

LAPORAN PRAKTIKUM II

LARAVEL

Disusun untuk Memenuhi Mata Pelajaran Praktikum Pemrograman Web
Dibimbing oleh Bapak Ricky Septiandy Pratama Putra Husmildin, S.Pd.

Oleh :

Dimas Yuda Pratama (17691)

XI TKJ 2



SEKOLAH MENENGAH KEJURUAN NEGERI 2 MALANG
KONSENTRASI KEAHLIAN TEKNIK KOMPUTER DAN JARINGAN
MARET 2025

A. LATIHAN

1. Persiapan Database:

- Kode program

```
C:\xampp\htdocs\penjualan-chart> php artisan make:migration create_penjualans_table
```

```
database > migrations > 2025_03_29_162549_create_penjualans_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      public function up(): void
9      {
10         Schema::create(table: 'penjualans', callback: function (Blueprint $table): void {
11             $table->id();
12             $table->date(column: 'tanggal');
13             $table->decimal(column: 'total_penjualan', total: 10, places: 2);
14             $table->timestamps();
15         });
16     }
17
18     public function down(): void
19     {
20         Schema::dropIfExists(table: 'penjualans');
21     }
22 };
```

```
PS C:\xampp\htdocs\penjualan-chart> php artisan migrate
```

```
PS C:\xampp\htdocs\penjualan-chart> php artisan tinker
Psy Shell v0.12.8 (PHP 8.2.12 - cli) by Justin Hileman
> use App\Models\Penjualan;
> use Carbon\Carbon;
>
> // Hapus semua data penjualan
> Penjualan::truncate();
= Illuminate\Database\Eloquent\Builder {#5196
    +pendingAttributes: [],
}
>
> $kategori = ['Makanan Berat', 'Minuman', 'Snack'];
= [
    "Makanan Berat",
    "Minuman",
    "Snack",
]
>
> $now = Carbon::now();
= Carbon\Carbon @1744222208 {#1303
    date: 2025-04-09 18:10:08.201338 UTC (+00:00),
}
>
> // === 7 Hari Terakhir ===
> for ($i = 6; $i >= 0; $i--) {
    .   Penjualan::create([
    .       'tanggal' => $now->copy()->subDays($i)->format('Y-m-d'),
    .       'kategori' => $kategori[array_rand($kategori)],
    .       'total_penjualan' => rand(10000, 50000),
    .   ]);
    . }
>
```

- Keterangan
 - Buat migrasi untuk tabel 'penjualan'
 - Buat tabel 'penjualan' dengan kolom: id, tanggal, total_penjualan
 - Isi data dummy untuk seminggu terakhir

2. Membuat Model dan Controller:

- Kode program

```
PS C:\xampp\htdocs\penjualan-chart> php artisan make:model Penjualan
PS C:\xampp\htdocs\penjualan-chart> php artisan make:controller PenjualanController
```

```
use App\Models\Penjualan;
use Carbon\Carbon;

class PenjualanController extends Controller
{
    public function index()
    {
        $penjualanSeminggu = Penjualan::whereBetween('tanggal', [Carbon::now()->subDays(6)
            ->orderBy('tanggal')
            ->get();

        $labels = $penjualanSeminggu->pluck('tanggal')->map(function($tanggal) {
            return Carbon::parse($tanggal)->format('d/m');
        });
        $data = $penjualanSeminggu->pluck('total_penjualan');

        return view('penjualan.chart', compact('labels', 'data'));
    }
}
```

- Keterangan
 - Buat model Penjualan
 - Buat PenjualanController untuk mengambil data penjualan
 - Edit file PenjualanController

3. Integrasi Chart.js:

- Kode program

```
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
```

- Keterangan
 - Install Chart.js ke layout utama (resources/views/layouts/app.blade.php) menggunakan npm atau tambahkan CDN
 - Buat view untuk menampilkan grafik

4. Implementasi Chart:

- Kode program

```
@extends('layouts.app')

@section('content')
<div class="container">
    <h1>Grafik Penjualan Kantin Seminggu Terakhir</h1>
    <canvas id="penjualanChart"></canvas>
</div>

<script>
var ctx = document.getElementById('penjualanChart').getContext('2d');
var chart = new Chart(ctx, {
    type: 'bar',
    data: {
        labels: @json($labels),
        datasets: [{
            label: 'Total Penjualan',
            data: @json($data),
            backgroundColor: 'rgba(75, 192, 192, 0.6)',
            borderColor: 'rgba(75, 192, 192, 1)',
            borderWidth: 1
        }]
    },
    options: {
        scales: {
            y: {
                beginAtZero: true
            }
        }
    }
});
</script>
@endsection
```

- Keterangan
 - Buat view baru di resources/views/penjualan/chart.blade.php
 - Ambil data dari database melalui controller
 - Kirim data ke view
 - Buat grafik menggunakan Chart.js di view

#untuk output program dari project ini akan saya tampilkan dibagian studi kasus

B. STUDI KASUS

1. Setelah menyelesaikan grafik dasar, coba tambahkan fitur-fitur berikut :

- Tambahkan filter untuk memilih rentang tanggal
- Buat grafik pie untuk menampilkan persentase penjualan per kategori makanan
- Implementasikan grafik garis untuk membandingkan penjualan bulan ini dengan bulan sebelumnya

HASIL

1. Kode Program:

#membuat kategori migrations

```
PS C:\xampp\htdocs\penjualan-chart> php artisan make:migration add_kategori_to_penjualans_table --table=penjualans

data\base\migrations\2025_04_06_172034_add_kategori_to_penjualans_table.php > ...
chart\data\base\migrations\2025_03_29_162549_create_penjualans_table.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     public function up(): void
10    {
11        Schema::table('penjualans', callback: function (Blueprint $table): void {
12            $table->string(column: 'kategori')->after(column: 'tanggal');
13        });
14    }
15
16    public function down(): void
17    {
18        Schema::table('penjualans', callback: function (Blueprint $table): void {
19            $table->dropColumn(columns: 'kategori');
20        });
21    }
22
23 };
```

#controller

```
app > Http > Controllers > PenjualanController.php > PHP > PenjualanController > index()
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Penjualan;
6  use Carbon\Carbon;
7  use Illuminate\Http\Request;
8  use Illuminate\Support\Facades\DB;
9
10 5 references | 0 implementations
11 class PenjualanController extends Controller
12 {
13     2 references | 0 overrides
14     public function index(Request $request): View
15     {
16         $startDate = $request->input(key: 'tanggal_mulai');
17         $endDate = $request->input(key: 'tanggal_akhir');
18
19         if (!$startDate || !$endDate) {
20             $startDate = Carbon::now()->subDays(value: 6)->format(format: 'Y-m-d');
21             $endDate = Carbon::now()->format(format: 'Y-m-d');
22         }
23
24         // === Data untuk tabel berdasarkan filter tanggal ===
25         $filteredData = Penjualan::whereBetween(column: 'tanggal', values: [$startDate, $endDate])
26             ->orderBy(column: 'tanggal')
27             ->get();
28
29         // === GRAFIK BAR ===
30         $barData = $filteredData;
31
32         $labels = $barData->pluck(value: 'tanggal')
33             ->map(callback: fn($tanggal): string => Carbon::parse(time: $tanggal)->format(format: 'd/m/Y'))
34             ->toArray();
35         $data = $barData->pluck(value: 'total_penjualan')->toArray();
36
37         // === GRAFIK PIE ===
38         $kategori = Penjualan::select(columns: 'kategori', DB::raw(value: 'SUM(total_penjualan) as total'))
39             ->whereBetween(column: 'tanggal', values: [$startDate, $endDate])
40             ->groupBy(groups: 'kategori')
41             ->get();
42
43         $kategoriLabels = $kategori->pluck(value: 'kategori')->toArray();
44         $kategoriData = $kategori->pluck(value: 'total')->toArray();
45
46         // === GRAFIK GARIS ===
47         $startThisMonth = Carbon::now()->startOfMonth();
48         $endThisMonth = Carbon::now()->endOfMonth();
49
50         $startLastMonth = Carbon::now()->subMonth()->startOfMonth();
```

```

49     $sendLastMonth = Carbon::now()->subMonth()->endOfMonth();
50
51     $labelline = [];
52     for ($date = $startThisMonth->copy(); $date <= $endThisMonth; $date->addDay()) {
53         $labelline[] = $date->format(format: 'd/m');
54     }
55
56     $penjualanBulanIni = Penjualan::whereBetween(column: 'tanggal', values: [$startThisMonth, $endThisMonth])
57         ->select(columns: DB::raw(value: 'DATE(tanggal) as tanggal'), DB::raw(value: 'SUM(total_penjualan) as total'))
58         ->groupBy(groups: DB::raw(value: 'DATE(tanggal)'))
59         ->pluck(column: 'total', key: 'tanggal');
60
61     $dataBulanIni = [];
62     foreach ($labelline as $day) {
63         $tanggalFormat = Carbon::createFromFormat(format: 'd/m', time: $day)->format(format: 'Y-m-d');
64         $dataBulanIni[] = $penjualanBulanIni[$tanggalFormat] ?? 0;
65     }
66
67     $penjualanBulanLalu = Penjualan::whereBetween(column: 'tanggal', values: [$startLastMonth, $endLastMonth])
68         ->select(columns: DB::raw(value: 'DATE(tanggal) as tanggal'), DB::raw(value: 'SUM(total_penjualan) as total'))
69         ->groupBy(groups: DB::raw(value: 'DATE(tanggal)'))
70         ->pluck(column: 'total', key: 'tanggal');
71
72     $dataBulanLalu = [];
73     foreach ($labelline as $day) {
74         $tanggalLalu = Carbon::createFromFormat(format: 'd/m', time: $day)->subMonth()->format(format: 'Y-m-d');
75         $dataBulanLalu[] = $penjualanBulanLalu[$tanggalLalu] ?? 0;
76     }
77
78     return view(view: 'penjualan.chart', data: compact(
79         var_name: 'labels', var_names: 'data',
80         'kategoriLabels', 'kategoriData',
81         'dataBulanIni', 'dataBulanLalu',
82         'labelline',
83         'startDate', 'endDate',
84         'filteredData'
85     ))->with(key: 'salesData', value: $filteredData);
86
87
88 // === TAMPILKAN FORM INPUT ===
89 1 reference | 0 overrides
90 public function form(): View
91 {
92     return view(view: 'penjualan.form');
93 }

```

```

94 // === SIMPAN DATA DARI FORM ===
95 1 reference | 0 overrides
96 public function store(Request $request): RedirectResponse
97 {
98     $request->validate(rules: [
99         'tanggal' => 'required|date',
100         'total_penjualan' => 'required|numeric',
101         'kategori' => 'required|string|max:50',
102     ]);
103
104     Penjualan::create(attributes: [
105         'tanggal' => $request->tanggal,
106         'total_penjualan' => $request->total_penjualan,
107         'kategori' => $request->kategori,
108     ]);
109
110     return redirect()->route(route: 'penjualan.form')->with(key: 'success', value: 'Data penjualan berhasil ditambahkan!');
111 }

```

#app/models/penjualan.php

```
app > Models > Penjualan.php > PHP Intelephense > Penjualan
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  6 references | 0 implementations
9  class Penjualan extends Model
10
11      use HasFactory;
12
13      0 references
14      protected $table = 'penjualans'; // Nama tabel di database
15
16      0 references
17      protected $fillable = ['tanggal', 'total_penjualan', 'kategori']; // Sesuaikan dengan kolom di database
```

#route/web.php

```
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\PenjualanController;
5
6  // Redirect root (/) ke /penjualan/chart
7  Route::redirect(uri: '/', destination: '/penjualan/chart');
8
9  // Route lainnya
10 Route::get(uri: '/penjualan/chart', action: [PenjualanController::class, 'index'])->name(name: 'penjualan.chart');
11 Route::get(uri: '/penjualan/form', action: [PenjualanController::class, 'form'])->name(name: 'penjualan.form');
12 Route::get(uri: '/penjualan', action: [PenjualanController::class, 'index'])->name(name: 'penjualan.index');
13 Route::post(uri: '/penjualan/form', action: [PenjualanController::class, 'store'])->name(name: 'penjualan.store');
14
```


#view/layouts/app.blade.php

```
Resources > Views > layouts > app.blade.php > HTML > head > style > header h1
1 <!DOCTYPE html>
2 <html lang="id">
3
4 <head>
5 <meta charset="UTF-8" />
6 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7 <title>Kantin Sekolah</title>
8 <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
9 <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;600&display=swap" rel="stylesheet">
10 <style>
11 body {
12   margin: 0;
13   font-family: 'Poppins', sans-serif;
14   background-color: #ffffff;
15   text-align: center;
16   padding-top: 100px;
17 }
18
19 .header {
20   background: linear-gradient(135deg, #ff7b00, #ffae42);
21   padding: 20px 40px;
22   display: flex;
23   align-items: center;
24   justify-content: space-between;
25   color: white;
26   box-shadow: 0 4px 12px rgba(0, 0, 0, 0.2);
27   z-index: 10;
28   position: fixed;
29   top: 0;
30   left: 0;
31   right: 0;
32 }
33
34 .header-left {
35   display: flex;
36   align-items: center;
37   gap: 15px;
38 }
39
40 .header-img {
41   width: 50px;
42   height: 60px;
43   filter: drop-shadow(3px 3px 8px rgba(0, 0, 0, 0.5));
44 }
45
46 .header-h1 {
47   margin: 0;
48   font-size: 20px;
49   font-weight: 600;
50   text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.4);
```

```
51 }
52
53 .navbar {
54   display: flex;
55   gap: 20px;
56 }
57
58 .navbar a {
59   position: relative;
60   color: white;
61   text-decoration: none;
62   font-weight: 500;
63   padding: 8px 16px;
64   border: 2px solid rgba(255, 255, 255, 0.56);
65   border-radius: 8px;
66   background: rgba(255, 255, 255, 0.1);
67   overflow: hidden;
68   z-index: 0;
69 }
70
71 .navbar a::before {
72   content: "";
73   position: absolute;
74   top: -20px;
75   left: -20px;
76   right: -20px;
77   bottom: -20px;
78   background: conic-gradient(from 0deg,
79     #ff7b00,
80     #ffc107,
81     #ff7b00,
82     #ffc107,
83     #ff7b00);
84   z-index: -2;
85   animation: rotateGlow 3s linear infinite;
86   filter: blur(3px);
87   opacity: 0.8;
88 }
89
90 .navbar a::after {
91   content: "";
92   position: absolute;
93   inset: 0;
94   background: rgba(255, 255, 255, 0.1);
```

```

95     border-radius: 8px;
96     z-index: -1;
97 }
98
99 .navbar a:hover {
100     transform: scale(1.05);
101     box-shadow: 0 4px 20px #rgb(255, 255, 255);
102 }
103
104 .main-content {
105     padding: 30px 20px;
106 }
107
108 .footer {
109     background: linear-gradient(135deg, #ffae42, #ff47b0);
110     padding: 15px;
111     color: white;
112     width: 100%;
113     box-shadow: 0 -4px 20px #rgba(0, 0, 0, 0.2);
114     bottom: 0;
115 }
116
117 @keyframes rotateGlow {
118     0% {
119         transform: rotate(0deg);
120     }
121
122     100% {
123         transform: rotate(360deg);
124     }
125 }
126
127 </style>
128 </head>
129
130 <body>
131     <!-- Header -->
132     <div class="header">
133         <div class="header-left">
134             
135             <h1>Grafik Penjualan Kantin Sekolah</h1>
136         </div>
137         <div class="navBar">
138             <a href="/penjualan/chart" class="nav-link active">Chart</a>
139             <a href="/penjualan/form" class="nav-link">Form</a>
140         </div>
141     </div>
142
143     <!-- Main Content -->
144     <div class="main-content" id="chart">
145         @yield(section: 'content')
146     </div>
147
148     <!-- Footer -->
149     <div class="footer">
150         <p>&copy; 2025 Kantin Sekolah. SMK Negeri 2 Malang.</p>
151     </div>
152
153     <!-- Scripts -->
154     @yield(section: 'scripts')
155 </body>
156
157 </html>

```

#view/penjualan/chart.blade.php

```
resources > views > penjualan > chart.blade.php > div.container.mt-4.
1  @extends(view: 'layouts.app')
2
3  @section(section: 'content')
4
5  <div class="container mt-4">
6      <!-- FILTER TANGGAL -->
7      <div class="filter-container">
8          <div class="filter-box mb-4">
9              <form method="GET" action="{{ route(name: 'penjualan.index') }}">
10                  <div class="filter-row">
11                      <div class="filter-group">
12                          <label for="tanggal_mulai">Dari Tanggal:</label>
13                          <input type="date" id="tanggal_mulai" name="tanggal_mulai" value="{{ request(key: 'tanggal_mulai',
14                          default: now()->subDays(value: 7)->format(format: 'Y-m-d')) }}">
15                      </div>
16
17                      <div class="filter-group">
18                          <label for="tanggal_akhir">Sampai Tanggal:</label>
19                          <input type="date" id="tanggal_akhir" name="tanggal_akhir" value="{{ request(key: 'tanggal_akhir',
20                          default: now()->format(format: 'Y-m-d')) }}">
21                      </div>
22
23                      <div class="filter-buttons">
24                          <button type="submit" class="btn-filter">Filter</button>
25                          <a href="{{ route(name: 'penjualan.index') }}" class="btn-reset">Reset</a>
26                      </div>
27                  </div>
28              </form>
29          </div>
30      </div>
```

```
32  <!-- DATA PENJUALAN -->
33  <div class="sales-data mb-5">
34      <h4>Data Penjualan</h4>
35      <div class="table-responsive">
36          <table class="table table-bordered table-striped">
37              <thead class="table-warning">
38                  <tr>
39                      <th>#</th>
40                      <th>Tanggal</th>
41                      <th>Kategori</th>
42                      <th>Total (Rp)</th>
43                  </tr>
44              </thead>
45              <tbody>
46                  @if(isset($salesData) && count(value: $salesData) > 0)
47                      @foreach($salesData as $index => $sale)
48                          <tr>
49                              <td>{{ $index + 1 }}</td>
50                              <td>{{ \Carbon\Carbon::parse(time: $sale->tanggal ?? now())->format(format: 'd-m-Y') }}</td>
51                              <td>{{ $sale->kategori ?? '-' }}</td>
52                              <td>Rp {{ number_format(num: $sale->total_penjualan ??
53                              0, decimals: 0, decimal_separator: ',', thousands_sepa...') }}</td>
54                          </tr>
55                      @endforeach
56                  @else
57                      <tr>
58                          <td colspan="4" class="text-center">Tidak ada data penjualan untuk periode yang dipilih.</td>
59                      </tr>
60                  @endif
61              </tbody>
62              <tfoot class="table-warning">
63                  <tr>
64                      <th colspan="2">Total</th>
65                      <th></th>
66                      <th>Rp {{ number_format(num: isset($salesData) ? $salesData->sum('total_penjualan') ??
67                      0 : 0, decimals: 0, decimal_separator: ',', thousands_sepa...') }}</th>
68                  </tr>
69              </tfoot>
70          </table>
71      </div>
```

```

72
73 <div class="chart-container">
74   <div class="chart-box">
75     <h3>Grafik Batang: Penjualan 7 Hari Terakhir</h3>
76     <canvas id="barChart"></canvas>
77   </div>
78
79   <div class="chart-box">
80     <h3>Grafik Garis: Perbandingan Penjualan Bulanan</h3>
81     <canvas id="lineChart"></canvas>
82   </div>
83
84   <div class="chart-box">
85     <h3>Grafik Pie: Persentase Penjualan per Kategori</h3>
86     <canvas id="pieChart"></canvas>
87   </div>
88 </div>
89 </div>
90
91 <style>
92   @import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;600&display=swap');
93
94   body {
95     background-color: #235556;
96     font-family: 'Poppins', sans-serif;
97     text-align: center;
98   }
99
100   .filter-container {
101     margin-bottom: 30px;
102     display: flex;
103     justify-content: center;
104   }
105
106   .filter-box {
107     background: #f0f0f0;
108     padding: 25px;
109     width: 70%;
110     border-radius: 12px;
111     box-shadow: 0 4px 8px #0000000a;
112   }
113
114   .filter-row {
115     display: flex;
116     flex-wrap: wrap;
117     gap: 20px;
118     justify-content: center;
119     align-items: flex-end;
120   }
121
122   .filter-group {
123     display: flex;
124     flex-direction: column;
125     align-items: flex-start;
126     min-width: 200px;
127   }
128
129   .filter-group label {
130     font-weight: 600;
131     margin-bottom: 5px;

```

```

132     color: #e65100;
133 }
134
135 .filter-group input {
136     width: 90%;
137     padding: 10px;
138     border: 1px solid #ffa726;
139     border-radius: 6px;
140     background-color: #fff8e1;
141 }
142
143 .filter-buttons {
144     display: flex;
145     gap: 10px;
146 }
147
148 .btn-filter {
149     background-color: #fb8c00;
150     color: white;
151     padding: 10px 20px;
152     border: none;
153     border-radius: 6px;
154     font-weight: bold;
155     cursor: pointer;
156     transition: all 0.3s ease;
157 }
158
159 .btn-reset {
160     background-color: #e0e0e0;
161     color: #424242;
162     padding: 10px 20px;
163     border: none;
164     border-radius: 6px;
165     font-weight: bold;
166     cursor: pointer;
167     text-decoration: none;
168     transition: all 0.3s ease;
169 }
170
171 .btn-filter:hover {
172     background-color: #ef6c00;
173     transform: translateY(-2px);
174 }
175
176 .btn-reset:hover {
177     background-color: #bdbdbd;
178     transform: translateY(-2px);
179 }
180
181 .sales-data {
182     margin: 0 auto;
183     background: #ffcc80;
184     width: 70%;
185     padding: 25px;
186     border-radius: 12px;
187     box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
188     margin-bottom: 30px;
189 }
190

```

```

191 .sales-data h4 {
192     margin-bottom: 20px;
193     font-weight: 600;
194     color: #e65100;
195 }
196
197 .table {
198     width: 100%;
199     border-collapse: collapse;
200 }
201
202 .table th,
203 .table td {
204     padding: 10px;
205     text-align: left;
206 }
207
208 .table-warning {
209     background-color: #ffe082;
210 }
211
212 .table-striped tbody tr:nth-of-type(odd) {
213     background-color: rgba(255, 248, 225, 0.5);
214 }
215
216 .chart-container {
217     display: flex;
218     flex-direction: column;
219     gap: 30px;
220     align-items: center;
221 }
222
223 .chart-box {
224     background: #fffcc8;
225     padding: 20px;
226     border-radius: 10px;
227     box-shadow: 2px 2px 10px rgba(0, 0, 0, 0.1);
228     width: 70%;
229 }
230
231 table {
232     font-size: 14px;
233     text-align: left;
234 }
235
236 @media (max-width: 768px) {
237     .filter-row {
238         flex-direction: column;
239         align-items: stretch;
240     }
241
242     .filter-group {
243         width: 100%;
244     }
245
246     .chart-box {
247         width: 95%;
248     }
249 }
250 </style>

```



```

251
252 <script>
253   var labels = @json(value: $labels ?? []);
254   var data = @json(value: $data ?? []);
255   var dataBulanIni = @json(value: $dataBulanIni ?? []);
256   var dataBulanLalu = @json(value: $dataBulanLalu ?? []);
257   var kategoriLabels = @json(value: $kategoriLabels ?? []);
258   var kategoriData = @json(value: $kategoriData ?? []);
259
260   document.addEventListener('DOMContentLoaded', function() {
261     if (document.getElementById('barChart')) {
262       new Chart(document.getElementById('barChart'), {
263         type: 'bar',
264         data: {
265           labels: labels,
266           datasets: [{
267             label: 'Total Penjualan (Rp)',
268             data: data,
269             backgroundColor: 'rgba(230, 81, 0, 0.6)',
270             borderColor: 'rgba(230, 81, 0, 1)',
271             borderWidth: 1
272           }]
273         },
274         options: {
275           responsive: true,
276           scales: {
277             y: {
278               beginAtZero: true
279             }
280           }
281         }
282       });
283     }
284
285     if (document.getElementById('lineChart')) {
286       new Chart(document.getElementById('lineChart'), {
287         type: 'line',
288         data: {
289           labels: labels,
290           datasets: [{
291             label: 'Bulan Ini',
292             data: dataBulanIni,
293             backgroundColor: 'rgba(33, 150, 243, 0.4)',
294             borderColor: 'rgba(33, 150, 243, 1)',
295             borderWidth: 2
296           },
297           {
298             label: 'Bulan Lalu',
299             data: dataBulanLalu,
300             backgroundColor: 'rgba(76, 175, 80, 0.4)',
301             borderColor: 'rgba(76, 175, 80, 1)',
302             borderWidth: 2
303           }
304         ],
305         options: {
306           responsive: true,
307           scales: {
308             y: {
309               beginAtZero: true
310             }
311           }
312         }
313       });
314     }
315   });
316 </script>

```

```

312     }
313   }
314   });
315 }
316
317 if (document.getElementById('pieChart')) {
318   new Chart(document.getElementById('pieChart'), {
319     type: 'pie',
320     data: {
321       labels: kategoriLabels,
322       datasets: [{
323         data: kategoriData,
324         backgroundColor: [
325           '#ff8a65', '#ff7043', '#ff5722', '#e64a19',
326           '#bf360c', '#d84315', '#ff6e40'
327         ]
328       }]
329     },
330     options: {
331       responsive: true
332     }
333   });
334 }
335 });
336 </script>
337
338 @endsection

```


#membuat view form input

```
resources > views > penjualan > form.blade.php > ...
1  @extends(view: 'layouts.app')
2
3  @section(section: 'content')
4  <style>
5      @import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600&display=swap');
6
7      html,
8      body {
9          margin: 0;
10         padding: 0;
11         min-height: 100%;
12         font-family: 'Poppins', sans-serif;
13         background: #rgb(235, 136, 37);
14         display: flex;
15         flex-direction: column;
16     }
17
18     .form-container {
19         flex: 1;
20         display: flex;
21         align-items: center;
22         justify-content: center;
23         padding: 2rem;
24         padding-bottom: 100px;
25     }
26
27     .form-box {
28         background: #rgb(255, 163, 58);
29         padding: 0;
30         overflow: hidden;
31         display: flex;
32         flex-direction: column;
33         border-radius: 1rem;
34         box-shadow: 0 12px 28px #rgba(0, 0, 0, 0.1);
35         width: 100%;
36         max-width: 500px;
37         transition: all 0.3s ease-in-out;
38     }
39
40     .form-box:hover {
41         transform: translateY(-3px);
42         box-shadow: 0 18px 38px #rgba(0, 0, 0, 0.15);
43     }
44
45     .form-header {
46         background: #rgb(255, 98, 14);
47         padding: 1.5rem 2rem;
48         text-align: center;
49     }
50
51     .form-title {
52         color: #rgb(255, 255, 255);
53         font-size: 1.9rem;
54         font-weight: 600;
55         margin: 0;
56     }
57
58     .form-body {
59         padding: 2rem;
60         display: flex;
61         flex-direction: column;
62     }
```

```

63     gap: 1.4rem;
64 }
65
66 .form-content {
67     display: flex;
68     flex-direction: column;
69     gap: 1.4rem;
70 }
71
72 .form-group label {
73     font-weight: 500;
74     color: #333;
75     margin-bottom: 0.5rem;
76     display: block;
77 }
78
79 .form-group input {
80     width: 90%;
81     padding: 0.75rem 1rem;
82     border: 1px solid #ddd;
83     border-radius: 0.6rem;
84     background: #fafafa;
85     transition: all 0.25s ease;
86     font-size: 1rem;
87 }
88
89 .form-group select {
90     width: 97%;
91     padding: 0.75rem 1rem;
92     border: 1px solid #ddd;
93     border-radius: 0.6rem;
94     background: #fafafa;
95     transition: all 0.25s ease;
96     font-size: 1rem;
97 }
98
99 .form-group input:hover,
100 .form-group select:hover {
101     background: #fff;
102     border-color: #ffcc80;
103     transform: scale(1.02);
104     box-shadow: 0 0 5px rgba(255, 112, 67, 0.2);
105 }
106
107 .form-group input:focus,
108 .form-group select:focus {
109     border-color: #ffa726;
110     background-color: #ffdddf;
111     outline: none;
112     box-shadow: 0 0 6px rgba(255, 143, 0, 0.25);
113     transform: scale(1.03);
114 }
115
116 .error-text {
117     color: #e53935;
118     font-size: 0.85rem;
119     margin-top: 0.3rem;
120 }
121
122 .alert-success {

```

```

123     background: #f1f8e9;
124     border: 1px solid #aed581;
125     padding: 0.8rem 1rem;
126     border-radius: 0.5rem;
127     color: #558b2f;
128     font-weight: 500;
129     text-align: center;
130     margin-bottom: 1rem;
131 }
132
133 .btn-wrapper {
134     text-align: center;
135 }
136
137 .btn-submit {
138     background: linear-gradient(to right, rgb(255, 154, 30), #f4511e);
139     color: #fff;
140     padding: 0.9rem 2.2rem;
141     border: none;
142     border-radius: 0.6rem;
143     font-weight: 600;
144     font-size: 1rem;
145     cursor: pointer;
146     transition: all 0.3s ease;
147     box-shadow: 0 5px 15px rgba(245, 81, 30, 0.3);
148 }
149
150 .btn-submit:hover {
151     background: linear-gradient(to right, #f57c00, rgb(226, 86, 43));
152     transform: translateY(-2px);
153     box-shadow: 0 8px 25px rgba(245, 81, 30, 0.4);
154 }
155
156 @media (max-width: 600px) {
157     .form-box {
158         padding: 0;
159     }
160
161     .form-header {
162         padding: 1rem;
163     }
164
165     .form-title {
166         font-size: 1.6rem;
167     }
168
169     .form-body {
170         padding: 1.5rem;
171     }
172
173     .btn-submit {
174         width: 100%;
175     }
176 }
177 </style>
178 <br><br><br><br><br>
179 <div class="form-container">
180     <div class="form-box">
181         <div class="form-header">
182             <h2 class="form-title">Form Input Penjualan</h2>

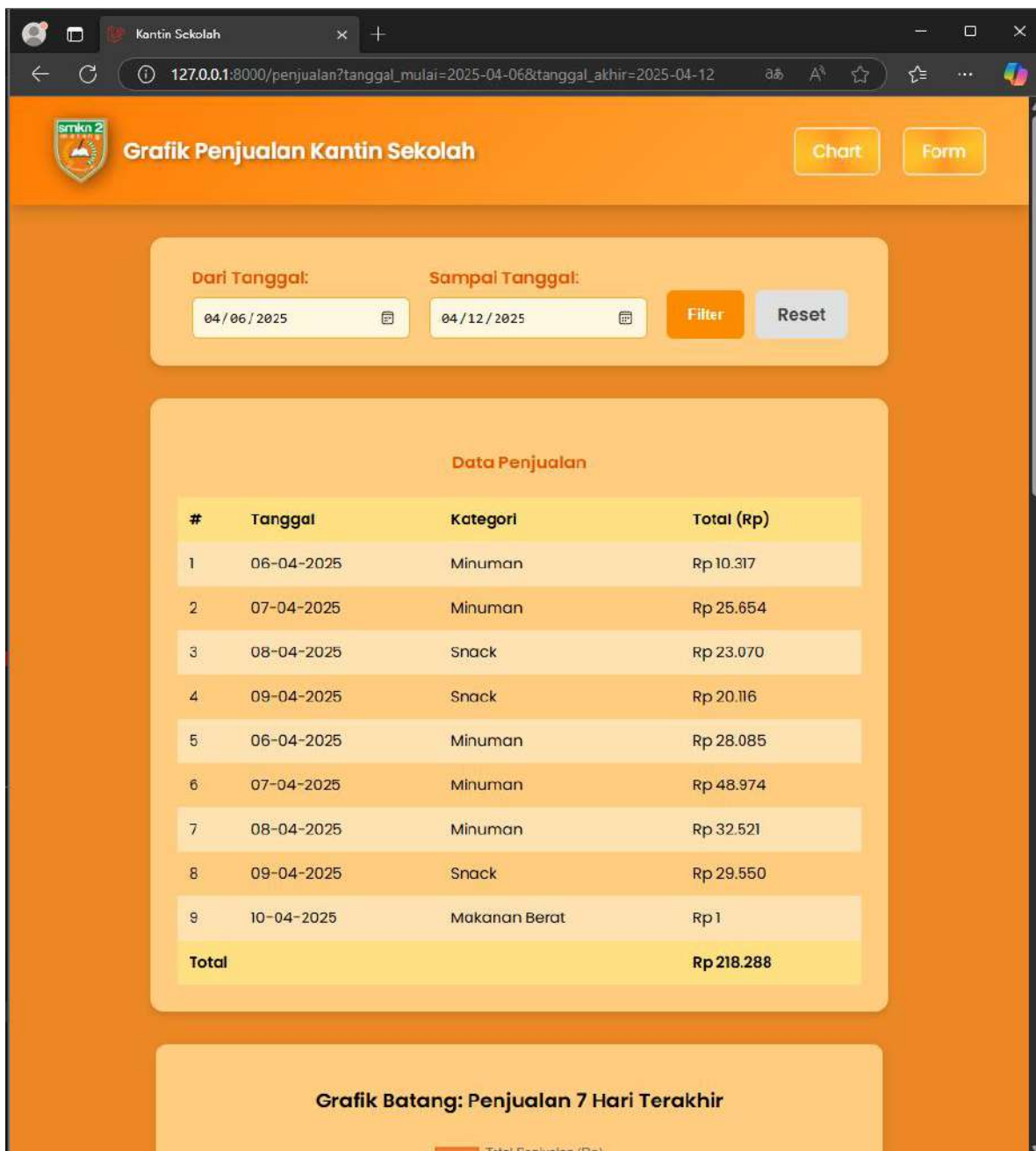
```

```

183 </div>
184
185 <div class="form-body">
186     @if(session(key: 'success'))
187     <div class="alert-success">{{ session(key: 'success') }}</div>
188     @endif
189
190     <form action="{{ route(name: 'penjualan.store') }}" method="POST" class="form-content">
191         @csrf
192
193         <div class="form-group">
194             <label for="tanggal">Tanggal</label>
195             <input type="date" name="tanggal" id="tanggal" value="{{ old(key: 'tanggal') }}" required>
196             @error('tanggal')
197             <small class="error-text">{{ $message }}</small>
198             @enderror
199         </div>
200
201         <div class="form-group">
202             <label for="total_penjualan">Total Penjualan</label>
203             <input type="number" name="total_penjualan" id="total_penjualan"
204             placeholder="harga total dalam 1 hari" value="{{ old(key: 'total_penjualan') }}" required>
205             @error('total_penjualan')
206             <small class="error-text">{{ $message }}</small>
207             @enderror
208         </div>
209
210         <div class="form-group">
211             <label for="kategori">Kategori</label>
212             <select name="kategori" id="kategori" required>
213                 <option value="">-- Pilih Kategori --</option>
214                 <option value="Makanan Berat" {{ old(key: 'kategori') == 'Makanan Berat' ? 'selected' : '' }}>Makanan Berat</option>
215                 <option value="Minuman" {{ old(key: 'kategori') == 'Minuman' ? 'selected' : '' }}>Minuman</option>
216                 <option value="Snack" {{ old(key: 'kategori') == 'Snack' ? 'selected' : '' }}>Snack</option>
217             </select>
218             @error('kategori')
219             <small class="error-text">{{ $message }}</small>
220             @enderror
221         </div>
222
223         <div class="btn-wrapper">
224             <button type="submit" class="btn-submit">Simpan</button>
225         </div>
226     </form>
227 </div>
228 </div>
229 <br>
230 <br>
231 <br>
232 @endsection

```

2. Output Program:





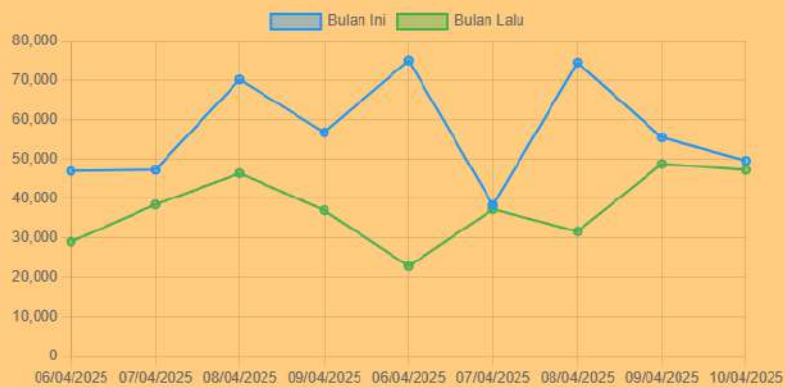
Grafik Penjualan Kantin Sekolah

[Chart](#)[Form](#)

Grafik Batang: Penjualan 7 Hari Terakhir



Grafik Garis: Perbandingan Penjualan Bulanan





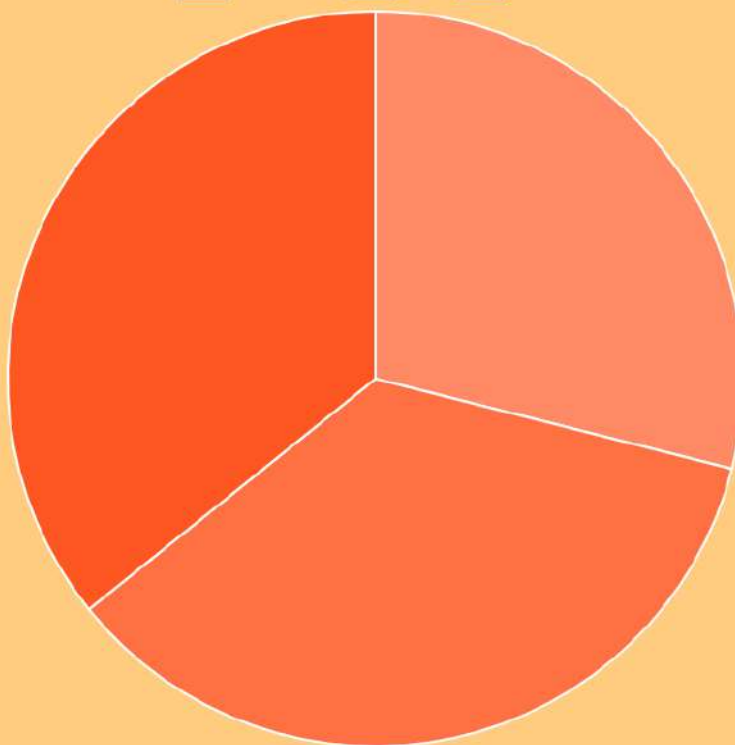
Grafik Penjualan Kantin Sekolah

Chart

Form


Grafik Pie: Persentase Penjualan per Kategori

Makanan Berat Snack Minuman



Kantin Sekolah

127.0.0.1:8000/penjualan/form



Grafik Penjualan Kantin Sekolah

ChartForm

Form Input Penjualan

Tanggal

mm/dd/yyyy

Total Penjualan

harga total dalam 1 hari

Kategori


-- Pilih Kategori --

Simpan

© 2025 Kantin Sekolah, SMK Negeri 2 Malang

Kantin Sekolah


127.0.0.1:8000/penjualan/form

 **Grafik Penjualan Kantin Sekolah** [Chart](#) [Form](#)

Form Input Penjualan

Data penjualan berhasil ditambahkan!

Tanggal

mm/dd/yyyy 

Total Penjualan

harga total dalam 1 hari

Kategori

-- Pilih Kategori --

Simpan

3. Keterangan:

Struktur Kode

- **Controller PenjualanController:**
 - Metode `index` menangani logika untuk menampilkan data berdasarkan filter tanggal yang diminta dan menyiapkan data untuk grafik (Batang, Pie, dan Garis).
 - Di dalam `index`, kita melakukan beberapa query database untuk mendapatkan data yang diperlukan, seperti penjualan berdasarkan tanggal, total per kategori, dan perbandingan penjualan antar bulan.
- **Penyusunan Data untuk Grafik:**
 - **Grafik Batang:** Mengambil data penjualan untuk 7 hari terakhir dan menyiapkan label serta data yang akan ditampilkan di grafik.
 - **Grafik Pie:** Menampilkan total penjualan berdasarkan kategori dengan menggunakan `groupBy()` untuk mengelompokkan data penjualan berdasarkan kategori dan menghitung totalnya.
 - **Grafik Garis:** Menyiapkan data per hari untuk bulan ini dan bulan sebelumnya, dan menggabungkan data untuk menghasilkan perbandingan yang jelas.
- **Form Input dan Penyimpanan Data:**
 - Metode `form` menampilkan form untuk memasukkan data penjualan baru.
 - Metode `store` menangani penyimpanan data penjualan baru setelah diverifikasi.
- **Routing:**
 - Pastikan bahwa setiap proses seperti menampilkan grafik, form input, dan penyimpanan data dapat diakses melalui routing yang telah ditentukan dalam `web.php`.

Kesimpulan

Dalam studi kasus ini, saya berhasil menambahkan fitur-fitur baru yang memberikan fleksibilitas bagi pengguna untuk memfilter data berdasarkan tanggal dan menampilkan analisis penjualan menggunakan grafik. Struktur kode yang dibangun terorganisir dengan baik dan menggunakan query database yang efisien untuk memproses data sesuai kebutuhan tampilan yang diinginkan. Sistem ini memberikan pengalaman pengguna yang lebih baik dalam menganalisis data penjualan..

Setelah menambahkan fitur filter tanggal, grafik pie, dan grafik garis, saya juga menambahkan form input untuk memasukkan data penjualan baru ke dalam sistem. Form input ini memungkinkan pengguna untuk memasukkan tanggal, total penjualan, dan kategori makanan. Ketika form disubmit, data akan divalidasi terlebih dahulu untuk memastikan bahwa tanggal yang dimasukkan benar, total penjualannya berupa angka, dan kategori sesuai dengan list.

Jika semua data valid, saya menggunakan metode `create()` untuk menyimpan data tersebut ke dalam database. Setelah data berhasil disimpan, pengguna akan mendapatkan notifikasi sukses dan diarahkan kembali ke form input. Dengan cara ini, pengguna dapat terus menambah data penjualan dengan mudah.