

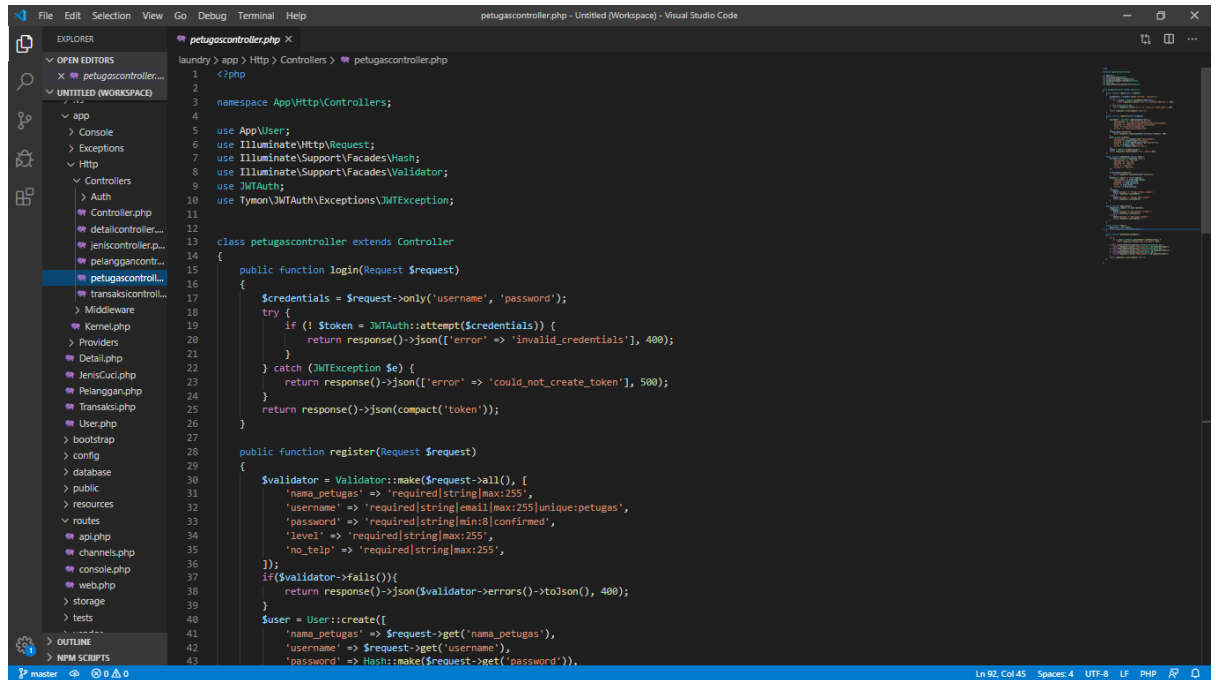
Nama : Saddam Raihan Ramadhan

No.Absen : 31

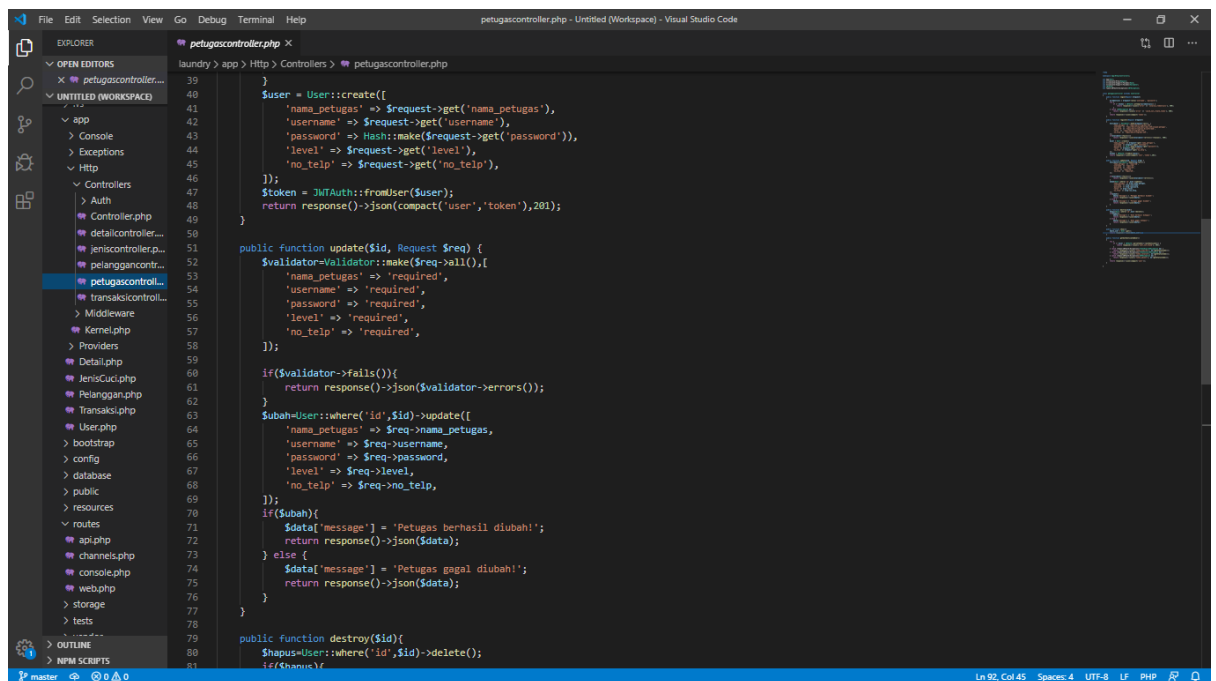
PROJECT : LAUNDRY

1. CRUD Master

Login / Petugas untuk menambahkan, mengurangi admin atau petugas



```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\User;
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\Hash;
8 use Illuminate\Support\Facades\Validator;
9 use JWTAuth;
10 use Tymon\JWTAuth\Exceptions\JWTException;
11
12 class petugascontroller extends Controller
13 {
14     public function login(Request $request)
15     {
16         $credentials = $request->only('username', 'password');
17         try {
18             if (! $token = JWTAuth::attempt($credentials)) {
19                 return response()->json(['error' => 'invalid_credentials'], 400);
20             }
21         } catch (JWTException $e) {
22             return response()->json(['error' => 'could_not_create_token'], 500);
23         }
24         return response()->json(compact('token'));
25     }
26
27     public function register(Request $request)
28     {
29         $validator = Validator::make($request->all(), [
30             'nama_petugas' => 'required|string|max:255',
31             'username' => 'required|string|email|max:255|unique:petugas',
32             'password' => 'required|string|min:8|confirmed',
33             'level' => 'required|string|max:255',
34             'no_telp' => 'required|string|max:255',
35         ]);
36         if ($validator->fails()) {
37             return response()->json($validator->errors()->toArray(), 400);
38         }
39         $user = User::create([
40             'nama_petugas' => $request->get('nama_petugas'),
41             'username' => $request->get('username'),
42             'password' => Hash::make($request->get('password')),
43         ]);
```



```
44         $user = User::create([
45             'nama_petugas' => $request->get('nama_petugas'),
46             'username' => $request->get('username'),
47             'password' => Hash::make($request->get('password')),
48             'level' => $request->get('level'),
49             'no_telp' => $request->get('no_telp'),
50         ]);
51         $token = JWTAuth::fromUser($user);
52         return response()->json(compact('user', 'token'), 201);
53     }
54
55     public function update($id, Request $req) {
56         $validator=Validator::make($req->all(),[
57             'nama_petugas' => 'required',
58             'username' => 'required',
59             'password' => 'required',
60             'level' => 'required',
61             'no_telp' => 'required',
62         ]);
63         if($validator->fails()){
64             return response()->json($validator->errors());
65         }
66         $ubah=User::where('id',$id)->update([
67             'nama_petugas' => $req->nama_petugas,
68             'username' => $req->username,
69             'password' => $req->password,
70             'level' => $req->level,
71             'no_telp' => $req->no_telp,
72         ]);
73         if($ubah){
74             $data['message'] = 'Petugas berhasil diubah!';
75             return response()->json($data);
76         } else {
77             $data['message'] = 'Petugas gagal diubah!';
78             return response()->json($data);
79         }
80     }
81
82     public function destroy($id){
83         $hapus=User::where('id',$id)->delete();
84         if($hapus){
```

```
1 laundry > app > Http > Controllers > petugascontroller.php
2 <?php
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Pelanggan;
7 use Illuminate\Support\Facades\Validator;
8 use Illuminate\Support\Facades\Validador;
9 use JWTAuth;
10 use Auth;
11 use Tymon\JWTAuth\Exceptions\JWTException;
12
13 class petugascontroller extends Controller
14 {
15     public function store(Request $req){
16         if(Auth::user()->level=="petugas"){
17             $validator=Validator::make($req->all(),[
18                 'nama' => 'required',
19                 'alamat' => 'required',
20                 'no_telp' => 'required',
21             ]);
22             if($validator->fails()){
23                 return Response()->json($validator->errors());
24             }
25             $simpan=Pelanggan::create([
26                 'nama' => $req->nama,
27                 'alamat' => $req->alamat,
28                 'no_telp' => $req->no_telp,
29             ]);
30             if($simpan){
31                 $data["status"]="Berhasil";
32                 $data["message"]="Data berhasil disimpan!";
33                 return Response()->json($data);
34             }else{
35                 $data["status"]="Gagal";
36                 $data["message"]="Data gagal disimpan!";
37                 return Response()->json($data);
38             }
39         } else {
40             $data["status"]="Gagal";
41             $data["message"]="Anda bukan Petugas!";
42             return Response()->json($data);
43         }
44     }
45 }
```

CRUD Master Pelanggan yang dapat menginput hanya petugas untuk memasukkan data pelanggan dan yang menghapus, melihat dan mengedit adalah admin

```
1 laundry > app > Http > Controllers > pelanggancontroller.php
2 <?php
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Pelanggan;
7 use Illuminate\Support\Facades\Validator;
8 use Illuminate\Support\Facades\Validador;
9 use JWTAuth;
10 use Auth;
11 use Tymon\JWTAuth\Exceptions\JWTException;
12
13 class pelanggancontroller extends Controller
14 {
15     public function store(Request $req){
16         if(Auth::user()->level=="petugas"){
17             $validator=Validator::make($req->all(),[
18                 'nama' => 'required',
19                 'alamat' => 'required',
20                 'no_telp' => 'required',
21             ]);
22             if($validator->fails()){
23                 return Response()->json($validator->errors());
24             }
25             $simpan=Pelanggan::create([
26                 'nama' => $req->nama,
27                 'alamat' => $req->alamat,
28                 'no_telp' => $req->no_telp,
29             ]);
30             if($simpan){
31                 $data["status"]="Berhasil";
32                 $data["message"]="Data berhasil disimpan!";
33                 return Response()->json($data);
34             }else{
35                 $data["status"]="Gagal";
36                 $data["message"]="Data gagal disimpan!";
37                 return Response()->json($data);
38             }
39         } else {
40             $data["status"]="Gagal";
41             $data["message"]="Anda bukan Petugas!";
42             return Response()->json($data);
43         }
44     }
45 }
```

```
38         return Response()->json($data);
39     }
40 }
41
42 $data['status']="Gagal";
43 $data['Message']="Anda bukan Petugas!";
44 return Response()->json($data);
45 }
46
47 public function update($id, Request $req)
48 {
49     if (Auth::user()->level=="admin") {
50         $validator=Validator::make($req->all(),
51         [
52             'nama' => 'required',
53             'alamat' => 'required',
54             'no_telp' => 'required',
55         ]);
56
57         if($validator->fails()){
58             return Response()->json($validator->errors());
59         }
60         $ubah=Pelanggan::where('id',$id)->update([
61             'nama' => $req->nama,
62             'alamat' => $req->alamat,
63             'no_telp' => $req->no_telp,
64         ]);
65         if($ubah){
66             $data['status']="Berhasil";
67             $data['message']="Data berhasil diubah!";
68             return Response()->json($data);
69         }else{
70             $data['status']="Gagal";
71             $data['message']="Data gagal diubah!";
72             return Response()->json($data);
73         }
74     } else {
75         $data['status']="Gagal";
76         $data['Message']="Anda bukan Admin!";
77         return Response()->json($data);
78     }
79 }
80 }
```

```
81 public function destroy($id){
82     if (Auth::user()->level=="admin") {
83         $hapus=Pelanggan::where('id',$id)->delete();
84         if($hapus){
85             $data['status']="Berhasil";
86             $data['message']="Data berhasil dihapus!";
87             return Response()->json($data);
88         }else{
89             $data['status']="Gagal";
90             $data['message']="Data gagal dihapus!";
91             return Response()->json($data);
92         }
93     } else {
94         $data['status']="Gagal";
95         $data['Message']="Anda bukan Admin!";
96         return Response()->json($data);
97     }
98 }
99
100 public function show(){
101     if(Auth::user()->level=="admin"){
102         $data_pelanggan=Pelanggan::get();
103         $arr_data=array();
104         foreach($data_pelanggan as $dp) {
105             $arr_data = array(
106                 'nama' => $dp->nama,
107                 'alamat' => $dp->alamat,
108                 'no_telp'=> $dp->no_telp,
109             );
110         }
111         $data['Pelanggan']=$arr_data;
112         return Response()->json($data_pelanggan);
113     } else {
114         $data['status']="Gagal";
115         $data["Anda bukan Admin!"];
116         return Response()->json($data);
117     }
118 }
119 }
```

CRUD Master (jenis) dalam cucian menginputkan jenis apa saja dalam cucian sehingga pelanggan dapat mengetahui harga dari pelayanan tersebut

```
File Edit Selection View Go Debug Terminal Help
jeniscontroller.php - Untitled (Workspace) - Visual Studio Code

EXPLORER
jeniscontroller.php X
OPEN EDITORS
jeniscontroller.php...
UNTITLE (WORKSPACE)
app
Console
Exceptions
Http
Controllers
Auth
Controller.php
detailcontroller...
jeniscontroller.p...
pelanggancontr...
petugascontr...
transaksicontr...
Middleware
Kernel.php
Providers
Detail.php
JenisCuci.php
Pelanggan.php
Transaksi.php
bootstrap
config
database
public
resources
routes
api.php
channels.php
console.php
web.php
storage
tests
OUTLINE
NPM SCRIPTS

jeniscontroller.php X
laundry > app > Http > Controllers > jeniscontroller.php
1 {
2 }
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\JenisCuci;
7 use Illuminate\Support\Facades\Hash;
8 use Illuminate\Support\Facades\Validator;
9 use JWTAuth;
10 use Auth;
11 use Tymon\JWTAuth\Exceptions\JWTException;
12
13 class jeniscontroller extends Controller
14 {
15     public function store(Request $req){
16         if (Auth::user()->level=='petugas') {
17             $validator=Validator::make($req->all(),[
18                 'nama_jenis' => 'required',
19                 'harga_per_kilo' => 'required',
20             ]);
21             if($validator->fails()){
22                 return response()->json($validator->errors());
23             }
24
25             $simpan=JenisCuci::create([
26                 'nama_jenis' => $req->nama_jenis,
27                 'harga_per_kilo' => $req->harga_per_kilo,
28             ]);
29             if($simpan){
30                 $data['status']="Berhasil";
31                 $data['message']="Data berhasil disimpan!";
32                 return Response()->json($data);
33             }else{
34                 $data['status']="Gagal";
35                 $data['message']="Data gagal disimpan!";
36                 return Response()->json($data);
37             }
38         } else {
39             $data['status']="Gagal";
40             $data['message']="Anda bukan Petugas!";
41             return Response()->json($data);
42         }
43     }
44 }
```

```
File Edit Selection View Go Debug Terminal Help
jeniscontroller.php - Untitled (Workspace) - Visual Studio Code

EXPLORER
jeniscontroller.php X
OPEN EDITORS
jeniscontroller.php...
UNTITLE (WORKSPACE)
app
Console
Exceptions
Http
Controllers
Auth
Controller.php
detailcontroller...
jeniscontroller.p...
pelanggancontr...
petugascontr...
transaksicontr...
Middleware
Kernel.php
Providers
Detail.php
JenisCuci.php
Pelanggan.php
Transaksi.php
bootstrap
config
database
public
resources
routes
api.php
channels.php
console.php
web.php
storage
tests
OUTLINE
NPM SCRIPTS

jeniscontroller.php X
laundry > app > Http > Controllers > jeniscontroller.php
43 }
44
45 public function update($id, Request $req){
46     if (Auth::user()->level=='admin') {
47         $validator=Validator::make($req->all(),[
48             'nama_jenis' => 'required',
49             'harga_per_kilo' => 'required'
50         ]);
51         if($validator->fails()){
52             return response()->json($validator->errors());
53         }
54         $ubah=JenisCuci::where('id', $id)->update([
55             'nama_jenis' => $req->nama_jenis,
56             'harga_per_kilo' => $req->harga_per_kilo,
57         ]);
58         if($ubah){
59             $data['status']="Berhasil";
60             $data['message']="Data berhasil diubah!";
61             return Response()->json($data);
62         }else{
63             $data['status']="Gagal";
64             $data['message']="Data gagal diubah!";
65             return Response()->json($data);
66         }
67     } else {
68         $data['status']="Gagal";
69         $data['message']="Anda bukan Admin!";
70         return Response()->json($data);
71     }
72 }
73
74 public function destroy($id){
75     if (Auth::user()->level=='admin') {
76         $hapus=JenisCuci::where('id', $id)->delete();
77         if($hapus){
78             $data['status']="Berhasil";
79             $data['message']="Data berhasil dihapus!";
80             return Response()->json($data);
81         }else{
82             $data['status']="Gagal";
83             $data['message']="Data gagal dihapus!";
84             return Response()->json($data);
85         }
86     }
87 }
```

```
64 $data['status']="Gagal";
65 $data['message']="Data gagal diubah!";
66 return Response()->json($data);
67 }
68 } else {
69 $data['status']="Gagal";
70 $data['message']="Anda bukan Admin!";
71 return Response()->json($data);
72 }
73 }
74 }
75
76 public function destroy($id){
77 if (Auth::user()->level=="admin") {
78 $hapus=JenisCuci::where('id', $id)->delete();
79 if($hapus){
80 $data['status']="Berhasil";
81 $data['message']="Data berhasil dihapus!";
82 return Response()->json($data);
83 }else{
84 $data['status']="Gagal";
85 $data['message']="Data gagal dihapus!";
86 return Response()->json($data);
87 }
88 } else {
89 $data['status']="Gagal";
90 $data['message']="Anda bukan Admin!";
91 return Response()->json($data);
92 }
93 }
94
95 public function show(){
96 if (Auth::user()->level=="admin") {
97 $data_cuci=JenisCuci::get();
98 $data['Jenis Cucian']=$data_cuci;
99 return response()->json($data);
100 } else {
101 $data['status']="Gagal";
102 $data['message']="Anda bukan Admin!";
103 return Response()->json($data);
104 }
105 }
106 }
```

CRUD Master table transaksi menampilkan data transaksi pelanggan

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Transaksi;
7 use Illuminate\Support\Facades\Validator;
8 use Illuminate\Support\Facades\Auth;
9 use Tymon\JWTAuth;
10 use Auth;
11 use DB;
12 use Tymon\JWTAuth\Exceptions\JWTException;
13
14 class transaksikontrol extends Controller
15 {
16
17     public function store(Request $req){
18         if (Auth::user()->level=="petugas") {
19             $validator=Validator::make($req->all(),[
20                 'id_pelanggan' => 'required',
21                 'id_petugas' => 'required',
22                 'tgl_transaksi' => 'required',
23                 'tgl_selesai' => 'required',
24             ]);
25             if($validator->fails()){
26                 return Response()->json($validator->errors());
27             }
28
29             $simpan=Transaksi::create([
30                 'id_pelanggan' => $req->id_pelanggan,
31                 'id_petugas' => $req->id_petugas,
32                 'tgl_transaksi' => $req->tgl_transaksi,
33                 'tgl_selesai' => $req->tgl_selesai
34             ]);
35             if($simpan){
36                 $data['status']="Berhasil";
37                 $data['message']="Data berhasil disimpan!";
38                 return Response()->json($data);
39             }else{
40                 $data['status']="Gagal";
41                 $data['message']="Data gagal disimpan!";
42                 return Response()->json($data);
43             }
44         } else {
45 }
```

```
File Edit Selection View Go Debug Terminal Help
transaksicontroller.php - Untitled (Workspace) - Visual Studio Code

EXPLORER
transaksicontroller.php X
laundry > app > Http > Controllers > transaksicontroller.php
X transaksicontroller...
UNLOADED (WORKSPACE)
app
  Console
  Exceptions
  Http
  Controllers
    Auth
    Controller.php
    detailcontroller...
    jeniscucic...
    pelanggancon...
    petugascontroll...
    transaksicontroll...
  Middleware
  Kernel.php
  Providers
  Detail.php
  Jeniscuci.php
  Pelanggan.php
  Transaksi.php
  User.php
  bootstrap
  config
  database
  public
  resources
  routes
    api.php
    channels.php
    console.php
    web.php
  storage
  tests
  ...
  OUTLINE
  NPM SCRIPTS

transaksicontroller.php
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82

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', 'no_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->no_telp,
                    'Deadline' => $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;
        }
    }
}
```

```
File Edit Selection View Go Debug Terminal Help
transaksicontroller.php - Untitled (Workspace) - Visual Studio Code

EXPLORER
transaksicontroller.php X
laundry > app > Http > Controllers > transaksicontroller.php
X transaksicontroller...
UNLOADED (WORKSPACE)
app
  Console
  Exceptions
  Http
  Controllers
    Auth
    Controller.php
    detailcontroller...
    jeniscucic...
    pelanggancon...
    petugascontroll...
    transaksicontroll...
  Middleware
  Kernel.php
  Providers
  Detail.php
  Jeniscuci.php
  Pelanggan.php
  Transaksi.php
  User.php
  bootstrap
  config
  database
  public
  resources
  routes
    api.php
    channels.php
    console.php
    web.php
  storage
  tests
  ...
  OUTLINE
  NPM SCRIPTS

transaksicontroller.php
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90

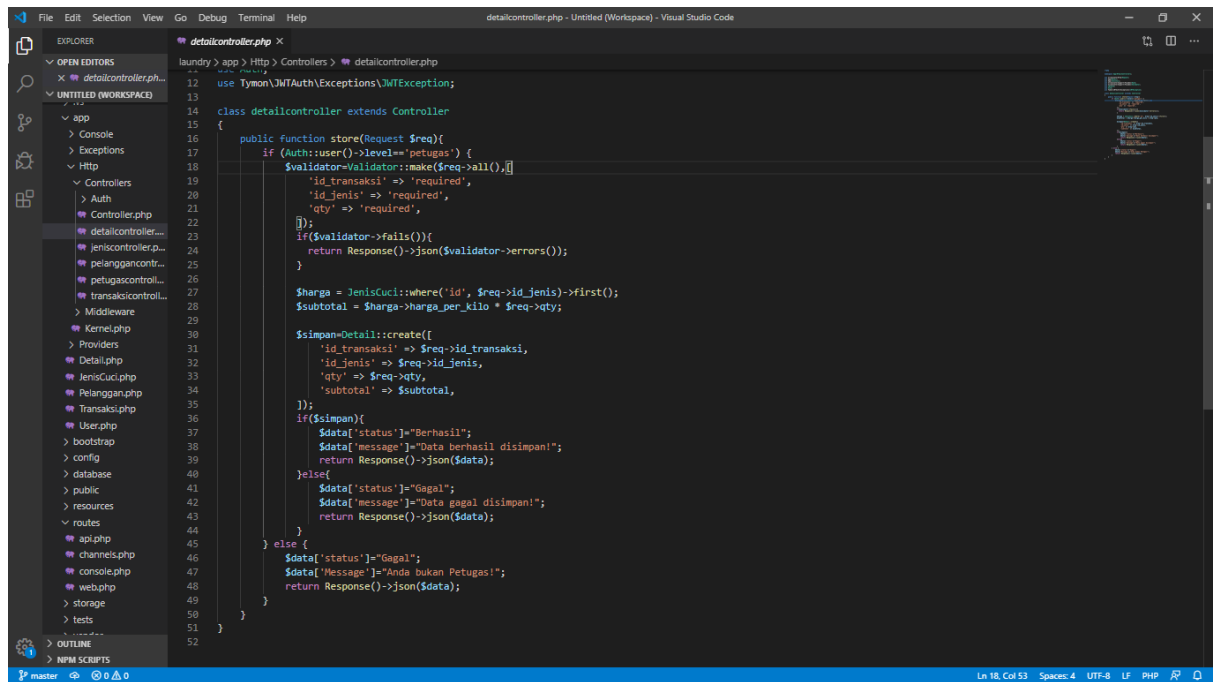
public function show(Request $req){
    if(Auth::user()->level == "petugas"){
        $transaksi = DB::table('transaksi')->join('pelanggan','pelanggan.id','=', 'transaksi.id_pelanggan')
        ->where('transaksi.tgl_transaksi', '>=', $req->tgl_transaksi)
        ->where('transaksi.tgl_transaksi', '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

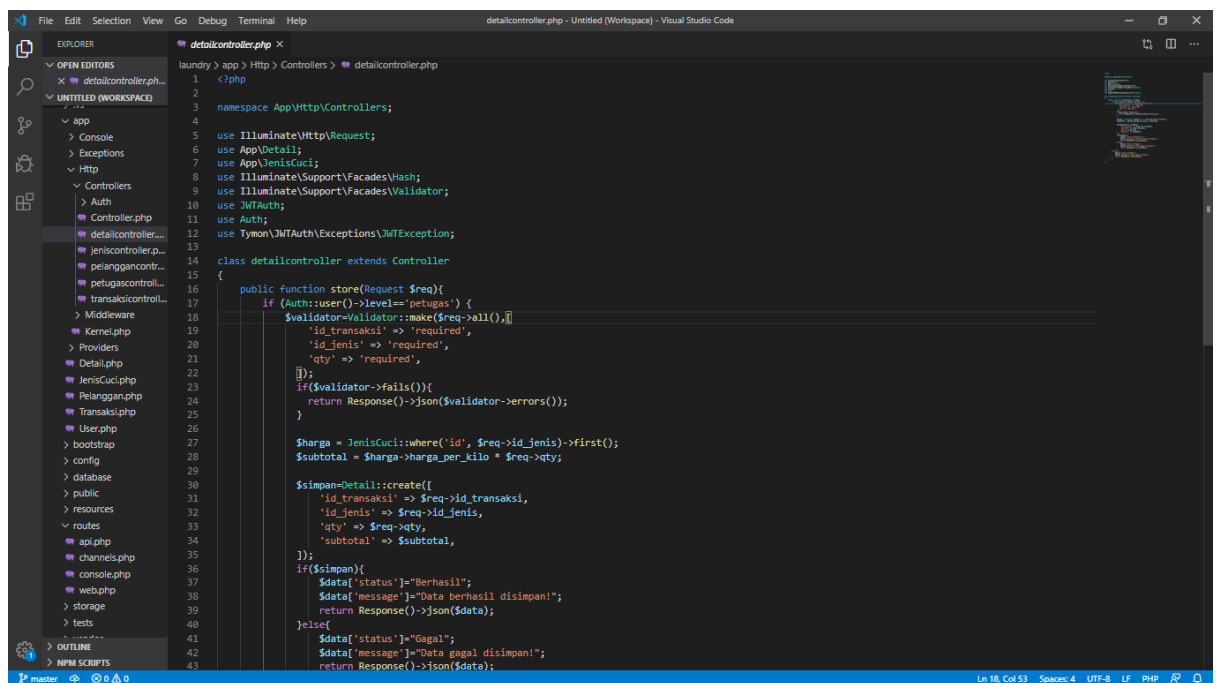
                $detail = DB::table('detail_transaksi')->join('jenis_cuci','detail_transaksi.id_jenis','=', 'jenis_cuci.id')
                ->where('id_transaksi','=', $t->id)
                ->get();

                $data_transaksi[] = array(
                    'Tanggal' => $t->tgl_transaksi,
                    'Nama Pelanggan' => $t->nama,
                    'Alamat' => $t->alamat,
                    'No Telp' => $t->no_telp,
                    'Deadline' => $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');
    }
}
```

CRUD Master dari table detail ini digunakan untuk menampilkan detail yang dipilih pelanggan



```
12 use Tymon\JWTAuth\Exceptions\JWTException;
13
14 class detailcontroller extends Controller
15 {
16     public function store(Request $req){
17         if (Auth::user()->level=='petugas') {
18             $validator=Validator::make($req->all(),[
19                 'id_transaksi' => 'required',
20                 'id_jenis' => 'required',
21                 'qty' => 'required',
22             ]);
23             if($validator->fails()){
24                 return Response()->json($validator->errors());
25             }
26             $harga = JenisCuci::where('id', $req->id_jenis)->first();
27             $subtotal = $harga->harga_per_kilo * $req->qty;
28
29             $simpan=Detail::create([
30                 'id_transaksi' => $req->id_transaksi,
31                 'id_jenis' => $req->id_jenis,
32                 'qty' => $req->qty,
33                 'subtotal' => $subtotal,
34             ]);
35             if($simpan){
36                 $data['status']="Berhasil";
37                 $data['message']="Data berhasil disimpan!";
38                 return Response()->json($data);
39             }else{
40                 $data['status']="Gagal";
41                 $data['message']="Data gagal disimpan!";
42                 return Response()->json($data);
43             }
44         } else {
45             $data['status']="Gagal";
46             $data['message']="Anda bukan Petugas!";
47             return Response()->json($data);
48         }
49     }
50 }
51
52 }
```



```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Detail;
7 use App\JenisCuci;
8 use Illuminate\Support\Facades\Hash;
9 use Illuminate\Support\Facades\Validator;
10 use Tymon\JWTAuth;
11 use Auth;
12 use Tymon\JWTAuth\Exceptions\JWTException;
13
14 class detailcontroller extends Controller
15 {
16     public function store(Request $req){
17         if (Auth::user()->level=='petugas') {
18             $validator=Validator::make($req->all(),[
19                 'id_transaksi' => 'required',
20                 'id_jenis' => 'required',
21                 'qty' => 'required',
22             ]);
23             if($validator->fails()){
24                 return Response()->json($validator->errors());
25             }
26             $harga = JenisCuci::where('id', $req->id_jenis)->first();
27             $subtotal = $harga->harga_per_kilo * $req->qty;
28
29             $simpan=Detail::create([
30                 'id_transaksi' => $req->id_transaksi,
31                 'id_jenis' => $req->id_jenis,
32                 'qty' => $req->qty,
33                 'subtotal' => $subtotal,
34             ]);
35             if($simpan){
36                 $data['status']="Berhasil";
37                 $data['message']="Data berhasil disimpan!";
38                 return Response()->json($data);
39             }else{
40                 $data['status']="Gagal";
41                 $data['message']="Data gagal disimpan!";
42                 return Response()->json($data);
43             }
44         }
45     }
46 }
```

Transaksi

NEW

Runner

Import

BuilderTeam Library

IN SYNC

Saddam Ra...

There's a newer version of Postman which has free teams and many requested features! [Download](#)

Filter

HistoryCollections

Clear all

▼ Today

POSThttp://localhost/laundry/public/api/ nota

POSThttp://localhost/laundry/public/api/ nota

POSThttp://localhost/laundry/public/api/ nota

POSThttp://localhost/laundry/public/api/ login

▼ February 26

POSThttp://localhost/laundry/public/api/ nota

POSThttp://localhost/laundry/public/api/ detail

POSThttp://localhost/laundry/public/api/ transaksi

POSThttp://localhost/laundry/public/api/ transaksi

POSThttp://localhost/laundry/public/api/ login

POSThttp://localhost/laundry/public/api/ transaksi

http://localhost/laundhttp://localhost/laund+***

No Environment

POST

http://localhost/laundry/public/api/nota

Params

Send

Save

tgl_transaksi

2020-02-27

tgl_selesai

2020-02-28

New key

Value

Description

Body

Cookies

Headers (11)

Test Results

Status: 200 OK

Time: 822 ms

PrettyRawPreviewJSON

```
1 {
2   "data_transaksi": [
3     {
4       "Tanggal": "2020-02-27",
5       "Nama Pelanggan": "Saddam",
6       "Alamat": "Jl.Dirgantara",
7       "No Telp": "082140322672",
8       "Deadline": "2020-02-28",
9       "Grand Total": {
10        "grandtotal": 36000
11      },
12      "Detail": [
13        {
14          "id": 2,
15          "id_transaksi": 2,
16          "id_jenis": 2,
17          "qty": 6,
18          "subtotal": 36000,
19          "nama_jenis": "CUCI BIASA",
20          "harga_per_kilo": 6000
21        }
22      ]
23    }
24  ]
25 }
```