

DAFTAR PUSTAKA

- Anggraeni, E. Y., & Irviani, R. (2017). *Pengantar Sistem Informasi*. CV. Andi Offset.
- Azwar, A. (1996). *Pengantar Administrasi Kesehatan*. Jakarta: Binarupa Aksara.
- Hatta, G. (2012). *Pedoman Manajemen Informasi Kesehatan di Sarana Pelayanan Kesehatan*. Jakarta: UI Presss.
- Helling, L. S., Wahyudi, E., & Hasanudin. (2019). Siremis : Sistem Informasi Rekam Medis Puskesmas Kecamatan Matraman Jakarta. *INTENSIF: Jurnal Ilmiah Penelitian Dan Penerapan Teknologi Sistem Informasi*, 3(2), 116–129.
- Iflahah, D., Aknuranda, I., & Setiawan, N. Y. (2018). Analisis dan Perancangan Sistem Informasi Rekam Medis Poli Gigi (Studi Kasus : Puskesmas Sumbersari Kecamatan Saradan Kabupaten Madiun). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(6).
- Jogiyanto. (2005). *Analisis & Desain : Sistem Informasi : Pendekatan Terstruktur Teori dan Praktik Aplikasi Bisnis*. Yogyakarta : Andi Offset.
- Mulyani, S. (2016). *Metode Analisis dan Perancangan Sistem*. Bandung: Abdi Sistematika.
- Peraturan Menteri Kesehatan Tentang Kepemilikan Rekam Medis, (2008).
- Putra, I., Dawood, R., & Roslidar. (2017). Rancang Bangun Aplikasi Rekam Medis untuk Praktik Pribadi Dokter Berbasis Android dan Memanfaatkan Layanan Web. *KITEKTRO: Jurnal Online Teknik Elektro*, 2(1).
- Rahman, T., Hakim, L., & Sari, O. M. (2020). Sistem Informasi Rekam Medis Pada Dinas Kesehatan Kabupaten Musirawas Berbasis Web Mobile. *Jurnal Teknik Informatika Musirawas*, 05(02), 141–156.
- Rahmawati, E., Wijianto, R., & Komputer, I. (2019). *Penerapan Model Waterfall Pada Pengembangan Sistem*. 15(1), 53–60.
- Raymond Mcleod, J., & Schell, G. P. (2008). *Management Information Systems*.

Jakarta : Salemba Empat.

Rustiyanto, E. (2010). *Sistem Informasi Manajemen Rumah Sakit Yang Terintegrasi*. Gosyen Publishing : Yogyakarta.

Sulhan, M. (2006). *Pengembangan Aplikasi Barbasis Web dengan PHP dan ASP*. Malang: Gava Media.

LAMPIRAN

.env

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:cYZhAvNfhOLCZQmMDJg5SeylGFmp556cFnfJ6S
0gR2I=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=sistem_rekam_medis
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
FILESYSTEM_DISK=local
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120

MEMCACHED_HOST=127.0.0.1

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379
```

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.gmail.com
MAIL_PORT=465
MAIL_USERNAME=admpuskesmaswp@gmail.com
MAIL_PASSWORD=mflbfqodjnbdxgdf
MAIL_ENCRYPTION=ssl
MAIL_FROM_ADDRESS="admpuskesmaswp@gmail.com"
MAIL_FROM_NAME="Reset Password"

AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_DEFAULT_REGION=us-east-1
AWS_BUCKET=
AWS_USE_PATH_STYLE_ENDPOINT=false

PUSHER_APP_ID=
PUSHER_APP_KEY=
PUSHER_APP_SECRET=
PUSHER_HOST=
PUSHER_PORT=443
PUSHER_SCHEME=https
PUSHER_APP_CLUSTER=mt1

VITE_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
VITE_PUSHER_HOST="${PUSHER_HOST}"
VITE_PUSHER_PORT="${PUSHER_PORT}"
VITE_PUSHER_SCHEME="${PUSHER_SCHEME}"
VITE_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
```

Penjelasan:

Untuk menghubungkan web ke database. Dan untuk mengatur Email yang akan digunakan sebagai email pengirim fitur *Forget Password*.

Database/migrations/2023_02_14_101442_pasien.php

```
<?php

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

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('pasien', function (Blueprint $table) {
            $table->id();
            $table->string('nama');
            $table->string('nKepalaKeluarga');
            $table->string('nIbuKandung');
            $table->string('kode_desa');
            $table->string('nik')->unique();
            $table->string('agama');
            $table->date('tglLahir');
            $table->char('jk'); //Jenis Kelamin
            $table->string('status');
```

```

        $table->string('pekerjaan');
        $table->string('alamat');
        $table->string('noTelp');
        $table->string('jenis_asuransi');
        $table->string('no_kartu')->nullable();
        $table->timestamps();
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('pasien');
}
};

```

Penjelasan:

Untuk membuat tabel pasien.

Database/migrations/ 2023_02_14_102424_poli.php

```

<?php

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

return new class extends Migration

```

```

{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('polis', function (Blueprint
$table) {
            $table->id();
            $table->string('nama');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('polis');
    }
};

```

Penjelasan:

Untuk membuat tabel polis.

Database/migrations/ 2023_02_14_102530_icd10.php

```

<?php

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

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('icd10s', function (Blueprint $table) {
            $table->id();
            $table->string('nama');
            $table->string('deskripsi');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('icd10s');
    }
}

```

```
}  
};
```

Penjelasan:

Untuk membuat tabel icd10s.

Database/migrations/ 2023_02_14_102724_obat.php

```
<?php  
  
use Illuminate\Database\Migrations\Migration;  
use Illuminate\Database\Schema\Blueprint;  
use Illuminate\Support\Facades\Schema;  
  
return new class extends Migration  
{  
    /**  
     * Run the migrations.  
     *  
     * @return void  
     */  
    public function up()  
    {  
        Schema::create('obats', function (Blueprint  
$table) {  
            $table->id();  
            $table->string('nama');  
            $table->string('jenis_obat');  
            $table->string('deskripsi');  
            $table->timestamps();  
        });  
    }  
}
```

```

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('obats');
}
};

```

Penjelasan:

Untuk membuat tabel obats.

Database/migrations/ 2023_02_14_122533_user.php

```

<?php

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

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {

```

```

        Schema::create('users', function (Blueprint
$table) {
            $table->id();
            $table->string('nama');
            $table->string('email')->unique();
            $table->string('password');
            $table->string('jabatan');
            $table->unsignedBigInteger('poli_id')-
>nullable();
            $table->timestamps();
            $table->foreign('poli_id')
                ->references('id')->on('polis')-
>onDelete('cascade');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('users');
    }
};

```

Penjelasan:

Untuk membuat tabel users.

Database/migrations/ 2023_02_14_122912_keluhan.php

```

<?php

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

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('keluhans', function (Blueprint $table) {
            $table->id();
            $table->unsignedBigInteger('pasien_id');
            $table->unsignedBigInteger('icd10_id')->nullable();
            $table->unsignedBigInteger('dokter_id');
            $table->unsignedBigInteger('poli_id');
            $table->string('keluhan');
            $table->string('tindakan')->nullable();
            $table->timestamps();
            $table->foreign('poli_id')
                ->references('id')->on('polis')->onDelete('cascade');
            $table->foreign('pasien_id')
                ->references('id')->on('pasien')->onDelete('cascade');
            $table->foreign('icd10_id')

```

```

        ->references('id')->on('icd10s')->onDelete('cascade');

        $table->foreign('dokter_id')
            ->references('id')->on('users')->onDelete('cascade');
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('keluhans');
}
};

```

Penjelasan:

Untuk membuat tabel keluhan.

Database/migrations/ 2023_02_16_065222_obat_keluhan.php

```

<?php

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

return new class extends Migration
{
    /**

```

```

    * Run the migrations.
    *
    * @return void
    */
    public function up()
    {
        Schema::create('keluhan_obat', function (Blueprint $table) {
            $table->id();
            $table->unsignedBigInteger('keluhan_id');
            $table->unsignedBigInteger('obat_id')->nullable();
            $table->timestamps();
            $table->foreign('keluhan_id')
                ->references('id')->on('keluhans')->onDelete('cascade');
            $table->foreign('obat_id')
                ->references('id')->on('obats')->onDelete('cascade');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('obat_keluhans');
    }
};

```

Penjelasan:

Untuk membuat tabel obat_keluhans.

App/Http/Controllers/Admin/adminController.php

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\dokter as Admin;
use Illuminate\Support\Facades\Auth;

class adminController extends Controller
{
    public function showProfileAdmin()
    {
        $data = Admin::where('id', '=', Auth::user()->id)->get();
        return view('admin.menu.inner-menu.profil')
            ->with(compact('data'));
    }

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

        $request->validate([
            'nama' => 'required',
            'email' => 'required|email|unique:users,email, '.$id,
        ], [
            'email.unique' => 'Email Sudah Terdaftar!
Gunakan Email Lain!'
        ]
    )
    }
```

```

    });

    $Admin = Admin::find($id);
    $Admin->nama = $request->nama;
    $Admin->email = $request->email;
    $Admin->save();

    return back()->with('success', 'Data Anda
Berhasil Diedit!');

}

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

    $request->validate([
        'password' => 'required',
    ]);

    $Admin = Admin::find($id);
    $Admin->password = bcrypt($request-
>password);
    $Admin->save();

    return back()->with('success', 'Password Anda
Berhasil Diubah!');

}

}

```

Penjelasan:

Untuk menampilkan dan mengedit data Admin.

App/Http/Controllers/Dokter/dokterController.php

```
<?php

namespace App\Http\Controllers\Dokter;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\dokter;
use App\Models\keluhan;
use DB;
use Illuminate\Support\Facades\Auth;

class dokterController extends Controller
{

    // Interaksi Admin Ke Dokter

    public function show()
    {

        $data = dokter::where('jabatan', '=', 'dokter')->get();
        $data2 = DB::table('polis')->get();
        return view('admin.menu.dataDokter')
            ->with(compact('data'))
            ->with(compact('data2'));
    }

    public function create(Request $request)
    {

        $request->validate([
```

```

        'nama' => 'required',
        'email' => 'required|unique:users',
        'password' => 'required',
        'poli' => 'required',
    ], [
        'email.unique' => 'Email Sudah Terdaftar!
Gunakan Email Lain!'
    ]);

    $dokter = new dokter();
    $dokter->nama = $request->nama;
    $dokter->email = $request->email;
    $dokter->password = bcrypt($request-
>password);
    $dokter->jabatan = "dokter";
    $dokter->poli_id = $request->poli;
    $dokter->save();

    return redirect()->to('/dataDokter')->send()-
>with('success', 'Data Dokter Berhasil Ditambahkan!');
}

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

    $request->validate([
        'nama' => 'required',
        'email' => 'required|unique:users,email, '.$id,
        'poli' => 'required',
    ], [
        'email.unique' => 'Email Sudah Terdaftar!
Gunakan Email Lain!'
    ]);
}

```

```

        $dokter = dokter::find($id);
        $dokter->nama = $request->nama;
        $dokter->email = $request->email;
        $dokter->poli_id = $request->poli;
        $dokter->save();

        return redirect()->to('/profilDokter/'.$id)-
>send()->with('success', 'Data Dokter Berhasil
Ditambahkan!');

    }

    public function detail($id)
    {
        $data = dokter::where('id','=', $id)->get();
        $data2 = keluhan::where('dokter_id', '=',
$id)->get();
        $data3 = DB::table('polis')->get();
        return view('admin.menu.inner-
menu.profilDokter')
        ->with(compact('data'))
        ->with(compact('data2'))
        ->with(compact('data3'));
    }

    public function hapus($id)
    {
        $deleted = DB::table('users')-
>where('id','=', $id)->delete();
        return redirect()->to('/dataDokter')->send()-
>with('success', 'Data Dokter Berhasil Di Hapus!');
    }

```

```

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

        $request->validate([
            'password' => 'required',
        ]);

        $dokter = dokter::find($id);
        $dokter->password = bcrypt($request-
>password);
        $dokter->save();

        return redirect()->to('/profilDokter/'.$id)-
>send()->with('success', 'Password Berhasil Diubah!');

    }

    // Interaksi Dokter Ke Dokter

    public function showProfileDokter()
    {
        $data = dokter::where('id', '=', Auth::user()-
>id)->get();
        return view('dokter.menu.inner-menu.profil')
            ->with(compact('data'));
    }

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

        $request->validate([
            'nama' => 'required',

```

```

        'email'
        =>
        'required|email|unique:users,email, '.$id,
        ], [
            'email.unique' => 'Email Sudah Terdaftar!
Gunakan Email Lain!'
        ]);

        $dokter = dokter::find($id);
        $dokter->nama = $request->nama;
        $dokter->email = $request->email;
        $dokter->save();

        return back()->with('success', 'Data Anda
Berhasil Diedit!');
    }

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

        $request->validate([
            'password' => 'required',
        ]);

        $dokter = dokter::find($id);
        $dokter->password = bcrypt($request-
>password);
        $dokter->save();

        return back()->with('success', 'Password Anda
Berhasil Diubah!');
    }

```

```
}
```

Penjelasan:

Untuk menampilkan, menambah, mengedit dan menghapus data dokter di database.

App/Http/Controllers/ICD10/icd10Controller.php

```
<?php

namespace App\Http\Controllers\ICD10;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\icd10;
use DB;

class icd10Controller extends Controller
{
    public function show()
    {
        $data = DB::table('icd10s')->get();
        return view('admin.menu.dataICD10')
            ->with(compact('data'));
    }

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

```

        $icd10 = new icd10();
        $icd10->nama = $request->kode;
        $icd10->deskripsi = $request->deskripsi;
        $icd10->save();

        return redirect()->to('/dataICD10')->send()-
>with('success', 'ICD10 Berhasil Ditambahkan!');
    }

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

        $request->validate([
            'kode' => 'required',
            'deskripsi' => 'required',
        ]);

        $icd10 = icd10::find($id);
        $icd10->nama = $request->kode;
        $icd10->deskripsi = $request->deskripsi;
        $icd10->save();

        return redirect()->to('/dataICD10')->send()-
>with('success', 'ICD10 Berhasil Diedit!');
    }

    public function delete($id)
    {
        $deleted = DB::table('icd10s')-
>where('id','=', $id)->delete();

        return redirect()->to('/dataICD10')->send()-
>with('success', 'Data ICD10 berhasil di hapus!');
    }

```

```
}
```

Penjelasan:

Untuk menampilkan, menambah, mengedit dan menghapus data icd10 di database. (CRUD tabel icd10s)

App/Http/Controllers/Obat/obatController.php

```
<?php

namespace App\Http\Controllers\Obat;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\obat;
use DB;

class obatController extends Controller
{
    public function show()
    {
        $data = DB::table('obats')->get();
        return view('admin.menu.dataObat')
            ->with(compact('data'));
    }

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

```

        $obat = new obat();
        $obat->nama = $request->nama;
        $obat->jenis_obat = $request->jenis;
        $obat->deskripsi = $request->deskripsi;
        $obat->save();

        return redirect()->to('/dataObat')->send()-
>with('success', 'Obat Berhasil Ditambahkan!');
    }

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

        $request->validate([
            'nama' => 'required',
            'jenis' => 'required',
        ]);

        $obat = obat::find($id);
        $obat->nama = $request->nama;
        $obat->jenis_obat = $request->jenis;
        $obat->deskripsi = $request->deskripsi;
        $obat->save();

        return redirect()->to('/dataObat')->send()-
>with('success', 'Data Obat Berhasil Diedit!');

    }

    public function delete($id)
    {

```

```

        $deleted = DB::table('obats')-
>where('id','=', $id)->delete();

        return redirect()->to('/dataObat')->send()-
>with('success', 'Data Obat berhasil di hapus!');
    }
}

```

Penjelasan:

Untuk menampilkan, menambah, mengedit dan menghapus data obat di database. (CRUD tabel obats).

App/Http/Controllers/Pasien/pasienController.php

```

<?php

namespace App\Http\Controllers\Pasien;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\pasien;
use App\Models\obat;
use App\Models\poli;
use App\Models\icd10;
use App\Models\keluhan;
use App\Models\obat_keluhan;
use App\Models\dokter;
use DB;
use Illuminate\Support\Facades\Auth;

```

```

class pasienController extends Controller
{

    // Interaksi Admin Ke Pasien

    public function show()
    {
        $data = pasien::get();
        return view('admin.menu.dataPasien')
            ->with(compact('data'));
    }

    public function create(Request $request)
    {
        $request->validate([
            'nama' => 'required',
            'nKepalaKeluarga' => 'required',
            'nIbuKandung' => 'required',
            'kodeDesa' => 'required',
            'nik' => 'required|unique:pasien',
            'agama' => 'required',
            'tglLahir' => 'required',
            'jk' => 'required',
            'status' => 'required',
            'pekerjaan' => 'required',
            'alamat' => 'required',
            'noTelp' => 'required',
            'jenisAsuransi' => 'required',
        ], [
            'nik.unique' => 'NIK Sudah Terdaftar!'
        ]);
    }
}

```

```

        $pasien = new pasien();
        $pasien->nama = $request->nama;
        $pasien->nKepalaKeluarga      =      $request-
>nKepalaKeluarga;
        $pasien->nIbuKandung = $request->nIbuKandung;
        $pasien->kode_desa = $request->kodeDesa;
        $pasien->nik = $request->nik;
        $pasien->agama = $request->agama;
        $pasien->tglLahir = $request->tglLahir;
        $pasien->jk = $request->jk;
        $pasien->status = $request->status;
        $pasien->pekerjaan = $request->pekerjaan;
        $pasien->alamat = $request->alamat;
        $pasien->noTelp = $request->noTelp;
        $pasien->jenis_asuransi      =      $request-
>jenisAsuransi;
        $pasien->no_kartu = $request->noAsuransi;

        $pasien->save();

        return redirect()->to('/kunjunganPasien')-
>send()->with('success', 'Data Pasien Berhasil
Ditambahkan!');
    }

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

        $request->validate([
            'nama' => 'required',
            'nKepalaKeluarga' => 'required',
            'nIbuKandung' => 'required',
            'kodeDesa' => 'required',

```

```

        'nik' => 'required|unique:pasiens,nik, '.$id,
        'agama' => 'required',
        'tglLahir' => 'required',
        'jk' => 'required',
        'status' => 'required',
        'pekerjaan' => 'required',
        'alamat' => 'required',
        'noTelp' => 'required',
        'jenisAsuransi' => 'required',
    ], [
        'nik.unique' => 'NIK Sudah Terdaftar!'
    ]);

    $pasien = pasien::find($id);
    $pasien->nama = $request->nama;
    $pasien->nKepalaKeluarga = $request->nKepalaKeluarga;
    $pasien->nIbuKandung = $request->nIbuKandung;
    $pasien->kode_desa = $request->kodeDesa;
    $pasien->nik = $request->nik;
    $pasien->agama = $request->agama;
    $pasien->tglLahir = $request->tglLahir;
    $pasien->jk = $request->jk;
    $pasien->status = $request->status;
    $pasien->pekerjaan = $request->pekerjaan;
    $pasien->alamat = $request->alamat;
    $pasien->noTelp = $request->noTelp;
    $pasien->jenis_asuransi = $request->jenisAsuransi;
    $pasien->no_kartu = $request->noAsuransi;

```

```

        $pasien->save();

        return redirect()->to('/rmPasien/'.$id)-
>send()->with('success', 'Data Pasien Berhasil Di
Edit');

    }

    public function detail($id)
    {
        $data = pasien::where('id','=', $id)->get();
        $data2 = keluhan::where('pasien_id', '=',
$id)->get();
        $data3 = poli::get();
        $data4 = dokter::where('jabatan', '=',
'dokter')->get();
        return view('admin.menu.inner-menu.rmPasien')
            ->with(compact('data'))
            ->with(compact('data2'))
            ->with(compact('data3'))
            ->with(compact('data4'));
    }

    public function hapus($id)
    {
        $deleted = DB::table('pasien')-
>where('id','=', $id)->delete();
        return redirect()->to('/kunjunganPasien')-
>send()->with('success', 'Data Pasien Berhasil Di
Hapus!');
    }

    public function rm()

```

```

{
    $data = keluhan::get();
    return view('admin.menu.riwayatMedis')
        ->with(compact('data'));
}

public function tambahRM(Request $request, $id)
{
    $request->validate([
        'keluhan' => 'required',
        'poli' => 'required',
        'dokter' => 'required',
    ]);

    $keluhan = new keluhan();
    $keluhan->pasien_id = $id;
    $keluhan->keluhan = $request->keluhan;
    $keluhan->poli_id = $request->poli;
    $keluhan->dokter_id = $request->dokter;
    $keluhan->save();

    return redirect()->to('/rmPasien/'.$id)-
>send()->with('success', 'Riwayat Medis Pasien
Berhasil Ditambahkan!');
}

public function editRM(Request $request, $id,
$pId)
{
    $request->validate([
        'keluhan' => 'required',
        'poli' => 'required',

```

```

        'dokter' => 'required',
    ]));

    $keluhan = keluhan::find($id);
    $keluhan->keluhan = $request->keluhan;
    $keluhan->poli_id = $request->poli;
    $keluhan->dokter_id = $request->dokter;
    $keluhan->save();

    return redirect()->to('/rmPasien/' . $pId)-
>send()->with('success', 'Riwayat Medis Pasien
Berhasil Diedit!');
}

// Interaksi Dokter Ke Pasien

public function detailDokter($id)
{
    $data = pasien::where('id', '=', $id)->get();
    $data2 = keluhan::where('pasien_id', '=',
$id)->get();
    $data3 = obat::get();
    $data4 = icd10::get();
    return view('dokter.menu.inner-
menu.rmPasien')
        ->with(compact('data'))
        ->with(compact('data2'))
        ->with(compact('data3'))
        ->with(compact('data4'));
}

public function rmDokter()
{

```

```

        $data = keluhan::where('dokter_id', '=',
Auth::user()->id)->whereNull('tindakan')->get();

        return view('dokter.menu.riwayatMedis')
            ->with(compact('data'));
    }

    public function editRMDokter(Request $request,
$id, $pId)
    {
        $request->validate([
            'tindakan' => 'required',
        ]);

        $keluhan = keluhan::find($id);
        $keluhan->tindakan = $request->tindakan;
        $keluhan->icd10_id = $request->icd10;
        $keluhan->save();

        return redirect()-
>to('/rmPasienDokter/'.$pId)->send()->with('success',
'Riwayat Medis Pasien Berhasil Diedit!');
    }

    public function editObatDokter(Request $request,
$id, $pId)
    {
        $obat = $request->input('obat_id', []);

        foreach ($obat as $index => $value) {
            $units[] = [
                "keluhan_id" => $id,
                "obat_id" => $obat[$index],
            ];
        }
    }

```

```

    }

    $created = obat_keluhan::insert($units);

    return redirect()-
>to('/rmPasienDokter/'.$pId)->send()->with('success',
'Riwayat Medis Pasien Berhasil Diedit!');
}

public function hapusRM($id)
{
    $deleted = DB::table('keluhans')-
>where('id','=', $id)->delete();

    return redirect()->to('/rmPasien/'.$id)-
>send()->with('success', 'Riwayat Penyakit Pasien
berhasil di hapus!');
}
}

```

Penjelasan:

Untuk menampilkan, menambah, mengedit dan menghapus data pasien dan riwayat medis pasien.

App/Http/Controllers/Poli/poliController.php

```

<?php

namespace App\Http\Controllers\Poli;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\poli;
use DB;

```

```

class poliController extends Controller
{
    public function show()
    {
        $data = DB::table('polis')->get();
        return view('admin.menu.dataPoli')
            ->with(compact('data'));
    }

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

        $poli = new poli();
        $poli->nama = $request->nama;
        $poli->save();

        return redirect()->to('/dataPoli')->send()-
>with('success', 'Poli Berhasil Ditambahkan!');
    }

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

        $request->validate([
            'nama' => 'required',
        ]);

        $poli = poli::find($id);
        $poli->nama = $request->nama;
    }
}

```

```

        $poli->save();

        return redirect()->to('/dataPoli')->send()-
>with('success', 'Poli Berhasil Di Edit!');

    }

    public function delete($id)
    {
        $deleted = DB::table('polis')-
>where('id','=', $id)->delete();

        return redirect()->to('/dataPoli')->send()-
>with('success', 'Data Poli berhasil di hapus!');
    }
}

```

Penjelasan:

Untuk menampilkan, menambah, mengedit dan menghapus data Poli.

App/Models/dokter.php

```

<?php

namespace App\Models;

use
Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class dokter extends Model
{
    use HasFactory;
}

```

```

        protected $table = 'users';

        public function poli()
        {
            return $this->belongsTo('App\Models\poli');
        }

        public function keluhan()
        {
            return $this->hasMany('App\Models\keluhan');
        }
    }

```

Penjelasan:

Untuk menghubungkan relasi pada tabel users ke tabel polis dan keluhan

App/Models/icd10.php

```

<?php

namespace App\Models;

use
Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class icd10 extends Model
{
    use HasFactory;

    public function keluhan()
    {

```

```

        return $this->hasMany('App\Models\keluhan');
    }
}

```

Penjelasan:

Untuk menghubungkan relasi pada tabel icd10s ke tabel keluhan.

App/Models/keluhan.php

```

<?php

namespace App\Models;

use
Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Carbon\Carbon;

class keluhan extends Model
{
    use HasFactory;
    protected $table = 'keluhans';

    public function poli()
    {
        return $this->belongsTo('App\Models\poli');
    }

    public function dokter()
    {
        return $this->belongsTo('App\Models\dokter');
    }
}

```

```

    public function pasien()
    {
        return $this->belongsTo('App\Models\pasien');
    }

    public function icd10()
    {
        return $this->belongsTo('App\Models\icd10');
    }

    public function obat()
    {
        return $this->belongsToMany('App\Models\obat');
    }

    public function getCreatedAtAttribute($value){
        return Carbon::parse($value)->format('d/m/Y');
    }
}

```

Penjelasan:

Untuk menghubungkan relasi pada tabel keluhan ke tabel poli, dokter, pasien, icd10 dan obat.

(getCreatedAtAttribute adalah accessor yang digunakan untuk mengubah format tanggal yang diambil dari database)

App/Models/obat.php

```
<?php

namespace App\Models;

use
Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class obat extends Model
{
    use HasFactory;

    public function keluhan()
    {
        return $this->belongsToMany('App\Models\keluhan');
    }
}
```

Penjelasan:

Untuk menghubungkan tabel obat ke tabel keluhan melalui tabel many to many obat_keluhans.

App/Models/pasien.php

```
<?php

namespace App\Models;

use
Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
```

```

use Carbon\Carbon;

class pasien extends Model
{
    use HasFactory;
    protected $table = 'pasiens';

    public function keluhan()
    {
        return $this->hasMany('App\Models\keluhan');
    }

    public function getTglLahirAttribute($value){
        return Carbon::parse($value)->format('d/m/Y');
    }

    public function setTglLahirAttribute($value){
        $this->attributes['tglLahir'] = Carbon::createFromFormat('d/m/Y', $value)->format('Y-m-d');
    }
}

```

Penjelasan:

Untuk menghubungkan relasi pada tabel pasiens ke tabel keluhan.

(setTglLahirAttribute adalah mutator untuk mengubah format tanggal lahir sebelum dimasukkan ke database)

```
<?php

namespace App\Models;

use
Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class poli extends Model
{
    use HasFactory;
    protected $table = 'polis';

    public function dokter()
    {
        return $this->hasMany('App\Models\dokter');
    }

    public function keluhan()
    {
        return $this->hasMany('App\Models\keluhan');
    }
}
```

Penjelasan:

Untuk menghubungkan relasi pada tabel polis ke tabel keluhan dan tabel dokter.