

1. Source Code

1.1 Migration

```
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  {
9      /**
10       * Run the migrations.
11       */
12     public function up(): void
13     {
14         Schema::create('failed_jobs', function (Blueprint $table) {
15             $table->id();
16             $table->string('uuid')->unique();
17             $table->text('connection');
18             $table->text('queue');
19             $table->longText('payload');
20             $table->longText('exception');
21             $table->timestamp('failed_at')->useCurrent();
22         });
23     }
24
25     /**
26      * Reverse the migrations.
27      */
28     public function down(): void
29     {
30         Schema::dropIfExists('failed_jobs');
31     }
32 };
33
```



```
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  {
9      /**
10       * Run the migrations.
11       */
12     public function up(): void
13     {
14         Schema::create('users', function (Blueprint $table) {
15             $table->id();
16             $table->string('name');
17             $table->string('email')->unique();
18             $table->timestamp('email_verified_at')->nullable();
19             $table->string('password');
20             $table->rememberToken();
21             $table->timestamps();
22         });
23     }
24
25     /**
26      * Reverse the migrations.
27      */
28     public function down(): void
29     {
30         Schema::dropIfExists('users');
31     }
32 };
33
```



```
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  {
9      /**
10       * Run the migrations.
11       */
12     public function up(): void
13     {
14         Schema::create('personal_access_tokens', function (Blueprint $table) {
15             $table->id();
16             $table->morphs('tokenable');
17             $table->string('name');
18             $table->string('token', 64)->unique();
19             $table->text('abilities')->nullable();
20             $table->timestamp('last_used_at')->nullable();
21             $table->timestamp('expires_at')->nullable();
22             $table->timestamps();
23         });
24     }
25
26     /**
27      * Reverse the migrations.
28      */
29     public function down(): void
30     {
31         Schema::dropIfExists('personal_access_tokens');
32     }
33 };
34
```



```
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  {
9      /**
10       * Run the migrations.
11       */
12     public function up(): void
13     {
14         Schema::create('divisis', function (Blueprint $table) {
15             $table->id();
16             $table->string('nama_divisi');
17             $table->timestamps();
18         });
19     }
20
21     /**
22      * Reverse the migrations.
23      */
24     public function down(): void
25     {
26         Schema::dropIfExists('divisi');
27     }
28 };
29
```

```
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  {
9      /**
10       * Run the migrations.
11       */
12     public function up(): void
13     {
14         Schema::create('karyawans', function (Blueprint $table) {
15             $table->id();
16             $table->string('nama');
17             $table->unsignedBigInteger('divisi_id');
18             $table->foreign('divisi_id')->references('id')->on('divisis');
19             $table->timestamps();
20         });
21     }
22
23     /**
24      * Reverse the migrations.
25      */
26     public function down(): void
27     {
28         Schema::dropIfExists('karyawan');
29     }
30 };
31
```



```
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  {
9      /**
10       * Run the migrations.
11       */
12     public function up(): void
13     {
14         Schema::create('kompetensi', function (Blueprint $table) {
15             $table->id();
16             $table->string('kompetensi');
17             $table->timestamps();
18         });
19     }
20
21     /**
22      * Reverse the migrations.
23      */
24     public function down(): void
25     {
26         Schema::dropIfExists('kompetensi');
27     }
28 };
29
```

```
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  {
9      /**
10       * Run the migrations.
11       */
12     public function up(): void
13     {
14         Schema::create('penilais', function (Blueprint $table) {
15             $table->id();
16             $table->unsignedBigInteger('karyawan_id');
17             $table->foreign('karyawan_id')->references('id')->on('karyawans')->onDelete('cascade');
18             $table->unsignedBigInteger('kompetensi_id');
19             $table->foreign('kompetensi_id')->references('id')->on('kompetensis');
20             $table->integer('nilai');
21             $table->timestamps();
22         });
23     }
24
25     /**
26      * Reverse the migrations.
27      */
28     public function down(): void
29     {
30         Schema::dropIfExists('penilaian');
31     }
32 };
33
```



```
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  {
9      /**
10       * Run the migrations.
11       */
12     public function up(): void
13     {
14         Schema::create('password_reset_tokens', function (Blueprint $table) {
15             $table->string('email')->primary();
16             $table->string('token');
17             $table->timestamp('created_at')->nullable();
18         });
19     }
20
21     /**
22      * Reverse the migrations.
23      */
24     public function down(): void
25     {
26         Schema::dropIfExists('password_reset_tokens');
27     }
28 };
29
```


1.2 CRUD

```
<?php

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');
    }

    public function viewtambahkaryawan()
    {
        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 = new Penilaian();

$n2->karyawan_id = $nki[0]->id;

$n2->kompetensi_id = 2;

$n2->nilai = 0;

$n2->save();

$n3 = new Penilaian();

$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');

}

public function lihatdash()

{

    $ky = Karyawan::with('divisi', 'penilaian')->get();

    // dd($ky[0]->penilaian[0]->nilai);

    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);

    $ky->nama = $r->nama;

    $ky->divisi_id = $r->divisi;

    $ky->save();

    return redirect('/dashboard');

}

public function hapuskaryawan($id)

{

    $k = Karyawan::find($id);

    $k->delete();

}
```

```
        return redirect('/dashboard');
    }

    public function viewnilaikaryawan($id)
    {
        $ky = Karyawan::find($id);

        return view('formnilaikaryawan', compact('ky'));
    }

    public function updatenilai(Request $r)
    {
        $x = Penilaian::where('karyawan_id', $r->id)->delete();

        $n = new Penilaian();
        $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');

}

}
```

1.3 Model



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



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



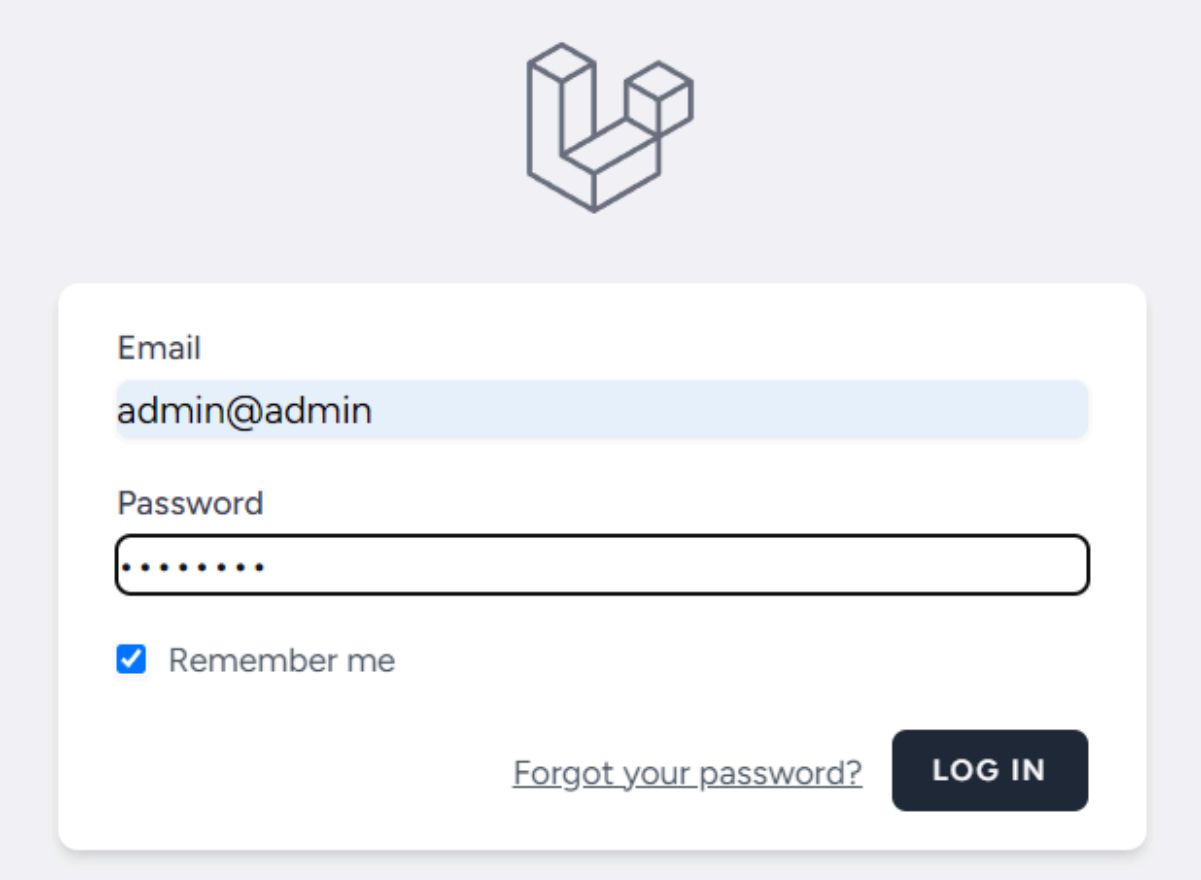

```
1  <?php
2
3  namespace App\Models;
4
5  use App\Models\Karyawan;
6  use App\Models\Kompetensi;
7  use Illuminate\Database\Eloquent\Model;
8  use Illuminate\Database\Eloquent\Factories\HasFactory;
9
10 class Penilaian extends Model
11 {
12     use HasFactory;
13     public function karyawan()
14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



```
1  <?php
2
3  namespace App\Models;
4
5  // use Illuminate\Contracts\Auth\MustVerifyEmail;
6  use Illuminate\Database\Eloquent\Factories\HasFactory;
7  use Illuminate\Foundation\Auth\User as Authenticatable;
8  use Illuminate\Notifications\Notifiable;
9  use Laravel\Sanctum\HasApiTokens;
10
11 class User extends Authenticatable
12 {
13     use HasApiTokens, HasFactory, Notifiable;
14
15     /**
16      * The attributes that are mass assignable.
17      *
18      * @var array<int, string>
19      */
20     protected $fillable = [
21         'name',
22         'email',
23         'password',
24     ];
25
26     /**
27      * The attributes that should be hidden for serialization.
28      *
29      * @var array<int, string>
30      */
31     protected $hidden = [
32         'password',
33         'remember_token',
34     ];
35
36     /**
37      * The attributes that should be cast.
38      *
39      * @var array<string, string>
40      */
41     protected $casts = [
42         'email_verified_at' => 'datetime',
43         'password' => 'hashed',
44     ];
45 }
46
```

2. Tampilan Aplikasi

2.1 Login



The login form is centered on a light gray background. At the top, there is a 3D cube logo. Below it, the form contains an email input field with the text 'admin@admin', a password input field with masked dots, a 'Remember me' checkbox, a 'Forgot your password?' link, and a 'LOG IN' button.


Email
admin@admin

Password
.....

☒ Remember me

[Forgot your password?](#) **LOG IN**

2.2 Dashboard

 Dashboard Admin ▾

Dashboard

TAMBAH KARYAWAN				Search: <input type="text"/>
Show 10 ▾ entries	Nama	Divisi	Nilai Total	Aksi
	Ardan	Manager	23	<div>KASIH NILAI EDIT PROFIL</div> <div>HAPUS KARYAWAN</div>
	Destian	HRD	0	<div>KASIH NILAI EDIT PROFIL</div> <div>HAPUS KARYAWAN</div>
	Hamasip	IT	25	<div>KASIH NILAI EDIT PROFIL</div> <div>HAPUS KARYAWAN</div>
	Nama	Divisi	Nilai	Aksi

2.3 Tambah Karyawan



Dashboard

Admin ▾

Form Tambah Karyawan

Tambah Data Karyawan

Destian Ardan Alfatanu

IT ▾

TAMBAH

2.4 Penilaian



Dashboard

Admin ▾

Form Penilaian Karyawan

Penilaian Karyawan

Destian Ardan Alfatanu

Penilaian Kehadiran

★★★★★

Penilaian Tugas Selesai

★★★★★

Penilaian Kedisiplinan

★★★★★

Penilaian Kinerja

★★★★★

Penilaian Keterampilan

★★★★★

SIMPAN

Muncul Nilai Skor pada Halaman Dashboard Setelah Klik Simpan

Destian Ardan Alfatanu	IT	15
------------------------	----	----

2.5 Edit Karyawan



Dashboard

Admin ▾

Form Edit Karyawan

Edit Data Karyawan

Destian Ardan Alfatanu

IT ▾

-- Pilih Divisi --
HRD
Manager
IT

SAV

2.6 Hapus Karyawan

Destian Ardan Alfatanu	IT	15	KASIH NILAI	EDIT PROFIL	HAPUS KARYAWAN
------------------------	----	----	-------------	-------------	----------------

2.7 Logout

Admin ▾

Profile

Log Out

3. Relasi Antar Table

