

L

A

M

P

I

R

A

N

```
<?php
```

```
use App\Models\Absensi;
```

```
use App\Models\Hasil;
```

```
use App\Models\NilaiLainnya;
```

```
use App\Models\Quis;
```

```
use Illuminate\Support\Facades\Storage;
```

```
function uploadImage($pDir,$idFoto,$fileFoto)
```

```
{
```

```
    if (count(Storage::files('public/'. $pDir . md5($idFoto))) > 0) {
```

```
        Storage::deleteDirectory('public/'. $pDir . md5($idFoto));
```

```
    }
```

```
    $eks = strtolower($fileFoto->extension());
```

```
    $image = $fileFoto->storeAs('public/'. $pDir . md5($idFoto), uniqid() . '.' . $eks);
```

```
    $image = explode("/", $image);
```

```
    return asset('storage/'. $pDir . $image[count($image) - 2] . '/' . $image[count($image) - 1]);
```

```
}
```

```
function nilai_lainnya($id_murid,$kelas,$semester,$tahun)
```

```
{
```

```
    $nilai = NilaiLainnya::where('kelas',$kelas)
```

```
        ->where('semester',$semester)
```

```
        ->where('tahun',$tahun)
```

```
        ->where('id_murid',$id_murid);
```

```
    if($nilai->count() <= 0){
```

```
        return 0;
```

```
    }
```

```
    return $nilai->first()->nilai;
```

```
}
```

```

function nilai_quis($id_murid,$kelas,$semester,$tahun)
{

    $quis = Quis::where('kelas',$kelas)
        ->where('semester',$semester)
        ->where('tahun',$tahun)
        ->where('status_quis','AKTIF')
        ->get();

    $nilai_total=0;
    foreach ($quis as $k1 => $v1) {
        $hasil = Hasil::select('hasils.*')
            ->join('quis','quis.id_quis','=','hasils.id_quis')
            ->where('kelas',$kelas)
            ->where('semester',$semester)
            ->where('tahun',$tahun)
            ->where('tema',$v1->tema)
            ->where('subtema',$v1->subtema)
            ->where('no_quis',$v1->no_quis)
            ->where('id_murid',$id_murid)
            ->get();

        // dump($hasil);

        $nilai = 0;
        foreach ($hasil as $k => $v) {
            $nilai += round(($v->nilai/$v->total_nilai)*100,2);
        }
        $nilai_total += round($nilai,2);
    }
    if(count($quis)<=0){
        return 0;
    }
}

```

```

    return round($nilai_total/count($quis),2);
}

function nilai_kehadiran($id_murid,$jml_pertemuan,$kelas,$semester,$tahun)
{
    // dump($id_murid);
    // dump($jml_pertemuan);
    // for ($i=01; $i <= $jml_pertemuan; $i++) {
    //     # code...
    // }

    $absen = Absensi::where('kelas',$kelas)
        ->where('semester',$semester)
        ->where('tahun',$tahun)
        ->where('id_murid',$id_murid)
        ->get();

    $nilai = 0;

    foreach ($absen as $k => $v) {
        if($v->ket=='hadir'){
            $nilai += 100;
        }elseif($v->ket=='sakit'){
            $nilai += 75;
        }elseif($v->ket=='izin'){
            $nilai += 50;
        }elseif($v->ket=='tanpa keterangan'){
            $nilai += 0;
        }
    }

    $persen = 0;

    if($nilai>0){
        $persen = ($nilai / ($jml_pertemuan*100)) * 100;
    }

    return $persen;
}

```

```
}
```

```
function ubahLinkDB($linkA) {
    // Ganti bagian yang perlu diubah
    $linkA1 = preg_replace('/^(https?:)?\\//', '', $linkA);
    $linkB = str_replace('www.dropbox.com/scl/fi/', 'dl.dropbox.com/s/', $linkA1);

    return '//'. $linkB;
}
```

```
function getDriveId($url) {
    // Cek apakah URL valid
    if (strpos($url, 'drive.google.com') === false) {
        return false;
    }

    // Cek apakah URL berisi ID
    if (strpos($url, '/d/') !== false) {
        $start = strpos($url, '/d/') + 3;
        $end = strpos($url, '/', $start);
        $id = ($end === false) ? substr($url, $start) : substr($url, $start, $end - $start);
        return $id;
    }
}
```

```
// Cek apakah URL berisi uc?id=
if (strpos($url, 'uc?id=') !== false) {
    $start = strpos($url, 'uc?id=') + 7;
    $end = strpos($url, '&', $start);
    $id = ($end === false) ? substr($url, $start) : substr($url, $start, $end - $start);
    return $id;
}
```

```

    }

    // Cek apakah URL berisi /view
    if (strpos($url, '/view') !== false) {
        $start = strpos($url, '/d/') + 3;
        $end = strpos($url, '/view', $start);
        $id = ($end === false) ? substr($url, $start) : substr($url, $start, $end - $start);
        return $id;
    }

    // Cek apakah URL berisi /edit
    if (strpos($url, '/edit') !== false) {
        $start = strpos($url, '/d/') + 3;
        $end = strpos($url, '/edit', $start);
        $id = ($end === false) ? substr($url, $start) : substr($url, $start, $end - $start);
        return $id;
    }

    return false;
}
<?php

```

```

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use App\Providers\RouteServiceProvider;
use Illuminate\Foundation\Auth\AuthenticatesUsers;

class LoginController extends Controller
{
    /*
    |-----
    */

```

```

| Login Controller
|-----
|
| This controller handles authenticating users for the application and
| redirecting them to your home screen. The controller uses a trait
| to conveniently provide its functionality to your applications.
|
*/

use AuthenticatesUsers;

/**
 * Where to redirect users after login.
 *
 * @var string
 */
protected $redirectTo = RouteServiceProvider::HOME;

/**
 * Create a new controller instance.
 *
 * @return void
 */
public function __construct()
{
    $this->middleware('guru.guest:guru', ['except' => 'logout']);
    $this->middleware('murid.guest:murid', ['except' => 'logout']);
    $this->middleware('guest')->except('logout');
}
}

```

```
<?php
```

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

```
use App\Http\Controllers\Controller;
```

```
use App\Providers\RouteServiceProvider;
```

```
use App\Models\User;
```

```
use Illuminate\Foundation\Auth\RegistersUsers;
```

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

```
use Illuminate\Support\Facades\Validator;
```

```
class RegisterController extends Controller
```

```
{
```

```
    /*
```

```
    |-----
```

```
    | Register Controller
```

```
    |-----
```

```
    |
```

```
    | This controller handles the registration of new users as well as their
```

```
    | validation and creation. By default this controller uses a trait to
```

```
    | provide this functionality without requiring any additional code.
```

```
    |
```

```
    */
```

```
    use RegistersUsers;
```

```
    /**
```

```
     * Where to redirect users after registration.
```

```
     *
```

```
     * @var string
```

```
     */
```

```

protected $redirectTo = RouteServiceProvider::HOME;

/**
 * Create a new controller instance.
 *
 * @return void
 */
public function __construct()
{
    $this->middleware('guru.guest:guru');
    $this->middleware('murid.guest:murid');
    $this->middleware('guest');
}

/**
 * Get a validator for an incoming registration request.
 *
 * @param array $data
 * @return \Illuminate\Contracts\Validation\Validator
 */
protected function validator(array $data)
{
    return Validator::make($data, [
        'name' => ['required', 'string', 'max:255'],
        'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
        'password' => ['required', 'string', 'min:8', 'confirmed'],
    ]);
}

/**
 * Create a new user instance after a valid registration.

```

```

*
* @param array $data
* @return \App\Models\User
*/
protected function create(array $data)
{
    return User::create([
        'name' => $data['name'],
        'email' => $data['email'],
        'password' => Hash::make($data['password']),
    ]);
}
}

```

```
<?php
```

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

```

use App\Http\Controllers\Controller;
use App\Models\Absensi;
use App\Murid;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Crypt;
use Illuminate\Support\Facades\DB;

```

```

class AbsensiController extends Controller
{
    public function __construct()
    {
        $this->middleware('guru.auth:guru');
        $this->dataName = 'Absensi';
    }
}

```

```

$this->dataContent['title'] = 'Absensi';
}

public function index() {
    $this->dataContent['subtitle'] = 'Daftar Absensi';
    $this->dataContent['kelas'] = Absensi::select('kelas')->groupBy('kelas')->get();
    $this->dataContent['semester'] = Absensi::select('semester')->groupBy('semester')->get();
    $this->dataContent['tahun'] = Absensi::select('tahun')->groupBy('tahun')->get();
    return view('guru.absensi.list',$this->dataContent);
}

public function load_absensi(Request $request) {
    $murid = Murid::where('kelas',$request->kelas)
        ->where('semester',$request->semester)
        ->where('tahun',$request->tahun);
    if($murid->count() <= 0){
        return '<div class="card mb-4">
            <div class="card-body">
                <b>Murid tidak ditemukan!<b>
            </div>
        </div>';
    }
    $absensi = Absensi::where('kelas',$request->kelas)
        ->where('semester',$request->semester)
        ->where('tahun',$request->tahun)
        ->orderBy('pertemuan','desc');
    $jml_pertemuan=1;
    if($absensi->count() > 0){
        $jml_pertemuan=$absensi->first()->pertemuan;
    }
    $this->dataContent['jml_pertemuan'] = $jml_pertemuan;
}

```

```

$this->dataContent['murid'] = $murid->get();
$this->dataContent['kelas'] = $request->kelas;
$this->dataContent['semester'] = $request->semester;
$this->dataContent['tahun'] = $request->tahun;
return view('guru.absensi.load_absensi',$this->dataContent);
}

```

```

public function add()
{
    $this->dataContent['subtitle']='Tambah Absensi';
    return view('guru.absensi.add',$this->dataContent);
}

```

```

public function add2($data)
{
    try {
        $data = Crypt::decrypt($data);
    } catch (\Throwable $th) {
        return back()->withErrors(['Data ' . $this->dataName . ' gagal di dekripsi!']);
    }
}

```

```

$murid = Murid::where('id', $data['id_murid'])->first();
if ($murid == null) {
    return back()->withErrors(['Data Murid tidak ditemukan!']);
}

```

```

$this->dataContent['subtitle']='Tambah Absensi';
$this->dataContent['absensi']=(object)$data;
$this->dataContent['murid']=$murid;
return view('guru.absensi.add2',$this->dataContent);
}

```

```

public function load_murid(Request $request)
{
    $murid = Murid::where('kelas',$request->kelas)
        ->where('semester',$request->semester)
        ->where('tahun',$request->tahun);
    if($murid->count() <= 0){
        return '<div class="card mb-4">
            <div class="card-body">
                <b>Murid tidak ditemukan!<b>
            </div>
        </div>';
    }
    $this->dataContent['murid'] = $murid->get();

    return view('guru.absensi.load_murid', $this->dataContent);
}

public function add_process(Request $request)
{
    // dd($request->all());

    $absensi = Absensi::where('pertemuan',$request->pertemuan)
        ->where('kelas',$request->kelas)
        ->where('semester',$request->semester)
        ->where('tahun',$request->tahun);
    if($absensi->count()>0){
        return back()->withErrors(['Data ' . $this->dataName . ' sudah Ada!']);
    }

    $data['status'] = false;

```

```

DB::beginTransaction();

try {

    foreach ($request->murid as $k => $v) {

        $murid=json_decode($v);

        $ket=$request->ket[$k];

        $datasave = [

            'id_murid' => $k,

            'ket' => $ket,

            'pertemuan' => $request->pertemuan,

            'kelas' => $request->kelas,

            'semester' => $request->semester,

            'tahun' => $request->tahun,

        ];

        $create = Absensi::create($datasave);

        if ($create) {

            $data['status'] = true;

            DB::commit();

        } else {

            DB::rollback();

            $data['status'] = false;

            $data['response'] = [

                'message' => 'Data Absensi gagal disimpan!',

                'error' => $v

            ];

            break;

        }

    }

} catch (\Throwable $th) {

    DB::rollback();

```

```

        $data['status'] = false;
        $data['response'] = [
            'message' => 'Data Absensi gagal disimpan!',
            'error' => json_encode($th)
        ];
    }

    if($data['status']){
        return redirect()->route('guru.absensi')->with('message', $this->dataName . ' berhasil
disimpan!');
    }else{
        return back()->withErrors(['Data ' . $this->dataName . ' gagal disimpan!']);
    }
}

public function edit($id)
{
    try {
        $id = Crypt::decrypt($id);
    } catch (\Throwable $th) {
        return back()->withErrors(['Data ' . $this->dataName . ' gagal di dekripsi!']);
    }

    $absensi = Absensi::with('Murid')->where('id_absensi', $id)->first();
    if ($absensi == null) {
        return back()->withErrors(['Data ' . $this->dataName . ' tidak ditemukan!']);
    }

    $this->dataContent['absensi'] = $absensi;
    $this->dataContent['subtitle']='Ubah Absensi';
    return view('guru.absensi.edit', $this->dataContent);
}

```

```

public function add2_process(Request $request, $id)
{
    try {
        $id_murid = Crypt::decrypt($id);
    } catch (\Throwable $th) {
        return back()->withErrors(['ID Murid gagal di dekripsi!']);
    }

    $murid = Murid::where('id', $id_murid);
    if ($murid->first() == null) {
        return back()->withErrors(['Data Murid tidak ditemukan!']);
    }

    $datasave = [
        'id_murid' => $id_murid,
        'ket' => $request->ket,
        'pertemuan' => $request->pertemuan,
        'kelas' => $request->kelas,
        'semester' => $request->semester,
        'tahun' => $request->tahun,
    ];

    $create = Absensi::create($datasave);

    if ($create) {
        return redirect()->route('guru.absensi')->with('message', $this->dataName . ' berhasil
disimpan!');
    } else {
        return back()->withErrors(['Data ' . $this->dataName . ' gagal disimpan!']);
    }
}

```

```

public function edit_process(Request $request, $id)
{
    try {
        $id = Crypt::decrypt($id);
    } catch (\Throwable $th) {
        return back()->withErrors(['Data ' . $this->dataName . ' gagal di dekripsi!']);
    }

    $absensi = Absensi::where('id_absensi', $id);
    if ($absensi->first() == null) {
        return back()->withErrors(['Data ' . $this->dataName . ' tidak ditemukan!']);
    }

    $datasave = [
        'ket' => $request->ket,
    ];

    // if($request->password!=" && $request->password!=null){
    //     $datasave = Hash::make($request->password);
    // }

    $update = $absensi->update($datasave);

    if ($update) {
        return redirect()->route('guru.absensi')->with('message', $this->dataName . ' berhasil
disimpan!');
    } else {
        return back()->withErrors(['Data ' . $this->dataName . ' gagal disimpan!']);
    }
}

public function delete_process(Request $request)

```

```

{
    try {
        $id = Crypt::decrypt($request->id_absensi);
    } catch (\Throwable $th) {
        return back()->withErrors(['Data ' . $this->dataName . ' gagal di dekripsi!']);
    }

    $absensi = Absensi::where('id_absensi', $id);
    if ($absensi->first() == null) {
        return back()->withErrors(['Data ' . $this->dataName . ' tidak ditemukan!']);
    }

    $del = $absensi->delete();

    if ($del) {
        return redirect()->route('guru.absensi')->with('message', $this->dataName . ' berhasil
dihapus!');
    } else {
        return back()->withErrors(['Data ' . $this->dataName . ' gagal dihapus!']);
    }
}
}

```

```
<?php
```

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

```
use App\Http\Controllers\Controller;
```

```
use App\Http\Resources\Guru\HasilResource;
```

```
use App\Models\Hasil;
```

```
use App\Models\Quis;
```

```
use Illuminate\Http\Request;
```

```

use Illuminate\Support\Facades\Crypt;
use Yajra\DataTables\Facades\DataTables;

class HasilController extends Controller
{
    public function __construct()
    {
        $this->middleware('guru.auth:guru');
        $this->dataName = 'Hasil';
        $this->dataContent['title'] = 'Hasil';
    }

    public function index() {
        $this->dataContent['subtitle'] = 'Daftar Hasil';
        $quis = Quis::latest()->get();
        $this->dataContent['quis']=$quis;
        return view('guru.hasil.list',$this->dataContent);
    }

    public function list(Request $request,$id)
    {
        if ($request->ajax()) {
            $data = Hasil::with('Murid','Quis')
                ->where('id_quis',$id)
                ->orderBy('benar','DESC')
                ->orderBy('waktu_selesai','ASC')
                ->get();

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

```

```

        $actionBtn = '<div style="white-space: nowrap;"><a href="javascript:void(0)"
onclick="hapus(\''.Crypt::encrypt($row['id_hasil']).'\') class="delete btn btn-danger btn-sm"><i
class="bx bx-trash"></i></a><div>';

```

```

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

```

```

public function delete_process(Request $request)
{
    try {
        $id = Crypt::decrypt($request->id_hasil);
    } catch (\Throwable $th) {
        return back()->withErrors(['Data ' . $this->dataName . ' gagal di dekripsi!']);
    }

    $hasil = Hasil::where('id_hasil', $id);
    if ($hasil->first() == null) {
        return back()->withErrors(['Data ' . $this->dataName . ' tidak ditemukan!']);
    }

    $del = $hasil->delete();

    if ($del) {
        return redirect()->route('guru.hasil')->with('message', $this->dataName . ' berhasil
dihapus!');
    } else {
        return back()->withErrors(['Data ' . $this->dataName . ' gagal dihapus!']);
    }
}
}

```

```

<?php

namespace App\Http\Controllers\Guru;

use App\Http\Controllers\Controller;
use App\Http\Resources\Guru\HasilLatihanUjianResource;
use App\Models\HasilLatihanUjian;
use App\Models\LatihanUjian;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Crypt;
use Yajra\DataTables\Facades\DataTables;

class HasilLatihanUjianController extends Controller
{
    public function __construct()
    {
        $this->middleware('guru.auth:guru');
        $this->dataName = 'Hasil Latihan Ujian';
        $this->dataContent['title'] = 'Hasil Latihan Ujian';
    }

    public function index() {
        $this->dataContent['subtitle'] = 'Daftar Hasil Latihan Ujian';
        $latihan_ujian = LatihanUjian::latest()->get();
        $this->dataContent['latihan_ujian']=$latihan_ujian;
        return view('guru.hasil_latihan_ujian.list',$this->dataContent);
    }

    public function list(Request $request,$id)
    {

```

```

if ($request->ajax()) {
    $data = HasilLatihanUjian::with('Murid', 'LatihanUjian')
        ->where('id_latihan_ujian', $id)
        ->orderBy('benar', 'DESC')
        ->orderBy('waktu_selesai', 'ASC')
        ->get();

    return Datatables::of(HasilLatihanUjianResource::collection($data))
        ->addIndexColumn()
        ->addColumn('action', function($row){
            $actionBtn = '<div style="white-space: nowrap;"><a href="javascript:void(0)"
onclick="hapus(\".Crypt::encrypt($row[\'id_hasil_latihan_ujian\'])."\") class="delete btn btn-danger
btn-sm"><i class="bx bx-trash"></i></a></div>';

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

public function delete_process(Request $request)
{
    try {
        $id = Crypt::decrypt($request->id_hasil_latihan_ujian);
    } catch (\Throwable $th) {
        return back()->withErrors(['Data ' . $this->dataName . ' gagal di dekripsi!']);
    }

    $hasil_latihan_ujian = HasilLatihanUjian::where('id_hasil_latihan_ujian', $id);
    if ($hasil_latihan_ujian->first() == null) {
        return back()->withErrors(['Data ' . $this->dataName . ' tidak ditemukan!']);
    }
}

```

```

$del = $hasil_latihan_ujian->delete();

if ($del) {
    return redirect()->route('guru.hasil_latihan_ujian')->with('message', $this->dataName . '
berhasil dihapus!');
} else {
    return back()->withErrors(['Data ' . $this->dataName . ' gagal dihapus!']);
}
}
}
}

```

```
<?php
```

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

```
use App\Http\Controllers\Controller;
```

```
use App\Models\Absensi;
```

```
use App\Models\Materi;
```

```
use App\Models\Quis;
```

```
use App\Models\Soal;
```

```
use App\Murid;
```

```
use Illuminate\Support\Facades\Auth;
```

```
use Illuminate\Http\Request;
```

```
class HomeController extends Controller
```

```

{
    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()

```

```

{
    $this->middleware('guru.auth:guru');
    $this->dataName = 'Dashboard';
    $this->dataContent['title'] = 'Dashboard';
}

/**
 * Show the Guru dashboard.
 *
 * @return \Illuminate\Contracts\Support\Renderable
 */
public function index() {
    $this->dataContent['user']=Auth::guard('guru')->user();
    $hari = date('w');
    switch ($hari) {
        case 0:
            $hari = 'Minggu';
            break;
        case 1:
            $hari = 'Senin';
            break;
        case 2:
            $hari = 'Selasa';
            break;
        case 3:
            $hari = 'Rabu';
            break;
        case 4:
            $hari = 'Kamis';
            break;
        case 5:

```

```

        $hari = 'Jumat';
        break;
    case 6:
        $hari = 'Sabtu';
        break;

    default:
        break;
}

$this->dataContent['tgl'] = $hari.', '.date('d/m/Y');
$this->dataContent['murid'] = Murid::count();
$this->dataContent['quis'] = Quis::count();
$this->dataContent['materi'] = Materi::count();
$absensi = Absensi::select('kelas','semester','tahun')
    ->groupBy('kelas','semester','tahun')
    ->get();
$this->dataContent['absensi']=count($absensi);

$this->dataContent['kelas'] = Murid::select('kelas')->groupBy('kelas')->get();
$this->dataContent['semester'] = Murid::select('semester')->groupBy('semester')->get();
$this->dataContent['tahun'] = Murid::select('tahun')->groupBy('tahun')->get();
return view('guru.home',$this->dataContent);
}

public function grafik(Request $request) {
    $data['status']=true;
    $murid = Murid::where('kelas',$request->kelas)
        ->where('semester',$request->semester)
        ->where('tahun',$request->tahun);
    if($murid->count() <= 0){
        $data['status']=false;
    }
}

```

```

        $data['response']['label'][];
        $data['response']['data1'][];
        $data['response']['data2'][];
        return json_encode($data);
    }

    $absensi = Absensi::where('kelas',$request->kelas)
        ->where('semester',$request->semester)
        ->where('tahun',$request->tahun)
        ->orderBy('pertemuan','desc');

    $jml_pertemuan=1;
    if($absensi->count() > 0){
        $jml_pertemuan=$absensi->first()->pertemuan;
    }

    // $data['data']=$murid->get()->toArray();
    $label=[];
    $data1=[];
    $data2=[];
    foreach ($murid->get() as $k => $v) {
        $label[$k]=$v->name;

        $nilai_kehadiran = nilai_kehadiran($v->id,$jml_pertemuan,$request->kelas,$request->semester,$request->tahun);
        $nilai_quis = nilai_quis($v->id,$request->kelas,$request->semester,$request->tahun);
        $nilai_lainnya = nilai_lainnya($v->id,$request->kelas,$request->semester,$request->tahun);

        $total_nilai = (0.4 * $nilai_kehadiran) + (0.3 * $nilai_quis) + (0.3 * $nilai_lainnya);
        $total_nilai = round($total_nilai,2);
        $data1[$k]=$total_nilai;

        $data2[$k]=$v->login;
    }

```

```

    }
    $data['response']['label']=$label;
    $data['response']['data1']=$data1;
    $data['response']['data2']=$data2;

    return json_encode($data);

    // return view('guru.grafik',$this->dataContent);
}
}

```

```
<?php
```

```

namespace App\Http\Controllers\Guru;

use App\Http\Controllers\Controller;
use App\Models\Absensi;
use App\Models\NilaiLainnya;
use App\Murid;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Crypt;
use Illuminate\Support\Facades\DB;

class InputNilaiController extends Controller
{
    public function __construct()
    {
        $this->middleware('guru.auth:guru');
        $this->dataName = 'Input Nilai';
    }
}

```

```

$this->dataContent['title'] = 'Input Nilai';
}

public function index() {
    $this->dataContent['subtitle'] = "";
    $this->dataContent['kelas'] = Murid::select('kelas')->groupBy('kelas')->get();
    $this->dataContent['semester'] = Murid::select('semester')->groupBy('semester')->get();
    $this->dataContent['tahun'] = Murid::select('tahun')->groupBy('tahun')->get();
    return view('guru.input_nilai.list',$this->dataContent);
}

public function load_penilaian(Request $request) {
    $murid = Murid::where('kelas',$request->kelas)
        ->where('semester',$request->semester)
        ->where('tahun',$request->tahun);
    if($murid->count() <= 0){
        return '<div class="card mb-4">
            <div class="card-body">
                <b>Murid tidak ditemukan!<b>
            </div>
        </div>';
    }
    $absensi = Absensi::where('kelas',$request->kelas)
        ->where('semester',$request->semester)
        ->where('tahun',$request->tahun)
        ->orderBy('pertemuan','desc');
    $jml_pertemuan=1;
    if($absensi->count() > 0){
        $jml_pertemuan=$absensi->first()->pertemuan;
    }
    $this->dataContent['jml_pertemuan'] = $jml_pertemuan;
}

```

```

$this->dataContent['murid'] = $murid->get();
$this->dataContent['kelas'] = $request->kelas;
$this->dataContent['semester'] = $request->semester;
$this->dataContent['tahun'] = $request->tahun;
return view('guru.input_nilai.load_penilaian',$this->dataContent);
}

public function add_nilai_lainnya(Request $request,$kelas,$semester,$tahun)
{
    $status = true;
    DB::beginTransaction();
    foreach ($request->nilai_lainnya as $id_murid => $nilai) {
        $datasave['id_murid']=$id_murid;
        $datasave['nilai']=$nilai;
        $datasave['kelas']=$kelas;
        $datasave['semester']=$semester;
        $datasave['tahun']=$tahun;

        $nilai_lainnya = NilaiLainnya::where('kelas',$kelas)
            ->where('semester',$semester)
            ->where('tahun',$tahun)
            ->where('id_murid',$id_murid);
        if($nilai_lainnya->count() <= 0){
            $save = NilaiLainnya::create($datasave);
            if(!$save){
                $status = false;
                break;
            }
        }else{
            $save = $nilai_lainnya->update($datasave);
            if(!$save){

```

```

        $status = false;
        break;
    }
}
}

if($status){
    DB::commit();
    return redirect()->route('guru.input_nilai')->with('message', 'Data Nilai Lainnya berhasil
disimpan!');
}else{
    DB::rollback();
    return back()->withErrors(['Data Nilai Lainnya gagal disimpan!']);
}
}

```

```
<?php
```

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

```
use App\Http\Controllers\Controller;
```

```
use App\Http\Resources\Guru\KomentarResource;
```

```
use App\Models\Komentar;
```

```
use Illuminate\Http\Request;
```

```
use Illuminate\Support\Facades\Crypt;
```

```
use Yajra\DataTables\Facades\DataTables;
```

```
class KomentarController extends Controller
```

```

{
    public function __construct()
    {
        $this->middleware('guru.auth:guru');
    }
}

```

```

        $this->dataName = 'Komentar';
        $this->dataContent['title'] = 'Komentar';
    }

    public function index() {
        $this->dataContent['subtitle'] = 'Daftar Komentar';
        return view('guru.komentar.list',$this->dataContent);
    }

    public function list(Request $request)
    {
        if ($request->ajax()) {
            $data = Komentar::with('Murid')->latest()->get();
            return Datatables::of(KomentarResource::collection($data))
                ->addIndexColumn()
                ->addColumn('action', function($row){
                    $actionBtn = '<div style="white-space: nowrap;"><a href="javascript:void(0)"
onclick="hapus(\'".Crypt::encrypt($row['id_komentar'])."\")" class="delete btn btn-danger btn-
sm"><i class="bx bx-trash"></i></a><div>';
                    return $actionBtn;
                })
                ->rawColumns(['action'])
                ->make(true);
        }
    }

    public function delete_process(Request $request)
    {
        try {
            $id = Crypt::decrypt($request->id_murid);
        } catch (\Throwable $th) {
            return back()->withErrors(['Data ' . $this->dataName . ' gagal di dekripsi!']);
        }
    }

```

```

    }

    $komentar = Komentar::where('id', $id);
    if ($komentar->first() == null) {
        return back()->withErrors(['Data ' . $this->dataName . ' tidak ditemukan!']);
    }

    $del = $komentar->delete();

    if ($del) {
        return redirect()->route('guru.komentar')->with('message', $this->dataName . ' berhasil
dihapus!');
    } else {
        return back()->withErrors(['Data ' . $this->dataName . ' gagal dihapus!']);
    }
}
}
}

```

```
<?php
```

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

```
use App\Http\Controllers\Controller;
```

```
use App\Models\Absensi;
```

```
use App\Murid;
```

```
use PDF;
```

```
use Illuminate\Http\Request;
```

```
class LaporanNilaiController extends Controller
```

```

{
    public function __construct()
    {

```

```

$this->middleware('guru.auth:guru');
$this->dataName = 'Laporan Nilai';
$this->dataContent['title'] = 'Laporan Nilai';
}

public function index() {
    $this->dataContent['subtitle'] = "";
    $this->dataContent['kelas'] = Murid::select('kelas')->groupBy('kelas')->get();
    $this->dataContent['semester'] = Murid::select('semester')->groupBy('semester')->get();
    $this->dataContent['tahun'] = Murid::select('tahun')->groupBy('tahun')->get();
    return view('guru.laporan_nilai.list',$this->dataContent);
}

public function pdf($data) {
    $data=base64_decode($data);
    $data=json_decode($data,true);
    // dd($data);
    $date = date('d M Y');
    $bulan = array(
        'Jan' => 'Januari',
        'Feb' => 'Februari',
        'Mar' => 'Maret',

```

```

'Apr' => 'April',
'May' => 'Mei',
'Jun' => 'Juni',
'Jul' => 'Juli',
'Aug' => 'Agustus',
'Sep' => 'September',
'Oct' => 'Oktober',
'Nov' => 'November',
'Dec' => 'Desember'
);

```

```

$murid = Murid::where('kelas',$data['kelas'])
    ->where('semester',$data['semester'])
    ->where('tahun',$data['tahun']);
if($murid->count() <= 0){
    return '<div class="card mb-4">
        <div class="card-body">
            <b>Murid tidak ditemukan!<b>
        </div>
    </div>';
}
$absensi = Absensi::where('kelas',$data['kelas'])
    ->where('semester',$data['semester'])
    ->where('tahun',$data['tahun'])
    ->orderBy('pertemuan','desc');
$jml_pertemuan=1;
if($absensi->count() > 0){
    $jml_pertemuan=$absensi->first()->pertemuan;
}

```

```

$ranking=[];
foreach ($murid->get() as $k => $v){
    $nilai_kehadiran = nilai_kehadiran($v-
>id,$jml_pertemuan,$data['kelas'],$data['semester'],$data['tahun']);
    $nilai_quis = nilai_quis($v->id,$data['kelas'],$data['semester'],$data['tahun']);
    $nilai_lainnya = nilai_lainnya($v->id,$data['kelas'],$data['semester'],$data['tahun']);
    $total_nilai = (0.4 * $nilai_kehadiran) + (0.3 * $nilai_quis) + (0.3 * $nilai_lainnya);
    $total_nilai = round($total_nilai,2);

    $nilai_murid=$v->toArray();
    $nilai_murid['total_nilai']=$total_nilai;
    $ranking[$k]=$nilai_murid;
}

// dump(json_encode($ranking));
usort($ranking, function($a, $b) {
    return $b['total_nilai'] - $a['total_nilai'];
});

$this->dataContent['ranking'] = $ranking;
$this->dataContent['data'] = [
    'tahun' => $data['tahun'],
    'kelas' => $data['kelas'],
    'semester' => $data['semester'],
    'guru' => 'Saskia',
    'tempat_ttd' => 'Parepare',
    'tgl_ttd' => strtr($date, $bulan),
];

// dd($this->dataContent);

```

```

        $pdf = PDF::loadView('guru.laporan_nilai.pdf', $this->dataContent);
        return $pdf->stream();
    }
}

```

```
<?php
```

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

```

use App\Http\Controllers\Controller;
use App\Http\Requests\Guru\LatihanUjianRequest;
use App\Http\Resources\Guru\LatihanUjianResource;
use App\Models\LatihanUjian;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Crypt;
use Yajra\DataTables\Facades\DataTables;

```

```
class LatihanUjianController extends Controller
```

```

{
    public function __construct()
    {
        $this->middleware('guru.auth:guru');
        $this->dataName = 'Latihan Ujian';
        $this->dataContent['title'] = 'Latihan Ujian';
    }

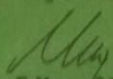
    public function index() {
        $this->dataContent['subtitle'] = 'Daftar Latihan Ujian';
        return view('guru.latihan_ujian.list',$this->dataContent);
    }
}

```


Lampiran

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 6 Revisi Sistem hasil Kode Program harus sesuai, fungsi nama konstanta, variabel dan konstanta, Pada ini ada, quiz Parameter, kompilan Ulat berburu dan rekaman, hasil Foto		Konsultasi 6 Tambah rekapan nilai pd aplikasi	
Konsultasi 7 Sesuaikan revisi dengan dokumen, gunakan foto Profil sendiri sebagai intelektual		Konsultasi 7 Daftar pustaka dilengkapi sesuai dgn bahasan	
Konsultasi 8 UPIL 3 Xchivly Diagram 4 ... dbase		Konsultasi 8 Lec	
Konsultasi 9		Konsultasi 9	
Konsultasi 10		Konsultasi 10	

Mengetahui
Ketua Program Studi


Marlina, S.Kom., M.Kom.
NBM. 1162 680

Parepare,

Mahasiswa

Nurfadillah
NIM. 217280090

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak timbal balik

KARTU MONITORING BIMBINGAN
 MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
 FAKULTAS TEKNIK
 UNIVERSITAS MUHAMMADIYAH PAREPARE

SKRIPSI

Mahasiswa : Nurfaidillah	Pembimbing I : Ade Hastuty, S.T., S.Kom., M.T
NIM : 217280090	Pembimbing II : Andi Wafiah, S.Kom., M.Kom
Judul Skripsi : Aplikasi Media Pembelajaran Pada Buku Tematik Kelas 3 SDN 142 Kajuangin Berbasis Website	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 Abstrak 350 kata		Konsultasi 1 Tambahkan Nilai Siswa di akun Siswa.	
Konsultasi 2 Tambahkan rumusan masalah di awal di kesimpulan		Konsultasi 2 Tambahkan nilai siswa di Admin	
Konsultasi 3 di blackBox dan whiteBox ~~di blackBox dan whiteBox~~ ~~di blackBox dan whiteBox~~		Konsultasi 3 Tambahkan Laporan Nilai ujian siswa	
Konsultasi 4		Konsultasi 4 ACC Ujian Hasil	
Konsultasi 5		Konsultasi 5	

Lanjut ke halaman sebelah...