<iDev />

https://wshlink.idevsemarang.my.id/

Werdani Sulistya Hadi
Developer & Tutor

Building your first BOM for general use

2017 **Kotlin Trainer**
UNIKA Soegijopranoto

2018 **Code Camp for Kids**
Kreazona Semarang

2019 **Developer Director**
Rectmedia

2019 **Founder & CEO**
iDev Semarang

2022 **Outsource Director**
Multi Reva Parama

2023 **Developer Trainer**
MMT Surabaya

2024 **CTO**
AWM Indonesia

Perkenalan

iDev Client & Experience

ERP

↓

????

↓

Bill Of Material

Ilustrasi & Lingkup ERP

Mie Lorso

BOM dalam kehidupan sehari-hari

Mie Lorso

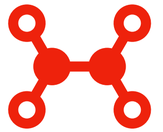
Mie (1 bks)	1 x 10.000 = 10.000
Sosis (6 Buah)	6 x 500 = 3.000
Telor (2 Buah)	2 x 3.000 = 6.000

Minimum Price

19.000

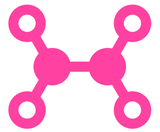
BOM dalam kehidupan sehari-hari

T-imuno

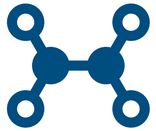


Vitamin E (30mg)

Folic Acid (400mg)



Zinc Picolinat (22mg)



BOM dalam Lingkup Industri Farmasi

Berikut ini merupakan langkah yang baik dalam rangkaian proses development sistem, yakni….

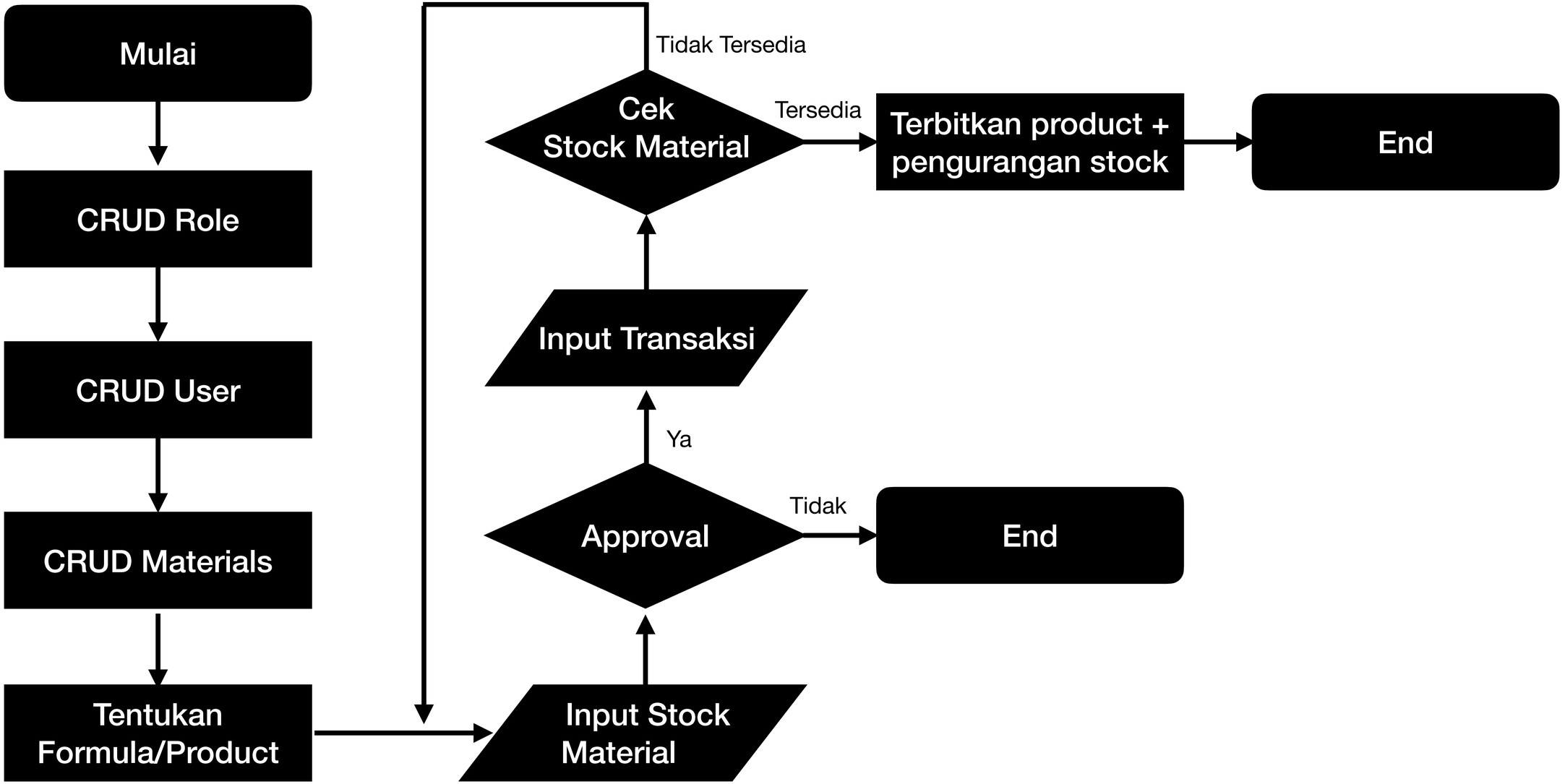
a. menyusun function -> design database -> pemilihan teknologi -> menyusun flowchart

b. design database -> pemilihan teknologi -> menyusun flowchart -> menyusun function

c. pemilihan teknologi -> menyusun flowchart -> menyusun function -> design database

d. menyusun flowchart -> design database -> pemilihan teknologi -> menyusun function

Mini Quiz



Basic Flow

Berikut ini merupakan flow dasar yang bisa dikembangkan sesuai dengan kebutuhan industri.

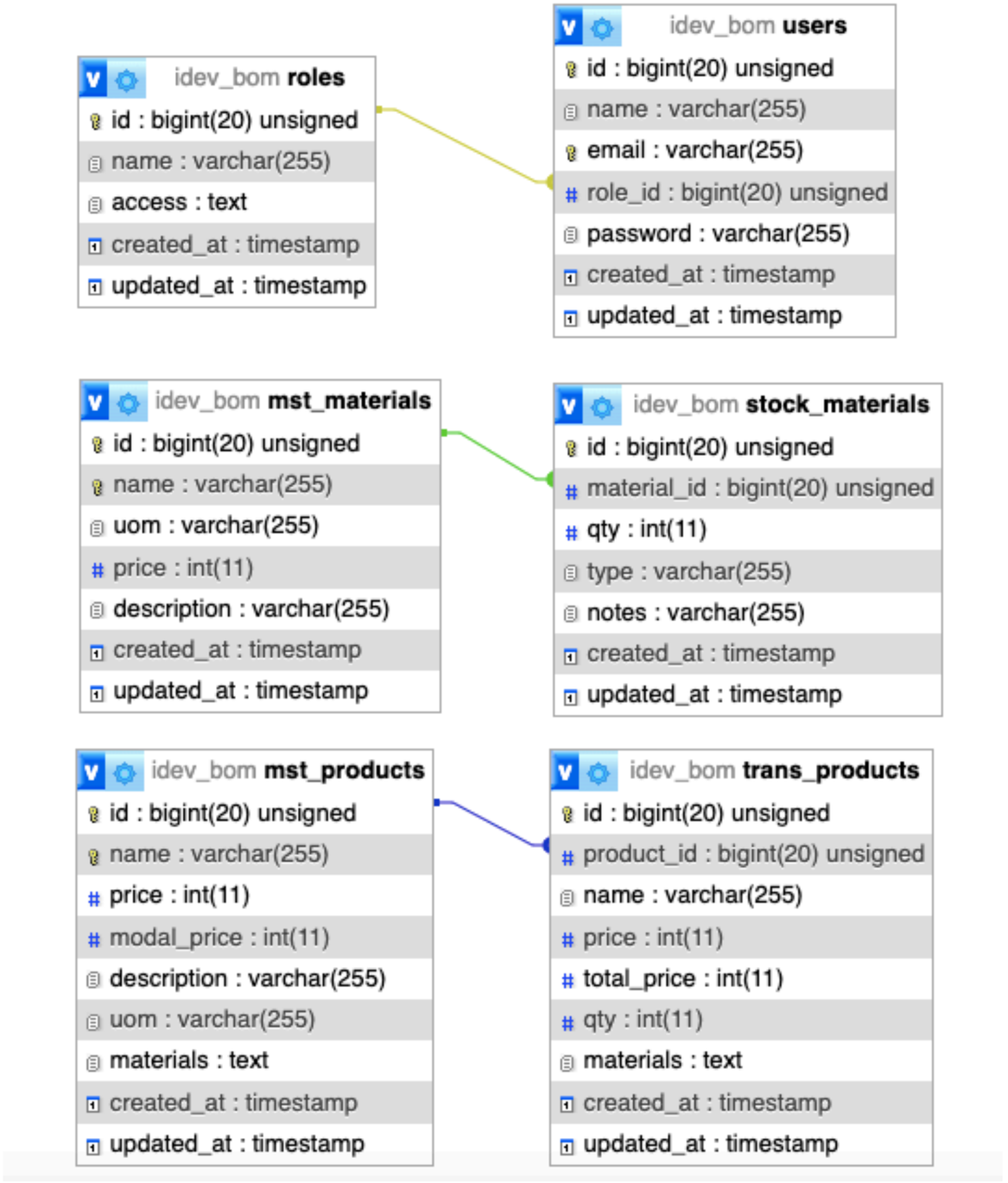
Notes

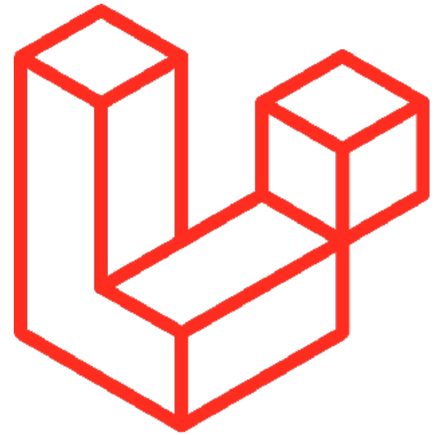
Pastikan Anda menggunakan id, maupun uuid sebagai identitas dan primary key dari tiap data.

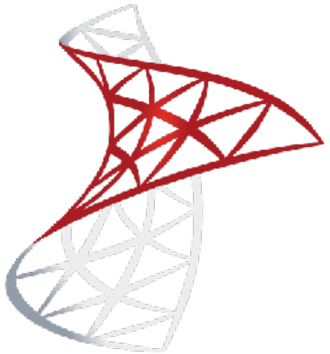
Selain itu, Perlu adanya created_at, updated_at sebagai pecatat waktu create maupun update.

Pada tingkat lanjut dianjurkan untuk menggunakan tabel log/reivision.

Struktur Database



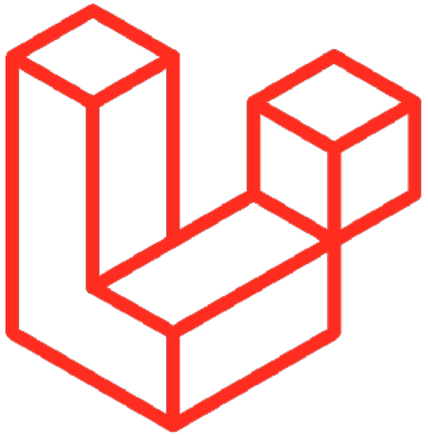


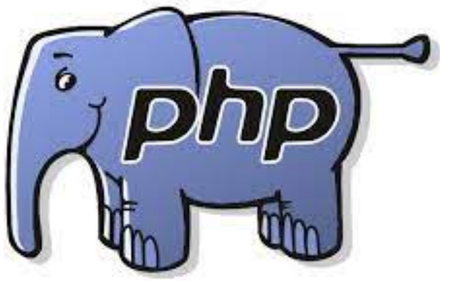


Teknologi & Framework

Notes

Pada pembelajaran kali ini kita akan menggunakan php dengan framework laravel serta database mysql. Untuk mempercepat proses development, maka kita akan menggunakan package ideveasyadmin yang dikembangkan Oleh Tim iDev Semarang.

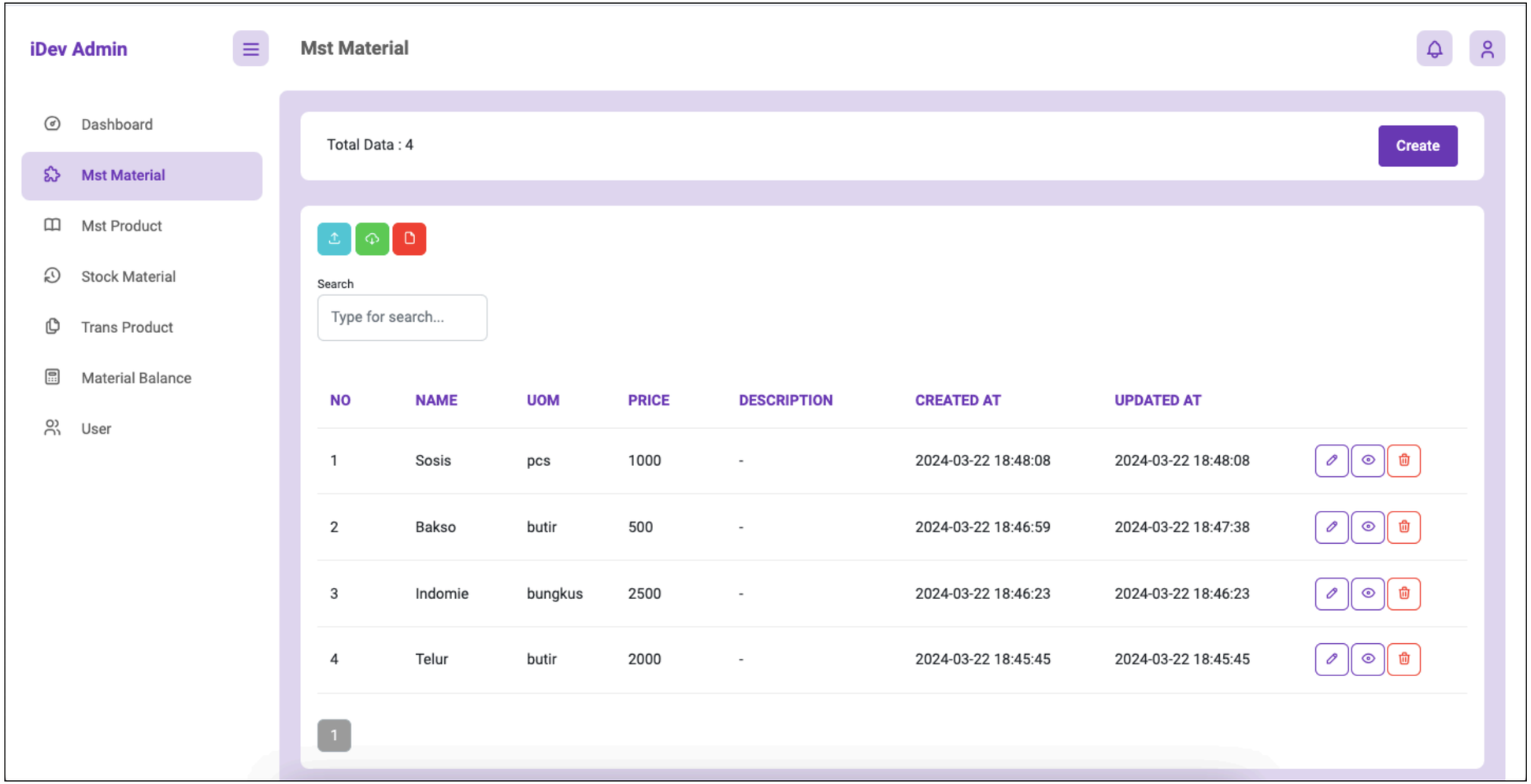




Tools yang kita digunakan

idev easyadmin memungkinkan developer dalam membuat halaman admin dengan mudah, yakni:

1. Menggunakan perintah artisan,
2. terdapat fitur authentikasi,
3. role management,
4. import-export data.
5. Crud
6. Dan masih banyak lagi



Gambaran idev easyadmin

Install Laravel (disarankan menggunakan laravel 9 atau 10)

composer create-project laravel/laravel:^10.0 bom-app

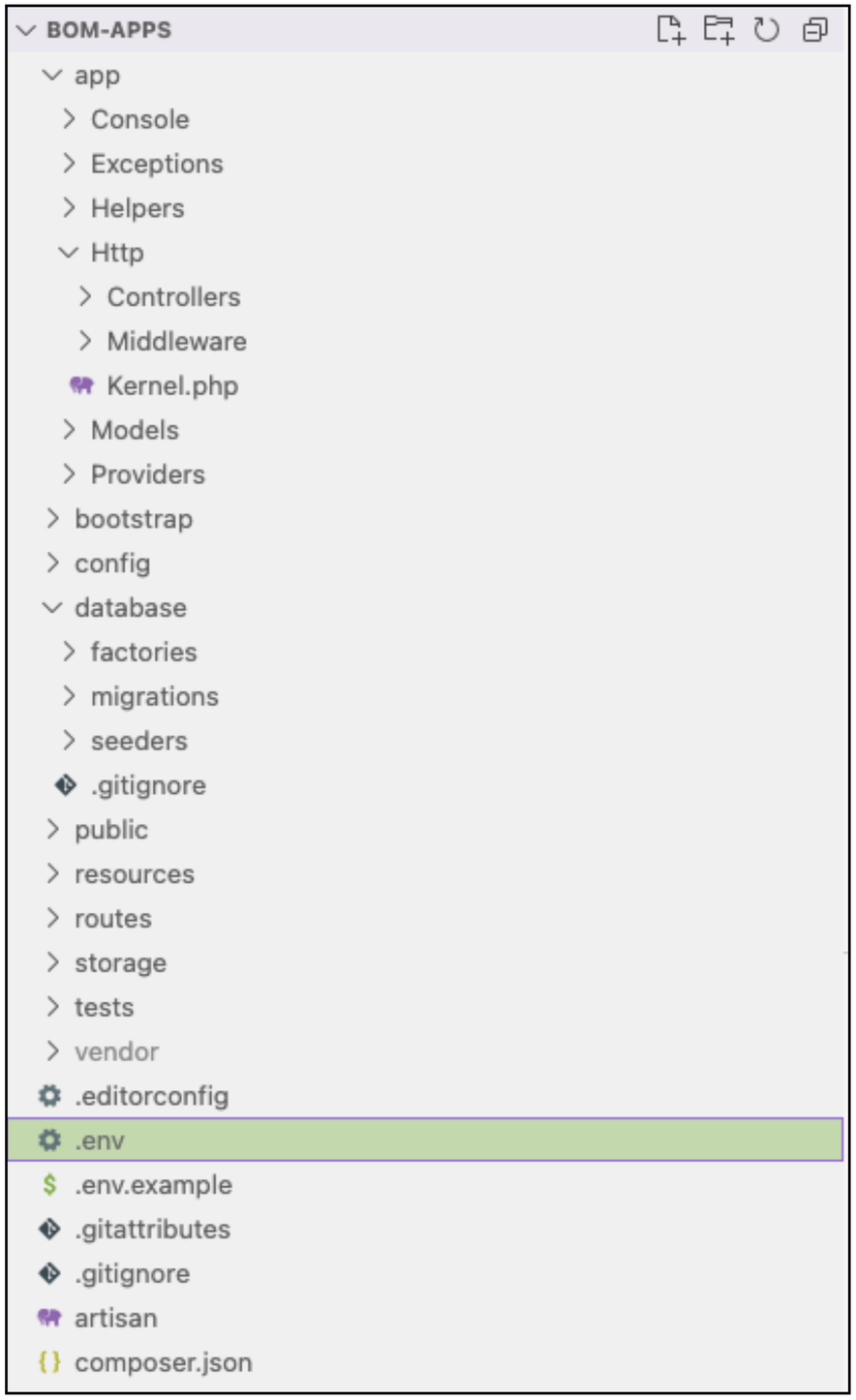
Install iDev Easyadmin

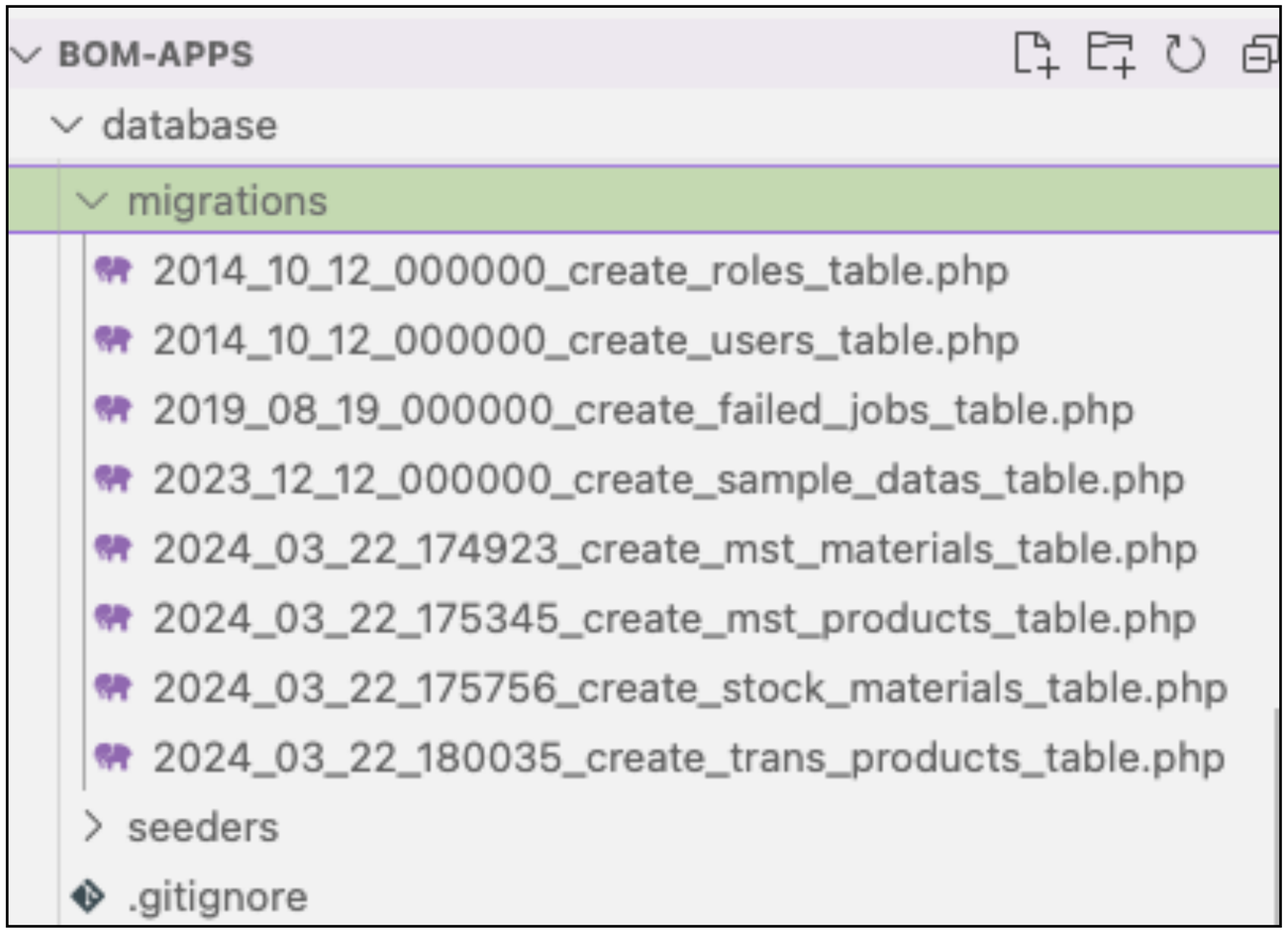
https://github.com/idevsemarang/easyadmin

.env

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=idev_bom
DB_USERNAME=root
DB_PASSWORD=
```

Mulai Project





Persiapkan File Migrasi Master Material

php artisan make:migration create_mst_materials_table

Persiapkan File Migrasi Master Product

php artisan make:migration create_mst_products_table

Persiapkan File Migrasi Stock Material

php artisan make:migration create_stock_materials_table

Persiapkan File Migrasi Transaction Product

php artisan make:migration create_trans_products_table

Membuat Migration

```
        Schema::create('mst_materials', function (Blueprint $table) {
            $table->id();
            $table->string('name')->unique();
            $table->string('uom');
            $table->integer('price')->default(0);
            $table->string('description')->nullable();
            $table->timestamps();
        });
```

Pembuatan file migrasi bertujuan untuk menyiapkan konsep tabel yang akan dibuat. Berikut ini merupakan minimum columns yang harus ada pada tabel master material

Master Material

```php
Schema::create('mst_products', function (Blueprint $table) {
    $table->id();
    $table->string('name')->unique();
    $table->integer('price')->default(0);
    $table->integer('modal_price')->default(0);
    $table->string('description')->nullable();
    $table->string('uom');
    $table->text('materials')->nullable();
    $table->timestamps();
});
```

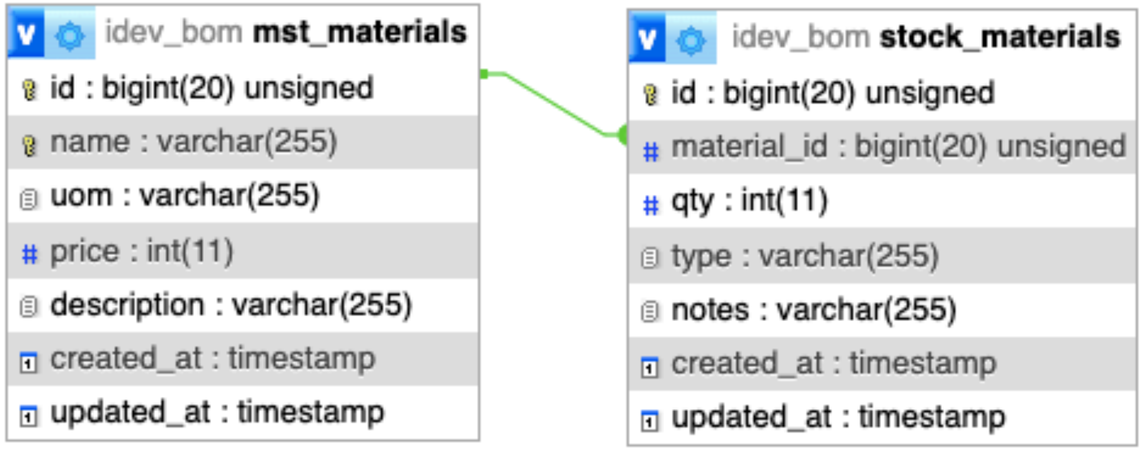
Berikut ini merupakan minimum columns yang harus ada pada tabel master product. Yang menarik di sini, penentuan material akan disimpan dalam kolom materials dengan memanfaatkan string json seperti demikian. Sehingga akan lebih menghemat space dan mempercepat performa.

[{“material_id":2,"name":"Indomie","price":2500,"qty":"1"},
{"material_id":3,"name":"Bakso","price":500,"qty":"4"}]

Master Product

```php
Schema::create('stock_materials', function (Blueprint $table) {
    $table->id();
    $table->unsignedBigInteger('material_id');
    $table->integer('qty')->default(0);
    $table->string('type')->nullable();
    $table->string('notes')->nullable();
    $table->timestamps();

    $table->foreign('material_id')->references('id')->on('mst_materials');
});
```

Berikut ini merupakan minimum columns yang harus ada pada tabel stock materials. Pada bagian ini terlihat adana relasi tabel sesuai konsep yang kita bahas pada halaman sebelumnya.



Stock Material

```php
Schema::create('trans_products', function (Blueprint $table) {
    $table->id();
    $table->unsignedBigInteger('product_id');
    $table->string('name');
    $table->integer('price')->default(0);
    $table->integer('total_price')->default(0);
    $table->integer('qty')->default(0);
    $table->text('materials')->nullable();
    $table->timestamps();

    $table->foreign('product_id')->references('id')->on('mst_products');
});
```

Ini merupakan tabel transaksi yang berperan penting dalam proses seperti penjualan, pengeluaran stock, pembuatan MO, dan lain sebagainya. Meski tabel ini berelasi dengan master product, tetapi kita tetap perlu menambahkan informasi product karena harus bersifat historical.

Trans Products

```
php artisan migrate

php artisan db:seed
```

Notes

Dengan menggunakan migrate dan seeder bersama-sama, Anda dapat mengontrol struktur dan isi basis data dengan baik, serta memastikan bahwa aplikasi Laravel Anda memiliki lingkungan pengembangan yang konsisten dan dapat diandalkan.

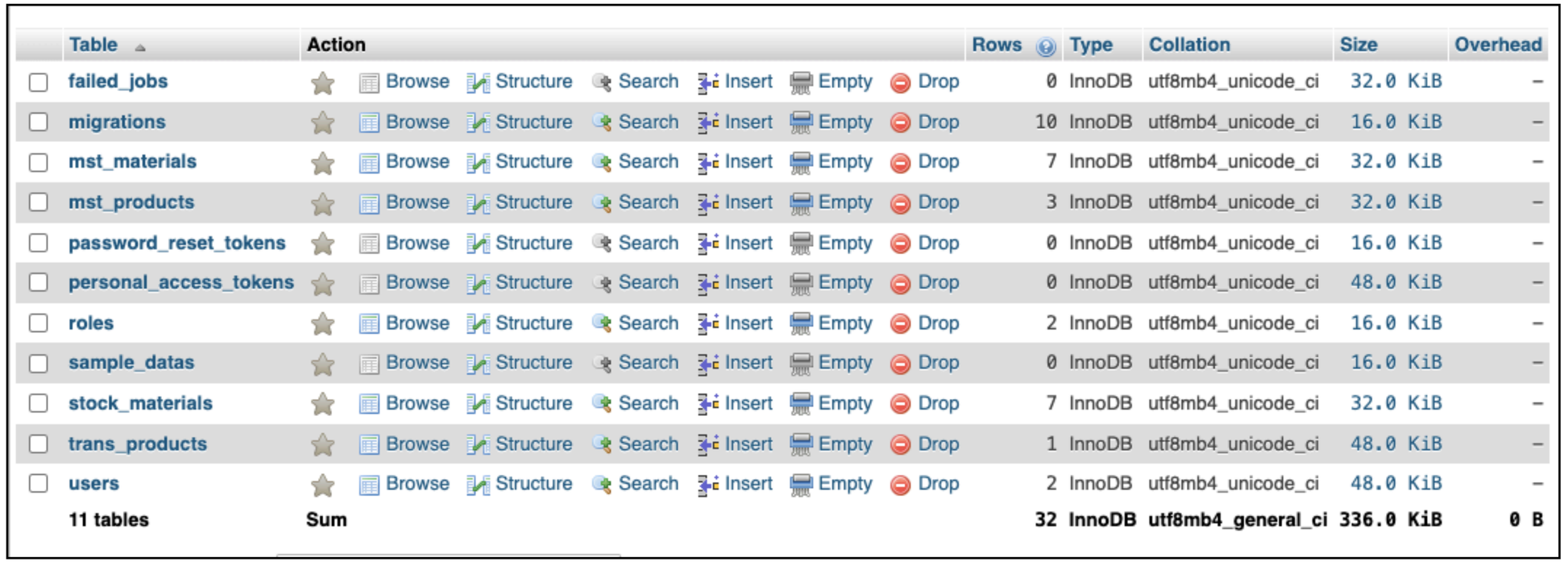


	Table	Action	Rows	Type	Collation	Size	Overhead
☐	failed_jobs	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	–
☐	migrations	Browse Structure Search Insert Empty Drop	10	InnoDB	utf8mb4_unicode_ci	16.0 KiB	–
☐	mst_materials	Browse Structure Search Insert Empty Drop	7	InnoDB	utf8mb4_unicode_ci	32.0 KiB	–
☐	mst_products	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_unicode_ci	32.0 KiB	–
☐	password_reset_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	–
☐	personal_access_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 KiB	–
☐	roles	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	16.0 KiB	–
☐	sample_datas	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	–
☐	stock_materials	Browse Structure Search Insert Empty Drop	7	InnoDB	utf8mb4_unicode_ci	32.0 KiB	–
☐	trans_products	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	48.0 KiB	–
☐	users	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	48.0 KiB	–
	11 tables	**Sum**	**32**	**InnoDB**	**utf8mb4_general_ci**	**336.0 KiB**	**0 B**

Migrate & Seeder

Perbedaan Antara string() dan text() dalam pembuatan migration adalah…

a. string() Berfungsi untuk menampung numerik, dan text() Berfungsi untuk menampung kata-kata

b. string() hanya memiliki maksimal 11 karakter, dan text() bisa menampung hingga 256 karakter

c. string() hanya bisa menampung maksimal 256 karakter, dan text() bisa menampung lebih dari 256 karakter

d. string() bisa menampung lebih dari 256 karakter, dan text() Hanya bisa menampung maksimal 256 karakter

Mini Quiz

MVC Mst Material

php artisan idev:controller-maker --slug=mst-material --table=mst_materials

MVC Mst Product

php artisan idev:controller-maker --slug=mst-product --table=mst_products

MVC Stock Material

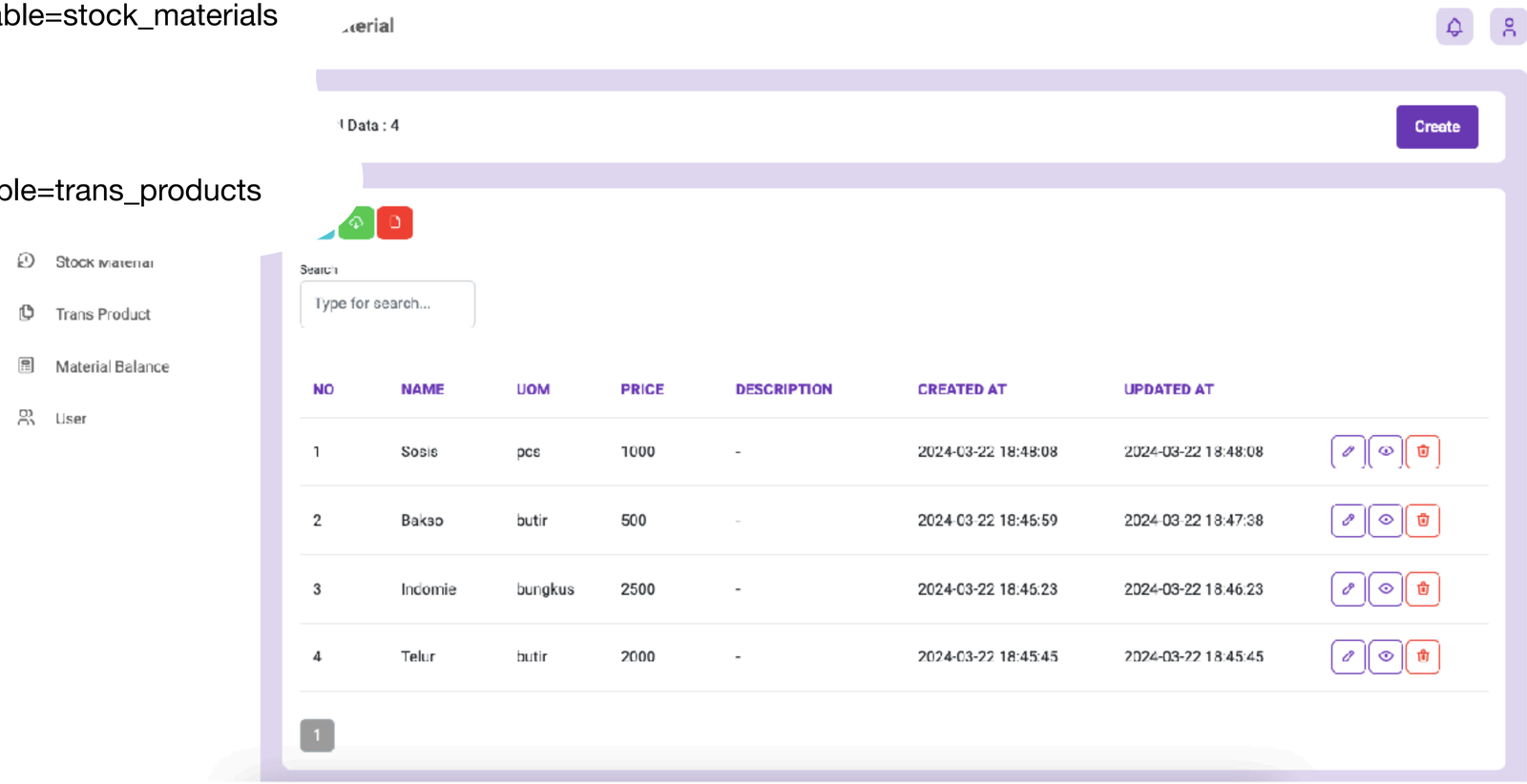
php artisan idev:controller-maker --slug=stock-material --table=stock_materials

MVC Trans Product

php artisan idev:controller-maker --slug=trans-product --table=trans_products

Notes

Command di samping akan otomatis menggenerate model, view, controller.
Kita perlu menambahkan ke dałam routes dań Sidebar Helper.
Controller hasil generate berada di folder App\Http\Controllers



Generate MVC

```php
Route::group(['middleware' => ['web', 'auth']], function () {
    Route::resource('mst-material', MstMaterialController::class);
    Route::get('mst-material-api', [MstMaterialController::class, 'indexApi'])
     ->name('mst-material.listapi');
    Route::get('mst-material-export-pdf-default', [MstMaterialController::class, 'exportPdf'])
     ->name('mst-material.export-pdf-default');
    Route::get('mst-material-export-excel-default', [MstMaterialController::class, 'exportExcel'])
     ->name('mst-material.export-excel-default');
    Route::post('mst-material-import-excel-default', [MstMaterialController::class, 'importExcel'])
     ->name('mst-material.import-excel-default');
});
```

#1

Contoh hasil generator yang diterapkan pada routes/web.php

#2

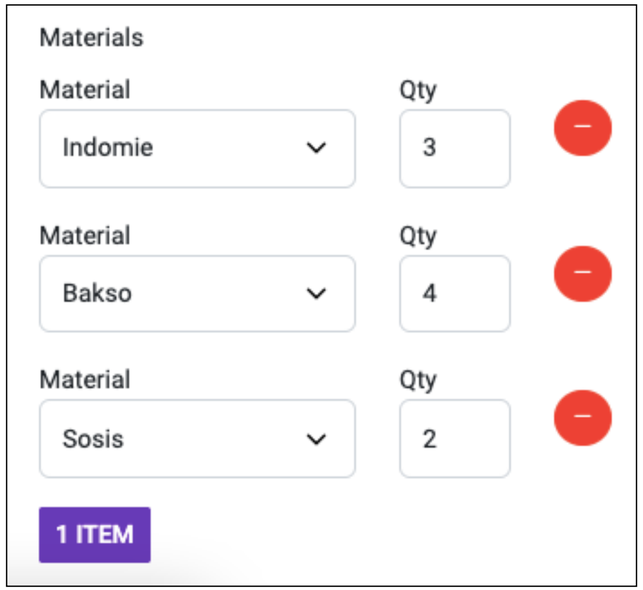
Contoh hasil generator yang diterapkan pada app/Helpers/Sidebar.php

```php
[
  'name' => 'Mst Material',
  'icon' => 'ti ti-puzzle',
  'key' => 'mst-material',
  'base_key' => 'mst-material',
  'visibility' => true,
  'ajax_load' => false,
  'childrens' => []
],
```

Generate MVC

#1

Pada MstProductController terdapat function fields() dan lakukan kustomisasi pada materials, dengan mengganti tipe field menjadi model repeatable.




```php
[
    'type' => 'repeatable',
    'label' => 'Materials',
    'name' =>  'materials',
    'class' => 'col-md-12 my-2',
    'required' => $this->flagRules('materials', $id),
    'value' => (isset($edit)) ? $edit->materials : '',
    'enable_action' => true,
    'html_fields' => [
        [
            'type' => 'select',
            'label' => 'Material',
            'name' =>  'material_id',
            'class' => 'col-md-7 my-2',
            'value' => '',
            'options' => MstMaterial::get(['id', 'name'])
            ->map(function ($item) {
                return [
                    'value' => $item->id,
                    'text' => $item->name
                ];
            })
            ->toArray()
        ],
        [
            'type' => 'number',
            'label' => 'Qty',
            'name' =>  'qty',
            'class' => 'col-md-3 my-2',
        ],
    ]
],
```

Product Customization

Pada proses input multiple data, bisa menggunakan beberapa metode ini, kecuali….

a. Menggunakan DB transaction

b. Menggunakan chunk

c. Menggunakan queue, jobs

d. Menggunakan table view

Mini Quiz

#2

Apabila anda menggunakan model repeatable, pastikan juga untuk melakukan adaptasi dengan mengubah array json repeatable menjadi string, sehingga bisa disimpan ke dalam database.

```php
$materialIds = $request->material_id;
$qtys = $request->qty;

$arrQty = [];
foreach ($materialIds as $key => $mi) {
    $arrQty[$mi] = $qtys[$key];
}

$mstMaterials = MstMaterial::whereIn('id', $materialIds)->get();
$arrMaterials = [];
$modalPrice = 0;
foreach ($mstMaterials as $key => $mm) {
    $arrMaterials[] = [
        'material_id' => $mm->id,
        'name' => $mm->name,
        'price' => $mm->price,
        'qty' => $arrQty[$mm->id]
    ];

    $modalPrice += $mm->price*$arrQty[$mm->id];
}

$materials = json_encode($arrMaterials);
```

Product Customization

#3

Json string yang tersimpan dalam database bisa kita tampilkan dengan bantuan append pada model MstProduct dan sesuaikan dengan kebutuhan Anda. Serta tambahkan juga hasil append tersebut ke dalam $this->tableHeaders

```php
protected $appends = ['text_materials'];

public function getTextMaterialsAttribute()
{
    $arrMaterials = json_decode($this->materials, true);
    $str = "";
    foreach ($arrMaterials as $key => $am) {
        $str .= $am['name'].", ";
    }

    return $str;
}
```

```php
$this->tableHeaders = [
    ['name' => 'No', 'column' => '#', 'order' => true],
    ['name' => 'Name', 'column' => 'name', 'order' => true],
    ['name' => 'Price', 'column' => 'price', 'order' => true],
    ['name' => 'Modal price', 'column' => 'modal_price', 'order' => true],
    ['name' => 'Description', 'column' => 'description', 'order' => true],
    ['name' => 'Uom', 'column' => 'uom', 'order' => true],
    ['name' => 'Materials', 'column' => 'text_materials',
     'order' => true, 'search' => false],
    ['name' => 'Created at', 'column' => 'created_at', 'order' => true],
    ['name' => 'Updated at', 'column' => 'updated_at', 'order' => true],
];
```

Product Customization

#1

Kita bisa merubah material id pada fungsi fields() di dalam kelas StockMaterialController dengan model select atau dropdown dengan cara di samping supaya user lebih mudah dalam melakukan pengambilan data, yakni dengan mengambil data dari tabel master material. Ubah juga tipe stock dengan in / out.

```
[
    'type' => 'select',
    'label' => 'Material id',
    'name' =>  'material_id',
    'class' => 'col-md-12 my-2',
    'required' => $this->flagRules('material_id', $id),
    'value' => (isset($edit)) ? $edit->material_id : '',
    'options' => MstMaterial::get(['id', 'name'])
        ->map(function ($item) {
            return [
                'value' => $item->id,
                'text' => $item->name
            ];
        })
        ->toArray()
],
```

```
[
    'type' => 'select',
    'label' => 'Type',
    'name' =>  'type',
    'class' => 'col-md-12 my-2',
    'required' => $this->flagRules('type', $id),
    'value' => (isset($edit)) ? $edit->type : '',
    'options' => [
        ['value' => 'in', 'text' => 'in'],
        ['value' => 'out', 'text' => 'out'],
    ]
],
```

Stock Material

#2

Di sini kita melakukan join antara table stock_materials dengan mst_materials sesuai dengan skema database yang telah kita definisikan sebelumnya. Kustomisasi ini ada pada function defaultDataQuery().

```php
protected function defaultDataQuery()
{
    $filters = [];
    $orThose = null;
    $orderBy = 'id';
    $orderState = 'DESC';
    if (request('search')) {
        $orThose = request('search');
    }
    if (request('order')) {
        $orderBy = request('order');
        $orderState = request('order_state');
    }

    $dataQueries = $this->modelClass::where($filters)
            ->join('mst_materials', 'mst_materials.id', 'stock_materials.material_id')
            ->select('stock_materials.*', 'mst_materials.name as material_name')
            ->where(function ($query) use ($orThose) {
                $query->where('mst_materials.name', 'LIKE', '%' . $orThose . '%');
                $query->orWhere('notes', 'LIKE', '%' . $orThose . '%');
            })
            ->orderBy($orderBy, $orderState);

    return $dataQueries;
}
```

Stock Material

#1

Pada transaksi product yang kita perlukan hanyalah pengambilan product_id serta quantity product yang akan dikeluarkan.
Dalam srtuasi ini, kita perlu menambahkan logic yang berfungsi untuk melakukan pengecekan stock material.

Trans Product

```php
$productId = $request->product_id;
$qtyProduct = $request->qty;

$product = MstProduct::where('id', $productId)->first();
$errorsOutOfStocks = [];
$stockOuts = [];

if ($product) {
    $arrMaterials = [];
    if ($product->materials) {
        $arrMaterials = json_decode($product->materials, true);
    }
    $materialIds = array_column($arrMaterials, 'material_id');

    $currentStocks = StockMaterial::whereIn('material_id', $materialIds)
        ->select('material_id', DB::raw('SUM(qty * IF(type = "in", 1, -1)) as total_stock'))
        ->groupBy('material_id')
        ->pluck('total_stock', 'material_id')
        ->toArray();

    foreach ($arrMaterials as $am) {
        $materialId = $am['material_id'];
        $materialName = $am['name'];
        $stockReduce = $qtyProduct * $am['qty'];

        $currentStock = $currentStocks[$materialId] ?? 0;

        if ($currentStock - $stockReduce < 0) {
            $errorsOutOfStocks[] = $materialName;
        } else if ($currentStock -  $stockReduce >= 0) {
            $stockOuts[] = [
                'material_id' => $materialId,
                'qty' => $stockReduce,
            ];
        }
    }
}

if (sizeof($errorsOutOfStocks) > 0) {
    return response()->json([
        'status' => false,
        'message' => "Stock material " . implode(",", $errorsOutOfStocks) . " tidak mencukupi",
    ], 404);
}
```

#2

Setelah semi lolos validasi, kemudian insert data juga dengan melakukan pengurangan pada stock material. Ini ditandai dengan type “out”.

```
$insert = new TransProduct();
$insert->name = $product->name;
$insert->price = $product->price;
$insert->total_price = $qtyProduct*$product->price;
$insert->product_id = $productId;
$insert->qty = $qtyProduct;
$insert->materials = $product->materials;
$insert->save();

foreach ($stockOuts as $key => $so) {
    $insertSm = new StockMaterial();
    $insertSm->material_id = $so['material_id'];
    $insertSm->qty = $so['qty'];
    $insertSm->type = 'out';
    $insertSm->notes = 'tp-'.$insert->id;
    $insertSm->save();
}
```

Trans Product

#3

Masukkan step 1 dan 2 ke dalam function store(). Pastikan kita menggunakan try catch serta db transcaction.

```
protected function store(Request $request)
{
    // step #1
    DB::beginTransaction();
    try {
            // step #2
        DB::commit();
        return response()->json([
            'status' => true,
            'alert' => 'success',
            'message' => 'Data Was Created Successfully',
        ], 200);
    } catch (Exception $e) {
        DB::rollBack();

        return response()->json([
                'status' => false,
                'message' => $e->getMessage(),
        ], 500);
    }
}
```

#4

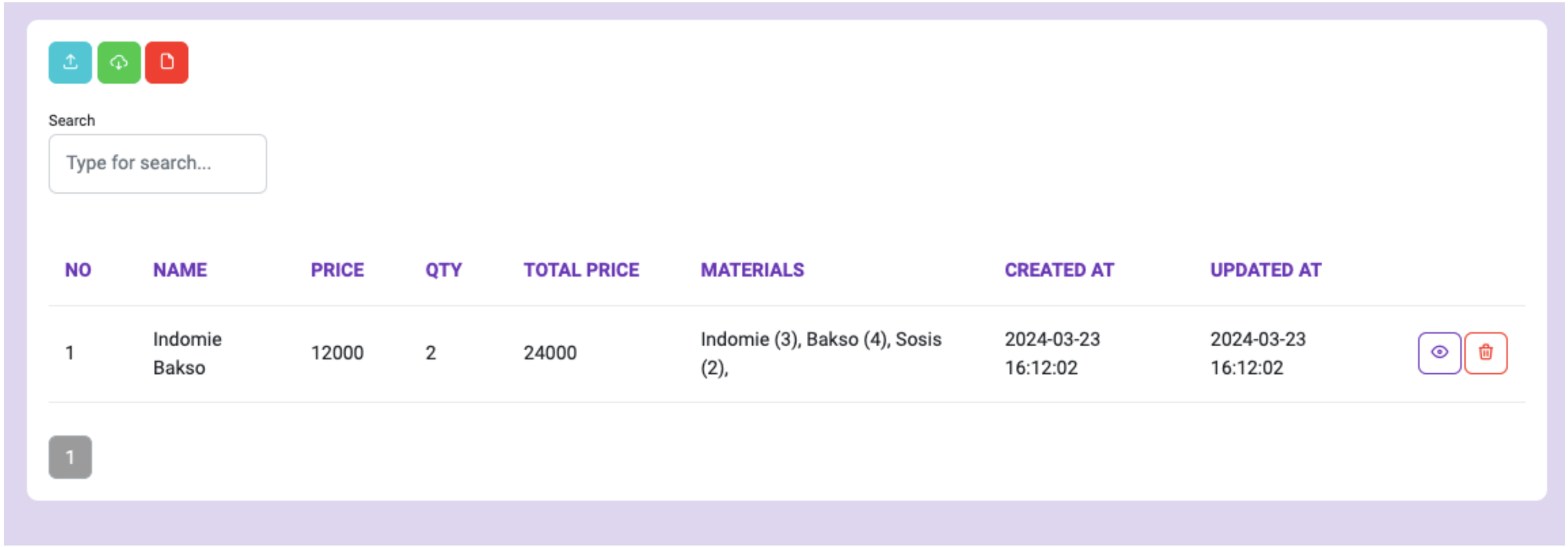
Perlu diperhatikan saat melakukan delete data, maka stock material yang terkait juga perlu dihapus.

```php
protected function destroy($id)
{
    $this->modelClass::where('id', $id)->delete();

    StockMaterial::where('notes', 'tp-'.$id)->delete();

    return response()->json([
        'status' => true,
        'alert' => 'success',
        'message' => 'Data Was Deleted Successfully',
    ], 200);
}
```

Trans Product

Lakukan kustomisasi pada Trans Product sehingga bisa menampilkan data seperti demikian :

1. Lakukan Join table untuk mendapatkan nama product
2. Lakukan append pada model untuk menampilkan materials dengan format seperti gambar



Challenge

#1

Pergerakan Stock bisa kita amati dengan menggunakan Stock Balance dengan kode sebagai berikut.

Kita perlu melakukan pemisahan antara stok keluar dan stok masuk, yakni dengan menambahkan type in / out.

Sehingga ini akan menjadi acuan dalam penjumlahan atau pengurangan kuantitas stok.

```php
protected function defaultDataQuery()
{
    $filters = [];
    $orderBy = 'material_id';
    $orderState = 'DESC';
    if (request('order')) {
        $orderBy = request('order');
        $orderState = request('order_state');
    }

    $dataQueries = $this->modelClass::where($filters)
        ->join('mst_materials', 'mst_materials.id', 'stock_materials.material_id')
        ->select('material_id',
            DB::raw('MIN(mst_materials.name) as material_name'),
            DB::raw('MAX(stock_materials.updated_at) as updated_at'),
            DB::raw('SUM(qty * IF(type = "in", 1, -1)) as total_stock'))
        ->groupBy('material_id')
        ->orderBy($orderBy, $orderState);

    return $dataQueries;
}
```

Stock Balance (BONUS)

Terimakasih

idev.project

idevsemarang.my.id