

## MODUL MINGGU 10 – IMPLEMENTASI CHAINED SELECTBOX & MORE CUSTOM FILTER DATATABLES

### 1.1 Tujuan Instruksi Khusus

Setelah mengikuti modul ini, mahasiswa diharapkan mampu:

- Mampu membuat fitur chained selectbox dengan AJAX
- Mampu membuat fitur filter data lebih dari 1 parameter menggunakan AJAX Datatables

### 1.2 Instruksi Modul

#### 1.2.1 Setup project

- Siapkan kembali project **cart\_online** yang sudah dibuat sebelumnya menggunakan framework Laravel
- Siapkan file **.env** beserta konfigurasi databasenya (jika belum)
- Jangan lupa menyalakan Apache dan Mysql pada XAMPP
- Buka PHPMyAdmin dengan web browser dan buka database kalian masing-masing
- Import file **indonesia.sql** yang sudah dibagikan kedalam database masing-masing.
- Selanjutnya buka file manager dan masuk ke folder project kalian, klik kanan dan pilih **open git bash here** (pada windows 11 klik **show more option** terlebih dahulu)
- Update project dengan kode baru dari repo dengan perintah pada terminal/git bash :

```
git pull old-origin main
```

- Proses download dan menyatukan kode akan terjadi
- Lakukan git push untuk mengupdate repo kalian masing-masing (opsional)

#### 1.2.2 Implementasi Chained Selectbox

- Buka file Controllers **Keranjang.php** lalu tambahkan kode untuk pemanggilan model-model sebelum deklarasi class sebagai berikut :

```
use App\Models\KecamatanModel;  
use App\Models\KeranjangModel;  
use App\Models\KotaModel;  
use App\Models\ProvinsiModel;  
use Illuminate\Http\Request;
```

- 2) Masih di file yang sama, buat fungsi API untuk mendapatkan data provinsi sebagai berikut :

```
public function get_prov()
{
    $list = ProvinsiModel::select('id', 'name')
        ->orderBy('name','ASC')
        ->get();

    $html_select = '<option value ="">-- Pilih Salah Satu --</option>';
    foreach ($list as $key => $value) {
        $html_select .= '<option value = "'.$value->id.'|'.$value->name.'">'.
        $value->name.'</option>';
    }

    return $html_select;
}
```

- 3) Fungsi API untuk mendapatkan kota sesuai provinsi sebagai berikut :

```
public function get_kota($kode)
{
    $list = KotaModel::select('id', 'name')
        ->where('province_id', $kode)
        ->orderBy('name','ASC')
        ->get();

    $html_select = '<option value ="">-- Pilih Salah Satu --</option>';
    foreach ($list as $key => $value) {
        $html_select .= '<option value = "'.$value->id.'|'.$value->name.'">'.
        $value->name.'</option>';
    }

    return $html_select;
}
```

- 4) Terakhir fungsi API untuk mendapatkan kecamatan sesuai kota sebagai berikut

```
141 public function get_kec($kode)
142 {
143     $list = KecamatanModel::select('id', 'name')
144         ->where('regency_id', $kode)
145         ->orderBy('name','ASC')
146         ->get();
147
148     $html_select = '<option value ="">-- Pilih Salah Satu --</option>';
149     foreach ($list as $key => $value) {
150         $html_select .= '<option value = "'.$value->id.'|'.$value->name.'">'.
            $value->name.'</option>';
151     }
152
153     return $html_select;
154 }
```

- 5) Buka file Views **transaksi/checkout.blade.php** dan tambahkan event onchange pada selectbox provinsi dan kota. Kodenya :

```
<select name="provinsi" id="provinsi" class="form-control" required=""  
onchange="ambil_kota()">
```

dan

```
<select name="kota" id="kota" class="form-control" required=""  
onchange="ambil_kecamatan()">
```

- 6) Kemudian, didalam `@section('script_custom')` tambahkan script untuk pemanggilan AJAX untuk API provinsi, kota, dan kecamatan. Kodenya :

```
<script>  
function ambil_prov() {  
    var link = '{{ url("api/provinsi") }}';  
  
    $.ajax(link, {  
        type: 'GET',  
        success : function (data, status, xhr) {  
            $('#provinsi').html(data);  
        },  
        error : function (jqXHR, textStatus, errorMsg) {  
            alert('Error Pengambilan Data Provinsi : ' + errorMsg);  
        }  
    })  
}  
  
ambil_prov();  
  
function ambil_kota() {  
    var prov = $('#provinsi').val().split("||");  
    var link = '{{ url("api/kota/") }}' + '/' + prov[0];  
  
    ???  
}  
  
function ambil_kecamatan() {  
    var kota = $('#kota').val().split("||");  
    var link = '{{ url("api/kecamatan/") }}' + '/' + kota[0];  
  
    ???  
}  
</script>
```

Lengkapi “???” sesuai screenshot kode berikut :

```
75 function ambil_kota() {
76     var prov = $('#provinsi').val().split("||");
77     var link = '{ url("api/kota/") }' + '/' + prov[0];
78
79     $.ajax(link, {
80         type: 'GET',
81         success : function (data, status, xhr) {
82             $('#kota').html(data);
83         },
84         error : function (jqXHR, textStatus, errorMsg) {
85             alert('Error Pengambilan Data Kota : ' + errorMsg);
86         }
87     })
88 }
89
90 function ambil_kecamatan() {
91     var kota = $('#kota').val().split("||");
92     var link = '{ url("api/kecamatan/") }' + '/' + kota[0];
93
94     $.ajax(link, {
95         type: 'GET',
96         success : function (data, status, xhr) {
97             $('#kecamatan').html(data);
98         },
99         error : function (jqXHR, textStatus, errorMsg) {
100             alert('Error Pengambilan Data Kecamatan : ' + errorMsg);
101         }
102     })
103 }
104 }
```

- 7) Buka file Controllers **Transaksi.php**, ubah fungsi store-nya menjadi sebagai berikut :

```
public function store(Request $request)
{
    //
    $data = KeranjangModel::leftJoin('produk', 'produk.id_produk', '=', 'keranjang.id_produk')
    ->selectRaw('sum(produk.harga * keranjang.jumlah_beli) as total')
    ->first();

    $kecamatan = explode('||', $request->get('kecamatan'));
    $provinsi = explode('||', $request->get('provinsi'));
    $kota = explode('||', $request->get('kota'));

    $trans = new TransaksiModel([
        'tgl_transaksi' => date("Y-m-d H:i:s"),
        'total_bayar' => $data->total,
        'nama' => $request->get('nama'),
        'alamat_jalan' => $request->get('alamat_jalan'),
        'kecamatan' => $kecamatan[1],
        'kota' => $kota[1],
```

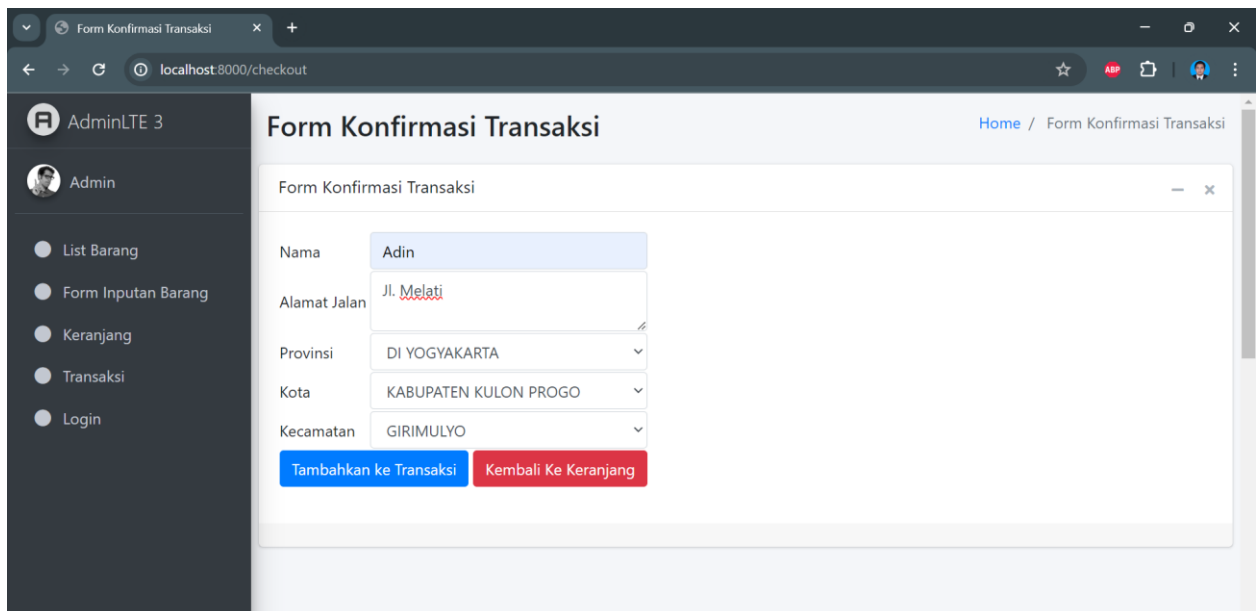
```
'provinsi' => $provinsi[1],  
'id_user' => 1  
]);  
  
$saved = $trans->save();  
  
$all_detail = KeranjangModel::leftJoin('produk', 'produk.id_produk', '=', 'keranjang.id_produk')  
->select('keranjang.*', 'produk.nama_produk', 'produk.harga')  
->get();  
  
foreach($all_detail as $value) {  
    $detail = new DetailTransaksiModel([  
        'id_transaksi' => $trans->id_transaksi,  
        'id_produk' => $value->id_produk,  
        'harga' => $value->harga,  
        'jumlah_beli' => $value->jumlah_beli  
    ]);  
  
    $save_detail = $detail->save();  
}  
  
$keranjang = KeranjangModel::where('id_user', 1)->delete();  
  
return redirect(url('transaksi'));  
}
```

8) Buka file **routes/api.php** dan tambahkan kode route baru di akhir file :

```
Route::get('/provinsi', [Keranjang::class, 'get_prov']);  
Route::get('/kota/{kode}', [Keranjang::class, 'get_kota']);  
Route::get('/kecamatan/{kode}', [Keranjang::class, 'get_kec']);
```

9) Lakukan Git add, Git Commit, dan Git Push

10) Jalankan php artisan serve melalui command prompt dan ujicoba checkout keranjang



### 1.2.3 Custom Pencarian lebih dari 1 parameter

- 1) Buka file Views **transksi/list.blade.php**, tambahkan kode form sebelum <table>. Kodenya:

```
<form>
  <div class="row">
    <div class="col">
      <label>Tanggal Awal</label>
      <input class="form-control" type="date" name="tgl_awal" id="tgl_awal">
    </div>
    <div class="col">
      <label>Tanggal Akhir</label>
      <input class="form-control" type="date" name="tgl_akhir" id="tgl_akhir">
    </div>
    <div class="col">
      <label>Provinsi</label>
      <input class="form-control" type="text" name="prov" id="prov">
    </div>
    <div class="col">
      <label>Min. Nominal</label>
      <input class="form-control" type="number" name="min_total" id="min_total">
    </div>
    <div class="col">
      <input type="submit" value="Cari" class="btn btn-primary" style="margin-top:33px" id="">
    </div>
  </div>
</form>
```

- 2) Masih di file yang sama, ubah fungsi pemanggilan AJAX-nya menjadi seperti berikut :

```
var tabel = $("#data-list").DataTable({
  "processing": true,
  "serverSide": true,
  "ajax": {
    url: url,
    data: function (d) {
      d.tgl_awal = $("#tgl_awal").val();
      d.tgl_akhir = $("#tgl_akhir").val();
      d.prov = $("#prov").val();
      d.min_total = $("#min_total").val();
    }
  },
});
```

- 3) Buka file **Controllers Transaksi.php** dan tambahkan kode-kode berikut pada fungsi dataDatables untuk pemanggilan variable request dan juga Query pencarian :

```
$tgl_awal = $request->query('tgl_awal');
$tgl_akhir = $request->query('tgl_akhir');
$prov = $request->query('prov');
$min_total = $request->query('min_total');
```

Lalu,

```
if ($tgl_awal != " && $tgl_awal != null) {  
    $data_db_filtered = $data_db_filtered->whereRaw('DATE(tgl_transaksi) >= "'. $tgl_awal.'");  
}  
  
if ($tgl_akhir != " && $tgl_akhir != null) {  
    $data_db_filtered = $data_db_filtered->whereRaw('DATE(tgl_transaksi) <= "'. $tgl_akhir.'");  
}  
  
if ($prov != " && $prov != null) {  
    $data_db_filtered = $data_db_filtered->where('provinsi', 'like', '%'.$prov.'%');  
}  
  
if ($min_total != " && $min_total != null) {  
    $data_db_filtered = $data_db_filtered->where('total_bayar', '>=', $min_total);  
}
```

Dan

```
if ($tgl_awal != " && $tgl_awal != null) {  
    $data_db = $data_db->whereRaw('DATE(tgl_transaksi) >= "'. $tgl_awal.'");  
}  
  
if ($tgl_akhir != " && $tgl_akhir != null) {  
    $data_db = $data_db->whereRaw('DATE(tgl_transaksi) <= "'. $tgl_akhir.'");  
}  
  
if ($prov != " && $prov != null) {  
    $data_db = $data_db->where('provinsi', 'like', '%'.$prov.'%');  
}  
  
if ($min_total != " && $min_total != null) {  
    $data_db = $data_db->where('total_bayar', '>=', $min_total);  
}
```

- 4) Lakukan Git add, Git Commit, dan Git Push
- 5) Jalankan php artisan serve melalui command prompt dan ujicoba halaman transaksi

The screenshot shows a web application interface for 'List Transaksi'. On the left is a dark sidebar with the user 'Admin' and a menu containing 'List Barang', 'Form Inputan Barang', 'Keranjang', 'Transaksi', and 'Login'. The main content area has a light blue header with 'List Transaksi' and a 'Home / List Transaksi' breadcrumb. Below the header is a filter section with inputs for 'Tanggal Awal' (12/06/2024), 'Tanggal Akhir' (13/06/2024), 'Provinsi' (Bali), and 'Min. Nominal'. A 'Cari' button is next to the 'Min. Nominal' input. Below the filters, it says 'Show 10 entries' and 'Search:'. A table displays one transaction entry. The table has columns: No., Tanggal, Nama, Alamat, and Nominal. The entry shows '1' for No., '12-Jun-2024 18:42:51' for Tanggal, 'Adin' for Nama, 'panggungan KUTA SELATAN KABUPATEN BADUNG BALI' for Alamat, and 'Rp 8,000' for Nominal. At the bottom, it says 'Showing 1 to 1 of 1 entries (filtered from 2 total entries)' and has 'Previous', '1', and 'Next' pagination links.

**List Transaksi**

Tanggal Awal: 12/06/2024 Tanggal Akhir: 13/06/2024 Provinsi: Bali Min. Nominal: Cari

Show 10 entries Search:

No.	Tanggal	Nama	Alamat	Nominal
1	12-Jun-2024 18:42:51	Adin	panggungan KUTA SELATAN KABUPATEN BADUNG BALI	Rp 8,000

Showing 1 to 1 of 1 entries (filtered from 2 total entries) Previous 1 Next