    </div>
                    @if ($message = Session::get('success'))
                        <div class="alert alert-success mt-3 mb-0">
                            <p>{{ $message }}</p>
                        </div>
                    @endif
                    <div class="card-body">
                        <table class="table table-bordered">
                            <thead>
                                <tr>
                                    <th class="text-center">User ID</th>
                                    <th class="text-center">Level ID</th>
                                    <th class="text-center">Username</th>
                                    <th class="text-center">Nama</th>
                                    <th class="text-center">Password</th>
                                    <th class="text-center">Actions</th>
                                </tr>
                            </thead>
                            <tbody>
                                @foreach ($m_user as $user)
                                    <tr>
                                        <td>{{ $user->user_id }}</td>
                                        <td>{{ $user->level_id }}</td>
                                        <td>{{ $user->username }}</td>
                                        <td>{{ $user->nama }}</td>
                                        <td>{{ $user->password }}</td>
                                        <td class="text-center">
                                            <form action="{{ route('user.destroy', $user->user_id) }}" method="post">
                                                <a href="{{ route('user.show', $user->user_id) }}"
                                                    class="btn btn-info btn-sm">Show</a>
                                                <a href="{{ route('user.edit', $user->user_id) }}"
                                                    class="btn btn-primary btn-sm">Edit</a>
                                                @csrf
                                                @method('DELETE')
                                                <button type="submit" class="btn btn-danger btn-sm">Delete</button>
                                            </form>
                                        </td>
                                    </tr>
                                </tbody>
                            </table>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
@endsection
```

```

@csrf
@method('DELETE')
<button type="submit" class="btn btn-danger btn-sm"
    onclick="return confirm('Are you sure you want to delete this item?')
">Delete</button>
</form>
</td>
</tr>
@endforeach
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
@endsection

```

c. create.blade.php

```

@extends('user/template')
@section('content')
    <div class="row mt-5 mb-5">
        <div class="col-lg-12 margin-tb">
            <div class="float-left">
                <h2>Membuat Daftar User</h2>
            </div>
            <div class="float-right">
                <a href="{{ route('user.index') }}" class="btn btn-secondary">Kembali</a>
            </div>
        </div>
    </div>
    @if ($errors->any())
        <div class="alert alert-danger">
            <strong>Ops!</strong> Input gagal <br><br>
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif

    <form action="{{ route('user.store') }}" method="post">
        @csrf
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>Username:</strong>
                <input type="text" name="username" class="form-control" placeholder="Masukan username">
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>nama:</strong>
                <input type="text" name="nama" class="form-control" placeholder="Masukan nama">
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>password:</strong>
                <input type="password" name="password" class="form-control" placeholder="Masukan password">
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12 text-center">
            <button type="submit" class="btn btn-primary">Submit</button>
        </div>
    </form>
@endsection

```

d. show.blade.php

```

@extends('user/template')
@section('content')
    <div class="row mt-5 mb-5">
        <div class="col-lg-12 margin-tb">
            <div class="float-left">
                <h2>Show User</h2>
            </div>
            <div class="float-right">
                <a href="{{ route('user.index') }}" class="btn btn-secondary">Kembali</a>
            </div>
        </div>
    </div>
    <div class="row">
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>User_id:</strong>
                {{ $m_user->user_id }}
            </div>
            <div class="col-xs-12 col-sm-12 col-md-12">
                <div class="form-group">
                    <strong>Level_id:</strong>
                    {{ $m_user->level_id }}
                </div>
            </div>
            <div class="col-xs-12 col-sm-12 col-md-12">
                <div class="form-group">
                    <strong>Username:</strong>
                    {{ $m_user->username }}
                </div>
            </div>
            <div class="col-xs-12 col-sm-12 col-md-12">
                <div class="form-group">
                    <strong>Nama:</strong>
                    {{ $m_user->nama }}
                </div>
            </div>
            <div class="col-xs-12 col-sm-12 col-md-12">
                <div class="form-group">
                    <strong>Password:</strong>
                    {{ $m_user->password }}
                </div>
            </div>
        </div>
    </div>
@endsection

```

e. edit.blade.php


```

@extends('user/template')
@section('content')
    <div class="row mt-5 mb-5">
        <div class="col-lg-12 margin-tb">
            <div class="float-left">
                <h2>Edit User</h2>
            </div>
            <div class="float-right">
                <a href="{{ route('user.index') }}" class="btn btn-secondary">Kembali</a>
            </div>
        </div>
    </div>
    @if ($errors->any())
        <div class="alert alert-danger">
            <strong>Ops!</strong> Input gagal <br><br>
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif

    <form action="{{ route('user.update') }}" method="post">
        @csrf
        @method('PUT')
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>User_id:</strong>
                <input type="text" name="userid" class="form-control" placeholder="Masukan user id"
                    value="{{ $user->user_id }}">
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>Level_id:</strong>
                <input type="text" name="levelid" class="form-control" placeholder="Masukan level">
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>Username:</strong>
                <input type="text" name="username" class="form-control" placeholder="Masukan username"
                    value="{{ $user->username }}">
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>nama:</strong>
                <input type="text" name="nama" class="form-control" placeholder="Masukan nama"
                    value="{{ $user->nama }}">
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>password:</strong>
                <input type="password" name="password" class="form-control" placeholder="Masukan password"
                    value="{{ $user->password }}">
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12 text-center">
            <button type="submit" class="btn btn-primary">Update</button>
        </div>
    </form>
@endsection

```

8. Silahkan akses localhost/m_user

127.0.0.1:8000/user

CRUD user Input User

User_id	Level_id	Username	nama	password	
1	1	admin	AdministratorSatu	\$2y\$12\$RCJIT1qTNPxccqj255jVA.p9xot85RJ3fWMa5Zvu77lecKaxRoWaa	<a>Show <a>Edit <a>Delete
2	2	manager	Manager	\$2y\$12\$LK5atJjNPxSe5pcVTYQzuuplvExfJXylxKP6xSr4n3SIWipeVM6.	<a>Show <a>Edit <a>Delete
3	3	staff	Staff/Kasir	\$2y\$12\$TzxT7uMIEvrME6Z4A2p.a9BfT6W89YEqi.pPx8veP5wBqRdS7IK	<a>Show <a>Edit <a>Delete
6	5	customer-1	Pelanggan Pertama	\$2y\$12\$L6szSUm3vQW9kj1xlwxQzRFwd5CTUCLJ/w1uH5JISeTqWMrSDb6	<a>Show <a>Edit <a>Delete

Tugas

1. Coba tampilkan level_id pada halaman web tersebut dimana field ini merupakan foreign key

CRUD user Input User

User_id	Level_id	Username	nama	password	
1	1	admin	AdministratorSatu	\$2y\$12\$RCJIT1qTNPxcccq25SjVA.p9xot85RJ3fWMA5Zvu77lecKaxRoWaa	<a>Show <a>Edit <a>Delete
2	2	manager	Manager	\$2y\$12\$LK5atJlNPx5e5pcVTYQzuupJvExfJXyJxK6xSr4n3SIWlpeVM6.	<a>Show <a>Edit <a>Delete
3	3	staff	Staff/Kasir	\$2y\$12\$ZTzTJuMIEvrRME6Z4A2p.a9BfT6W89YEqi.pPx8veP5wBqRdS7IK	<a>Show <a>Edit <a>Delete
6	5	customer-1	Pelanggan Pertama	\$2y\$12\$Le6sz5Um3vQW9kj1xlwxQzRFwdSCTUCIJ/w1uHSJISeTqWMrSDb6	<a>Show <a>Edit <a>Delete

```
<table class="table table-bordered">
  <tr>
    <th width="20px" class="text-center">User_id</th>
    <th width="150px" class="text-center">Level_id</th>
    <th width="200px" class="text-center">Username</th>
    <th width="200px" class="text-center">nama</th>
    <th width="150px" class="text-center">password</th>
  </tr>
  @foreach ($m_user as $user)
    <tr>
      <td>{{ $user->user_id }}</td>
      <td>{{ $user->level_id }}</td>
      <td>{{ $user->username }}</td>
      <td>{{ $user->nama }}</td>
      <td>{{ $user->password }}</td>
      <td class="text-center d-flex">
        <form action="{{ route('user.destroy', $user->user_id) }}" method="post">
          <a href="{{ route('user.show', $user->user_id) }}" class="btn btn-info btn-sm">Show</a>
          <a href="{{ route('user.show', $user->user_id) }}" class="btn btn-primary btn-sm">Edit</a>
          @csrf
          @method('DELETE')
          <button type="submit" class="btn btn-danger btn-sm"
            onclick="return confirm('Apakah Anda ingin menghapus data ini?')">Delete</button>
        </form>
      </td>
    </tr>
  @endforeach
</table>
```

2. Modifikasi dengan tema/ template kesukaan Anda

CRUD User Input User

User ID	Level ID	Username	Nama	Password	Actions
1	1	admin	AdministratorSatu	\$2y\$12\$RCJIT1qTNPxcccq25SjVA.p9xot85RJ3fWMA5Zvu77lecKaxRoWaa	<a>Show <a>Edit <a>Delete
2	2	manager	Manager	\$2y\$12\$LK5atJlNPx5e5pcVTYQzuupJvExfJXyJxK6xSr4n3SIWlpeVM6.	<a>Show <a>Edit <a>Delete
3	3	staff	Staff/Kasir	\$2y\$12\$ZTzTJuMIEvrRME6Z4A2p.a9BfT6W89YEqi.pPx8veP5wBqRdS7IK	<a>Show <a>Edit <a>Delete
6	5	customer-1	Pelanggan Pertama	\$2y\$12\$Le6sz5Um3vQW9kj1xlwxQzRFwdSCTUCIJ/w1uHSJISeTqWMrSDb6	<a>Show <a>Edit <a>Delete

user/template

```
<style>
  body {
    background-color: #f8f9fa;
    color: #333;
  }

  .container {
    margin-top: 50px;
  }

  .card-header {
    background-color: #007bff;
    color: #fff;
  }

  .card-body {
    background-color: #fff;
  }

  .btn {
    margin-right: 5px;
  }

  .alert {
    margin-top: 10px;
  }

  .table th,
  .table td {
    vertical-align: middle;
  }
</style>
```

3. Apa fungsi `$request->validate`, `$error` dan `alert` yang ada pada halaman CRUD tersebut
- Method 'validate' pada objek '\$request' digunakan untuk melakukan validasi terhadap input yang diterima dari pengguna
 - Variable '\$errors' menyimpan pesan-pesan kesalahan yang dihasilkan dari validasi
 - Alert digunakan untuk memberikan atau menampilkan informasi kepada pengguna tentang status operasi yang dilakukan.