

LISTING PROGRAM

HomeController.php

```

<?php

namespace
App\Http\Controllers;

use Illuminate\Http\Request;
use App\Helpers\Helper;
use
Illuminate\Support\Facades\DB
;
use
PhpParser\Node\Stmt\TryCatch;

class HomeController extends
Controller
{
    /**
     * Create a new
     controller instance.
     *
     * @return void
     */
    public function
__construct()
    {
        $this->middleware('auth');
    }

    /**
     * Show the application
     dashboard.
     *
     * @return
     \Illuminate\Contracts\Support
     \Renderable
     */
    public function index()
    {
        $data = new
        \stdClass();
        return view('home');
    }
}

```

UserController.php

```

<?php

namespace
App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\User;
use App\Helpers\Helper;
use
PhpParser\Node\Stmt\TryCatch;
use
Illuminate\Support\Facades\Hash;
use
Illuminate\Support\Facades\Auth;

class UserController extends
Controller
{
    public function
__construct()
    {
        $this->middleware('auth');
    }

    public function index()
    {
        $user =
        User::whereNotIn('role',
        ['nasabah'])->orderBy('id',
        'DESC')->get();
        $dataTables = [];
        $no = 1;
        foreach ($user as
        $key => $item) {
            $deleteLink =
            route('user.destroy', $item->id);

            $dataTables[] = [
                $no++,
                $item->name,
                $item->email,

```

```

        $item->role,
        $item-
>status,
        Helper::btnEd
it('/user/' . $item->id .
'/edit'),
        ];
    }

    $data = new
\stdClass();
    $data->heads = [
        'No',
        'User Name',
        'Email',
        'Role',
        'Status',
        ['label' =>
'Actions', 'no-export' =>
true, 'width' => 15],
    ];

    $data->config = [
        'data' =>
$dataTables,
        'columnDefs' =>
[['targets' => "_all"]],
        'order' => [[0,
'asc']],
    ];

    return
view('pages.user.index',
compact('data'));
}

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

    public function
store(Request $request)
    {
        $request->validate([
            'name' =>
'required',
            'email' =>
'required',
            'password' =>
'required',
            'role' =>
'required',
            'status' =>
'required',
        ]);

        $data = [];
        $data['name'] =
$request->name;
        $data['role'] =
$request->role;
        $data['status'] =
$request->status;
        $data['email'] =
$request->email;
        $data['password']
= Hash::make($request-
>password);
        $store =
User::Create($data);
        $user_id = $store-
>id;

        return redirect()-
>route('user.index')-
>with('success', 'Data user
berhasil ditambahkan');
    }

    public function edit($id)
    {
        $data = new
\stdClass();
        $data->user =
user::find($id);
        if (!$data->user) {
            return
redirect()-
>route('user.index')-
>with('error', 'Data user
tidak ditemukan');
        }
    }

```

```

        return
view('pages.user.edit',
compact('data'));
    }

    public function
update(Request $request, $id)
    {
        $request->validate([
            'name' =>
'required',
            'email' =>
'required',
            'role' =>
'required',
            'status' =>
'required',
        ]);

        $data = [];
        $data['name'] =
$request->name;
        $data['role'] =
$request->role;
        $data['status'] =
$request->status;
        $data['email'] =
$request->email;
        if($request-
>password){
            $data['password']
= Hash::make($request-
>password);
        }
        $update =
User::findOrFail($id)-
>update($data);

        return redirect()-
>route('user.index')-
>with('success', 'Data user
berhasil diubah');
    }

    public function
destroy($id)
    {
        $delete =
user::find($id);

```

```

        try {
            $delete-
>delete();

            return
redirect()-
>route('user.index')-
>with('success', 'Data user
berhasil dihapus');
        } catch
(\Illuminate\Database\QueryEx
ception $e) {
            return
redirect()-
>route('user.index')-
>with('error', 'Data user
tidak dapat dihapus');
        }
    }

    public function
laporan_nasabah()
    {
        $user =
user::select('users.*',
'indoregion_villages.name as
village_name')-
>with('get_province', 'get_reg
ency', 'get_district', 'get_vil
lage')-
>leftJoin('indoregion_village
s', 'users.village_id',
'indoregion_villages.id')-
>where('role', 'nasabah')-
>orderBy('users.id', 'DESC')-
>get();

        if (Auth::user()-
>role == 'nasabah') {
            $user =
user::select('users.*',
'indoregion_villages.name as
village_name')-
>with('get_province', 'get_reg
ency', 'get_district', 'get_vil
lage')-
>leftJoin('indoregion_village
s', 'users.village_id',
'indoregion_villages.id')-
>where('users.id',
Auth::user()->id)-

```

```

>orderBy('users.id', 'DESC')-
>get();
    }
    $dataTables = [];
    $no = 1;
    foreach ($user as
$key => $item) {
        $deleteLink =
route('user.destroy', $item-
>id);

        $dataTables[] = [
            $no++,
            strtoupper($i
tem->name),
            strtoupper($i
tem->gender),
            $item-
>brith_date ?
Helper::tanggal_indo($item-
>brith_date) : '',
            $item-
>phone_number,
            $item-
>get_province->name,
            $item-
>get_regency->name,
            $item-
>get_district->name,
            $item-
>village_name,
            $item-
>address,
            Helper::tangg
al_indo($item->created_at),

```

RegionalController.php

```

<?php

namespace
App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\District;
use App\Models\Province;
use App\Models\Regency;
use App\Models\Village;
use App\Helpers\Helper;

```

```

    ];
    }

    $data = new
stdClass();
    $data->heads = [
        'No',
        'Nama',
        'Jenis Kelamin',
        'Tanggal Lahir',
        'No. Telepon',
        'Provinsi',
        'Kabupaten',
        'Kecamatan',
        'Kelurahan',
        'Alamat',
        'Tanggal Masuk',
    ];

    $data->config = [
        'data' =>
$dataTables,
        'columnDefs' =>
[['targets' => "_all"],
        'order' => [[0,
'asc']],
    ];

    return
view('pages.laporan.nasabah',
compact('data'));
    }
}

```

```

use
PhpParser\Node\Stmt\TryCatch;
use
Illuminate\Support\Facades\Ha
sh;
use
Illuminate\Support\Facades\Au
th;

class RegionalController
extends Controller
{

```

```

        public function
province()
    {
        return
Province::select('id', 'name
as text')->get();
    }
    public function
regency($id)
    {
        return
Regency::select('id', 'name
as text')-
>where('province_id',$id)-
>get();
    }
    public function
district($id)
    {
        return
District::select('id', 'name
as text')-
>where('regency_id',$id)-
>get();
    }
    public function
village($id)
    {
        return
Village::select('id', 'name
as text')-
>where('district_id',$id)-
>get();
    }
}

```

JenisKreditController.php

```

<?php

namespace
App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\JenisKredit;
use
App\Models\JaminanJenisKredit
;
use App\Helpers\Helper;

```

```

use
Illuminate\Support\Facades\Au
th;

class JenisKreditController
extends Controller
{
    public function
__construct()
    {
        $this-
>middleware('auth');
    }

    public function index()
    {

        $jenis_kredit =
JenisKredit::orderBy('jenis_k
redit_id', 'DESC')->get();
        $dataTables = [];
        $no = 1;
        foreach
($jenis_kredit as $key =>
$item) {
            $deleteLink =
route('jenis_kredit.destroy',
$item->jenis_kredit_id);
            $del =
Helper::btnDelete("deleteData
('$item->jenis_kredit_id',
'$deleteLink')");

            $action =
Helper::btnEdit('/jenis_kredi
t/' . $item->jenis_kredit_id
. '/edit');

            $dataTables[] = [
                $no++,
                strtoupper($i
tem->jenis_kredit_nama),
                Helper::rp($i
tem->jenis_kredit_min),
                Helper::rp($i
tem->jenis_kredit_max),
                $action,
            ];
        }
    }
}

```

```

        $data = new
\stdClass();
        $data->heads = [
            'No',
            'Nama',
            'Min',
            'Max',
        ];
        $data->heads[] =
['label' => 'Actions', 'no-
export' => true, 'width' =>
15];
        $data->config = [
            'data' =>
$dataTables,
            'columnDefs' =>
[['targets' => "_all"],
            'order' => [[0,
'asc']],
        ];

        return
view('pages.jenis_kredit.inde
x', compact('data'));
    }

    public function create()
    {
        return
view('pages.jenis_kredit.crea
te');
    }

    public function
store(Request $request)
    {
        // return $request-
>all();
        $request->validate([
            'jenis_kredit_nam
a' => 'required',
            'jenis_kredit_min
' => 'required',
            'jenis_kredit_max
' => 'required',
        ]);
    }

```

```

        $store =
JenisKredit::create($request-
>all());
        if(count($request-
>jaminan) > 0){
            foreach($request-
>jaminan as $item){
                $jaminan =
JaminanJenisKredit::create([
                    'jaminan_
nama' => $item,
                    'jaminan_
jenis_kredit_id' => $store-
>jenis_kredit_id,
                ]);
            }
        }

        return redirect()-
>route('jenis_kredit.index')-
>with('success', 'Data Jenis
Kredit berhasil
ditambahkan');
    }

    public function edit($id)
    {
        $data = new
\stdClass();
        $data->jenis_kredit =
JenisKredit::find($id);
        $data->jaminan =
JaminanJenisKredit::where('ja
minan_jenis_kredit_id',$id)-
>get();
        if (!$data-
>jenis_kredit) {
            return
redirect()-
>route('jenis_kredit.index')-
>with('error', 'Data Jenis
Kredit tidak ditemukan');
        }
        return
view('pages.jenis_kredit.edit
', compact('data'));
    }

```

```

    public function
update(Request $request, $id)
    {
        $this->validate($request, [
            'jenis_kredit_nama' => 'required',
            'jenis_kredit_min' => 'required',
            'jenis_kredit_max' => 'required',
        ]);

        $update =
JenisKredit::findorfail($id)->update($request->all());
        if(count($request->jaminan) > 0){
            $deleteNotIn =
JaminanJenisKredit::where('jaminan_jenis_kredit_id', $id)->whereNotIn('jaminan_nama', $request->jaminan)->delete();

            foreach($request->jaminan as $item){
                $cek =
JaminanJenisKredit::where('jaminan_jenis_kredit_id', $id)->where('jaminan_nama', $item)->first();
                if(!$cek){
                    $jaminan =
JaminanJenisKredit::create([
                        'jaminan_nama' => $item,
                        'jaminan_jenis_kredit_id' => $id,
                    ]);
                }
            }
            return redirect()->route('jenis_kredit.index')->with('success', 'Data Jenis Kredit berhasil diubah');
        }
    }

```

```

    public function
destroy($id)
    {
        $delete =
JenisKredit::find($id);
        try {
            $delete->delete();
            return
redirect()->route('jenis_kredit.index')->with('success', 'Data Jenis Kredit berhasil dihapus');
        } catch
(\Illuminate\Database\QueryException $e) {
            return
redirect()->route('jenis_kredit.index')->with('error', 'Data Jenis Kredit tidak dapat dihapus');
        }
    }

    public function laporan()
    {
        $jenis_kredit =
JenisKredit::orderBy('jenis_kredit_id', 'DESC')->get();
        $dataTables = [];
        $no = 1;
        foreach
($jenis_kredit as $key => $item) {
            $dataTables[] = [
                $no++,
                strtoupper($item->jenis_kredit_nama),
                Helper::rp($item->jenis_kredit_min),
                Helper::rp($item->jenis_kredit_max),
            ];
        }

        $data = new
stdClass();
    }

```

```

        $data->heads = [
            'No',
            'Nama',
            'Min',
            'Max',
        ];
        $data->config = [
            'data' =>
$dataTables,
            'columnDefs' =>
[['targets' => "_all"]],
            'order' => [[0,
'asc']],
        ];

        return
view('pages.laporan.jenis_kre
dit', compact('data'));
    }
}

```

BungaController.php

```

<?php

namespace
App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Bunga;
use App\Helpers\Helper;
use
Illuminate\Support\Facades\Au
th;

class BungaController extends
Controller
{
    public function
__construct()
    {
        $this->
middleware('auth');
    }

    public function index()
    {

```

```

        $bunga =
Bunga::orderBy('bunga_id',
'DESC')->get();
        $dataTables = [];
        $no = 1;
        foreach ($bunga as
$key => $item) {
            $deleteLink =
route('bunga.destroy', $item->
bunga_id);
            $del =
Helper::btnDelete("deleteData
('$item->bunga_id',
'$deleteLink')");

            $action =
Helper::btnEdit('/bunga/' .
$item->bunga_id . '/edit');

            $dataTables[] = [
                $no++,
                strtoupper($i
tem->bunga_jenis),
                $item->
>bunga_persentase,
                $action,
            ];
        }

        $data = new
stdClass();
        $data->heads = [
            'No',
            'Nama',
            'Persentase',
        ];
        $data->heads[] =
['label' => 'Actions', 'no-
export' => true, 'width' =>
15];
        $data->config = [
            'data' =>
$dataTables,
            'columnDefs' =>
[['targets' => "_all"]],
            'order' => [[0,
'asc']],
        ];

```



```

        return
view('pages.bunga.index',
compact('data'));
    }

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

    public function
store(Request $request)
    {
        $request->validate([
            'bunga_jenis' =>
'required',
            'bunga_persentase
' => 'required',
        ]);

        $store =
Bunga::create($request-
>all());
        $store->save();

        return redirect()-
>route('bunga.index')-
>with('success', 'Data Bunga
berhasil ditambahkan');
    }

    public function edit($id)
    {
        $data = new
stdClass();
        $data->bunga =
Bunga::find($id);
        if (!$data->bunga) {
            return
redirect()-
>route('bunga.index')-
>with('error', 'Data Bunga
tidak ditemukan');
        }
        return
view('pages.bunga.edit',
compact('data'));
    }

```

```

    public function
update(Request $request, $id)
    {
        $this-
>validate($request, [
            'bunga_jenis' =>
'required',
            'bunga_persentase
' => 'required',
        ]);

        $update =
Bunga::findorfail($id)-
>update($request->all());
        return redirect()-
>route('bunga.index')-
>with('success', 'Data Bunga
berhasil diubah');
    }

    public function
destroy($id)
    {
        $delete =
Bunga::find($id);
        try {
            $delete-
>delete();
            return
redirect()-
>route('bunga.index')-
>with('success', 'Data Bunga
berhasil dihapus');
        } catch
(\\Illuminate\\Database\\QueryEx
ception $e) {
            return
redirect()-
>route('bunga.index')-
>with('error', 'Data Bunga
tidak dapat dihapus');
        }
    }

    public function laporan()
    {

```

```

        $bunga =
Bunga::orderBy('bunga_id',
'DESC')->get();
        $dataTables = [];
        $no = 1;
        foreach ($bunga as
$key => $item) {

            $dataTables[] = [
                $no++,
                strtoupper($i
tem->bunga_jenis),
                $item-
>bunga_persentase,
            ];
        }

        $data = new
\stdClass();
        $data->heads = [
            'No',
            'Nama',
            'Persentase',
        ];
        $data->config = [
            'data' =>
$dataTables,
            'columnDefs' =>
[['targets' => "_all"],
            'order' => [[0,
'asc']],
        ];

        return
view('pages.laporan.bunga',
compact('data'));
    }
}

```

PinjamanController.php

```

<?php

namespace
App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Pengajuan;
use App\Models\Pencairan;
use App\Models\Angsuran;

```

```

use App\Models\JenisKredit;
use
App\Models\JaminanJenisKredit
;
use
App\Models\JaminanPengajuan;
use App\Models\Bunga;
use App\Models\User;
use App\Helpers\Helper;
use Carbon\Carbon;
use
Illuminate\Support\Facades\DB
;
use
Illuminate\Support\Facades\Au
th;

```

```

class PinjamanController
extends Controller
{
    public function
__construct()
    {
        $this-
>middleware('auth');
    }

    public function index()
    {

        $pengajuan =
Pengajuan::with('get_nasabah'
,'get_jenis_kredit','get_bung
a')->orderBy('pengajuan_id',
'DESC')->get();
        if (Auth::user()-
>role == 'nasabah') {
            $pengajuan =
Pengajuan::with('get_nasabah'
,'get_jenis_kredit','get_bung
a')-
>where('pengajuan_nasabah_id'
, Auth::user()->id)-
>orderBy('pengajuan_id',
'DESC')->get();
        }
        $dataTables = [];
        $no = 1;

```

```

        foreach ($pengajuan
as $key => $item) {
            $deleteLink =
route('pinjaman.destroy',
$item->pengajuan_id);
            $detail =
Helper::btnDetails('/pinjaman
/' . $item->pengajuan_id .
'/detail');
            if ($item-
>pengajuan_status ==
"diajukan") {
                $edit = '';
                $del = '';
                // $edit =
Helper::btnEdit('/pinjaman/
. $item->pengajuan_id .
'/edit');
                // $del =
Helper::btnDelete("deleteData
('$item->pengajuan_id',
'$deleteLink')");
            } else {
                $edit = '';
                $del = '';
            }

            $action =
$detail;
            if (
Auth::user()->role ==
'karyawan') {
                $action =
$edit.$del.$detail;
            }

            $dataTables[] = [
                $no++,
                strtoupper($i
tem->get_nasabah->name),
                strtoupper($i
tem->get_jenis_kredit-
>jenis_kredit_nama),
                strtoupper($i
tem->get_bunga-
>bunga_jenis).' '.$item-
>get_bunga-
>bunga_persentase.'%',
                Helper::tangg
al_indo($item-
>pengajuan_tanggal),
                Helper::rp($i
tem->pengajuan_nominal),
                $item-
>pengajuan_angsuran,
                strtoupper($i
tem->pengajuan_status),
                $action
            ];
        }

        $data = new
\stdClass();
        $data->heads = [
            'No',
            'Nasabah',
            'Jenis Kredit',
            'Bunga',
            'Tanggal
Pengajuan',
            'Nominal',
            'Angsuran',
            'Status',
        ];
        $data->heads[] =
['label' => 'Actions', 'no-
export' => true, 'width' =>
15];
        $data->config = [
            'data' =>
$dataTables,
            'columnDefs' =>
[['targets' => "_all"],
            'order' => [[0,
'asc']],
        ];

        return
view('pages.pinjaman.index',
compact('data'));
    }

    public function create()
    {
        $data = new
\stdClass();

```

```

        $data->jenis_kredit =
JenisKredit::all();
        $data->bunga =
Bunga::all();
        $data->nasabah =
User::where('role',
'nasabah')->get();
        if (Auth::user()-
>role == 'nasabah') {
            $data->nasabah =
User::where('role',
'nasabah')->where('id',
Auth::user()->id)->get();
        }
        $data->tgl =
Helper::tanggal_indo(now());
        return
view('pages.pinjaman.create',
compact('data'));
    }

    public function
list_jaminan($id)
    {
        return
JaminanJenisKredit::select('j
aminan_id as id',
'jaminan_nama as text')-
>where('jaminan_jenis_kredit_
id',$id)->get();
    }

    public function
store(Request $request)
    {
        $request->validate([
            'pengajuan_tangga
l' => 'required',
            'pengajuan_nomina
l' => 'required',
            'pengajuan_angsur
an' => 'required',
            'pengajuan_nasaba
h_id' => 'required',
            'pengajuan_jenis_
kredit_id' => 'required',
            'pengajuan_bunga_
id' => 'required',
        ]);
    }

        $store =
Pengajuan::create($request-
>all());
        if(count($request-
>jaminan) > 0){
            foreach($request-
>jaminan as $item){
                $jaminan =
JaminanPengajuan::create([
                    'jaminan_
id' => $item,
                    'pengajua
n_id' => $store-
>pengajuan_id,
                ]);
            }
        }

        return redirect()-
>route('pinjaman.index')-
>with('success', 'Data
Pengajuan Pinjaman berhasil
ditambahkan');
    }

    public function edit($id)
    {
        $data = new
stdClass();
        $data->jenis_kredit =
JenisKredit::all();
        $data->bunga =
Bunga::all();
        $data->pengajuan =
Pengajuan::find($id);
        $data->nasabah =
User::where('role',
'nasabah')->get();
        if (Auth::user()-
>role == 'nasabah') {
            $data->nasabah =
User::where('role',
'nasabah')->where('id',
Auth::user()->id)->get();
        }
        if (!$data-
>pengajuan) {

```

```

        return
    redirect()-
>route('pinjaman.index')-
>with('error', 'Data
Pengajuan Pinjaman tidak
ditemukan');
    }
    $data->tgl =
    Helper::tanggal_indo($data-
>pengajuan-
>pengajuan_tanggal);
    return
    view('pages.pinjaman.edit',
compact('data'));
    }

    public function
update(Request $request, $id)
    {
        $this-
>validate($request, [
            'pengajuan_tangga
l' => 'required',
            'pengajuan_nomina
l' => 'required',
            'pengajuan_angsur
an' => 'required',
            'pengajuan_jamina
n' => 'required',
            'pengajuan_nasaba
h_id' => 'required',
            'pengajuan_jenis_
kredit_id' => 'required',
            'pengajuan_bunga_
id' => 'required',
        ]);

        $update =
    Pengajuan::findorfail($id)-
>update($request->all());
        return redirect()-
>route('pinjaman.index')-
>with('success', 'Data
Pengajuan Pinjaman berhasil
diubah');
    }

    public function
destroy($id)

```

```

    {
        $delete =
    Pengajuan::find($id);
        try {
            $delete-
>delete();
            return
            redirect()-
>route('pinjaman.index')-
>with('success', 'Data
Pengajuan Pinjaman berhasil
dihapus');
        } catch
        (\Illuminate\Database\QueryEx
ception $e) {
            return
            redirect()-
>route('pinjaman.index')-
>with('error', 'Data
Pengajuan Pinjaman tidak
dapat dihapus');
        }

        public function
detail($id)
        {
            $data = new
            \stdClass();
            $data->jenis_kredit =
            JenisKredit::all();
            $data->bunga =
            Bunga::all();
            $data->pengajuan =
            Pengajuan::with('get_nasabah',
            'get_bunga', 'get_jenis_kredi
            t')->find($id);
            $data->jaminan =
            JaminanPengajuan::with('get_j
            aminan')-
            >where('pengajuan_id', $id)-
            >get();
            $data->nasabah =
            User::all();
            if (!$data-
            >pengajuan) {
                return
                redirect()-
                >route('pinjaman.index')-
            }
        }
    }

```

```

>with('error', 'Data Pinjaman
tidak ditemukan');
    }
    $data->tgl =
Helper::tanggal_indo($data-
>pengajuan-
>pengajuan_tanggal);
    $data->tgl_now =
Helper::tanggal_indo(now());
    $data->nominal =
Helper::rp($data->pengajuan-
>pengajuan_nominal);
    if($data->pengajuan-
>pengajuan_status ==
"dicairkan" || $data-
>pengajuan->pengajuan_status
== "diangsur" || $data-
>pengajuan->pengajuan_status
== "lunas" ){
        $data->pencairan
=
Pencairan::where('pencairan_p
engajuan_id', $id)->first();
        $angsuran =
Angsuran::where("angsuran_pen
cairan_id", $data->pencairan-
>pencairan_id)-
>orderBy('angsuran_ke',
'ASC')->get();
        $dataTables = [];
        $no = 1;

        foreach
($angsuran as $key => $item)
{
            $tglIndo =
""$.Helper::tanggal_indo($ite
m->angsuran_jatuh_tempo)."";
            $tgljatuhTemp
o = ""$. $item-
>angsuran_jatuh_tempo."";
            $detail =
'<button onclick="angsuran('
. $item->angsuran_id . ', '
$item->angsuran_total . ', '
. $tgljatuhTempo . ',
'$. $tglIndo.')" class="btn
btn-xs btn-success text-white

```

```

mx-1 shadow" title="Bayar
Angsuran">
        <i class="fa
fa-lg fa-fw fa-money-
bill"></i>
        </button>';

        if($item-
>angsuran_status == 'lunas'){
            $detail =
'<button disabled class="btn
btn-xs btn-success text-white
mx-1 shadow" title="Bayar
Angsuran">
                <i
class="fa fa-lg fa-fw fa-
money-bill"></i>
                </button>';
            }

            $set_row = [
                $no++,
                $item-
>angsuran_ke,
                Helper::t
anggal_indo($item-
>angsuran_jatuh_tempo),
                Helper::r
p($item->angsuran_pokok),
                Helper::r
p($item->angsuran_bunga),
                Helper::r
p($item->angsuran_denda),
                Helper::r
p($item->angsuran_total),
                strtoupe
r($item->angsuran_status),
                ($item-
>angsuran_status ==
'tagihan') ? '-' :
Helper::tanggal_indo($item-
>angsuran_tanggal)
            ];
            if(Auth::user
()->role == 'karyawan'){
                $set_row
= [
                    $no++
,

```

```

        $item
    }
    ->angsuran_ke,
        Helpe
    r::tanggal_indo($item-
    >angsuran_jatuh_tempo),
        Helpe
    r::rp($item->angsuran_pokok),
        Helpe
    r::rp($item->angsuran_bunga),
        Helpe
    r::rp($item->angsuran_denda),
        Helpe
    r::rp($item->angsuran_total),
        strt
    upper($item-
    >angsuran_status),
        ($ite
    m->angsuran_status ==
    'tagihan') ? '-' :
    Helper::tanggal_indo($item-
    >angsuran_tanggal),
        $deta
    il
        ];
    }
    $dataTables[]
    = $set_row;
    $data->heads
    = [
        'No',
        'Ke',
        'Jatuh
    Tempo',
        'Pokok',
        'Bunga',
        'Denda',
        'Total',
        'Status',
        'Tanggal
    Bayar',
    ];
    if(Auth::user
    (->role == 'karyawan'){
        $data-
    >heads[] = ['label' =>
    'Actions', 'no-export' =>
    true, 'width' => 15];
    }
    }
    $data->config
    = [
        'data' =>
    $dataTables,
        'columnDe
    fs' => [['targets' =>
    "_all"]],
        'order'
    => [[0, 'asc']],
    ];
    }
    return
    view('pages.pinjaman.detail',
    compact('data'));
    }
    public function
    disetujui(Request $request,
    $id)
    {
        $update =
    Pengajuan::findorfail($id)-
    >update(['pengajuan_status'
    => 'disetujui']);
        return redirect()-
    >route('pinjaman.detail',[$id
    ])->with('success', 'Data
    Pinjaman berhasil
    disetujui');
    }
    public function
    ditolak(Request $request,
    $id)
    {
        $this-
    >validate($request, [
        'pengajuan_reason
    s' => 'required',
        'pengajuan_status
    ' => 'required',
    ]);
    }

```

```

        $update =
Pengajuan::findorfail($id)-
>update($request->all());
        return redirect()-
>route('pinjaman.detail',[$id
])->with('success', 'Data
Pinjaman ditolak');
    }

    public function
pencairan(Request $request,
$id)
    {
        $this-
>validate($request, [
            'pencairan_tgl'
=> 'required',
            'pencairan_total'
=> 'required',
        ]);

        try {
            DB::beginTransact
ion();
            $get_pengajuan =
Pengajuan::with('get_nasabah'
, 'get_bunga',
'get_jenis_kredit')-
>find($id);

            $pencairan = [];
            $pencairan['penca
iran_tgl'] = $request-
>pencairan_tgl;
            $pencairan['penca
iran_total'] = $request-
>pencairan_total;
            $pencairan['penca
iran_status'] = $request-
>pencairan_status;
            $pencairan['penca
iran_pengajuan_id'] =
$request-
>pencairan_pengajuan_id;
            $store =
Pencairan::create($pencairan)
;

            if
($get_pengajuan->get_bunga-
>bunga_jenis == "flat") {
                $angsuran_lis
t =
Helper::metode_flat($get_peng
ajuan->pengajuan_nominal,
$get_pengajuan-
>pengajuan_angsuran,
$get_pengajuan->get_bunga-
>bunga_persentase);
            } elseif
($get_pengajuan->get_bunga-
>bunga_jenis == "anuitas") {
                $angsuran_lis
t =
Helper::metode_anuitas($get_p
engajuan->pengajuan_nominal,
$get_pengajuan-
>pengajuan_angsuran,
$get_pengajuan->get_bunga-
>bunga_persentase);
            } elseif
($get_pengajuan->get_bunga-
>bunga_jenis == "efektif") {
                $angsuran_lis
t =
Helper::metode_efektif($get_p
engajuan->pengajuan_nominal,
$get_pengajuan-
>pengajuan_angsuran,
$get_pengajuan->get_bunga-
>bunga_persentase);
            }

            $tgl_jatuh_tempo
= Carbon::parse($request-
>pencairan_tgl)->addMonth()-
>format('Y-m-d');
            \Carbon\Carbon::s
etLocale('id');
            foreach
($angsuran_list as $item) {
                $angsuran =
[];
                $angsuran['an
gsuran_ke'] = $item['no'];

```



```

        $angsuran['an
gsuran_pokok'] =
$item['pokok'];
        $angsuran['an
gsuran_bunga'] =
$item['bunga'];
        $angsuran['an
gsuran_denda'] = 0;
        $angsuran['an
gsuran_total'] =
$item['jumlah_angsuran'];
        $angsuran['an
gsuran_jatuh_tempo'] =
$tgl_jatuh_tempo;
        $angsuran['an
gsuran_status'] = 'tagihan';
        $angsuran['an
gsuran_pencairan_id'] =
$store->pencairan_id;
        $store_angsur
an =
Angsuran::create($angsuran);
        $tgl_jatuh_te
mpo =
Carbon::parse($tgl_jatuh_temp
o)->addMonth()->format('Y-m-
d');
    }

    $pengajuan_update
= Pengajuan::findorfail($id)-
>update(['pengajuan_status'
=> 'dicairkan']);
    DB::commit();

    return
redirect()-
>route('pinjaman.detail',
[$id])->with('success', 'Data
Pinjaman dicairkan');
    } catch (exception
$e) {
        DB::rollBack();
        return
redirect()-
>route('pinjaman.detail',
[$id])->with('error', 'Gagal
Melakukan Pencairan');
    }
}

}

public function
angsuran(Request $request)
{
    try {
        DB::beginTransact
ion();
        $angsuran = [];
        $angsuran['angsur
an_denda'] = $request-
>angsuran_denda;
        $angsuran['angsur
an_total'] = $request-
>angsuran_total;
        $angsuran['angsur
an_tanggal'] = $request-
>angsuran_tanggal;
        $angsuran['angsur
an_status'] = 'lunas';
        $update =
Angsuran::findorfail($request
->angsuran_id)-
>update($angsuran);

        $count_angsuran =
Angsuran::where('angsuran_pen
cairan_id', $request-
>angsuran_pencairan_id)-
>where('angsuran_status',
'tagihan')->count();
        if($count_angsura
n < 1){
            $update_penga
juan =
Pengajuan::where('pengajuan_i
d', $request-
>angsuran_pengajuan_id)-
>update(['pengajuan_status'
=> 'lunas']);
        }else{
            $update_penga
juan =
Pengajuan::where('pengajuan_i
d', $request-
>angsuran_pengajuan_id)-
>update(['pengajuan_status'
=> 'diangsur']);

```

```

        }
        DB::commit();
        return
    }
    redirect()-
    >route('pinjaman.detail',
    [$request-
    >angsuran_pengajuan_id])-
    >with('success', 'Berhasil
    Melakukan Pembayaran
    Angsuran');
    } catch (exception
    $e) {
        DB::rollBack();
        return
    }
    redirect()-
    >route('pinjaman.detail',
    [$request-
    >angsuran_pengajuan_id])-
    >with('error', 'Gagal
    Melakukan Pembayaran
    Angsuran');
    }
}

```

LaporanController.php

```

<?php

namespace
App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Pengajuan;
use App\Models\Pencairan;
use App\Models\Angsuran;
use App\Models\JenisKredit;
use
App\Models\JaminanJenisKredit
;
use
App\Models\JaminanPengajuan;
use App\Models\Bunga;
use App\Models\User;
use App\Helpers\Helper;
use Carbon\Carbon;
use
Illuminate\Support\Facades\DB
;

```

```

use
Illuminate\Support\Facades\Auth;

class LaporanController
extends Controller
{
    public function
    __construct()
    {
        $this-
    >middleware('auth');
    }

    public function
    pengajuan(Request $request)
    {
        $data = new
        \stdClass();
        $model_pengajuan =
        Pengajuan::with('get_nasabah'
        , 'get_jenis_kredit', 'get_bunga')-
        >orderBy('pengajuan_id',
        'DESC');
        if (Auth::user()-
    >role == 'nasabah') {
            $model_pengajuan
        =
        Pengajuan::with('get_nasabah'
        , 'get_jenis_kredit', 'get_bunga')-
        >where('pengajuan_nasabah_id'
        , Auth::user()->id)-
        >orderBy('pengajuan_id',
        'DESC');
        }
        if($request->status){
            $data->status =
            $request->status;
            $model_pengajuan-
        >where('pengajuan_status',
            $request->status);
        }else{
            $data->status =
            "";
            $model_pengajuan-
        >whereIn('pengajuan_status',
            ['diajukan', 'disetujui',
            'ditolak']);
        }
    }
}

```

```

    }
    $data->date = "";

    $dates = [];
    if($request->date){
        $explode_date =
explode(" - ", $request-
>date);
        foreach($explode_
date as $item){
            $format =
Carbon::createFromFormat('m/d
/Y', $item)->format('Y-m-d');
            array_push($d
ates, $format);
        }
        $data->date =
$request->date;
    }else{
        array_push($dates
, Carbon::now()-
>startOfMonth()->format('Y-m-
d'));
        array_push($dates
, Carbon::now()-
>endOfMonth()->format('Y-m-
d'));
        $data->date =
Carbon::now()-
>startOfMonth()-
>format('m/d/Y').' -
'.Carbon::now()-
>endOfMonth()-
>format('m/d/Y');
    }
    $model_pengajuan-
>whereBetween('pengajuan_tang
gal', $dates);

    $data->periode = '';
    if($dates[0] ==
$dates[1]){
        $data->periode =
Helper::tanggal_indo($dates[0
]);
    }else{
        $data->periode =
'Periode

'.Helper::tanggal_indo($dates
[0]).' s/d
'.Helper::tanggal_indo($dates
[1]);
    }

    $pengajuan =
$model_pengajuan->get();
    $dataTables = [];
    $no = 1;
    foreach ($pengajuan
as $key => $item) {

        $dataTables[] = [
            $no++,
            strtoupper($i
tem->get_nasabah->name),
            strtoupper($i
tem->get_jenis_kredit-
>jenis_kredit_nama),
            strtoupper($i
tem->get_bunga-
>bunga_jenis).' '.$item-
>get_bunga-
>bunga_persentase.'%',
            Helper::tangg
al_indo($item-
>pengajuan_tanggal),
            Helper::rp($i
tem->pengajuan_nominal),
            $item-
>pengajuan_angsuran,
            strtoupper($i
tem->pengajuan_status),
            $item-
>pengajuan_nominal
        ];
    }

    $data->heads = [
        'No',
        'Nasabah',
        'Jenis Kredit',
        'Bunga',
        'Tanggal
Pengajuan',
        'Nominal',
        'Angsuran',
        'Status',

```

```

];
$data->config = [
    'data' =>
$dataTables,
    'columnDefs' =>
[['targets' => "_all"],
    'order' => [[0,
'asc']],
];
return
view('pages.laporan.pengajuan
', compact('data'));
}

public function
pencairan(Request $request)
{
    $data = new
    \stdClass();
    $model_pengajuan =
Pengajuan::with('get_nasabah'
, 'get_jenis_kredit', 'get_bung
a')-
>join("pencairan", "pengajuan_
id",
"pencairan_pengajuan_id")-
>orderBy('pengajuan_id',
'DESC');
    if (Auth::user()-
>role == 'nasabah') {
        $model_pengajuan
=
Pengajuan::with('get_nasabah'
, 'get_jenis_kredit', 'get_bung
a')-
>join("pencairan", "pengajuan_
id",
"pencairan_pengajuan_id")-
>where('pengajuan_nasabah_id'
, Auth::user()->id)-
>orderBy('pengajuan_id',
'DESC');
    }
    if($request->status){
        $data->status =
$request->status;

        $model_pengajuan-
>where('pengajuan_status',
$request->status);
    }else{
        $data->status =
"";
        $model_pengajuan-
>whereIn('pengajuan_status',
['dicairkan', 'diangsur',
'lunas']);
    }
    $data->date = "";

    $dates = [];
    if($request->date){
        $explode_date =
explode(" - ", $request-
>date);
        foreach($explode_
date as $item){
            $format =
Carbon::createFromFormat('m/d
/Y', $item)->format('Y-m-d');
            array_push($d
ates, $format);
        }
        $data->date =
$request->date;
    }else{
        array_push($dates
, Carbon::now()-
>startOfMonth()->format('Y-m-
d'));
        array_push($dates
, Carbon::now()-
>endOfMonth()->format('Y-m-
d'));
        $data->date =
Carbon::now()-
>startOfMonth()-
>format('m/d/Y').' -
'.Carbon::now()-
>endOfMonth()-
>format('m/d/Y');
    }
    $model_pengajuan-
>whereBetween('pencairan_tgl'
, $dates);

```

```

        $data->periode = '';
        if($dates[0] ==
$dates[1]){
            $data->periode =
Helper::tanggal_indo($dates[0
]);
        }else{
            $data->periode =
'Periode
'.Helper::tanggal_indo($dates
[0]).' s/d
'.Helper::tanggal_indo($dates
[1]);
        }

        $pengajuan =
$model_pengajuan->get();
        $dataTables = [];
        $no = 1;
        foreach ($pengajuan
as $key => $item) {

            $dataTables[] = [
                $no++,
                strtoupper($i
tem->get_nasabah->name),
                strtoupper($i
tem->get_jenis_kredit-
>jenis_kredit_nama),
                strtoupper($i
tem->get_bunga-
>bunga_jenis).' '.$item-
>get_bunga-
>bunga_persentase.'%',
                Helper::tangg
al_indo($item-
>pencairan_tgl),
                Helper::rp($i
tem->pengajuan_nominal),
                $item-
>pengajuan_angsuran,
                strtoupper($i
tem->pengajuan_status),
                $item-
>pengajuan_nominal
            ];
        }

        $data->heads = [
            'No',
            'Nasabah',
            'Jenis Kredit',
            'Bunga',
            'Tanggal
Pencairan',
            'Nominal',
            'Angsuran',
            'Status',
        ];
        $data->config = [
            'data' =>
$dataTables,
            'columnDefs' =>
[[['targets' => "_all"],
            'order' => [[0,
            'asc']],
        ];

        return
view('pages.laporan.pencairan
', compact('data'));
    }

    public function
angsuran(Request $request)
    {
        $data = new
stdClass();
        $model_angsuran =
Angsuran::join("pencairan","a
ngsuran_pencairan_id",
"pencairan_id")-
>orderBy('angsuran_id',
'DESC');
        if($request->status){
            $data->status =
$request->status;
            $model_angsuran-
>where('angsuran_status',
$request->status);
        }else{
            $data->status =
"";
            $model_angsuran-
>whereIn('angsuran_status',
['tagihan', 'lunas']);
        }
    }

```

```

        $data->date = "";

        $enddate =
Carbon::now()->endOfMonth()-
>format('m/d/Y');
        $startdate =
Carbon::now()-
>startOfMonth()-
>format('m/d/Y');
        $coldate =
$startdate.' - '.$enddate;

        if($request->date){
            $explode_date =
explode(" - ", $request-
>date);
            $data->date =
$request->date;
        }else{
            $explode_date =
explode(" - ", $coldate);
            $data->date =
$coldate;
        }

        $dates = [];
        foreach($explode_date
as $item){
            $format =
Carbon::createFromFormat('m/d
/Y', $item)->format('Y-m-d');
            array_push($dates
, $format);
        }
        $model_angsuran-
>whereBetween('angsuran_jatuh
_tempo', $dates);

        $data->periode = '';
        if($dates[0] ==
$dates[1]){
            $data->periode =
Helper::tanggal_indo($dates[0
]);
        }else{
            $data->periode =
'Periode
'.Helper::tanggal_indo($dates
[0]).' s/d
'.Helper::tanggal_indo($dates
[1]);
        }

        $angsuran =
$model_angsuran->get();
        $dataTables = [];
        $no = 1;
        foreach ($angsuran as
$key => $item) {
            $set_pengajuan =
Pengajuan::with('get_nasabah'
, 'get_jenis_kredit', 'get_bunga
a')->where("pengajuan_id",
$item-
>pencairan_pengajuan_id)-
>first();

            $dataTables[] = [
                $no++,
                strtoupper($s
et_pengajuan->get_nasabah-
>name),
                strtoupper($s
et_pengajuan-
>get_jenis_kredit-
>jenis_kredit_nama),
                strtoupper($s
et_pengajuan->get_bunga-
>bunga_jenis).',
                '$set_pengajuan->get_bunga-
>bunga_persentase.'%',
                Helper::tangg
al_indo($item-
>angsuran_jatuh_tempo),
                Helper::rp($i
tem->angsuran_total),
                strtoupper($i
tem->angsuran_status),
                $item-
>angsuran_total
            ];
        }

        $data->heads = [
            'No',
            'Nasabah',
            'Jenis Kredit',
            'Bunga',

```

```

        'Tanggal Jatuh
Tempo',
        'Nominal',
        'Status',
    ];
    $data->config = [
        'data' =>
$dataTables,
        'columnDefs' =>
[['targets' => "_all"]],
        'order' => [[0,
'asc']],
    ];

    return
view('pages.laporan.angsuran'
, compact('data'));
    }

    public function
kartu_piutang(Request
$request)
    {
        $data = new
\stdClass();
        $model_pengajuan =
Pengajuan::with('get_nasabah'
,'get_jenis_kredit','get_bung
a')-
>join("pencairan","pengajuan_
id",
"pencairan_pengajuan_id")-
>orderBy('pengajuan_id',
'DESC');
        if (Auth::user()-
>role == 'nasabah') {
            $model_pengajuan
=
Pengajuan::with('get_nasabah'
,'get_jenis_kredit','get_bung
a')-
>join("pencairan","pengajuan_
id",
"pencairan_pengajuan_id")-
>where('pengajuan_nasabah_id'
, Auth::user()->id)-
>orderBy('pengajuan_id',
'DESC');
        }

        $data->date = "";

        $dates = [];
        if($request->date){
            $explode_date =
explode(" - ", $request-
>date);
            foreach($explode_
date as $item){
                $format =
Carbon::createFromFormat('m/d
/Y', $item)->format('Y-m-d');
                array_push($d
ates, $format);
            }
            $data->date =
$request->date;
        }else{
            array_push($dates
, Carbon::now()-
>startOfMonth()->format('Y-m-
d'));
            array_push($dates
, Carbon::now()-
>endOfMonth()->format('Y-m-
d'));
            $data->date =
Carbon::now()-
>startOfMonth()-
>format('m/d/Y').' -
'.Carbon::now()-
>endOfMonth()-
>format('m/d/Y');
        }
        $model_pengajuan-
>whereBetween('pencairan_tgl'
, $dates);

        $data->periode = '';
        if($dates[0] ==
$dates[1]){
            $data->periode =
Helper::tanggal_indo($dates[0
]);
        }else{
            $data->periode =
'Periode

```

```

'.Helper::tanggal_indo($dates
[0]).' s/d
'.Helper::tanggal_indo($dates
[1]);
    }

    $pengajuan =
$model_pengajuan->get();
    $dataTables = [];
    $no = 1;
    foreach ($pengajuan
as $key => $item) {
        $detail_angsuran
=
Angsuran::where('angsuran_pen
cairan_id', $item-
>pencairan_id)->get();
        $item-
>total_tagihan = 0;
        $item-
>total_angsuran = 0;
        $item-
>sisangsuran = 0;
        foreach
($detail_angsuran as $key =>
$angsuran) {
            $angsuran_bul
anan = $angsuran-
>angsuran_pokok + $angsuran-
>angsuran_bunga;
            $item-
>total_tagihan = $item-
>total_tagihan +
$angsuran_bulanan;
            if($angsuran-
>angsuran_status !=
'tagihan'){
                $item-
>total_angsuran = $item-
>total_angsuran +
$angsuran_bulanan;
            }
        }
        $item-
>sisangsuran = $item-
>total_tagihan - $item-
>total_angsuran;
    }

    $data->heads = [
        'No',
        'Nasabah',
        'Tanggal
Pencairan',
        'Jenis Kredit',
        'Bunga',
        'Total Tagihan',
        'Total Angsuran',
        'Sisa Angsuran',
    ];
    $data->row =
    $pengajuan;
    return
view('pages.laporan.kartu_piu
tang', compact('data'));
}

    public function
detail_kartu_piutang($id)
    {

        $data = new
\stdClass();
        $data->pinjaman =
$id;
        $data->pengajuan =
Pengajuan::with('get_nasabah'
,'get_bunga','get_jenis_kredi
t')->find($id);
        $data->jaminan =
JaminanPengajuan::with('get_j
aminan')-
>where('pengajuan_id', $id)-
>get();
        $data->list_pinjaman
=
Pengajuan::with('get_nasabah'
,'get_jenis_kredit','get_bung
a')-
>whereIn('pengajuan_status',
['dicairkan', 'diangsur',
'lunas'])-
>orderBy('pengajuan_id',
'DESC')->get();
        if (Auth::user()-
>role == 'nasabah') {
            $data-
>list_pinjaman =

```



```

Pengajuan::with('get_nasabah'
, 'get_jenis_kredit', 'get_bunga'
a')-
>whereIn('pengajuan_status',
['dicairkan', 'diangsur',
'lunas'])-
>where('pengajuan_nasabah_id'
, Auth::user()->id)-
>orderBy('pengajuan_id',
'DESC')->get();
    }
    $data->tgl =
Helper::tanggal_indo($data-
>pengajuan-
>pengajuan_tanggal);
    $data->tgl_now =
Helper::tanggal_indo(now());
    $data->nominal =
Helper::rp($data->pengajuan-
>pengajuan_nominal);
    if($data->pengajuan-
>pengajuan_status ==
"dicairkan" || $data-
>pengajuan->pengajuan_status
== "diangsur" || $data-
>pengajuan->pengajuan_status
== "lunas" ){
        $data->pencairan
=
Pencairan::where('pencairan_p
engajuan_id', $id)->first();
        $angsuran =
Angsuran::where("angsuran_pen
cairan_id", $data->pencairan-
>pencairan_id)-
>orderBy('angsuran_ke',
'ASC')->get();
        $dataTables = [];
        $no = 1;

        foreach
($angsuran as $key => $item)
        {
            $dataTables[]
= [
                $no++,
                $item-
>angsuran_ke,
                Helper::t
anggal_indo($item-
>angsuran_jatuh_tempo),
                Helper::r
p($item->angsuran_pokok),
                Helper::r
p($item->angsuran_bunga),
                Helper::r
p($item->angsuran_denda),
                Helper::r
p($item->angsuran_total),
                strtoupp
e($item->angsuran_status),
                ($item-
>angsuran_status ==
'tagihan') ? '-' :
                Helper::tanggal_indo($item-
>angsuran_tanggal),
                $item-
>angsuran_pokok,
                $item-
>angsuran_bunga,
                $item-
>angsuran_denda,
                $item-
>angsuran_total,
            ];
            $data->heads
= [
                'No',
                'Ke',
                'Jatuh
Tempo',
                'Pokok',
                'Bunga',
                'Denda',
                'Total',
                'Status',
                'Tanggal
Bayar',
            ];
            $data->config
= [
                'data' =>
                $dataTables,
                'columnDe
fs' => [['targets' =>
                "_all"]],

```

```

        'order'
=> [[0, 'asc']],
    ];
    }
}

```

home.blade.php

```

@extends('adminlte::page')

@section('title',
'Dashboard')

@section('content_header')
<h1>Peminjaman Dana</h1>
@stop

@section('content')
<p> Selamat datang di Sistem
Informasi Kredit BPR Artajaya
Bhaktimulia</p>
@stop

@section('css')
<link rel="stylesheet"
href="/css/admin_custom.css">
@stop

@section('js')
<script>
</script>
@stop

user.index.php
@extends('adminlte::page')

@section('content_header')
<h3>User</h3>
@stop

@section('content')
@if ($message =
Session::get('success'))
<x-adminlte-alert
theme="success"
type="success">
    {{$message}}
</x-adminlte-alert>
@endif

```

```

        return
view('pages.laporan.detail_ka
rtu_piutang',
compact('data'));
    }
}

@if ($message =
Session::get('error'))
<x-adminlte-alert
theme="danger"
title="Danger">
    {{ $message }}
</x-adminlte-alert>
@endif

<div class="row">
    <div class="col-12">
        <div class="card">
            <div class="card-
header">
                <a
href="user/create" class="btn
btn-primary btn-sm shadow">
                    <i
class="fa fa-plus"></i>
                        Tambah
User
                    </a>
                </div>
            <!-- /.card-
header -->
            <div class="card-
body">
                <x-adminlte-
datatable id="example1"
:heads="$data->heads"
:config="$data->config"
striped hoverable
                    bordered
compressed with-buttons />
                </div>
            <!-- /.card-body
-->
        </div>
    </div>
</div>

```

```

@stop                                     </div>
                                           <!-- /.card-
@section('css')                          header -->
@stop                                     <div class="card-
                                           body">
@section('js')                             <x-adminlte-
@stop                                     datatable id="example1"
                                           :heads="$data->heads"
                                           :config="$data->config"
                                           striped hoverable
                                           bordered
                                           compressed with-buttons />
@stop                                     </div>
                                           <!-- /.card-body
                                           -->
                                           </div>
                                           </div>
                                           </div>
@stop
@section('css')
@stop
@section('js')
@stop

jenis_kredit.index.php
@extends('adminlte::page')
@section('title', 'Jenis
Kredit')
@section('content_header')
<h3>Jenis Kredit</h3>
@stop

@section('content')
@if ($message =
Session::get('success'))
<x-adminlte-alert
theme="success"
type="success">
    {{$message}}
</x-adminlte-alert>
@endif

@if ($message =
Session::get('error'))
<x-adminlte-alert
theme="danger"
title="Danger">
    {{ $message }}
</x-adminlte-alert>
@endif

<div class="row">
    <div class="col-12">
        <div class="card">
            <div class="card-
header">
                <a
href="jenis_kredit/create"
class="btn btn-primary btn-sm
shadow">
                    <i
class="fa fa-plus"></i>
                    Tambah
                    Jenis Kredit
                </a>
            </div>
        </div>
    </div>
</div>

```

```

bunga.index.php
@extends('adminlte::page')
@section('title', 'Bunga')
@section('content_header')
<h3>Bunga</h3>
@stop

```

```

@section('content')
@if ($message =
Session::get('success'))
<x-adminlte-alert
theme="success"
type="success">
    {{$message}}
</x-adminlte-alert>
@endif

```

```

@if ($message =
Session::get('error'))
<x-adminlte-alert
theme="danger"
title="Danger">

```

```

        {{ $message }}
    </x-adminlte-alert>
@endif

<div class="row">
    <div class="col-12">
        <div class="card">
            <div class="card-
header">
                <a
href="bunga/create"
class="btn btn-primary btn-sm
shadow" >
                    <i
class="fa fa-plus"></i>
                    Tambah
                    Bunga
                </a>
            </div>
            <!-- /.card-
header -->
            <div class="card-
body">
                <x-adminlte-
datatable id="example1"
:heads="$data->heads"
:config="$data->config"
striped hoverable
                bordered
compressed with-buttons />
            </div>
            <!-- /.card-body
-->
        </div>
    </div>
</div>

@stop

@section('css')
@stop

@section('js')
@stop

pinjaman.index.php
@extends('adminlte::page')
@section('content_header')
<h3>Pinjaman</h3>

```

```

@stop

@section('content')
@if ($message =
Session::get('success'))
<x-adminlte-alert
theme="success"
type="success">
    {{ $message }}
</x-adminlte-alert>
@endif

@if ($message =
Session::get('error'))
<x-adminlte-alert
theme="danger"
title="Danger">
    {{ $message }}
</x-adminlte-alert>
@endif

<div class="row">
    <div class="col-12">
        <div class="card">
            @if(Auth::user()-
>role == 'karyawan')
                <div class="card-
header">
                    <a
href="pinjaman/create"
class="btn btn-primary btn-sm
shadow">
                        <i
class="fa fa-plus"></i>
                        Tambah
                        Pengajuan Pinjaman
                    </a>
                </div>
            @endif
            <!-- /.card-
header -->
            <div class="card-
body">
                <x-adminlte-
datatable id="example1"
:heads="$data->heads"
:config="$data->config"
striped hoverable

```

```

                                bordered
compressed with-buttons />
                                </div>
                                <!-- /.card-body
-->
                                </div>
                                </div>
</div>

@stop

@section('css')
@stop

@section('js')
@stop

angsuran.blade.php
@extends('adminlte::page')

@section('content_header')
<h3>Laporan Angsuran
Pinjaman</h3>
@stop

@section('content')
@if ($message =
Session::get('success'))
<x-adminlte-alert
theme="success"
type="success">
    {{$message}}
</x-adminlte-alert>
@endif

@if ($message =
Session::get('error'))
<x-adminlte-alert
theme="danger"
title="Danger">
    {{ $message }}
</x-adminlte-alert>
@endif

<div class="row">
    <div class="col-12">
        <div class="card">
            <!-- /.card-
header -->
                                <div class="card-
header">
                                    <div
class="row">
                                        <button
type="button" class="btn btn-
primary btn-sm shadow mr-2"
data-toggle="modal" data-
target="#modal"><i class="fa
fa-filter"></i> Filter
                                        </button>
                                                <button
type="button" class="btn btn-
primary btn-sm shadow"
onclick="printContent('print'
);">
                                                    <i
class="fa fa-print"></i>
Cetak
                                                </button>
                                            </div>
                                        </div>
                                        <div id="print"
class="card-body">
                                            <div
class="table-responsive">
                                                <div
class="text-center text-bold
text-uppercase">BPR Artajaya
Bhaktimulia</div>
                                                <div
class="text-center text-bold
text-uppercase">Laporan
Angsuran Pinjaman</div>
                                                <div
class="text-center mb-2 text-
bold text-uppercase">{{$data-
>periode}}</div>
                                                    <table
class="table table-hover
table-sm">
                                                        <thead>
                                                            <tr class="bg-secondary"
style="background-color:
#6c757d!important;color:
#fff!important;">

```

```

        @foreach($data->heads as
$col)
            <th
scope="col">{{$col}}</th>
        @endforeach
    /tr>
ad>
y>
if(count($data-
>config['data']) < 1) <tr
class="table-secondary">
    <th colspan="7"
scope="row">Data Kosong</th>
</tr>
@endif
@php $nominal = 0; @endphp
@foreach($data-
>config['data'] as $key =>
$item)
    @php $nominal =
$nominal+$item[7]; @endphp
    <tr>
        <th
scope="row">{{$item[0]}}</th>
        <td>{{$item[1]}}</td>
        <td>{{$item[2]}}</td>
        <td>{{$item[3]}}</td>
        <td>{{$item[4]}}</td>
        <td>{{$item[5]}}</td>
        <td>{{$item[6]}}</td>
    </tr>
    @if(count($data-
>config['data']) >= 1)
        <tr class="table-
secondary">
            <th colspan="5"
class="text-right"
scope="row">Total :</th>
            <th colspan="2"
scope="row">@rp($nominal)</th>
        </tr>
    @endif
dy>
    </tbody>
</table>
</div>
<!-- /.card-body
-->
</div>
</div>
</div>
<x-adminlte-modal id="modal"
hidden_footer="true"
title="Filter" size="md"
theme="info" icon="fas fa-
filter " v-centered static-
backdrop scrollable>
    <form method="GET"
action="{{route('laporan.angs
uran')}}">
        <div class="card-
body">
            <x-adminlte-
select2 name="status"

```

```

label="Status Angsuran"
style="width: 100%;">
    <option
value="">Pilih Status
Angsuran</option>
    <option
value="tagihan" @if($data-
>status == "tagihan")
selected
@endif>Tagihan</option>
    <option
value="lunas" @if($data-
>status == "lunas") selected
@endif>Lunas</option>
</x-adminlte-
select2>
    <div class="form-
group">
        <label
for="date">
            Tanggal
        </label>
        <div
class="input-group input-
group-sm">
            <input
id="date" name="date"
value="{{ $data->date }}"
class="form-control"
type="text">
        </div>
    </div>
    <div class="modal-
footer">
        <hr />
        <x-adminlte-
button type="submit"
class="mr-auto" theme="info"
label="Filter" />
        <x-adminlte-
button type="reset"
theme="danger" label="Batal"
data-dismiss="modal" />
    </div>
</form>
</x-adminlte-modal>

@stop

@section('css')
@stop

@section('js')
<script
type="text/javascript"
src="https://cdn.jsdelivr.net
/momentjs/latest/moment.min.j
s"></script>
<script
type="text/javascript"
src="https://cdn.jsdelivr.net
/npm/daterangepicker/daterang
epicker.min.js"></script>
<link rel="stylesheet"
type="text/css"
href="https://cdn.jsdelivr.ne
t/npm/daterangepicker/dateran
gepicker.css" />

<script>
    $(function() {
        $('input[name="date"]
').daterangepicker({
            opens: 'left',
            locale: {
                cancelLabel:
                'Clear'
            }
        });
        $('#date').on('cancel
.daterangepicker',
function(ev, picker) {
            //do something,
            like clearing an input
            $('#date').val(''
);
        });
    });
</script>
@stop

kartu_piutang.blade.php
@extends('adminlte::page')

@section('content_header')
<h3>Laporan Kartu
Piutang</h3>

```

```

@stop
@section('content')
<div class="row">
  <div class="col-12">
    <div class="card">
      <!-- /.card-
header -->
      <div class="card-
header">
        <div
class="row">
          <button
type="button" class="btn btn-
primary btn-sm shadow mr-2"
data-toggle="modal" data-
target="#modal"><i class="fa
fa-filter"></i> Filter
</button>
          <button
type="button" class="btn btn-
primary btn-sm shadow"
onclick="printContent('print'
);">
            <i
class="fa fa-print"></i>
Cetak
          </button>
        </div>
      </div>
      <div id="print"
class="card-body">
        <div
class="table-responsive">
          <div
class="text-center text-bold
text-uppercase">BPR Artajaya
Bhaktimulia</div>
          <div
class="text-center text-bold
text-uppercase">Laporan Kartu
Piutang</div>
          <div
class="text-center mb-2 text-
bold text-uppercase">{{$data-
>periode}}</div>

```

```

      <table
class="table table-hover
table-sm">
        <thead>
          <tr class="bg-secondary"
style="background-color:
#6c757d!important;color:
#fff!important;">
            @foreach($data->heads as
$col)
              <th
scope="col">{{$col}}</th>
            @endforeach
            <th scope="col">Aksi</th>
          </tr>
        </thead>
        <tbody>
          @if(count($data->row) < 1)
            <tr class="table-secondary">
              <th colspan="9"
scope="row">Data Kosong</th>
            </tr>
          @endif
          @php $tagihan = 0; @endphp
          @php $angsuran = 0; @endphp
          @php $siswa = 0; @endphp
          @foreach($data->row as $key =>
$item)
            <tr>

```



```

        <th
scope="row">{{ $key+1 }}</th>
        <td>{{ strtoupper($item-
>get_nasabah->name) }}</td>
        <td>@date($item-
>pencairan_tgl)</td>
        <td>{{ strtoupper($item-
>get_jenis_kredit-
>jenis_kredit_nama) }}</td>
        <td>{{ strtoupper($item-
>get_bunga->bunga_jenis) }}
{{ strtoupper($item-
>get_bunga-
>bunga_persentase) }}%</td>
        <td>@rp($item-
>total_tagihan)</td>
        <td>@rp($item-
>total_angsuran)</td>
        <td>@rp($item-
>sisangsuran)</td>
        <td><a
href="/laporan_kartu_piutang/
{{ $item-
>pengajuan_id }}">Detail</a></
td>
    </tr>
    @
    php $tagihan = $tagihan +
$item->total_tagihan; @endphp
    @
    php $angsuran = $angsuran +
$item->total_angsuran;
@endphp
    @
    php $sisangsuran = $sisangsuran + $item-
>sisangsuran; @endphp
    @
endforeach
    @
    if(count($data->row) >= 1)
    <
    tr class="table-secondary">
        <th colspan="5"
class="text-right"
scope="row">Total :</th>
        <th
scope="row">@rp($tagihan)</th>
        >
        <th
scope="row">@rp($angsuran)</th>
        <th colspan="2"
scope="row">@rp($sisangsuran)</th>
        </tr>
    @
endif
    </tbody>
</div>
</div>
<!-- /.card-body
-->
</div>
</div>
</div>
<x-adminlte-modal id="modal"
hidden_footer="true"
title="Filter" size="md"
theme="info" icon="fas fa-
filter " v-centered static-
backdrop scrollable>
    <form method="GET"
action="{{ route('laporan.kart
u_piutang') }}">
        <div class="card-
body">
            <div class="form-
group">
                <label
for="date">

```

```

                Tanggal
            </label>
        <div
class="input-group input-
group-sm">
            <input
id="date" name="date"
value="{{ $data->date }}"
class="form-control"
type="text">
        </div>
    </div>
</div>
<div class="modal-
footer">
    <hr />
    <x-adminlte-
button type="submit"
class="mr-auto" theme="info"
label="Filter" />
    <x-adminlte-
button type="reset"
theme="danger" label="Batal"
data-dismiss="modal" />
</div>
</form>
</x-adminlte-modal>

@stop

@section('css')
@stop

@section('js')
<script
type="text/javascript"

```

```

src="https://cdn.jsdelivr.net
/momentjs/latest/moment.min.j
s"></script>
<script
type="text/javascript"
src="https://cdn.jsdelivr.net
/npm/daterangepicker/daterang
epicker.min.js"></script>
<link rel="stylesheet"
type="text/css"
href="https://cdn.jsdelivr.ne
t/npm/daterangepicker/dateran
gepicker.css" />

<script>
    $(function() {
        $('input[name="date"]
').daterangepicker({
            opens: 'left',
            locale: {
                cancelLabel:
'Clear'
            }
        });
        $('#date').on('cancel
.daterangepicker',
function(ev, picker) {
    //do something,
like clearing an input
    $('#date').val(''
);
    });
});
</script>
@stop

```



**KARTU BIMBINGAN PROYEK AKHIR - DIPLOMA III
UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA**

Nama : SANGEETA A.C. SIMANJUNTAK
 NIM : 203210024
 Program Studi : SIA
 Pembimbing : Al. Agus Subagyo, S.E., M.Si.
 Judul : Sistem Informasi Kredit BPR Artajaya Bhaktimulia

NO	HARI/ TANGGAL	TOPIK	PARAF PEMBIMBING
1	Rabu / 17 Mei 2023	Pengajuan judul	
2	Rabu / 7 Juni 2023	Bab I & II revisi	
3	Kamis / 15 Juni 2023	Bab I & II acc, DAD & Relasi revisi	
4	Senin / 19 Juni 2023	Bab I, II, III acc → ada Catatan Surat Lomol.	
5	Kamis / 6 Juli 2023	Demo Aplikasi → Validasi ②	
6	Selasa / 11 Juli 2023	Demo Aplikasi → Laporan ③	
7	Senin / 17 Juli 2023	Demo Aplikasi → Hak akses. [terayaknadi]	
8	Kamis / 20 Juli 2023	Demo Aplikasi → Belanja FRI 20/7	
9	Kamis 20/7 2023	Acc Master - Cek nama field antara master dg aplikasi 20/7	
10			

Dosen Pembimbing TGA

Nama Pembimbing
NIDN

PROYEK AKHIR
SISTEM INFORMASI KREDIT BANK PERKREDITAN RAKYAT (BPR)

ARTAJAYA BHAKTIMULIA



acc review,
[Signature] 3/8 2023

(ENDANG W.)

acc review 3/8-23

[Signature]

(DANI K.)

[Signature]
acc review
ALAGUS
3/8 23

SANGEETA ANGGUN CHYNTA SIMANJUNTAK

NIM : 203210024

PROGRAM STUDI SISTEM INFORMASI AKUNTANSI

PROGRAM DIPLOMA TIGA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA

YOGYAKARTA

2023

PEMBERITAHUAN SEBELUM UJIAN : Pengumpulan akhir dokumen Tugas Akhir/Skripsi melewati semester berjalan, mahasiswa harus menyelesaikan registrasi dan KRS semester berikutnya.					
KRITERIA KELULUSAN UJIAN SIDANG / PENDADARAN					
1. Lulus ujian tanpa syarat, disebut kriteria 1.					
2. Lulus bersyarat, disebut kriteria 2, yaitu dengan sedikit perbaikan atau penyempurnaan text dan atau program dalam waktu maks sampai tanggal dan tidak ada ujian lagi. Jika dalam waktu yang ditentukan mahasiswa tersebut tidak dapat menyelesaikan, maka, mahasiswa yang bersangkutan dianggap tidak lulus ujian.					
3. Tidak lulus ujian sidang/pendadaran, disebut kriteria 3, dijelaskan, disarankan Ketua Tim Penguji untuk mempelajari ulang materi, merombak program/teks, atau mengganti judul.					
Ketentuan bagi peserta yang tidak lulus ujian sidang / pendadaran.					
1) Mahasiswa wajib menempuh ujian sidang/pendadaran ulang					
2) Kesempatan ujian sidang/pendadaran ulang hanya diberikan dalam rentang waktu maksimum 6 bulan, setelah ujian sidang/pendadaran					
3) Jika sampai batas waktu maksimum 6 bulan tersebut belum dapat diajukan/diselesaikan, maka calon peserta ujian dinyatakan sebagai mahasiswa peserta Skripsi/TGA baru, dengan segala ketentuan yang berlaku bagi peserta baru					
4) Mahasiswa yang akan menempuh ujian sidang/pendadaran ulang ini diwajibkan membayar biaya ujian setara 2 SKS praktik, sesuai tahun angkatan					
					Yogyakarta, _____
					Memahami dan bersedia
					Mematuhi peraturan di atas,
					Nama Mahasiswa
					Sangeeta Simanjuntak

Catatan Pendadaran

Hari, tanggal	:	Senin, 31 Juli 2023
Waktu	:	08.00
Nama	:	Sangeeta Simanjuntak
No. Mahasiswa / Jurusan	:	203210024 / SIA
Nama Dosen	:	Al. Agus Subagyo, S.E., M.Si.
	:	
		Penguji
Hal yang harus diperbaiki	:	
	1.	Penamaan Tabel harusnya di atas, bukan di bawah. Sitasi untuk penulisan dasar teori
	2.	tabel user diganti tabel nasabah. tabel bunga keterangan di atasnya direvisi, ada tambahan keterangan di naskah untuk jenis kredit
	3.	Penulisan DAD dan Tabel, field tabel dalam naskah disesuaikan dengan yang ada dalam
	4.	
*coret yang tidak perlu		

Catatan Pendadaran

Hari, tanggal : Senin, 31 Juli 2023
 Waktu : 08.00
 Nama : Sangeeta Simanjuntak
 No. Mahasiswa / Jurusan : 203210024 / SIA
 Nama Dosen : Al. Agus Subagyo, S.E., M.Si.

	Hal yang harus diperbaiki	Pembimbing
1.	nama tabel User disesuaikan	Al. Agus Subagyo, S.E., M.Si.
2.	dalam saran dimasukkan saat direktur mau approve pinjaman, ditunjukkan siapa pegawai	Al. Agus Subagyo, S.E., M.Si.
3.	pelembutan nama tabel dibagian atas tabel	
4.	penulisan citation dalam naskah	

*coret yang tidak perlu

KEPUTUSAN HASIL UJIAN PENDADARAN					
Sesuai dengan hasil sidang pendadaran pada tanggal		maka			
Nama Mahasiswa	Sangeeta Simanjuntak				
NIM / Program Studi	203210024 / SIA				
Jenjang	D3				
	dinyatakan	LULUS	dengan kriteria	2	
Ketua Penguji	Endang Wahyuningsih, S.Kom., M. Cs.				