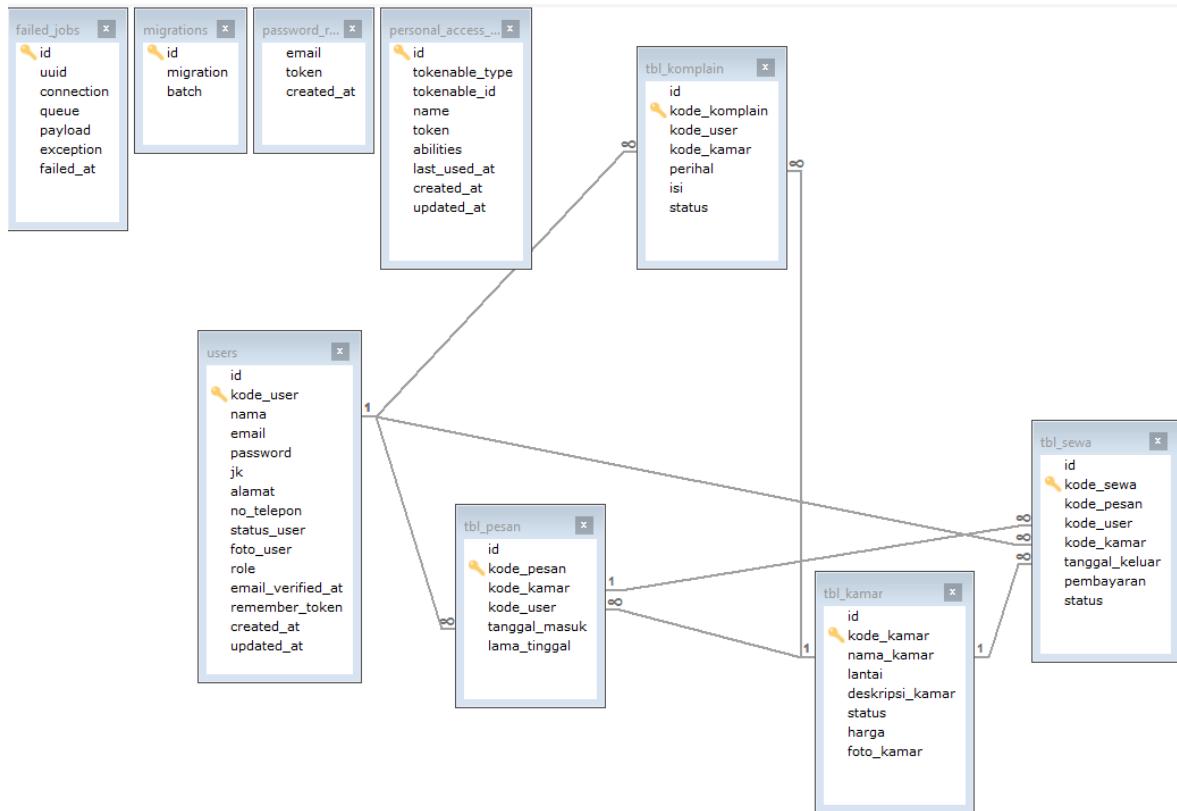


Nama : Dhimas Rizaldy

Npm : 20312030

Bagian : Backend Api

## 1. Database : db\_e-kos



## 2. Tampilan kodingan Models :

- **UserModel.php**

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure under the "BACKEND" folder, including "app", "Models" (containing KamarModel.php, KomplainModel.php, PesanModel.php, SewaModel.php, User.php, and UserModel.php), and various configuration and vendor files.
- Editor:** The "UserModel.php" file is open, displaying PHP code for a database model. The code includes methods for viewing data, getting detail data by parameter, and deleting data by parameter.
- Bottom Status Bar:** Shows the current file path (Backend/UserModel.php), line number (Ln 20), column number (Col 26), and other settings like spaces (4), encoding (UTF-8), and line endings (LF).

## • KamarModel.php

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure under the "BACKEND" folder, including "app", "Models" (containing KamarModel.php, KomplainModel.php, PesanModel.php, SewaModel.php, User.php, and UserModel.php), and various configuration and vendor files.
- Editor:** The "KamarModel.php" file is open, displaying PHP code for a database model. The code includes methods for viewing data, getting detail data by parameter, and deleting data by parameter.
- Bottom Status Bar:** Shows the current file path (Backend/KamarModel.php), line number (Ln 10), column number (Col 25), and other settings like spaces (4), encoding (UTF-8), and line endings (LF).

## • PesanModel.php

```

<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Facades\DB;

class PesanModel extends Model
{
    // Tampil Data pesan
    function viewData()
    {
        $query = DB::table('tbl_pesan')
            ->select(
                'kode_pesan',
                'kode_kamar',
                'kode_user',
                'tanggal_masuk',
                'lama_tinggal'
            )
            ->orderBy('kode_pesan')
            ->get();
        return $query;
    }

    // Tampil Data tbl_pesan Join tb_users Join tbl_kamar
    function viewDataPesanhJoin()
    {
        $query = DB::table('tbl_pesan')
            ->join('users', 'users.kode_user', '=', 'tbl_pesan.kode_user')
            ->join('tbl_kamar', 'tbl_kamar.kode_kamar', '=', 'tbl_pesan.kode_kamar')
            ->select([
                'tbl_pesan.kode_pesan',
                'tbl_pesan.tanggal_masuk',
                'tbl_pesan.tanggal_keluar',
                'tbl_pesan.nama_kamar',
                'tbl_pesan.kode_user',
                'tbl_pesan.nama',
                'tbl_pesan.tanggal_masuk',
                'tbl_pesan.lama_tinggal'
            ])
            ->get();
        return $query;
    }

    // Detail Data pesan
    function detailData($parameter)
    {
        $query = DB::table('tbl_pesan')
            ->select(
                'kode_pesan',
                'kode_kamar',
                'kode_user',
                'tanggal_masuk',
                'lama_tinggal'
            )
            ->where('kode_pesan', '=', $parameter)
            ->orderBy('kode_pesan')
            ->get();
        return $query;
    }
}

```

- **SewaModel.php**

```

<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Facades\DB;

class SewaModel extends Model
{
    // Tampil Data sewa
    function viewData()
    {
        $query = DB::table('tbl_sewa')
            ->select(
                'kode_sewa',
                'kode_pesan',
                'kode_user',
                'kode_kamar',
                'tanggal_keluar',
                'pembayaran',
                'status'
            )
            ->orderBy('kode_sewa')
            ->get();
        return $query;
    }

    // Tampil Data tol_sewa Join tb_users Join tbl_kamar join tbl_pesan
    function viewDataSewahlJoin()
    {
        $query = DB::table('tbl_sewa')
            ->join('users', 'users.kode_user', '=', 'tbl_sewa.kode_user')
            ->join('tbl_pesan', 'tbl_pesan.kode_pesan', '=', 'tbl_sewa.kode_pesan')
            ->join('tbl_kamar', 'tbl_kamar.kode_kamar', '=', 'tbl_sewa.kode_kamar')
            ->select([
                'tbl_sewa.kode_sewa',
                'tbl_sewa.tanggal_keluar',
                'tbl_sewa.pembayaran',
                'tbl_sewa.status',
                'users.nama',
                'tbl_sewa.kode_kamar',
                'tbl_kamar.nama_kamar',
                'tbl_kamar.harga',
                'tbl_pesan.tanggal_masuk',
                'tbl_pesan.lama_tinggal',
                'tbl_sewa.tanggal_keluar'
            ])
            ->get();
        return $query;
    }

    // Tampil View Detail Data tbl_sewa Join tb_users Join tbl_kamar join tbl_pesan
    function viewDetailDataSewahlJoin()
    {
        $query = DB::table('tbl_sewa')
            ->join('users', 'users.kode_user', '=', 'tbl_sewa.kode_user')
            ->join('tbl_pesan', 'tbl_pesan.kode_pesan', '=', 'tbl_sewa.kode_pesan')
            ->join('tbl_kamar', 'tbl_kamar.kode_kamar', '=', 'tbl_sewa.kode_kamar')
            ->select([
                'tbl_sewa.kode_sewa',
                'tbl_sewa.kode_user',
                'users.nama',
                'users.email'
            ]);
        return $query;
    }
}

```

- **KomplainModel.php**

```

<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Facades\DB;
class KomplainModel extends Model
{
    // Tampil Data komplain
    function viewData()
    {
        $query = DB::table("tbl_komplain")
            ->select(
                "kode_komplain",
                "kode_user",
                "kode_kamar",
                "perihal",
                "isi",
                "status"
            )
            ->orderBy("kode_komplain")
            ->get();
        return $query;
    }
    // Tampil Data tbl_komplain join tb_users join tb_kamar
    function viewDataKomplainId($id)
    {
        $query = DB::table("tbl_komplain")
            ->join("tbl_user", "tbl_komplain.kode_user", '=', "tbl_user.kode_user")
            ->join("tbl_kamar", "tbl_komplain.kode_kamar", '=', "tbl_kamar.kode_kamar")
            ->select("tbl_komplain.kode_komplain", "tbl_komplain.kode_user", "users.name", "tbl_komplain.kode_kamar", "tbl_kamar.nama_kamar", "tbl_komplain.perihal", "tbl_komplain.isi", "tbl_komplain.status")
            ->get();
        return $query;
    }
    // Detail Data komplain
    function detailData($parameter)
    {
        $query = DB::table("tbl_komplain")
            ->select(
                "kode_komplain",
                "kode_user",
                "kode_kamar",
                "perihal",
                "isi",
                "status"
            );
    }
}

```

Ln 1, Col 1 Spaces:4 UTF-8 LF PHP Go Live Prettier 9:23 PM 5/31/2023

### 3. Tampilan Kodingan Controller :

- UserController.php

```

<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\UserModel;
class UserController extends Controller
{
    // Function Construct
    function __construct()
    {
        $this->model = new UserModel();
    }
    // Function Untuk View Data User
    function viewUser()
    {
        // Ambil Function viewData dari UserModel
        $data = $this->model->viewData();
        // Tampilan hasil dari "tbl_user"
        return response([
            "DataUser" => $data
        ], http_response_code());
    }
    // Function Untuk Detail Data User
    function detailUser($parameter)
    {
        // Ambil Fungsi detailData dari UserModel
        $data = $this->model->detailData($parameter);
        // Tampilan hasil dari "tbl_user"
        return response([
            "DetailUser" => $data
        ], http_response_code());
    }
    // Function Untuk Delete Data User
    function deletedUser($parameter)
    {
        // Cek data dari "tbl_user" berdasarkan kodeUser
        $data = $this->model->deleteData($parameter);
        // Jika data ditemukan
        if (count($data) != 0) {
            // Buat pesan dan status berhasil
            $status = 1;
            $pesan = "Data Berhasil dihapus";
        }
    }
}

```

Ln 1, Col 1 Spaces:4 UTF-8 LF PHP Go Live Prettier 9:50 PM 5/31/2023

- **KamarController.php**

```

<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\KamarModel;

class KamarController extends Controller
{
    //Function Construct()
    function __construct()
    {
        $this->model = new KamarModel();
    }

    // Function Untuk View Data
    function viewKamar()
    {
        // Ambil Function viewData dari UserModel
        $data = $this->model->viewData();

        // Tampilkan hasil dari "tbl_user"
        return response([
            "DataKamar" => $data
        ], 200);
    }

    // Function Untuk Detail Data User
    function detailKamar($parameter)
    {
        // Ambil Function detailData dari UserModel
        $data = $this->model->detailData($parameter);

        // Tampilkan hasil dari "tbl_user"
        return response([
            "Detail Kamar" => $data
        ], 200);
    }

    // Function Untuk Delete Data User
    function deleteKamar($parameter)
    {
        // cek data dari tbl_user (berdasarkan kode_kamar)
        $data = $this->model->deleteData($parameter);

        // Jika data ditemukan
        if(count($data) != 0) {
            // Data berhasil dihapus
            $data = $this->model->deleteData($parameter);
            // Buat pesan dan status berhasil Penghapusan Data
            $status = 1;
            $pesan = "Data Berhasil dihapus";
        }
    }
}

```

- **PesanController.php**

```

<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\PesanModel;

class PesanController extends Controller
{
    //Function Construct()
    function __construct()
    {
        $this->model = new PesanModel();
    }

    // Function Untuk View Data
    function viewPesan()
    {
        // Ambil Function viewData dari PesanModel
        $data = $this->model->viewData();

        // Tampilkan hasil dari "tbl_pesan"
        return response([
            "DataPesan" => $data
        ], 200);
    }

    // Function untuk view data tbl_pesan join tbl_users join tbl_kamar
    function viewPesanJoin()
    {
        // Ambil Function viewDataPesanJoin dari PesanModel
        $data = $this->model->viewDataPesanJoin();

        // Tampilkan hasil dari "tbl_pesan" join "tbl_users" join "tbl_kamar"
        return response([
            "Data Pesan Join" => $data
        ], 200);
    }

    // Function Untuk Detail Data pesan
    function detailPesan($parameter)
    {
        // Ambil Function detailData dari PesanModel
        $data = $this->model->detailData($parameter);

        // Tampilkan hasil dari "tbl_pesan"
        return response([
            "Detail Pesan" => $data
        ], 200);
    }

    // Function Untuk Delete Data pesan
    function deletePesan($parameter)
    {
    }
}

```

- **SewaController.php**

```

<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\SewaModel;
use App\Models\TblPesanModel;

class SewaController extends Controller
{
    //Function Construct()
    function __construct()
    {
        $this->model = new SewaModel();
    }

    // Function Untuk View Data
    function viewSewa()
    {
        // Ambil Function viewData dari SewaModel
        $data = $this->model->viewData();

        // Tampilkan hasil dari "tbl_sewa"
        return response([
            "Data Sewa" => $data
        ], http_response_code());
    }

    // Function untuk view data tbl_pesan join tbl_users join tbl_kamar join tbl_pesan
    function viewSewaJoin()
    {
        // Ambil Function viewDataJoin dari SewaModel
        $data = $this->model->viewDataJoin();

        // Tampilkan hasil dari "tbl_sewa" join "tbl_users" join "tbl_kamar" join "tbl_pesan"
        return response([
            "Data Sewa Join" => $data
        ], http_response_code());
    }

    // Function untuk view detail data tbl_pesan join tbl_users join tbl_kamar join tbl_pesan
    function viewDetailSewaJoin()
    {
        // Ambil Function viewDataJoin dari SewaModel
        $data = $this->model->viewDetailDataJoin();

        // Tampilkan hasil dari "tbl_sewa" join "tbl_users" join "tbl_kamar" join "tbl_pesan"
        return response([
            "Detail Tampil Data" => $data
        ], http_response_code());
    }

    // Function untuk view detail data tgl_pesan join tbl_users join tbl_kamar join tgl_pesan
    function viewDetailSewa()
    {
        // Ambil Function viewData dari SewaModel
        $data = $this->model->viewDetailData();

        // Tampilkan hasil dari "tbl_sewa"
        return response([
            "Data Sewa" => $data
        ], http_response_code());
    }

    // Function Untuk Detail Data sewa
    function detailSewa($parameter)
    {
        // Function Untuk Delete Data sewa
        function deleteSewa($parameter)
        {
            // Ambil fungsi deleteData dari SewaModel
            $data = $this->model->deleteData($parameter);

            // Tampilkan Hasil dari "tbl_sewa"
            return response([
                "Delete Data" => $data
            ], http_response_code());
        }
    }
}

```

- **KomplainController.php**

```

<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\KomplainModel;
use App\Models\TblKamarModel;

class KomplainController extends Controller
{
    //Function Construct()
    function __construct()
    {
        $this->model = new KomplainModel();
    }

    // Function Untuk View Data
    function viewKomplain()
    {
        // Ambil Function viewData dari KomplainModel
        $data = $this->model->viewData();

        // Tampilkan hasil dari "tbl_komplain"
        return response([
            "Data Komplain" => $data
        ], http_response_code());
    }

    // Function untuk view data tgl_komplain join tbl_users join tgl_kamar
    function viewKomplainJoin()
    {
        // Ambil Function viewDataKomplainJoin dari KomplainModel
        $data = $this->model->viewDataKomplainJoin();

        // Tampilkan hasil dari "tbl_komplain" join "tbl_users" join "tbl_kamar"
        return response([
            "Data Komplain Join" => $data
        ], http_response_code());
    }

    // Function Untuk Detail Data komplain
    function detailKomplain($parameter)
    {
        // Ambil fungsi detailData dari KomplainModel
        $data = $this->model->detailData($parameter);

        // Tampilkan Hasil dari "tbl_komplain"
        return response([
            "Detail Komplain" => $data
        ], http_response_code());
    }

    // Function Untuk Delete Data komplain
    function deleteKomplain($parameter)
    {
        // Ambil fungsi deleteData dari KomplainModel
        $data = $this->model->deleteData($parameter);

        // Tampilkan Hasil dari "tbl_komplain"
        return response([
            "Delete Data" => $data
        ], http_response_code());
    }
}

```

## 4. Tampilan Codingan Route Api

```

Backend > routes > api.php

1 // **** Routes ****
2
3 // Here is where you can register API routes for your application. These
4 // routes are loaded by the RouteServiceProvider within a group which
5 // is assigned the "api" middleware group. Enjoy building your API!
6
7 /*
8 * Route:middleware('auth:sanctum')>get('/user', function (Request $request) {
9 *     return $request->user();
10 * });
11 */
12
13 // **** ROUTE REGISTER ****
14 // **** ROUTE LOGIN ****
15 // **** ROUTE LOGOUT ****
16 // **** ROUTE USER ****
17
18 Route::post('/register', [RegisterController::class, 'register']);
19
20 Route::get('/login', [LoginController::class, 'login']);
21
22 Route::post('/logout', [LogoutController::class, 'logout']);
23
24 // **** ROUTE DETAIL DATA USER ****
25
26 Route::get('/viewUser', [UserController::class, 'viewUser']);
27
28 //**** ROUTE DETAIL DATA USER ****
29 Route::get('/detailUser/{parameter}', [UserController::class, 'detailUser']);
30
31 //**** ROUTE DELETE DATA USER ****
32 Route::delete('/deleteUser/{parameter}', [UserController::class, 'deleteUser']);
33
34 // Route untuk insert data user
35 Route::post('/insertUser', [UserController::class, 'insertUser']);
36
37 // Route untuk update data user
38 Route::put('/updateUser/{parameter}', [UserController::class, 'updateUser']);
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

## 5. Tampilan Seetting .env

```

Backend > .env

1 APP_NAME=Sistem-EKOS
2 APP_ENV=local
3 APP_KEY=base64:tbfSmpAUcvJuC0JPvn5y3d1195k10m+eNvXK8G8Y=
4 APP_DEBUG=true
5 APP_URL=http://127.0.0.1:8000/
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=db_ekos
15 DB_USERNAME=root
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 FILESYSTEM_DRIVER=local
21 QUEUE_DRIVER=sync
22 SESSION_DRIVER=file
23 SESSION_LIFETIME=120
24
25 REDIS_HOST=127.0.0.1
26 REDIS_PASSWORD=null
27 REDIS_PORT=6379
28
29 MAIL_MAILER=set
30 MAIL_HOST=mailing
31 MAIL_PORT=1025
32 MAIL_USERNAME=null
33 MAIL_PASSWORD=null
34 MAIL_ENCRYPTION=null
35 MAIL_FROM_ADDRESS=null
36 MAIL_FROM_NAME="${APP_NAME}"
37
38 AWS_ACCESS_KEY_ID=
39 AWS_SECRET_ACCESS_KEY=
40 AWS_DEFAULT_REGION=us-east-1
41 AWS_BUCKET=
42 AWS_USE_PATH_STYLE_ENDPOINT=false
43
44 PUSHER_APP_ID=
45 PUSHER_APP_KEY=
46 PUSHER_APP_SECRET=
47 PUSHER_APP_CLUSTER=
48
49 PUSHER_APP_CLUSTER=$cluster
50
51 MIGRATION_PUSHER_APP_KEY=${PUSHER_APP_KEY}
52 MIGRATION_PUSHER_APP_CLUSTER=${PUSHER_APP_CLUSTER}
53

```

## 6. Tampilan Testing Backend Api via postman

## a. Test Api Tbl\_user

- Test API tambah user :

POST http://127.0.0.1:8000/api/insertUser

Key	Value	Description
jk	pria	
alamat	Jakarta	
no_telepon	0892772872	
status_user	Pegawai	
foto_user	toni.png	
role	customer	

Status: 200 OK Time: 223 ms Size: 353 B Save as Example

```
1 "status": 1,
2 "pesan": "Data berhasil Disimpan"
```

- Test API Edit data user :

PUT http://127.0.0.1:8000/api/updateUser?kode\_user=10129919&nama=Toni&email=toni1234@gmail.com&password=tori12345678&jk=Pria&alamat=jln.kelapa.no.128 bandar lampung&no\_telepon=0892772882

Key	Value	Description
kode_user	10129919	
nama	Toni	
email	toni1234@gmail.com	
password	toni12345678	
jk	Pria	
alamat	jln.kelapa.no.128 bandar lampung	
no_telepon	0892772882	
status_user	Karyawan	
foto_user	toni123.jpg	
role	Customer	

Status: 200 OK Time: 3.07 s Size: 354 B Save as Example

```
1 "status": 1,
2 "pesan": "Data berhasil di Ubah"
```

## Hasil update data user :

The screenshot shows the Postman application interface. A collection named 'My Workspace' is open, containing various API requests. The current request is a PUT operation titled 'PUT Update\_User' with the URL `http://127.0.0.1:8000/api/detailUser/10129919`. The request body is set to 'x-www-form-urlencoded' and contains the following data:

Key	Value
kode_user	18710001
name	Drimas Rizaldy
email	drimas@gmail.com
password	drimas123

The response status is 200 OK, and the JSON response body is:

```
1  "DetailUser": [
2    {
3      "kode_user": 10129919,
4      "name": "Tonii",
5      "email": "tonii2345@gmail.com",
6      "password": "tonii12345678",
7      "jk": "Pria",
8      "alamat": "Jln.kelapa no.128 bandar lampung",
9      "no_telepon": "089277280292",
10     "status_user": "Karyawan",
11     "role": "Customer",
12     "foto_user": "tonii223.jpg"
13   }
14 ]
15 ]
```

## • Test API Delete data user :

The screenshot shows the Postman application interface. A collection named 'My Workspace' is open, containing various API requests. The current request is a DELETE operation titled 'DELETE\_Delete\_User' with the URL `http://127.0.0.1:8000/api/deleteUser/10129919`. The request body is set to 'x-www-form-urlencoded' and contains the following data:

Key	Value
Key	

The response status is 200 OK, and the JSON response body is:

```
1  "status": 1,
2  "pesan": "Data Berhasil diHapus"
```

## b. Test Api Tbl\_kamar

- Test API tambah kamar :

The screenshot shows the Postman application interface. The left sidebar lists collections and environments, with 'My Workspace' selected. In the main workspace, a collection named 'ApL-E-KOS' is expanded, showing various API endpoints. One endpoint, 'POST Add\_Kamar', is selected and highlighted in grey. The request details panel shows a POST method to the URL `http://127.0.0.1:8000/api/insertKamar`. The 'Body' tab is selected, showing a form-data payload with fields: 'kode\_kamar' (kmr06), 'nama\_kamar' (Kamar C02), 'lantai' (Lantai 2), 'deskripsi\_kamar' (Spesifikasi : -), 'status' (Kosong), 'harga' (750000), and 'foto\_kamar' (KamarC2.png). The response panel shows a status 200 OK with a response body containing JSON: 

```
1 {  
2     "status": 1,  
3     "pesan": "Data berhasil Disimpan"  
4 }
```

- Test API Edit data kamar :

The screenshot shows the Postman application interface. The left sidebar lists collections and environments, with 'My Workspace' selected. In the main workspace, a collection named 'ApL-E-KOS' is expanded, showing various API endpoints. One endpoint, 'PUT Update\_Kamar', is selected and highlighted in grey. The request details panel shows a PUT method to the URL `http://127.0.0.1:8000/api/updateKamar/kmr06?kode_kamar=km06&nama_kamar=Kamar C06&lantai=lantai 3&deskripsi_kamar=Kamar Lengkap : sudah tersedia Kasur,Meja Belajar Kursi dan Lemari&status=Penuh&harga=750000`. The 'Body' tab is selected, showing a form-data payload with fields: 'kode\_kamar' (kmr06), 'nama\_kamar' (Kamar C06), 'lantai' (Lantai 3), 'deskripsi\_kamar' (Kamar Lengkap : sudah tersedia Kasur,Meja Belajar Kursi dan Lemari), 'status' (Penuh), 'harga' (750000), and 'foto\_kamar' (Kamar C3.png). The response panel shows a status 200 OK with a response body containing JSON: 

```
1 {  
2     "status": 1,  
3     "pesan": "Data berhasil di Ubah"  
4 }
```

**Hasil update data kamar :**

The screenshot shows the Postman interface with a successful API call. The URL is `http://127.0.0.1:8000/api/detailKamar/km06`. The response body is:

```

1  [
2     "Detail_Kamar": [
3         {
4             "name_kamar": "Kamar C06",
5             "lantai": "Lantai 3",
6             "deskripsi_kamar": "Kamar Lengkap : sudah tersedia Kasur,Meja Belajar Kursi dan Lemari",
7             "status": "Penuh",
8             "harga": "750000",
9             "foto_kamar": "Kamar C3.png"
10        }
11    ]
12 ]

```

## • Test API Delete data kamar :

The screenshot shows the Postman interface with a successful API call. The URL is `http://127.0.0.1:8000/api/deleteKamar/km06`. The response body is:

```

1  [
2     "status": 1,
3     "pesan": "Data Berhasil diHapus"
4 ]

```

## c. Test Api Tbl\_pesan

- Test API tambah pesan :

The screenshot shows the Postman interface with a collection named 'My Workspace' containing various API endpoints. The current request is a POST to `http://127.0.0.1:8000/api/insertPesan`. The body is set to 'x-www-form-urlencoded' and contains the following parameters:

Key	Value	Description
kode_pesan	KPSN03	
kode_kamar	kmr03	
kode_user	23432424	
tanggal_awal	2023-06-01	
lama_tinggal	2 Bulan	

The response status is 200 OK, time 488 ms, size 353 B.

- Test API Edit data pesan :

The screenshot shows the Postman interface with a collection named 'My Workspace' containing various API endpoints. The current request is a PUT to `http://127.0.0.1:8000/api/updatePesan/KPSN03&kode_pesan=KPSN03&kode_kamar=kmr04&kode_user=23432424&tanggal_masuk=2023-06-01&lama_tinggal=4 Bulan`. The body is set to 'x-www-form-urlencoded' and contains the following parameters:

Key	Value	Description
kode_pesan	KPSN03	
kode_kamar	kmr04	
kode_user	23432424	
tanggal_masuk	2023-06-01	
lama_tinggal	4 Bulan	

The response status is 200 OK, time 404 ms, size 354 B.

**Hasil update data pesan :**

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists various API requests. In the center, a specific request is selected: 'GET View\_Pesan' with the URL `http://127.0.0.1:8000/api/detailPesan/KPSN03`. The 'Params' tab is active, showing three query parameters: 'name', 'email', and 'password'. The 'Body' tab shows the response in JSON format:

```
1  [
2    "Detail Pesan": [
3      {
4        "kode_pesan": "KPSN03",
5        "kode_kamar": "kps04",
6        "kode_user": "23432424",
7        "tanggalsusuk": "2023-06-01",
8        "lama_tinggal": "4 Bulan"
9      }
10    ]
11  ]
```

The status bar at the bottom indicates a 200 OK response with a time of 403 ms and a size of 446 B.

## • Test API Delete data pesan :

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists various API requests. In the center, a specific request is selected: 'DELETE Pesan' with the URL `http://127.0.0.1:8000/api/deletePesan/KPSN03`. The 'Params' tab is active, showing one parameter: 'Key'. The 'Body' tab shows the response in JSON format:

```
1  [
2    "status": 1,
3    "pesan": "Data Berhasil diHapus"
4  ]
```

The status bar at the bottom indicates a 200 OK response with a time of 327 ms and a size of 352 B.

## d. Test Api Tbl\_sewa

### • Test API tambah sewa :

The screenshot shows the Postman interface with a collection named 'ApI\_E-KOS'. A POST request is made to `http://127.0.0.1:8000/api/insertSewa`. The request body contains the following data:

Key	Value
kode_sewa	KDSW03
kode_pesan	KPSN03
kode_user	29897728
kode_kamar	kmr02
tanggal_keluar	2023-8-21
pembayaran	750000
status	Dicuci

The response status is 200 OK with a time of 454 ms and a size of 353 B.

### • Test API Edit data sewa :

The screenshot shows the Postman interface with a collection named 'ApI\_E-KOS'. A PUT request is made to `http://127.0.0.1:8000/api/updateSewa/KDSW03?kode_sewa=KDSW03&kode_pesan=KPSN03&kode_user=29897728&kode_kamar=kmr02&tanggal_keluar=2023-08-21&pembayaran=750000&status=Sudah Bayar`. The request body contains the following data:

Key	Value
status	1
pesan	"Data berhasil di Ubah"

The response status is 200 OK with a time of 12.12 s and a size of 354 B.

## Hasil update data sewa :

The screenshot shows the Postman interface with a collection named "My Workspace". A specific request titled "Api\_E-KOS / Show\_Sewa" is selected. The method is set to "GET" and the URL is "http://127.0.0.1:8000/api/detailSewa/KDSW03". The "Params" tab is active, showing three query parameters: "nama", "email", and "password". The "Body" tab shows a JSON response with the following structure:

```
1  "Detail Sewa": [
2    {
3      "kode_sewa": "KDSW03",
4      "kode_pesan": "KPSN03",
5      "kode_user": "29897728",
6      "kode_kamar": "kmr02",
7      "tanggal_keluar": "2023-08-21",
8      "pembayaran": "750000",
9      "status": "Sudah Bayar"
10    }
11 }
```

The status bar at the bottom indicates a 200 OK response with a time of 479 ms and a size of 487 B.

## • Test API Delete data sewa :

The screenshot shows the Postman interface with the same "My Workspace" collection. A request titled "Api\_E-KOS / Delete\_Sewa" is selected, using the "DELETE" method and the URL "http://127.0.0.1:8000/api/deleteSewa/KDSW03". The "Params" tab is active, showing a single query parameter "Key". The "Body" tab shows a JSON response with the following structure:

```
1  "status": 1,
2  "pesan": "Data Berhasil diHapus"
```

The status bar at the bottom indicates a 200 OK response with a time of 382 ms and a size of 352 B.

## e. Test Api Tbl\_komplain

- Test API tambah komplain :

The screenshot shows the Postman interface with a collection named "Api\_E-KOS". A POST request is being made to "http://127.0.0.1:8000/api/insertKomplain". The request body is set to "form-data" and contains the following fields:

Key	Value
kode_komplain	KDKMP02
kode_user	10102939
kode_kamar	kmr01
perihal	keran air rusak
isi	pak tolong keran air saya rusak, tolong diperbaiki pak
status	Belum

The response status is 200 OK, time 500 ms, size 353 B.

- Test API Edit data komplain :

The screenshot shows the Postman interface with a collection named "Api\_E-KOS". A PUT request is being made to "http://127.0.0.1:8000/api/updateKomplain/KDKMP02?kode\_komplain=KDKMP02&kode\_user=10102939&kode\_kamar=kmr01&perihal=Genteng Bocor&isi=Pak ini kamar saya genteng nya bocor, mohon diperbaiki...&status=Belum". The request body is set to "form-data" and contains the same fields as the previous request, with the "perihal" field updated to "Genteng Bocor, Air Mati, Token Listrik Habis". The response status is 200 OK, time 397 ms, size 354 B.

Hasil update data komplain :

My Workspace

GET View\_All\_Pesan JOIN users J...  
GET View\_All\_Sewa JOIN users J...  
GET View\_All\_Detail Sewa JOIN u...  
GET View\_All\_Sewa  
GET View\_All\_Komplain  
GET View\_All\_Kamar  
GET View\_All\_Pesan  
PUT Update\_User  
PUT Update\_Sewa  
PUT Update\_Pesan  
PUT Update\_Kamar  
DEL Delete\_User  
DEL Delete\_Kamar  
DEL Delete\_Pesan  
DEL Delete\_Sewa  
DEL Delete\_Komplain  
GET Show\_User  
GET Show\_Pesan  
GET Show\_Sewa  
GET Show\_SewaJoin  
GET Show\_Komplain  
GET Show\_Kamar  
New Collection  
RESTful API basics: CRUD, test & va...

Api\_E-KOS / Show\_Komplain

GET http://127.0.0.1:8000/api/detailKomplain/KDKMP02

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description
nama		
email		
password		

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON

```
1   "Detail Komplain": [
2     {
3       "kode_komplain": "KDKMP02",
4       "kode_user": "10102939",
5       "kode_kamar": "kme01",
6       "perihal": "Genteng Bocor, Air Mati, Token Listrik Habis",
7       "isi": "pak ini kamar saya genteng nya bocor, mohon diperbaiki ya pak",
8       "status": "Belum"
9     }
10   ]
11 ]
```

Status: 200 OK Time: 432 ms Size: 543 B Save as Example

Runner Capture requests Cookies Trash 10:05 AM 6/2/2023

- Test API Delete data komplain :

My Workspace

GET View\_All\_Pesan JOIN users J...  
GET View\_All\_Sewa JOIN users J...  
GET View\_All\_Detail Sewa JOIN u...  
GET View\_All\_Sewa  
GET View\_All\_Komplain  
GET View\_All\_Kamar  
GET View\_All\_Pesan  
PUT Update\_User  
PUT Update\_Sewa  
PUT Update\_Pesan  
PUT Update\_Kamar  
DEL Delete\_User  
DEL Delete\_Kamar  
DEL Delete\_Pesan  
DEL Delete\_Sewa  
DEL Delete\_Komplain  
GET Show\_User  
GET Show\_Pesan  
GET Show\_Sewa  
GET Show\_SewaJoin  
GET Show\_Komplain  
GET Show\_Kamar  
New Collection  
RESTful API basics: CRUD, test & va...

Api\_E-KOS / Delete\_Komplain

DELETE http://127.0.0.1:8000/api/deleteKomplain/KDKMP02

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description
Key		

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON

```
1   "status": 1,
2   "pesan": "Data berhasil dihapus"
3
4 ]
```

Status: 200 OK Time: 375 ms Size: 352 B Save as Example

Runner Capture requests Cookies Trash 10:06 AM 6/2/2023