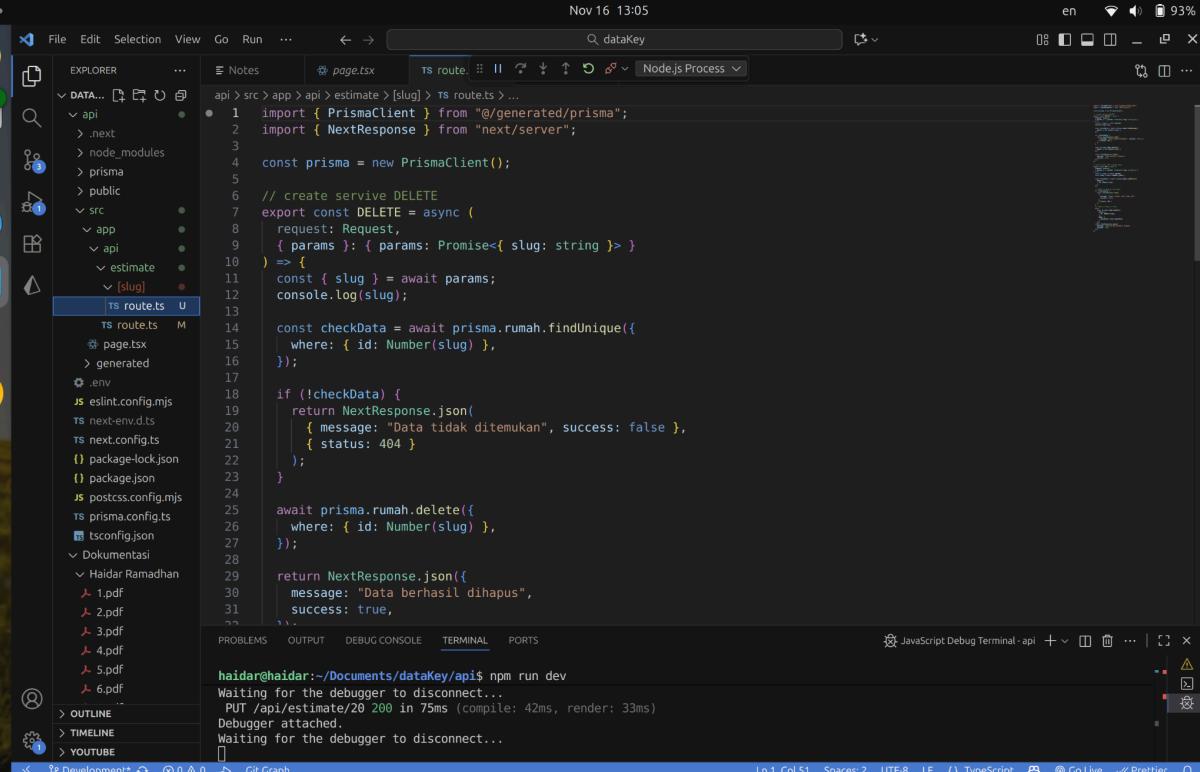


1. Pemindahan logika DELETE dan PUT to exact file, di dalam estimate/[slug]/route.ts.

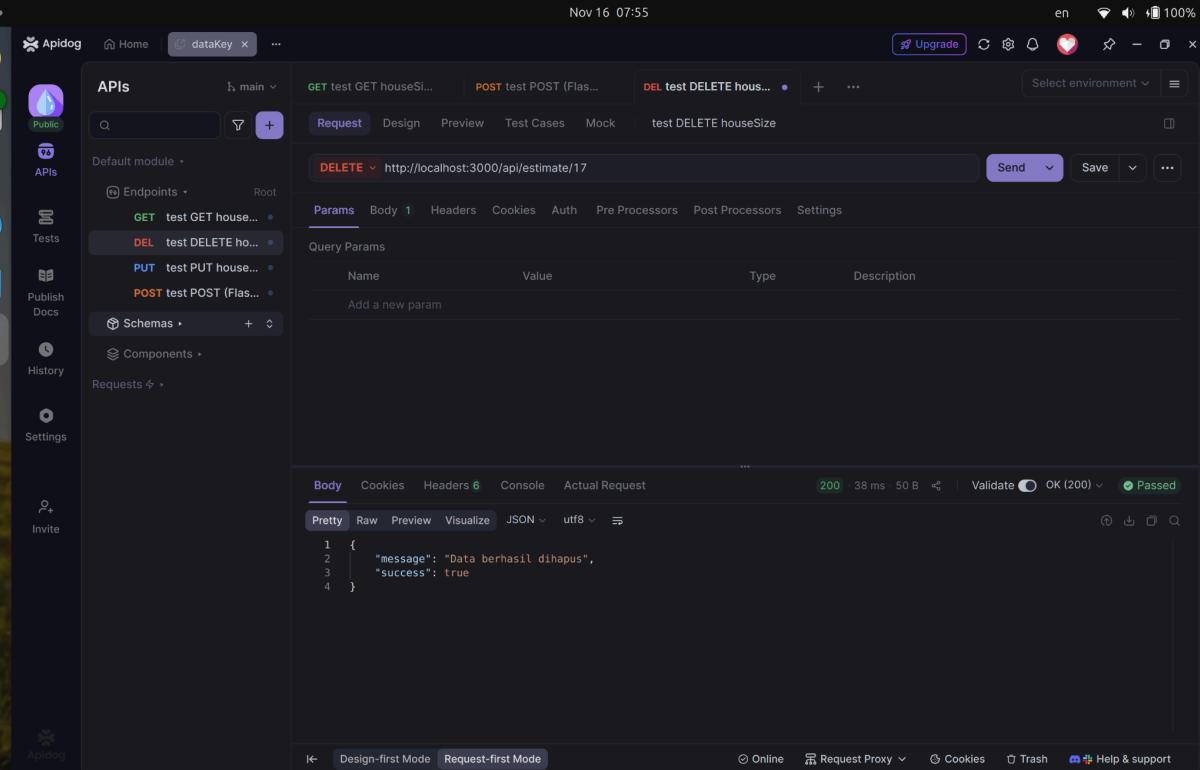


The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure with a focus on the `api/estimate/[slug]/route.ts` file.
- Editor:** Displays the code for a `DELETE` endpoint. The code imports PrismaClient and NextResponse, defines a function to check if a house exists by ID, and then performs a delete operation on the Prisma database. It returns a response indicating success or failure.
- Terminal:** Shows the command `npm run dev` being run, and the output indicates the debugger is attached.
- Status Bar:** Shows the current file is `route.ts`, the line count is 31, and the file size is 200 bytes.

test it...!!!

Pengujian untuk DELETE (estimate/route.ts)



The screenshot shows the Apidog API testing tool interface with the following details:

- Sidebar:** Shows sections for Public, APIs, Tests, Publish, Docs, History, Settings, and Invite.
- APIs Tab:** Shows a list of endpoints: `GET test GET houseSize`, `POST test POST (Flas...`, and `DEL test DELETE hous...`.
- Request Tab:** A `DELETE` request is selected, pointing to `http://localhost:3000/api/estimate/17`. The `Send` button is visible.
- Params Tab:** Shows parameters for the request, including `Body` (with value 1), `Headers`, `Cookies`, `Auth`, `Pre Processors`, `Post Processors`, and `Settings`.
- Body Tab:** Shows the response body in JSON format:

```
1 {
2   "message": "Data berhasil dihapus",
3   "success": true
4 }
```
- Bottom Navigation:** Includes links for Design-first Mode, Request-first Mode, Online, Request Proxy, Cookies, Trash, and Help & support.

Pengujian untuk PUT (estimate/route.ts)

The screenshot shows the Apidog API testing interface. On the left sidebar, under the 'APIs' section, there is a list of endpoints: GET test GET house..., DEL test DELETE house..., PUT test PUT house..., and POST test POST (Flas...). The 'PUT' endpoint is selected. In the main panel, the URL is set to `http://localhost:3000/api/estimate/20`. The 'Body' tab is active, showing a JSON payload:

```
1 {
2   "houseSize": 57
3 }
```

Below the body, the response is displayed in a table:

Body	Cookies	Headers	Console	Actual Request
1 { 2 "message": "Data barang berhasil diubah", 3 "success": true 4 }				200 95 ms 56 B Validate OK (200) Passed

The response body is also shown in a code editor-like view:

```
1 {
2   "message": "Data barang berhasil diubah",
3   "success": true
4 }
```