1. Source Code

1.1 Migration

```
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
return new class extends Migration
{
    public function up(): void
    {
        Schema::create('failed_jobs', function (Blueprint $table) {
            $table->id();
            $table->string('uuid')->unique();
            $table->text('connection');
            $table->text('queue');
            $table->longText('payload');
            $table->longText('exception');
            $table->timestamp('failed_at')->useCurrent();
        });
    }
    public function down(): void
        Schema::dropIfExists('failed_jobs');
};
```

```
<?php
    use Illuminate\Database\Migrations\Migration;
    use Illuminate\Database\Schema\Blueprint;
    use Illuminate\Support\Facades\Schema;
    return new class extends Migration
        public function up(): void
            Schema::create('users', function (Blueprint $table) {
                $table->id();
                $table->string('name');
                $table->string('email')->unique();
                $table->timestamp('email_verified_at')->nullable();
                $table->string('password');
                $table->rememberToken();
                $table->timestamps();
            });
        }
        public function down(): void
            Schema::dropIfExists('users');
        }
   };
```

```
<?php
   use Illuminate\Database\Migrations\Migration;
   use Illuminate\Database\Schema\Blueprint;
   use Illuminate\Support\Facades\Schema;
   return new class extends Migration
       public function up(): void
           Schema::create('personal_access_tokens', function (Blueprint $table) {
               $table->id();
               $table->morphs('tokenable');
               $table->string('name');
               $table->string('token', 64)->unique();
               $table->text('abilities')->nullable();
                $table->timestamp('last_used_at')->nullable();
                $table->timestamp('expires_at')->nullable();
                $table->timestamps();
       public function down(): void
           Schema::dropIfExists('personal_access_tokens');
```

```
<?php
   use Illuminate\Database\Migrations\Migration;
   use Illuminate\Database\Schema\Blueprint;
   use Illuminate\Support\Facades\Schema;
   return new class extends Migration
       public function up(): void
           Schema::create('divisis', function (Blueprint $table) {
                $table->id();
                $table->string('nama_divisi');
               $table->timestamps();
           });
        }
       public function down(): void
           Schema::dropIfExists('divisi');
       }
28 };
```

```
<?php
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
   return new class extends Migration
       public function up(): void
           Schema::create('karyawans', function (Blueprint $table) {
               $table->id();
               $table->string('nama');
               $table->unsignedBigInteger('divisi_id');
               $table->foreign('divisi_id')->references('id')->on('divisis');
               $table->timestamps();
           });
       public function down(): void
           Schema::dropIfExists('karyawan');
        }
   };
```

```
namespace App\Http\Controllers;
use App\Models\Karyawan;
use App\Models\Penilaian;
use Illuminate\Http\Request;
use Illuminate\Http\RedirectResponse;
use Illuminate\Routing\Controller as BaseController;
use Illuminate\Foundation\Validation\ValidatesRequests;
use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
class Controller extends BaseController
   use AuthorizesRequests, ValidatesRequests;
   public function datatable()
       return view('datatable');
       return view('tambah-karyawan');
    public function tambahkaryawan(Request $r)
       $ky = new Karyawan();
       ky->nama = r->nama;
        $ky->divisi id = $r->divisi;
```

```
$ky->save();
$nki = Karyawan::latest()->get();
$n = new Penilaian();
$n->karyawan_id = $nki[0]->id;
$n->kompetensi_id = 1;
n->nilai = 0;
$n->save();
$n2->karyawan_id = $nki[0]->id;
$n2->kompetensi_id = 2;
n2->nilai = 0;
$n2->save();
$n3->karyawan_id = $nki[0]->id;
n3->kompetensi id = 3;
$n3->nilai = 0;
$n3->save();
$n4 = new Penilaian();
n4->karyawan id = nki[0]->id;
$n4->kompetensi_id = 4;
n4->nilai = 0;
$n4->save();
$n5 = new Penilaian();
$n5->karyawan_id = $nki[0]->id;
```

```
n5->kompetensi id = 5;
    n5->nilai = 0;
   $n5->save();
   return redirect('/dashboard');
   $ky = Karyawan::with('divisi', 'penilaian')->get();
   return view('dashboard', compact('ky'));
public function vieweditkaryawan($id)
   $k = Karyawan::find($id);
   return view('formeditkaryawan', compact('k'));
public function updatekaryawan(Request $r)
   $ky = Karyawan::find($r->id);
   x = x-  
   $ky->divisi id = $r->divisi;
   $ky->save();
   return redirect('/dashboard');
public function hapuskaryawan($id)
   $k = Karyawan::find($id);
   $k->delete();
```

```
return redirect('/dashboard');
    $ky = Karyawan::find($id);
    return view('formnilaikaryawan', compact('ky'));
public function updatenilai(Request $r)
    $x = Penilaian::where('karyawan_id', $r->id)->delete();
    $n->karyawan id = $r->id;
    n->kompetensi id = 1;
    $n->nilai = $r->penhadir;
    $n->save();
    $n2 = new Penilaian();
    n2->karyawan id = r->id;
    $n2->kompetensi_id = 2;
    $n2->nilai = $r->pentug;
    $n2->save();
    $n3 = new Penilaian();
    n3->karyawan_id = r->id;
    $n3->kompetensi_id = 3;
    $n3->nilai = $r->penked;
    $n3->save();
```

```
$n4 = new Penilaian();
$n4->karyawan_id = $r->id;
$n4->kompetensi_id = 4;
$n4->nilai = $r->penkin;
$n4->save();

$n5 = new Penilaian();
$n5->karyawan_id = $r->id;
$n5->kompetensi_id = 5;
$n5->nilai = $r->penket;
$n5->save();
return redirect('dashboard');
}
```

```
1  <?php
2
3  namespace App\Models;
4
5  use App\Models\Karyawan;
6  use Illuminate\Database\Eloquent\Model;
7  use Illuminate\Database\Eloquent\Factories\HasFactory;
8
9  class Divisi extends Model
10  {
11     use HasFactory;
12     public function karyawan()
13     {
14         return $this->hasMany(Karyawan::class);
15     }
16  }
17
```

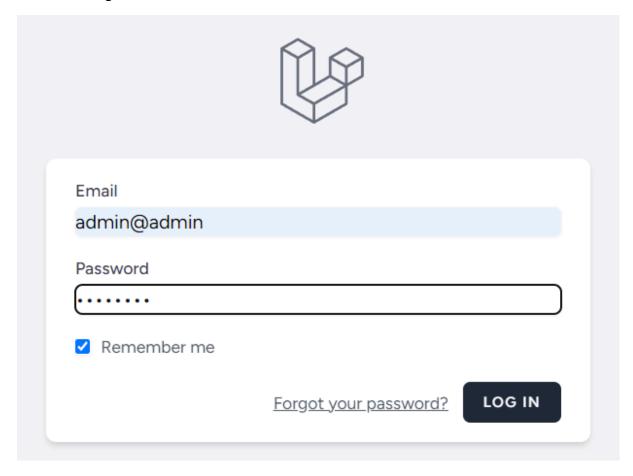
```
1 <?php
3 namespace App\Models;
5 use App\Models\Divisi;
6  use Illuminate\Database\Eloquent\Model;
   use Illuminate\Database\Eloquent\Factories\HasFactory;
9 class Karyawan extends Model
10 {
       use HasFactory;
        public function divisi()
       {
            return $this->belongsTo(Divisi::class);
        }
        public function penilaian()
           return $this->hasMany(Penilaian::class);
       }
20 }
```

```
1  <?php
2
3  namespace App\Models;
4
5  use App\Models\Karyawan;
6  use App\Models\Kompetensi;
7  use Illuminate\Database\Eloquent\Model;
8  use Illuminate\Database\Eloquent\Factories\HasFactory;
9
10  class Penilaian extends Model
11  {
12   use HasFactory;
13   public function karyawan()
14   {
15     return $this->belongsTo(Karyawan::class);
16   }
17   public function kompetensi()
18   {
19     return $this->belongsTo(Kompetensi::class);
20   }
21  }
22
```

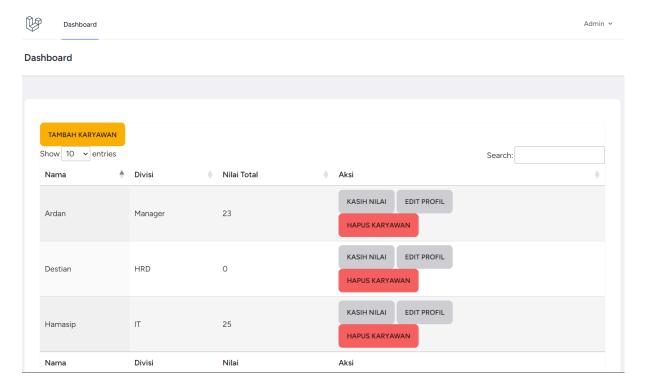
```
<?php
    namespace App\Models;
   use Illuminate\Database\Eloquent\Factories\HasFactory;
    use Illuminate\Foundation\Auth\User as Authenticatable;
    use Illuminate\Notifications\Notifiable;
    use Laravel\Sanctum\HasApiTokens;
    class User extends Authenticatable
        use HasApiTokens, HasFactory, Notifiable;
         * @var array<int, string>
        protected $fillable = [
            'name',
            'email',
            'password',
        ];
         * @var array<int, string>
        protected $hidden = [
            'password',
            'remember_token',
        ];
         * @var array<string, string>
        protected $casts = [
            'email_verified_at' => 'datetime',
            'password' => 'hashed',
        ];
45 }
```

2. Tampilan Aplikasi

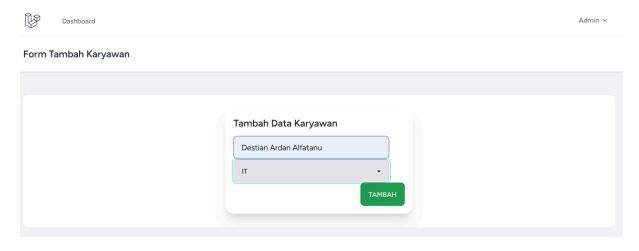
2.1 Login



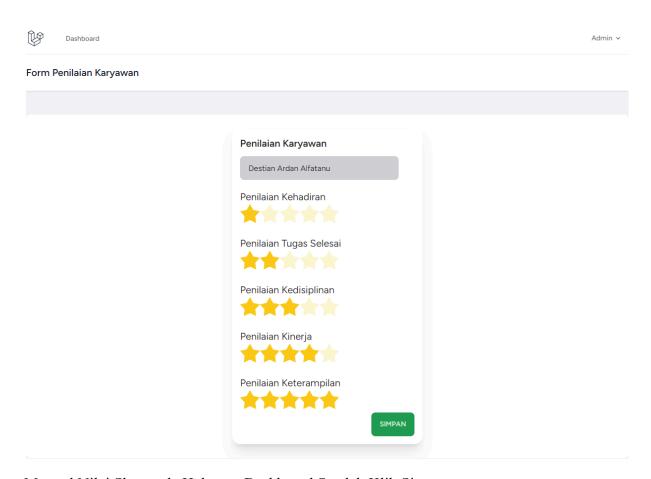
2.2 Dashboard



2.3 Tambah Karyawan



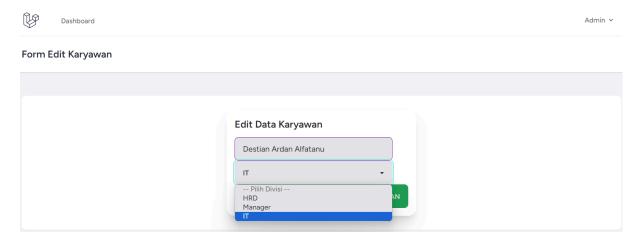
2.4 Penilaian



Muncul Nilai Skor pada Halaman Dashboard Setelah Klik Simpan



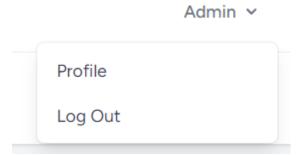
2.5 Edit Karyawan



2.6 Hapus Karyawan



2.7 Logout



3. Relasi Antar Table

