

## Materi Pertemuan 12 - Web Analytics

### Use Case

1. User bisa mengurutkan (sorting) produk berdasarkan best seller
2. User bisa mengurutkan (sorting) produk berdasarkan terbaik (rating tertinggi)
3. User bisa mengurutkan (sorting) produk berdasarkan harga termurah
4. User bisa mengurutkan (sorting) produk berdasarkan harga terbaru
5. User bisa mengurutkan (sorting) produk berdasarkan terbaru
6. Fitur ini bisa dilakukan di public product dan di admin

### Prasarat

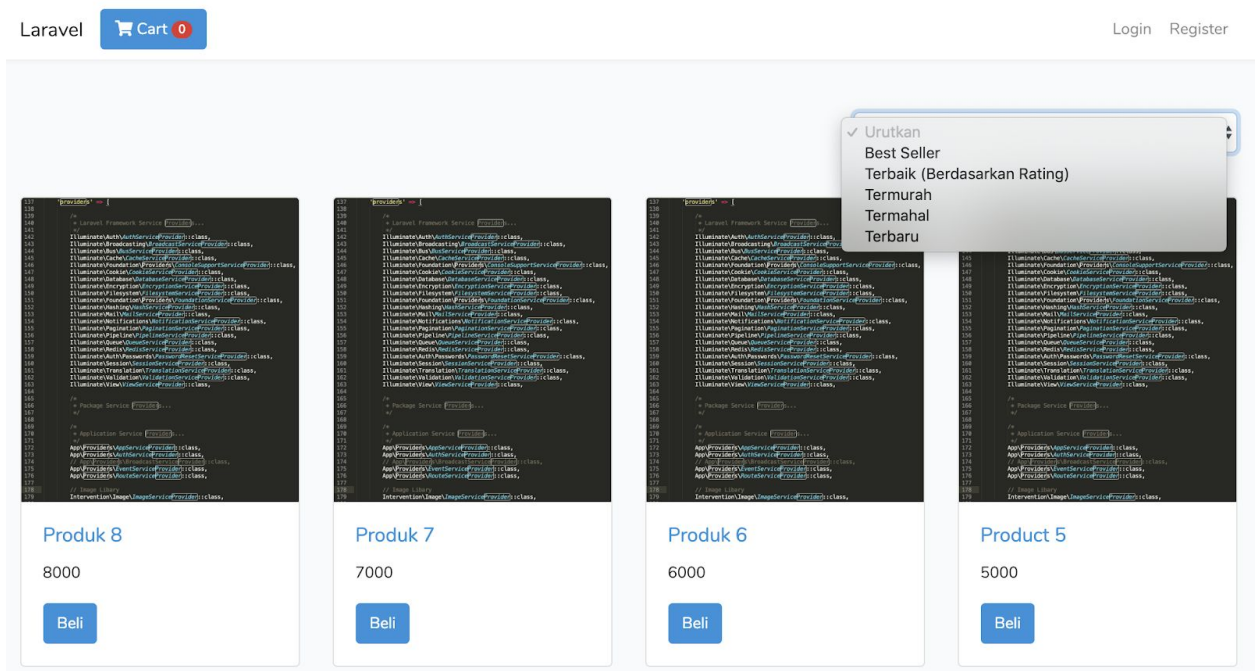
1. Laravel yang digunakan versi 5.8.4
2. Mengerjakan latihan dan tutorial pada 'Materi Pertemuan 11 - Product Review & Social Media Sharing'

### Implement Fitur Sorting

1. Buka file resources/views/products/index.blade.php, code nya seperti dibawah ini (line 4 - 17 dan line 45 - 53)

```
1 @extends('layouts.app')
2 @section('content')
3 <div class="container">
4   <div class="row mt-4">
5     <div class="col-md-4 offset-md-8">
6       <div class="form-group">
7         <select id="order_field" class="form-control">
8           <option value="" disabled selected>Urutkan</option>
9           <option value="best_seller">Best Seller</option>
10          <option value="terbaik">Terbaik (Berdasarkan Rating)</option>
11          <option value="termurah">Termurah</option>
12          <option value="termahal">Termahal</option>
13          <option value="terbaru">Terbaru</option>
14        </select>
15      </div>
16    </div>
17  </div>
18  @foreach($products as $idx => $product)
19    @if ($idx == 0 || $idx % 4 == 0)
20      <div class="row mt-4">
21        @endif
22      <div class="col">
23        <div class="card">
24          
26          <div class="card-body">
27            <h5 class="card-title">
28              <a href="{{ route('products.show', ['id' => $product->id]) }}">
29                {{ $product->name }}
30              </a>
31            </h5>
32            <p class="card-text">
33              {{ $product->price }}
34            </p>
35            <a href="{{ route('carts.add', ['id' => $product->id]) }}" class="btn btn-primary">Beli</a>
36          </div>
37        </div>
38      </div>
39      @if ($idx > 0 && $idx % 4 == 3)
40        </div>
41      @endif
42    @endforeach
43  </div>
44  <!-- JQuery -->
45  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
46  <script type="text/javascript">
47    $(document).ready(function(){
48      $('#order_field').change(function(){
49        window.location.href = '?order_by=' + $(this).val();
50      });
51    });
52  </script>
53  @endsection
```

2. Tampilanya akan seperti ini



3. Buka file `app/Http/Controllers/ProductController.php`, pada method `index` ubah mnejadi seperti ini

```

20  /**
21   * Show the application dashboard.
22   *
23   * @return \Illuminate\Contracts\Support\Renderable
24   */
25  public function index(Request $request)
26  {
27      $productInstance = new Product();
28      $products = $productInstance->orderProducts($request->get('order_by'));
29
30      return view('products.index', compact('products'));
31  }

```

4. Buka file `app/Models/Product.php`, ubah code nya menjadi seprti ini (Line 5)

```

3  namespace App\Models;
4
5  use Illuminate\Support\Facades\DB;
6  use Illuminate\Database\Eloquent\Model;
7

```

(Line 27 - 65)

```
19  /**
20  * Get the product_reviews for the product.
21  */
22  public function productReviews()
23  {
24      return $this->hasMany('App\Models\ProductReview', 'product_id');
25  }
26
27  public function orderProducts($order_by) {
28      // secara default product akan di urutkan berdasarkan created_at
29      $query = 'SELECT * FROM products ORDER BY created_at DESC';
30
31      if ($order_by == 'best_seller') {
32          // best seller
33          // Untuk lebih lanjut bisa pelajari MYSQL
34          // JOIN dan aggregation
35          $query = "SELECT p.*, oi.quantity FROM products AS p
36                  LEFT JOIN (
37                      SELECT product_id, SUM(quantity) as quantity from order_items
38                      GROUP BY product_id
39                  ) AS oi ON oi.product_id = p.id
40                  ORDER BY oi.quantity DESC;";
41
42      } else if ($order_by == 'terbaik') {
43          // terbaik
44          // Untuk lebih lanjut bisa pelajari MYSQL
45          // JOIN dan aggregation
46
47          // NOTE
48          // anda harus mengubah query ini supaya bisa mengurutkan product berdasarkan rating tertinggi
49          $query = "SELECT * FROM products ORDER BY created_at DESC";
50
51      } else if ($order_by == 'termurah') {
52          // termurah
53          $query = "SELECT * FROM products ORDER BY price ASC";
54
55      } else if ($order_by == 'termahal') {
56          // termahal
57          $query = "SELECT * FROM products ORDER BY price DESC";
58
59      } else if ($order_by == 'terbaru') {
60          // terbaru
61          $query = "SELECT * FROM products ORDER BY created_at DESC";
62      }
63
64      return DB::select($query);
65  }
```

5. Silahkan di coba functionality nya.
6. Finish

## Tantangan

1. Implementasi mengurutkan produk berdasarkan terbaik (rating tertinggi)
2. Implementasi fitur pengurutan pada halaman admin produk (hanya produk dari current user logged in saja yang muncul)

## Reference

1. <https://laravel.com/docs/5.8/controllers>
2. <https://laravel.com/docs/5.8/views>
3. <https://laravel.com/docs/5.8/routing>
4. <https://laravel.com/docs/5.8/queries>