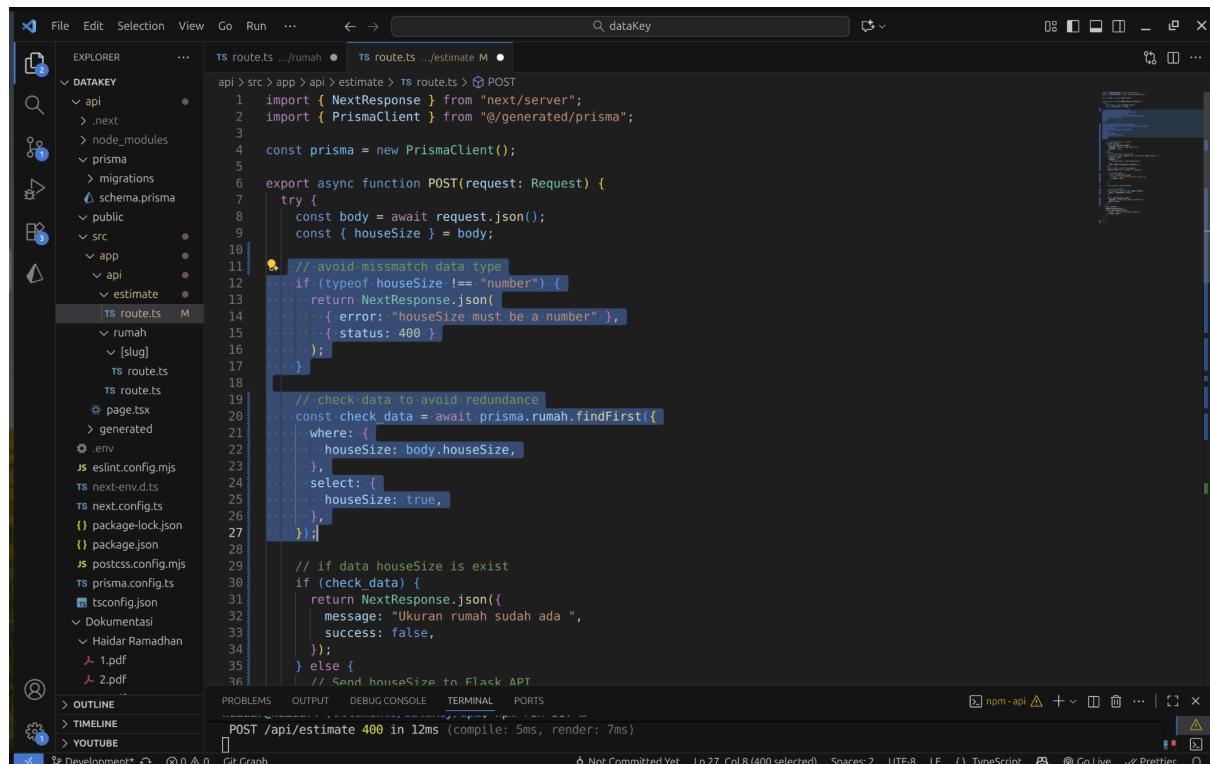


- Penyempurnaan/pengkondisian `route.ts` di dalam `app/api/estimate`, untuk menghindari `type mismatch`, dan redundancy.



```
import { NextResponse } from "next/server";
import { PrismaClient } from "@generated/prisma";
const prisma = new PrismaClient();

export async function POST(request: Request) {
  try {
    const body = await request.json();
    const { houseSize } = body;

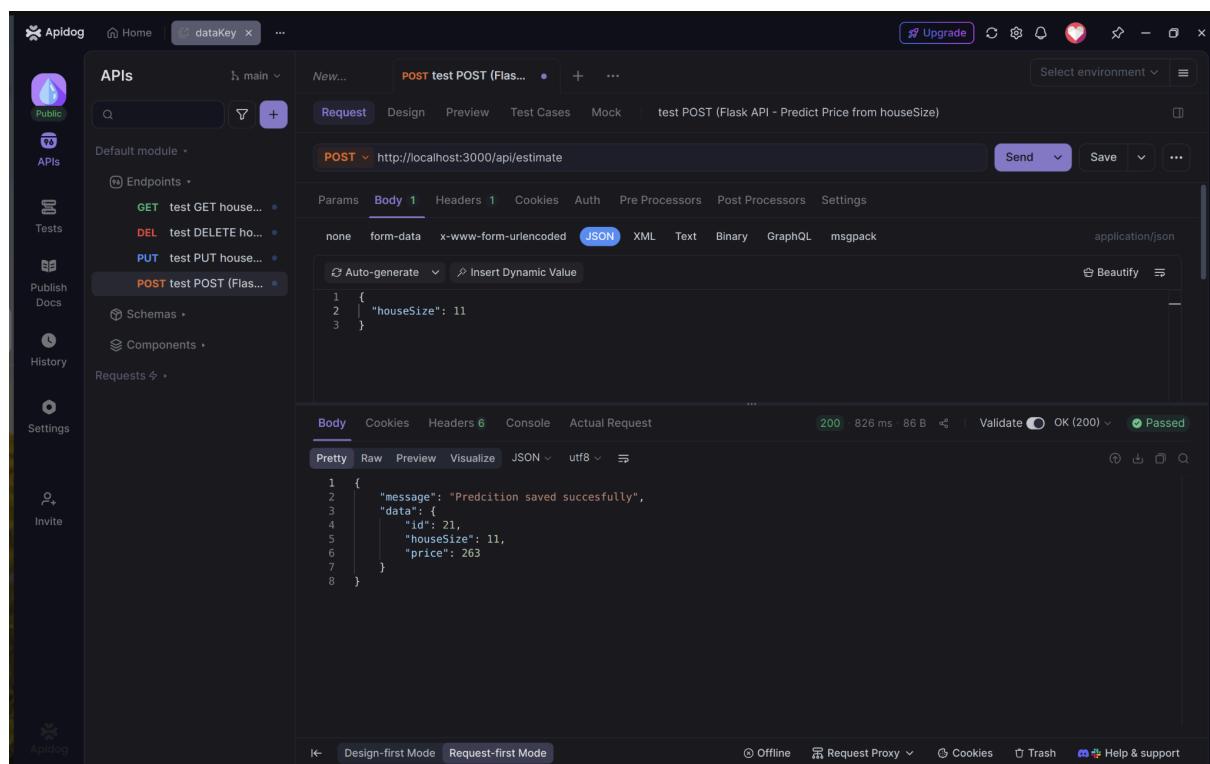
    // avoid mismatch data type
    if (typeof houseSize !== "number") {
      return NextResponse.json({
        error: "houseSize must be a number",
        status: 400
      });
    }

    // check data to avoid redundancy
    const checkData = await prisma.rumah.findFirst({
      where: {
        houseSize: body.houseSize,
      },
      select: {
        houseSize: true,
      },
    });

    // if data houseSize is exist
    if (checkData) {
      return NextResponse.json({
        message: "Ukuran rumah sudah ada",
        success: false,
      });
    } else {
      // Send houseSize to Flask API
    }
  } catch (error) {
    console.error(error);
    return NextResponse.json({
      error: "Internal Server Error",
      status: 500
    });
  }
}
```

test it...!!!

### Pengujian untuk POST (estimate/route.ts)



The screenshot shows the Apidog API testing interface. A POST request is being made to `http://localhost:3000/api/estimate`. The request body is set to `JSON` with the value:

```
1 {
2   "houseSize": 11
3 }
```

The response status is `200` and the response body is:

```
1 {
2   "message": "Prediction saved successfully",
3   "data": {
4     "id": 21,
5     "houseSize": 11,
6     "price": 263
7   }
8 }
```

jika houseSize sudah ada, maka akan mengembalikan nilai sebagai berikut.

The screenshot shows the Apidog API testing tool. On the left sidebar, there are sections for APIs, Tests, Publish, History, and Settings. The main area displays a POST request to 'http://localhost:3000/api/estimate'. The 'Body' tab shows a JSON payload with 'houseSize': 11. The response status is 200 OK (200 ms). The response body is: { "message": "Ukuran rumah sudah ada", "success": false }. The validation results show one error: 'Response data differs from endpoint spec'.

Jika tipe data tidak sesuai maka akan mengembalikan nilai sebagai berikut

The screenshot shows the Apidog API testing tool. The setup is identical to the previous one, but the JSON payload now contains 'houseSize': "Sebelas". The response status is 400 (14 ms). The response body is: { "error": "houseSize must be a number" }. The validation results show two errors: 'HTTP Status Code should be 200' and 'Response data differs from endpoint spec'.

