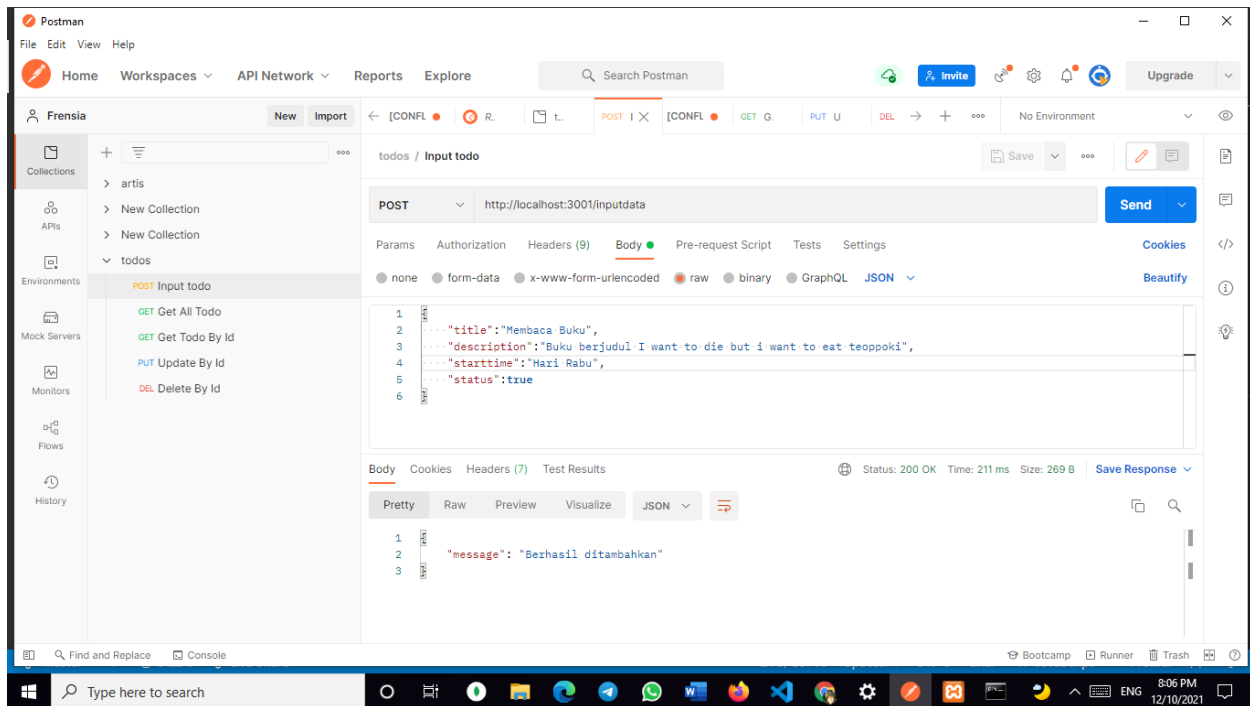


Nama:Frensia Tanaga Anaclaudia

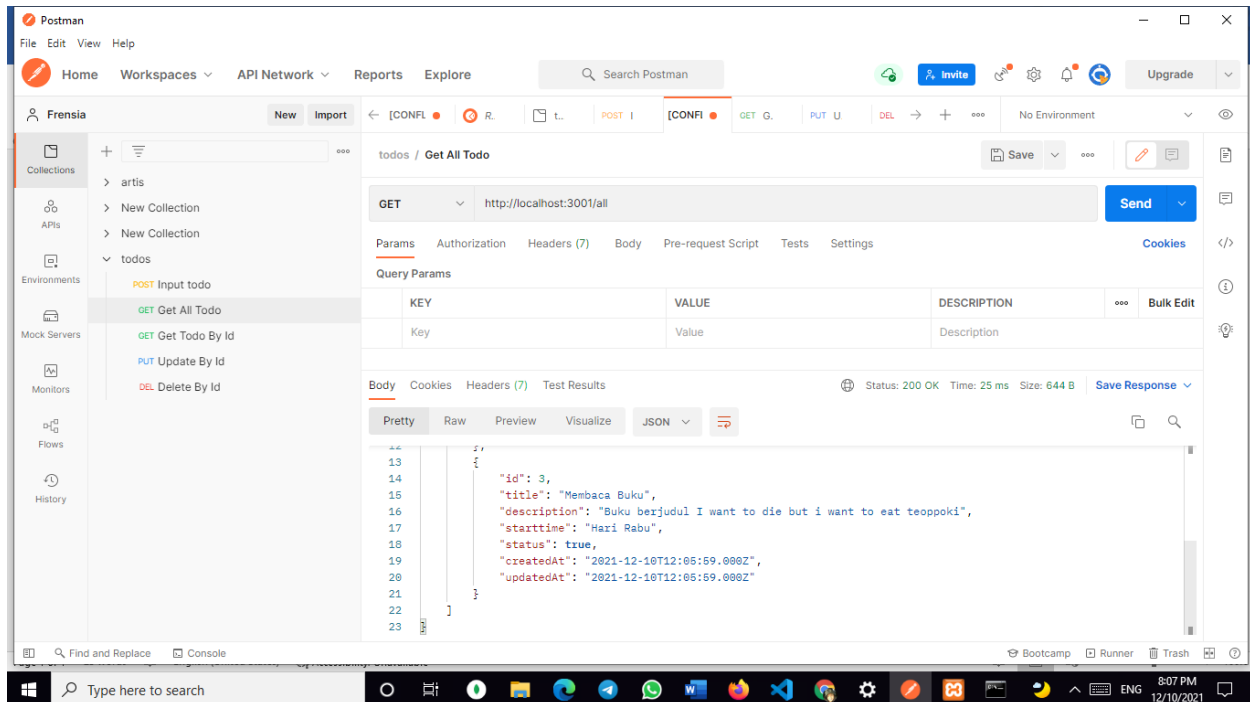
Grup 15

Tugas Testing-API-and-Documentation

-Testing POST dengan memasukan data



-Testing GET ALL untuk mendapatkan semua data yang ada

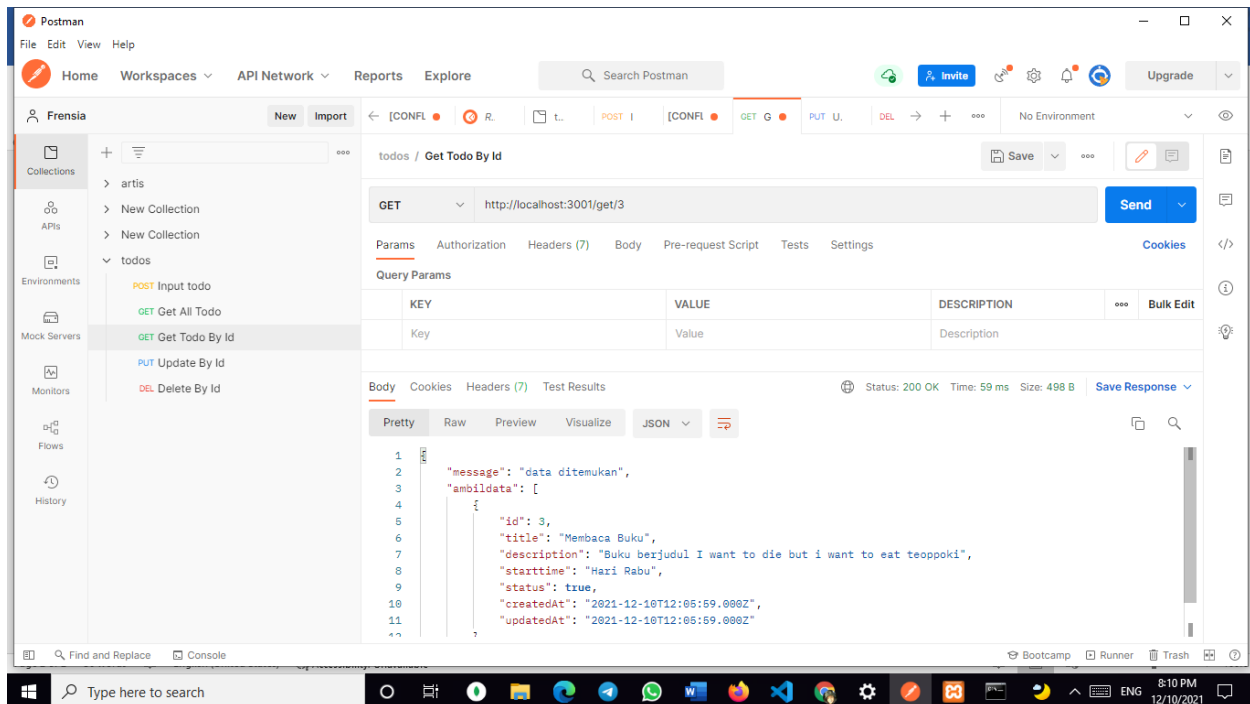


The screenshot shows the Postman application interface. On the left sidebar, the 'Collections' panel is open, showing a collection named 'todos' with several endpoints. The 'GET All Todo' endpoint is selected. The main panel displays the details of this endpoint, including the URL 'http://localhost:3001/all' and the 'Send' button. Below the URL bar, the 'Params' tab is active, showing a table with columns 'KEY', 'VALUE', and 'DESCRIPTION'. The 'Body' tab is also visible, showing the response in 'Pretty' format. The response is a JSON array with one object:

```
[{"id": 3, "title": "Membaca Buku", "description": "Buku berjudul I want to die but i want to eat teoppoki", "starttime": "Hari Rabu", "status": true, "createdAt": "2021-12-10T12:05:59.000Z", "updatedAt": "2021-12-10T12:05:59.000Z"}]
```

The status bar at the bottom indicates 'Status: 200 OK', 'Time: 25 ms', and 'Size: 644 B'.

-Testing GET by ID untuk mendapatkan data berdasarkan ID



The screenshot shows the Postman application interface. On the left sidebar, the 'Collections' panel is open, showing a collection named 'todos' with several endpoints. The 'GET Todo By Id' endpoint is selected. The main panel displays the details of this endpoint, including the URL 'http://localhost:3001/get/3' and the 'Send' button. Below the URL bar, the 'Params' tab is active, showing a table with columns 'KEY', 'VALUE', and 'DESCRIPTION'. The 'Body' tab is also visible, showing the response in 'Pretty' format. The response is a JSON object:

```
{ "message": "data ditemukan", "ambildata": [{"id": 3, "title": "Membaca Buku", "description": "Buku berjudul I want to die but i want to eat teoppoki", "starttime": "Hari Rabu", "status": true, "createdAt": "2021-12-10T12:05:59.000Z", "updatedAt": "2021-12-10T12:05:59.000Z"}]}
```

The status bar at the bottom indicates 'Status: 200 OK', 'Time: 59 ms', and 'Size: 498 B'.

-Testing UPDATE jika terjadi pembaharuan data

The screenshot shows the Postman application interface. On the left sidebar, the 'Collections' panel is open, showing a collection named 'todos' with several endpoints. The 'PUT Update By Id' endpoint is selected. The main panel displays the details of the PUT request to 'http://localhost:3001/update/3'. The request body is a JSON object with the following fields: 'title', 'description', 'starttime', and 'status'. The response is a 200 OK status with a JSON body containing a 'message' and the updated 'payload'.

```
PUT http://localhost:3001/update/3
```

```
{
  "title": "Pergi ke supermarket",
  "description": "Membeli makanan hewan",
  "starttime": "Sore hari",
  "status": true
}
```

```
{
  "message": "data berhasil di update",
  "payload": {
    "title": "Pergi ke supermarket",
    "description": "Membeli makanan hewan",
    "starttime": "Sore hari",
    "status": true
  }
}
```

-Testing DELETE jika ada data yang dihapus

The screenshot shows the Postman application interface. On the left sidebar, the 'Collections' panel is open, showing a collection named 'todos' with several endpoints. The 'DEL Delete By Id' endpoint is selected. The main panel displays the details of the DELETE request to 'http://localhost:3001/delete/1'. The response is a 200 OK status with a JSON body containing a 'message'.

```
DELETE http://localhost:3001/delete/1
```

```
{
  "message": "data di hapus"
}
```