README.md

Sismenkes Bengkel Koding

Repositori tugas bengkel koding website sismenkes dengan laravel.

```
Nama : Heryawan Eko Saputro
NIM  : A11.2022.14237
```

Cara menjalankan projek secara lokal

1. Clone repositori:

```
git clone https://github.com/yourusername/RavaelaBkod.git
cd RavaelaBkod
```

2. Install dependensi PHP:

```
composer install
```

3. Install dependensi node:

```
npm install
```

4. Setup environment lokal:

```
cp .env.example .env
php artisan key:generate
```

5. Jalankan migrasi beserta seeder:

```
php artisan migrate --seed
```

6. Jalankan projek:

```
php artisan serve
```

Route Documentation

Authentication Routes

Route	Controller	Middleware	Deskripsi
/register	AuthController@showRegisterForm	-	Halaman register
/register	AuthController@register	-	Redirect register
/login	AuthController@showLoginForm	-	Halaman login
/login	AuthController@login	-	Redirect login
/logout	AuthController@logout	auth	Redirect logout

Doctor Routes

Route	Controller	Middleware	Description
/dokter/dashboard	-	auth, role:dokter	Halaman dokter
/dokter/obat	ObatController@index	auth, role:dokter	List obat
/dokter/obat/create	ObatController@create	auth, role:dokter	Halaman tambah obat
/dokter/obat	ObatController@store	auth, role:dokter	Simpan obat
/dokter/obat/{id}/edit	ObatController@edit	auth, role:dokter	Halaman edit obat
/dokter/obat/{id}	ObatController@update	auth, role:dokter	Update obat
/dokter/obat/{id}	ObatController@destroy	auth, role:dokter	Hapus obat

Patient Routes

URI	Controller	Middleware	Description
/pasien/dashboard	-	auth, role:pasien	Patient dashboard

Penjelasan implementasi RBAC

1. Role user

- Terdefinisi di kode migrasi user:

```
$table->enum('role', ['pasien', 'dokter'])->default('pasien');
```

2. Role Middleware

- Terdefinisi in `bootstrap/app.php` :

```
$middleware->alias([
    'role' => \App\Http\Middleware\RoleMiddleware::class,
]);
```

3. Proteksi Route

- Route Dokter terhalangi dengan middleware `role:dokter`
- Route Pasien terhalangi dengan middleware `role:pasien`
- Dua duanya memerlukan otentikasi melalui middleware `auth`

4. User Model Relations

```
// Relationship for patient's medical checks
public function pasiens(): HasMany
{
    return $this->hasMany(Periksa::class, 'id_pasien');
}

// Relationship for doctor's medical checks
public function dokters(): HasMany
{
    return $this->hasMany(Periksa::class, 'id_dokter');
}
```

Testing RBAC

Anda dapan memastikan apakah fungsi RBAC berfungsi atau tidak dengan:

1. Login sebagai dokter - route /dokter/* dapat diakses tetapi route /pasien/* tidak dapat diakses
2. Login sebagai pasien - route /pasien/* dapat diakses tetapi route /dokter/* tidak dapat diakses
3. User ilegal - Otomatis redirect ke halaman login jika membuka halaman yang memerlukan login

Screenshot akses ditolak

Dokter ketika mencoba mengakses halaman pasien

Skema Database & Model

File Migrasi

1. Migrasi Detail Periksa

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('detail_periksas', function (Blueprint $table) {
            $table->id();
            $table->foreignId('id_periksa')->constrained('periksas')->onDelete('cascade');
            $table->foreignId('id_obat')->constrained('obats')->onDelete('cascade');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('detail_periksas');
    }
};
```

Migrasi untuk tabel detail_periksas

2. Obat Migration

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('obats', function (Blueprint $table) {
            $table->id();
            $table->string('nama_obat');
            $table->string('kemasan');
            $table->integer('harga');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('obats');
    }
};
```

Migrasi untuk tabel obats

Model Files

1. Model Periksa

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Database\Eloquent\Relations\HasMany;

class Periksa extends Model
{
    protected $fillable = [
        'id_pasien',
        'id_dokter',
        'tgl_periksa',
        'catatan',
        'biaya_periksa',
    ];

    /**
     * Relasi ke tabel User sebagai pasien
     */
    public function pasien(): BelongsTo
    {
        return $this->belongsTo(User::class, 'id_pasien');
    }

    /**
     * Relasi ke tabel User sebagai dokter
     */
    public function dokter(): BelongsTo
    {
        return $this->belongsTo(User::class, 'id_dokter');
    }

    /**
     * Relasi ke tabel DetailPeriksa
     */
    public function detailPeriksas(): HasMany
    {
        return $this->hasMany(DetailPeriksa::class, 'id_periksa');
    }
}
```

Implementasi model Periksa dengan relasinya

2. Model Detail Periksa

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;

class DetailPeriksa extends Model
{
    protected $fillable = [
        'id_periksa',
        'id_obat',
    ];

    /**
     * Relasi ke tabel Periksa
     */
    public function periksa(): BelongsTo
    {
        return $this->belongsTo(Periksa::class, 'id_periksa');
    }

    /**
     * Relasi ke tabel Obat
     */
    public function obat(): BelongsTo
    {
        return $this->belongsTo(Obat::class, 'id_obat');
    }
}
```

Implementasi model DetailPeriksa dengan relasinya

3. Model Obat

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\HasMany;

class Obat extends Model
{
    protected $fillable = [
        'nama_obat',
        'kemasan',
        'harga'
    ];

        /**
     * Relasi ke tabel DetailPeriksa
     */
    public function detailPeriksas(): HasMany
    {
        return $this->hasMany(DetailPeriksa::class, 'id_obat');
    }
}
```

Implementasi model Obat dengan relasinya

Database Setup

Screenshot terminal menunjukkan migrasi database berhasil

User Interface

Halaman Publik

1. Landing Page
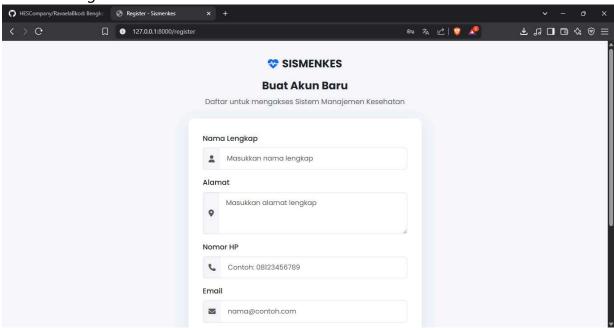

Halaman awal website yang dapat diakses publik

2. Halaman Login
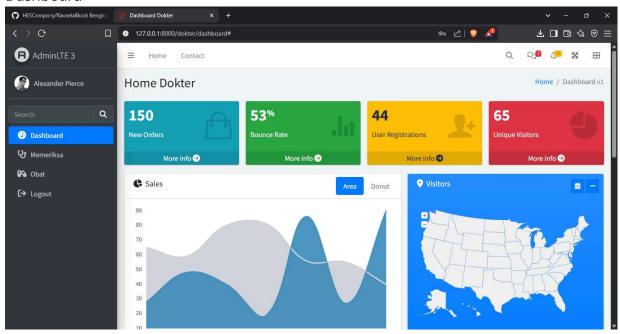

Halaman login untuk dokter dan pasien

3. Halaman Register

Halaman pendaftaran untuk pasien baru

Halaman Dokter

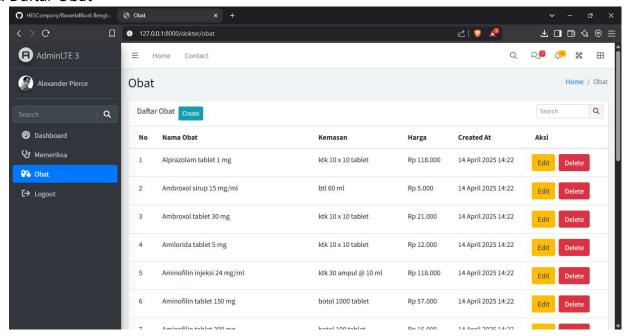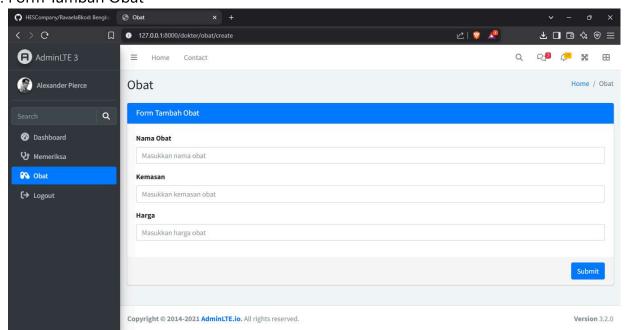### 1. Dashboard

Halaman dashboard untuk dokter

2. Daftar Obat

Halaman pengelolaan daftar obat

3. Form Tambah Obat

Form untuk menambahkan obat baru