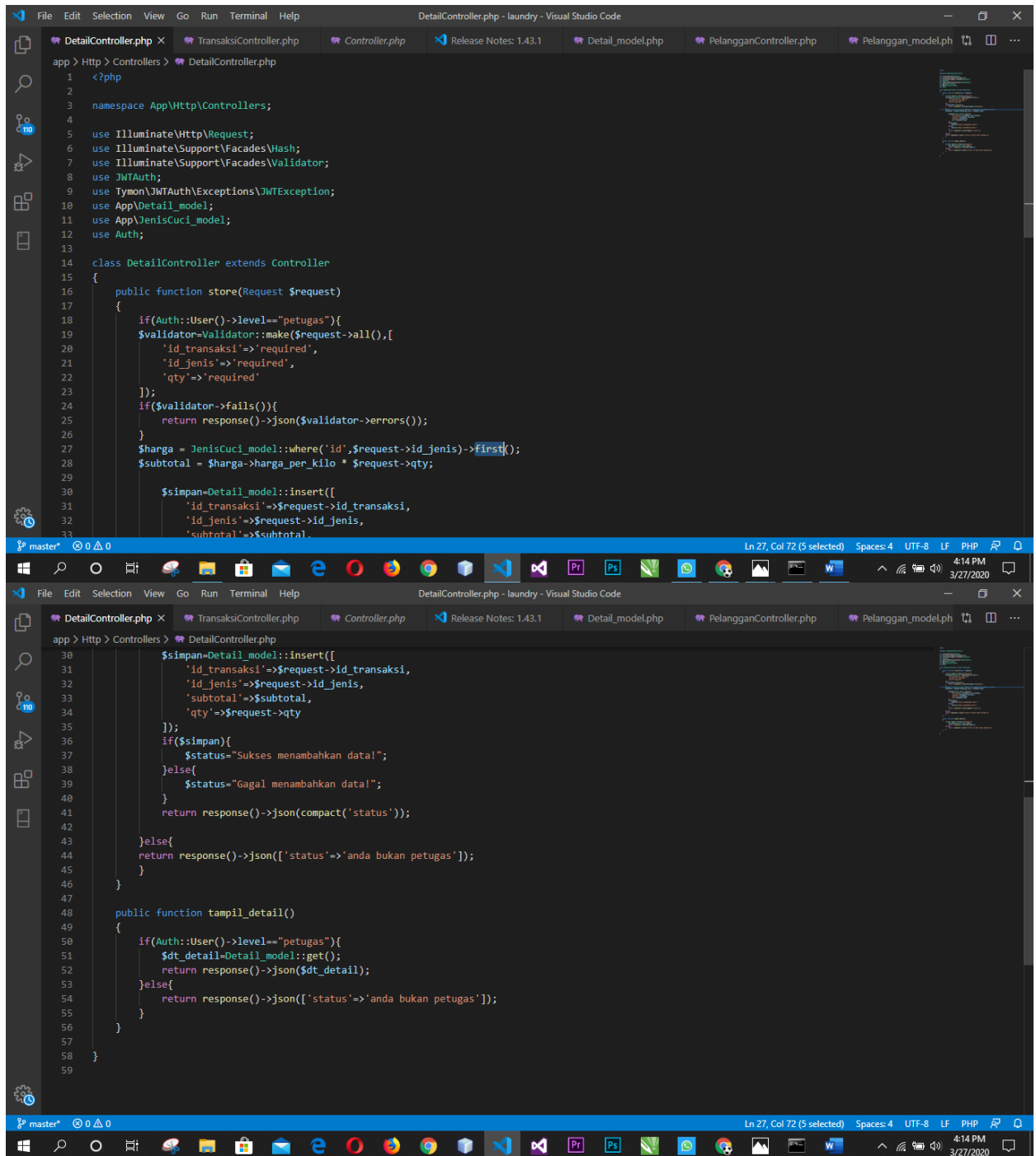
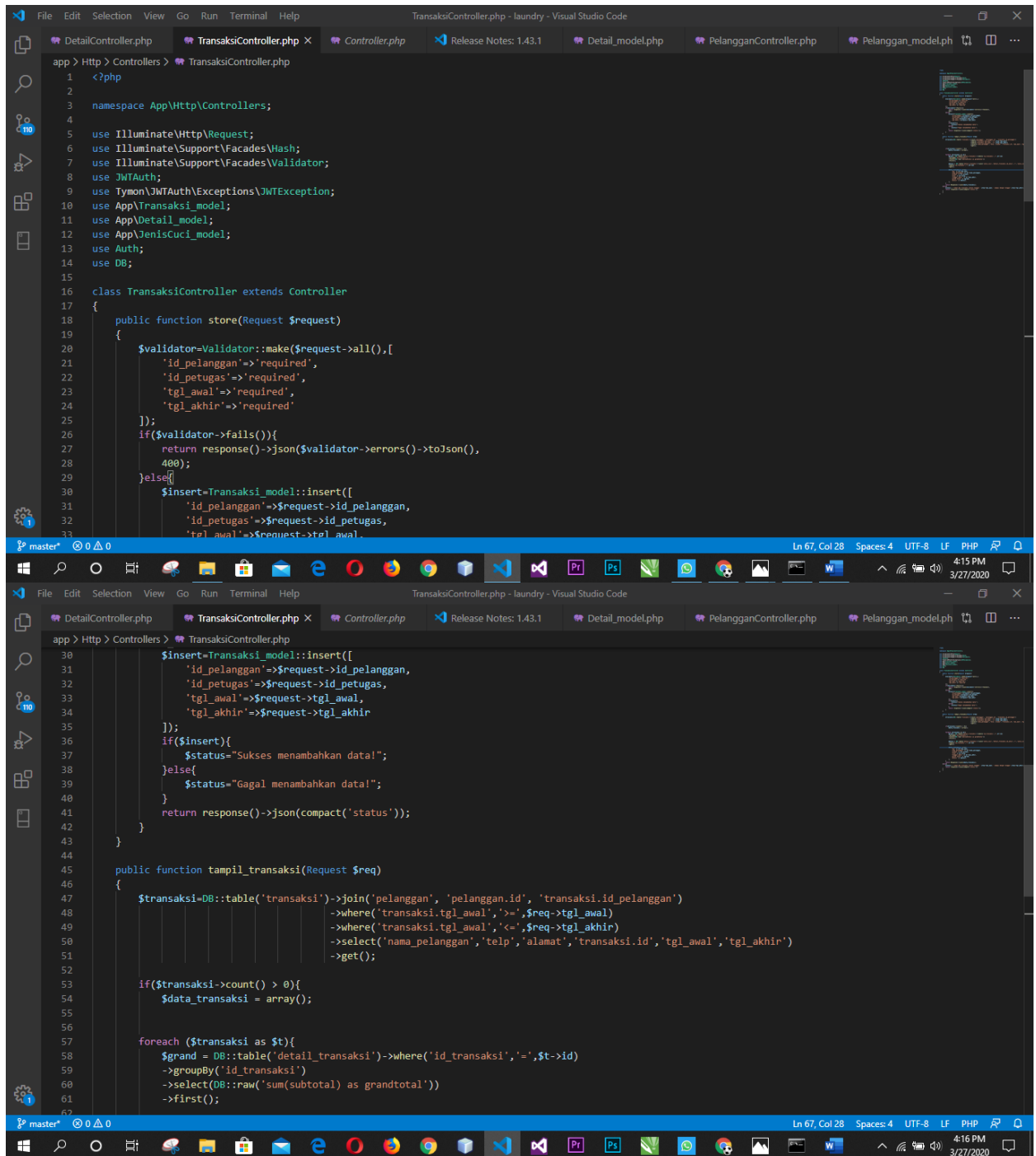


- **Script Controller DetailController.php**



```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\Hash;
7 use Illuminate\Support\Facades\Validator;
8 use JWTAuth;
9 use Tymon\JWTAuth\Exceptions\JWTException;
10 use App\Detail_model;
11 use App\JenisCuci_model;
12 use Auth;
13
14 class DetailController extends Controller
15 {
16     public function store(Request $request)
17     {
18         if(Auth::User()->level=="petugas"){
19             $validator=Validator::make($request->all(),[
20                 'id_transaksi'=>'required',
21                 'id_jenis'=>'required',
22                 'qty'=>'required'
23             ]);
24             if($validator->fails()){
25                 return response()->json($validator->errors());
26             }
27             $harga = JenisCuci_model::where('id',$request->id_jenis)->first();
28             $subtotal = $harga->harga_per_kilo * $request->qty;
29
30             $simpan=Detail_model::insert([
31                 'id_transaksi'=>$request->id_transaksi,
32                 'id_jenis'=>$request->id_jenis,
33                 'subtotal'=>$subtotal,
34                 'qty'=>$request->qty
35             ]);
36             if($simpan){
37                 $status="Sukses menambahkan data!";
38             }else{
39                 $status="Gagal menambahkan data!";
40             }
41             return response()->json(compact('status'));
42         }else{
43             return response()->json(['status'=>'anda bukan petugas']);
44         }
45     }
46
47     public function tampil_detail()
48     {
49         if(Auth::User()->level=="petugas"){
50             $dt_detail=Detail_model::get();
51             return response()->json($dt_detail);
52         }else{
53             return response()->json(['status'=>'anda bukan petugas']);
54         }
55     }
56 }
57
58 }
59
```

- **Script Controller TransaksiController.php**

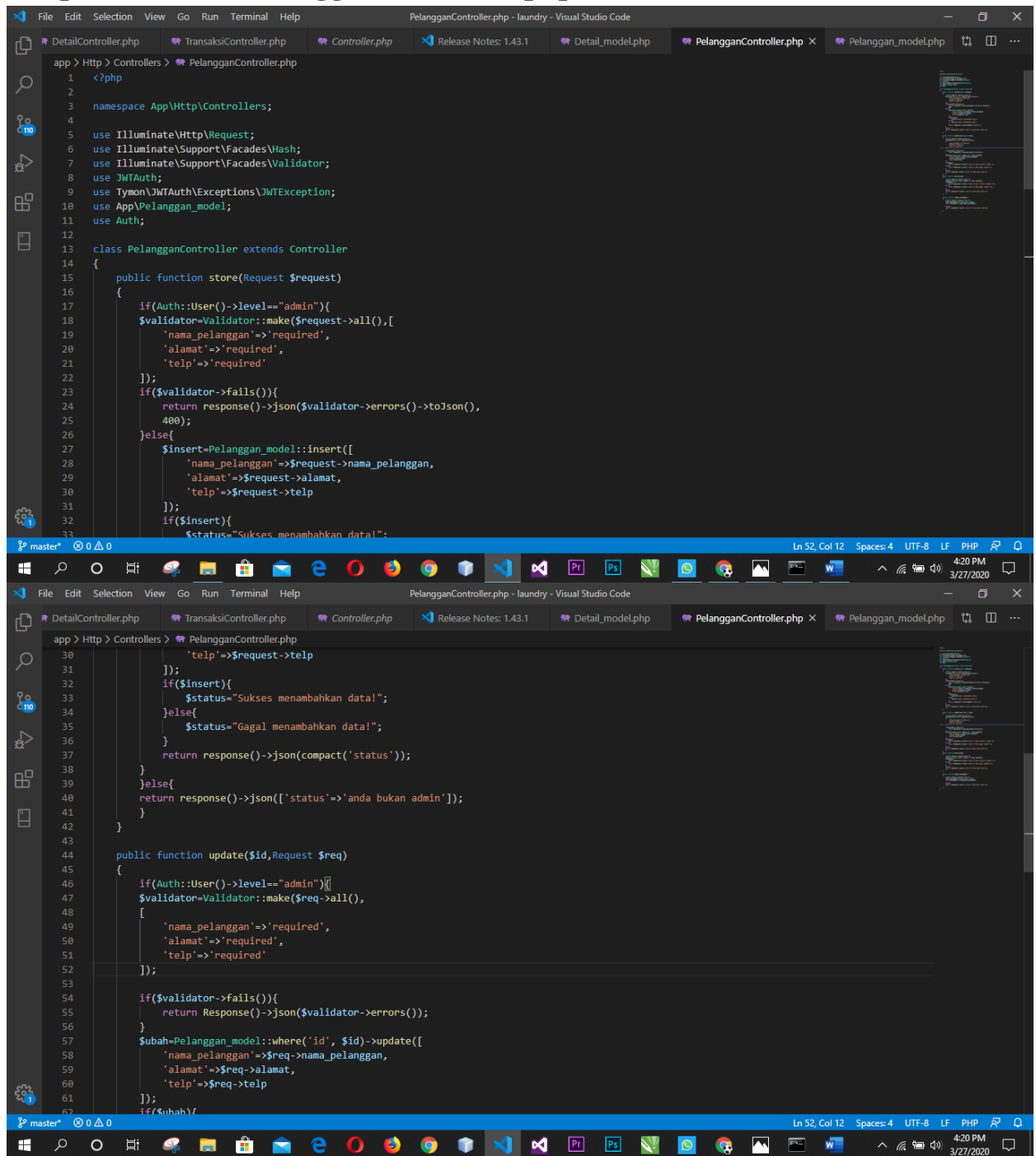


```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\Hash;
7 use Illuminate\Support\Facades\Validator;
8 use JWTAuth;
9 use Tymon\JWTAuth\Exceptions\JWTException;
10 use App\Transaksi_model;
11 use App\Detail_model;
12 use App\JenisCuci_model;
13 use Auth;
14 use DB;
15
16 class TransaksiController extends Controller
17 {
18     public function store(Request $request)
19     {
20         $validator=Validator::make($request->all(),[
21             'id_pelanggan'=>'required',
22             'id_petugas'=>'required',
23             'tgl_awal'=>'required',
24             'tgl_akhir'=>'required'
25         ]);
26         if($validator->fails()){
27             return response()->json($validator->errors()->toJson(),
28                 400);
29         }else{
30             $insert=Transaksi_model::insert([
31                 'id_pelanggan'=>$request->id_pelanggan,
32                 'id_petugas'=>$request->id_petugas,
33                 'tgl_awal'=>$request->tgl_awal,
34                 'tgl_akhir'=>$request->tgl_akhir
35             ]);
36             if($insert){
37                 $status="Sukses menambahkan data!";
38             }else{
39                 $status="Gagal menambahkan data!";
40             }
41             return response()->json(compact('status'));
42         }
43     }
44
45     public function tampil_transaksi(Request $req)
46     {
47         $transaksi=DB::table('transaksi')->join('pelanggan', 'pelanggan.id', 'transaksi.id_pelanggan')
48             ->where('transaksi.tgl_awal','>=', $req->tgl_awal)
49             ->where('transaksi.tgl_akhir','<=', $req->tgl_akhir)
50             ->select('nama_pelanggan','telp','alamat','transaksi.id','tgl_awal','tgl_akhir')
51             ->get();
52
53         if($transaksi->count() > 0){
54             $data_transaksi = array();
55
56             foreach ($transaksi as $t){
57                 $grand = DB::table('detail_transaksi')->where('id_transaksi','=',$t->id)
58                     ->groupBy('id_transaksi')
59                     ->select(DB::raw('sum(subtotal) as grandtotal'))
60                     ->first();
61             }
62         }
63     }
64 }
```

```
TransaksiController.php - laundry - Visual Studio Code
DetailController.php TransaksiController.php Controller.php Release Notes: 1.43.1 Detail_model.php PelangganController.php Pelanggan_model.php

app > Http > Controllers > TransaksiController.php
56
57     foreach ($transaksi as $t){
58         $grand = DB::table('detail_transaksi')->where('id_transaksi','=',$t->id)
59         ->groupBy('id_transaksi')
60         ->select(DB::raw('sum(subtotal) as grandtotal'))
61         ->first();
62
63         $detail = DB::table('detail_transaksi')->join('jenis_cuci','detail_transaksi.id_jenis','=',$jenis_cuci.id')
64         ->where('id_transaksi','=',$t->id)
65         ->get();
66
67         $data_transaksi = array(
68             'tgl' => $t->tgl_awal,
69             'nama pelanggan' => $t->nama_pelanggan,
70             'alamat' => $t->alamat,
71             'telp' => $t->telp,
72             'tanggal ambil' => $t->tgl_akhir,
73             'grand total' => $grand,
74             'detail' => $detail,
75         );
76     }
77     return Response()->json($data_transaksi);
78 }else{
79     $status = 'tidak ada transaksi antara tanggal '.$req->tgl_awal.' sampai dengan tanggal '.$req->tgl_akhir;
80     return response()->json(compact('status'));
81 }
82 }
83 }
84 }
```

- **Script Controller PelangganController.php**



```
app > Http > Controllers > PelangganController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use Illuminate\Support\Facades\Hash;
7  use Illuminate\Support\Facades\Validator;
8  use JWTAuth;
9  use Tymon\JWTAuth\Exceptions\JWTException;
10 use App\Pelanggan_model;
11 use Auth;
12
13 class PelangganController extends Controller
14 {
15     public function store(Request $request)
16     {
17         if(Auth::User()->level=="admin"){
18             $validator=Validator::make($request->all(),[
19                 'nama_pelanggan'=>'required',
20                 'alamat'=>'required',
21                 'telp'=>'required'
22             ]);
23             if($validator->fails()){
24                 return response()->json($validator->errors()->toJson(),
25                     400);
26             }else{
27                 $insert=Pelanggan_model::insert([
28                     'nama_pelanggan'=>$request->nama_pelanggan,
29                     'alamat'=>$request->alamat,
30                     'telp'=>$request->telp
31                 ]);
32                 if($insert){
33                     $status="Sukses menambahkan data!";
34                 }else{
35                     $status="Gagal menambahkan data!";
36                 }
37                 return response()->json(compact('status'));
38             }
39         }else{
40             return response()->json(['status'=>'anda bukan admin']);
41         }
42     }
43
44     public function update($id,Request $req)
45     {
46         if(Auth::User()->level=="admin"){
47             $validator=Validator::make($req->all(),
48                 [
49                     'nama_pelanggan'=>'required',
50                     'alamat'=>'required',
51                     'telp'=>'required'
52                 ]);
53             if($validator->fails()){
54                 return Response()->json($validator->errors());
55             }
56             $ubah=Pelanggan_model::where('id', $id)->update([
57                 'nama_pelanggan'=>$req->nama_pelanggan,
58                 'alamat'=>$req->alamat,
59                 'telp'=>$req->telp
60             ]);
61             if($ubah){
62                 $status="Sukses update data!";
63             }else{
64                 $status="Gagal update data!";
65             }
66             return response()->json(compact('status'));
67         }else{
68             return response()->json(['status'=>'anda bukan admin']);
69         }
70     }
71 }
```

The image displays two instances of the Visual Studio Code editor, showing PHP code for a laundry management system. The top window shows the 'update' method in 'PelangganController.php', and the bottom window shows the 'destroy' and 'tampil_pelanggan' methods in the same file.

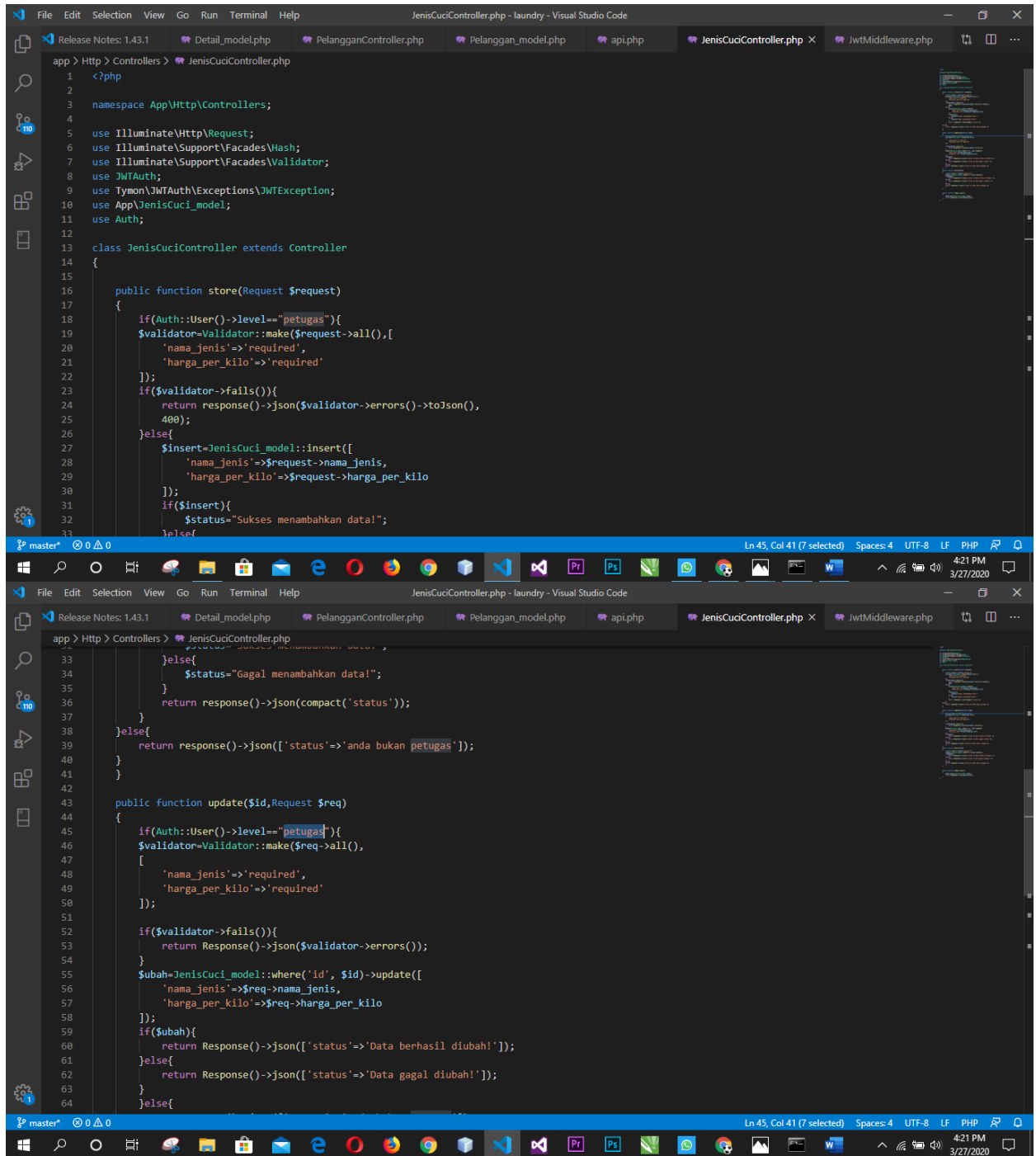
Top Window: PelangganController.php - update method

```
54     if($validator->fails()){
55         return Response()->json($validator->errors());
56     }
57     $ubah=Pelanggan_model::where('id', $id)->update([
58         'nama_pelanggan'=>$req->nama_pelanggan,
59         'alamat'=>$req->alamat,
60         'telp'=>$req->telp
61     ]);
62     if($ubah){
63         return Response()->json(['status'=>'Data berhasil diubah!']);
64     }else{
65         return Response()->json(['status'=>'Data gagal diubah!']);
66     }
67 }
68 return response()->json(['status'=>'anda bukan admin']);
69 }
70 }
71 public function destroy($id)
72 {
73     if(Auth::User()->level=="admin"){
74         $hapus=Pelanggan_model::where('id',$id)->delete();
75     }if($hapus){
76         return Response()->json(['status'=>'Data berhasil dihapus!']);
77     }else{
78         return Response()->json(['status'=>'Data gagal dihapus!']);
79     }
80 }
81 return response()->json(['status'=>'anda bukan admin']);
82 }
83 }
84
85 public function tampil_pelanggan()
```

Bottom Window: PelangganController.php - destroy and tampil_pelanggan methods

```
75     if($hapus){
76         return Response()->json(['status'=>'Data berhasil dihapus!']);
77     }else{
78         return Response()->json(['status'=>'Data gagal dihapus!']);
79     }
80 }
81 return response()->json(['status'=>'anda bukan admin']);
82 }
83 }
84
85 public function tampil_pelanggan()
86 {
87     if(Auth::User()->level=="admin"){
88         $data_pelanggan=Pelanggan_model::get();
89         return Response()->json($data_pelanggan);
90     }
91     }else{
92         return response()->json(['status'=>'anda bukan admin']);
93     }
94 }
95 }
96 }
```

- **Script Controller JenisCuciController.php**



```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\Hash;
7 use Illuminate\Support\Facades\Validator;
8 use JWTAuth;
9 use Tymon\JWTAuth\Exceptions\JWTException;
10 use App\JenisCuci_model;
11 use Auth;
12
13 class JenisCuciController extends Controller
14 {
15
16     public function store(Request $request)
17     {
18         if(Auth::User()->level=="petugas"){
19             $validator=Validator::make($request->all(),[
20                 'nama_jenis'=>'required',
21                 'harga_per_kilo'=>'required'
22             ]);
23             if($validator->fails()){
24                 return response()->json($validator->errors()->toJson(),
25                     400);
26             }else{
27                 $insert=JenisCuci_model::insert([
28                     'nama_jenis'=>$request->nama_jenis,
29                     'harga_per_kilo'=>$request->harga_per_kilo
30                 ]);
31                 if($insert){
32                     $status="Sukses menambahkan data!";
33                 }else{
34                     $status="Gagal menambahkan data!";
35                 }
36                 return response()->json(compact('status'));
37             }
38         }else{
39             return response()->json(['status'=>'anda bukan petugas']);
40         }
41     }
42
43     public function update($id,Request $req)
44     {
45         if(Auth::User()->level=="petugas"){
46             $validator=Validator::make($req->all(),
47             [
48                 'nama_jenis'=>'required',
49                 'harga_per_kilo'=>'required'
50             ]);
51
52             if($validator->fails()){
53                 return Response()->json($validator->errors());
54             }
55             $ubah=JenisCuci_model::where('id', $id)->update([
56                 'nama_jenis'=>$req->nama_jenis,
57                 'harga_per_kilo'=>$req->harga_per_kilo
58             ]);
59             if($ubah){
60                 return Response()->json(['status'=>'Data berhasil diubah!']);
61             }else{
62                 return Response()->json(['status'=>'Data gagal diubah!']);
63             }
64         }else{
65             return Response()->json(['status'=>'anda bukan petugas']);
66         }
67     }
68 }
```

```
File Edit Selection View Go Run Terminal Help
JenisCuciController.php - laundry - Visual Studio Code
Release Notes: 1.43.1 Detail_model.php PelangganController.php Pelanggan_model.php api.php JenisCuciController.php jwtMiddleware.php
app > Http > Controllers > JenisCuciController.php
59
60     return Response()->json(['status'=>'Data berhasil diubah!']);
61 }else{
62     return Response()->json(['status'=>'Data gagal diubah!']);
63 }
64 }else{
65     return response()->json(['status'=>'anda bukan petugas']);
66 }
67 }
68 public function destroy($id)
69 {
70     if(Auth::User()->level=="petugas"){
71         $hapus=JenisCuci_model::where('id',$id)->delete();
72         if($hapus){
73             return Response()->json(['status'=>'Data berhasil dihapus!']);
74         }else{
75             return Response()->json(['status'=>'Data gagal dihapus!']);
76         }
77     }else{
78         return response()->json(['status'=>'anda bukan petugas']);
79     }
80 }
81
82
83 public function tampil_jenis()
84 {
85     $data_jenis=JenisCuci_model::get();
86     return Response()->json($data_jenis);
87 }
88
89 }
```

Ln 45, Col 41 (7 selected) Spaces: 4 UTF-8 LF PHP 4:21 PM 3/27/2020

- Script Controller UserController.php

```
UserController.php
app > Http > Controllers > UserController.php
30 {
31     $validator = Validator::make($request->all(), [
32         'nama_petugas' => 'required|string|max:255',
33         'telp' => 'required|string|max:11',
34         'username' => 'required|string|max:255',
35         'password' => 'required|string|min:6|confirmed',
36         'level' => 'required|string',
37     ]);
38
39     if($validator->fails()){
40         return response()->json($validator->errors()->toJson(), 400);
41     }
42
43     $user = User::create([
44         'nama_petugas' => $request->get('nama_petugas'),
45         'telp' => $request->get('telp'),
46         'username' => $request->get('username'),
47         'password' => Hash::make($request->get('password')),
48         'level' => $request->get('level'),
49     ]);
50
51     $token = JWTAuth::fromUser($user);
52
53     return response()->json(compact('user', 'token'), 201);
54 }
55
56 public function update($id, Request $req)
57 {
58     $validator=Validator::make($req->all(),
59     [
60         'nama_petugas'=>'required',
61         'telp'=>'required',
62         'username'=>'required'
```

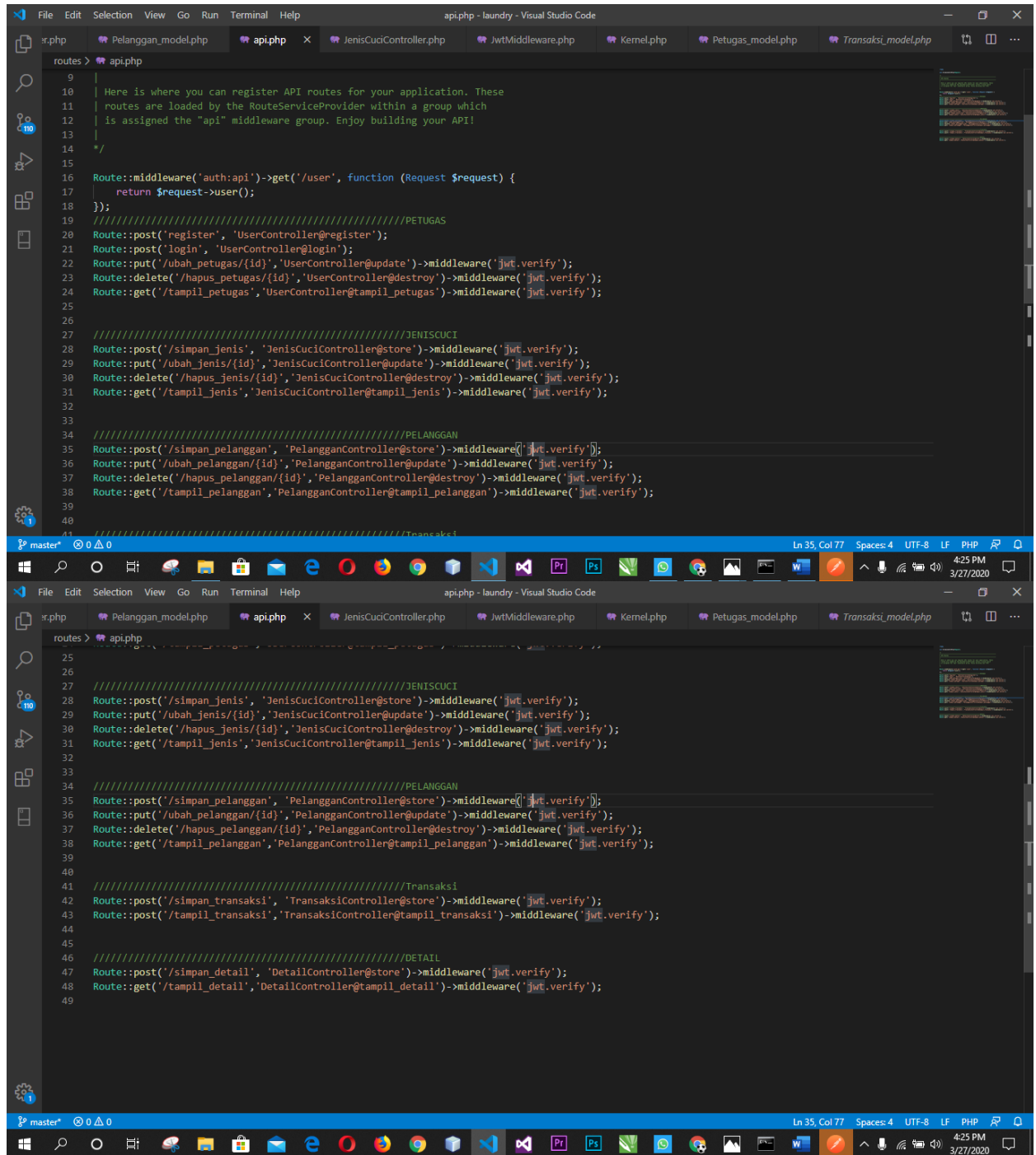
```
UserController.php
app > Http > Controllers > UserController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\User;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Hash;
8  use Illuminate\Support\Facades\Validator;
9  use JWTAuth;
10 use Tymon\JWTAuth\Exceptions\JWTException;
11
12 class UserController extends Controller
13 {
14     public function login(Request $request)
15     {
16         $credentials = $request->only('username', 'password');
17
18         try {
19             if (! $token = JWTAuth::attempt($credentials)) {
20                 return response()->json(['error' => 'invalid_credentials'], 400);
21             }
22         } catch (JWTException $e) {
23             return response()->json(['error' => 'could_not_create_token'], 500);
24         }
25
26         return response()->json(compact('token'));
27     }
28
29     public function register(Request $request)
30     {
31         $validator = Validator::make($request->all(), [
32             'nama_petugas' => 'required|string|max:255',
33             'telp' => 'required|string|max:11',
```



```
UserController.php - laundry - Visual Studio Code
UserController.php x DetailController.php TransaksiController.php Controller.php Release Notes: 1.43.1 Detail_model.php PelangganController.php

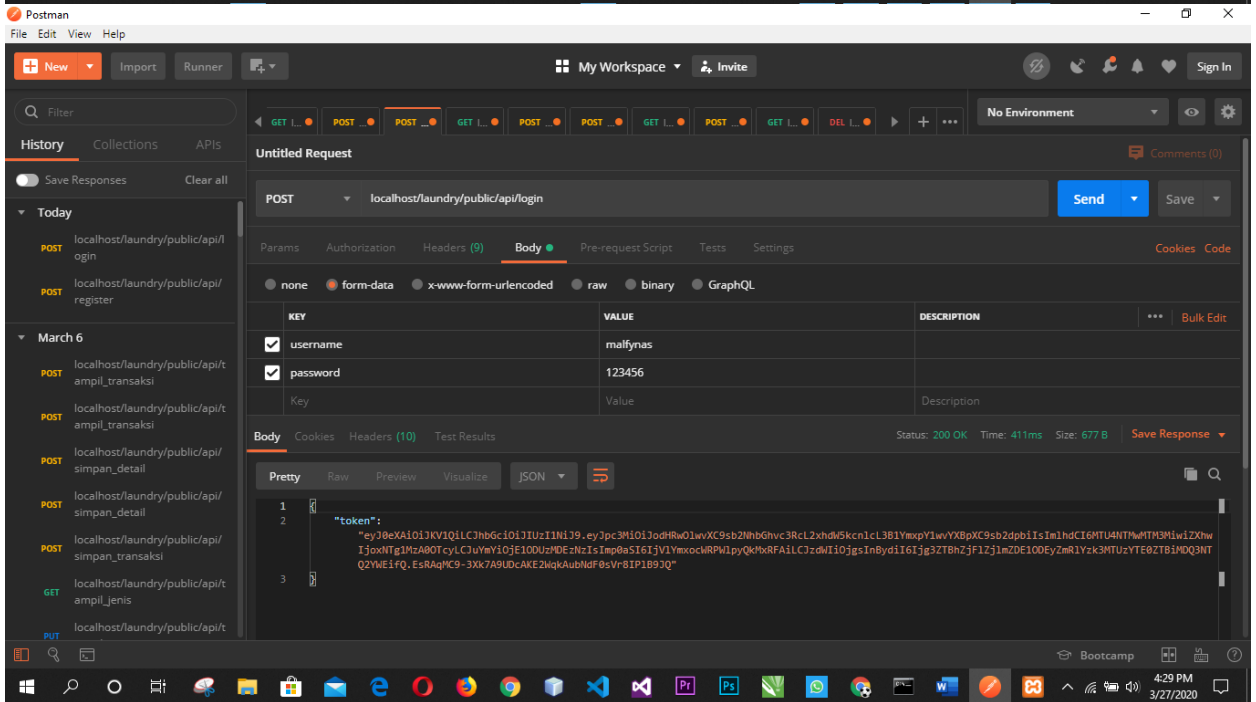
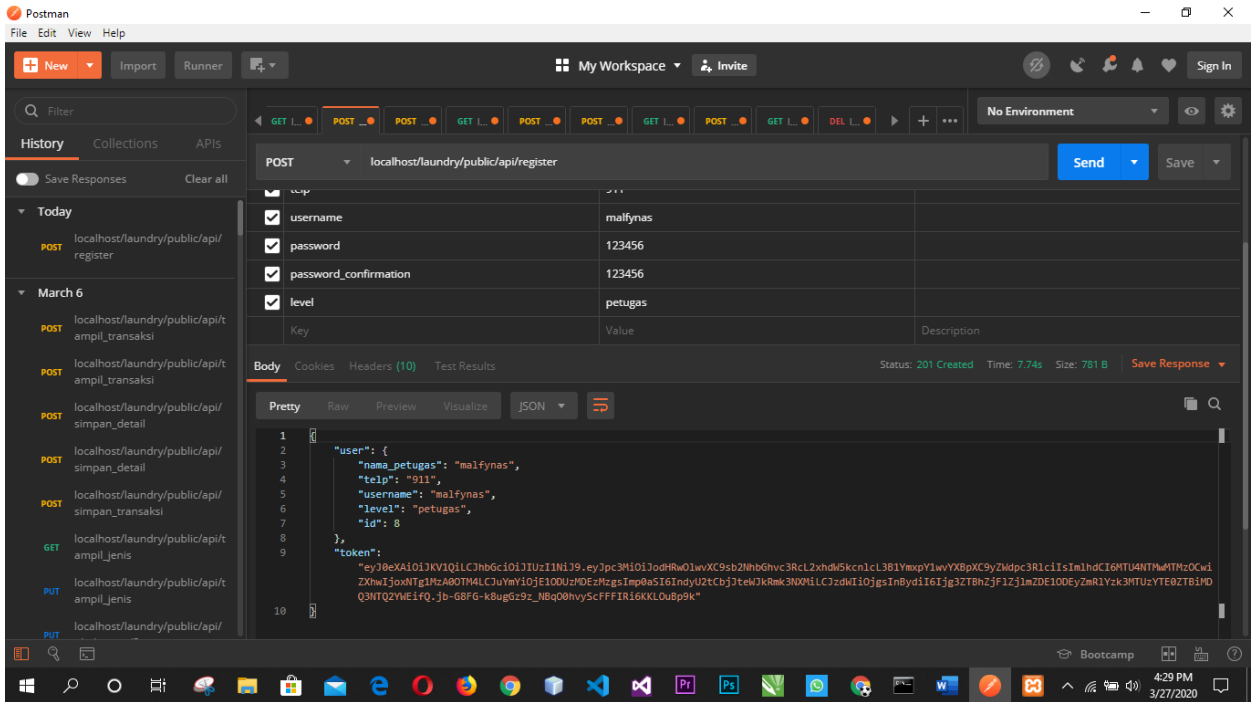
app > Http > Controllers > UserController.php
62     'username'=>'required',
63     'password'=>'required',
64     'level'=>'required'
65 ];
66
67     if($validator->fails()){
68         return Response()->json($validator->errors());
69     }
70     $ubah=User::where('id', $id)->update([
71         'nama_petugas'=>$req->nama_petugas,
72         'telp'=>$req->telp,
73         'username'=>$req->username,
74         'password'=>$req->password,
75         'level'=>$req->level
76     ]);
77     if($ubah){
78         return Response()->json(['status'=>'Sukses mengubah data!']);
79     }else{
80         return Response()->json(['status'=>'Sukses mengubah data!']);
81     }
82 }
83
84 public function destroy($id)
85 {
86     $hapus=User::where('id',$id)->delete();
87     if($hapus){
88         return Response()->json(['status'=>'Sukses menghapus data!']);
89     }else{
90         return Response()->json(['status'=>'Gagal menghapus data!']);
91     }
92 }
93
94
95 public function getAuthenticatedUser()
96 {
97     try {
98         if (! $user = JWTAuth::parseToken()->authenticate()) {
99             return response()->json(['user_not_found'], 404);
100         }
101     } catch (Tymon\JWTAuth\Exceptions\TokenExpiredException $e) {
102         return response()->json(['token_expired'], $e->getStatusCode());
103     } catch (Tymon\JWTAuth\Exceptions\TokenInvalidException $e) {
104         return response()->json(['token_invalid'], $e->getStatusCode());
105     } catch (Tymon\JWTAuth\Exceptions\JWTException $e) {
106         return response()->json(['token_absent'], $e->getStatusCode());
107     }
108     return response()->json(compact('user'));
109 }
```

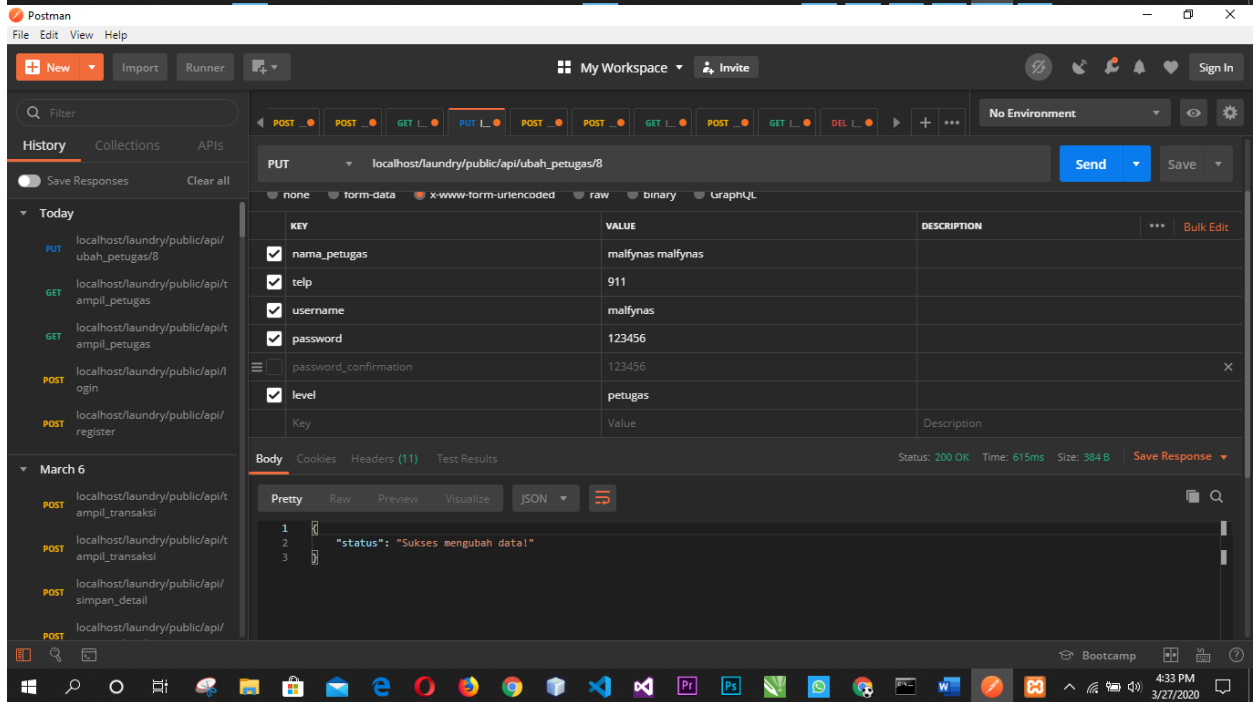
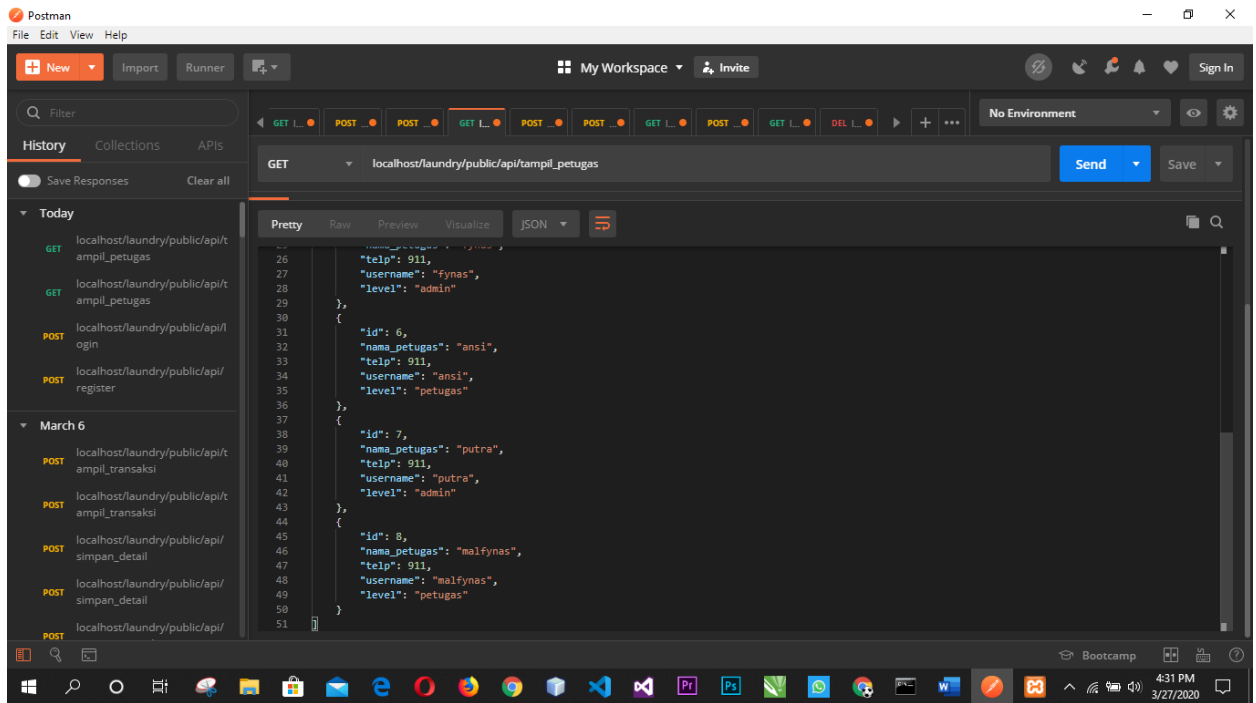
- **Script Controller api.php**

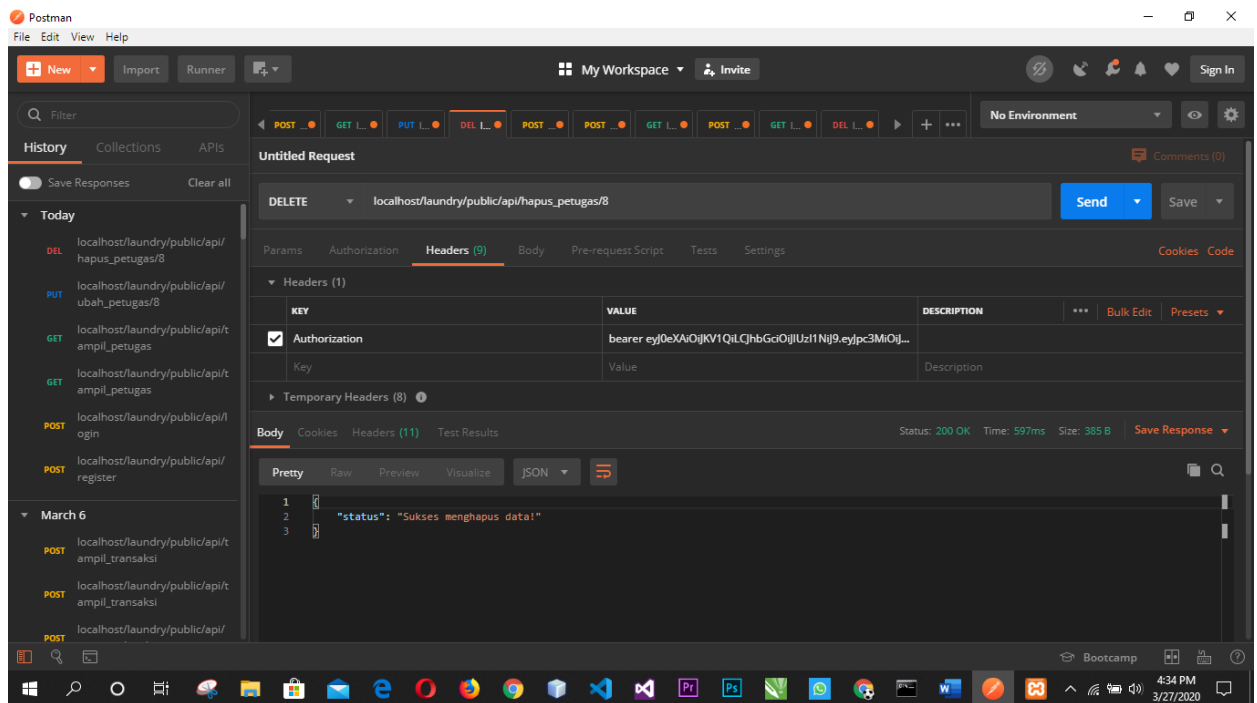


```
9
10 Here is where you can register API routes for your application. These
11 routes are loaded by the RouteServiceProvider within a group which
12 is assigned the "api" middleware group. Enjoy building your API!
13
14 */
15
16 Route::middleware('auth:api')->get('/user', function (Request $request) {
17     return $request->user();
18 });
19
20 ///////////////////////////////////////////////////PETUGAS
21 Route::post('register', 'UserController@register');
22 Route::post('login', 'UserController@login');
23 Route::put('/ubah_petugas/{id}', 'UserController@update')->middleware('jwt.verify');
24 Route::delete('/hapus_petugas/{id}', 'UserController@destroy')->middleware('jwt.verify');
25 Route::get('/tampil_petugas', 'UserController@tampil_petugas')->middleware('jwt.verify');
26
27 ///////////////////////////////////////////////////JENISCUCI
28 Route::post('/simpan_jenis', 'JenisCuciController@store')->middleware('jwt.verify');
29 Route::put('/ubah_jenis/{id}', 'JenisCuciController@update')->middleware('jwt.verify');
30 Route::delete('/hapus_jenis/{id}', 'JenisCuciController@destroy')->middleware('jwt.verify');
31 Route::get('/tampil_jenis', 'JenisCuciController@tampil_jenis')->middleware('jwt.verify');
32
33
34 ///////////////////////////////////////////////////PELANGGAN
35 Route::post('/simpan_pelanggan', 'PelangganController@store')->middleware('jwt.verify');
36 Route::put('/ubah_pelanggan/{id}', 'PelangganController@update')->middleware('jwt.verify');
37 Route::delete('/hapus_pelanggan/{id}', 'PelangganController@destroy')->middleware('jwt.verify');
38 Route::get('/tampil_pelanggan', 'PelangganController@tampil_pelanggan')->middleware('jwt.verify');
39
40
41 ///////////////////////////////////////////////////Transaksi
42 Route::post('/simpan_transaksi', 'TransaksiController@store')->middleware('jwt.verify');
43 Route::post('/tampil_transaksi', 'TransaksiController@tampil_transaksi')->middleware('jwt.verify');
44
45
46 ///////////////////////////////////////////////////DETAIL
47 Route::post('/simpan_detail', 'DetailController@store')->middleware('jwt.verify');
48 Route::get('/tampil_detail', 'DetailController@tampil_detail')->middleware('jwt.verify');
49
```

- **CRUD Petugas + Login**







• CRUD Jenis

The first screenshot shows a POST request to `localhost/laundry/public/api/simpan_jenis` with a body containing `nama_jenis: Cuci Basah` and `harga_per_kilo: 3000`. The response status is 200 OK with the message `"status": "Sukses menambahkan data!"`.

The second screenshot shows a GET request to `localhost/laundry/public/api/tampil_jenis` returning a JSON array of three laundry items:

```
1 {
2   "id": 1,
3   "nama_jenis": "Cuci Kering Setrika",
4   "harga_per_kilo": 4500
5 },
6 {
7   "id": 2,
8   "nama_jenis": "Cuci Kering",
9   "harga_per_kilo": 4000
10 },
11 {
12   "id": 3,
13   "nama_jenis": "Cuci Basah",
14   "harga_per_kilo": 3000
15 }
16 }
```

Postman

File Edit View Help

New Import Runner My Workspace Invite

Filter

History Collections APIs

Save Responses Clear all

Today

- PUT localhost/laundry/public/api/ubah_jenis/4
- PUT localhost/laundry/public/api/ubah_jenis/4
- PUT localhost/laundry/public/api/ubah_jenis/4
- GET localhost/laundry/public/api/ambil_jenis
- PUT localhost/laundry/public/api/ubah_jenis/4
- PUT localhost/laundry/public/api/ubah_jenis/4
- DEL localhost/laundry/public/api/hapus_jenis/4
- POST localhost/laundry/public/api/simpan_jenis
- DEL localhost/laundry/public/api/simpan_jenis
- localhost/laundry/public/api/

Untitled Request

PUT localhost/laundry/public/api/ubah_jenis/4

Send Save

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> nama_jenis	Cuci Basah	
<input checked="" type="checkbox"/> harga_per_kilo	90000	
Key	Value	Description

Body Cookies Headers (11) Test Results Status: 200 OK Time: 259ms Size: 354 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "Data berhasil diubah!"
3 }
```

Postman

File Edit View Help

New Import Runner My Workspace Invite

Filter

History Collections APIs

Save Responses Clear all

Today

- DEL localhost/laundry/public/api/hapus_jenis/3
- DEL localhost/laundry/public/api/hapus_jenis/3
- GET localhost/laundry/public/api/ambil_jenis
- POST localhost/laundry/public/api/simpan_jenis
- POST localhost/laundry/public/api/simpan_jenis
- POST localhost/laundry/public/api/simpan_pelanggan
- POST localhost/laundry/public/api/simpan_pelanggan
- POST localhost/laundry/public/api/simpan_pelanggan
- POST localhost/laundry/public/api/simpan_pelanggan
- localhost/laundry/public/api/

DELETE localhost/laundry/public/api/hapus_jenis/3

Send Save

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "Data berhasil dihapus!"
3 }
```

• CRUD Pelanggan

The image displays two screenshots of the Postman API client interface, demonstrating a REST API for customer management.

Top Screenshot: GET Request

The top screenshot shows a GET request to the endpoint `localhost/laundry/public/api/tampil_pelanggan`. The response is a JSON array of four customer records:

```
1 {
2   "id": 1,
3   "nama_pelanggan": "Athallah",
4   "alamat": "Jl. Mangga",
5   "telp": "098787"
6 },
7
8 {
9   "id": 2,
10  "nama_pelanggan": "putra",
11  "alamat": "malang",
12  "telp": "08123"
13 },
14
15 {
16  "id": 3,
17  "nama_pelanggan": "ansi",
18  "alamat": "malang",
19  "telp": "08123"
20 },
21
22 {
23  "id": 4,
24  "nama_pelanggan": "fynas",
25  "alamat": "malang",
26  "telp": "08123"
27 }
```

Bottom Screenshot: PUT Request

The bottom screenshot shows a PUT request to the endpoint `localhost/laundry/public/api/ubah_pelanggan/4`. The request body is a JSON object with the following data:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> nama_pelanggan	fynas fynas	
<input checked="" type="checkbox"/> alamat	malang	
<input checked="" type="checkbox"/> telp	08123	
Key	Value	Description

The response status is 200 OK, with a time of 270ms and a size of 384 B. The response body is a JSON object:

```
1 {
2   "status": "Data berhasil diubah!"
3 }
```


Postman

File Edit View Help

New Import Runner My Workspace Invite

Filter

History Collections APIs

Save Responses Clear all

Today

- DEL localhost/laundry/public/api/hapus_pelanggan/4
- PUT localhost/laundry/public/api/ubah_pelanggan/4
- GET localhost/laundry/public/api/ampil_pelanggan
- POST localhost/laundry/public/api/simpan_pelanggan
- DEL localhost/laundry/public/api/hapus_jenis/4
- DEL localhost/laundry/public/api/hapus_jenis/4
- PUT localhost/laundry/public/api/ubah_jenis/4
- PUT localhost/laundry/public/api/ubah_jenis/4
- PUT localhost/laundry/public/api/ubah_jenis/4
- localhost/laundry/public/api/t

Untitled Request

DELETE localhost/laundry/public/api/hapus_pelanggan/4

Send Save

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> nama_pelanggan	fynas fynas	
<input checked="" type="checkbox"/> alamat	malang	
<input checked="" type="checkbox"/> telp	08123	
Key	Value	Description

Body Cookies Headers (11) Test Results Status: 200 OK Time: 252ms Size: 385 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "Data berhasil dihapus!"
3 }
```

Postman

File Edit View Help

New Import Runner My Workspace Invite

Filter

History Collections APIs

Save Responses Clear all

Today

- POST localhost/laundry/public/api/simpan_pelanggan
- DEL localhost/laundry/public/api/hapus_jenis/4
- DEL localhost/laundry/public/api/hapus_jenis/4
- PUT localhost/laundry/public/api/ubah_jenis/4
- PUT localhost/laundry/public/api/ubah_jenis/4
- PUT localhost/laundry/public/api/ubah_jenis/4
- GET localhost/laundry/public/api/ampil_jenis
- PUT localhost/laundry/public/api/ubah_jenis/4
- PUT localhost/laundry/public/api/ubah_jenis/4
- localhost/laundry/public/api/

Untitled Request

POST localhost/laundry/public/api/simpan_pelanggan

Send Save

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL

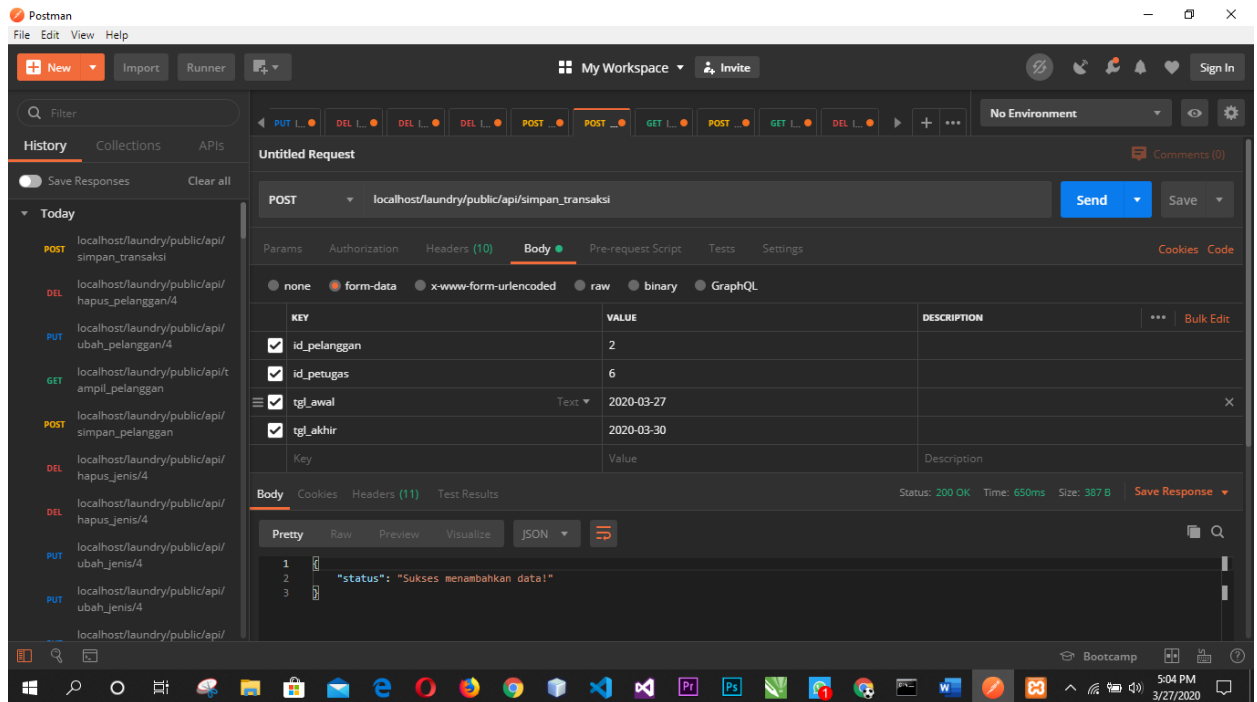
KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> nama_pelanggan	fynas	
<input checked="" type="checkbox"/> alamat	malang	
<input checked="" type="checkbox"/> telp	08123	
Key	Value	Description

Body Cookies Headers (11) Test Results Status: 200 OK Time: 304ms Size: 387 B Save Response

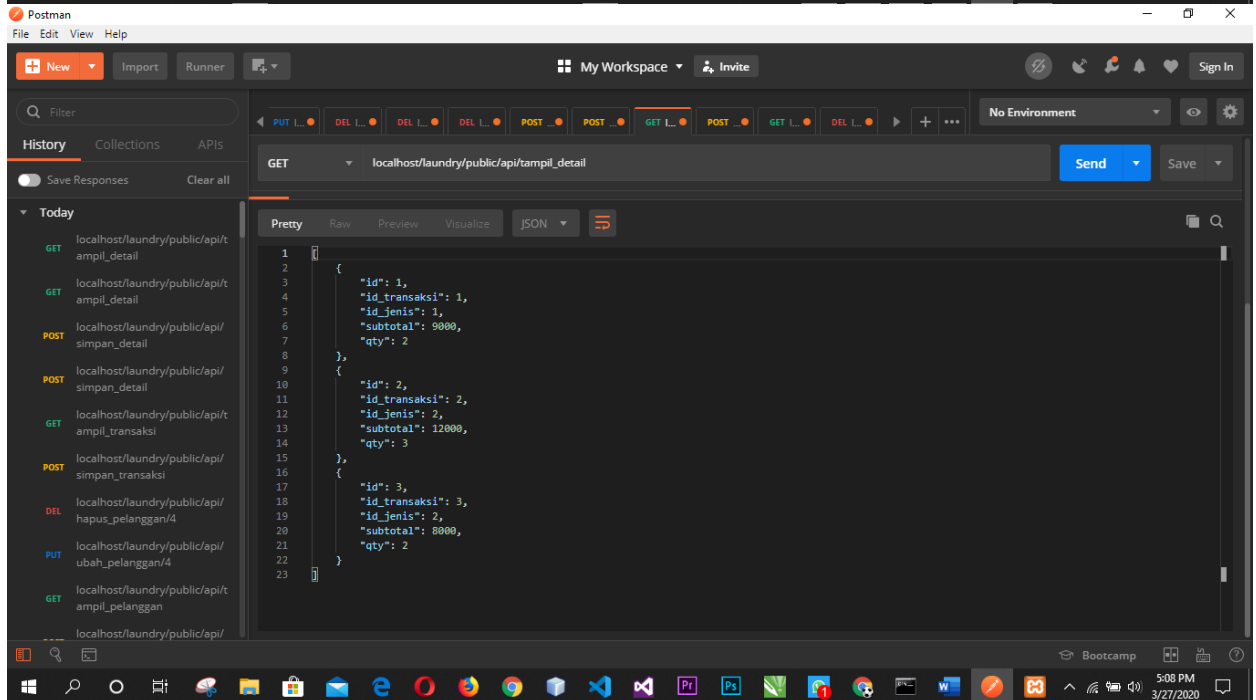
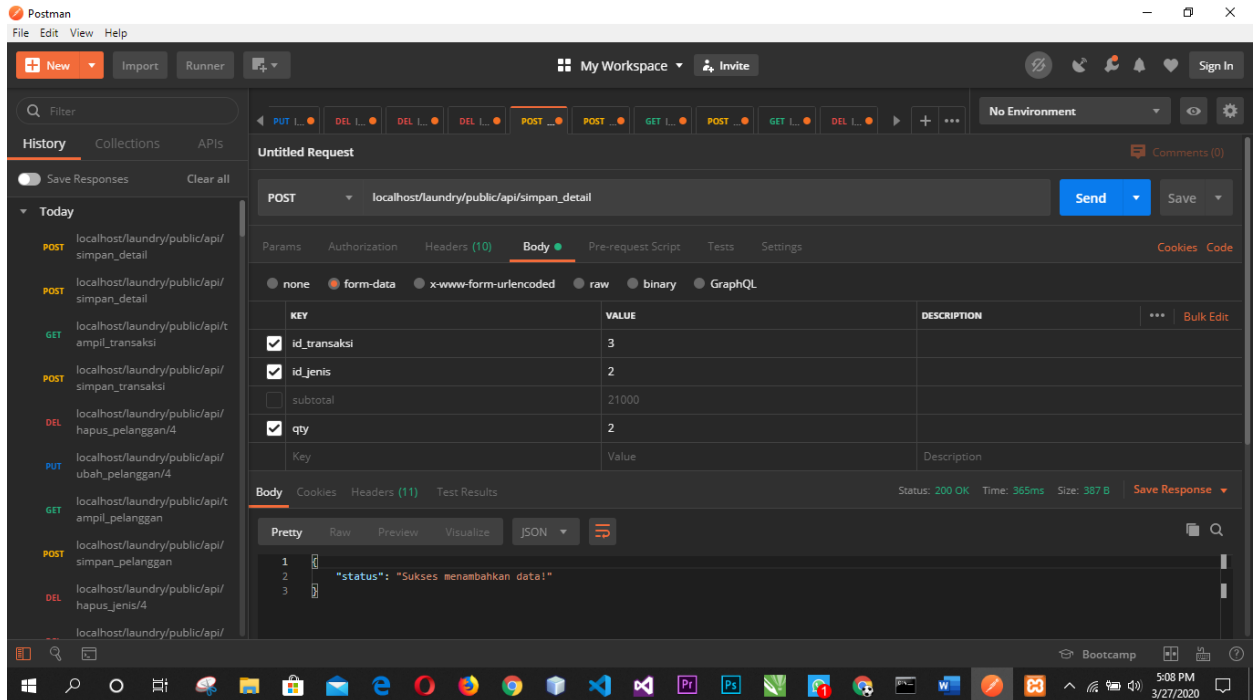
Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "Sukses menambahkan data!"
3 }
```

• Create Detail



• Create dan Read Detail



- Report transaksi berdasarkan tanggal awal sampai tanggal akhir

The screenshot shows the Postman application interface. The left sidebar displays a list of recent requests, all POST requests to the endpoint `localhost/laundry/public/api/tampil_transaksi`. The main panel shows a POST request configuration for the same endpoint. The request body is a JSON object with the following structure:

```
1 {
2   "tgl": "2020-03-27",
3   "nama pelanggan": "putra",
4   "alamat": "malang",
5   "telp": "08123",
6   "tanggal ambil": "2020-03-30",
7   "grand total": {
8     "grandtotal": "8000"
9   },
10  "detail": [
11    {
12      "id": 2,
13      "id_transaksi": 3,
14      "id_jenis": 2,
15      "subtotal": 8000,
16      "qty": 2,
17      "nama_jenis": "Cuci Kering",
18      "harga_per_kilo": 4000
19    }
20  ]
21 }
```

The response status is 200 OK, with a time of 249ms and a size of 617 B. The response body is displayed in the 'Body' tab, showing the same JSON structure as the request body.