1. Migration

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('pelanggans', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('users');
    }
};
```

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
```

```php
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('produks', function (Blueprint $table) {
            $table->id();
            $table->string('nama');
            $table->string('keterangan');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('produks');
    }
};
```

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
```

```
    {
        Schema::create('pesanans', function (Blueprint $table) {
            $table->id();

$table->foreignId('pelanggan_id')->constrained()->onDelete('cascade')->onU
pdate('cascade');

$table->foreignId('produk_id')->constrained()->onDelete('cascade')->onUpda
te('cascade');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('pesanans');
    }
};
```

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('pembayarans', function (Blueprint $table) {
            $table->id();
```

```
$table->foreignId('pesanan_id')->constrained()->onDelete('cascade')->onUpd
ate('cascade');
            $table->integer('status')->default('0');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('pembayarans');
    }
};
```

```
<?php

namespace Database\Seeders;

// use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;

class DatabaseSeeder extends Seeder
{
    /**
     * Seed the application's database.
     */
    public function run(): void
    {
        // \App\Models\User::factory(10)->create();

        // \App\Models\User::factory()->create([
        //     'name' => 'Test User',
        //     'email' => 'test@example.com',
        // ]);
        $this->call(ProductSeeder::class);
```

```
    }
}
```

```php
<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use App\Models\Produk;

class ProductSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        Produk::factory()->count(5)->create();
    }
}
```

```php
<?php

namespace Database\Factories;

use Illuminate\Database\Eloquent\Factories\Factory;

/**
* @extends
\Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Produk>
*/
class ProdukFactory extends Factory
{
    /**
     * Define the model's default state.
```

```
     *
     * @return array<string, mixed>
     */
    public function definition(): array
    {
        return [
            'nama' => $this->faker->word,
            'keterangan' => $this->faker->sentence,
        ];
    }
}
```

2. CRUD

```
<?php

namespace App\Models;

// use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'email',
        'password',
    ];
    protected $table = 'pelanggans';
```

```
    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * The attributes that should be cast.
     *
     * @var array<string, string>
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
        'password' => 'hashed',
    ];
    public function pesanan(){
        return $this->hasMany(Pesanan::class);
    }
}
```

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Produk extends Model
{
    use HasFactory;
    protected $fillable = [
        'nama',
        'keterangan',
```

```
    ];
    public function pesanan(){
        $this->hasMany(Pesanan::class);
    }
}
```

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use App\Models\Pembayaran;

class Pesanan extends Model
{
    use HasFactory;
    protected $fillable = [
        'pelanggan_id',
        'produk_id',
    ];
    protected static function booted()
    {
        static::created(function () {
            $pesanan = Pesanan::orderByDesc('id')->first();
            $baru = [
                'pesanan_id' => $pesanan->id,
            ];
            Pembayaran::insert($baru);
        });
    }
    public function user(){
        return $this->belongsTo(User::class);
    }
    public function produk(){
        return $this->belongsTo(Produk::class);
    }
    public function pembayaran(){
```

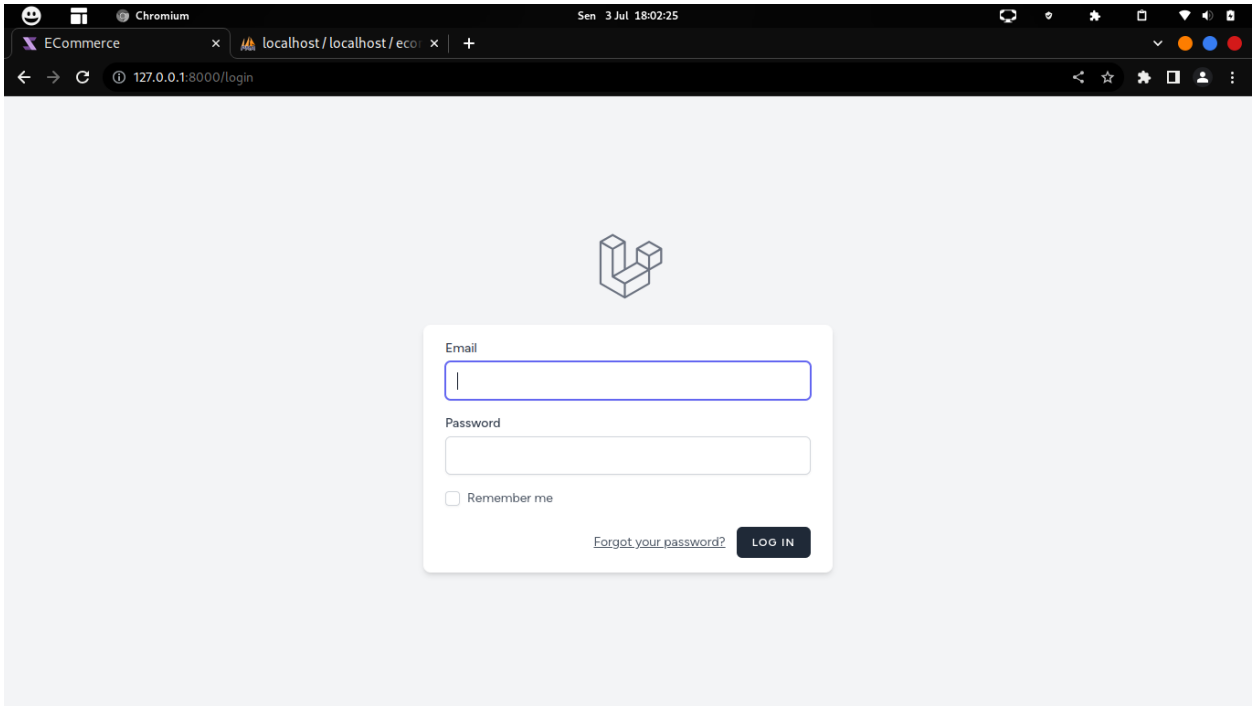
```
        return $this->hasOne(Pembayaran::class);
    }
}
```

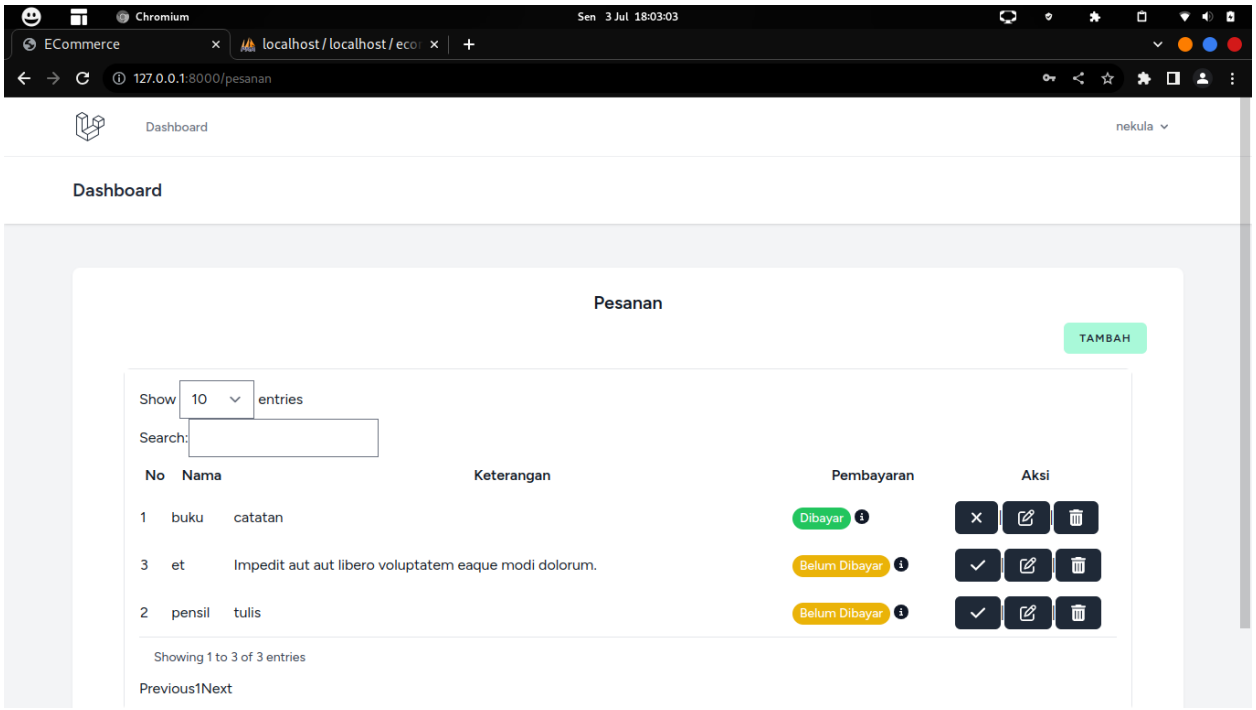
```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Pembayaran extends Model
{
    use HasFactory;
    protected $fillable = [
        'pesanan_id',
        'status'
    ];
    public function pesanan(){
        return $this->belongsTo(Pesanan::class);
    }
}
```

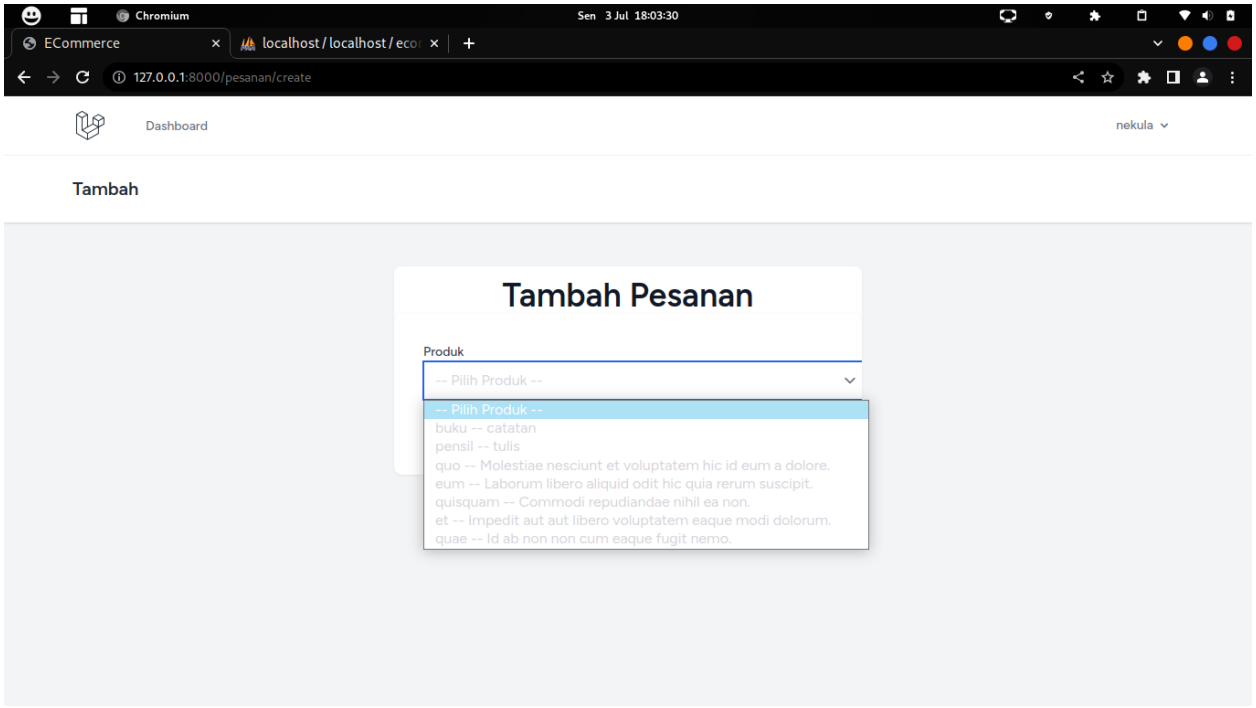
```
<?php

namespace App\Http\Controllers;

use App\Models\Pesanan;
use App\Models\Produk;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class PesananController extends Controller
{
    public function index(){
        $data = Pesanan::where('pelanggan_id', Auth::id())->get();
```

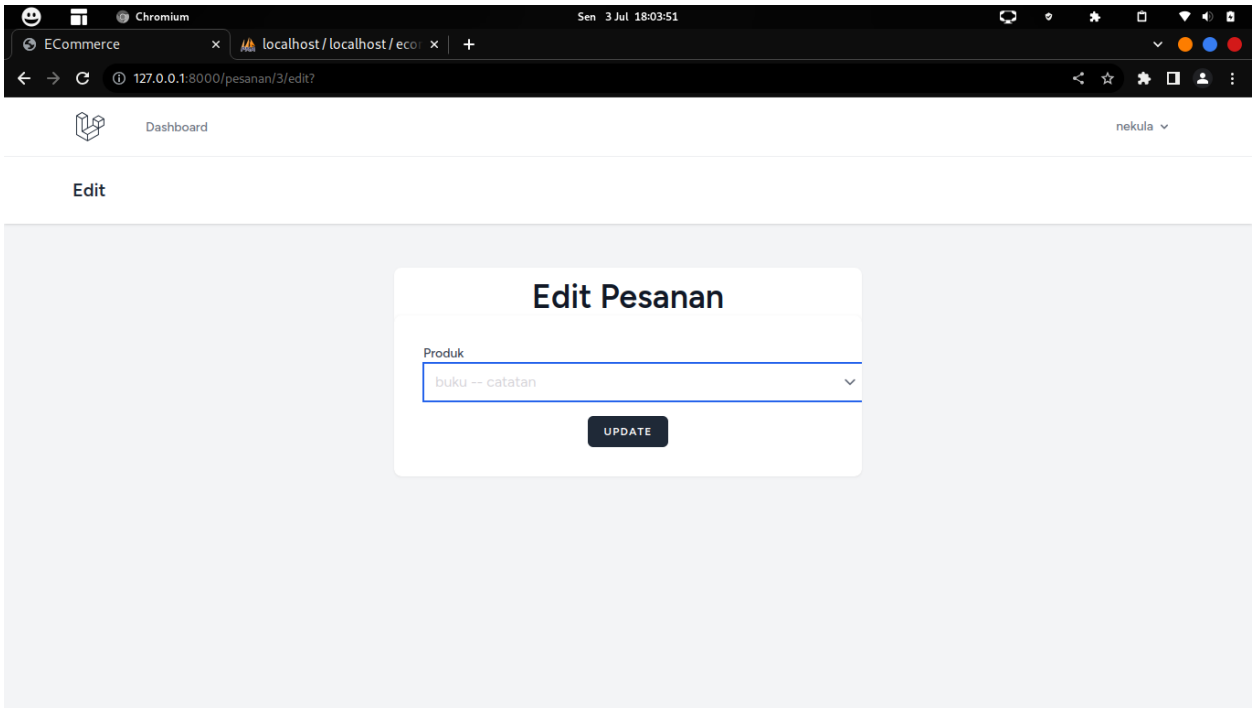
```
        return view('dashboard')->with('data', $data);
    }
    public function create(){
        $produk = Produk::get();
        return view('tambah')->with(['data'=>'', 'prod'=>$produk]);
    }
    public function store(Request $request){
        $cek = Pesanan::create([
            'pelanggan_id' =>Auth::id(),
            'produk_id' => $request->produk
        ]);
        if($cek) return redirect()->route('dashboard');
        return redirect()->route('pesanan.create');
    }
    public function edit($id){
        $data = Pesanan::find($id);
        $data2 = Produk::get();
        return view('edit')->with(['data'=>$data, 'data2'=>$data2,
'prod'=>[]]);
    }
    public function update(Request $request, $id){
        $id = Pesanan::find($id);
        $data = [
            'produk_id'=>$request->produk
        ];
        $cek = $id->update($data);
        if($cek) return redirect()->route('dashboard');
        return redirect()->back();
    }
    public function status($id){
        $data = Pesanan::find($id);
        $stat = $data->pembayaran->status;
        $cek = ($stat == 1) ? $data->pembayaran->update(['status' => 0]) :
$data->pembayaran->update(['status' => 1]);
        if ($cek) {
            return back();
        }
        return redirect()->route('dashboard');
    }
    public function destroy($id){
```

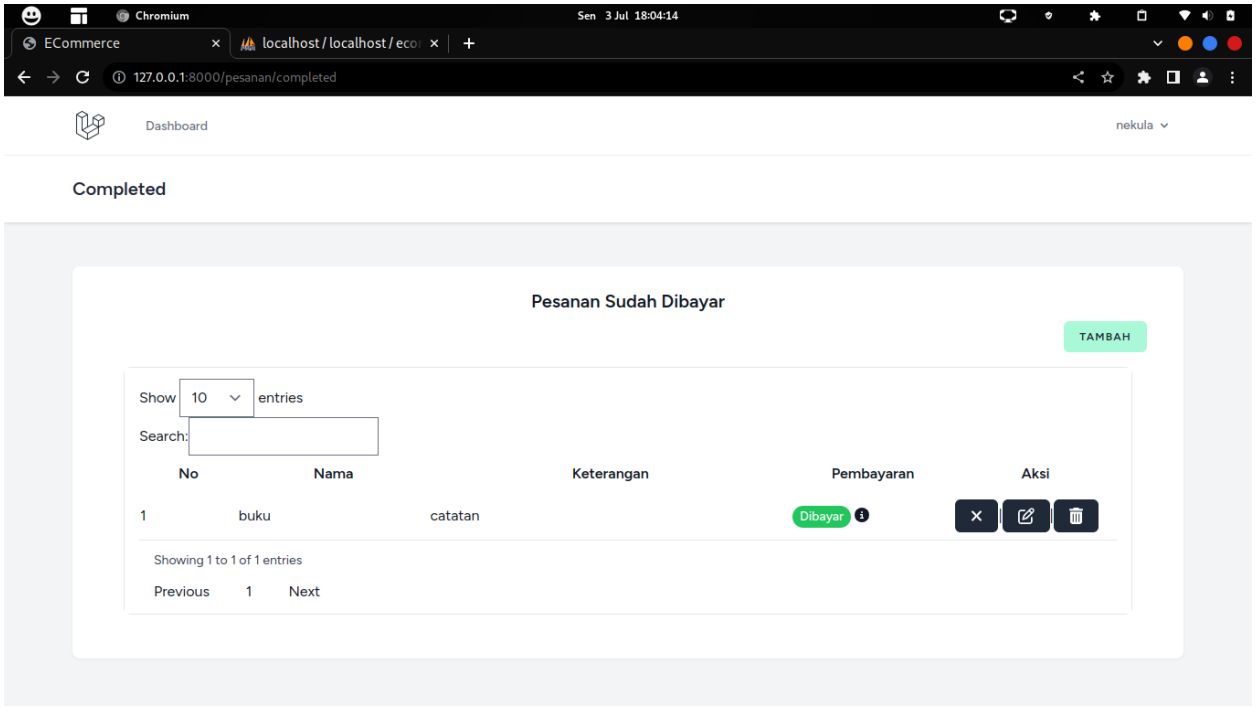
```
        $cek = Pesanan::find($id)->delete();
        if($cek) return back();
        return redirect()->route('dashboard');
    }
    public function show($id){
        if($id=='completed'){
            $data = Pesanan::whereHas('pembayaran', function ($query) {
                $query->where('status', 1);
            })
            ->where('pelanggan_id', Auth::id())
            ->get();
            return view('completed')->with('data', $data);
        }else{
            $data = Pesanan::whereHas('pembayaran', function ($query) {
                $query->where('status', 0);
            })
            ->where('pelanggan_id', Auth::id())
            ->get();
            return view('incomplete')->with('data', $data);
        }
    }
}
```

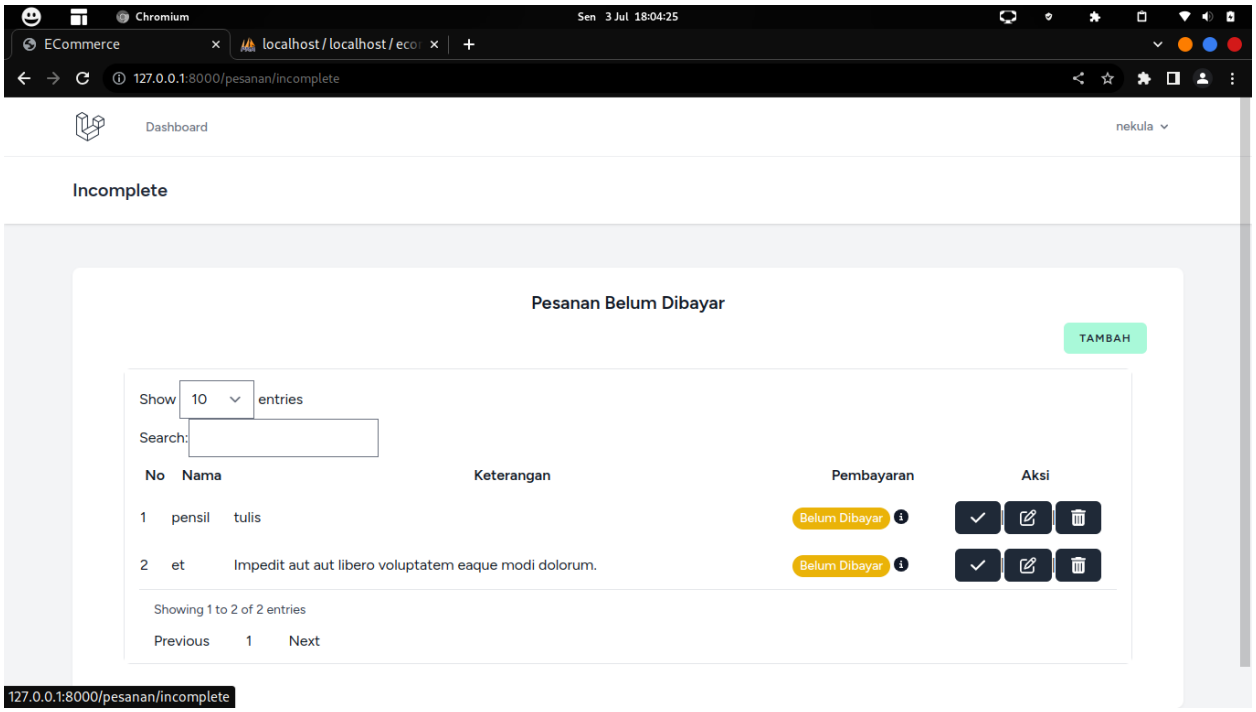

Chromium
Sen 3 Jul 18:02:25
ECommerce
localhost / localhost / eco
127.0.0.1:8000/login
Email
Password
Remember me
Forgot your password?
LOG IN


Chromium
Sen 3 Jul 18:03:03
ECommerce
localhost / localhost / eco
127.0.0.1:8000/pesanan
Dashboard
nekula
Dashboard
Pesanan
TAMBAH
Show 10 entries
Search:
No Nama Keterangan Pembayaran Aksi
1 buku catatan Dibayar
3 et Impedit aut aut libero voluptatem eaque modi dolorum. Belum Dibayar
2 pensil tulis Belum Dibayar
Showing 1 to 3 of 3 entries
Previous1Next


Chromium
Sen 3 Jul 18:03:30
ECommerce
localhost / localhost / eco
127.0.0.1:8000/pesanan/create
Dashboard
nekula
Tambah
Tambah Pesanan
Produk
-- Pilih Produk --
-- Pilih Produk --
buku -- catatan
pensil -- tulis
quo -- Molestiae nesciunt et voluptatem hic id eum a dolore.
eum -- Laborum libero aliquid odit hic quia rerum suscipit.
quisquam -- Commodi repudiandae nihil ea non.
et -- Impedit aut aut libero voluptatem eaque modi dolorum.
quae -- Id ab non non cum eaque fugit nemo.


Chromium
Sen 3 Jul 18:03:51
ECommerce
localhost / localhost / eco
127.0.0.1:8000/pesanan/3/edit?
Dashboard
nekula
Edit
Edit Pesanan
Produk
buku -- catatan
UPDATE





Pembayaran : info

Aksi : edit status, edit, delete