Login Register Petugas

```php
public function login(Request $request)
    {
        $credentials = $request->only('username', 'password');

        try {
            if (! $token = JWTAuth::attempt($credentials)) {
                return response()->json(['error' => 'invalid_credentials'], 400);
            }
        } catch (JWTException $e) {
            return response()->json(['error' => 'could_not_create_token'], 500);
        }

        return response()->json(compact('token'));
    }

    public function register(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'nama_petugas' => 'required|string|max:255',
            'no_telp' => 'required',
            'level' => 'required',
            'username' => 'required|string|max:255',
            'password' => 'required|string|min:6|confirmed',
        ]);

        if($validator->fails()){
            return response()->json($validator->errors()->toJson(), 400);
        }

        $user = Petugas::create([
            'nama_petugas' => $request->get('nama_petugas'),
            'no_telp' => $request->get('no_telp'),
            'level' => $request->get('level'),
            'username' => $request->get('username'),
            'password' => Hash::make($request->get('password')),
        ]);

        $token = JWTAuth::fromUser($user);

        return response()->json(compact('user','token'),201);
    }
```

CRUD Jenis Cuci

```
public function index($id)
    {
        if(Auth::user()->level=="admin"){
            $jeniscuci=DB::table('jenis_cuci')
            ->where('jenis_cuci.id',$id)
            ->get();
            return response()->json($jeniscuci);
        }else{
            return response()->json(['status'=>'anda bukan admin']);
        }
    }
    public function store(Request $req)
    {
        if(Auth::user()->level=="admin"){
        $validator=Validator::make($req->all(),
        [
            'nama_jenis'=>'required',
            'harga_per_kilo'=>'required'
        ]
        );
        if($validator->fails()){
            return Response()->json($validator->errors());
        }

        $simpan=Jeniscuci::create([
            'nama_jenis'=>$req->nama_jenis,
            'harga_per_kilo'=>$req->harga_per_kilo
        ]);
        $status=1;
        $message="Jenis Cuci Berhasil Ditambahkan";
        if($simpan){
          return Response()->json(compact('status','message'));
        }else {
          return Response()->json(['status'=>0]);
        }
      }
      else {
          return response()->json(['status'=>'anda bukan admin']);
      }
  }
    public function update($id,Request $request){
        if(Auth::user()->level=="admin"){
        $validator=Validator::make($request->all(),
```

```
            [
                'nama_jenis'=>'required',
                'harga_per_kilo'=>'required'
            ]
        );

        if($validator->fails()){
        return Response()->json($validator->errors());
        }

        $ubah=Jeniscuci::where('id',$id)->update([
            'nama_jenis'=>$request->nama_jenis,
            'harga_per_kilo'=>$request->harga_per_kilo
        ]);
        $status=1;
        $message="Jenis Cuci Berhasil Diubah";
        if($ubah){
        return Response()->json(compact('status','message'));
        }else {
        return Response()->json(['status'=>0]);
        }
        }
    else {
    return response()->json(['status'=>'anda bukan admin']);
    }
}
    public function destroy($id){
        if(Auth::user()->level=="admin"){
        $hapus=Jeniscuci::where('id',$id)->delete();
        $status=1;
        $message="Jenis Cuci Berhasil Dihapus";
        if($hapus){
        return Response()->json(compact('status','message'));
        }else {
        return Response()->json(['status'=>0]);
        }
    }
    else {
        return response()->json(['status'=>'anda bukan admin']);
        }
    }

    public function tampil(){
        if(Auth::user()->level=="admin"){
            $datas = Jeniscuci::get();
```

```
            $count = $datas->count();
            $jeniscuci = array();
            $status = 1;
            foreach ($datas as $dt_jc){
                $jeniscuci[] = array(
                    'id' => $dt_jc->id,
                    'nama_jenis' => $dt_jc->nama_jenis,
                    'harga_per_kilo' => $dt_jc->harga_per_kilo
                );
            }
            return Response()->json(compact('count','jeniscuci'));
        } else {
            return Response()->json(['status'=> 'Tidak bisa, anda bukan admin']);
        }
    }
```

CRUD Pelanggan

```
public function index($id)
    {
        if(Auth::user()->level=="admin"){
            $pelanggan=DB::table('pelanggan')
            ->where('pelanggan.id',$id)
            ->get();
            return response()->json($pelanggan);
        }else{
            return response()->json(['status'=>'anda bukan admin']);
        }
    }
    public function store(Request $req)
    {
        if(Auth::user()->level=="admin"){
        $validator=Validator::make($req->all(),
        [
            'nama'=>'required',
            'alamat'=>'required',
            'telp'=>'required'
        ]
        );
        if($validator->fails()){
            return Response()->json($validator->errors());
        }

        $simpan=Pelanggan::create([
            'nama'=>$req->nama,
```

```
            'alamat'=>$req->alamat,
            'telp'=>$req->telp
        ]);
        $status=1;
        $message="Pelanggan Berhasil Ditambahkan";
        if($simpan){
          return Response()->json(compact('status','message'));
        }else {
          return Response()->json(['status'=>0]);
        }
      }
      else {
          return response()->json(['status'=>'anda bukan admin']);
      }
}
  public function update($id,Request $request){
      if(Auth::user()->level=="admin"){
      $validator=Validator::make($request->all(),
          [
              'nama'=>'required',
              'alamat'=>'required',
              'telp'=>'required'
          ]
      );

      if($validator->fails()){
      return Response()->json($validator->errors());
      }

      $ubah=Pelanggan::where('id',$id)->update([
          'nama'=>$request->nama,
          'alamat'=>$request->alamat,
          'telp'=>$request->telp
      ]);
      $status=1;
      $message="Pelanggan Berhasil Diubah";
      if($ubah){
      return Response()->json(compact('status','message'));
      }else {
      return Response()->json(['status'=>0]);
      }
      }
  else {
  return response()->json(['status'=>'anda bukan admin']);
  }
```

```
}
    public function destroy($id){
        if(Auth::user()->level=="admin"){
        $hapus=Pelanggan::where('id',$id)->delete();
        $status=1;
        $message="Pelanggan Berhasil Dihapus";
        if($hapus){
        return Response()->json(compact('status','message'));
        }else {
        return Response()->json(['status'=>0]);
        }
    }
    else {
        return response()->json(['status'=>'anda bukan admin']);
        }
    }

    public function tampil(){
        if(Auth::user()->level=="admin"){
            $datas = Pelanggan::get();
            $count = $datas->count();
            $pelanggan = array();
            $status = 1;
            foreach ($datas as $dt_pl){
                $pelanggan[] = array(
                    'id' => $dt_pl->id,
                    'nama' => $dt_pl->nama,
                    'alamat' => $dt_pl->alamat,
                    'telp' => $dt_pl->telp
                );
            }
            return Response()->json(compact('count','pelanggan'));
        } else {
            return Response()->json(['status'=> 'Tidak bisa, anda bukan admin']);
        }
    }
```

Transaksi

```
public function store(Request $req){
        if(Auth::user()->level == 'petugas'){

        $validator = Validator::make($req->all(),
        [
            'id_pelanggan'=>'required',
```

```
            'id_petugas'=>'required',
            'tgl_transaksi'=>'required',
            'tgl_selesai'=>'required'
        ]
        );
        if($validator->fails()){
            return Response()->json($validator->errors());
        }

        $simpan = Transaksi::create([
            'id_pelanggan'=>$req->id_pelanggan,
            'id_petugas'=>$req->id_petugas,
            'tgl_transaksi'=>$req->tgl_transaksi,
            'tgl_selesai'=>$req->tgl_selesai

        ]);
        if($simpan){
            return Response()->json('Data Transaksi berhasil ditambahkan');
        }else{
            return Response()->json('Data Transaksi gagal ditambahkan');
        }
    }else{
        return Response()->json('Anda Bukan Petugas');
    }
    }

    public function update($id,Request $req){
        if(Auth::user()->level == 'petugas'){

        $validator = Validator::make($req->all(),
        [
            'id_pelanggan'=>'required',
            'id_petugas'=>'required',
            'tgl_transaksi'=>'required',
            'tgl_selesai'=>'required'
        ]
        );
        if($validator->fails()){
            return Response()->json($validator->errors());
        }

        $ubah = Transaksi::where('id', $id)->update([
            'id_pelanggan'=>$req->id_pelanggan,
            'id_petugas'=>$req->id_petugas,
            'tgl_transaksi'=>$req->tgl_transaksi,
```

```
            'tgl_selesai'=>$req->tgl_selesai
        ]);
        if($ubah){
            return Response()->json('Data Transaksi berhasil diubah');
        }else{
            return Response()->json('Data Transaksi gagal diubah');
        }
    }else{
        return Response()->json('Anda Bukan Petugas');
    }
    }

    public function destroy($id){
        if(Auth::user()->level == 'admin'){

        $hapus = Transaksi::where('id', $id)->delete();
        if($hapus){
            return Response()->json('Data Transaksi berhasil dihapus');
        }else{
            return Response()->json('Data Transaksi gagal dihapus');
        }
    }else{
        return Response()->json('Anda Bukan Admin');
    }
    }

    public function show(Request $req){
        if(Auth::user()->level == "petugas"){
            $transaksi = DB::table('transaksi')-
>join('pelanggan','pelanggan.id','=','transaksi.id_pelanggan')
            ->where('transaksi.tgl_transaksi','>=',$req->tgl_transaksi)
            ->where('transaksi.tgl_transaksi','<=',$req->tgl_selesai)
            -
>select('nama','telp','alamat','transaksi.id','tgl_transaksi','tgl_selesai')
            ->get();

            if($transaksi->count() > 0){
            $data_transaksi = array();
            foreach ($transaksi as $t){
                $grand = DB::table('detail_transaksi')-
>where('id_transaksi','=',$t->id)
                ->groupBy('id_transaksi')
                ->select(DB::raw('sum(subtotal) as grandtotal'))
                ->first();
```

```
                $detail = DB::table('detail_transaksi')-
>join('jenis_cuci','detail_transaksi.id_jenis','=','jenis_cuci.id')
                ->where('id_transaksi','=',$t->id)
                ->get();

                $data_transaksi[] = array(
                    'Tanggal' => $t->tgl_transaksi,
                    'Nama Pelanggan' => $t->nama,
                    'Alamat' => $t->alamat,
                    'No Telp' => $t->telp,
                    'Tanggal Ambil' => $t->tgl_selesai,
                    'Grand Total' => $grand,
                    'Detail' => $detail,
                );
            }
            return response()->json(compact('data_transaksi'));
    }else{
            $status = 'tidak ada transaksi antara tanggal '.$req-
>tgl_transaksi.' sampai dengan tanggal '.$req->tgl_selesai;
            return response()->json(compact('status'));
    }
        }else{
            return Response()->json('Anda Bukan Petugas');
        }
}
```

Detai Transaksi

```
public function store(Request $req){
        if (Auth::user()->level=='petugas') {
            $validator=Validator::make($req->all(),[
                'id_transaksi' => 'required',
                'id_jenis' => 'required',
                'qty' => 'required',
              ]);
              if($validator->fails()){
                return Response()->json($validator->errors());
              }

              $harga = JenisCuci::where('id', $req->id_jenis)->first();
              $subtotal = $harga->harga_per_kilo * $req->qty;

              $simpan=Detailtransaksi::create([
                  'id_transaksi' => $req->id_transaksi,
                  'id_jenis' => $req->id_jenis,
```

```
                'qty' => $req->qty,
                'subtotal' => $subtotal,
            ]);
            if($simpan){
                $data['status']="Berhasil";
                $data['message']="Data berhasil disimpan!";
                return Response()->json($data);
            }else{
                $data['status']="Gagal";
                $data['message']="Data gagal disimpan!";
                return Response()->json($data);
            }
        } else {
            $data['status']="Gagal";
            $data['Message']="Anda bukan Petugas!";
            return Response()->json($data);
        }
    }

    public function update($id,Request $req){
        if(Auth::user()->level == 'petugas'){

        $validator = Validator::make($req->all(),
        [
            'id_transaksi'=>'required',
            'id_jenis'=>'required',
            'qty'=>'required'
        ]
        );
        if($validator->fails()){
            return Response()->json($validator->errors());
        }

        $harga = Jeniscuci::where('id',$req->id_jenis)->first();
        $subtotal = $harga->harga_perkilo * $req->qty;

        $ubah = Detailtransaksi::where('id', $id)->update([
            'id_transaksi'=> $req->id_transaksi,
            'id_jenis'=> $req->id_jenis,
            'subtotal'=> $subtotal,
            'qty'=> $req->qty
        ]);
        if($ubah){
            return Response()->json('Data Detail Transaksi berhasil diubah');
        }else{
```

```
            return Response()->json('Data Detail Transaksi gagal diubah');
        }
    }else{
        return Response()->json('Anda Bukan Petugas');
    }
    }

    public function destroy($id){
        if(Auth::user()->level == 'admin'){

        $hapus = Detailtransaksi::where('id', $id)->delete();
        if($hapus){
            return Response()->json('Data Detail Transaksi berhasil dihapus');
        }else{
            return Response()->json('Data Detail Transaksi gagal dihapus');
        }
    }else{
        return Response()->json('Anda Bukan admin');
    }
    }
```

Menampilkan Transaksi berdasarkan tanggal awal dan akhir