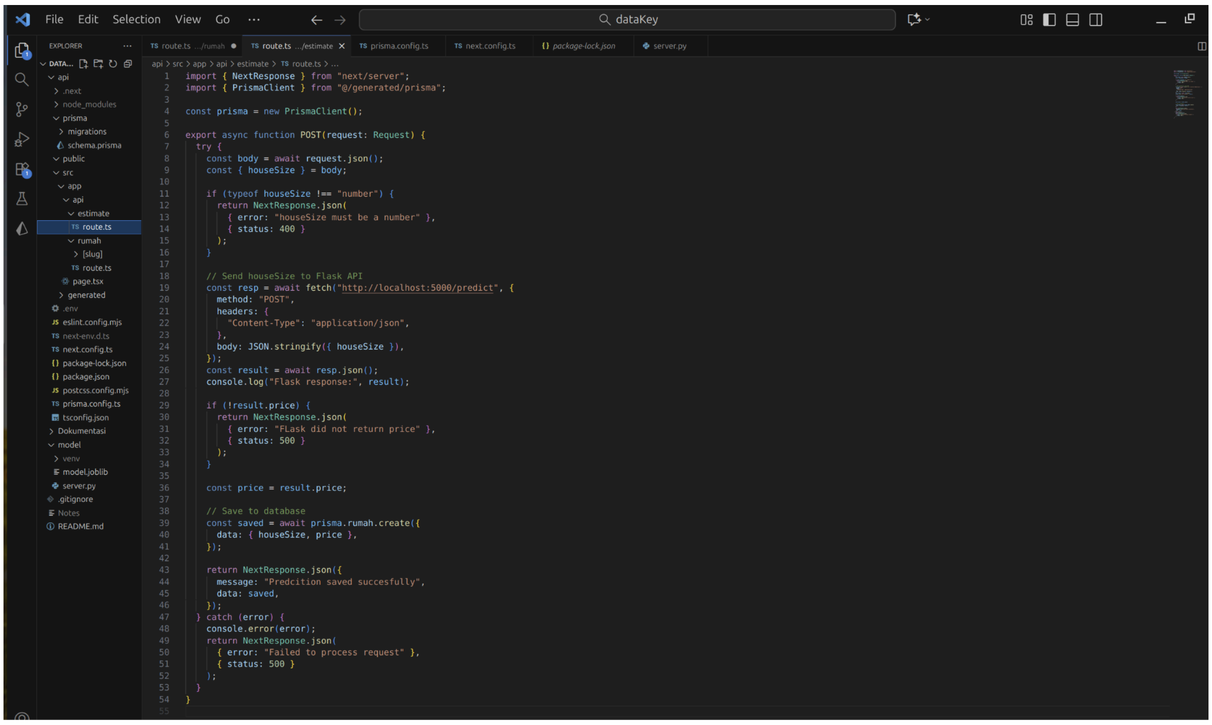
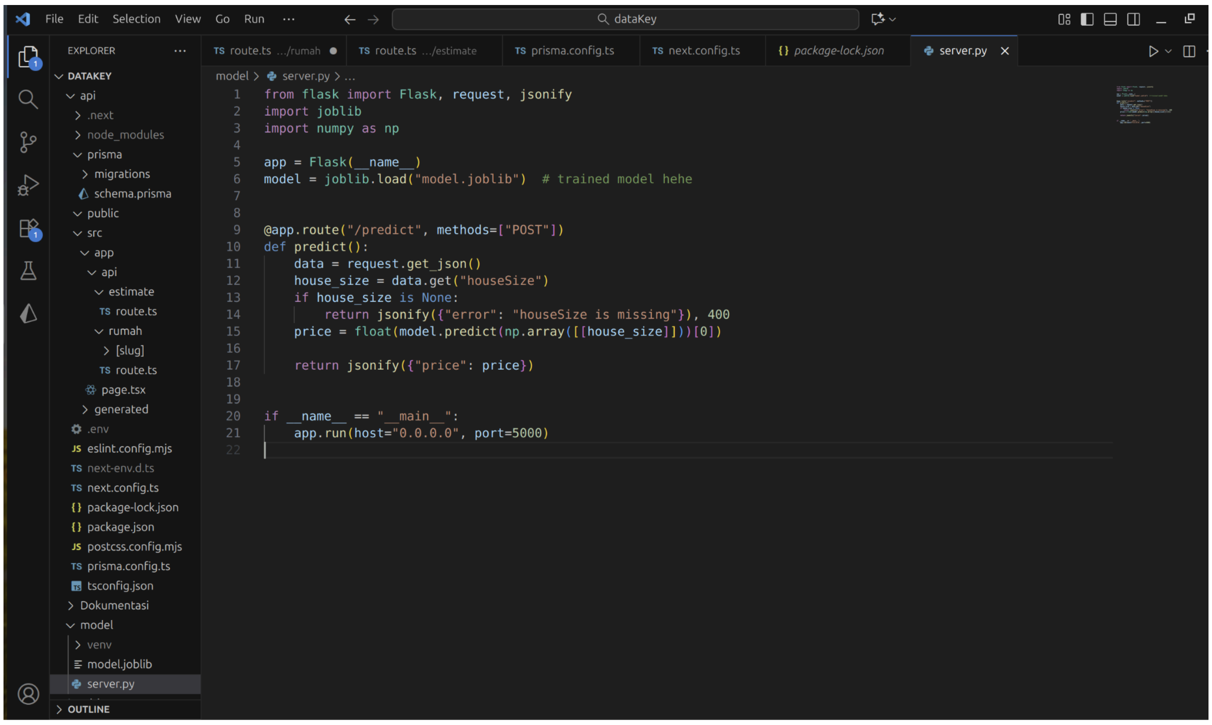
Haidar Ramadhan - 23382003P - Dokumentasi 8

1. Buat logic sebenarnya dalam di dalam src/app/api/estimate, untuk mengirim houseSize sebelumnya melanjutkannya ke python (dalam hal ini Flask), untuk kemudian diproses dan dikembalikan value berupa *price.*



```typescript
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;

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

    // Send houseSize to Flask API
    const resp = await fetch("http://localhost:5000/predict", {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify({ houseSize }),
    });
    const result = await resp.json();
    console.log("Flask response:", result);

    if (!result.price) {
      return NextResponse.json(
        { error: "FLask did not return price" },
        { status: 500 }
      );
    }

    const price = result.price;

    // Save to database
    const saved = await prisma.rumah.create({
      data: { houseSize, price },
    });

    return NextResponse.json({
      message: "Predcition saved succesfully",
      data: saved,
    });
  } catch (error) {
    console.error(error);
    return NextResponse.json(
      { error: "Failed to process request" },
      { status: 500 }
    );
  }
}
```

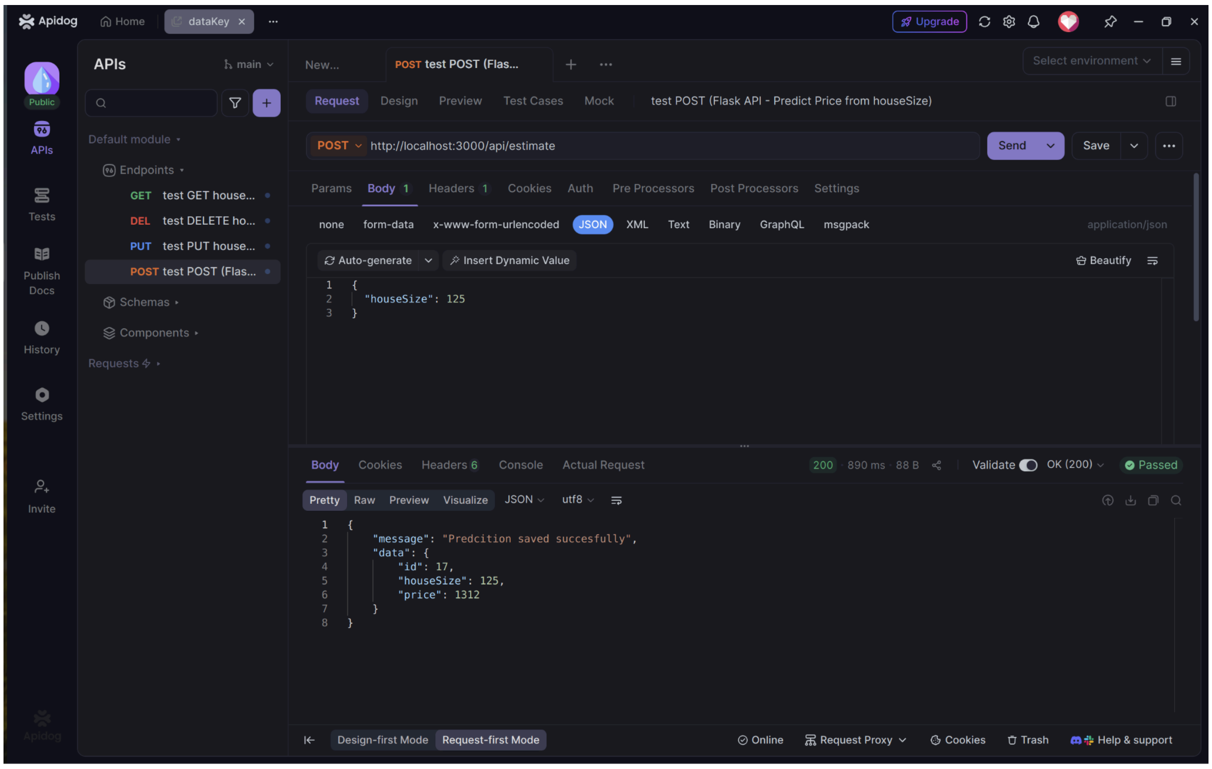
2. Gotcha!, inside model/server.py, you will find a logic code where's Flask serve value houseSize, to process it with the model we have trained, and then return it into *next.js*.



```python
from flask import Flask, request, jsonify
import joblib
import numpy as np

app = Flask(__name__)
model = joblib.load("model.joblib")  # trained model hehe


@app.route("/predict", methods=["POST"])
def predict():
    data = request.get_json()
    house_size = data.get("houseSize")
    if house_size is None:
        return jsonify({"error": "houseSize is missing"}), 400
    price = float(model.predict(np.array([[house_size]]))[0])

    return jsonify({"price": price})


if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
```

Haidar Ramadhan - 23382003P - Dokumentasi 8

test it…!!!



test POST (Flask API - Predict Price from houseSize)

POST http://localhost:3000/api/estimate

```
{
  "houseSize": 125
}
```

200 · 890 ms · 88 B

```
{
    "message": "Predcition saved succesfully",
    "data": {
        "id": 17,
        "houseSize": 125,
        "price": 1312
    }
}
```