

LISTRING PROGRAM

PVLController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Instansi;
use App\Models\Pegawai;
use App\Models\Pelapor;
use App\Models\PVL;
use App\Models\Subinstansi;
use App\Models\TipeLaporan;
use Carbon\Carbon;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Log;

class PVLController extends Controller
{
    function __construct()
    {
        $this->middleware('permission:pvl-list|pvl-create|pvl-edit|pvl-delete', ['only' => ['index', 'show']]);
        $this->middleware('permission:pvl-create', ['only' => ['create', 'store']]);
        $this->middleware('permission:pvl-edit', ['only' => ['edit', 'update']]);
        $this->middleware('permission:pvl-delete', ['only' => ['destroy']]);
    }

    public function index_old()
    {
        $pvls = PVL::latest()->paginate(5);
        return view('pvls.index', compact('pvls'))
            ->with('i', (request()->input('page', 1) - 1) * 5);
    }

    public function index(Request $request)
    {
        if ($request->ajax()) {
            $query_data = new PVL();
            if ($request->sSearch) {
                $search_value = '%' . $request->sSearch . '%';
                $query_data = $query_data-
                >where(function ($query) use ($search_value) {
                    $query->where('pvls.perihal', 'like', $search_value)

```

```

                ->orWhere('pvls.no_agenda', 'like', $search_value)
            });
        }
        $data = $query_data-
        >orderBy('id')->get();
        return datatables()->of($data)-
        >addIndexColumn()->addColumn('action', function ($row) {
            $btn = '';
            $btn .= '
                <form action="" .
                route('pvl.destroy', $row->id) . ""
                method="POST">
                    <a class="btn btn-info"
                    href="" . route('pvl.show', $row->id) . ""
                    style="margin-right: 5px;">Show</a>;
                    if ($row-
                    >tanggal_diserahkan_pvl_kaper == null) {
                        $btn .= '<a class="btn btn-info" href="" . route('sendtokaper', $row-
                        >id) . "" style="margin-right: 5px;">Send
                        To Kaper</a>;
                    }

                    if (Auth::user()->can('pvl-edit'))
                    {
                        $btn .= '<a class="btn btn-primary" href="" . route('pvl.edit', $row-
                        >id) . "">Edit</a>;
                    }

                    if (Auth::user()->can('pvl-delete')) {
                        $btn .= csrf_field() .
                        method_field('DELETE') . '
                        <button type="submit"
                        class="btn btn-danger">Delete</button>;
                    }

                    $btn .= '
                </form>
            ';

            return $btn;
        })->addColumn('TipeLaporan', function ($row) {
            return $row->TipeLaporan-
            >nama;

```

```

        })->addColumn('pegawai', function
($row) {
            return $row->pegawai->nama;
        })->addColumn('pelapor', function
($row) {
            return $row->pelapor->nama;
        })->addColumn('instansi', function
($row) {
            return $row->instansi->nama;
        })->addColumn('subinstansi',
function ($row) {
            return $row->subinstansi;
        })->addColumn('asisten_riksa',
function ($row) {
            return $row->asisteriksa;
        })
        ->rawColumns(['action'])-
>make(true);
    }
    return view('pvls.index');
}

public function sendtokaper($id)
{
    $data = PVL::findOrFail($id);
    if ($data) {
        $data->update([
            'status' => "send",

'tanggal_diserahkan_pvl_kaper' => now()
        ]);
    }

    return redirect()->back()-
>with('message', 'success send to kaper');
}

public function
tgl_diserahkan_riksa($id)
{
    $data = PVL::findOrFail($id);
    $data->update([
        'status' => "send",
        'tgl_diserahkan_riksa' => now()
    ]);

    return redirect()-
>route('rekap_laporan_page');
}

public function recap_laporan_page()
{
    $jumlah = 0;

```

```

    $data = [];
    return view('pvls.rekap',
compact('jumlah', 'data'));
}

public function recap_laporan()
{
    $jumlah = 0;
    $data = [];
    return view('pvls.rekap_laporan',
compact('jumlah', 'data'));
}

public function chart()
{
    $reports = PVL::select(
        DB::raw('YEAR(tgl_agenda) as
year'),
        'id_tipe_laporan',
        DB::raw('count(*) as count')
    )
    ->groupBy('year', 'id_tipe_laporan')
    ->with('TipeLaporan')
    ->get();

    $pvls =
        $data = $reports->map(function
($report) {
            return [
                'year' => $report->year,
                'category' => $report-
>TipeLaporan->nama, // Asumsi 'nama'
adalah kolom kategori di tabel
TipeLaporan
                'count' => $report->count
            ];
        });

    // barchart

    $reports = PVL::select(
        DB::raw('YEAR(tgl_agenda) as
year'),
        DB::raw('MONTH(tgl_agenda) as
month'),
        DB::raw('count(*) as count')
    )
    ->groupBy('year', 'month')
    ->get();

    $data_bar = $reports-
>groupBy('year')->map(function
($yearGroup) {

```

```

        return $yearGroup-
>groupBy('month')->map(function
($monthGroup) {
    return $monthGroup-
>sum('count');
});
});

$data_triwulan_table =
PVL::select(DB::raw('
CASE
    WHEN MONTH(tgl_agenda)
    BETWEEN 1 AND 3 THEN "TW I"
    WHEN MONTH(tgl_agenda)
    BETWEEN 4 AND 6 THEN "TW II"
    WHEN MONTH(tgl_agenda)
    BETWEEN 7 AND 9 THEN "TW III"
    ELSE "TW IV"
END as TW,
SUM(CASE WHEN
tipe_laporan.nama = "IN" THEN 1 ELSE 0
END) as `IN`,
SUM(CASE WHEN
tipe_laporan.nama = "Konsultasi" THEN 1
ELSE 0 END) as Konsultasi,
SUM(CASE WHEN
tipe_laporan.nama = "Laporan
Masyarakat" THEN 1 ELSE 0 END) as
LaporanMasyarakat,
SUM(CASE WHEN
tipe_laporan.nama = "RCO" THEN 1
ELSE 0 END) as RCO,
SUM(CASE WHEN
tipe_laporan.nama = "Tembusan" THEN 1
ELSE 0 END) as Tembusan,
COUNT(*) as Jumlah
'))
->join('tipe_laporan',
'pvl.id_tipe_laporan', '=', 'tipe_laporan.id')
->groupBy(DB::raw('CASE
    WHEN MONTH(tgl_agenda)
    BETWEEN 1 AND 3 THEN "TW I"
    WHEN MONTH(tgl_agenda)
    BETWEEN 4 AND 6 THEN "TW II"
    WHEN MONTH(tgl_agenda)
    BETWEEN 7 AND 9 THEN "TW III"
    ELSE "TW IV"
END'))
->get();

// Menambahkan total keseluruhan
$total = PVL::select(DB::raw('
SUM(CASE WHEN
tipe_laporan.nama = "IN" THEN 1 ELSE 0
END) as `IN`,

```

```

SUM(CASE WHEN
tipe_laporan.nama = "Konsultasi" THEN 1
ELSE 0 END) as Konsultasi,
SUM(CASE WHEN
tipe_laporan.nama = "Laporan
Masyarakat" THEN 1 ELSE 0 END) as
LaporanMasyarakat,
SUM(CASE WHEN
tipe_laporan.nama = "RCO" THEN 1
ELSE 0 END) as RCO,
SUM(CASE WHEN
tipe_laporan.nama = "Tembusan" THEN 1
ELSE 0 END) as Tembusan,
COUNT(*) as Jumlah
'))
->join('tipe_laporan',
'pvl.id_tipe_laporan', '=', 'tipe_laporan.id')
->first();

// dd($data_bar);
return view('dashboard.chart',
compact('data', 'data_bar',
'data_triwulan_table', 'totals'));
}

public function data_triwulan()
{
    // Mengambil data berdasarkan
    kuartal dan tipe laporan
    $data = PVL::select(DB::raw('
CASE
    WHEN MONTH(tgl_agenda)
    BETWEEN 1 AND 3 THEN "TW I"
    WHEN MONTH(tgl_agenda)
    BETWEEN 4 AND 6 THEN "TW II"
    WHEN MONTH(tgl_agenda)
    BETWEEN 7 AND 9 THEN "TW III"
    ELSE "TW IV"
END as TW,
SUM(CASE WHEN
tipe_laporan.nama = "IN" THEN 1 ELSE 0
END) as `IN`,
SUM(CASE WHEN
tipe_laporan.nama = "Konsultasi" THEN 1
ELSE 0 END) as Konsultasi,
SUM(CASE WHEN
tipe_laporan.nama = "Laporan
Masyarakat" THEN 1 ELSE 0 END) as
LaporanMasyarakat,
SUM(CASE WHEN
tipe_laporan.nama = "RCO" THEN 1
ELSE 0 END) as RCO,
SUM(CASE WHEN
tipe_laporan.nama = "Tembusan" THEN 1
ELSE 0 END) as Tembusan,
COUNT(*) as Jumlah

```

```

    ))
    ->join('tipe_laporan',
'pvl.id_tipe_laporan', '=', 'tipe_laporan.id')
    ->groupBy(DB::raw('CASE
    WHEN MONTH(tgl_agenda)
    BETWEEN 1 AND 3 THEN "TW I"
    WHEN MONTH(tgl_agenda)
    BETWEEN 4 AND 6 THEN "TW II"
    WHEN MONTH(tgl_agenda)
    BETWEEN 7 AND 9 THEN "TW III"
    ELSE "TW IV"
    END'))
    ->get();

    // Menambahkan total keseluruhan
    $totals = PVL::select(DB::raw('
    SUM(CASE WHEN
    tipe_laporan.nama = "IN" THEN 1 ELSE 0
    END) as `IN`,
    SUM(CASE WHEN
    tipe_laporan.nama = "Konsultasi" THEN 1
    ELSE 0 END) as Konsultasi,
    SUM(CASE WHEN
    tipe_laporan.nama = "Laporan
    Masyarakat" THEN 1 ELSE 0 END) as
    LaporanMasyarakat,
    SUM(CASE WHEN
    tipe_laporan.nama = "RCO" THEN 1
    ELSE 0 END) as RCO,
    SUM(CASE WHEN
    tipe_laporan.nama = "Tembusan" THEN 1
    ELSE 0 END) as Tembusan,
    COUNT(*) as Jumlah
    '))
    ->join('tipe_laporan',
'pvl.id_tipe_laporan', '=', 'tipe_laporan.id')
    ->first();

    return view('pvls.report', ['data' =>
    $data, 'totals' => $totals]);
}

public function create()
{
    $pelapors = Pelapor::all();
    $pegawais = Pegawai::all();
    $instasis = Instansi::all();
    $subinstasis = Subinstansi::all();
    $tipelaporans = TipeLaporan::all();
    $instansis = Instansi::all();
    return view('pvls.create',
compact('pelapors', 'pegawais', 'instasis',
'subinstansis', 'tipelaporans', 'instansis'));
}

```

```

/**
 * Store a newly created resource in
storage.
 *
 * @param \Illuminate\Http\Request
$request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    $request->validate([
        'id_tipe_laporan' => 'required',
        'no_agenda' => 'required',
        'no_arsip' => 'required',
        'id_pelapor' => 'required',
        'id_instansi' => 'required',
        'id_subinstansi' => 'required',
        'perihal' => 'required',
        'id_pegawai' => 'required',
    ]);
    $input = $request->all();
    $input['tahap_saat_ini'] = 'register';
    PVL::create($input);
    return redirect()->route('pvl.index')
    ->with('success', 'Product created
successfully. ');
}

public function rekap()
{
    $data = [];
    $jumlah = 0;
    return view('pvls.rekap',
compact('data', 'jumlah'));
}

public function getPvlByYear(Request
$request)
{
    $year = $request->year;

    $data = PVL::whereYear('created_at',
$year)->get();

    $jumlah =
PVL::whereYear('created_at', $year)-
>get()->count();

    return view('pvls.rekap',
compact('data', 'jumlah'));
}

```

```

    public function
    getPvlByYearRekap(Request $request)
    {
        $year = $request->year;

        $data = PVL::whereYear('created_at',
        $year)->orderBy('id_tipe_laporan')->get();

        $jumlah =
        PVL::whereYear('created_at', $year)-
        >get()->count();

        return view("pvls.rekap_laporan",
        compact('data', 'jumlah'));
    }

    public function cetak_data_pvl(Request
    $request)
    {
        $start_date = $request-
        >input('startDate');
        $end_date = $request-
        >input('endDate');
        $data =
        PVL::whereBetween('tgl_agenda',
        [$start_date, $end_date])->get();
        return view('pvls.cetak_pvl',
        compact('data'));
    }

    public function detail($id)
    {
        $data = PVL::findOrFail($id);
        return view('pvls.detail',
        compact('data'));
    }

    public function show(PVL $pvl)
    {
        return view('pvls.show',
        compact('pvl'));
    }

    public function edit($id)
    {
        $pvls = PVL::findOrFail($id);
        $tipelaporans = TipeLaporan::all();
        $pegawais = Pegawai::all();
        $subinstansis = Subinstansi::all();
        $instansis = Instansi::all();
        $pelapors = Pelapor::all();

        return view('pvls.edit', compact('pvls',
        'tipelaporans', 'pegawais', 'subinstansis',
        'instansis', 'pelapors'));
    }

```

```

    public function update(Request
    $request, PVL $pvl)
    {
        $pvl->update($request->all());

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

    public function destroy(PVL $pvl)
    {
        $pvl->delete();

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

```

DivisiController.php

```

<?php

namespace App\Http\Controllers;

use App\Models\Divisi;
use Illuminate\Http\Request;

class DivisiController extends Controller
{
    function __construct()
    {
        $this->middleware(
        'permission:divisi-list|divisi-
        create|divisi-edit|divisi-delete',
        ['only' => ['index', 'show']]
        );
        $this->middleware('permission:divisi-
        create', ['only' => ['create', 'store']]);
        $this->middleware('permission:divisi-
        edit', ['only' => ['edit', 'update']]);
        $this->middleware('permission:divisi-
        delete', ['only' => ['destroy']]);
    }

    public function index_old()
    {
        $divisis = Divisi::latest()-
        >paginate(5);
        return view('divisis.index',
        compact('divisis'))
    }

```

```

->with('i', (request()->input('page',
1) - 1) * 5);
}
public function index(Request $request)
{
    if ($request->ajax()) {
        $query_data = new Divisi();
        if ($request->sSearch) {
            $search_value = '%' . $request-
>sSearch . '%';
            $query_data = $query_data-
>where(function ($query) use
($search_value) {
                $query->where('nama',
'catatan', $search_value);
            });
            $data = $query_data-
>orderBy('nama', 'asc')->get();
            return datatables()->of($data)-
>addIndexColumn()->addColumn('action',
function ($row) {
                $btn = '<form action="" .
route('divisis.destroy', $row->id) .
""method="POST">
                <a class="btn btn-info" href="" .
route('divisis.show', $row->id) .
"">Show</a>
                <a class="btn btn-primary" href="" .
route('divisis.edit', $row->id) . "">Edit</a>'
                . csrf_field() . method_field('DELETE') .
                '<button type="submit" class="btn btn-
danger">Delete</button>
                </form>';
                return $btn;
            })
            ->rawColumns(['action'])-
>make(true);
        }
        return view('divisis.index');
    }

    public function create()
    {
        return view('divisis.create');
    }
    public function store(Request $request)
    {
        $request->validate([
            'nama' => 'required',
        ]);
        Divisi::create($request->all());

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

```

```

->with('success', 'divisi created
successfully.');
```

```

    }

    public function show(Divisi $divisi)
    {
        return view('divisis.show',
compact('divisi'));
    }
    public function edit(Divisi $divisi)
    {
        return view('divisis.edit',
compact('divisi'));
    }
    public function update(Request
$request, Divisi $divisi)
    {
        $request->validate([
            'nama' => 'required',
        ]);
        $divisi->update($request->all());

        return redirect()->route('divisis.index')
        ->with('success', 'divisi updated
successfully');
    }
    public function destroy(Divisi $divisi)
    {
        $divisi->delete();

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

```

InstansiController.php

```
<?php
```

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

```
use App\Models\Instansi;
use Illuminate\Http\Request;
```

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

```

```

        $this->middleware('permission:instansi-create',
        ['only' => ['create', 'store']]);
        $this->middleware('permission:instansi-edit',
        ['only' => ['edit', 'update']]);
        $this->middleware('permission:instansi-delete',
        ['only' => ['destroy']]);
    }

    public function index_old()
    {
        $instansis = Instansi::latest()->paginate(5);
        return view('instansis.index',
        compact('instansis'))
        ->with('i', (request()->input('page',
        1) - 1) * 5);
    }

    public function index(Request $request)
    {
        if ($request->ajax()) {
            // $data = instansi::select('*');
            $query_data = new Instansi();
            if ($request->sSearch) {
                $search_value = '%' . $request->sSearch . '%';
                $query_data = $query_data->where(function ($query) use
                ($search_value) {
                    $query->where('nama',
                    'keterangan', $search_value);
                });
            }
            $data = $query_data->orderBy('nama', 'asc')->get();
            return datatables()->of($data)->addIndexColumn()->addColumn('action',
            function ($row) {
                // $btn = '<a
                href="javascript:void(0)" class="edit btn
                btn-primary btn-sm">View</a>';
                $btn = '<form action="" .
                route('instansis.destroy', $row->id) .
                ""method="POST">
                <a class="btn btn-info" href="" .
                route('instansis.show', $row->id) .
                "">Show</a>
                <a class="btn btn-primary" href="" .
                route('instansis.edit', $row->id) .
                "">Edit</a>' . csrf_field() .
                method_field('DELETE') . '<button
                type="submit" class="btn btn-
                danger">Delete</button>

```

```

        </form>';
        return $btn;
    }->addColumn('keterangan',
    function ($row) {
        if ($row->keterangan == null) {
            return "-";
        }
        return $row->keterangan;
    }->rawColumns(['action'])->make(true);
    }
    return view('instansis.index');
}

/**
 * Show the form for creating a new
 resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
    return view('instansis.create');
}

/**
 * Store a newly created resource in
 storage.
 *
 * @param \Illuminate\Http\Request
 $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    $request->validate([
        'nama' => 'required',
        'keterangan' => 'required',
    ]);

    Instansi::create($request->all());

    return redirect()->route('pvl.create')
    ->with('success', 'Product created
    successfully.');
```

```

        return view('Instansis.show',
compact('instansi'));
    }

/**
 * Show the form for editing the
specified resource.
 *
 * @param \App\Instansis $instansi
 * @return \Illuminate\Http\Response
 */
public function edit(Instansi $instansi)
{
    return view('instansis.edit',
compact('instansi'));
}

/**
 * Update the specified resource in
storage.
 *
 * @param \Illuminate\Http\Request
$request
 * @param \App\Instansi $instansi
 * @return \Illuminate\Http\Response
 */
public function update(Request
$request, Instansi $instansi)
{
    $request->validate([
        'nama' => 'required',
        'keterangan' => 'required',
    ]);

    $instansi->update($request->all());

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

/**
 * Remove the specified resource from
storage.
 *
 * @param \App\Product $product
 * @return \Illuminate\Http\Response
 */
public function destroy(Instansi
$instansi)
{
    $instansi->delete();

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

```

```

        ->with('success', 'Product deleted
successfully');
    }
}

PegawaiController.php
<?php

namespace App\Http\Controllers;

use App\Models\Divisi;
use App\Models\Pegawai;
use Illuminate\Http\Request;
use Yajra\DataTables\DataTables;

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_old()
    {
        $pegawais = Pegawai::latest()-
>paginate(5);
        return view('pegawais.index',
compact('pegawais'))
            ->with('i', (request()->input('page',
1) - 1) * 5);
    }

    public function index(Request $request)
    {
        if ($request->ajax()) {
            $query_data = new Pegawai();
            if ($request->sSearch) {
                $search_value = '%' . $request-
>sSearch . '%';
                $query_data = $query_data-
>where(function ($query) use
($search_value) {

```



```

        $query->where('nip', 'nama',
'email', 'nohp', $search_value);
    });
    }
    $data = $query_data->orderBy('id',
'asc')->get();
    return DataTables::of($data)-
>addIndexColumn()->addColumn('action',
function ($row) {
        $btn = '<form id="deleteForm_' .
$row->id . '" action="" .
route('pegawais.destroy', $row->id) . '"
method="POST">' .
        '<a class="btn btn-info"
href="" . route('pegawais.show', $row->id) .
"">Show</a>' .
        '<a class="btn btn-primary"
href="" . route('pegawais.edit', $row->id) .
"">Edit</a>' .
        csrf_field() .
        method_field('DELETE') .
        '<button type="submit"
class="btn btn-danger" onclick="return
confirm('\Are you sure you want to delete
this item?\')">Delete</button>' .
        '</form>';
        return $btn;
    }->addColumn('divisi', function
($row) {
        return $row->divisi->nama;
    })
    ->rawColumns(['action'])-
>make(true);
    }
    return view('pegawais.index');
}

public function create()
{
    $divisis = Divisi::all();
    return view('pegawais.create',
compact('divisis'));
}

public function store(Request $request)
{
    $request->validate([
        'nip' => 'required',
        'nama' => 'required',
        'nohp' => 'required',
        'email' => 'required',
        'id_divisi' => 'required',
    ]);
    Pegawai::create($request->all());

```

```

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

```

    }

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

    public function edit(Pegawai $pegawai)
    {
        $divisis = Divisi::all();
        return view('pegawais.edit',
compact('pegawai', 'divisis'));
    }

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

        return redirect()-
>route('pegawais.index')
        ->with('success', 'Product updated
successfully');
```

```

    }

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

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

```

    }

```

```

    public function
    cetak_data_pegawai(Request $request)
    {
        $data = Pegawai::all();
        return
        view('pegawais.cetak_pegawai',
        compact('data'));
    }
}

```

PelaporController.php

```
<?php
```

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

```
use App\Models\Pelapor;
use Illuminate\Http\Request;
```

```

class PelaporController extends Controller
{
    function __construct()
    {
        $this->middleware(
            'permission:pelapor-list|pelapor-
            create|pelapor-edit|pelapor-delete',
            ['only' => ['index', 'show']]
        );
        $this-
        >middleware('permission:pelapor-create',
        ['only' => ['create', 'store']]);
        $this-
        >middleware('permission:pelapor-edit',
        ['only' => ['edit', 'update']]);
        $this-
        >middleware('permission:pelapor-delete',
        ['only' => ['destroy']]);
    }
}

```

```

    public function index_old()
    {
        $pelapors = Pelapor::latest()-
        >paginate(5);
        return view('pelapors.index',
        compact('pelapors'))
        ->with('i', (request()->input('page',
        1) - 1) * 5);
    }
}

```

```

    public function index(Request $request)
    {
        if ($request->ajax()) {
            $query_data = new Pelapor();
            if ($request->sSearch) {
                $search_value = '%' . $request-
                >sSearch . '%';
            }
        }
    }
}

```

```

        $query_data = $query_data-
        >where(function ($query) use
        ($search_value) {
            $query->where('NIK', 'nama',
            'email', 'no.hp', $search_value);
        });
    }
    $data = $query_data-
    >orderBy('nama', 'asc')->get();
    return datatables()->of($data)-
    >addIndexColumn()->addColumn('action',
    function ($row) {
        $btn = '<form action="" .
        route('pelapor.destroy', $row->id) . ""
        method="POST">
            <a class="btn btn-info" href="" .
            route('pelapor.show', $row->id) .
            "">Show</a>
            <a class="btn btn-primary"
            href="" . route('pelapor.edit', $row->id) .
            "">Edit</a>' . csrf_field() .
            method_field('DELETE') . '<button
            type="submit" class="btn btn-
            danger">Delete</button>
            </form>;
        return $btn;
    })
    ->rawColumns(['action'])-
    >make(true);
}
return view('pelapors.index');
}

```

```

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

```

```

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

```

```

        Pelapor::create($request->all());

        return redirect()->route('pvl.create')
        ->with('success', 'Pelapor created
        successfully.');
```

```

    public function show(Pelapor $pelapor)
    {
        return view('pelapors.show', compact('pelapor'));
    }
}

```

```

    {
        return view('pelapors.show',
compact('pelapor'));
    }

    public function edit(Pelapor $pelapor)
    {
        return view('pelapors.edit',
compact('pelapor'));
    }

    public function update(Request
$request, Pelapor $pelapor)
    {
        $request->validate([
            'nik' => 'required',
            'nama' => 'required',
            'asal_daerah' => 'required',
            'email' => 'required',
            'nohp' => 'required',
        ]);

        $pelapor->update($request->all());

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

    public function destroy(Pelapor
$pelapor)
    {
        $pelapor->delete();

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

    public function
cetak_data_pelapor(Request $request)
    {
        $data = Pelapor::all();
        return view('pelapors.cetak_pelapor',
compact('data'));
    }
}

```

TipeLaporanController.php

<?php

namespace App\Http\Controllers;

use App\Models\TipeLaporan;
use Illuminate\Http\Request;

```

class TipeLaporanController extends
Controller
{
    function __construct()
    {
        $this->middleware(
            'permission:tipelaporan-
list|tipelaporan-create|tipelaporan-
edit|tipelaporan-delete',
            ['only' => ['index', 'show']]
        );
        $this-
>middleware('permission:tipelaporan-
create', ['only' => ['create', 'store']]);
        $this-
>middleware('permission:tipelaporan-edit',
['only' => ['edit', 'update']]);
        $this-
>middleware('permission:tipelaporan-
delete', ['only' => ['destroy']]);
    }
}

```

```

    public function index_old()
    {
        $tipelaporans = TipeLaporan::latest()-
>paginate(5);
        return view('tipelaporans.index',
compact('tipelaporans'))
        ->with('i', (request()->input('page',
1) - 1) * 5);
    }
}

```

```

    public function index(Request $request)
    {
        if ($request->ajax()) {
            // $data = divisi::select('*');
            $query_data = new TipeLaporan();
            if ($request->sSearch) {
                $search_value = '%' . $request-
>sSearch . '%';
                $query_data = $query_data-
>where(function ($query) use
($search_value) {
                    $query->where('nama', 'kode',
$search_value);
                });
            }
            $data = $query_data-
>orderBy('kode')->get();
            return datatables()->of($data)-
>addIndexColumn()->addColumn('action',
function ($row) {

```

```

        // $btn = '<a
href="javascript:void(0)" class="edit btn
btn-primary btn-sm">View</a>';
        $btn = '<form action="" .
route('tipelaporans.destroy', $row->id) .
""method="POST">
        <a class="btn btn-info" href="" .
route('tipelaporans.show', $row->id) .
"">Show</a>
        <a class="btn btn-primary" href="" .
route('tipelaporans.edit', $row->id) .
"">Edit</a>' . csrf_field() .
method_field('DELETE') . '<button
type="submit" class="btn btn-
danger">Delete</button>
        </form>;
        return $btn;
    })
    ->rawColumns(['action'])-
>make(true);
    }
    return view('tipelaporans.index');
}

```

```

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

```

```

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

```

```

TipeLaporan::create($request->all());

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

```

public function show(TipeLaporan
$tipelaporan)
{
    return view('tipelaporans.show',
compact('tipelaporan'));
}

```

```

public function edit(TipeLaporan
$tipelaporan)
{

```

```

        return view('tipelaporans.edit',
compact('tipelaporan'));
    }
    public function update(Request
$request, TipeLaporan $tipelaporan)
    {
        $request->validate([
            'nama' => 'required',
            'kode' => 'required',
        ]);

        $tipelaporan->update($request-
>all());

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

    public function destroy(TipeLaporan
$tipelaporan)
    {
        $tipelaporan->delete();
    }

```

```

        return redirect()-
>route('tipelaporans.index')
        ->with('success', 'Product 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
{
    //
    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);
    }
}

```

```

    }

    public function create()
    {
        $roles = Role::pluck('name', 'name')-
>all();
        return view('users.create',
compact('roles'));
    }

    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');
    }

    public function show($id)
    {
        $user = User::find($id);
        return view('users.show',
compact('user'));
    }

    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'));
    }

    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');
    }

    public function destroy($id)
    {
        User::find($id)->delete();
        return redirect()->route('users.index')
->with('success', 'User deleted
successfully');
    }
}

```

PVL.php (Models)

```

<?php

namespace App\Models;

use
Illuminate\Database\Eloquent\Factories\Ha
sFactory;
use Illuminate\Database\Eloquent\Model;

class PVL extends Model
{
    use HasFactory;
    protected $table = "PVL";
    protected $guarded = ['id'];

    public function pegawai()
    {
        return $this->hasOne(pegawai::class,
'id', 'id_pegawai');
    }

    public function TipeLaporan()
    {

```

```

        return $this->hasOne(TipeLaporan::class, 'id',
'id_tipe_laporan');
    }

    public function pelapor()
    {
        return $this->hasOne(Pelapor::class,
'id', 'id_pelapor');
    }

    public function subinstansi()
    {
        return $this->hasOne(Subinstansi::class, 'id',
'id_subinstansi');
    }

    public function instansi()
    {
        return $this->hasOne(Instansi::class,
'id', 'id_instansi');
    }

    public function asisteriksa()
    {
        return $this->hasOne(Pegawai::class,
'id', 'asisten_riksa');
    }
}

```

Divisi.php (Models)

```
<?php
```

```
namespace App\Models;
```

```

class Divisi extends Model
{
    use HasFactory;

    protected $guarded = ['id'];
}

```

Instansi.php (Models)

```
<?php
```

```
namespace App\Models;
```

```

class Instansi extends Model
{
    use HasFactory;
    protected $table = "instansi";
    protected $guarded = ['id'];
}

```

Pegawai.php (Models)

```
<?php
```

```
namespace App\Models;
```

```

class Pegawai extends Model
{
    use HasFactory;
    protected $table = "pegawai";
    protected $fillable = ['nip', 'nama',
'email', 'nohp', 'id_divisi'];

    public function divisi()
    {
        return $this->belongsTo(Divisi::class,
'id_divisi', 'id');
    }
}

```

Pelapor.php (Models)

```
<?php
```

```
namespace App\Models;
```

```

class Pelapor extends Model
{
    use HasFactory;
    protected $table = "pelapor";
    protected $guarded = ['id'];
}

```

TipeLaporan.php (Models)

```
<?php
```

```
namespace App\Models;
```

```

class TipeLaporan extends Model
{
    use HasFactory;
    protected $table = "tipe_laporan";
    protected $guarded = ['id'];
}

```

User.php (Models)

```
<?php
```

```
namespace App\Models;
```

```

class User extends Authenticatable
{
    use HasApiTokens, HasFactory,
Notifiable, HasRoles;

    protected $fillable = [
        'name',
        'email',
        'password',
    ];
}

```

```
];

protected $hidden = [
    'password',
    'remember_token',
];

protected $casts = [
    'email_verified_at' => 'datetime',
];
}
```

Create_users_table.php

```
<?php

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

return new class extends Migration
{

    public function up()
    {
        Schema::create('users', function
(Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email')->unique();
            $table-
>timestamp('email_verified_at')-
>nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('users');
    }
};
```

Create_tabel_divisi.php

```
<?php

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

return new class extends Migration
{
    public function up()
```

```
{
    Schema::create('divisi', function
(Blueprint $table) {
        $table->id();
        $table->string('nama');
        $table->string('catatan')-
>nullable();
        $table->string('created_by')-
>nullable();
        $table->string('updated_by')-
>nullable();
        $table->string('deleted_by')-
>nullable();
        $table->softDeletes();
        $table->timestamps();
    });
}

    public function down()
    {
        Schema::dropIfExists('tabel_divisi');
    }
};
```

Create_tabel_pegawai.php

```
<?php

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

return new class extends Migration
{
    public function up()
    {
        Schema::create('pegawai', function
(Blueprint $table) {
            $table->id();
            $table->string('nip');
            $table->string('nama');
            $table->string('email');
            $table->string('nohp');
            $table->foreignId('id_divisi')-
>constrained('divisi')-
>cascadeOnUpdate()->cascadeOnDelete();
            $table->string('created_by')-
>nullable();
            $table->string('updated_by')-
>nullable();
            $table->string('deleted_by')-
>nullable();
            $table->softDeletes();
            $table->timestamps();
        });
    }
};
```

```

        public function down()
        {
            Schema::dropIfExists('pegawai');
        }
    };

```

Create_tabel_tipe_laporan.php

```
<?php
```

```

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

```

```
return new class extends Migration
```

```

{
    public function up()
    {
        Schema::create('tipe_laporan',
function (Blueprint $table) {
            $table->id();
            $table->string('nama');
            $table->string('kode');
            $table->string('created_by')-
>nullable();
            $table->string('updated_by')-
>nullable();
            $table->string('deleted_by')-
>nullable();
            $table->softDeletes();
            $table->timestamps();
        });
    }

```

```

        public function down()
        {
            Schema::dropIfExists('tipe_laporan');
        }
    };

```

Create_tabel_pelapor.php

```
<?php
```

```

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

```

```
return new class extends Migration
```

```

{
    public function up()
    {
        Schema::create('pelapor', function
(Blueprint $table) {
            $table->id();
            $table->string('nama');

```

```

            $table->string('nik');
            $table->string('asal_daerah');
            $table->string('email');
            $table->string('nohp');
            $table->string('created_by')-
>nullable();
            $table->string('updated_by')-
>nullable();
            $table->string('deleted_by')-
>nullable();
            $table->softDeletes();
            $table->timestamps();
        });
    }

```

```

        public function down()
        {
            Schema::dropIfExists('pelapor');
        }
    };

```

Create_tabel_instansi.php

```
<?php
```

```

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

```

```
return new class extends Migration
{
```

```

    public function up()
    {
        Schema::create('instansi', function
(Blueprint $table) {
            $table->id();
            $table->string('nama');
            $table->string('keterangan')-
>nullable();
            $table->string('created_by')-
>nullable();
            $table->string('updated_by')-
>nullable();
            $table->string('deleted_by')-
>nullable();
            $table->softDeletes();
            $table->timestamps();
        });
    }

```

```

        public function down()
        {
            Schema::dropIfExists('instansi');
        }
    };

```


Create_tabel_pvl.php

```
<?php

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

return new class extends Migration
{
    public function up()
    {
        Schema::create('pvl', function
(Blueprint $table) {
            $table->id();
            $table-
>foreignId('id_tipe_laporan')-
>constrained('tipe_laporan')-
>cascadeOnUpdate()->cascadeOnDelete();
            $table->foreignId('id_pelapor')-
>constrained('pelapor')-
>cascadeOnUpdate()->cascadeOnDelete();
            $table->foreignId('id_subinstansi')-
>constrained('subinstansi')-
>cascadeOnUpdate()->cascadeOnDelete();
            $table->foreignId('id_instansi')-
>constrained('instansi')-
>cascadeOnUpdate()->cascadeOnDelete();
            $table->foreignId('id_pegawai')-
>constrained('pegawai')-
>cascadeOnUpdate()->cascadeOnDelete();
            $table->string('perihal');
            $table->string('no_agenda');
            $table->string('no_arsip');
            $table->string('tgl_agenda');
            $table->string('status')->nullable();
            // $table->foreignId('asisten_riksa')-
>constrained('pegawai')-
>cascadeOnUpdate()->cascadeOnDelete();
            $table-
>string('tanggal_diserahkan_pvl_kaper')-
>nullable();
            $table->string('tgl_BAPL')-
>nullable();
            $table-
>string('tgl_diserahkan_riksa')->nullable();
            $table->string('tahap_saat_ini')-
>nullable();
            $table->string('created_by')-
>nullable();
            $table->string('updated_by')-
>nullable();
            $table->string('deleted_by')-
>nullable();
            $table->softDeletes();
        });
    }
}
```

```
        $table->timestamps();
    });
}

public function down()
{
    Schema::dropIfExists('pvl');
}
};
```

CreateAdminUserSeeder.php

```
<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;
use App\Models\User;
use Spatie\Permission\Models\Role;
use Spatie\Permission\Models\Permission;

class CreateAdminUserSeeder extends
Seeder
{
    public function run()
    {
        $user = User::create([
            'name' => 'cyborg',
            'email' => 'cyborg@yopmail.com',
            'password' => bcrypt('12345678')
        ]);
        $role = Role::create(['name' =>
'Admin']);
        $permissions = Permission::pluck('id',
'id')->all();
        $role-
>syncPermissions($permissions);
        $user->assignRole([$role->id]);
    }
}
```

DatabaseSeeder.php

```
<?php

namespace Database\Seeders;
use Illuminate\Database\Seeder;

class DatabaseSeeder extends Seeder
{
    public function run()
    {
        $this->call([
            PermissionTableSeeder::class,
            CreateAdminUserSeeder::class
        ]);
    }
}
```

```

    }
}

```

PermissionTableSeeder.php

```

<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;
use Spatie\Permission\Models\Permission;

class PermissionTableSeeder extends
Seeder
{
    public function run()
    {
        $permissions = [
            'role-list',
            'role-create',
            'role-edit',
            'role-delete',
            'user-list',
            'user-create',
            'user-edit',
            'user-delete',
            'divisi-list',
            'divisi-create',
            'divisi-edit',
            'divisi-delete',
            'instansi-list',
            'instansi-create',
            'instansi-edit',
            'instansi-delete',
            'pegawai-list',
            'pegawai-create',
            'pegawai-edit',
            'pegawai-delete',
            'pelapor-list',
            'pelapor-create',
            'pelapor-edit',
            'pelapor-delete',
            'pvl-list',
            'pvl-create',
            'pvl-edit',
            'pvl-delete',
            'subinstansi-list',
            'subinstansi-create',
            'subinstansi-edit',
            'subinstansi-delete',
            'tipelaporan-list',
            'tipelaporan-create',
            'tipelaporan-edit',
            'tipelaporan-delete',
            'laporanmasuk-list',
            'laporanmasuk-create',
            'laporanmasuk-edit',

```

```

        'laporanmasuk-delete',
        'laporan-kaper',

```

```

    ];
    foreach ($permissions as $permission)
    {
        Permission::create(['name' =>
        $permission]);
    }
}

```

Web.php (Routes)

```

<?php

use App\Http\Controllers\DivisiController;
use
App\Http\Controllers\InstansiController;
use
App\Http\Controllers\LaporanMasukContr
oller;
use
App\Http\Controllers\PegawaiController;
use
App\Http\Controllers\PelaporController;
use App\Http\Controllers\PVLController;
use
App\Http\Controllers\SubinstansiControlle
r;
use
App\Http\Controllers\TipeLaporanControll
er;
use
App\Http\Controllers\Auth\AuthController
;
use App\Http\Controllers\RoleController;
use App\Http\Controllers\UserController;
use App\Models\Subinstansi;
use App\Models\TipeLaporan;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    return view('welcome');
});
Route::resource('pegawais',
PegawaiController::class);
Route::resource('tipelaporans',
TipeLaporanController::class);
Route::resource('instansis',
InstansiController::class);
Route::resource('pvl',
PVLController::class);

```

```

Route::resource('subinstansi',
SubinstansiController::class);
Route::resource('pelapor',
PelaporController::class);
Route::resource('divisi',
DivisiController::class);
Route::get('login', [AuthController::class,
'index'])->name('login');
Route::post('post-login',
[AuthController::class, 'postLogin'])-
>name('login.post');
Route::get('registration',
[AuthController::class, 'registration'])-
>name('register');
Route::post('post-registration',
[AuthController::class, 'postRegistration'])-
>name('register.post');
Route::get('dashboard',
[AuthController::class, 'dashboard'])-
>name('dashboard');
Route::get('logout', [AuthController::class,
'logout'])->name('logout');
Route::get('rekap_laporan',
[PVLController::class, 'rekap'])-
>name('rekap.laporan');
Route::post('rekap-search',
[PVLController::class, 'getPvlByYear'])-
>name('rekap.search');
Route::post('rekap-search-laporan',
[PVLController::class,
'getPvlByYearRekap'])-
>name('rekapsearch.laporan');
Route::get('rekap_laporan_page',
[PVLController::class,
'rekap_laporan_page'])-
>name('rekap_laporan_page');

Route::post('cetak-rekap-pvl',
[PVLController::class, 'cetak_data_pvl'])-
>name('cetak.pvl');
Route::get('cetak-data-pelapor',
[PelaporController::class,
'cetak_data_pelapor'])-
>name('cetak.pelapor');
Route::get('cetak-data-pegawai',
[PegawaiController::class,
'cetak_data_pegawai'])-
>name('cetak.pegawai');
Route::resource('users',
UserController::class);
Route::resource('roles',
RoleController::class);
Route::get('sendtokaper/{id}',
[PVLController::class, 'sendtokaper'])-
>name('sendtokaper');

```

```

Route::get('detail/{id}',
[PVLController::class, 'detail'])-
>name('detail');
Route::get('chart', [PVLController::class,
'chart'])->name('chart');

Route::get('process-riksa/{id}',
[PVLController::class,
'tgl_diserahkan_riksa'])-
>name('process.riksa');
Route::get('rekap-laporan',
[PVLController::class, 'rekap_laporan'])-
>name('rekaplaporan');
Route::get('data-triwulan',
[PVLController::class, 'data_triwulan']);

```

Dashboard-->chart.blade.php

```

@extends('layout')
@section('content')
<style>
    .tw-1 {
        background-color: #f2dede;
        /* Triwulan 1 - Light Red */
    }

    .tw-2 {
        background-color: #d9edf7;
        /* Triwulan 2 - Light Green */
    }

    .tw-3 {
        background-color: #d9edf7;
        /* Triwulan 3 - Light Blue */
    }

    .tw-4 {
        background-color: #fcf8e3;
        /* Triwulan 4 - Light Yellow */
    }
</style>

<div class="row column1">
    <div class="col-lg-6">
        <div class="white_shd full
margin_bottom_30">
            <div class="full graph_head">
                <div class="heading1
margin_0 center">
                    <div id="chartContainer">
                        <select
id="quarterSelect" style="margin-bottom:
20px;">
                            <option
value="1">Triwulan 1</option>

```

```

        <option
value="2">Triwulan 2</option>
        <option
value="3">Triwulan 3</option>
        <option
value="4">Triwulan 4</option>
    </select>
    <canvas
id="myBarChart" width="400"
height="400"></canvas>
    </div>
</div>
</div>
</div>
    { {-- Pertahun --} }
    <div class="col-lg-6">
        <div class="white_shd full
margin_bottom_30">
            <div class="full graph_head">
                <div class="heading1
margin_0 center">
                    <div id="chartContainer">
                        <select id="yearSelect"
style="margin-bottom: 20px;"></select>
                        <div
id="chartContainer">
                            <canvas
id="myPieChart" width="400"
height="400"></canvas>
                        </div>
                        <style>
                            #chartContainer {
                                max-width: 400px;
                                margin: auto;
                            }
                        </style>

                        <script>
                            const rawData = <?=$data ?>;

                            const yearSelect =
document.getElementById('yearSelect');
                            const years = [...new
Set(rawData.map(item => item.year))];
                            years.forEach(year =>
{
                                const option =
document.createElement('option');
                                option.value = year;
                                option.text = year;

                                yearSelect.appendChild(option);
                            });

```

```

// Fungsi untuk
memfilter data berdasarkan tahun
function
filterDataByYear(year) {
    return
    rawData.filter(item => item.year == year);
}

// Fungsi untuk
memperbarui chart
function
updateChart(chart, year) {

    console.log(year);
    const filteredData =
filterDataByYear(year);

    console.log(filteredData);

    const labels =
filteredData.map(item => item.category);
    console.log(labels);
    const data =
filteredData.map(item => item.count);
    console.log(data);

    chart.data.labels =
labels;

    chart.data.datasets[0].data = data;
    chart.update();
}

// Inialisasi chart
const ctx =
document.getElementById('myPieChart').g
etContext('2d');

const myPieChart =
new Chart(ctx, {
    type: 'pie',
    data: {
        labels: [],
        datasets: [{
            label: 'Jumlah
Laporan',

            data: [],

            backgroundColor: [
                'rgba(255,
99, 132, 0.2)',
                'rgba(54,
162, 235, 0.2)',
                'rgba(255,
206, 86, 0.2)',
                'rgba(75,
192, 192, 0.2)',

```

```

        'rgba(153,
102, 255, 0.2)',
        'rgba(255,
159, 64, 0.2)'
    ],
    borderColor: [
        'rgba(255,
99, 132, 1)',
        'rgba(54,
162, 235, 1)',
        'rgba(255,
206, 86, 1)',
        'rgba(75,
192, 192, 1)',
        'rgba(153,
102, 255, 1)',
        'rgba(255,
159, 64, 1)'
    ],
    borderWidth: 1
  }},
  options: {
    responsive: true
  }
});

// Event listener untuk
dropdown

yearSelect.addEventListener('change',
function() {
    // const selectedYear
    = this.value;

    updateChart(myPieChart, selectedYear);

    const selectedYear =
this.value;

    const
selectedQuarter =
document.getElementById('quarterSelect').
value;

    updateBarChart(myBarChart,
selectedYear, selectedQuarter);
});

// Set default tahun saat
pertama kali di-load
if (rawData.length > 0)
{

```

```

        yearSelect.value =
years[0];

        updateChart(myPieChart, years[0]);
    }
</script>

<script>
    const rawData2 =
@json($data_bar);
    console.log(rawData2);

    // Fungsi untuk
memfilter data berdasarkan triwulan
function
filterDataByQuarter(year, quarter) {
    const months = {
        1: [1, 2, 3], //
Triwulan 1: Januari, Februari, Maret
        2: [4, 5, 6], //
Triwulan 2: April, Mei, Juni
        3: [7, 8, 9], //
Triwulan 3: Juli, Agustus, September
        4: [10, 11, 12] //
Triwulan 4: Oktober, November,
Desember
    };

    return
months[quarter].map(month => ({
        month,
        count:
rawData2[year][month] || 0
    }));
}

// Fungsi untuk
memperbarui chart
function
updateBarChart(chart, year, quarter) {
    const filteredData =
filterDataByQuarter(year, quarter);

    console.log(filteredData);
    const labels =
filteredData.map(item => new Date(0,
item.month - 1).toLocaleString('default', {
        month: 'long'
    }));
    const data =
filteredData.map(item => item.count);

    chart.data.labels =
labels;

```

```

chart.data.datasets[0].data = data;
    chart.update();
}

// Inisialisasi chart
const ctx2 =
document.getElementById('myBarChart').
getContext('2d');
const myBarChart =
new Chart(ctx2, {
    type: 'bar',
    data: {
        labels: [],
        datasets: [{
            label: 'Jumlah
Laporan',
            data: [],
            backgroundColor: 'rgba(75, 192, 192, 0.2)',
            borderColor:
'rgba(75, 192, 192, 1)',
            borderWidth: 1
        }]
    },
    options: {
        responsive: true,
        scales: {
            y: {
                beginAtZero:
true
            }
        }
    }
});

// Event listener untuk
dropdown

document.getElementById('quarterSelect').
addEventListener('change', function() {
    const
selectedQuarter = this.value;

console.log(selectedQuarter);
    const selectedYear =
document.getElementById('yearSelect').val
ue;

updateBarChart(myBarChart,
selectedYear, selectedQuarter);
});

// Set default triwulan
saat pertama kali di-load

//
updateBarChart(myBarChart, new
Date().getFullYear(), 1);
    if (years.length > 0) {
        yearSelect.value =
years[0];

updateBarChart(myBarChart, years[0], 1);
    }
</script>
</div>
</div>
</div>
</div>
</div>
<div class="container mt-4">
    <h1>Report</h1>
    <table class="table table-bordered
table-striped">
        <thead class="thead-dark">
            <tr>
                <th>TW</th>
                <th>IN</th>
                <th>Konsultasi</th>
                <th>Laporan Masyarakat</th>
                <th>RCO</th>
                <th>Tembusan</th>
                <th>Jumlah</th>
            </tr>
        </thead>
        <tbody>
            @foreach ($data_triwulan_table
as $index => $row)
                <tr class="tw-{{ ($index % 4)
+ 1 }}">
                    <td>{{ $row->TW }}</td>
                    <td>{{ $row->IN }}</td>
                    <td>{{ $row->Konsultasi
}}</td>
                    <td>{{ $row-
>LaporanMasyarakat }}</td>
                    <td>{{ $row->RCO
}}</td>
                    <td>{{ $row->Tembusan
}}</td>
                    <td>{{ $row->Jumlah
}}</td>
                </tr>
            @endforeach
        </tbody>
        <tr>
            <td><strong>Total</strong></td>
            <td><strong>{{ $totals->IN
}}</strong></td>

```

```

        <td><strong>{{ $totals-
>Konsultasi }}</strong></td>
        <td><strong>{{ $totals-
>LaporanMasyarakat }}</strong></td>
        <td><strong>{{ $totals->RCO
}}</strong></td>
        <td><strong>{{ $totals-
>Tembusan }}</strong></td>
        <td><strong>{{ $totals-
>Jumlah }}</strong></td>
    </tr>
</tbody>
</table>
</div>
@endsection

```

Divisis.index.blade.php

```

@extends('layout')
@section('content')
    <div class="row">
        <div class="col-lg-12 margin-tb">
            <div class="pull-right">
                <a class="btn btn-success my-3
ml-4" href="{{ route('divisis.create') }}">
Create new</a>
            </div>
        </div>
    </div>

    @if ($message = Session::get('success'))
        <div class="alert alert-success">
            <p>{{ $message }}</p>
        </div>
    @endif

    <div class="center">
        <h4><strong> DAFTAR DATA
KEASISTENAN</strong></h4>
    </div>
    <div class="container p-4">
        <table class="table table-bordered
data-table">
            <thead>
                <tr>
                    <th>No</th>
                    <th>nama</th>
                    <th>catatan</th>
                    <th
width="280px">Action</th>
                </tr>
            </thead>
            <tbody>
            </tbody>
        </table>
    </div>
    <script type="text/javascript">
        $(function() {

```

```

        var table = $('<div>
data-
table').DataTable({
            processing: true,
            serverSide: true,
            ajax: "{{ route('divisis.index')
}}",
            columns: [{
                data: 'DT_RowIndex',
                name: 'DT_RowIndexid'
            },
            {
                data: 'nama',
                name: 'nama'
            },
            {
                data: 'catatan',
                name: 'catatan'
            },
            {
                data: 'action',
                name: 'action',
                orderable: false,
                searchable: false
            },
        ]
    });
</script>
@endsection

```

Instansis.index.blade.php

```

@extends('layout')
@section('content')
    <div class="row">
        <div class="col-lg-12 margin-tb">
            <div class="pull-right">
                <a class="btn btn-success my-3
ml-4" href="{{ route('instansis.create')
}}"> Create New</a>
            </div>
        </div>
    </div>

    @if ($message = Session::get('success'))
        <div class="alert alert-success">
            <p>{{ $message }}</p>
        </div>
    @endif

    <div class="center">
        <h4><strong> DAFTAR DATA
INSTANSI</strong></h4>
    </div>
    <div class="container mr-4">
        <table class="table table-bordered
data-table">

```

```

        <thead>
        <tr>
            <th>No</th>
            <th>nama</th>
            <th>keterangan</th>
            <th
width="280px">Action</th>
        </tr>
        </thead>
        <tbody>
        </tbody>
    </table>
</div>
<script type="text/javascript">
    $(function() {
        var table = $('#data-
table').DataTable({
            processing: true,
            serverSide: true,
            ajax: "{{ route('instansis.index')
}}",
            columns: [{
                data: 'DT_RowIndex',
                name: 'DT_RowIndex'
            },
            {
                data: 'nama',
                name: 'nama'
            },
            {
                data: 'keterangan',
                name: 'keterangan'
            },
            {
                data: 'action',
                name: 'action',
                orderable: false,
                searchable: false
            },
        ]
    });
</script>
@endsection

```

Pegawais.index.blade.php

```

@extends('layout')
@section('content')
    <div class="row">
        <div class="col-lg-12 margin-tb">
            <div class="pull-right">
                <a class="btn btn-success my-3
ml-4" href="{{ route('pegawais.create')
}}"> Create New</a>
            </div>
        </div>
    </div>

```

```

</div>
@if ($message = Session::get('success'))
    <div class="alert alert-success">
        <p>{{ $message }}</p>
    </div>
@endif
<div class="center">
    <h4><strong> DAFTAR DATA
PEGAWAI</strong></h4>
</div>
<form id="printForm" action="{{ {
route('cetak.pegawai') }}" method="GET"
target="_blank">
    <button type="submit">Print
Data</button>
</form>
<div class="container p-4">
    <table class="table table-bordered
data-table">
        <thead>
            <tr>
                <th>nip</th>
                <th>nama</th>
                <th>nohp</th>
                <th>email</th>
                <th>divisi</th>
                <th
width="280px">Action</th>
            </tr>
        </thead>
        <tbody>
        </tbody>
    </table>
</div>
<script type="text/javascript">
    $(function() {
        var table = $('#data-
table').DataTable({
            processing: true,
            serverSide: true,
            ajax: "{{ route('pegawais.index')
}}",
            columns: [{
                data: 'nip',
                name: 'nip'
            },
            {
                data: 'nama',
                name: 'nama'
            },
            {
                data: 'nohp',
                name: 'nohp'
            },
            {
                data: 'email',

```



```

        name: 'email'
    },
    {
        data: 'divisi',
        name: 'divisi'
    },
    {
        data: 'action',
        name: 'action',
        orderable: false,
        searchable: false
    },
    ]
});
</script>
@endsection

```

Pelapors.index.blade.php

```

@extends('layout')
@section('content')
    <div class="row">
        <div class="col-lg-12 margin-tb">
            <div class="pull-left">
                {{-- <h2>Halaman Pelapor</h2>
            --}}
            </div>
        </div>
    </div>

    @if ($message = Session::get('success'))
        <div class="alert alert-success">
            <p>{{ $message }}</p>
        </div>
    @endif

    <h2>Cetak Data Pelapor</h2>
    @csrf
    <form id="printForm" action="{{ {
route('cetak.pelapor') }}" method="GET"
target="_blank">
        <button type="submit">Print
Data</button>
    </form>
    <div class="container p-4">
        <table class="table table-bordered
data-table">
            <thead>
                <tr>
                    <th>NIK</th>
                    <th>nama</th>
                    <th>asal</th>
                    <th>email</th>
                    <th>no.hp</th>
                    <th
width="280px">Action</th>

```

```

        </tr>
    </thead>
    <tbody>
    </tbody>
    </table>
</div>
<script type="text/javascript">
    $(function() {
        var table = $('<table>').DataTable({
            processing: true,
            serverSide: true,
            ajax: "{{ route('pelapor.index')
        }}",
            columns: [{
                data: 'nik',
                name: 'nik'
            },
            {
                data: 'nama',
                name: 'nama'
            },
            {
                data: 'asal_daerah',
                name: 'asal_daerah'
            },
            {
                data: 'email',
                name: 'email'
            },
            {
                data: 'nohp',
                name: 'nohp'
            },
            {
                data: 'action',
                name: 'action',
                orderable: false,
                searchable: false
            },
        ]
    });
</script>
@endsection

```

Pvl.index.blade.php

```

@extends('layout')
@section('content')
    <div class="row">
        <div class="col-lg-12 margin-tb">
            <div class="pull-right">
                <a class="btn btn-success my-3
ml-4" href="{{ { route('pvl.create') }}">
Create New</a>
            </div>

```

```

</div>
</div>

@if ($message = Session::get('success'))
    <div class="alert alert-success">
        <p>{{ $message }}</p>
    </div>
@endif
@csrf
<div class="center">
    <h2><strong> Laporan
Masuk</strong> </h2>
</div>

<div class="container p-4">
    <table class="table table-bordered
data-table">
        <thead>
            <tr>
                <th>Tipe Laporan</th>
                <th>No Agenda</th>
                <th>No Arsip</th>
                <th>Pelapor</th>
                <th>Instansi</th>
                {{ -- <th>Subinstansi</th> -- }}
                {{ -- <th>Perihal</th> -- }}
                {{ -- <th>Tgl Agenda</th> -- }}
                <th>Status Laporan</th>
                <th>Asisten PVL</th>
                <th>Tgl Diserahkan ke
Kaper</th>
                <th>Tahap Saat Ini</th>
                {{ -- <th>Asisten PJ
Riksa</th> -- }}
                {{ -- <th>Tanggal Diserahkan
Riksa</th> -- }}
                <th>Tgl BA Pleno</th>
                <th
width="280px">Action</th>
            </tr>
        </thead>
        <tbody>
        </tbody>
    </table>
</div>
<script type="text/javascript">
    $(function() {
        var table = $('#data-
table').DataTable({
            processing: true,
            scrollX: true,
            serverSide: true,
            ajax: "{{ route('pvl.index') }}",
            columns: [{
data: 'TipeLaporan',
name: 'TipeLaporan'
},
{
data: 'no_agenda',
name: 'no_agenda'
},
{
data: 'no_arsip',
name: 'no_arsip'
},
{
data: 'pelapor',
name: 'pelapor'
},
{
data: 'instansi',
name: 'instansi'
},
// {
// data: 'subinstansi',
// name: 'subinstansi'
// },
// {
// data: 'perihal',
// name: 'perihal'
// },
// {
// data: 'tgl_agenda',
// name: 'tgl_agenda'
// },
{
data: 'status',
name: 'status'
},
{
data: 'pegawai',
name: 'pegawai'
},
{
data:
'tanggal_diserahkan_pvl_kaper',
name:
'tanggal_diserahkan_pvl_kaper'
},
{
data: 'tahap_saat_ini',
name: 'tahap_saat_ini'
},
// {
// data: 'asisten_riksa',
// name: 'asisten_riksa'
// },
// {
// data:
'tgl_diserahkan_riksa',

```

```

        // name:
        'tgl_diserahkan_riksa'
        // },
        {
            data: 'tgl_BAPL',
            name: 'tgl_BAPL'
        },
        {
            data: 'action',
            name: 'action',
            orderable: false,
            searchable: false
        },
    ]
    });
    });
</script>
@endsection

```

TipeLaporan.index.blade.php

```

@extends('layout')
@section('content')
    <div class="row">
        <div class="col-lg-12 margin-tb">
            <div class="pull-left">
                {{-- <h2>Halaman Tipe
                Laporan</h2> --}}
            </div>
            <div class="pull-right">
                <a class="btn btn-success my-3
                ml-4" href="{{ route('tipelaporans.create')
                }}"> Create New</a>
            </div>
        </div>
    </div>

```

```

    @if ($message = Session::get('success'))
        <div class="alert alert-success">
            <p>{{ $message }}</p>
        </div>
    @endif
    <div class="container p-4">
        <table class="table table-bordered
        data-table">
            <thead>
                <tr>
                    <th>No</th>
                    <th>nama</th>
                    <th>kode</th>
                    <th
                    width="280px">Action</th>
                </tr>
            </thead>
            <tbody>
            </tbody>
        </table>

```

```

</div>
<script type="text/javascript">
    $(function() {
        var table = $('#data-
        table').DataTable({
            processing: true,
            serverSide: true,
            ajax: "{{
            route('tipelaporans.index') }}" ,
            columns: [{
                data: 'id',
                name: 'id'
            },
            {
                data: 'nama',
                name: 'nama'
            },
            {
                data: 'kode',
                name: 'kode'
            },
            {
                data: 'action',
                name: 'action',
                orderable: false,
                searchable: false
            },
        ]
    });
    });
</script>
@endsection

```



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 : Popi Nabilla
NPM : 062130800241
Jurusan : Manajemen Informatika
Program Studi : DIII Manajemen Informatika

Pihak Kedua

Nama : Muhammad Noval, S.E., M.Si.
NIP : 197511082005011003
Jurusan : Manajemen Informatika
Program Studi : DIII Manajemen Informatika

Pada hari ini Kamis tanggal 14 Maret 2024 telah sepakat untuk melakukan konsultasi bimbingan Laporan Akhir.

Konsultasi bimbingan sekurang-kurangnya 1 (satu) kali dalam satu minggu. Pelaksanaan bimbingan pada setiap hari Rabu/Kamis pukul 13.00 WIB - 14.00 WIB, tempat di Politeknik Negeri Sriwijaya.

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

Palembang, 14 Maret 2024

Pihak Pertama,

Popi Nabilla
NIM 062130800241

Pihak Kedua,

Muhammad Noval, S.E., M.Si.
NIP 197511082005011003

Mengetahui,
Ketua Jurusan

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



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

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 : Popi Nabilla
NPM : 062130800241
Jurusan : Manajemen Informatika
Program Studi : DIII Manajemen Informatika

Pihak Kedua

Nama : Meivi Kusnandar, S.Kom., M.Kom.
NIP : 197407052002121014
Jurusan : Manajemen Informatika
Program Studi : DIII Manajemen Informatika

Pada hari ini Kamis tanggal 14 Maret 2024 telah sepakat untuk melakukan konsultasi bimbingan Laporan Akhir.

Konsultasi bimbingan sekurang-kurangnya 1 (satu) kali dalam satu minggu. Pelaksanaan bimbingan pada setiap hari Selasa/Rabu pukul 13.00 WIB - 14.00 WIB, tempat di Politeknik Negeri Sriwijaya.

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

Palembang, 14 Maret 2024

Pihak Pertama,

Popi Nabilla
NIM 062130800241

Pihak Kedua,

Meivi Kusnandar, S.Kom., M.Kom.
NIP 197407052002121014

Mengetahui,
Ketua Jurusan

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

LEMBAR PENGAJUAN JUDUL LAPORAN AKHIR (LA)

Kepada Yth,
Dosen Pembimbing I
Muhammad Noval, S.E.,M.Si
Politeknik Negeri Sriwijaya

Dengan Hormat,

Dalam rangka memenuhi syarat penyusunan Laporan Kerja Praktik, maka kami :

Nama : Popi Nabilla
NIM : 062130800241
Jurusan / Prodi : Manajemen Informatika / DIII Manajemen Informatika

Bermaksud untuk mengajukan judul laporan kerja praktek. Adapun judul laporan kerja praktik yang kami ajukan, yaitu:

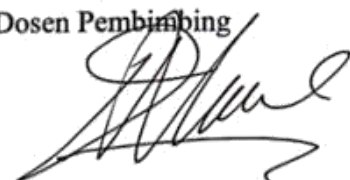
1. Aplikasi Pendaftaran Magang Pada Kantor Ombudsman RI Perwakilan Sumatera Selatan
2. Aplikasi Pengelolaan Data Peminjaman Barang Milik Negara (BMN) di Kantor Ombudsman RI Perwakilan Sumatera Selatan
3. Sistem Informasi Eksekutif Dalam Manajemen Laporan Masuk Pada Kantor Ombudsman RI Perwakilan Sumatera Selatan

Demikian surat pengajuan judul laporan kerja praktik ini dibuat. Sekiranya judul diatas dapat disetujui. Atas persetujuan Bapak kami ucapkan terima kasih.

Palembang, Maret 2024

Menyetujui,

Dosen Pembimbing


Muhammad Noval, S.E.,M.Si
NIP 197511082005011003

Mahasiswa


Popi Nabilla
NIM 062130800241





LEMBAR PENGAJUAN JUDUL LAPORAN AKHIR (LA)

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

Dengan Hormat,

Dalam rangka memenuhi syarat penyusunan Laporan Kerja Praktik, maka kami :

Nama : Popi Nabilla
NIM : 062130800241
Jurusan / Prodi : Manajemen Informatika / DIII Manajemen Informatika

Bermaksud untuk mengajukan judul laporan kerja praktek. Adapun judul laporan kerja praktik yang kami ajukan, yaitu:

1. Aplikasi Pendaftaran Magang Pada Kantor Ombudsman RI Perwakilan Sumatera Selatan
2. Aplikasi Pengelolaan Data Peminjaman Barang Milik Negara (BMN) di Kantor Ombudsman RI Perwakilan Sumatera Selatan
3. Sistem Informasi Eksekutif Dalam Manajemen Laporan Masuk Pada Kantor Ombudsman RI Perwakilan Sumatera Selatan

Demikian surat pengajuan judul laporan kerja praktik ini dibuat. Sekiranya judul diatas dapat disetujui. Atas persetujuan Bapak kami ucapkan terima kasih.

Palembang, Maret 2024

Menyetujui,
Dosen Pembimbing

Meivi Kusnandar, S.Kom.,M.Kom
NIP 197407052002121014

Mahasiswa

Popi Nabilla
NIM 062130800241





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 : Popi Nabilla
NIM : 062130800241
Jurusan : Manajemen Informatika
Program Studi : DIII Manajemen Informatika
Judul : Sistem Informasi Eksekutif Dalam Manajemen Laporan Masuk
Pada Kantor Ombudsman RI Perwakilan Sumatera Selatan

Palembang, 22 Maret 2024

Tim Pembimbing :

Dosen Pembimbing I

Muhammad Noval, SE., M.Si
NIP 197511082005011003

Dosen Pembimbing II

Meivi Kusnandar, S.Kom., M.Kom
NIP 197407052002121014

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

3151 27/3/24

PERMOHONAN SURAT PENGANTAR PENGAMBILAN DATA

Hal : Permohonan Surat Pengantar
Pengambilan Data

Palembang, 20 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 ditujukan kepada :

Yth. Kepala Perwakilan
Ombudsman RI Perwakilan Sumatera Selatan
Jl. Radio No.1 Kel.20 Ilir DIV, Kec. Ilir timur 1

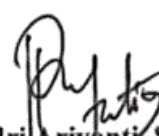
Untuk pengambilan data di Kantor Ombudsman RI Perwakilan Sumatera Selatan dalam pembuatan aplikasi yang berjudul "Sistem Informasi Eksekutif Dalam Manajemen Laporan Masuk Pada Kantor Ombudsman RI Perwakilan Sumatera Selatan".

Pengambilan data ini akan saya mulai pada tanggal 25 Maret 2024 s.d. 5 April 2024. Saya mahasiswa yang akan melakukan pengambilan data adalah :

| No | NAMA | NIM | KELAS | JURUSAN |
|----|--------------|--------------|-------|-----------------------|
| 1. | Popi Nabilla | 062130800241 | 6 IF | Manajemen Informatika |

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

Mengetahui,
Ketua Jurusan Manajemen Informatika


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

Hormat saya,



Popi Nabilla
NIM 062130800241





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara Bukit Besar – Palembang 30139
Telepon 0711-353414 Faksimili 0711-355918
Laman : <http://polsri.ac.id>, Pos El : info@polsri.ac.id

Nomor : 3151/PL6.3.1/SP/2024
Hal : Izin Pengambilan Data

25 Maret 2024

Yth. Kepala
Perwakilan Ombudsman Republik Indonesia
Perwakilan Sumatera Selatan
Jalan Radio No. 1 Kelurahan 20 Ilir D IV
Kecamatan Ilir Timur I
Palembang

Dengan hormat,

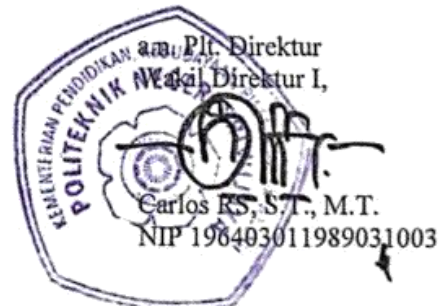
Sesuai denn kurikulum Jurusan Manajemen Informatika pada Politeknik Negeri Sriwijaya, Laporan Akhir merupakan mata kuliah wajib pada semester VI (enam). Untuk itu mahasiswa kami memerlukan data untuk penyusunan Laporan Akhir tersebut.

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

Mahasiswa kami yang akan mengumpulkan data tersebut adalah :

| No | Nama | NPM | Kelas | Jurusan/
Program Studi |
|----|--------------|----------------|-------|---------------------------|
| 1 | Popi Nabilla | 0621 3080 0241 | 6 IF | Manajemen Informatika |

Atas perhatian dan bantuannya diucapkan terima kasih.



Tembusan:

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

F:MS.Word/D/TM



Nomor : B/219/HM.02.03-07/IV/2024
Sifat : Biasa
Lampiran : -
Hal : Izin Pengambilan Data

04 April 2024

Yth. Wakil Direktur I Politeknik Negeri Sriwijaya
di Palembang

Berdasarkan Surat Wakil Direktur I Politeknik Negeri Sriwijaya Nomor:3151/PL6.3.1/SP/2024 Hal: Izin Pengambilan Data, Tanggal: 25 Maret 2024, dengan ini diberitahukan bahwa Kantor Perwakilan Ombudsman RI Provinsi Sumatera Selatan memberikan izin untuk melakukan kegiatan yang dimaksud yang dimaksud kepada mahasiswa dengan:

| No. | Nama | NIM | Progam Studi |
|-----|--------------|----------------|-----------------------|
| 1. | Popi Nabilla | 0621 3080 0241 | Manajemen Informatika |

Selama melaksanakan kegiatan tersebut, mahasiswa yang bersangkutan dapat berkoodinasi secara langsung dengan Sdri. Vidya Nirmala Sari selaku asisten di bidang Pencegahan Maladministrasi.

Demikian disampaikan, atas perhatian dan kerja sama yang baik diucapkan terima kasih.

Kepala Perwakilan,



M. Adrian Agustiansyah



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

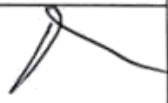
Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414

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

LEMBAR BIMBINGAN LAPORAN AKHIR

Lembar : 1

Nama : Popi Nabilla
 NIM : 062130800241
 Jurusan/Program Studi : Manajemen Informatika/DIII Manajemen Informatika
 Judul Laporan Akhir : Sistem Informasi Eksekutif Dalam Manajemen Laporan Masuk Pada Kantor Ombudsman RI Perwakilan Sumatera Selatan
 Pembimbing I : Muhammad Noval, SE.,M.Si

| No. | Tanggal | Uraian Bimbingan | Tanda Tangan Pembimbing |
|-----|-----------------|-------------------------|---|
| 1. | 20 / 2024
03 | ACC Judul LAPORAN AKHIR |  |
| 2. | 22 / 2024
03 | Pengajuan BAB I |  |
| 3. | 28 / 2024
03 | ACC BAB I |  |
| 4. | 02 / 2024
04 | Pengajuan BAB II |  |
| 5. | 25 / 2024
04 | ACC BAB II |  |
| 6. | 25 / 2024
04 | Perbaikan BAB III |  |

Lembar : 2

| No. | Tanggal | Uraian Bimbingan | Tanda Tangan Pembimbing |
|-----|------------|---------------------|---|
| 7. | 11-06-2024 | ACC BAB III |  |
| 8. | 04-07-2024 | Perbaikan BAB N-S V |  |
| 9. | 08-07-2024 | Perbaikan BAB N-S V |  |
| 10. | 09-07-2024 | ACC BAB N-S V |  |
| 11. | | | |
| 12. | | | |

Palembang,

2024

Ketua Jurusan,


 Dr. Indri Ariyanti, SE., M.Si
 NIP 197806032008012008
Catatan:

*) melingkari angka yang sesuai.

- Ketua Jurusan harus memeriksa jumlah pelaksanaan bimbingan sesuai yang dipersyaratkan dalam Pedoman Laporan Akhir sebelum menandatangani lembar bimbingan ini.
- Lembar pembimbingan LA 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

LEMBAR BIMBINGAN LAPORAN AKHIR

Lembar : 1

Nama : Popi Nabilla
 NIM : 062130800241
 Jurusan/Program Studi : Manajemen Informatika/DIII Manajemen Informatika
 Judul Laporan Akhir : Sistem Informasi Eksekutif Dalam Manajemen Laporan Masuk Pada Kantor Ombudsman RI Perwakilan Sumatera Selatan
 Pembimbing II : Meivi Kusnandar, S.Kom., M.Kom

| No. | Tanggal | Uraian Bimbingan | Tanda Tangan Pembimbing |
|-----|--------------|--|-------------------------|
| 1. | 20 / 03 2024 | det judul Capran akhir | |
| 2. | 29 / 3 2024 | pengajuan bab I, revisi pada bab I dan bebel | |
| 3. | 1 / 4 2024 | det bab I, lengkap bab II | |
| 4. | 2 / 4 2024 | pengajuan bab II, tambah referensi | |
| 5. | 4 / 4 2024 | tambahan kebagian bab I dan bab II | |
| 6. | 5 / 4 2024 | det bab II lengkap bab III | |

Lembar : 2

| No. | Tanggal | Uraian Bimbingan | Tanda Tangan Pembimbing |
|-----|---------------|---|-------------------------|
| 7. | 25/2023
/4 | pengantar sb II, peni di bgs fide
yg kepala & syten yg diuolue | |
| 8. | 2/5 2023 | akt sb II, langt sb II | |
| 9. | 12/6 2023 | pengantar sb II, peni jata fide | |
| 10. | 13/6 2024 | akt sb II, langt sb II | |
| 11. | 13/6 2024 | pengantar sb II dan tdk dapt mndr apkr. | |
| 12. | 20/6 2024 | akt sb II, Dekonasi, Sdy LX | |

Palembang,

2024

Ketua Jurusan,

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

*) melingkari angka yang sesuai.

- Ketua Jurusan harus memeriksa jumlah pelaksanaan bimbingan sesuai yang dipersyaratkan dalam Pedoman Laporan Akhir sebelum menandatangani lembar bimbingan ini.
- Lembar pembimbingan LA 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

REKOMENDASI UJIAN LAPORAN AKHIR (LA)

Pembimbing Laporan Akhir memberikan rekomendasi kepada,

Nama : Popi Nabilla
NPM : 062130800241
Jurusan/Program Studi : Manajemen Informatika/D3 Manajemen Informatika
Judul Laporan Akhir : Sistem Informasi Eksekutif Dalam Manajemen Laporan Masuk
Pada Kantor Ombudsman RI Perwakilan Sumatera Selatan

Mahasiswa tersebut telah memenuhi persyaratan dan dapat mengikuti Ujian Laporan Akhir (LA)
pada Tahun Akademik 2023/2024

Palembang, 13 Juni 2024

Pembimbing I,

Muhammad Noval, SE., M.Si
NIP.197511082005011003

Pembimbing II,

Meivi Kushandar, S.Kom., M.Kom
NIP. 197407052002121014

Mengetahui,
Ketua Jurusan Manajemen Informatika

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



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA

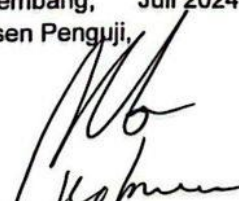
Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id>, Pos El : info@polsri.ac.id

REVISI UJIAN LAPORAN AKHIR (LA)

Ruang : 4
Dosen Penguji : Robinson, S.kom., M.kom.
Nama Mahasiswa : Popi Nabilla
NIM : 062130800241
Jurusan/Program Studi : Manajemen Informatika/DIII Manajemen Informatika
Judul Laporan Akhir : Sistem Informasi Eksekutif Dalam Manajemen Laporan Masuk
Pada Kantor Ombudsman RI Perwakilan Sumatera Selatan

| No | Uraian Revisi | Paraf |
|----|-------------------------------|-------|
| 1 | DRCM tabul
s174 rek sssvri | |

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

LEMBAR PELAKSANAAN REVISI LAPORAN AKHIR

Mahasiswa berikut,

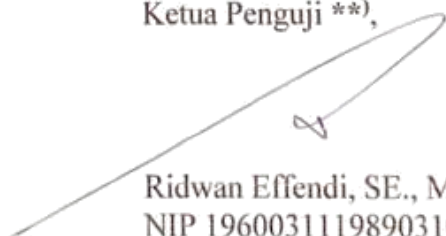
Nama : Popi Nabilla
NPM : 062130800241
Jurusan/Program Studi : Manajemen Informatika/DIII Manajemen Infotmatika
Judul Laporan Akhir : Sistem Informasi Eksekutif Dalam Manajemen Laporan Masuk
Pada Kantor Ombudsman RI Perwakilan Sumatera Selatan

Telah melaksanakan revisi terhadap Laporan Akhir yang diujikan pada hari Selasa tanggal 16 Bulan 07 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. | | Ridwan Effendi, SE., M.Si | 16-07-2024 |  |
| 2. | Desain Tabel
Size Tidak Sesuai | Robinson, S.Kom., M.Kom | 16-07-2024 |  |
| 3. | | Devi Sartika, S.Kom., MAB | 16-07-2024 |  |

Palembang, Juli 2024

Ketua Penguji **),


Ridwan Effendi, SE., M.Si
NIP 196003111989031005

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;