

Listing Program

dashboard.blade.php

```
@extends('layouts.layout')

@section('content')

    <div class="row lg-12">

        <div class="col-lg-4 col-sm-12">

            <div class="card card-statistic-2">

                <div class="card-icon shadow-primary" style="background-color: #105a70">

                    <i class="fas fa-building"></i>

                </div>

                <div class="card-wrap">

                    <div class="card-header">

                        <h4>AGEN</h4>

                    </div>

                    <div class="card-body">

                        {{ $agenCount }}

                        Agen

                    </div>

                </div>

            </div>

            <div class="col-lg-4 col-sm-12">

                <div class="card card-statistic-2">

                    <div class="card-icon shadow-primary" style="background-color: #105a70">

                        <i class="fas fa-shopping-bag"></i>

                    </div>

                    <div class="card-wrap">

                        <div class="card-header">

                            <h4>PENJUALAN</h4>

                        </div>

                        <div class="card-body">

                            {{ $penjualCount }}

                            Terjual

                        </div>

                    </div>

                </div>

            </div>

            <div class="col-lg-4 col-md-4 col-sm-12">

                <div class="card card-statistic-2">


```

```

        <div class="card-icon
shadow-primary"
style="background-color: #105a70">

            <i class="fas fa-
truck"></i>

        </div>

        <div class="card-wrap">

            <div class="card-
header">

<h4>PENGIRIMAN</h4>

            </div>

            <div class="card-
body">

                {{ $pengirimanCount
}} Terkirim

            </div>

        </div>

    </div>

</div>

<div class="row">

    <div class="col-md-12">

    </div>

</div>
</div>

```

```
@endsection
```

layout.blade.php

```

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta content="width=device-
width, initial-scale=1, maximum-
scale=1, shrink-to-fit=no"
name="viewport">

    <title>Login &mdash;
Stisla</title>

    <!-- General CSS Files -->

    <link rel="stylesheet"
href="/templates/modules/bootstrap/
css/bootstrap.min.css">

    <link rel="stylesheet"
href="/templates/modules/fontaweso
me/css/all.min.css">

    <!-- CSS Libraries -->

    <link rel="stylesheet"
href="/templates/modules/bootstrap-
social/bootstrap-social.css">

    <!-- Template CSS -->

    <link rel="stylesheet"
href="/templates/css/style.css">

```

```
<link rel="stylesheet"
href="/templates/css/components.css
">
```

```
<!-- Start GA -->
```

```
<script async
src="https://www.googletagmanager.
com/gtag/js?id=UA-94034622-
3"></script>
```

```
<script>
```

```
    window.dataLayer =
window.dataLayer || [];
```

```
    function gtag() {
        dataLayer.push(arguments);
    }
```

```
    gtag('js', new Date());
```

```
    gtag('config', 'UA-94034622-
3');
```

```
</script>
```

```
<!-- /END GA -->
```

```
<style>
```

```
    .auth{
```

```
        background: radial-
gradient(circle, #9ddbd4, #37898c);
```

```
    }
```

```
</style>
```

```
</head>
```

```
<body class="auth " >
```

```
<div id="app">
```

```
<section class="section">
```

```
<div class="container mt-5">
```

```
<div class="row">
```

```
<div
```

```
        class="col-12 col-sm-
8 offset-sm-2 col-md-6 offset-md-3
col-lg-6 offset-lg-3 col-xl-4 offset-xl-
4">
```

```
        @yield('content')
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</section>
```

```
</div>
```

```
<!-- General JS Scripts -->
```

```
<script
src="/templates/modules/jquery.min.j
s"></script>
```

```
<script
src="/templates/modules/popper.js">
</script>
```

```
<script
src="/templates/modules/tooltip.js">
</script>
```

```
<script
src="/templates/modules/bootstrap/js
/bootstrap.min.js"></script>
```

```

<script
src="/templates/modules/nicescroll/j
query.nicescroll.min.js"></script>

<script
src="/templates/modules/moment.mi
n.js"></script>

<script
src="/templates/js/stisla.js"></scrip
t>

```

```

<!-- JS Libraies -->

```

```

<!-- Page Specific JS File -->

```

```

<!-- Template JS File -->

```

```

<script
src="/templates/js/scripts.js"></scrip
t>

```

```

<script
src="/templates/js/custom.js"></scri
pt>

```

```

</body>

```

```

</html>

```

login.blade.php

```

@extends('auth.layout')

```

```

@section('content')

```

```

<main class="login-form vh-100 d-
flex align-items-center justify-
content-center">

```

```

<div class="container mt-5">

```

```

<div class="row justify-content-
center">

```

```

<div class="col-md-">

```

```

<div class="card" >

```

```

<div class="text-
center">

```

```



```

```

</div>

```

```

<div class="card-
body">

```

```

<form action="{{
route('login.post') }}"
method="POST">

```

```

@csrf

```

```

<div class="form-
group row">

```

```

<label
for="email_address" class="col-md-
4 col-form-label text-md-
right">Email</label>

```

```

<div class="col-
md-8">

```

```

<input
type="text" id="email_address"
class="form-control" name="email"
required autofocus>

```

```

@if ($errors-
>has('email'))

```

```

<span
class="text-danger">{{ $errors-
>first('email') }}</span>

```

```

        @endif
    </div>
</div>

<div class="form-
group row">

    <label
for="password" class="col-md-4 col-
form-label text-md-
right">Password</label>

    <div class="col-
md-8">

        <input
type="password" id="password"
class="form-control"
name="password" required>

        @if ($errors-
>has('password'))

            <span
class="text-danger">{{ $errors-
>first('password') }}</span>

        @endif
    </div>
</div>

```

```

<div class="form-
group row">

    <div class="col-
md-6 offset-md-4">

        <div

class="checkbox">

            <label>

```

```

        <input
type="checkbox"
name="remember"> Remember Me

    </label>

</div>
</div>
</div>

```

```

<div class="form-
group row">

    <div class="col-
md-4 offset-md-4 d-flex justify-
content-center">

        <button
type="submit" class="btn btn-info
rounded-pill" style="width: 100px;
height: 40px; background-color:
#105a70">

```

```

            Login

        </button>

    </div>

</div>
</form>

```

```

</div>
</div>
</div>
</div>
</div>
</main>
@endsection

```

AgenController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Agen;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Gate;

use
Yajra\DataTables\Facades\DataTables;

class AgenController extends
Controller
{
    function __construct()
    {
        $this->middleware('permission:agen-
list|agen-create|agen -edit|agen-
delete', ['only' => ['index', 'show']]);

        $this->middleware('permission:agen-
create', ['only' => ['create', 'store']]);

        $this->middleware('permission:agen-edit',
['only' => ['edit', 'update']]);

        $this->middleware('permission:agen-
delete', ['only' => ['destroy']]);
    }
}
```

```
public function index(Request
$request)
{
    if ($request->ajax()) {
        $data = Agen::select('*');
        return DataTables::of($data)
            ->addIndexColumn()
            ->addColumn('action',
function ($row) {

                $actionBtn = '<form
method="POST" action="" .
route('agen.destroy', $row->id) . ""
style="display:inline;">'

                    . csrf_field()

                    .
method_field('DELETE');

                if (Gate::allows('agen-
delete')) {

                    $actionBtn .= '<button
type="submit" class="btn btn-danger
btn-sm" onclick="return
confirm(\'Are you sure you want to
delete this
item?\')">Delete</button>';

                }

                $actionBtn .= '</form>';

                if (Gate::allows('agen-
edit')) {

                    $actionBtn .= '<a
href="" . route('agen.edit', $row->id) .
```

```

    class="edit btn btn-success btn-sm">Edit</a>;

    }

    if (Gate::allows('agen-list')) {

        $actionBtn .= '<a href=' . route('agen.show', $row->id) . ' class="show btn btn-info btn-sm">Show</a>';

    }

    return $actionBtn;

})

->rawColumns(['action'])

->make(true);

}

return view('agen.index');

}

/**

 * Show the form for creating a new resource.

 */

public function create()

{

    return view('agen.create');

}

}

/**

```

```

 * Store a newly created resource in storage.

 */

public function store(Request $request)

{

    $request->validate([

        'nama' => 'required',

        'nohp' => 'required',

        'kode_agen' => 'required',

        'alamat' => 'required',

        'email' => 'required',

    ]);

    $input = $request->all();

    Agen::create([

        'nama'=> $input['nama'],

        'nohp' => $input['nohp'],

        'alamat' => $input['alamat'],

        'email' => $input['email'],

        'kode_agen' => $input['kode_agen']

    ]);

    return redirect()->route('agen.index');

}

/**

```

```

* Display the specified resource.
*/

public function show(Agen $agen)
{
    return view('agen.show',
compact('agen'));
}

```

```
/**
```

```

* Show the form for editing the
specified resource.

```

```

*/

public function edit(Agen $agen)
{
    return view('agen.edit',
compact('agen'));
}

```

```
/**
```

```

* Update the specified resource in
storage.

```

```

*/

public function update(Request
$request, Agen $agen)
{
    $request->validate([

```

```

        'nama' => 'required',

```

```

        'nohp' => 'required',

```

```

        'kode_agen' => 'required',

```

```

        'alamat' => 'required',
        'email' => 'required',
    ]);
    $input = $request->all();
    $agen->update($input);

    return redirect()-
>route('agen.index');
}

```

```
/**
```

```

* Remove the specified resource
from storage.

```

```

*/

public function destroy(Agen
$agen)
{
    $agen->delete();

```

```

    return redirect()-
>route('agen.index')
->with('success', 'Agen
deleted successfully');
}
}

```

AlamatController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```

use App\Models\Agen;
use App\Models\Alamat;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Gate;
use
Yajra\DataTables\Facades\DataTable
s;

```

```

class AlamatController extends
Controller

{
    function __construct()
    {
        $this-
>middleware('permission:alamat-
list|alamat-create|alamat -edit|alamat-
delete',['only' => ['index','show']]);

        $this-
>middleware('permission:alamat-
create', ['only' => ['create','store']]);

        $this-
>middleware('permission:alamat-
edit', ['only' => ['edit','update']]);

        $this-
>middleware('permission:alamat-
delete', ['only' => ['destroy']]);
    }

    public function index(Request
$request)
    {
        if ($request->ajax()) {

```

```

        $data = Alamat::select('*');
        return DataTables::of($data)
            ->addIndexColumn()
            ->addColumn('action',
function($row){
            $actionBtn = '<form
method="POST" action="" .
route('alamat.destroy', $row->id) . ""
style="display:inline;">'

                . csrf_field()

                .
method_field('DELETE');

            if (Gate::allows('alamat-
delete')) {

                $actionBtn .= '<button
type="submit" class="btn btn-danger
btn-sm" onclick="return
confirm(\'Are you sure you want to
delete this
item?\')">Delete</button>';

            }

            $actionBtn .= '</form>';

            if (Gate::allows('alamat-
edit')) {

                $actionBtn .= '<a
href="" . route('alamat.edit', $row-
>id) . "" class="edit btn btn-success
btn-sm">Edit</a>';

            }

```

```

        if (Gate::allows('alamat-
list')) {

            $actionBtn .= '<a
href=' . route('alamat.show', $row-
>id) . "' class='show btn btn-info
btn-sm'">Show</a>';

            }

```

```

        return $actionBtn;

```

```

    })-
    >addColumn('agen',function($row){

        return $row->agen-
        >nama;

    })-
    >addColumn('nohp',function($row){

        return $row->agen-
        >nohp;

    })

```

```

        ->rawColumns(['action'])
        ->make(true);

```

```

    }

    return view('alamat.index');
}

```

```

/**

```

```

    * Show the form for creating a
new resource.

```

```

    */

```

```

    public function create()

```

```

    {

        $agens = Agen::all();

        return
view('alamat.create',compact('agens')
);

    }

```

```

    /**

```

```

    * Store a newly created resource
in storage.

```

```

    */

```

```

    public function store(Request
$request)

```

```

    {

        $request->validate([

            'nama' => 'required',

            'id_agen' =>
'required|exists:agens,id',

            'kode_shipto' => 'required',

        ]);

```

```

        $input = $request->all();

```

```

        Alamat::create($input);

```

```

        return redirect()-
>route('alamat.index');

```

```

    }

```

```

    /**

```

```

    * Display the specified resource.
    */

    public function show(Alamat
$alamat)
    {
        return view('alamat.show',
compact('alamat'));
    }

/**

    * Show the form for editing the
specified resource.
    */

    public function edit($id)
{
    // Fetch the specific alamat record
    $alamat = Alamat::find($id);

    // Check if the alamat record is
found

    if (!$alamat) {
        return redirect()-
>route('alamat.index')->with('error',
'Alamat not found');
    }

    // Fetch all agens for the dropdown
    $agens = Agen::all();

```

```

    // Pass the alamat and agens data
to the view

    return view('alamat.edit',
compact('alamat', 'agens'));
}

/**

    * Update the specified resource in
storage.
    */

    public function update(Request
$request, Alamat $alamat)
    {
        $request->validate([
            'nama' => 'required',
            'id_agen' =>
'required|exists:agens,id',
            'kode_shipto' => 'required',
        ]);

        $input = $request->all();
        $alamat->update($input);

        return redirect()-
>route('alamat.index');
    }

/**

    * Remove the specified resource
from storage.

```

```

        */

        public function destroy(Alamat
$alamat)

        {

            $alamat->delete();

            return redirect()-
>route('alamat.index')

            ->with('success', 'Alamat
deleted successfully');

        }
    }
}

```

KendaraanController.php

```

<?php

namespace App\Http\Controllers;

use App\Models\Kendaraan;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Gate;
use
Yajra\DataTables\Facades\DataTable
s;

class KendaraanController extends
Controller

{

    function __construct()

```

```

    {

        $this-
>middleware('permission:kendaraan-
list|kendaraan-create|kendaraan-
edit|kendaraan-delete',['only' =>
['index','show']]);

        $this-
>middleware('permission:kendaraan-
create', ['only' => ['create','store']]);

        $this-
>middleware('permission:kendaraan-
edit', ['only' => ['edit','update']]);

        $this-
>middleware('permission:kendaraan-
delete', ['only' => ['destroy']]);

    }

    public function index(Request
$request)

    {

        if ($request->ajax()) {

            $data =
Kendaraan::select('*');

            return DataTables::of($data)

                ->addIndexColumn()

                ->addColumn('action',
function($row){

                    $actionBtn = '<form
method="POST" action="" .
route('kendaraan.destroy', $row->id)
. "" style="display:inline;">'

                        . csrf_field()

                            .

                    method_field('DELETE');

```

```

        if
(Gate::allows('kendaraan-delete')) {

            $actionBtn .= '<button
type="submit" class="btn btn-danger
btn-sm" onclick="return
confirm(\'Are you sure you want to
delete this
item?\')">Delete</button>';

        }

        $actionBtn .= '</form>';

        if
(Gate::allows('kendaraan-edit')) {

            $actionBtn .= '<a
href="" . route('kendaraan.edit', $row-
>id) . "" class="edit btn btn-success
btn-sm">Edit</a>';

        }

        if
(Gate::allows('kendaraan-list')) {

            $actionBtn .= '<a
href="" . route('kendaraan.show',
$row->id) . "" class="show btn btn-
info btn-sm">Show</a>';

        }

        return $actionBtn;

    })
->rawColumns(['action'])
->make(true);

```

```

    }

    return view('kendaraan.index');
}

/**
 * Show the form for creating a
new resource.
 */

public function create()
{
    return view('kendaraan.create');
}

/**
 * Store a newly created resource
in storage.
 */

public function store(Request
$request)
{
    $request->validate([

        'foto' =>
'required|image|mimes:jpeg,png,jpg,
gif,svg|max:2048',

        'napol' => 'required',

        'nomesin' => 'required',

        'merk' => 'required',

    ]);

    $input = $request->all();

```

```

        if ($foto = $request-
>file('foto')) {
            $destinationPath = 'foto/';

            $profileFoto =
date('YmdHis') . "." . $foto-
>getClientOriginalExtension();

            $foto-
>move($destinationPath,
$profileFoto);

            $input['foto'] =
"$profileFoto";
        }

        Kendaraan::create($input);

        return redirect()-
>route('kendaraan.index')
        ->with('success', 'Teman
created successfully.');
```

```

    }

/**
 * Display the specified resource.
 */

    public function show(Kendaraan
$kendaraan)
    {
        return view('kendaraan.show',
compact('kendaraan'));
    }

```

```

/**
 * Show the form for editing the
specified resource.
 */

    public function edit(Kendaraan
$kendaraan)
    {
        return
view('kendaraan.edit',compact('kenda
raan'));
    }

/**
 * Update the specified resource in
storage.
 */

    public function update(Request
$request, Kendaraan $kendaraan)
    {
        $request->validate([
            'foto' =>
'required|image|mimes:jpeg,png,jpg,
gif,svg|max:2048',
            'napol' => 'required',
            'nomesin' => 'required',
            'merk' => 'required',
        ]);
        $input = $request->all();

        if ($foto = $request-
>file('foto')) {

```

```

        $destinationPath = 'foto/';
    }

    $profileFoto =
date('YmdHis') . "." . $foto-
>getClientOriginalExtension();

    $foto-
>move($destinationPath,
$profileFoto);

    $input['foto'] =
"$profileFoto";
    }

    $kendaraan->update($input);

    return redirect()-
>route('kendaraan.index')
        ->with('success', 'Kendaraan
updated successfully');
    }

/**
 * Remove the specified resource
from storage.
 */
    public function destroy(Kendaraan
$kendaraan)
    {
        $kendaraan->delete();

        return redirect()-
>route('kendaraan.index')
            ->with('success', 'Kendaraan
deleted successfully');
    }

```

PegawaiController.php

```

<?php

namespace App\Http\Controllers;

use App\Models\Pegawai;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Gate;
use
Yajra\DataTables\Facades\DataTable
s;

class PegawaiController extends
Controller
{
    function __construct()
    {
        $this-
>middleware('permission:pegawai-
list|pegawai-create|pegawai -
edit|pegawai-delete',['only' =>
['index','show']]);

        $this-
>middleware('permission:pegawai-
create', ['only' => ['create','store']]);

        $this-
>middleware('permission:pegawai-
edit', ['only' => ['edit','update']]);
    }
}

```

```

$this->middleware('permission:pegawai-delete', ['only' => ['destroy']]);
}

public function index(Request $request)
{
    if ($request->ajax()) {
        $data = Pegawai::select('*');
        return DataTables::of($data)
            ->addIndexColumn()
            ->addColumn('action',
function($row){
                $actionBtn = '<form
method="POST" action="' .
route('pegawai.destroy', $row->id) .
'" style="display:inline;">'

                . csrf_field()

                .
method_field('DELETE');

                if
(Gate::allows('pegawai-delete')) {

                    $actionBtn .= '<button
type="submit" class="btn btn-danger
btn-sm" onclick="return
confirm('\Are you sure you want to
delete this
item?\'")>Delete</button>';

                }

                $actionBtn .= '</form>';

```

```

        if
(Gate::allows('pegawai-edit')) {

            $actionBtn .= '<a
href="' . route('pegawai.edit', $row-
>id) . '" class="edit btn btn-success
btn-sm">Edit</a>';

        }

        if
(Gate::allows('pegawai-list')) {

            $actionBtn .= '<a
href="' . route('pegawai.show', $row-
>id) . '" class="show btn btn-info
btn-sm">Show</a>';

        }

        return $actionBtn;

    })
    ->rawColumns(['action'])
    ->make(true);

}

return view('pegawai.index');
}

/**
 * Show the form for creating a
new resource.
 */

public function create()
{

```

```

        return view('pegawai.create');
    }

    /**
     * Store a newly created resource
     in storage.
     */

    public function store(Request
$request)
    {
        $request->validate([
            'nama' => 'required',
            'nohp' => 'required',
            'email' => 'required',
            'alamat' => 'required',
        ]);

        $input = $request->all();
        Pegawai::create($input);

        return redirect()-
>route('pegawai.index')
        ->with('success', 'Pegawai
created successfully.');
```

```

    }

    /**
     * Display the specified resource.
     */

```

```

    public function show(Pegawai
$pegawai)
    {
        return view('pegawai.show',
compact('pegawai'));
    }

    /**
     * Show the form for editing the
specified resource.
     */

    public function edit(Pegawai
$pegawai)
    {
        return
view('pegawai.edit',compact('pegawa
i'));
    }

    /**
     * Update the specified resource in
storage.
     */

    public function update(Request
$request, Pegawai $pegawai)
    {
        $request->validate([
            'nama' => 'required',
            'nohp' => 'required',
            'email' => 'required',
            'alamat' => 'required',

```

```

]);

$input = $request->all();

$pegawai->update($input);

return redirect()-
>route('pegawai.index')
->with('success', 'Pegawai
updated successfully');
}

/**
 * Remove the specified resource
from storage.
 */

public function destroy(Pegawai
$pegawai)
{
    $pegawai->delete();

    return redirect()-
>route('pegawai.index')
->with('success', 'Pegawai
deleted successfully');
}
}

```

PengirimanController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```

use App\Models\Kendaraan;
use App\Models\Pengiriman;
use App\Models\Penjualan;
use App\Models\Sopir;
use Barryvdh\DomPDF\Facade\Pdf;
use Illuminate\Http\Request;

use
Illuminate\Support\Facades\Gate;

use
Yajra\DataTables\Facades\DataTable
s;

use
Illuminate\Support\Facades\Auth;

class PengirimanController extends
Controller
{
    function __construct()
    {
        $this-
>middleware('permission:pengiriman
-list|pengiriman-create|pengiriman -
edit|pengiriman-delete',['only' =>
['index','show']]);

        $this-
>middleware('permission:pengiriman
-create', ['only' => ['create','store']]);

        $this-
>middleware('permission:pengiriman
-edit', ['only' => ['edit','update']]);

```

```

$this->middleware('permission:pengiriman-
delete', ['only' => ['destroy']]);
}

public function index(Request
$request)
{
    if ($request->ajax()) {

        $data =
        Pengiriman::select('*');

        return DataTables::of($data)
            ->addIndexColumn()
            ->addColumn('action',
            function($row){

                $actionBtn = '<form
                method="POST" action="" .
                route('pengiriman.destroy', $row-
                >id) . "" style="display:inline;">'

                . csrf_field()

                .
                method_field('DELETE');

                if
                (Gate::allows('pengiriman-delete')) {

                    $actionBtn .= '<button
                    type="submit" class="btn btn-danger
                    btn-sm" onclick="return
                    confirm(\'Are you sure you want to
                    delete this
                    item?\')">Delete</button>';

                }

                $actionBtn .= '</form>';
            }
        );
    }
}

```

```

        if
        (Gate::allows('pengiriman-edit')) {

            $actionBtn .= '<a
            href="" . route('pengiriman.edit',
            $row->id) . "" class="edit btn btn-
            success btn-sm">Edit</a>';

        }

        if
        (Gate::allows('pengiriman-list')) {

            $actionBtn .= '<a
            href="" . route('pengiriman.show',
            $row->id) . "" class="show btn btn-
            info btn-sm">Show</a>';

        }

        if(Auth::user()-
        >can('confirmation-access')){

            if($row-
            >status_laporan_pengiriman ==
            null){

                $actionBtn .= '<a
                href="" . route('laporan.confirmation',
                $row->id) . "" class="show btn btn-
                success btn-sm">Confirmation</a>';

            }

        }

        if(Gate::allows('make-
        surat-jalan')){

            if($row-
            >status_laporan_pengiriman ==
            "confirmed"){

```

```

                $actionBtn .= '<a
href="' . route('pengiriman.pdf',
$row->id) . '" class="show btn btn-
success btn-sm">Cetak Surat
Jalan</a>';

            }

        }

        return $actionBtn;

    })-
    >addColumn('penjualan',function($r
ow){

        return $row->penjualan-
        >no_transaksi;

    })-
    >addColumn('sopir',function($row){

        return $row->sopir-
        >nama;

    })-
    >addColumn('kendaraan',function($r
ow){

        return $row->kendaraan-
        >napol;

    })
    ->rawColumns(['action'])
    ->make(true);

}

return view('pengiriman.index');

}

/**

```

```

    * Show the form for creating a
    new resource.

    */

    public function create()
    {

        $penjualans = Penjualan::all();

        $sopirs = Sopir::all();

        $kendaraans = Kendaraan::all();

        return
        view('pengiriman.create',compact('pe
njualans','sopirs','kendaraans'));

    }

    public function cetak()
    {

        $data = Pengiriman::all();

        return view('pengiriman.cetak-
surat-jalan',compact('data'));

    }

    public function exportPdf($id){

        $pengiriman =
        Pengiriman::findOrFail($id);

        return view('pengiriman.cetak-
surat-jalan',compact('pengiriman'));

    }

    /**

    * Store a newly created resource
    in storage.

```

```

*/

public function store(Request
$request)
{
    $request->validate([
        'id_penjualan' => 'required',
        'id_sopir' => 'required',
        'id_kendaraan' => 'required',
        'jumlah' => 'required',
        'ijom' => 'required',
        'rit' => 'required',
        'kode_kirim' => 'required',
        'nomor_do' => 'required',
        'jam_keluar' => 'required',
        'jam_tiba' => 'required',
    ]);

    $input = $request->all();

    Pengiriman::create([
        'id_penjualan' =>
    $input['id_penjualan'],

        'id_sopir' =>
    $input['id_sopir'],

        'id_kendaraan' =>
    $input['id_kendaraan'],

        'jumlah' => $input['jumlah'],

        'ijom' => $input['ijom'],

        'rit' => $input['rit'],

        'kode_kirim' =>
    $input['kode_kirim'],

```

```

        'nomor_do' =>
    $input['nomor_do'],

        'jam_keluar' =>
    $input['jam_keluar'],

        'jam_tiba' =>
    $input['jam_tiba']
    ]);

    return redirect()-
>route('pengiriman.index');
}

public function
confirmationPengiriman($id){
    Pengiriman::where('id','=',$id)-
>update([
        'status_laporan_pengiriman'
    => 'confirmed'
    ]);

    return redirect()->back()-
>with('message','data confirmed');
}

public function kirim($id)
{
    // $pengiriman =
    Pengiriman::where('id','=',$id)-
>first();

    // return view('penjualan.kirim',
    compact('pengiriman'));

    $pengiriman =
    Pengiriman::where('id','=',$id);

    $pengirimanupdatestatus =
    $pengiriman->update([

```

```

        'status_laporan_pengiriman'
=> ' kirim'
    ]);

    if($pengirimanupdatestatus){
        $datapengiriman =
$pengiriman->first();
        dd($datapengiriman);
        return view('cetak-surat-
jalan','datapengiriman');
    }

    return redirect()->back()-
>with('message','data confirmed');
}

/**
 * Display the specified resource.
 */

public function show(Pengiriman
$pengiriman)
{
    return view('pengiriman.show',
compact('pengiriman'));
}

/**
 * Show the form for editing the
specified resource.
 */

public function edit(Pengiriman
$pengiriman)

```

```

{
    $penjualans = Penjualan::all();
    $kendaraans = Kendaraan::all();
    $sopirs = Sopir::all();

    return
view('pengiriman.edit',compact('pen
giriman','kendaraans','penjualans','so
pirs'));
}

/**
 * Update the specified resource in
storage.
 */

public function update(Request
$request, Pengiriman $pengiriman)
{
    $request->validate([
        'id_penjualan' => 'required',
        'id_sopir' => 'required',
        'id_kendaraan' => 'required',
        'jumlah' => 'required',
        'ijom' => 'required',
        'rit' => 'required',
        'kode_kirim' => 'required',
        'nomor_do' => 'required',
        'jam_keluar' => 'required',
        'jam_tiba' => 'required',
    ]);
    $input = $request->all();

```

```

        $pengiriman->update($input);

        return redirect()-
>route('pengiriman.index');
    }

    /**
     * Remove the specified resource
     from storage.
     */

    public function
    destroy(Pengiriman $pengiriman)
    {
        $pengiriman->delete();

        return redirect()-
>route('pengiriman.index')
        ->with('success', 'Pengiriman
        deleted successfully');
    }

```

```

    // function cetakPdf() {

        // $pengirimans =
        Pengiriman::all();

        // $pdf =
        PDF::loadview('pengiriman.cetakPD
        F',['pengirimans'=>$pengirimans]);

        // $pdf->setPaper('A4',
        'landscape');

        // return $pdf-
        >stream('Laporan.pdf');
    }

```

```

        // // return
        view('pengiriman.cetakPDF',['pengiri
        mans'=>$pengirimans]);

        // }

        function cetakPdf() {

            $pengirimans =
            Pengiriman::all();

            // $pdf =
            PDF::loadview('pengiriman.cetakPD
            F',['pengirimans'=>$pengirimans]);

            // $pdf->setPaper('A4',
            'landscape');

            // return $pdf-
            >stream('Laporan.pdf');

            return
            view('pengiriman.cetakPDF',['pengiri
            mans'=>$pengirimans]);

            // return
            view('pengiriman.cetakPDF',['pengiri
            mans'=>$pengirimans]);

            }

        }
    }

```

PenjualanController.php

```

<?php

namespace App\Http\Controllers;

use App\Models\Agen;
use App\Models\Alamat;
use App\Models\Kendaraan;
use App\Models\Pengiriman;

```

```

use App\Models\Penjualan;
use App\Models\Sopir;
use Illuminate\Http\Request;

use
Illuminate\Support\Facades\Gate;

use
Yajra\DataTables\Facades\DataTables;

```

```

class PenjualanController extends
Controller
{
    function __construct()
    {
        $this->middleware('permission:penjualan-
list|penjualan-create|penjualan -
edit|penjualan-delete', ['only' =>
['index', 'show']]);

        $this->middleware('permission:penjualan-
create', ['only' => ['create', 'store']]);

        $this->middleware('permission:penjualan-
edit', ['only' => ['edit', 'update']]);

        $this->middleware('permission:penjualan-
delete', ['only' => ['destroy']]);
    }

    public function index(Request
$request)
    {
        if ($request->ajax()) {

```

```

        $data = Penjualan::select('*');
        return DataTables::of($data)
            ->addIndexColumn()
            ->addColumn('action',
function ($row) {

            $actionBtn = '<form
method="POST" action="" .
route('penjualan.destroy', $row->id) .
"" style="display:inline;">'

                . csrf_field()

                .

            method_field('DELETE');

            if
(Gate::allows('penjualan-delete')) {

                $actionBtn .= '<button
type="submit" class="btn btn-danger
btn-sm" onclick="return
confirm(\'Are you sure you want to
delete this
item?\')">Delete</button>';

            }

            $actionBtn .= '</form>';

            if
(Gate::allows('penjualan-edit')) {

                $actionBtn .= '<a
href="" . route('penjualan.edit', $row-
>id) . "" class="edit btn btn-success
btn-sm">Edit</a>';

            }

```

```

        if
        (Gate::allows('penjualan-list')) {

            $actionBtn .= '<a
href="" . route('penjualan.show',
$row->id) . "" class="show btn btn-
info btn-sm">Show</a>';

        }

```

```

        return $actionBtn;

    }->addColumn('agen',
function ($row) {

    return $row->agen-
>nama;

    }->addColumn('alamat',
function ($row) {

    return $row->alamat-
>nama;

    })

    ->rawColumns(['action'])
    ->make(true);

    }

    return view('penjualan.index');

}

```

```

/**

    * Show the form for creating a
    new resource.

    */

    public function create()

    {

        $agens = Agen::all();

```

```

        $alamats = Alamat::all();

        return view('penjualan.create',
compact('agens', 'alamats'));

    }

```

```

/**

    * Store a newly created resource
    in storage.

```

```

    */

    public function store(Request
$request)

    {

```

```

        $validatedData = $request-
>validate([

            'id_agen' => 'required',

            'jumlah' => 'required',

            'id_alamat' => 'required',

            'harga' => 'required',

            'nomor_so' => 'required',

        ]);

```

```

        $jumlah = Penjualan::count();

```

```

        if ($jumlah < 10) {

            $prefix = 'NT00';

        } elseif ($jumlah < 100) {

            $prefix = 'NT0';

        } else {

            $prefix = 'NT';

```

```

    }

    $noTransaksi = $prefix .
($jumlah + 1);

Penjualan::create(array_merge($valida
tedData, ['no_transaksi' =>
$noTransaksi]));

    return redirect()-
>route('penjualan.index');
}

/**
 * Display the specified resource.
 */
public function show(Penjualan
$penjualan)
{
    return view('penjualan.show',
compact('penjualan'));
}

/**
 * Show the form for editing the
specified resource.
 */
public function edit(Penjualan
$penjualan)
{

```

```

    $alamat = Alamat::all();

    $agens = Agen::all();

    return view('penjualan.edit',
compact('penjualan','alamat','agens')
);
}

/**
 * Update the specified resource in
storage.
 */
public function update(Request
$request, Penjualan $penjualan)
{
    $request->validate([

        'id_agen' => 'required',
        'jumlah' => 'required',
        'id_alamat' => 'required',
        'harga' => 'required',
        'nomor_so' => 'required',

    ]);

    $input = $request->all();

    $penjualan->update($input);

    return redirect()-
>route('penjualan.index');
}

```

```

/**
 * Remove the specified resource
 from storage.
 */
public function destroy(Penjualan
$penjualan)
{
    $penjualan->delete();

    return redirect()-
>route('penjualan.index')
    ->with('success', 'Penjualan
deleted successfully');
}
}

```

SopirController.php

```

<?php

namespace App\Http\Controllers;

use App\Models\Sopir;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Gate;
use
Yajra\DataTables\Facades\DataTable
s;

class SopirController extends
Controller

```

```

{
    function __construct()
    {
        $this-
>middleware('permission:sopir-
list|sopir-create|sopir -edit|sopir-
delete',['only' => ['index','show']]);

        $this-
>middleware('permission:sopir-
create', ['only' => ['create','store']]);

        $this-
>middleware('permission:sopir-edit',
['only' => ['edit','update']]);

        $this-
>middleware('permission:sopir-
delete', ['only' => ['destroy']]);
    }

    public function index(Request
$request)
    {
        if ($request->ajax()) {
            $data = Sopir::select('*');

            return DataTables::of($data)
                ->addIndexColumn()
                ->addColumn('action',
function($row){
                    $actionBtn = '<form
method="POST" action="" .
route('sopir.destroy', $row->id) . "'
style="display:inline;">'

                    . csrf_field()

                    .
method_field('DELETE');

```

```

        if (Gate::allows('sopir-
delete')) {

            $actionBtn .= '<button
type="submit" class="btn btn-danger
btn-sm" onclick="return
confirm(\'Are you sure you want to
delete this
item?\')">Delete</button>';

            }

            $actionBtn .= '</form>';

            if (Gate::allows('sopir-
edit')) {

                $actionBtn .= '<a
href="" . route('sopir.edit', $row->id) .
"" class="edit btn btn-success btn-
sm">Edit</a>';

                }

                if (Gate::allows('sopir-
list')) {

                    $actionBtn .= '<a
href="" . route('sopir.show', $row-
>id) . "" class="show btn btn-info
btn-sm">Show</a>';

                    }

                    return $actionBtn;

                })

                ->rawColumns(['action'])

                ->make(true);

            }

            return view('sopir.index');

        }

        /**

        * Show the form for creating a
new resource.

        */

        public function create()

        {

            return view('sopir.create');

        }

        /**

        * Store a newly created resource
in storage.

        */

        public function store(Request
$request)

        {

            $request->validate([

                'nama' => 'required',

                'nohp' => 'required',

                'alamat' => 'required',

                'nomor_do' => 'required',

            ]);

            $input = $request->all();

            Sopir::create($input);

```

```

        return redirect()-
>route('sopir.index');
    }

/**
 * Display the specified resource.
 */
public function show(Sopir $sopir)
{
    return view('sopir.show',
compact('sopir'));
}

/**
 * Show the form for editing the
specified resource.
 */
public function edit(Sopir $sopir)
{
    return
view('sopir.edit',compact('sopir'));
}

/**
 * Update the specified resource in
storage.
 */
public function update(Request
$request, Sopir $sopir)

```

```

{
    $request->validate([
        'nama' => 'required',
        'nohp' => 'required',
        'alamat' => 'required',
        'nomor_do' => 'required',
    ]);
    $input = $request->all();
    $sopir->update($input);

    return redirect()-
>route('sopir.index')
        ->with('success', 'Sopir
updated successfully');
}

/**
 * Remove the specified resource
from storage.
 */
public function destroy(Sopir
$sopir)
{
    $sopir->delete();

    return redirect()-
>route('sopir.index')
        ->with('success', 'Sopir deleted
successfully');
}

```

<pre> } RoleController.php <?php namespace App\Http\Controllers; use Illuminate\Http\Request; use App\Http\Controllers\Controller; use Illuminate\Support\Facades\DB; use Spatie\Permission\Models\Role; use Spatie\Permission\Models\Permission; class RoleController extends Controller { /** * Display a listing of the resource. * * @return \Illuminate\Http\Response */ function __construct() { \$this->middleware('permission:role-list role- create role-edit role-delete', </pre>	<pre> ['only' => ['index', 'store']]); \$this->middleware('permission:role- create', ['only' => ['create', 'store']]); \$this->middleware('permission:role-edit', ['only' => ['edit', 'update']]); \$this->middleware('permission:role- delete', ['only' => ['destroy']]); } /** * Display a listing of the resource. * * @return \Illuminate\Http\Response */ public function index(Request \$request) { \$roles = Role::orderBy('id', 'DESC')->paginate(5); return view('roles.index', compact('roles')) ->with('i', (\$request-> input('page', 1) - 1) * 5); } /** * Show the form for creating a new resource. * </pre>
--	--

```

    * @return
    \Illuminate\Http\Response

    */

    public function create()
    {
        $permission =
        Permission::get();

        return view('roles.create',
        compact('permission'));
    }
    /**
     * Store a newly created resource
     in storage.
     *
     * @param
     \Illuminate\Http\Request $request
     * @return
     \Illuminate\Http\Response
     */

    public function store(Request
    $request)
    {
        $this->validate($request, [
            'name' =>
            'required|unique:roles,name',
            'permission' => 'required',
        ]);

        $role = Role::create(['name' =>
        $request->input('name')]);

        $role-
        >syncPermissions($request-
        >input('permission'));

```

```

        return redirect()-
        >route('roles.index')

        ->with('success', 'Role
        created successfully');
    }
    /**
     * Display the specified resource.
     *
     * @param int $id
     * @return
     \Illuminate\Http\Response
     */

    public function show($id)
    {
        $role = Role::find($id);
        $rolePermissions =

        Permission::join("role_has_permissi
        ons",
        "role_has_permissions.permission_id
        ", "=", "permissions.id")

        -
        >where("role_has_permissions.role_i
        d", $id)

        ->get();

        return view('roles.show',
        compact('role', 'rolePermissions'));
    }
    /**
     * Show the form for editing the
     specified resource.
     *

```

```

    * @param int $id
    * @return
    \Illuminate\Http\Response

    */

    public function edit($id)
    {
        $role = Role::find($id);

        $permission =
        Permission::get();

        $rolePermissions =
        DB::table("role_has_permissions")-
        >where("role_has_permissions.role_id", $id)
        -
        >pluck('role_has_permissions.permission_id',
        'role_has_permissions.permission_id'
        )

        ->all();

        return view('roles.edit',
        compact('role', 'permission',
        'rolePermissions'));
    }

    /**

    * Update the specified resource in
    storage.

    *

    * @param
    \Illuminate\Http\Request $request

    * @param int $id

    * @return
    \Illuminate\Http\Response

    */

```

```

    public function update(Request
    $request, $id)

    {

        $this->validate($request, [

            'name' => 'required',

            'permission' => 'required',

        ]);

        $role = Role::find($id);

        $role->name = $request-
        >input('name');

        $role->save();

        $role-
        >syncPermissions($request-
        >input('permission'));

        return redirect()-
        >route('roles.index')->with('success',
        'Role updated successfully');

    }

    /**

    * Remove the specified resource
    from storage.

    *

    * @param int $id

    * @return
    \Illuminate\Http\Response

    */

    public function destroy($id)

    {

        DB::table("roles")->where('id',
        $id)->delete();
    }

```

```

        return redirect()-
>route('roles.index')->with('success',
'Role deleted successfully');

    }
}

```

UserController.php

```
<?php
```

```

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\User;
use Illuminate\Support\Arr;
use Illuminate\Support\Facades\DB;
use
Illuminate\Support\Facades\Hash;
use Spatie\Permission\Models\Role;

class UserController extends
Controller
{
    function __construct()
    {
        $this-
>middleware('permission:user-
list|user-create| user -edit| user-
delete', ['only' => ['index', 'show']]);

        $this-
>middleware('permission:user-
create', ['only' => ['create', 'store']]);

```

```

        $this-
>middleware('permission:user-edit',
['only' => ['edit', 'update']]);

        $this-
>middleware('permission:user-
delete', ['only' => ['destroy']]);

    }

    //

    public function index(Request
$request)
    {
        $data = User::orderBy('id',
'DESC')->paginate(5);

        return view('users.index',
compact('data'))->with('i', ($request-
>input('page', 1) - 1) * 5);
    }

    /**

    * Show the form for creating a
new resource.

    *

    * @return
\Illuminate\Http\Response

    */

    public function create()
    {
        $roles = Role::pluck('name',
'name')->all();

        return view('users.create',
compact('roles'));
    }

    /**

```

```

    * Store a newly created resource
    in storage.
    *

    * @param
    \Illuminate\Http\Request $request

    * @return
    \Illuminate\Http\Response

    */

    public function store(Request
    $request)

    {

        $this->validate($request, [

            'name' => 'required',

            'email' =>
            'required|email|unique:users,email',

            'password' =>
            'required|same:confirm-password',

            'roles' => 'required'

        ]);

        $input = $request->all();

        $input['password'] =
        Hash::make($input['password']);

        $user = User::create($input);

        $user->assignRole($request-
        >input('roles'));

        return redirect()-
        >route('users.index')->with('success',
        'User created successfully');

    }

    /**

    * Display the specified resource.

    *

```

```

    * @param int $id

    * @return
    \Illuminate\Http\Response

    */

    public function show($id)

    {

        $user = User::find($id);

        return view('users.show',
        compact('user'));

    }

    /**

    * Show the form for editing the
    specified resource.

    *

    * @param int $id

    * @return
    \Illuminate\Http\Response

    */

    public function edit($id)

    {

        $user = User::find($id);

        $roles = Role::pluck('name',
        'name')->all();

        $userRole = $user->roles-
        >pluck('name', 'name')->all();

        return view('users.edit',
        compact('user', 'roles', 'userRole'));

    }

    /**

    * Update the specified resource in
    storage.

```

```

*
* @param
\Illuminate\Http\Request $request
* @param int $id
* @return
\Illuminate\Http\Response
*/

public function update(Request
$request, $id)
{
    $this->validate($request, [
        'name' => 'required',
        'email' =>
'required|email|unique:users,email,' .
$id,
        'password' => 'same:confirm-
password',
        'roles' => 'required'
    ]);
    $input = $request->all();
    if (!empty($input['password']))
    {
        $input['password'] =
Hash::make($input['password']);
    } else {
        $input = Arr::except($input,
array('password'));
    }
    $user = User::find($id);
    $user->update($input);
    DB::table('model_has_roles')-
>where('model_id', $id)->delete();

```

```

$user->assignRole($request-
>input('roles'));

return redirect()-
>route('users.index')->with('success',
'User updated successfully');
}
/**

* Remove the specified resource
from storage.

*
* @param int $id
* @return
\Illuminate\Http\Response
*/

public function destroy($id)
{
    User::find($id)->delete();

    return redirect()-
>route('users.index')->with('success',
'User deleted successfully');
}
}

<?php

namespace App\Http\Controllers;

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

```

```
use
Illuminate\Support\Facades\Hash;

use Spatie\Permission\Models\Role;
```

```
class UserController extends
Controller

{

    function __construct()

    {

        $this-
>middleware('permission:user-
list|user-create| user -edit| user-
delete', ['only' => ['index', 'show']]);

        $this-
>middleware('permission:user-
create', ['only' => ['create', 'store']]);

        $this-
>middleware('permission:user-edit',
['only' => ['edit', 'update']]);

        $this-
>middleware('permission:user-
delete', ['only' => ['destroy']]);

    }

    //

    public function index(Request
$request)

    {

        $data = User::orderBy('id',
'DESC')->paginate(5);

        return view('users.index',
compact('data'))->with('i', ($request-
>input('page', 1) - 1) * 5);

    }
```

```
/**

    * Show the form for creating a
    new resource.

    *

    * @return
    \Illuminate\Http\Response

    */

    public function create()

    {

        $roles = Role::pluck('name',
'name')->all();

        return view('users.create',
compact('roles'));

    }

    /**

    * Store a newly created resource
    in storage.

    *

    * @param
    \Illuminate\Http\Request $request

    * @return
    \Illuminate\Http\Response

    */

    public function store(Request
$request)

    {

        $this->validate($request, [

            'name' => 'required',

            'email' =>
            'required|email|unique:users,email',

            'password' =>
            'required|same:confirm-password',
```

```

        'roles' => 'required'
    ]);

    $input = $request->all();

    $input['password'] =
    Hash::make($input['password']);

    $user = User::create($input);

    $user->assignRole($request-
    >input('roles'));

    return redirect()-
    >route('users.index')->with('success',
    'User created successfully');
}

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return
 \Illuminate\Http\Response
 */

public function show($id)
{
    $user = User::find($id);

    return view('users.show',
    compact('user'));
}

/**
 * Show the form for editing the
 specified resource.
 *
 * @param int $id

```

```

 * @return
 \Illuminate\Http\Response
 */

public function edit($id)
{
    $user = User::find($id);

    $roles = Role::pluck('name',
    'name')->all();

    $userRole = $user->roles-
    >pluck('name', 'name')->all();

    return view('users.edit',
    compact('user', 'roles', 'userRole'));
}

/**
 * Update the specified resource in
 storage.
 *
 * @param
 \Illuminate\Http\Request $request
 * @param int $id
 * @return
 \Illuminate\Http\Response
 */

public function update(Request
$request, $id)
{
    $this->validate($request, [

        'name' => 'required',

        'email' =>
        'required|email|unique:users,email,' .
        $id,

```

```

        'password' => 'same:confirm-
password',
        'roles' => 'required'
    ]);
    $input = $request->all();
    if (!empty($input['password']))
    {
        $input['password'] =
Hash::make($input['password']);
    } else {
        $input = Arr::except($input,
array('password'));
    }
    $user = User::find($id);
    $user->update($input);
    DB::table('model_has_roles')-
>where('model_id', $id)->delete();
    $user->assignRole($request-
>input('roles'));
    return redirect()-
>route('users.index')->with('success',
'User updated successfully');
}
/**
 * Remove the specified resource
from storage.
 *
 * @param int $id
 * @return
\Illuminate\Http\Response
 */
public function destroy($id)
{
    User::find($id)->delete();
    return redirect()-
>route('users.index')->with('success',
'User deleted successfully');
}
}

```



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA**

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414

Laman : <http://polsri.ac.id>, Pos El : info@polsri.ac.id

KESEPAKATAN BIMBINGAN LAPORAN AKHIR (LA)

Kami yang bertanda tangan di bawah ini,

Pihak Pertama

Nama : Putri Thessa
NPM : 062130800229
Jurusan : Manajemen Informatika
Program Studi : DIII Manajemen Informatika

Pihak Kedua

Nama : Rika Sadariawati, S.E., M.Si.
NIP : 197302232002122001
Jurusan : Manajemen Informatika
Program Studi : DIII Manajemen Informatika

Pada hari ini Selasa tanggal 19 Maret 2024 telah sepakat untuk melakukan konsultasi bimbingan Laporan Akhir. Konsultasi bimbingan sekurang-kurangnya 2 (Dua) kali dalam satu minggu. Pelaksanaan bimbingan pada setiap hari Senin - Sabtu di Politeknik Negeri Sriwijaya.

Demikianlah kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Laporan Akhir.

Pihak Pertama,

**Putri Thessa
NIM 062130800229**

Palembang, 19 Maret 2024

Pihak Kedua,

**Rika Sadariawati, S.E., M.Si.
NIP. 197302232002122001**

Mengetahui,
Ketua Jurusan

**Dr. Indri Ariyanti, SE., M.Si.
NIP. 197306032008012008**



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA
JURUSAN MANAJEMEN INFORMATIKA

Jalan Sriwijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id>, Pos El : info@polsri.ac.id

KESEPAKATAN BIMBINGAN LAPORAN AKHIR (LA)

Kami yang bertanda tangan di bawah ini,

Pihak Pertama

Nama : Putri Thessa
NIM : 062130800229
Jurusan : Manajemen Informatika
Program Studi : DIII Manajemen Informatika

Pihak Kedua

Nama : Robinson, S.Kom., M.Kom.
NIP : 197503172002121003
Jurusan : Manajemen Informatika
Program Studi : DIII Manajemen Informatika

Pada hari ini Kamis tanggal 21 Maret 2024 telah sepakat untuk melakukan konsultasi bimbingan Laporan Akhir. Konsultasi bimbingan sekurang-kurangnya minimal 2 (dua) kali dalam satu minggu, tempat di Politeknik Negeri Sriwijaya.

Demikianlah kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Laporan Akhir.

Pihak Pertama,

Putri Thessa
NIM. 062130800229

Palembang, 21 Maret 2024

Pihak Kedua,

Robinson, S.Kom., M.Kom.
NIP. 197503172002121003

Mengetahui,
Ketua Jurusan Manajemen Informatika

Dr. Indri Ariyanti, S.E., M.Si.
NIP. 197306032008012008





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA
JURUSAN MANAJEMEN INFORMATIKA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id>, Pos El : info@polsri.ac.id

PENGAJUAN JUDUL LAPORAN AKHIR (LA)

Kepada Yth,

Dosen Pembimbing I

Rika Sadariawati, S.E., M.Si.

Politeknik Negeri Sriwijaya

Dengan Hormat,

Dalam rangka memenuhi syarat penyusunan Laporan Akhir, maka saya:

Nama : Putri Thessa

NIM : 062130800229

Jurusan/Prodi : Manajemen Informatika/DIII Manajemen Informatika

Bermaksud mengajukan judul Laporan Akhir. Adapun judul Laporan Akhir yang saya ajukan, yaitu:

- (1.) Aplikasi Pengelolaan Data Kendaraan Logistik Dan Penjualan Gas LPG Pada PT. Sarana Agung Gas
2. Sistem Manajemen Logistik Harian Agen Masuk Ke SPPBE PT. Sarana Agung Gas Berbasis Website

Demikian surat pengajuan judul Laporan Akhir ini dibuat, sekiranya judul di atas dapat disetujui. Atas persetujuan Ibu saya ucapkan terima kasih.

Menyetujui,

Dosen Pembimbing I

Rika Sadariawati, S.E., M.Si.
NIP. 197302232002122001

Palembang, 20 Maret 2024

Hormat Saya,
Mahasiswa

Putri Thessa
NIM. 062130800229





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA
JURUSAN MANAJEMEN INFORMATIKA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id>, Pos El : info@polsri.ac.id

PENGAJUAN JUDUL LAPORAN AKHIR (LA)

Kepada Yth,
Dosen Pembimbing II
Robinson, S.Kom., M.Kom.
Politeknik Negeri Sriwijaya

Dengan Hormat,

Dalam rangka memenuhi syarat penyusunan Laporan Akhir, maka saya:

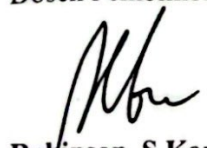
Nama : Putri Thessa
NIM : 062130800229
Jurusan/Prodi : Manajemen Informatika/DIII Manajemen Informatika

Bermaksud mengajukan judul Laporan Akhir. Adapun judul Laporan Akhir yang saya ajukan,
yaitu:

1. Aplikasi Pengelolaan Data Kendaraan Logistik Dan Penjualan Gas LPG Pada PT. Sarana Agung Gas
2. Sistem Manajemen Logistik Harian Agen Masuk Ke SPPBE PT. Sarana Agung Gas Berbasis Website

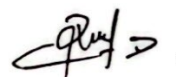
Demikian surat pengajuan judul Laporan Akhir ini dibuat, sekiranya judul di atas dapat
disetujui. Atas persetujuan Bapak saya ucapkan terima kasih.

Menyetujui,
Dosen Pembimbing II


Robinson, S.Kom., M.Kom.
NIP. 197503172002121003

Palembang, 20 Maret 2024

Hormat Saya,
Mahasiswa



Putri Thessa
NIM. 062130800229





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA
JURUSAN MANAJEMEN INFORMATIKA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id> Pos El : info@polsri.ac.id

LEMBAR PENGESAHAN JUDUL LAPORAN AKHIR (LA)

Nama : Putri Thessa
NIM : 062130800229
Jurusan / Program Studi : Manajemen Informatika/ DIII Manajemen Informatika
Judul Laporan Akhir : Aplikasi Pengelolaan Kendaraan Logistik Dan Penjualan
Gas LPG Pada PT. Sarana Agung Gas

Palembang, 21 Maret 2024

Tim Pembimbing:

Pembimbing I

Rika Sadariawati, S.E., M.Si.
NIP 197302232002122001

Pembimbing II

Robinson, S.Kom., M.Kom.
NIP 197503172002121003

Mengetahui,
Ketua Jurusan Manajemen Informatika

Dr. Indri Ariyanti, SE., M.Si
NIP 197306032008012008





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA
JURUSAN MANAJEMEN INFORMATIKA

Jalan Sriwijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414

Laman : <http://polsri.ac.id>, Pos El : info@polsri.ac.id

PERMOHONAN SURAT PENGANTAR PENGAMBILAN DATA

Hal : Permohonan Surat Pengantar
Pengambilan Data

Palembang, 21 Maret 2024

Kepada Yth.
Wakil Direktur I
Politeknik Negeri Sriwijaya
Di
Tempat

Dengan Hormat,

Sehubungan dengan adanya penulisan Laporan Akhir yang merupakan salah satu mata kuliah Semester VI dan sebagai syarat untuk menyelesaikan pendidikan Diploma III Jurusan Manajemen Informatika Politeknik Negeri Sriwijaya, untuk itu saya mohon kepada Bapak agar dibuatkan surat pengantar yang ditunjukkan kepada :

Yth. Direktur Utama
PT. Sarana Agung Gas
Jl. Lintas Sumatra, Celikah, Kayuagung,
Kabupaten Ogan Komering Ilir, Sumatera Selatan 30867

Untuk pengambilan data di PT. Sarana Agung Gas dalam pembuatan aplikasi yang berjudul "Aplikasi Pengelolaan Data Kendaraan Logistik Dan Penjualan Gas LPG Pada PT. Sarana Agung Gas".

Pengambilan data ini akan saya mulai pada tanggal 1 April 2024 s.d. 30 Juli 2024. Saya mahasiswa yang akan melakukan pengambilan data adalah :

No	NAMA	NIM	KELAS	JURUSAN
1.	Putri Thessa	062130800229	6 IB	Manajemen Informatika

Demikianlah surat permohonan ini saya buat, atas bantuan dan kerjasama Bapak saya ucapkan terima kasih.

Mengetahui,
Ketua Jurusan Manajemen Informatika

Hormat saya,


Dr. Indri Ariyanti, SE., M.Si
NIP 197306032008012008


Putri Thessa
062130800229



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara Bukit Besar – Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id>, Pos El : info@polsri.ac.id

Nomor : 3109/PL6.3.1/SP/2024
Perihal : Izin Pengambilan Data

25 Maret 2024

Yth. Direktur Utama
PT. Sarana Agung Gas
Jalan Lintas Sumatera Celikah Kayuagung
Kabupaten Ogan Komering Ilir Sumatera Selatan 30867

Dengan hormat,

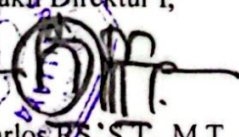
Sesuai dengan kurikulum Jurusan Manajemen Informatika pada Politeknik Negeri Sriwijaya, Laporan Akhir merupakan mata kuliah wajib pada semester 6 (enam). Untuk itu mahasiswa kami memerlukan data untuk penyusunan Laporan Akhir tersebut.

Sehubungan dengan hal tersebut di atas, kami mohon Bapak/Ibu dapat memberikan izin dan membantu mahasiswa kami ini untuk pengambilan data.

Mahasiswa kami yang akan mengumpulkan data tersebut adalah :

No	Nama	NPM	Kelas	Jurusan/ Program Studi
1	Putri Thessa	0621 3080 0229	6 IB	Manajemen Informatika

Atas perhatian dan bantuannya diucapkan terima kasih.


Plt. Direktur
Wakil Direktur I,

Carlos RS, S.T., M.T.
NIP 196403011989031003

Tembusan:

1. Plt. Direktur
2. Ketua Jurusan Manajemen Informatika
3. Yang bersangkutan
4. Arsip

MS.Word/E/AD/Dw





PT SARANA AGUNG GAS

Jl. Lintas Sumatra, Celikah, Kayuagung, Kabupaten Ogan
Komereng Ilir - Sumatera Selatan. No. Telp: 0813-6704-4605

Email : spbesaranaagunggas@gmail.com

Kayuagung, 09 April 2024

Perihal : Izin Pengambilan Data

**Kepada Yth,
Direktur Politeknik Negeri Sriwijaya
JL. Srijaya Negara Bukit Besar
Palembang**

Dengan Hormat,

Sehubungan dengan surat Nomor: 3109/PL.3.1/SP/2024 tanggal 25 Maret 2024, perihal izin pengambilan data perusahaan guna penyusunan Laporan Akhir, maka dengan ini kami bersedia memberikan izin pengambilan data kepada mahasiswa dibawah ini:

No.	Nama	NIM	Kelas	Program Studi/Jurusan
1.	Putri Thessa	062130800229	6IB	Manajemen Informatika

Demikianlah surat ini kami buat untuk dipergunakan sebagaimana mestinya, atas perhatiannya kami ucapkan terima kasih.

Hormat kami,
Manager

M. Fathoni



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA
JURUSAN MANAJEMEN INFORMATIKA

Jalan Sriwijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman: <http://polstri.ac.id> Pos El: info@polstri.ac.id

LEMBAR BIMBINGAN LAPORAN AKHIR

Lembar : 1

Nama : Putri Thessa
Nim : 062130800229
Jurusan / Program Studi : Manajemen Informatika / DIII Manajemen Informatika
Judul : Aplikasi Pengelolaan Data Kendaraan Logistik dan Penjualan Gas LPG Pada PT Sarana Agung Gas
Pembimbing : Rika Sadariawati, S.E., M.Si.

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
1.	19/3 - 2024	Pengantar judul	
2.	20/3 - 2024	Ace judul	
3.	27/3 - 2024	Perbaiki Bab I sesuai catatan	
4.	27/3 - 2024	Ace Bab I	
5.	17/4 - 2024	Perbaiki Bab II sesuai catatan	
6.	22/4 - 2024	ACC BAB II	
7.	03/5 - 2024	ACC BAB III	
8.	14/6 - 2024	ACC BAB IV	





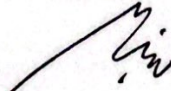
KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA
JURUSAN MANAJEMEN INFORMATIKA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414

Laman : <http://polsri.ac.id>, Pos El : info@polsri.ac.id

Lembar : 2

9.	20/6 - 2024	ACC BAB V	
10.	25/6 - 2024	Rekomendasi Skripsi	
11.			
12.			
13.			
14.			
15.			

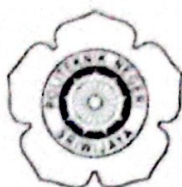
Palembang,

2024

Ketua Jurusan

Manajemen Informatika


Dr. Indri Ariyanti, SE., M.Si.
NIP 197306032008012008



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA
JURUSAN MANAJEMEN INFORMATIKA

Jalan Sriwijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polisri.ac.id> Pos El : info@polisri.ac.id

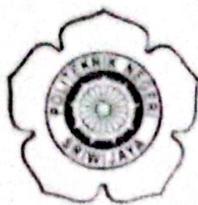
LEMBAR BIMBINGAN LAPORAN AKHIR

Lembar : 1

Nama : Putri Thessa
Nim : 062130800229
Jurusan / Program Studi : Manajemen Informatika / DIII Manajemen Informatika
Judul : Aplikasi Pengelolaan Data Kendaraan Logistik dan Penjualan Gas LPG Pada PT Sarana Agung Gas
Pembimbing : Robinson, S.Kom., M.Kom.

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
1.	20/3 - 2024	ACC JUDUL	
2.	01/4 - 2024	REVISI BAB I	
3.	22/4 - 2024	ACC BAB I	
4.	22/4 - 2024	REVISI BAB II	
5.	29/4 - 2024	ACC BAB II	
6.	29/4 - 2024	REVISI BAB III	
7.	13/5 - 2024	ACC BAB IV	
8.	19/6 - 2024	REVISI BAB IV	





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA
JURUSAN MANAJEMEN INFORMATIKA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id>, Pos El : info@polsri.ac.id

Lembar : 2

9.		ACC BAB IV	
10.		ACC BAB V	
11.		1. KUT + kesimpulan. laporan Akhir.	
12.			
13.			
14.			
15.			

Palembang,

2024

Ketua Jurusan

Manajemen Informatika

Dr. Indri Ariyanti, SE., M.Si.
NIP 197306032008012008





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414

Laman : <http://polsri.ac.id>, Pos El : info@polsri.ac.id

REKOMENDASI UJIAN LAPORAN AKHIR (LA)

Pembimbing Laporan Akhir memberikan rekomendasi kepada,

Nama : Putri Thessa
NPM : 062130800229
Jurusan/Program Studi : DIII-Manajemen Informatika
Judul Laporan Akhir : Aplikasi Pengelolaan Data Kendaraan Logistik Dan Penjualan
Gas LPG Pada PT Sarana Agung Gas

Mahasiswa tersebut telah memenuhi persyaratan dan dapat mengikuti Ujian Laporan Akhir (LA)
pada Tahun Akademik 2024

Pembimbing I,

Rika Sadariawati, SE., M.Si.
NIP197302232002122001

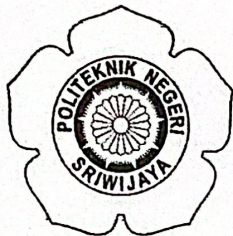
Palembang, 24 Juni 2024

Pembimbing II,

Robinson, S.Kom., M.Kom.
NIP 197503172002121003

Mengetahui,
Ketua Jurusan Manajemen Informatika

Dr. Indri Ariyanti, SE., M.Si.
NIP 197306032008012008



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414

Laman : <http://polsri.ac.id> Pos El : info@polsri.ac.id

LEMBAR PELAKSANAAN REVISI LAPORAN AKHIR

Mahasiswa berikut,

Nama : Putri Thessa
NIM : 062130800229
Jurusan/Program Studi : Manajemen Informatika/DIII Manajemen Informatika
Judul Laporan Akhir : Aplikasi Pengelolaan Data Kendaraan Logistik dan Penjualan Gas LPG Pada PT Sarana Agung Gas

Telah melaksanakan revisi terhadap Laporan Akhir yang diujikan pada hari Senin tanggal 15 bulan Juli tahun 2024 Pelaksanaan revisi terhadap Laporan Akhir tersebut telah disetujui oleh Dosen Penguji yang memberikan revisi :

No.	Komentar	Nama Dosen Penguji *)	Tanggal	Tanda Tangan
1.	1. Bab I 2. Use Case (4 Aktor) 3. Class Diagram 4. Kesimpulan	Indra Satriadi, S.T.,M.Kom	24/7-2024	
2.	1. Latar Belakang Tambah jurnal tahun 2020 2. Wawancara dibuat tanggal pelaksanaan 3. Use Case Diagram	Denny Alfian, S.Kom.,M.kom	9/8-2024	
3.	1. Tambahkan Data DO Pada Aplikasi Sopir	Yusniarti, S.Kom.,M.Kom	31/7/2024	

Palembang, Juli 2024

Ketua Penguji **),

Indra Satriadi, S.T.,M.Kom
NIP 197211162000031002

Catatan:

- *) 1. Dosen penguji yang memberikan revisi saat ujian Laporan Akhir;
**) 2. Dosen penguji yang ditugaskan sebagai Ketua Penguji saat ujian LA;
3. Lembaran pelaksanaan revisi ini harus dilampirkan dalam Laporan Akhir;



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id>, Pos El : info@polsri.ac.id

REVISI UJIAN LAPORAN AKHIR (LA)

Ruang : 6
Dosen Penguji : Indra Sutnadi.
Nama Mahasiswa : Putri Thessa (062130800229)
Jurusan/Program Studi : Manajemen Informatika / DIII Manajemen Informatika
Judul Laporan Akhir : Aplikasi Pengelolaan Data Kendaraan Logistik dan Penjualan
Gas LPG Pada PT Sarana Agung Gas

No	Uraian Revisi	Paraf
1.	Uraian Bab I.	
2.	Use Case → 4 aktor : Agen,	
3.	Class Diagram.	
4.	Kesimpulan	

Palembang, Juli 2024
Dosen Penguji,

(...Indra Sutnadi...)





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id>, Pos El : info@polsri.ac.id

REVISI UJIAN LAPORAN AKHIR (LA)

Ruang :
Dosen Penguji :
Nama Mahasiswa : Putri Thessa (062130800229)
Jurusan/Program Studi : Manajemen Informatika / DIII Manajemen Informatika
Judul Laporan Akhir : Aplikasi Pengelolaan Data Kendaraan Logistik dan Penjualan
Gas LPG Pada PT Sarana Agung Gas

No	Uraian Revisi	Paraf
	Tambahkan data Ds pd aplikasi sopir	

Palembang, Juli 2024
Dosen Penguji,

(.....)





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id>, Pos El : info@polsri.ac.id

REVISI UJIAN LAPORAN AKHIR (LA)

Ruang : 06
Dosen Penguji : DENNY SRAN
Nama Mahasiswa : Putri Thessa (062130800229)
Jurusan/Program Studi : Manajemen Informatika / DIII Manajemen Informatika
Judul Laporan Akhir : Aplikasi Pengelolaan Data Kendaraan Logistik dan Penjualan
Gas LPG Pada PT Sarana Agung Gas

No	Uraian Revisi	Paraf
1.	Cover belasan tambah final tahun 2020 >	
2.	Wawancara dibuat tangan plesman	
3.	Uji Case diadakan!	

Palembang, Juli 2024
Dosen Penguji,


(Denny Sran)

