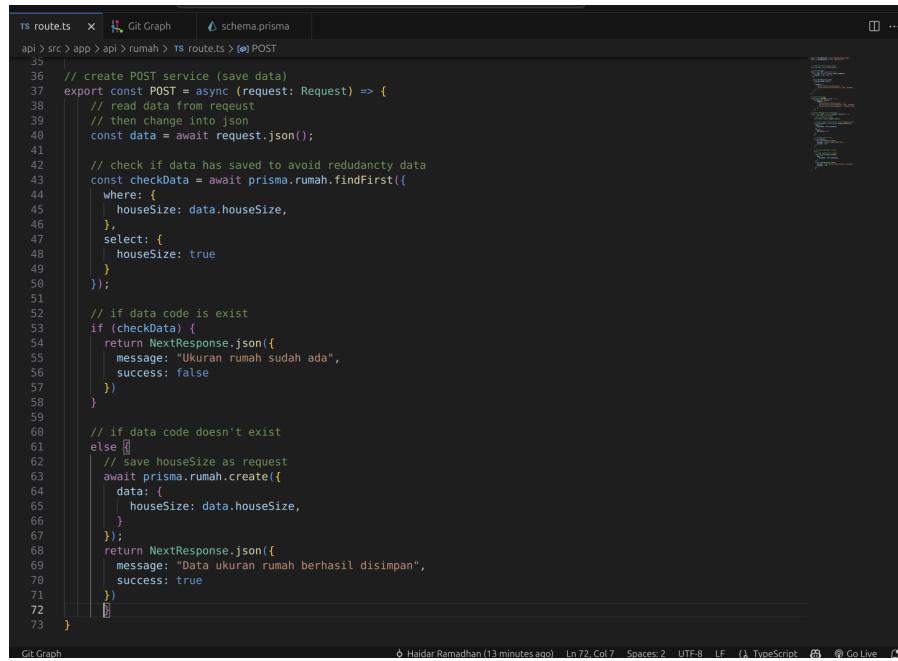
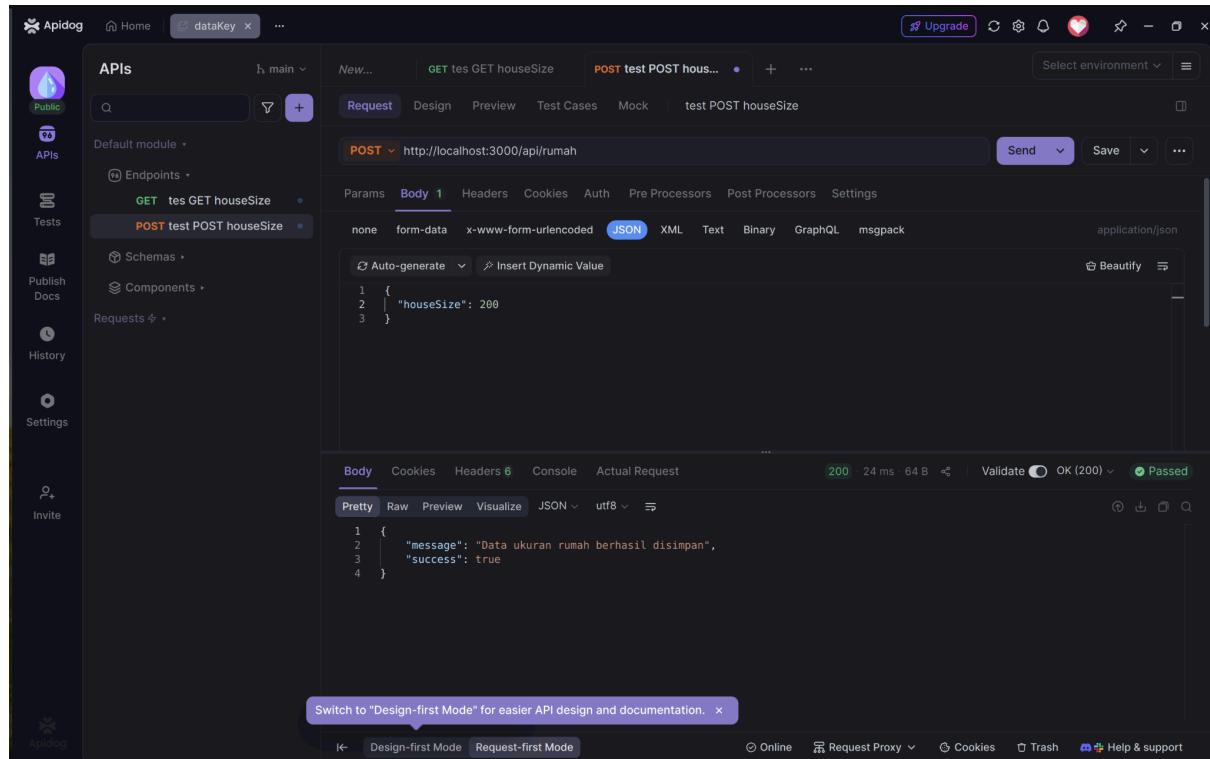


## 1. Tambahkan logika untuk **POST**, dalam hal ini houseSize



```
ts route.ts x Git Graph schema.prisma
api > src > app > api > rumah > ts route.ts > (e) POST
35
36 // create POST service (save data)
37 export const POST = async (request: Request) => {
38   // read data from request
39   // then change into json
40   const data = await request.json();
41
42   // check if data has saved to avoid redundancy data
43   const checkData = await prisma.rumah.findFirst({
44     where: {
45       houseSize: data.houseSize,
46     },
47     select: {
48       houseSize: true
49     }
50   });
51
52   // if data code is exist
53   if (checkData) {
54     return NextResponse.json({
55       message: "Ukuran rumah sudah ada",
56       success: false
57     })
58   }
59
60   // if data code doesn't exist
61   else {
62     // save houseSize as request
63     await prisma.rumah.create({
64       data: {
65         houseSize: data.houseSize,
66       }
67     });
68     return NextResponse.json({
69       message: "Data ukuran rumah berhasil disimpan",
70       success: true
71     })
72   }
73 }
```

## 2. Setelahnya dilakukan tes API melalui **Apidog**



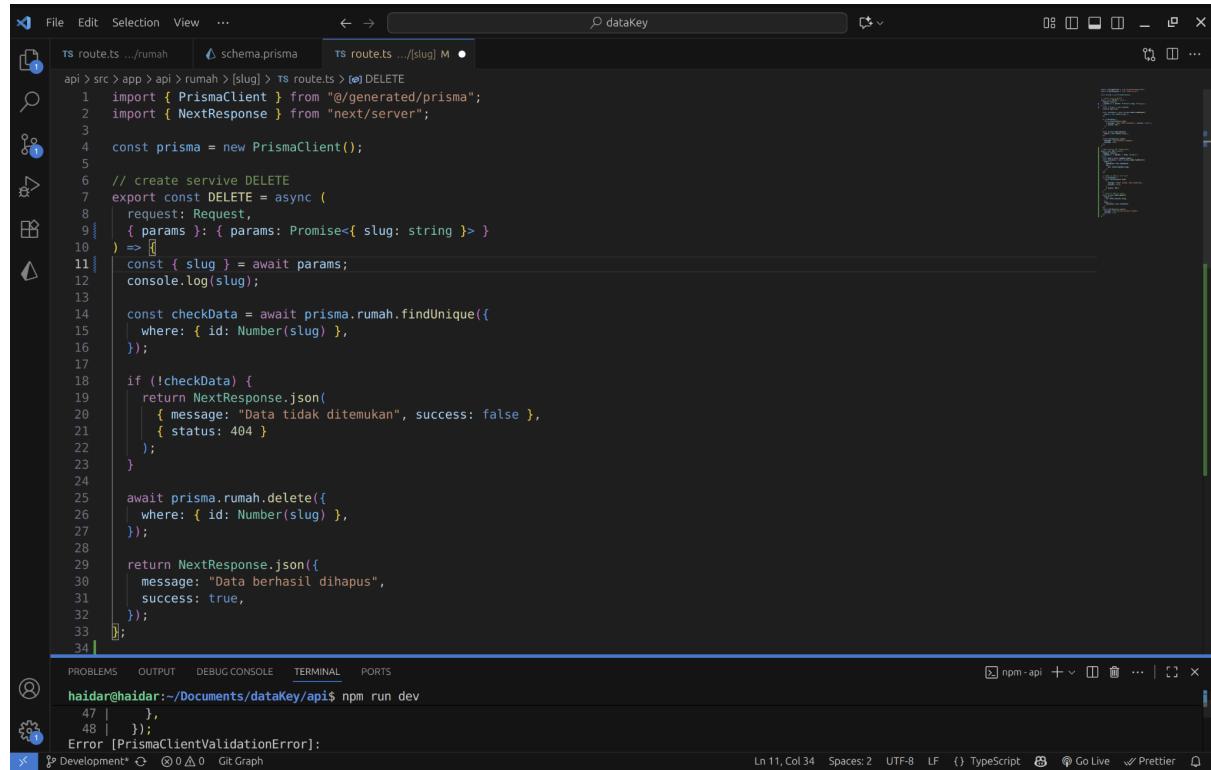
The screenshot shows the Apidog interface with the following details:

- APIs** sidebar: Public, APIs, Tests, Publish Docs, History, Settings, Invite.
- Endpoints**:
  - GET tes GET houseSize
  - POST test POST houseSize (selected)
- Request** tab: POST test POST houseSize, URL: http://localhost:3000/api/rumah
- Body** tab: Body type is JSON, value: { "houseSize": 200 }
- Response** tab: Status 200, Response body:

```
1 {  
2   "message": "Data ukuran rumah berhasil disimpan",  
3   "success": true  
4 }
```

Sementara kita hanya mengirim request berupa houseSize (Ukuran rumah), karena harga akan diproses oleh bahasa pemrograman python melalui algoritma linear regression untuk memprediksi harga rumah berdasarkan data.

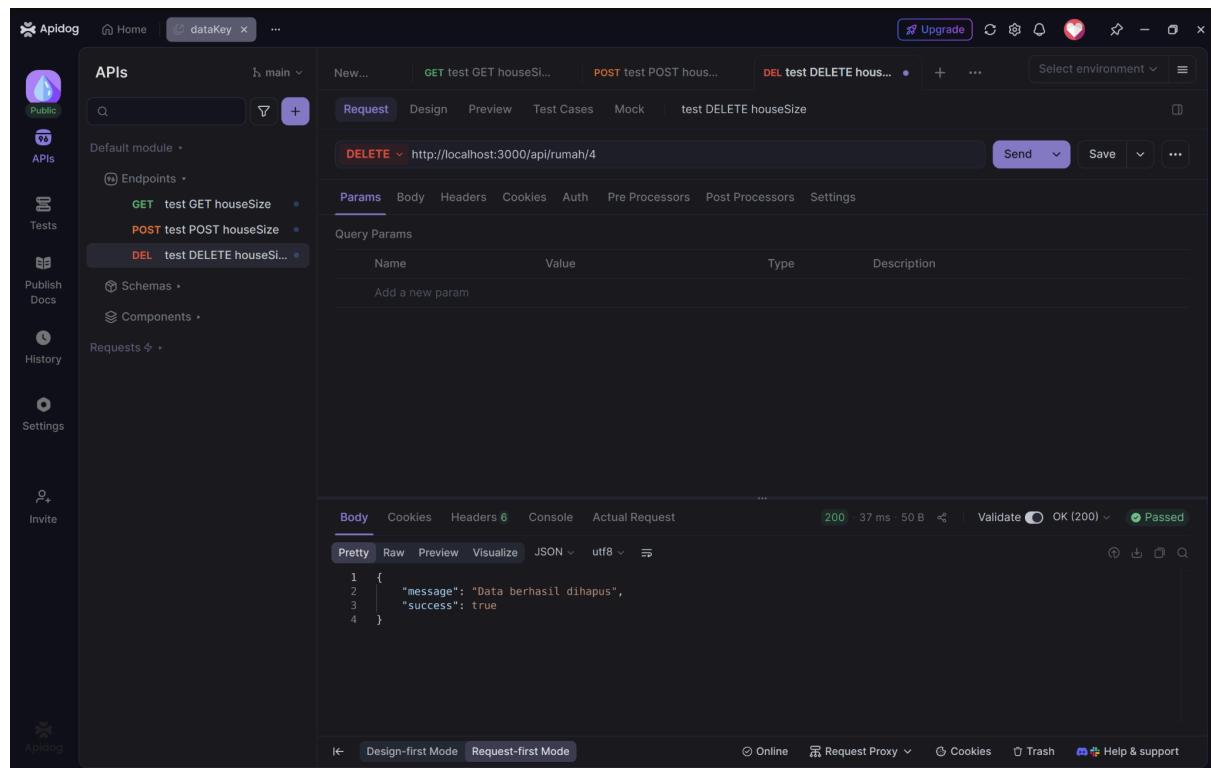
### 3. Buat logika **DELETE** di dalam `src/app/api/rumah/[slug]/`



The screenshot shows the `route.ts` file in VS Code. The code defines a `DELETE` endpoint for a house by slug. It first checks if the house exists. If it does, it is deleted from the database. Finally, a success message is returned.

```
api > src > app > api > rumah > [slug] > ts route.ts > [DELETE]
1 import { PrismaClient } from "@generated/prisma";
2 import { NextResponse } from "next/server";
3
4 const prisma = new PrismaClient();
5
6 // create service DELETE
7 export const DELETE = async (
8   request: Request,
9   { params }: { params: Promise<{ slug: string }> }
10 ) => {
11   const { slug } = await params;
12   console.log(slug);
13
14   const checkData = await prisma.rumah.findUnique({
15     where: { id: Number(slug) },
16   });
17
18   if (!checkData) {
19     return NextResponse.json(
20       { message: "Data tidak ditemukan", success: false },
21       { status: 404 }
22     );
23   }
24
25   await prisma.rumah.delete({
26     where: { id: Number(slug) },
27   });
28
29   return NextResponse.json({
30     message: "Data berhasil dihapus",
31     success: true,
32   });
33 };
34 
```

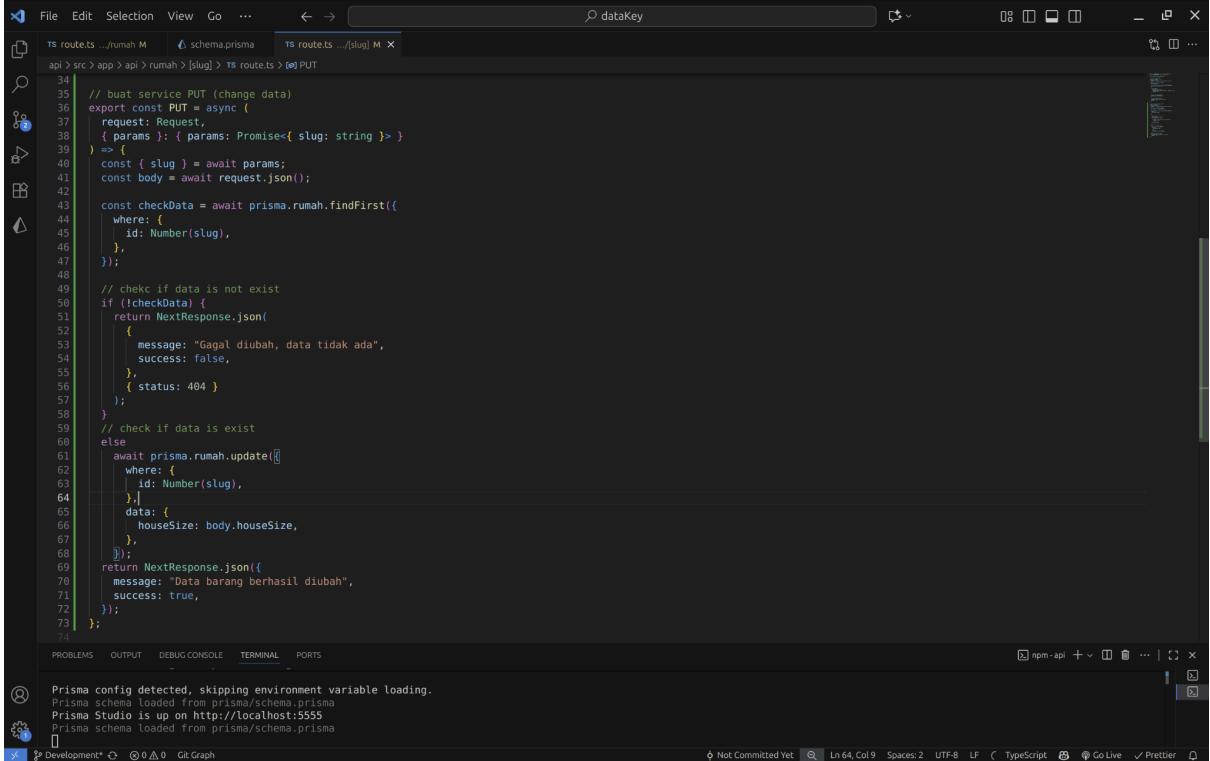
test it!



The screenshot shows the Apidog API testing tool. A `DELETE` request is being made to `http://localhost:3000/api/rumah/4`. The response body is a JSON object indicating success: `{"message": "Data berhasil dihapus", "success": true}`.

Name	Value	Type	Description
message	Data berhasil dihapus	String	
success	true	Boolean	

#### 4. Buat logika untuk **PUT** (Change data)



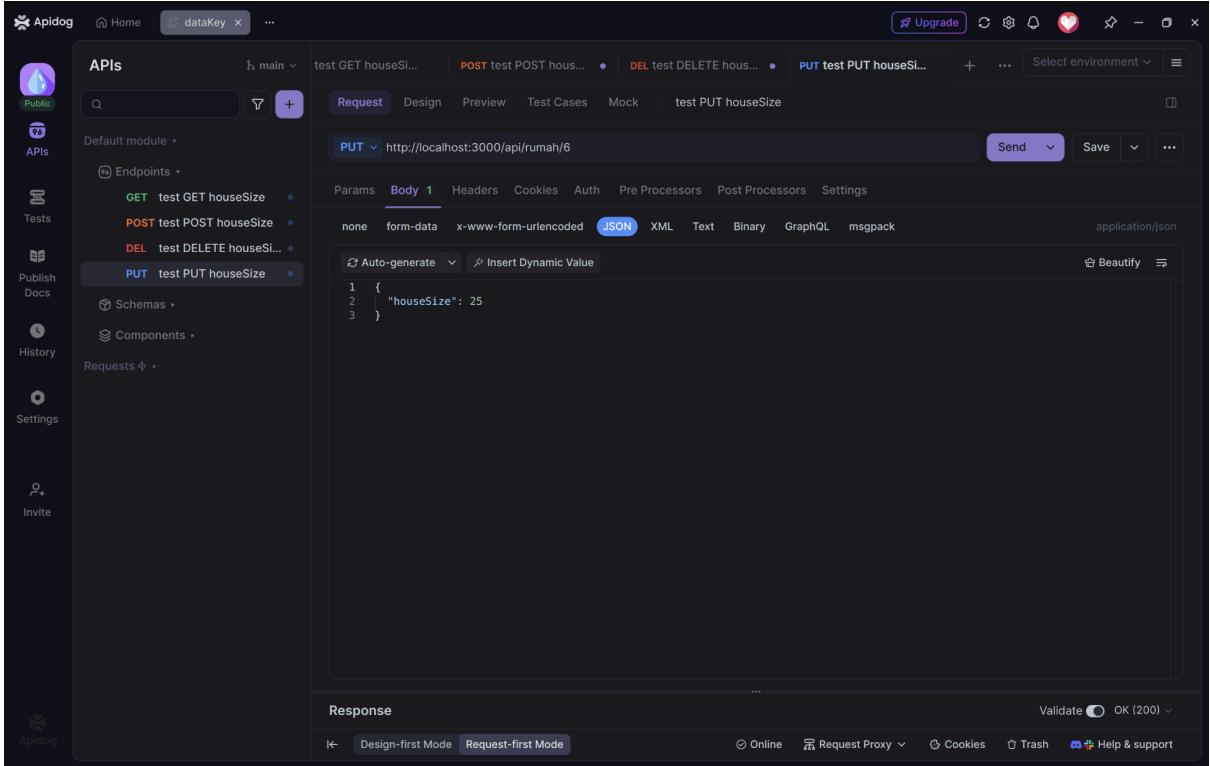
```
34 // buat service PUT (change data)
35 export const PUT = async (
36   request: Request,
37   { params }: { params: Promise<{ slug: string }> }
38 ) => {
39   const { slug } = await params;
40   const body = await request.json();
41
42   const checkData = await prisma.rumah.findFirst({
43     where: {
44       id: Number(slug),
45     },
46   });
47
48   // chekc if data is not exist
49   if (!checkData) {
50     return NextResponse.json(
51       {
52         message: "Gagal diubah, data tidak ada",
53         success: false,
54       },
55       { status: 404 }
56     );
57   }
58   // check if data is exist
59   else {
60     await prisma.rumah.update({
61       where: {
62         id: Number(slug),
63       },
64       data: {
65         houseSize: body.houseSize,
66       },
67     });
68   }
69   return NextResponse.json({
70     message: "Data barang berhasil diubah",
71     success: true,
72   });
73 };
74
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Prisma config detected, skipping environment variable loading.  
Prisma schema loaded from prisma/schema.prisma  
Prisma Studio is up on http://localhost:5555  
Prisma schema loaded from prisma/schema.prisma

Not Committed Yet | Ln 64, Col 9 | Spaces: 2 | UTF-8 | LF | TypeScript | Go Live | Prettier

test it...!!!



The screenshot shows the Apidog interface with the following details:

- APIs** section: A list of endpoints including **GET test GET houseSize**, **POST test POST houseSize**, **DEL test DELETE houseSize**, and **PUT test PUT houseSize**.
- Request** tab selected.
- URL**: `http://localhost:3000/api/rumah/6`
- Body** tab selected, showing `application/json` content type.
- Body Content**:

```
1  {
2   "houseSize": 25
3 }
```
- Response** tab: Shows a successful response with status `OK (200)`.