**Laravel DataTables dengan Filter Pencarian yang Berelasi**

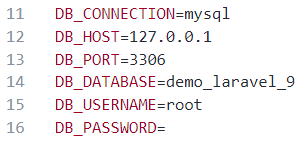
**Membuat Project Laravel**

Gunakan perintah berikut pada CMD / Terminal:

$ **composer create-project --prefer-dist laravel/laravel dt-col-rel-search**

**Konfigurasi Database**

Buka konfigurasi database pada file **.env** sebagai berikut:



**Mengkonfigurasi DataTables**

Jalankan perintah dibawah pada CMD / Terminal untuk mendapatkan DataTables:

$ **composer require yajra/laravel-datatables-oracle**

Kemudian daftarkan DataTables tersebut kedalam **Providers**, buka konfigurasinya pada **config/app.php** dan tuliskan konfigurasinya sebagai berikut:

'providers' => [

..........

**Yajra\DataTables\DataTablesServiceProvider::class,**

]

..........

    'aliases' => Facade::defaultAliases()->merge([

**'DataTables' => Yajra\DataTables\Facades\DataTables::class,**

    ])->toArray(),

**Mengkonfigurasi Model dan Migration**

Membuat sebuah model dan migration yang digunakan untuk mengelola sebuah tabel articles dengan perintah dibawah ini:

$ **php artisan make:model Article –m**

Buka file migrationnya pada **database/migrations/create\_articles\_table.php** dan tuliskan kode didalamnya:

    public function up()

    {

        Schema::create('articles', function (Blueprint $table) {

**$table->id();**

**$table->string('title', 191);**

**$table->text('detail')->nullable();**

**// foreign key terhubung ke tabel users**

**$table->integer('user\_id');**

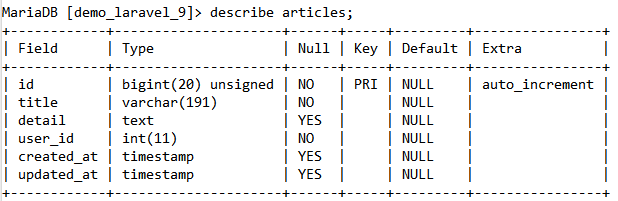
**$table->timestamps();**

        });

    }

Jalankan perintah migrate berikut untuk menciptakan tabel **articles**:

$ **php artisan migrate**



Kemudian buka file modelnya pada **app/Models/Article.php** dan tambahkan kode berikut:

class Article extends Model {

    use HasFactory;

**protected $fillable = ["title", "detail"];**

**public function users() {**

**return $this->belongsTo(User::class, "user\_id", "id");**

**}**

}

**Membuat Controller untuk Article**

Buka terminal dan tuliskan perintah berikut:

$ **php artisan make:controller ArticleController**

Buka file **ArticleController.php** tersebut dan tambahkan kode sebagai berikut:

    public function index(Request $request)

    {

        if ($request->ajax()) {

            $model = Article::with('users');

            return DataTables::eloquent($model)

->addColumn('users', function (Article $article) {

                return $article->users->name;

            })->toJson();

        }

        return view('users');

    }

**Mengkonfigurasi Perutean**

Buka file **routes/web.php** dan tambahkan kode berikut:

Route::get('articles', [ArticleController::class, 'index'])

->name('articles.index');

**Membuat Tampilan Users**

Buat sebuah file blade baru pada **views/users.blade.php** dan tuliskan kode didalamnya:

<!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>DataTables with Relationship Search</title>

    {{-- import --}}

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.1.3/css/bootstrap.min.css" />

    <link href="https://cdn.datatables.net/1.10.16/css/jquery.dataTables.min.css" rel="stylesheet">

    <link href="https://cdn.datatables.net/1.10.19/css/dataTables.bootstrap4.min.css" rel="stylesheet">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.js"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-validate/1.19.0/jquery.validate.js"></script>

    <script src="https://cdn.datatables.net/1.10.16/js/jquery.dataTables.min.js"></script>

    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script>

    <script src="https://cdn.datatables.net/1.10.19/js/dataTables.bootstrap4.min.js"></script>

    <style type="text/css">

        .paginate\_button {

            padding: 0px !important;

        }

    </style>

</head>

    <script src="https://cdn.datatables.net/1.10.16/js/jquery.dataTables.min.js"></script>

    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script>

    <script src="https://cdn.datatables.net/1.10.19/js/dataTables.bootstrap4.min.js"></script>

    <style type="text/css">

        .paginate\_button {

            padding: 0px !important;

        }

    </style>

</head>

<body>

    <div class="container" style="margin-top: 100px; margin-bottom: 100px">

        <div class="row justify-content-center">

            <div class="col-md-12">

                <div class="card">

                    <div class="card-header bg-info text-white">

                        Laravel DataTables with Column Relationship Search

                    </div>

                    <div class="card-body">

                        @if (session('status'))

                            <div class="alert alert-success" role="alert">

                                {{ sssion('status') }}

                            </div>

                        @endif

                        <table class="table table-bordered table-hover data-table">

                            <thead>

                                <tr>

                                    <th>NO</th>

                                    <th>TITLE</th>

                                    <th>DETAIL</th>

                                    <th>AUTHOR</th>

                                </tr>

                            </thead>

                            <tbody></tbody>

                        </table>

                    </div>

                </div>

            </div>

        </div>

    </div>

    <script type="text/javascript">

        $(document).ready(function() {

            $('.data-table').DataTable({

                processing: true,

                serverSide: true,

                ajax: "{{ route('articles.index') }}",

                columns: [{

                        data: 'id',

                        name: 'id'

                    },

                    {

                        data: 'title',

                        name: 'title'

                    },

                    {

                        data: 'detail',

                        name: 'detail'

                    },

                    {

                        data: 'users',

                        name: 'users.name'

                    },

                ]

            });

        });

    </script>

</body>

</html>

    <script type="text/javascript">

        $(document).ready(function() {

            $('.data-table').DataTable({

                processing: true,

                serverSide: true,

                ajax: "{{ route('articles.index') }}",

                columns: [{

                        data: 'id',

                        name: 'id'

                    },

                    {

                        data: 'title',

                        name: 'title'

                    },

                    {

                        data: 'detail',

                        name: 'detail'

                    },

                    {

                        data: 'users',

                        name: 'users.name'

                    },

                ]

            });

        });

    </script>

</body>

</html>

**Membuat Dummy Data**

Buat sebuah factory untuk membuat dummy data pada tabel **users** dan tabel **articles** dengan perintah dibawah ini:

$ **php artisan make:factory ArticleFactory**

    public function definition() {

**return [**

**'title' => $this->faker->sentence(3),**

**'detail' => $this->faker->sentence(5),**

**'user\_id' => $this->rand(1, 5),**

**];**

    }

$ **php artisan make:factory UserFactory**

    public function definition() {

**return [**

**'name' => fake()->name(),**

**'email' => fake()->unique()->safeEmail(),**

**'email\_verified\_at' => now(),**

**'password' => '$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', // password**

**'remember\_token' => Str::random(10),**

**];**

    }

Kemudian buka file **database/seeders/DatabaseSeeder.php** dan tambahkan kode berikut:

..........

**use App\Models\Article;**

**use App\Models\User;**

class DatabaseSeeder extends Seeder {

    public function run() {

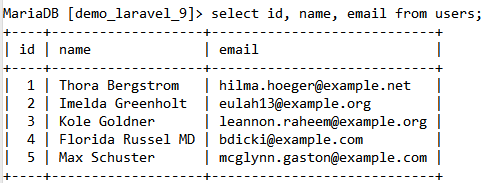
**User::factory(5)->create();**

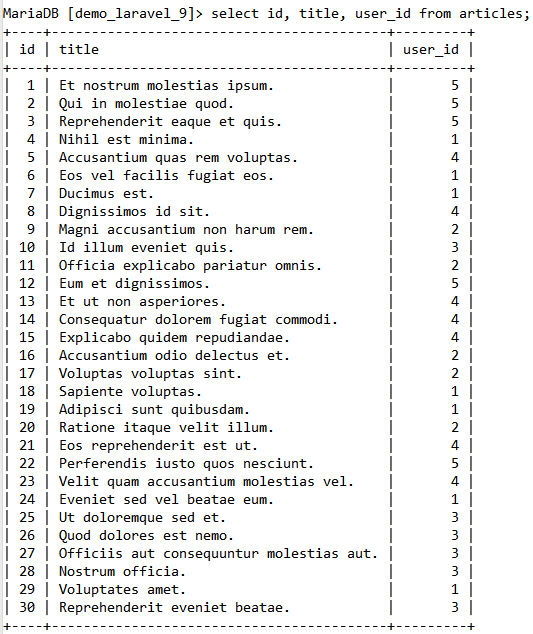
**Article::factory(30)->create();**

    }

}

Lihat pada tabel **users** dan tabel **articles**, sudah terisi dengan dummy data:





**Hasil di Browser**

