

DAFTAR PUSTAKA

- Andre. (2014). *Pengertian dan Fungsi PHP dalam Pemrograman Web*. 15 Desember 2014 <http://www.duniaikom.com/pengertian-dan-fungsi-php-dalam-pemrograman-web/>.
- Data, M., Ramadhan, G., & Amron, K. (2017). *Analisis Availabilitas dan Reliabilitas Multi-Master Database Server Dengan State Snapshot Transfers (SST) Jenis Rsync Pada MariaDB Galera Cluster*.
- Gumilar, S. K. Y., Suryatiningsih, & Muhammad, W. (2017). *Aplikasi Berbasis Web Pengelolaan Keuangan Sekolah Study Kasus MA Persis Tarogong Garut*.
- Kadir, A., Sarosa, S., Mcleod, R., P. Schell, G., Martono, R., & Mulyani, S. (2008). *Practical Inventory Management*.
- Kementerian Pendidikan. (2016). *Peraturan Menteri Pendidikan dan Kebudayaan No. 75 Tahun 2016 Tentang Komite Sekolah*.
- Munawar. (2003). *Pemodelan Visual dengan UML*. Graha Ilmu. Yogyakarta.
- Presiden Republik Indonesia. (2003). *Undang-Undang Republik Indonesia Nomor 20 Tahun 2003 Tentang Sistem Pendidikan Nasional*.
- Pressman, R. S., & Maxim, B. R. (2014). *Software Engineering: A Practitioner's Approach, Eight Edition*.
- Pribadi, A. (2016). *Konsep dan Implementasi Pemrograman Laravel 5*. Lokomedia. Yogyakarta.
- Satrya, F. (2016). *Pemrograman WEB {HTML, CSS, dan JavaScript}*. Andi. Yogyakarta.
- Setiawati, L. (2012). *Efektivitas Pengembangan Manajemen Pendidikan Tinggi*. Retrieved from http://jurnal.upi.edu/file/Linda_setiawati.pdf.
- Triandini, E. (2012). *Step Bby Step Desain Proyek Menggunakan UML*. Andi. Yogyakarta.
- Wardana. (2014). *Menjadi Master PHP dengan Framework Codeigniter*. Alex Media Komputindo. Jakarta

- Wulansari, N., Purnama, B. E., Wardati, I. U., & Rocki, K. (2013). *Sistem Informasi Pengelolaan Keuangan Sekolah di SMK PGRI 1 Pacitan*.
- Zakiyudin, A. (2012). *Sistem Informasi Manajemen*. Edisi ke 2. Mitra Wacana Media. Jakarta
- Agus Mulyanto. (2009). *Sistem Informasi Konsep dan Aplikasi*. Yogyakarta: Pustaka Pelajar.
- Akbar Muhammad, 2018. *Pengembangan RESTful API Untuk Application Specific High Level Location Service*. Skripsi. Universitas Islam Indonesia. Yogyakarta.
- Danandjaya Saputra dan Rizal Fathoni Aji, 2018. *Analisis Perbandingan Performa Web Service Rest Menggunakan Framework Laravel, Django Dan Ruby On Rails Untuk Akses Data Dengan Aplikasi Mobile (Studi Kasus: Portal E-Kampus Stt Indonesia Tanjungpinang)*. *Bangkit Indonesia*. Vol. 2, No.VII.

**L
A
M
P
I
R
A
N**

Lampiran 1 Source Code Program Model**Model (app/models) User.php**

```

<?php
namespace App\Models;
use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Spatie\Permission\Traits\HasRoles; use App\Models\Petugas;
use App\Models\Siswa;
class User extends Authenticatable
{
use HasFactory, Notifiable, HasRoles;
/**
 * The attributes that are mass assignable.
 *
 * @var array
 */
protected $fillable = [
    'username',
    'email',
    'password',
    'level',
];
/**
 * The attributes that should be hidden for arrays.
 *
 * @var array
 */
protected $hidden = [
    'password',

```

```

        'remember_token',
    ];
    /**
     * The attributes that should be cast to native types.
     *
     * @var array
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];
    public function petugas()
    {
        return $this->hasOne(Petugas::class);
    }
    public function siswa()
    {
        return $this->hasOne(Siswa::class);
    }
}

```

Kelas.php

```

<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use App\Models\Siswa;
class Kelas extends Model
{
    use HasFactory;
    protected $fillable = [
        'nama_kelas',
    ];
}

```

```

        'kompetensi_keahlian',
    ];
    public function siswa()
    {
        return $this->hasMany(Siswa::class);
    }
}

```

Pembayaran.php

```

<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use App\Models\Petugas;
use App\Models\Siswa;
use Carbon\Carbon;
class Pembayaran extends Model
{
    use HasFactory;
    protected $table = 'pembayaran';
    protected $fillable = [
        'kode_pembayaran',
        'petugas_id',
        'siswa_id',
        'nisn',
        'tanggal_bayar',
        'bulan_bayar',
        'tahun_bayar',
        'jumlah_bayar',
    ];
    public function getTanggalBayarAttribute($value)

```

```

    {
        return Carbon::parse($value)->format('d-m-Y');
    }
    public function getJumlahBayarAttribute($value)
    {
        return "Rp ".number_format($value, 0, 2, '.');
    }
    public function petugas()
    {
        return $this->belongsTo(Petugas::class);
    }
    public function siswa()
    {
        return $this->belongsTo(Siswa::class);
    }
}

```

Petugas.php

```

<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use App\Models\User;
use App\Models\Siswa;
use App\Models\Pembayaran;
class Petugas extends Model
{
    use HasFactory;
    protected $fillable = [
        'user_id',
        'kode_petugas',

```

```

        'nama_petugas',
        'jenis_kelamin',
    ];
    public function user()
    {
        return $this->belongsTo(User::class);
    }
    public function siswa()
    {
        return $this->hasMany(Siswa::class);
    }
    public function pembayaran()
    {
        return $this->hasMany(Pembayaran::class);
    }
}

```

Siswa.php

```

<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use App\Models\User;
use App\Models\Kelas; use App\Models\Spp;
use App\Models\Petugas;
use App\Models\Pembayaran;
class Siswa extends Model
{
    use HasFactory;
    protected $table = 'siswa';
    protected $fillable = [

```



```
'user_id',
    'kode_siswa',
    'nispn',
    'nis',
    'nama_siswa',
    'jenis_kelamin',
    'alamat',
    'no_telepon',
    'kelas_id',
];
public function user()
{
    return $this->belongsTo(User::class);
}
public function kelas()
{
    return $this->belongsTo(Kelas::class);
}
public function petugas()
{
    return $this->belongsTo(Petugas::class);
}
public function spp()
{
    return $this->belongsTo(Spp::class);
}
public function pembayaran()
{
    return $this->hasMany(Pembayaran::class);
}
}
```

Spp.php

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use App\Models\Siswa;
class Spp extends Model
{
    use HasFactory;
    protected $table = 'spp';
    protected $fillable = [
        'tahun',
        'nominal',
    ];
    public function siswa()
    {
        return $this->hasMany(Siswa::class);
    }
}
```

Lampiran 2 Source Code Program Controller**Controller (app/http/controller)****Siswacontroller.php**

```

<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request; use App\Models\Pembayaran;
use App\Models\Siswa;
use App\Models\Spp;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Str;
use DataTables;
use PDF;
class SiswaController extends Controller
{
    public function pembayaranSpp()
    {
        $spp = Spp::all();
        return view('siswa.pembayaran-spp', compact('spp'));
    }
    public function pembayaranSppShow(Spp $spp)
    {
        $siswa = Siswa::where('user_id', Auth::user()->id)
            ->first();
        $pembayaran = Pembayaran::with(['petugas', 'siswa'])
            ->where('siswa_id', $siswa->id)
            ->where('tahun_bayar', $spp->tahun)
            ->oldest()
            ->get();
        return view('siswa.pembayaran-spp-show', compact('pembayaran', 'siswa',
            'spp'));
    }
}

```

```

public function historyPembayaran(Request $request)
{
    if ($request->ajax()) {
        $siswa = Siswa::where('user_id', Auth::user()->id)
            ->first();
        $data = Pembayaran::with(['petugas', 'siswa' => function($query) {
            $query->with(['kelas']);
        }])
            ->where('siswa_id', $siswa->id)
            ->latest()
            ->get();
        return DataTables::of($data)
            ->addIndexColumn()
            ->addColumn('action', function($row) {
                $btn = '<div class="row"><a href="'.route('siswa.history-
pembayaran.preview', $row->id).'" class="btn btn-danger btn-sm ml-2"
target="_blank">
                    <i class="fas fa-print fa-fw"></i>
                    </a>';
                return $btn;
            })
            ->rawColumns(['action'])
            ->make(true);
    }
    return view('siswa.history-pembayaran');
}

public function previewHistoryPembayaran($id)
{
    $data['siswa'] = Siswa::where('user_id', Auth::user()->id)
        ->first();
    $data['pembayaran'] = Pembayaran::with(['petugas', 'siswa'])

```

```

->where('id', $id)
->where('siswa_id', $data['siswa']->id)
->first();
$pdf = PDF::loadView('siswa.history-pembayaran-preview',$data);
return $pdf->stream();
}
public function laporanPembayaran()
{
    $spp = Spp::all();
    return view('siswa.laporan', compact('spp'));
}
public function printPdf(Request $request)
{
    $siswa = Siswa::where('user_id', Auth::user()->id)
        ->first();
    $data['pembayaran'] = Pembayaran::with(['petugas', 'siswa'])
        >where('siswa_id', $siswa->id)
        ->where('tahun_bayar', $request->tahun_bayar)
        ->get();
    $data['data_siswa'] = $siswa;
    if ($data['pembayaran']->count() > 0) {
        $pdf = PDF::loadView('siswa.laporan-preview', $data);
        return $pdf->download('pembayaran-spp-'.$siswa->nama_siswa.'-'.
            $siswa->nisn.'-'.
            $request->tahun_bayar.'-'.
            Str::random(9).'pdf');
    }else{
        return back()->with('error', 'Data Pembayaran Spp Anda Tahun '.$request
->tahun_bayar.' tidak tersedia');
    }
}
}
}

```

Pembayarancontroller.php

```

<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request; use App\Models\User;
use App\Models\Siswa;
use App\Models\Spp;
use App\Models\Petugas;
use App\Models\Pembayaran;
use Carbon\Carbon;
use Illuminate\Support\Str;
use Illuminate\Support\Arr;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Validator;
use App\Helpers\Bulan;
use PDF;
use DataTables;

class PembayaranController extends Controller
{
    public function index(Request $request)
    {
        if ($request->ajax()) {
            $data = Siswa::with(['kelas'])->latest();
            return DataTables::of($data)
                ->addIndexColumn()
                ->addColumn('action', function($row) {
                    $btn = '<div class="row"><a href="'.route('pembayaran.bayar', $row-
>nisn).'"class="btn btn- primary btn-sm ml-2">
                        <i class="fas fa-money-check"></i> BAYAR
                    </a>';
                    return $btn;
                });
        }
    }
}

```

```

    })
    ->rawColumns(['action'])
    ->make(true);
}
return view('pembayaran.index');
}
public function bayar($nisn)
{
    $siswa = Siswa::with(['kelas'])
    ->where('nisn', $nisn)
    ->first();
    $spp = Spp::all();
    return view('pembayaran.bayar', compact('siswa', 'spp'));
}
public function spp($tahun)
{
    $spp = Spp::where('tahun', $tahun)
    ->first();
    return response()->json([
        'data' => $spp,
        'nominal_rupiah' => 'Rp '.number_format($spp->nominal, 0, 2, '.'),
    ]);
}
public function prosesBayar(Request $request, $nisn)
{
    $request->validate([
        'jumlah_bayar' => 'required',
    ], [
        'jumlah_bayar.required' => 'Jumlah bayar tidak boleh kosong!'
    ]);
}

```

```

        $petugas = Petugas::where('user_id', Auth::user()->id)
        ->first();
    $pembayaran = Pembayaran::whereIn('bulan_bayar', $request->bulan_bayar)
        ->where('tahun_bayar', $request->tahun_bayar)
        ->where('siswa_id', $request->siswa_id)
        ->pluck('bulan_bayar')
        ->toArray();
    if (!$pembayaran) {
        DB::transaction(function() use($request, $petugas) {
            foreach ($request->bulan_bayar as $bulan) {
                Pembayaran::create([
                    'kode_pembayaran' => 'SPPR'.Str::upper(Str::random(5)),
                    'petugas_id' => $petugas->id,
                    'siswa_id' => $request->siswa_id,
                    'nisn' => $request->nisn,
                    'tanggal_bayar' => Carbon::now('Asia/Jakarta'),
                    'tahun_bayar' => $request->tahun_bayar,
                    'bulan_bayar' => $bulan,
                    'jumlah_bayar' => $request->jumlah_bayar,
                ]);
            }
        });
        return redirect()->route('pembayaran.history-pembayaran')
            ->with('success', 'Pembayaran berhasil disimpan!');
    } else {
        return back()
            ->with('error', 'Siswa Dengan Nama : '.$request->nama_siswa.', NISN : ' .
                $request->nisn.' Sudah Membayar Spp di bulan yang diinput (' .
                implode($pembayaran,',').')".' , di Tahun : '.$request->tahun_bayar.',
                Pembayaran Dibatalkan');
    }
}

```



```

public function statusPembayaran(Request $request)
{
    if ($request->ajax()) {
        $data = Siswa::with(['kelas'])
            ->latest()
            ->get();

        Return DataTables::of($data)
            ->addIndexColumn()
            ->addColumn('action', function($row) {
                $btn = '<div class="row"><a href="'.route('pembayaran.status-
pembayaran.show',$row->nisn).
                '" class="btn btn-primary btn-sm">DETAIL</a>'; return $btn;
            })
            ->rawColumns(['action'])
            ->make(true);
    }
    return view('pembayaran.status-pembayaran');
}

public function statusPembayaranShow(Siswa $siswa)
{
    $spp = Spp::all();
    return view('pembayaran.status-pembayaran-tahun', compact('siswa', 'spp'));
}

public function statusPembayaranShowStatus($nisn, $tahun)
{
    $siswa = Siswa::where('nisn', $nisn)
        ->first();
    $spp = Spp::where('tahun', $tahun)
        ->first();
    $pembayaran = Pembayaran::with(['siswa'])

```

```

->where('siswa_id', $siswa->id)
->where('tahun_bayar', $spp->tahun)
->get();
return view('pembayaran.status-pembayaran-show', compact('siswa', 'spp',
'pembayaran'));
}
public function historyPembayaran(Request $request)
{
if ($request->ajax()) {
    $data = Pembayaran::with(['petugas', 'siswa' => function($query){
        $query->with('kelas');
    }])
    ->latest()->get();
return DataTables::of($data)
    ->addIndexColumn()
    ->addColumn('action', function($row) {
        $btn = '<div class="row"><a href="'.route('pembayaran.history-
pembayaran.print',$row->id).'" class="btn btn-danger btn-sm ml-2" target="_blank">
        <i class="fas fa-print fa-fw"></i></a>';
        return $btn;
    })
    ->rawColumns(['action'])
    ->make(true);
}
return view('pembayaran.history-pembayaran');
}
public function printHistoryPembayaran($id)
{
    $data['pembayaran'] = Pembayaran::with(['petugas', 'siswa'])
    ->where('id', $id)
    ->first();
}

```

```

$pdf = PDF::loadView('pembayaran.history-pembayaran-preview',$data);
return $pdf->stream();
}
public function laporan()
{
    return view('pembayaran.laporan');
}
public function printPdf(Request $request)
{
    $tanggal = $request->validate([
        'tanggal_mulai' => 'required',
        'tanggal_selesai' => 'required',
    ]);
    $data['pembayaran'] = Pembayaran::with(['petugas', 'siswa'])
        >whereBetween('tanggal_bayar', $tanggal)->get();
    if ($data['pembayaran']->count() > 0) {
        $pdf = PDF::loadView('pembayaran.laporan-preview', $data);
        return $pdf->download('pembayaran-spp-' .
            Carbon::parse($request->tanggal_mulai)->format('d-m-Y').'-' .
            Carbon::parse($request->tanggal_selesai)->format('d-m-Y') .
            Str::random(9).'pdf');
    }else{
        return back()->with('error', 'Data pembayaran spp tanggal ' .
            Carbon::parse($request->tanggal_mulai)->format('d-m-Y').' sampai dengan ' .
            Carbon::parse($request->tanggal_selesai)->format('d-m-Y').' Tidak Tersedia');
    }
}
}
}

```

Homecontroller.php

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
class HomeController extends Controller
{
    public function index()
    {
        return view('home');
    }
}
```

Profilecontroller.php

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request; use App\Models\User;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Validator;
use Illuminate\Support\Facades\Hash;
class ProfileController extends Controller
{
    public function index()
    {
        return view('profile');
    }
    public function update(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'old_password' => 'required',
            'new_password' => 'required|min:5',
        ], [
```

```

        'old_password.required' => 'Password sekarang harus diisi!',
        'new_password.required' => 'Password baru harus diisi!',
        'new_password.min' => 'Password baru harus lebih dari 5 karakter!',
    ];
    if ($validator->fails()) {
        return back()->withErrors($validator)->withInput();
    }
    if (!Hash::check($request->new_password, Auth::user()->password)) {
        if (Hash::check($request->old_password, Auth::user()->password)) {
            User::findOrFail(Auth::user()->id)->update([
                'password' => Hash::make($request->new_password),
            ]);
            return redirect()->route('home.index')->with('success', 'Password anda
berhasil diubah!');
        }else{
            return back()->with('error', 'Password sekarang salah!');
        }
    }else{
        return back()->with('error', 'Password baru tidak boleh sama dengan password
sekarang');
    }
}
}
}
}

```

Sppcontroller.php

```

<?php
namespace App\Http\Controllers\Admin;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request; use App\Models\Spp;
use Illuminate\Support\Facades\Validator;
use App\DataTables\SppDataTable;

```

```

class SppController extends Controller
{
    public function __construct()
    {
        $this->middleware(['permission:read-spp'])->only(['index', 'show']);
        $this->middleware(['permission:create-spp'])->only(['create', 'store']);
        $this->middleware(['permission:update-spp'])->only(['edit', 'update']);
        $this->middleware(['permission:delete-spp'])->only(['destroy']);
    }
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index(Request $request, SppDataTable $datatable)
    {
        if ($request->ajax()) {
            return $datatable->data();
        }
        return view('admin.spp.index');
    }
    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'tahun' => ['required', 'unique:spp'],

```

```

        'nominal' => ['required', 'numeric'],
    ];
    if ($validator->passes()) {
        Spp::create($request->all());
        return response()->json(['message' => 'Data berhasil disimpan!']);
    }
    return response()->json(['error' => $validator->errors()->all()]);
}
/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit($id)
{
    $spp = Spp::findOrFail($id);
    return response()->json(['data' => $spp]);
}
/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, $id)
{
    $validator = Validator::make($request->all(), [
        'nominal' => ['required', 'numeric']
    ]);

```

```

    if ($validator->passes()) {
        Spp::findOrFail($id)->update($request->all());
        return response()->json(['message' => 'Data berhasil diupdate!']);
    }
    return response()->json(['error' => $validator->errors()->all()]);
}
/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function destroy($id)
{
    Spp::findOrFail($id)->delete();
    return response()->json(['message' => 'Data berhasil dihapus!']);
}
}

```

Rolecontroller.php

```

<?php
namespace App\Http\Controllers\Admin;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Spatie\Permission\Models\Role;
use App\DataTables\RoleDataTable;
class RoleController extends Controller
{
    /**
     * Display a listing of the resource.
     *

```



```

* @return \Illuminate\Http\Response
*/
public function index(Request $request, RoleDataTable $datatable)
{
    if ($request->ajax()) {
        return $datatable->data();
    }
    return view('admin.roles.index');
}
/**
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
    return view('admin.roles.create');
}
/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    foreach($request->name as $name) {
        Role::create([
            'name' => $name,
        ]);
    }
}

```

```

        return response()->json(['message' => 'Data berhasil disimpan!']);
    }
/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function show($id)
{
    $role = Role::findOrFail($id);
    return view('admin.roles.show', compact('role'));
}
/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit($id)
{
    $role = Role::findOrFail($id);
    return response()->json(['data' => $role]);
}
/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */

```

```
public function update(Request $request, $id)
{
    Role::findOrFail($id)->update($request->all());
    return response()->json(['message' => 'Data berhasil diupdate!']);
}
/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function destroy($id)
{
    Role::findOrFail($id)->delete();
    return response()->json(['message' => 'Data berhasil dihapus!']);
}
}
```

Lampiran 3 Source Code Program View

View (resources/view)

Laporan.blade.php

```

@extends('layouts.backend.app')
@section('title', 'Laporan')
@section('content_title', 'Laporan')
@section('content')
<x-alert></x-alert>
<div class="row">
    <div class="col-lg-6">
        <div class="card">
            <div class="card-header">Laporan Pembayaran</div>
            <div class="card-body">
                <form method="POST" action="{{ route('siswa.laporan-pembayaran.print-pdf')
}}}">
                    @csrf
                    <div class="form-group">
                        <label for="tahun_bayar">Tahun</label>
                        <select name="tahun_bayar" required="" class="form-
control" id="tahun_bayar">
                            <option disabled="" selected="">- PILIH TAHUN-</option>
                            @foreach($spp as $row)
                                <option value="{{ $row->tahun }}">{{ $row->tahun }}</option>
                            @endforeach
                        </select>
                    </div>
                    <div class="form-group">
                        <button type="submit" class="btn btn-danger btn-sm">
                            <i class="fas fa-print fa-fw"></i> CETAK LAPORAN
                        </button>

```

```

        </div>
    </form>
</div>
</div>
</div>
</div>
</div>
@endsection

```

Status-pembayaran.blade.php

```

@extends('layouts.backend.app')
@section('title', 'Status Pembayaran Siswa')
@push('css')
<!-- DataTables -->
<link rel="stylesheet" href="{{ asset('templates/backend/AdminLTE-3.1.0')
}}/plugins/datatables-bs4/css/dataTables.bootstrap4.min.css">
<link rel="stylesheet" href="{{ asset('templates/backend/AdminLTE-3.1.0')
}}/plugins/datatables-responsive/css/dataTables.bootstrap4.min.css">
@endpush
@section('content_title', 'Data Siswa')
@section('content')
<x-alert></x-alert>
<div class="row">
    <div class="col-12">
        <div class="card">
            <div class="card-body">
                <table id="dataTable2" class="table table-bordered table-striped">
                    <thead>
                        <tr>
                            <th>No</th>
                            <th>Nama Siswa</th>
                            <th>Nisn</th>

```

```

        <th>Kelas</th>
        <th>Jenis Kelamin</th>
        <th>Detail</th>
    </tr>
</thead>
<tbody>
<tr>
    <td></td>
    <td></td>
    <td></td>
    <td></td>
    <td></td>
    <td></td>
</tr>
</tbody>
</table>
</div>
<!-- /.card-body -->
</div>
<!-- /.card -->
</div>
<!-- /.col -->
</div>
<!-- /.row -->
@stop
@push('js')
<!-- DataTables & Plugins -->
<script src="{{ asset('templates/backend/AdminLTE-3.1.0')
}}/plugins/datatables/jquery.dataTables.min.js"></script>
<script src="{{ asset('templates/backend/AdminLTE-3.1.0') }}/plugins/datatables-
bootstrap4/js/dataTables.bootstrap4.min.js"></script>

```

```

<script src="{{ asset('templates/backend/AdminLTE-3.1.0') }}/plugins/datatables-responsive/js/dataTables.responsive.min.js"></script>
<script src="{{ asset('templates/backend/AdminLTE-3.1.0') }}/plugins/datatables-responsive/js/responsive.bootstrap4.min.js"></script>
<script>
$(function () {
var table = $("#dataTable2").DataTable({
    processing: true,
    serverSide: true,
    "responsive": true,
    ajax: "{{ route('pembayaran.status-pembayaran') }}",
    columns: [
        {data: 'DT_RowIndex', name: 'id'},
        {data: 'nama_siswa', name: 'nama_siswa'},
        {data: 'nisn', name: 'nisn'},
        {data: 'kelas.nama_kelas', name: 'kelas.nama_kelas'},
        {data: 'jenis_kelamin', name: 'jenis_kelamin'},
        {data: 'action', name: 'action', orderable: false, searchable: true},
    ]
    });
});
</script>
@endpush
Dashboard.blade.php
@extends('layouts.backend.app')
@section('title', 'Dashboard')
@push('css')
<link rel="stylesheet" type="text/css" href="{{ asset('templates/backend/AdminLTE-3.1.0')
}}/plugins/chart.js/Chart.min.css">
@endpush
@section('content_title', 'Dashboard')

```

```

@section('content')
<!-- Small boxes (Stat box) -->
<div class="row">
  <div class="col-lg-3 col-6">
    <!-- small box -->
    <div class="small-box bg-info">
      <div class="inner">
        <h3>{{ $total_siswa }}</h3>
        <p>Siswa</p>
      </div>
      <div class="icon">
        <i class="fas fa-users"></i>
      </div>
      <a href="{{ route('siswa.index') }}" class="small-box-footer">More info <i
class="fas fa-arrow-circle-right"></i></a>
    </div>
  </div>
  <!-- ./col -->
  <div class="col-lg-3 col-6">
    <!-- small box -->
    <div class="small-box bg-danger">
      <div class="inner">
        <h3>{{ $total_kelas }}</h3>
        <p>Kelas</p>
      </div>
      <div class="icon">
        <i class="fas fa-school"></i>
      </div>
    </div>
  </div>
</div>

```



```

        <a href="{{ route('kelas.index') }}" class="small-box-footer">More info <i
class="fas fa-arrow-circle-right"></i></a>
    </div>
</div>
<!-- ./col -->
<div class="col-lg-3 col-6">
    <!-- small box -->
    <div class="small-box bg-success">
        <div class="inner">
            <h3>{{ $total_petugas }}</h3>
            <p>Petugas</p>
        </div>
        <div class="icon">
            <i class="fas fa-user-tie"></i>
        </div>
        <a href="{{ route('petugas.index') }}" class="small-box-footer">More
info <i class="fas fa-arrow-circle-right"></i></a>
    </div>
</div>
<!-- ./col -->
<div class="col-lg-3 col-6">
    <!-- small box -->
    <div class="small-box bg-warning">
        <div class="inner">
            <h3>{{ $total_admin }}</h3>
            <p>Admin</p>
        </div>
        <div class="icon">
            <i class="fas fa-user-secret"></i>
        </div>

```

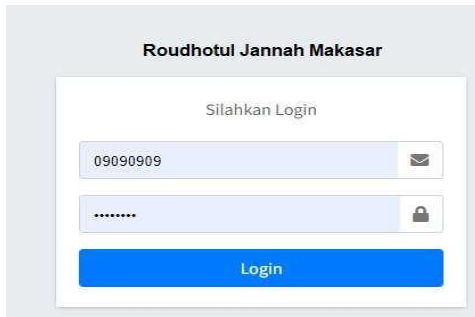
```

        <a href="{{ route('admin-list.index') }}" class="small-box-footer">More info <i
class="fas fa-arrow- circle-right"></i></a>
    </div>
</div>
</div>
<!-- /.row -->
<div class="container mt-5">
    <div class="row">
        <div class="col-md-10 offset-md-1">
            <div class="card">
                <div class="card-header">Siswa</div>
                <div class="card-body">
                    <canvas id="canvas" height="200" width="500"></canvas>
                </div>
            </div>
        </div>
    </div>
</div>
@endsection
@push('js')
<script type="text/javascript" src="{{ asset('templates/backend/AdminLTE-3.1.0')
}}/plugins/chart.js/Chart.min.js"></script>
<script>
var ctx = document.getElementById("canvas").getContext('2d');
var myChart = new Chart(ctx, {
    type: 'bar',
    data: {
        labels: ["Siswa Laki-laki", "Siswa Perempuan"],
        datasets: [{
            label: "",

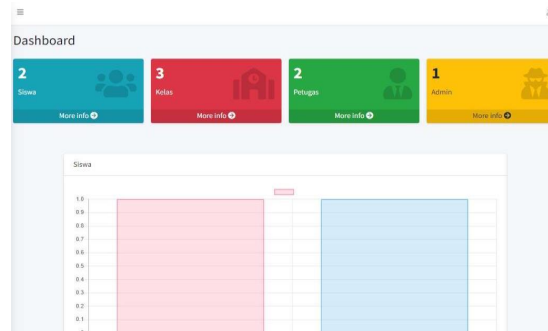
```

```
data: [  
  {!! $siswa_laki_laki !!},  
  {!! $siswa_perempuan !!},  
],  
backgroundColor: [  
  'rgba(255, 99, 132, 0.2)',  
  'rgba(54, 162, 235, 0.2)',  
],  
borderColor: [  
  'rgba(255,99,132,1)',  
  'rgba(54, 162, 235, 1)',  
],  
borderWidth: 1  
  ]]  
},  
options: {  
  scales: {  
    yAxes: [{  
      ticks: {  
        beginAtZero:true  
      }  
    }]  
  }  
}  
});  
</script>  
@endpush
```

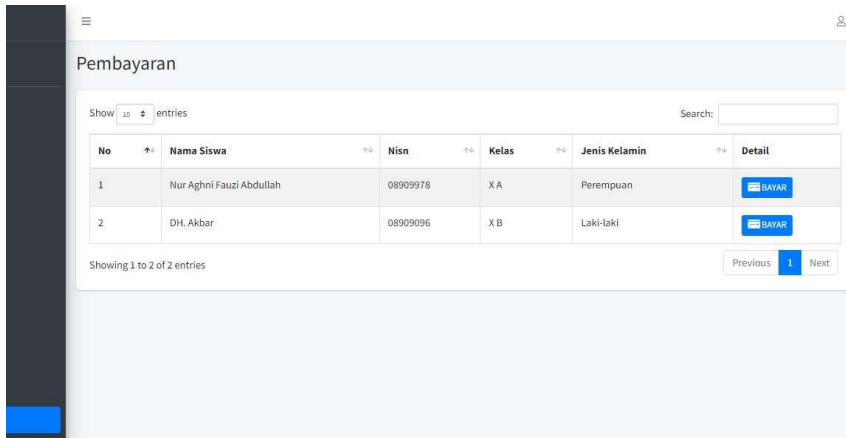
Lampiran 4 Tampilan Aplikasi



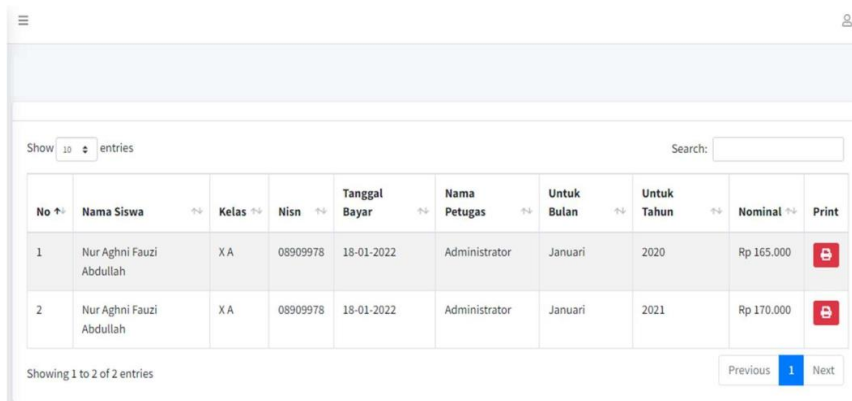
Tampilan Login



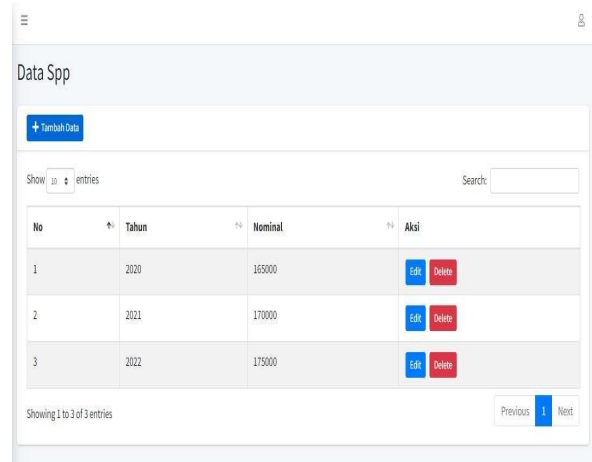
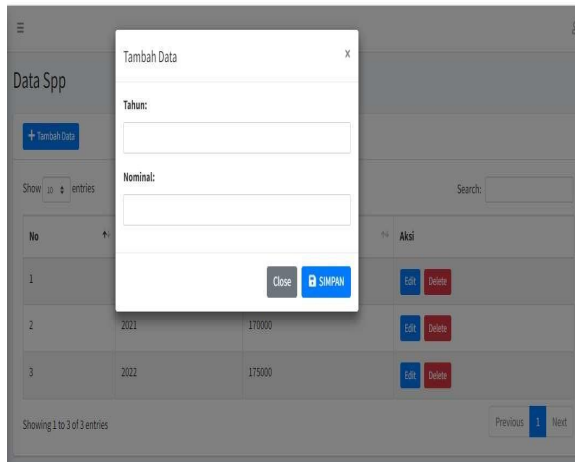
Tampilan Menu Dashboard



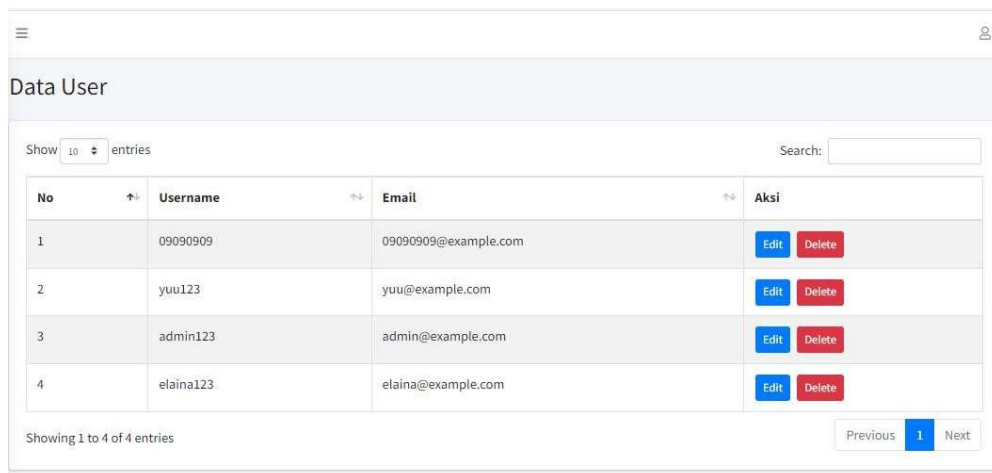
Tampilan Menu Pembayaran



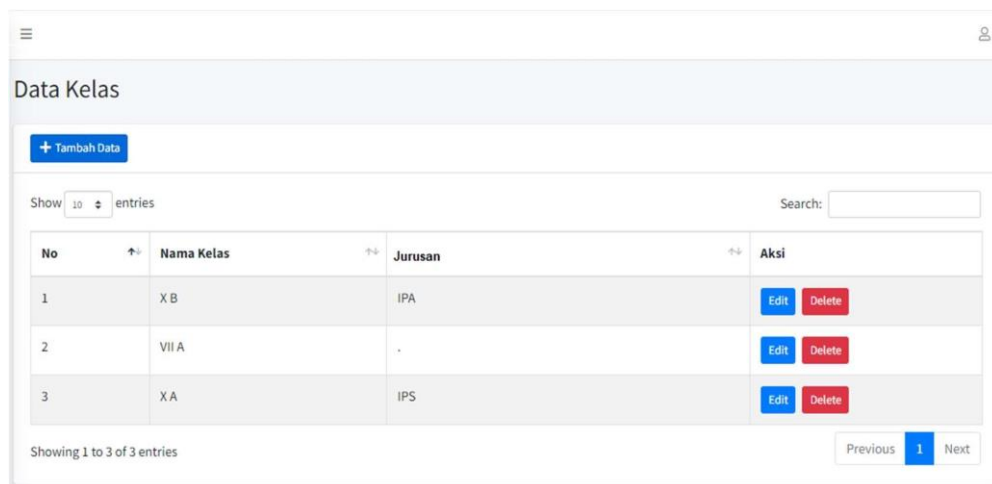
Tampilan Menu Rekapitulasi



Tampilan Menu Tahun Pembayaran



Tampilan Menu Data Pengguna



Tampilan Menu Data Kelas

Data Siswa

[+ Tambah Data](#)

Show 10 entries Search:

No	Nama Siswa	Nisn	Kelas	Jenis Kelamin	No Telepon	Aksi
1	Nur Aghni Fauzi Abdullah	08909978	X A	Perempuan	123456789	Edit Delete
2	DH. Akbar	08909096	X B	Laki-laki	08599865056	Edit Delete

Showing 1 to 2 of 2 entries [Previous](#) [1](#) [Next](#)

Tampilan Menu Data siswa

Show 10 entries Search:

No	Nama Siswa	Kelas	Nisn	Tanggal Bayar	Nama Petugas	Untuk Bulan	Untuk Tahun	Nominal	Status
1	Nur Aghni Fauzi Abdullah	X A	08909978	18-01-2022	Administrator	Januari	2021	Rp 170.000	DIBAYAR

Showing 1 to 1 of 1 entries [Previous](#) [1](#) [Next](#)

[● STATUS PEMBAYARAN](#)

Bulan	Status
Januari	DIBAYAR
Februari	BELUM DIBAYAR
Maret	BELUM DIBAYAR
April	BELUM DIBAYAR
Mei	BELUM DIBAYAR

Tampilan Menu Status Pembayaran

User Permission

[User List](#)

Show 10 entries Search:

No	Username	Role	Permission
1	admin123	admin	Handle
2	elaina123	petugas	Handle
3	09090909	siswa	Handle
4	yuu123	siswa	Handle

Showing 1 to 4 of 4 entries [Previous](#) [1](#) [Next](#)

Tampilan Menu User Permission