

LAMPIRAN

Lampiran 1. Source Code Model Laravel

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model; class Obat
extends Model
{ use HasFactory;
    protected $table = 'tabel_obat'; protected
    $primaryKey = 'kode_barang';
    protected $casts = [
        'kode_barang' => 'string',
    ];
    public $timestamps = false; public
    $incrementing = false;
    protected $fillable = ['kode_barang', 'nama_barang', 'stok_awal', 'stok_akhir',
'periode', 'harga', 'diskon']; public
    function transaksi()
    { return $this->hasMany(Transaksi::class, 'kode_barang', 'kode_barang');
    }
}

<?php
namespace App\Models; use
Awobaz\Compoships\Compoships; use
Illuminate\Database\Eloquent\Model; use
Illuminate\Database\Eloquent\Factories\HasFactory;
class Saldo extends Model
{ use HasFactory, Compoships;
    protected $table =
    'saldo_obat'; public
    $timestamps = false; public
    $incrementing = false;
    protected $primaryKey = ['periode', 'kode_barang']; // Define composite key
    protected $casts = [
        'stok_awal' => 'integer',
        'stok_akhir' => 'integer',
        'harga' => 'decimal:2',
    ];
    protected $fillable = ['stok_awal', 'stok_akhir', 'harga']; // Remove periode and
kode_barang from fillable
```

```

protected function setKeysForSaveQuery($query)
{ return $query->where('periode', $this->getAttribute('periode'))
    ->where('kode_barang', $this->getAttribute('kode_barang'));
}
}

```

```

<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model; class Obat
extends Model
{ use HasFactory;
    protected $table = 'tabel_obat'; protected
    $primaryKey = 'kode_barang';
    protected $casts = [
        'kode_barang' => 'string',
    ];
    public $timestamps = false; public
    $incrementing = false;
    protected $fillable = ['kode_barang', 'nama_barang', 'stok_awal', 'stok_akhir',
    'periode', 'harga', 'diskon']; public
    function transaksi()
    { return $this->hasMany(Transaksi::class, 'kode_barang', 'kode_barang');
    }
}

```

```

<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model; class
PrediksiPenjualan extends Model
{ use HasFactory;
    protected $table = 'prediksi_penjualan'; protected
    $fillable = [
        'kode_barang',
        'nama_barang',
        'periode',
        'stok_prediksi',
        'mse',
        'rmse',
        'mae',
        'mape',
    ];
}

```

Lampiran 2. Source Code Migration Laravel

```
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
return new class extends Migration
{
    public function up()
    {
        Schema::create('tabel_obat', function (Blueprint $table) {
            $table->string('kode_barang', 10)->primary();
            $table->string('nama_barang', 50);
            $table->integer('stok_awal');
            $table->integer('stok_akhir');
            $table->string('periode', 10);
            $table->double('harga');
            $table->double('diskon');
        });
    }

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

<?php
use
Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
return new class extends Migration
{
    public function up()
    {
        Schema::create('saldo_obat', function (Blueprint $table) {
            $table->string('periode', 10);
            $table->string('kode_barang', 10);
            $table->integer('stok_awal');
            $table->integer('stok_akhir');
            $table->double('harga');
            $table->primary(['periode',
            'kode_barang']);
        });
    }

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

```

<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
return new class extends Migration
{
    public function up()
    {
        Schema::create('tabel_transaksi', function (Blueprint $table) {
            $table->increments('transaksi_no'); // Primary Key Auto Increment
            $table->enum('kode_transaksi', ['BL', 'JL', 'PM', 'AJ']);
            $table->string('kode_barang', 10);
            $table->integer('stok_awal');
            $table->integer('jumlah'); // Ubah menjadi integer
            $table->integer('stok_akhir');
            $table->double('harga');
            $table->string('periode', 10);
            $table->timestamp('tgl_entry')->useCurrentOnUpdate()->useCurrent();
            $table->dateTime('tgl_transaksi');
        });
    }

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

```



```

<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
return new class extends Migration
{
    public function up()
    {
        Schema::create('prediksi_penjualan', function (Blueprint $table) {
            $table->id(); // Auto Increment ID
            $table->string('kode_barang', 10); // Kode barang
            $table->string('nama_barang', 50); // Nama barang
            $table->string('periode', 10); // Periode prediksi
            $table->double('stok_prediksi'); // Hasil prediksi stok
            $table->double('mse')->nullable(); // Mean Squared Error
            $table->double('rmse')->nullable(); // Root Mean Squared Error
            $table->double('mae')->nullable(); // Mean Absolute Error
            $table->double('mape')->nullable(); // Mean Absolute Percentage Error
            $table->timestamps(); // Untuk timestamp created_at dan updated_at
        });
    }
};

```

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



Lampiran 3. Source Code Controllers Laravel

```
<?php
namespace App\Http\Controllers;    use
App\Models\Obat;                    use
App\Models\Transaksi;               use
Illuminate\Http\Request;            use
App\Models\PrediksiPenjualan;       class
DashboardController extends Controller
{ public function jumlah()
{
    $jumlahObat = Obat::count();
    $jumlahTransaksi = Transaksi::count();

    // Ambil data prediksi terakhir dari session $prediksis
    = PrediksiPenjualan::get();
    return view("staff.obat.dashboard",compact("jumlahObat",
"jumlahTransaksi", "prediksis"));
}
public function hapusPrediksi($kode_barang)
{
    // Cari prediksi berdasarkan kode_barang
    $prediksi = PrediksiPenjualan::where('kode_barang', $kode_barang)->first();
    if (!$prediksi) { return redirect()->back()->with('error', 'Prediksi tidak
ditemukan. ');
}
    // Hapus prediksi $prediksi-
    >delete();
    return redirect()->back()->with('success', 'Data Prediksi berhasil dihapus. ');
}
}

<?php                                namespace
App\Http\Controllers;                use
Illuminate\Http\Request;             use
App\Models\Obat;
class ObatController extends Controller
{ public function index()
{
    $obat = Obat::orderBy('kode_barang', 'asc')->paginate(50); return
    view("staff.tabel.tabel", compact("obat"));
}
public function create()
{
```

```

        return view('staff.tabel.create');
    }
    public function show(Request $request)
    {
        $obat = Obat::where('kode_barang', 'LIKE', '%'.$request->cari.'%')
            ->orWhere('nama_barang', 'LIKE', '%'.$request->cari.'%')
            ->paginate(50);
        return view("staff.tabel.tabel", compact("obat"));
    }
    public function store(Request $request)
    {
        $request->validate([
            "kode_barang" => "required",
            "nama_barang" => "required",
            "stok_akhir" => "required|integer",
            "stok_awal" => "required|integer",
            "periode" => "required",
            "harga" => "required|numeric",
            "diskon" => "nullable|numeric", // Diskon boleh kosong, jika ada hanya
            boleh numeric
        ]);
        Obat::Create([
            "kode_barang"=>$request->kode_barang,
            "nama_barang"=>$request->nama_barang,
            "stok_akhir"=>$request->stok_akhir,
            "stok_awal"=>$request->stok_awal,
            "periode"=>$request->periode,
            "harga"=>$request->harga,
            "diskon"=>$request->diskon,
        ]);
        return redirect()->route('staff.obat')->with('success', 'Data Berhasil
        Ditambahkan');
    }
    public function edit(string $kode_barang)
    {
        $obat = Obat::findOrFail($kode_barang); return
        view('staff.tabel.edit', compact('obat'));
    }
    public function update(Request $request, string $kode_barang)
    {
        $request->validate([
            "kode_barang" => "required",
            "nama_barang" => "required",

```

```

        "stok_akhir" => "required|integer",
        "stok_awal" => "required|integer",
        "periode" => "required",
        "harga" => "required|numeric",
        "diskon" => "nullable|numeric", // Diskon boleh kosong, jika ada hanya
        boleh numeric
    ];

```

```

    $obat = Obat::findOrFail($kode_barang); $obat->update($request-
    >all());
    return redirect()->route('staff.obat')->with('success', 'Data Berhasil Di Edit');
}
public function destroy(string $kode_barang)
{
    Obat::findOrFail($kode_barang);
    return redirect()->route('staff.obat')->with('success', 'Data Berhasil Dihapus');
}
}

```

```

<?php
namespace App\Http\Controllers; use
App\Models\Saldo; use
Illuminate\Http\Request; use
Illuminate\Support\Facades\Log; class
SaldoObatController extends Controller
{ public function index(Request $request)
{
    $query = Saldo::query(); //
    Filter berdasarkan periode if
    ($request->filled('periode')) {
        $query->where('periode', $request->periode);
    }
    // Pencarian berdasarkan kode barang if
    ($request->filled('cari')) {
        $query->where('kode_barang', 'like', '%' . $request->cari . '%'); }
    $saldoObat = $query->paginate(50);
    $periode = Saldo::select('periode')->distinct()->pluck('periode'); return
    view('staff.saldo.index', compact('saldoObat', 'periode'));
}
}
}

```

```

<?php
namespace App\Http\Controllers;
use Carbon\Carbon; use

```

```

App\Models\Obat;          use
App\Models\Transaksi;     use
Illuminate\Http\Request;
class TransaksiController extends Controller
{
    // Controller example
    public function index(Request $request)
    {
        $query = Transaksi::query(); //
        Filter berdasarkan periode if
        ($request->filled('periode')) {
            $query->where('periode', $request->periode);
        }
        // Pencarian berdasarkan kode barang if
        ($request->filled('cari')) {
            $query->where('kode_barang', 'like', '%' . $request->cari . '%'); }
        $transaksi = $query->paginate(200);
        $periode = Transaksi::select('periode')->distinct()->pluck('periode');
        return view('staff.transaksi.index', compact('transaksi', 'periode')); }
    }
}

<?php
namespace App\Http\Controllers; use
DB; use Carbon\Carbon; use
App\Models\Obat;          use
App\Models\Transaksi;     use
App\Models\PrediksiPenjualan; use
Maatwebsite\Excel\Facades\Excel;
use App\Exports\PrediksiExport; use
Illuminate\Http\Request;
class PrediksiController extends Controller
{ public function index()
    { return view('staff.prediksi.index');
    }
    public function export()
    { return Excel::download(new PrediksiExport, 'prediksi.xlsx');
    }
    public function prediksi(Request $request)
    {
        $kode_barang = $request->input('kode_barang');

        // Ambil data transaksi berdasarkan kode_barang
        $transaksi = Transaksi::where('kode_barang', $kode_barang)

```

```

->where('kode_transaksi', 'JL') // Hanya transaksi penjualan -
>orderBy('periode')
->get(); if ($transaksi-
>isEmpty()) {
    return back()->withErrors('Data transaksi tidak ditemukan untuk kode
barang yang diberikan.');
```

```

    }
    // Format data transaksi untuk API Flask
    $dataTransaksi = $transaksi->map(function ($item) { return
    [
        'periode' => $item->periode,
        'jumlah' => $item->jumlah,
    ];
    }->toArray());
    // Panggil API Flask
    $client = new \GuzzleHttp\Client();
    $response = $client->post('http://127.0.0.1:5000/predict', [
        'json' => ['data_transaksi' => $dataTransaksi]
    ]);
    $result = json_decode($response->getBody()->getContents(), true); if
    (isset($result['error'])) {
        return back()->withErrors($result['error']);
    }
    // Format forecast data for view
    $forecasts = isset($result['forecasts']) ? $result['forecasts'] : [];
    // Simpan hasil prediksi ke dalam database
    $nama_barang = Obat::where('kode_barang', $kode_barang)-
    >value('nama_barang');
    PrediksiPenjualan::create([
        'kode_barang' => $kode_barang,
        'nama_barang' => $nama_barang,
        'periode' => $result['next_period'],
        'stok_prediksi' => $result['next_prediction'],
        'mse' => $result['mse'],
        'rmse' => $result['rmse'], 'mae'
        => $result['mae'],
        'mape' => $result['mape']
    ]);
    return view('staff.prediksi.index', compact('result', 'kode_barang',
'nama_barang', 'forecasts')); }
}

```

Lampiran 4. Surat Izin Mengambil Data



RUMAH SAKIT PETROKIMIA GRESIK

Jl. Achmad Yani No. 69, Gresik – Jawa Timur 61119
Telepon: (031) XXXX-XXXX | Website: www.rspetrokimiagresik.com

SURAT IZIN PENGAMBILAN DATA

Nomor: /06/NK.02.01//PGM.01/2024

Yang bertanda tangan di bawah ini:

Nama : Lilik Wijayati, S.Kep.,Ns.,M.Kes.

Jabatan : Manager Departemen SDM

Dengan ini memberikan izin kepada:

Nama : Muhammad Nur Assidiq Syafarrudin

NIM : 210602003

Program Studi : Teknik Informatika

Universitas : Universitas Muhammadiyah Gresik

Untuk melakukan pengambilan data terkait data obat pada Gudang Farmasi IGD Rumah Sakit Petrokimia Gresik guna keperluan penyusunan skripsi dengan judul:

“Implementasi Sistem Prediksi Permintaan Obat Menggunakan Support Vector Regression (SVR) pada Gudang Farmasi IGD Rumah Sakit Petrokimia Gresik”.

Demikian surat ini dibuat untuk digunakan sebagaimana mestinya.

Gresik, 11 Juni 2024

Mengetahui,
PT Petro Graha Medika

(Lilik Wijayati, S.Kep.,Ns.,M.Kes.)
Manager Departemen SDM