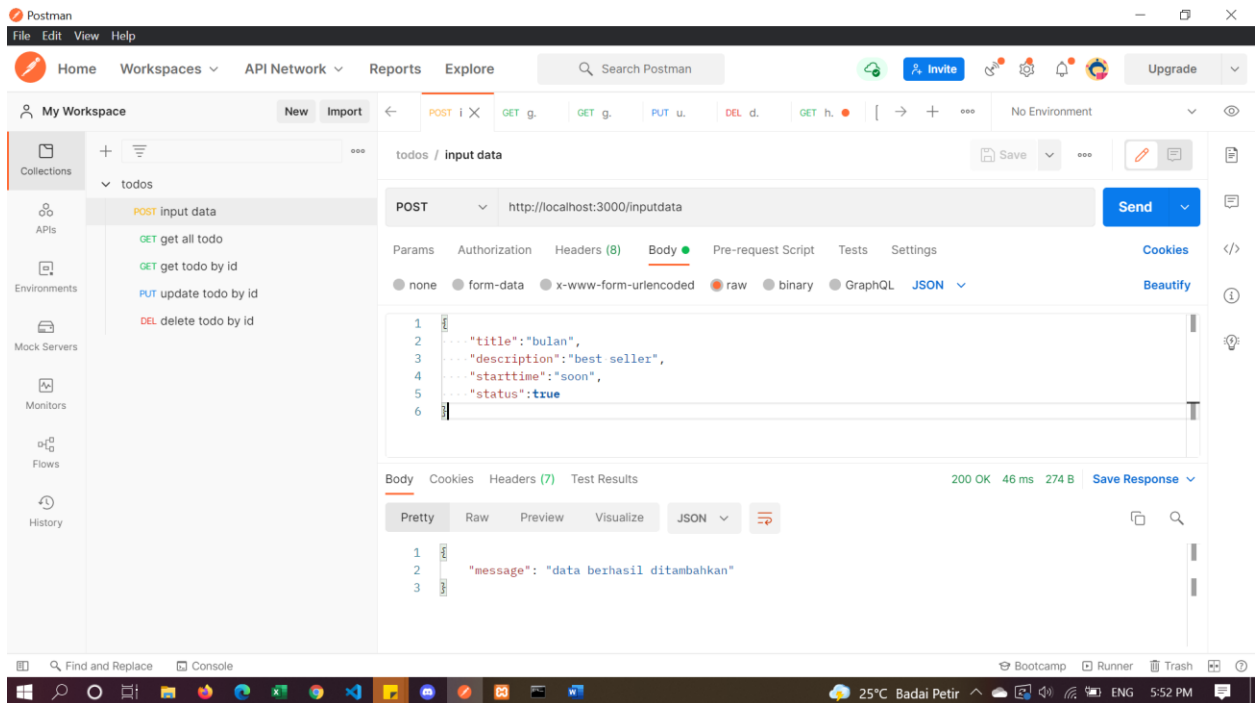
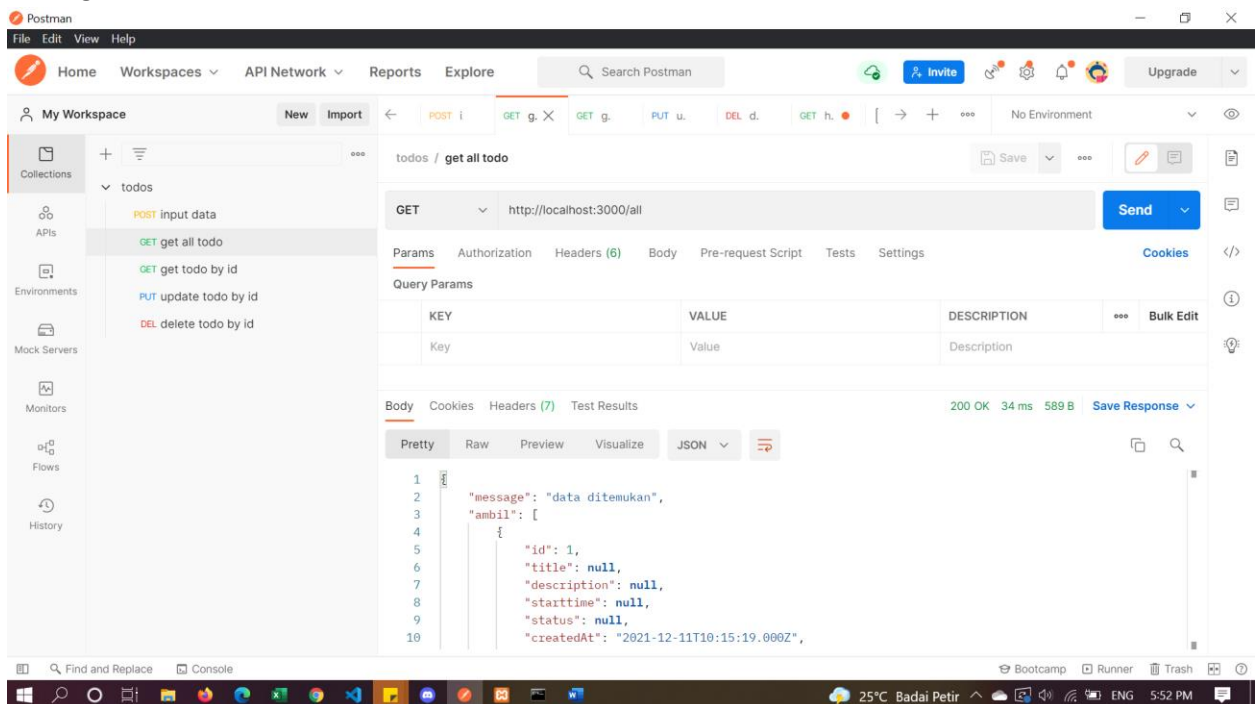


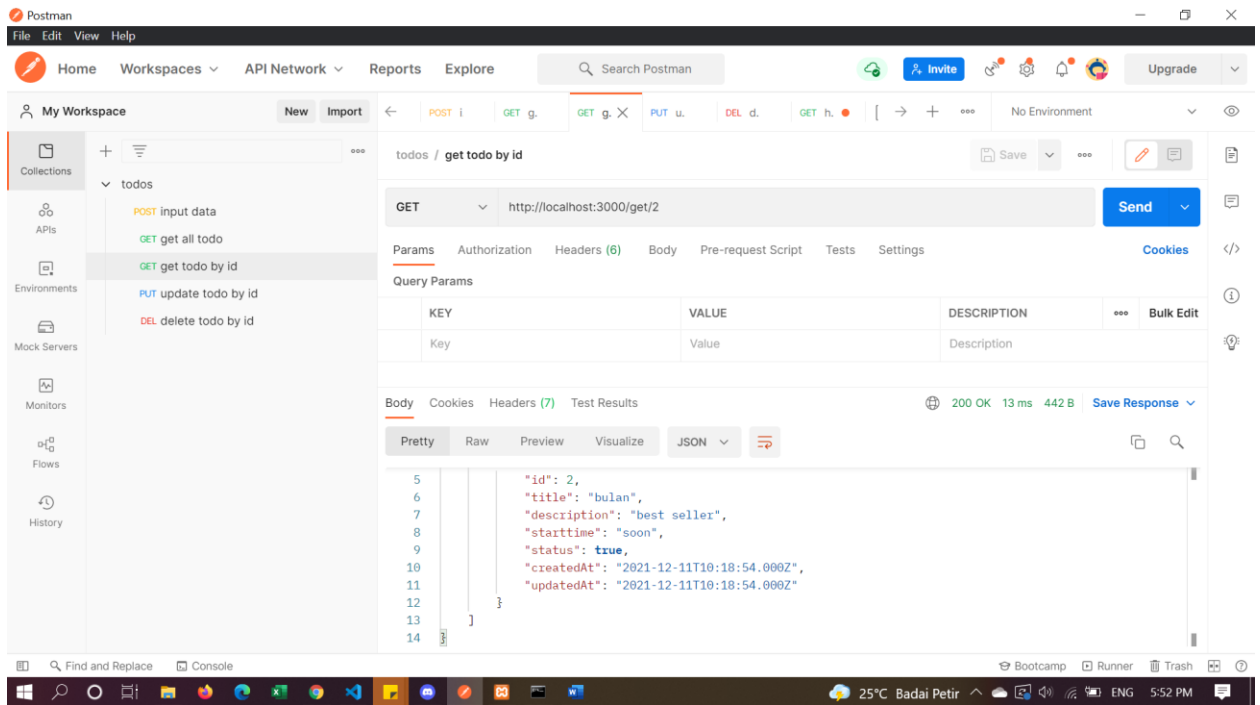
1. Testing POST



2. Testing GET ALL



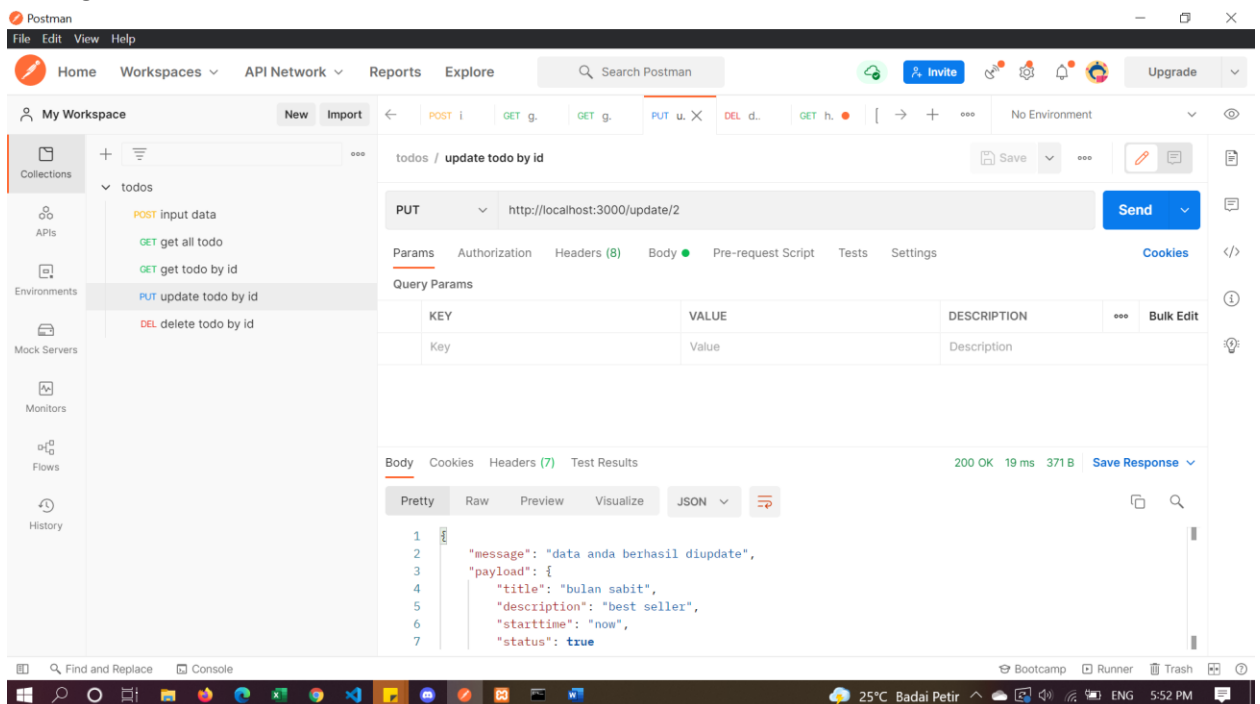
3. Testing GET by ID



The screenshot shows the Postman application with a GET request configured. The request is to the endpoint `http://localhost:3000/get/2`. The response is a 200 OK status with a response time of 13 ms and a body size of 442 B. The response body is displayed in JSON format, showing a todo item with the following details:

```
{
  "id": 2,
  "title": "bulan",
  "description": "best seller",
  "starttime": "soon",
  "status": true,
  "createdAt": "2021-12-11T10:18:54.000Z",
  "updatedAt": "2021-12-11T10:18:54.000Z"
}
```

4. Testing UPDATE



The screenshot shows the Postman application with a PUT request configured. The request is to the endpoint `http://localhost:3000/update/2`. The response is a 200 OK status with a response time of 19 ms and a body size of 371 B. The response body is displayed in JSON format, showing a success message and the updated todo item details:

```
{
  "message": "data anda berhasil diupdate",
  "payload": {
    "title": "bulan sabit",
    "description": "best seller",
    "starttime": "now",
    "status": true
  }
}
```

5. Testing DELETE

The screenshot shows the Postman application interface. On the left sidebar, under 'My Workspace', a collection named 'todos' is expanded, showing several endpoints. The 'delete todo by id' endpoint is selected, which is a DELETE request to 'http://localhost:3000/delete/2'. The main panel shows the request details, including the method 'DELETE', the URL, and the 'Send' button. Below the request details, the 'Query Params' section is empty. The 'Body' section is also empty. The 'Headers' section shows 7 headers. The 'Test Results' section shows a 200 OK status with a response time of 80 ms and a body size of 270 B. The response body is displayed in the 'Body' tab, showing a JSON object: {\"message\": \"data berhasil dihapus\"}.

Postman

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman

My Workspace New Import

collections

todos

- POST input data
- GET get all todo
- GET get todo by id
- PUT update todo by id
- DEL delete todo by id

DELETE http://localhost:3000/delete/2 Send

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

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (7) Test Results 200 OK 80 ms 270 B Save Response

Pretty Raw Preview Visualize JSON

```
1  
2 {  
3   "message": "data berhasil dihapus"
```

25°C Badai Petir ENG 5:53 PM