LISTING PROGRAM

AbsensiController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Absensis;
use App\Models\Pegawais;
use Carbon\Carbon;
use Illuminate\Http\Request;
use Barryvdh\DomPDF\Facade\Pdf;

class AbsensiController extends
Controller
{
    public function index()
    {
        $absensis =
Absensis::all();

        // $hadir = $absensis-
>where('status', '1')->count();
        // $izin = $absensis-
>where('status', '2')->count();
        // $alpha = $absensis-
>where('status', '3')->count();

        // $totalDays = $hadir
+ $izin + $alpha;

        // // Calculate the
realisasi_pendapatan based on
the rules
        // $totalDiscount =
$izin * 2 + $alpha * 3; //
Total discount based on
occurrences of izin and alpha

        // $realisasiPendapatan
= 0; // Initialize the variable

        // if ($totalDays > 0)
{
        //     $presentPercenta
ge = ($hadir / $totalDays) *
100;
        //     $realisasiPendap
atan = (100 - $totalDiscount) /
100 * 10000000;
        // }

        $groupedAbsensis =
$absensis-
>groupBy('id_pegawai');

        // Inisialisasi sebuah
array untuk menyimpan
realisasi_pendapatan untuk
setiap individu
        $realisasiPendapatanPer
Nama = [];

        foreach
($groupedAbsensis as $nama =>
$absensiGroup) {
            $hadir =
$absensiGroup->where('status',
'1')->count();
            $izin =
$absensiGroup->where('status',
'2')->count();
            $alpha =
$absensiGroup->where('status',
'3')->count();

            $totalDays = $hadir
+ $izin + $alpha;

            // Hitung
realisasi_pendapatan
berdasarkan aturan yang telah
ditetapkan
            $totalDiscount =
$izin * 2 + $alpha * 3;
            $realisasiPendapata
n = 0; // Inisialisasi variabel
```

```
            if ($totalDays > 0)
{
                $presentPercent
age = ($hadir / $totalDays) *
100;
                $realisasiPenda
patan = (100 - $totalDiscount)
/ 100 * 10000000;
            }

            // Simpan
realisasi_pendapatan untuk nama
ini dalam array
            $realisasiPendapata
nPerNama[$nama] =
$realisasiPendapatan;
        }

        //
var_dump($realisasiPendapatan);
        // Update the
realisasi_pendapatan in the
database

        return
view('absensi.index',
compact('absensis',
'realisasiPendapatan'));
    }

    public function create()
    {
        $pegawai =
Pegawais::all();
        $absensis =
Absensis::all();
        return
view('absensi.create',
compact('absensis',
'pegawai'));
    }

    // insert data to table
```

```
    public function
store(Request $request)
    {
        // dd($request->all());
        // data yg akan
diterima function store
        $today = Carbon::now()-
>format('Y-m-d');
        $id_pegawai = $request-
>id_pegawai;
        $status = $request-
>status;

        // validasi sebelum
insert ke tabel
        $request->validate([
            'id_pegawai' =>
'required|min:1',
            'status' =>
'required',
        ]);

        // buat object untuk
simpan data ke table
        $absensis = new
Absensis();

        //kirim nilai2 yg
didapat dari form ke table
        $absensis->id_pegawai =
$id_pegawai;
        $absensis->tanggal =
$today;
        $absensis->status =
$status;
        // simpan data yg telah
diterima ke dalam table
        $absensis->save();

        // redirect ke halaman
users
        return
redirect('/absensi')->with(
            'success',
            'Absensi Berhasil'
        );
    }

    public function edit($id)
    {
        $absensis =
Absensis::find($id); // SELECT
* FROM user WHERE id = $id
        // dd($user);
        $pegawai =
Pegawais::all();
        // tampilkan form edit
dan kirim datanya
        return
view('absensi.edit',
compact('absensis',
'pegawai'));
    }

    // update data selected
    public function
update(Request $request)
    {
        // data yg akan
diterima function store

        $idabsensi = $request-
>idabsensi;
        $today = Carbon::now()-
>format('Y-m-d');
        $id_pegawai = $request-
>id_pegawai;
        $status = $request-
>status;


        // buat object untuk
simpan data ke table
        $absensis =
Absensis::find($idabsensi);

        //kirim nilai2 yg
didapat dari form ke table
        $absensis->id_pegawai =
$id_pegawai;
        $absensis->tanggal =
$today;
        $absensis->status =
$status;

        // simpan data yg telah
diterima ke dalam table
        // $penerima->save();

        $absensis->update();

        // redirect ke halaman
users
        return
redirect('/absensi');
    }

    public function
delete($idabsensi)
    {
        // query/perintah hapus
data berdasarkan id
        Absensis::find($idabsen
si)->delete();

        // kembalikan ke
halaman users
        return
redirect('/absensi');
    }

    public function penerima()
    {
        $penerima =
Absensis::paginate(10);
        return
view('kepaladinas.penerima',
compact('penerima'));
    }


    public function
absensipdf($idabsensi)
    {
        $absensis =
Absensis::findOrFail($idabsensi
);

        $pdf =
PDF::loadview('absensi.absensip
df', ['absensis' =>
$absensis])-
>setOptions(['defaultFont' =>
'sans-serif']);
        return $pdf-
>download('laporan_Absensi.pdf'
);
    }
}
```

JabatanController.php

```
<?php

namespace App\Http\Controllers;


use App\Models\Jabatans;
use Illuminate\Http\Request;
use Barryvdh\DomPDF\Facade\Pdf;
use PDO;

class JabatanController extends
Controller
{
    public function index()
    {
        $jabatans =
Jabatans::all();
        return
view('jabatan.index',
compact('jabatans'));
    }

    public function create()
    {
        $jabatans =
Jabatans::all();
        return
view('jabatan.create',
compact('jabatans'));
    }

    // insert data to table
    public function
store(Request $request)
    {
        // dd($request->all());
        // data yg akan
diterima function store
        $kelasjabatan =
$request->kelasjabatan;
        $jabatan = $request-
>jabatan;

        // validasi sebelum
insert ke tabel
        $request->validate([
            'kelasjabatan' =>
'required|min:1',
            'jabatan' =>
'required',
        ]);

        // buat object untuk
simpan data ke table
        $jabatans = new
Jabatans();

        //kirim nilai2 yg
didapat dari form ke table
        $jabatans->kelasjabatan
= $kelasjabatan;
        $jabatans->jabatan =
$jabatan;
        // simpan data yg telah
diterima ke dalam table
        $jabatans->save();

        // redirect ke halaman
users
        return
redirect('/jabatan');
```

```
    }

    public function edit($id)
    {
        $jabatans =
Jabatans::find($id); // SELECT
* FROM user WHERE id = $id
        // dd($user);

        // tampilkan form edit
dan kirim datanya
        return
view('jabatan.edit',
compact('jabatans'));
    }

    // update data selected
    public function
update(Request $request)
    {
        // data yg akan
diterima function store

        $idjabatan = $request-
>idjabatan;
        $kelasjabatan =
$request->kelasjabatan;
        $jabatan = $request-
>jabatan;


        // buat object untuk
simpan data ke table
        $jabatans =
Jabatans::find($idjabatan);

        //kirim nilai2 yg
didapat dari form ke table
        $jabatans->kelasjabatan
= $kelasjabatan;
        $jabatans->jabatan =
$jabatan;;

        // simpan data yg telah
diterima ke dalam table
        // $penerima->save();

        $jabatans->update();

        // redirect ke halaman
users
        return
redirect('/jabatan');
    }

    public function
delete($idpenerima)
    {
        // query/perintah hapus
data berdasarkan id
        Jabatans::find($idpener
ima)->delete();

        // kembalikan ke
halaman users
        return
redirect('/jabatan');
    }

    public function penerima(){
        $penerima =
Jabatans::paginate(10);
        return
view('kepaladinas.penerima',
compact('penerima'));
    }

    public function
jabatanpdf()
    {
        $jabatans =
Jabatans::all();

        $pdf =
PDF::loadview('jabatan.jabatanp
df', ['jabatans' =>
$jabatans])-
>setOptions(['defaultFont' =>
'sans-serif']);
```

```
        return $pdf-
>download('laporan_jabatan.pdf'
);
        // return
view('aset.manager_pdf', [
        // 'aset' => $aset,
        // ]);

    }
}
```

LoginController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Absensis;
use App\Models\Jabatans;
use App\Models\Jenis;
use App\Models\Pegawais;
use App\Models\Penerima;
use App\Models\Realisasi;
use App\Models\Sumber;
use App\Models\User;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Auth
;

class LoginController extends
Controller
{
    public function
postlogin(Request $request)
    {
        // dd($request->all());
        if
(Auth::attempt($request-
>only('username', 'password')))
{
            switch
(Auth::user()['level']) {
                case 'Admin':
                    return
redirect('/dashboard');
                    break;
                case 'Pegawai':
                    return
redirect('/dashboard');
                    break;
                case
'Kepalasdm':
                    return
redirect('/dashboard');
                    break;
            }
        }
        return redirect('/');
    }


    public function welcome()
    {
        $totalPegawai =
Pegawais::count();
        $totalJabatan =
Jabatans::count();
        $totalAbsensi =
Absensis::count();
        $totalRealisasi =
Realisasi::count();
        $totalUser =
User::count();
        return view('welcome',
compact('totalPegawai',
'totalJabatan', 'totalAbsensi',
'totalRealisasi',
'totalUser'));
    }


    public function
logout(Request $request)
    {
        Auth::logout();

        request()->session()-
>invalidate();
```

```
        request()->session()-
>regenerateToken();

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

PegawaiController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Jabatans;
use App\Models\Pegawais;
use Illuminate\Http\Request;
use Barryvdh\DomPDF\Facade\Pdf;

class PegawaiController extends
Controller
{
    public function index()
    {
        $jabatan =
Jabatans::all();
        $pegawais =
Pegawais::all();
        return
view('pegawai.index',
compact('pegawais',
'jabatan'));
    }

    public function create()
    {
        $jabatan =
Jabatans::all();
        $Pegawais =
Pegawais::all();
        return
view('pegawai.create',
compact('Pegawais',
'jabatan'));
    }

    // insert data to table
    public function
store(Request $request)
    {
        // dd($request->all());
        // data yg akan
diterima function store
        $nip = $request->nip;
        $nama = $request->nama;
        $id_jabatan = $request-
>id_jabatan;
        $gaji = $request->gaji;
        $status = $request-
>status;

        // validasi sebelum
insert ke tabel
        $request->validate([
            'nip' =>
'required|min:1',
            'nama' =>
'required',


            'id_jabatan' =>
'required',
            'gaji' =>
'required',
            'status' =>
'required',
        ]);

        // buat object untuk
simpan data ke table
        $Pegawais = new
Pegawais();

        //kirim nilai2 yg
didapat dari form ke table
        $Pegawais->nip = $nip;
        $Pegawais->nama =
$nama;
```

```
        $Pegawais->id_jabatan =
$id_jabatan;
        $Pegawais->gaji =
$gaji;
        $Pegawais->status =
$status;
        // simpan data yg telah
diterima ke dalam table
        $Pegawais->save();

        // redirect ke halaman
users
        return
redirect('/pegawai');
    }

    public function edit($id)
    {
        $jabatan =
Jabatans::all();
        $pegawais =
Pegawais::find($id); // SELECT
* FROM user WHERE id = $id
        // dd($user);

        // tampilkan form edit
dan kirim datanya
        return
view('pegawai.edit',
compact('pegawais',
'jabatan'));
    }

    // update data selected
    public function
update(Request $request)
    {
        // data yg akan
diterima function store
        $idpegawai = $request-
>idpegawai;
        $nip = $request->nip;
        $nama = $request->nama;
        $id_jabatan = $request-
>id_jabatan;
        $gaji = $request->gaji;
        $status = $request-
>status;


        // buat object untuk
simpan data ke table
        $pegawais =
Pegawais::find($idpegawai);

        //kirim nilai2 yg
didapat dari form ke table
        $pegawais->nip = $nip;
        $pegawais->nama =
$nama;

        $pegawais->id_jabatan =
$id_jabatan;
        $pegawais->gaji =
$gaji;
        $pegawais->status =
$status;

        // simpan data yg telah
diterima ke dalam table
        // $penerima->save();

        $pegawais->update();

        // redirect ke halaman
users
        return
redirect('/pegawai');
    }

    public function
delete($idpegawai)
    {
        // query/perintah hapus
data berdasarkan id
        Pegawais::find($idpegaw
ai)->delete();
```

```
        // kembalikan ke
halaman users
        return
redirect('/pegawai');
    }

    public function penerima()
    {
        $penerima =
Jabatans::paginate(10);
        return
view('kepaladinas.penerima',
compact('penerima'));
    }



    public function
pegawaipdf($idpegawai)
    {
        $pegawais =
Pegawais::findOrFail($idpegawai
);

        $pdf =
PDF::loadView('pegawai.pegawaip
df', ['pegawais' =>
$pegawais])-
>setOptions(['defaultFont' =>
'sans-serif']);
        return $pdf-
>download('laporan_Pegawai.pdf'
);
    }
}
```

RealisasiController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Absensis;
use App\Models\Pegawais;
use App\Models\Realisasi;
use Illuminate\Http\Request;
use Barryvdh\DomPDF\Facade\Pdf;

class RealisasiController
extends Controller
{
    public function index()
    {
        $realisasis =
Realisasi::with('pegawai.absens
is')->get();
        return
view('realisasi.index',
compact('realisasis'));
    }

    public function create()
    {
        $pegawai =
Pegawais::all();
        $realisasis =
Realisasi::paginate(10);
        return
view('realisasi.create',
compact('realisasis',
'pegawai'));
    }

    // insert data to table
    public function
store(Request $request)
    {
        // dd($request->all());
        // data yg akan
diterima function store
        $tanggal = $request-
>tanggal;
        // $idtukin = $request-
>idtukin;
        $id_pegawai = $request-
>id_pegawai;
        // $tunjangan =
$request->tunjangan;
```

```
        // validasi sebelum
insert ke tabel
        $request->validate([
            'tanggal' =>
'required|min:1',
            // 'idtukin' =>
'required',
            'id_pegawai' =>
'required',
            // 'tunjangan' =>
'required',
        ]);

        // buat object untuk
simpan data ke table
        $realisasis = new
Realisasi();

        //kirim nilai2 yg
didapat dari form ke table
        $realisasis->tanggal =
$tanggal;
        // $realisasis->idtukin
= $idtukin;
        $realisasis->id_pegawai
= $id_pegawai;
        // $realisasis-
>tunjangan = $tunjangan;
        // simpan data yg telah
diterima ke dalam table
        $realisasis->save();

        // redirect ke halaman
users
        return
redirect('/realisasi');
    }

    public function edit($id)
    {
        $pegawai =
Pegawais::all();
        $realisasis =
Realisasi::find($id); // SELECT
* FROM user WHERE id = $id
```

```
        // dd($user);

        // tampilkan form edit
dan kirim datanya
        return
view('realisasi.edit',
compact('realisasis',
'pegawai'));
    }

    // update data selected
    public function
update(Request $request)
    {
        // data yg akan
diterima function store
        $idrealisasi =
$request->idrealisasi;
        $tanggal = $request-
>tanggal;
        // $idtukin = $request-
>idtukin;
        $id_pegawai = $request-
>id_pegawai;
        // $tunjangan =
$request->tunjangan;


        // buat object untuk
simpan data ke table
        $realisasis =
Realisasi::find($idrealisasi);

        //kirim nilai2 yg
didapat dari form ke table
        $realisasis->tanggal =
$tanggal;
        // $realisasis->idtukin
= $idtukin;
        $realisasis->id_pegawai
= $id_pegawai;
        // $realisasis-
>tunjangan = $tunjangan;
```

```
        // simpan data yg telah
diterima ke dalam table
        // $penerima->save();

        $realisasis->update();

        // redirect ke halaman
users
        return
redirect('/realisasi');
    }

    public function
delete($idrealisasi)
    {
        // query/perintah hapus
data berdasarkan id
        Realisasi::find($idreal
isasi)->delete();

        // kembalikan ke
halaman users
        return
redirect('/realisasi');
    }

    public function penerima()
    {
        $penerima =
Realisasi::paginate(10);
        return
view('kepaladinas.penerima',
compact('penerima'));
    }



    public function
realisasipdf($idrealisasi)
    {
        $realisasis =
Realisasi::findOrFail($idrealis
asi);
        $pdf =
PDF::loadview('realisasi.realis
asipdf', ['realisasis' =>
$realisasis])-
>setOptions(['defaultFont' =>
'sans-serif']);
        return $pdf-
>download('laporan_Realisasi.pd
f');
    }
}
```

UserController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\User;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Auth
;
use
Illuminate\Support\Facades\DB;
use
Illuminate\Support\Facades\Hash
;

class UserController extends
Controller
{
    public function index()
    {
        //ambil data dari tabel
users

        // Eloquent
(menggunakan model)
        $users = User::all();
        // dd($users);

        // Query builder
        // SELECT * FROM USER
        // $users =
DB::table('users')->get();

        return
view('user.index', [
            'users' => $users,
        ]);
    }

    // show add form
    public function create()
    {
        return
view('user.add');
    }

    // insert data to table
    public function
store(Request $request)
    {
        // data yg akan
diterima function store
        $username = $request-
>username;
        $level = $request-
>level;
        $password =
Hash::make($request->password);

        // validasi sebelum
insert ke tabel
        $request->validate([
            'username' =>
'required|min:3',
            'level' =>
'required',
            'password' =>
'required'
        ]);

        // buat object untuk
simpan data ke table
        $user = new User();
```

```
        //kirim nilai2 yg
didapat dari form ke table
        $user->username =
$username;
        $user->level = $level;
        $user->password =
$password;

        // simpan data yg telah
diterima ke dalam table
        $user->save();

        // redirect ke halaman
users
        return
redirect('/users');
    }

    public function edit($id)
    {
        $user =
User::find($id); // SELECT *
FROM user WHERE id = $id
        // dd($user);

        // tampilkan form edit
dan kirim datanya
        return
view('user.edit',
compact('user'));
    }

    // update data selected
    public function
update(Request $request)
    {
        // data yg akan
diterima function store

        $id = $request->id;
        $username = $request-
>username;
        $level = $request-
>level;
```

```
        // buat object untuk
simpan data ke table
        $user =
User::find($id);

        if($request->password)
{
            $password =
Hash::make($request->password);
        } else {
            $password = $user-
>password;
        }

        //kirim nilai2 yg
didapat dari form ke table
        $user->username =
$username;
        $user->level = $level;
        $user->password =
$password;

        // simpan data yg telah
diterima ke dalam table
        // $user->save();

        $user->update();

        // redirect ke halaman
users
        return
redirect('users');
    }

    public function delete($id)
    {
        // query/perintah hapus
data berdasarkan id
        User::find($id)-
>delete();

        // kembalikan ke
halaman users
        return
redirect('users');
```

```
    }

    public function profile()
    {
        $user =
User::find(Auth::user()->id);
        return
view('profile.profile',
compact('user'));
    }

    // update data selected
    public function
updateProfile(Request $request)
    {
        // data yg akan
diterima function store

        $id = $request->id;
        $name = $request->name;
        $email = $request-
>email;
        $jk = $request->jk;
        $no_telp = $request-
>no_telp;

        // buat object untuk
simpan data ke table
        $user =
User::find($id);

        if($request->password)
{
            $password =
Hash::make($request->password);
        } else {
            $password = $user-
>password;
        }
        //kirim nilai2 yg
didapat dari form ke table
        $user->name = $name;
        $user->email = $email;
        $user->jk = $jk;
        $user->no_telp =
$no_telp;
        $user->password =
$password;

        // simpan data yg telah
diterima ke dalam table
        // $user->save();

        $user->update();

        // redirect ke halaman
users
        return
redirect('profile')->with([
            'pesan' => 'Data
Berhasil diupdate!',
        ]);
    }
}
```

Absensipdf.blade.php

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-
Compatible" content="IE=edge">
    <meta name="viewport"
content="width=device-width,
initial-scale=1.0">
    <title>Document</title>
</head>

<body>
    <header>
        <center style="margin-
bottom: 10px; font-size: 24px;
font-weight: bold;">Laporan
Presensi Pada Pegawai Badan
Kepegawaian Dan Pengembangan
SDM Kota Palembang</center>
    </header></center>
    </header>
    <table id="example1"
class="table table-bordered
table-striped" border="1">
        <thead>
            <tr>

                <th>NIP</th>
                <th>Nama</th>
                <th>Tanggal
Presensi</th>
                <th>Status</th>
            </tr>
        </thead>
        <tbody>

            <tr>

                <td>{{
$absensis->pegawais->nip}}</td>
                <td>{{
$absensis->pegawais-
>nama}}</td>
                <td>{{
$absensis->tanggal }}</td>
                <td>{{
$absensis->status  }}</td>
            </tr>


        </tbody>
    </table>
</body>

</html>
```

Jabatanpdf.blade.php

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-
Compatible" content="IE=edge">
    <meta name="viewport"
content="width=device-width,
initial-scale=1.0">
    <title>Document</title>
</head>

<body>
    <header>
        <center style="margin-
bottom: 10px; font-size: 24px;
font-weight: bold;"> Laporan
Kategori Pada Badan Kepegawaian
Dan Pengembangan SDM Kota
Palembang</center>
    </header></center>
    </header>
    <center>
        <table id="example1"
class="table table-bordered
table-striped" border="1">
            <thead>
                <tr>
                    <th>No</th>
                    <th>Kelas
Jabatan</th>
                    <th>Jabatan
</th>
                </tr>
            </thead>
            <tbody>
                @forelse
($jabatans as $key => $pm)
                <tr>
                    <td>{{ $key
+ 1 }}</td>
                    <td>{{ $pm-
>kelasjabatan }}</td>
                    <td>{{ $pm-
>jabatan }}</td>
                </tr>
                @empty
                <tr>
                    <td
colspan="5" class="text-
center">Data tidak ada</td>
                </tr>
                @endforelse

            </tbody>
        </table>
    </center>

</body>

</html>
```

Pegawaipdf.blade.php

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-
Compatible" content="IE=edge">
    <meta name="viewport"
content="width=device-width,
initial-scale=1.0">
    <title>Document</title>
</head>

<body>
    <header>
        <center style="margin-
bottom: 10px; font-size: 24px;
font-weight: bold;"> Laporan
Pegawai Pada Badan Kepegawaian
Dan Pengembangan SDM Kota
Palembang</center>
    </header>
    <table id="example1"
class="table table-bordered
table-striped" border="1">
        <thead>
            <tr>
```

```
                <th>NIP</th>
                <th>Nama</th>
                <th>Jabatan</th>
                <th>Status</th>
            </tr>
        </thead>
        <tbody>

            <tr>

                <td>{{ $pegawais->nip }}</td>
                <td>{{ $pegawais->nama }}</td>
                <td>{{ $pegawais->jabatans->kelasjabatan}} - {{ $pegawais->jabatans->jabatan}}</td>
                <td>{{ $pegawais->status }}</td>
            </tr>


        </tbody>
    </table>
</body>

</html>
```

Realisasipdf.blade.php

```
DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>

<body>
    <header>
        <center style="margin-bottom: 10px; font-size: 24px; font-weight: bold;">Laporan Realisasi Tunjangan Kinerja Pada Pegawai Badan Kepegawaian Dan Pengembangan SDM Kota Palembang</center>
    </header></center>
    </header>
    <table id="example1" class="table table-bordered table-striped" border="1">
    <thead>

 <tr>


     <th>Tanggal</th>

     <!-- <th>ID Tukin</th> -->

     <th>NIP</th>

     <th>Nama</th>

     <th>Gaji</th>

     <th>Tunjangan Kinerja</th>



 </tr>
                            </thead>
                            <tbody>


 @php

 $pegawai = $realisasis->pegawai;
 $absensis = $pegawai-
>absensis;

 $hadir = $absensis-
>where('status', '1')->count();

 $izin = $absensis-
>where('status', '2')->count();

 $alpha = $absensis-
>where('status', '3')->count();


 $totalDays = $hadir + $izin +
$alpha;


 // Calculate the
realisasi_pendapatan based on
the rules

 $totalDiscount = $izin * 2 +
$alpha * 3; // Total discount
based on occurrences of izin
and alpha


 $realisasiPendapatan = 0; //
Initialize the variable


 if ($totalDays > 0) {

 $presentPercentage = ($hadir /
$totalDays) * 100;

 $realisasiPendapatan = (100 -
$totalDiscount) / 100 *
10000000;

 }

 @endphp
```

```
 <tr>



     <td>{{ $realisasis-
>tanggal }}</td>

     <td>{{ $pegawai->nip
}}</td>

     <td>{{ $pegawai->nama
}}</td>

     <td>{{ $pegawai->gaji
}}</td>

     <td>Rp {{
number_format($realisasiPendapa
tan, 2, ',', '.') }}</td>



 </tr>

                            </t
body>
    </table>
</body>

</html>
```

Login.blade.php

```
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport"
content="width=device-width,
initial-scale=1">
    <meta name="description"
content="">
```

```
    <meta name="author"
content="Mark Otto, Jacob
Thornton, and Bootstrap
contributors">
    <meta name="generator"
content="Hugo 0.84.0">
    <title>Login</title>

    <link
href="https://cdn.jsdelivr.net/
npm/bootstrap@5.3.0-
alpha1/dist/css/bootstrap.min.c
ss" rel="stylesheet"
integrity="sha384-
GLhlTQ8iRABdZLl6O3oVMWSktQOp6b7
In1Zl3/Jr59b6EGGoI1aFkw7cmDA6j6
gD" crossorigin="anonymous">
    <link rel="stylesheet"
href="assets/css/login.css">



    <style>
        .bd-placeholder-img {
            font-size:
1.125rem;
            text-anchor:
middle;
            -webkit-user-
select: none;
            -moz-user-select:
none;
            user-select: none;
        }

        @media (min-width:
768px) {
            .bd-placeholder-
img-lg {
                font-size:
3.5rem;
            }
        }
    </style>
```

```
    <!-- Custom styles for this
template -->
    <link
href="assets/css/style.css"
rel="stylesheet">
</head>

<body class="text-center">
        <img
src="assets/bkpsdm.jpeg"
alt="21" style="position:
absolute; width: 100%; height:
101%; margin-left: -720px;
margin-top: -208px;">
    <center>
        <main class="form-
signin text-center card p-2"
style="width: 400px; margin-
top: 0px; margin-left: 00px;
margin-top: 200px;">
            <img
src="assets/logo.png" alt="21"
width="80px" style="margin-
left: 150px;" >
            <h1 class="h3 mb-4
fw-normal" style="font-family:
'Fira Sans', sans-
serif;">Tunjangan Kinerja
<br>Pegawai BKPSDM</h1>
            <form
action="{{route('postlogin')}}"
method="POST">
                @csrf
                <div
class="form-floating">
                    <input
type="text" class="form-
control" name="username"
required id="floatingInput"
placeholder="username">
                    <label
for="floatingInput">Username</l
abel>
                </div>
                <div
class="form-floating mt-2">
                    <input
type="password" class="form-
control" name="password"
required id="floatingPassword"
placeholder="Password">
                    <label
for="floatingPassword">Password
</label>
                </div>
                <button
type="submit" style="width:
300px;" class="btn btn-primary
mb-1 mt-1">Submit</button>
                <!-- <button
class="w-100 btn btn-lg btn-
primary mt-3" name="submit"
type="submit">Login</button> --
>
            </form>
        </main>
    </center>
</body>

</html>
```