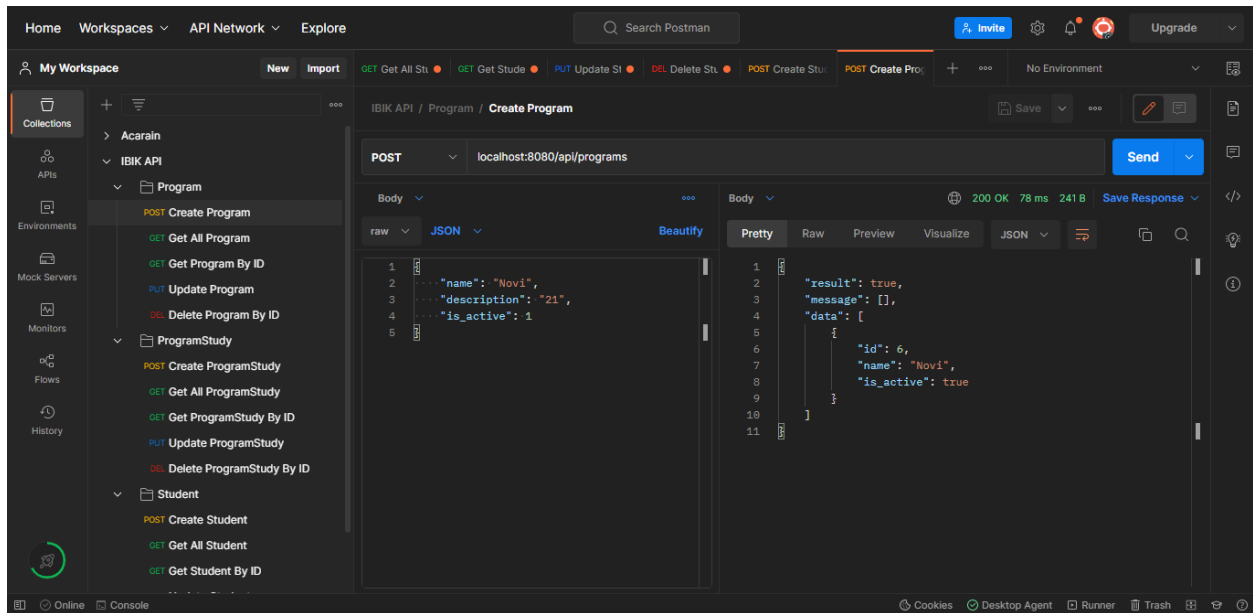


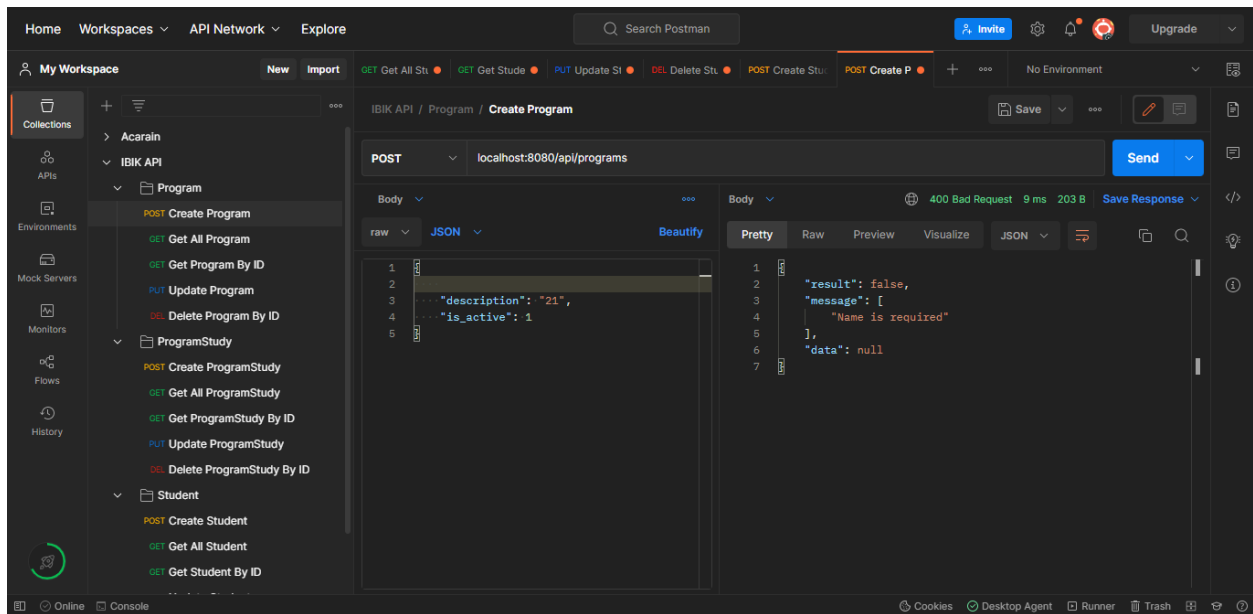
Materi 4 – Validation and Handler

Program

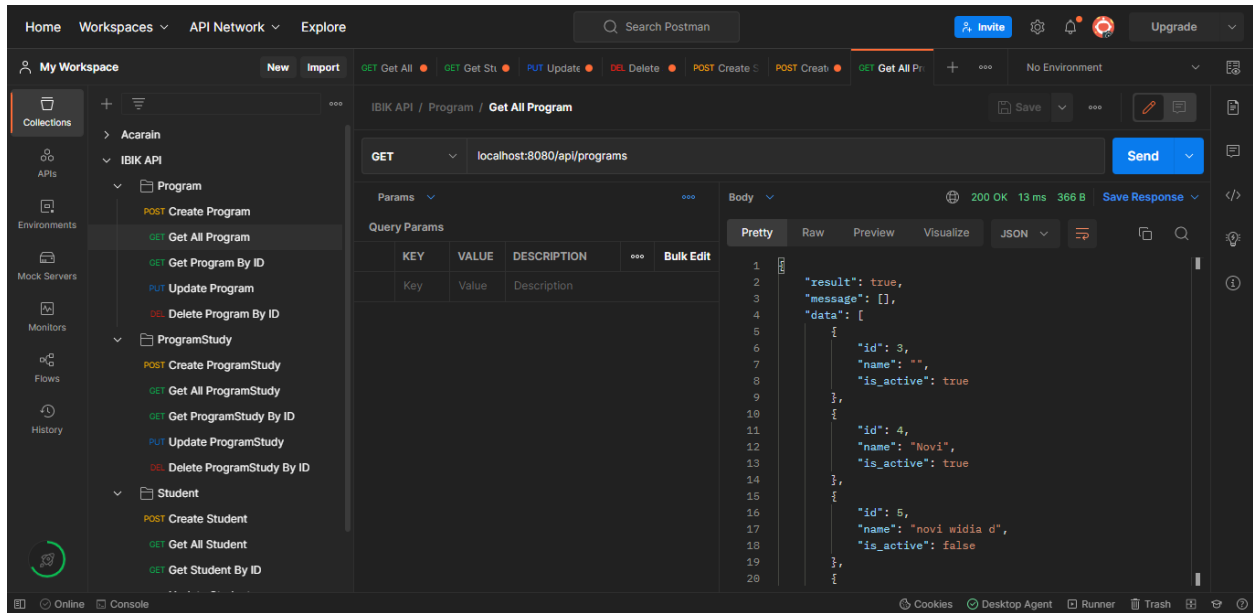
Create Program – Success



Create Program – Error



Get All Programs



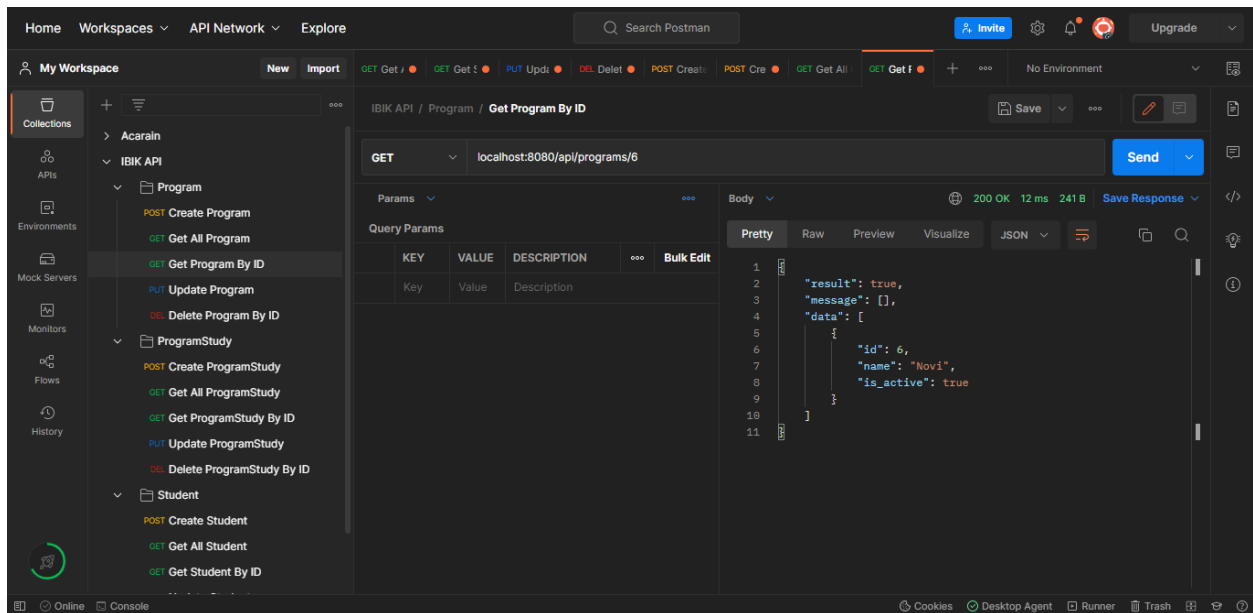
Postman interface showing a successful GET request to `localhost:8080/api/programs`. The response is a JSON array of three program objects.

Request: GET `localhost:8080/api/programs`

Response: 200 OK, 13 ms, 366 B

```
{
  "result": true,
  "message": [],
  "data": [
    {
      "id": 3,
      "name": "",
      "is_active": true
    },
    {
      "id": 4,
      "name": "Novi",
      "is_active": true
    },
    {
      "id": 5,
      "name": "novi widia d",
      "is_active": false
    }
  ]
}
```

Get Program By Id – Success



Postman interface showing a successful GET request to `localhost:8080/api/programs/6`. The response is a JSON object for a program with id 6.

Request: GET `localhost:8080/api/programs/6`

Response: 200 OK, 12 ms, 241 B

```
{
  "result": true,
  "message": [],
  "data": [
    {
      "id": 6,
      "name": "Novi",
      "is_active": true
    }
  ]
}
```

Get Program By Id – Error

The screenshot shows the Postman interface with a GET request to `localhost:8080/api/programs/2`. The response is a 400 Bad Request with a status of 18 ms and 203 B. The response body is:

```
1 {
2   "result": false,
3   "message": [
4     "No value present"
5   ],
6   "data": null
7 }
```

Update Program – Success

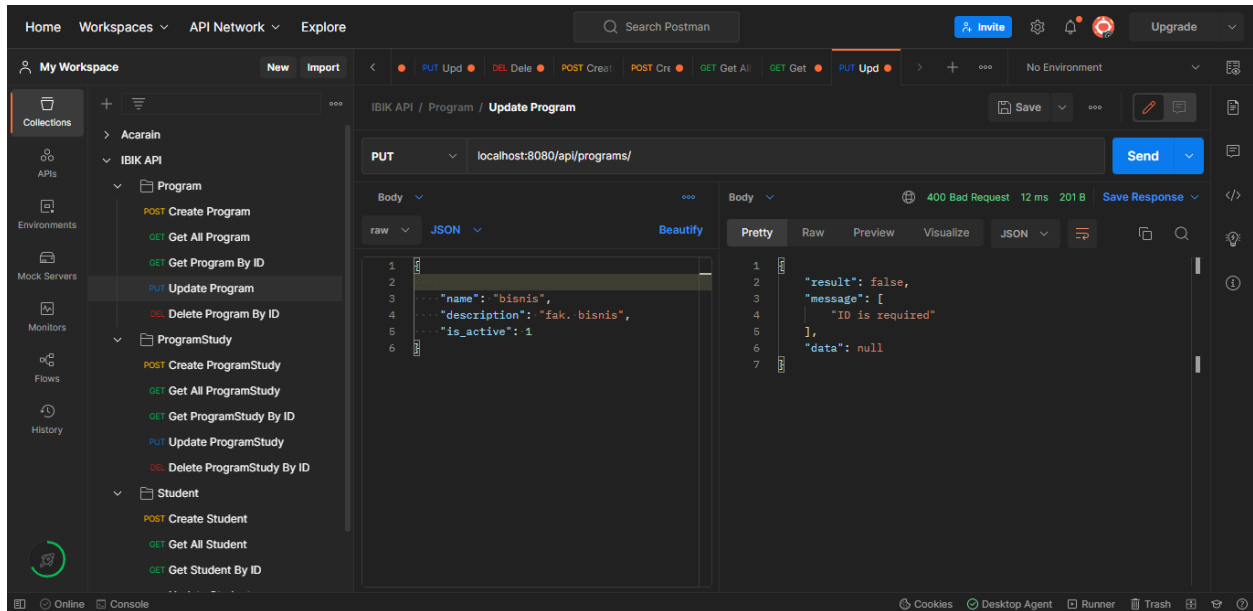
The screenshot shows the Postman interface with a PUT request to `localhost:8080/api/programs/`. The request body is:

```
1 {
2   "id": 6,
3   "name": "bisnis",
4   "description": "fak. bisnis",
5   "is_active": 1
6 }
```

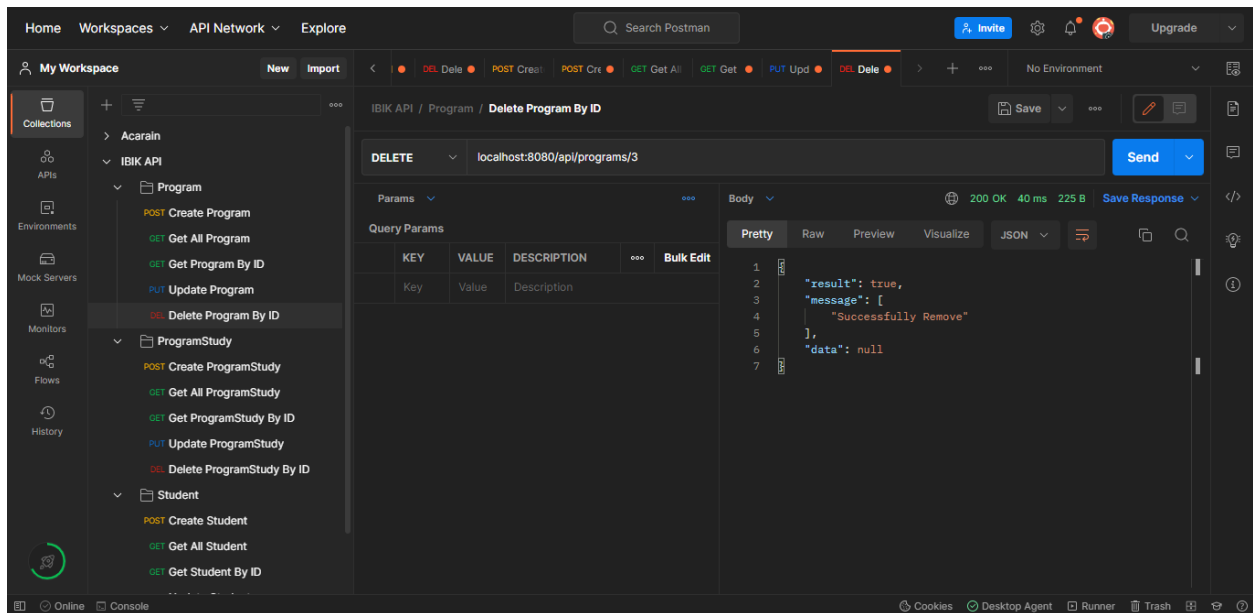
The response is a 200 OK with a status of 38 ms and 243 B. The response body is:

```
1 {
2   "result": true,
3   "message": [],
4   "data": [
5     {
6       "id": 6,
7       "name": "bisnis",
8       "is_active": true
9     }
10  ]
11 }
```

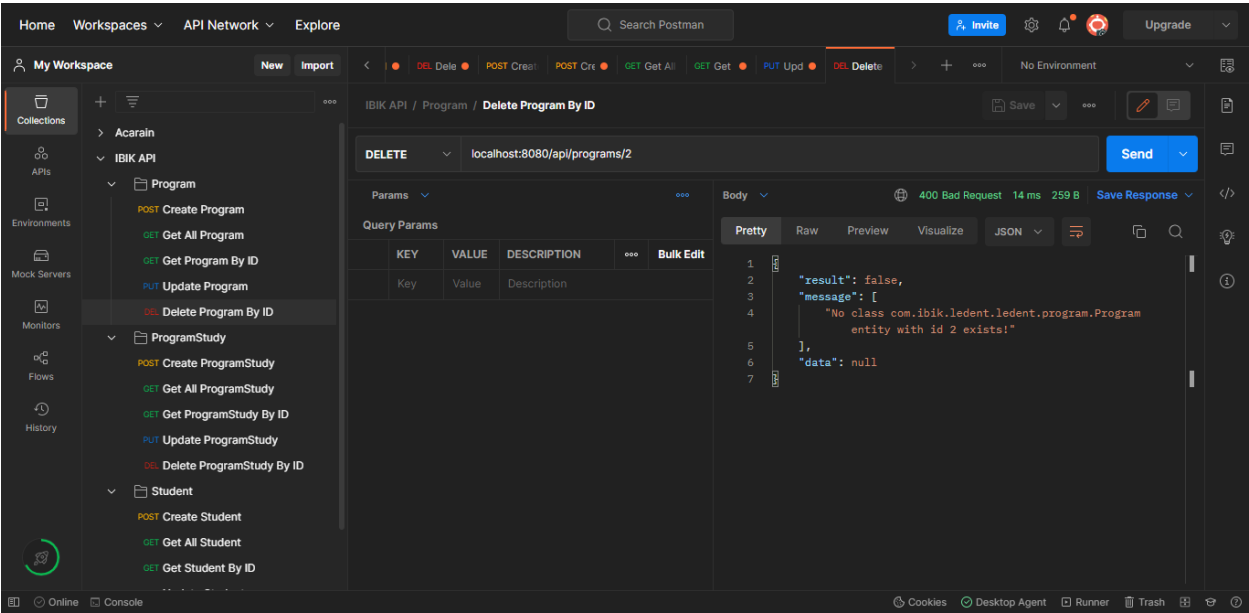
Update Program – Error



Delete Program By Id – Success

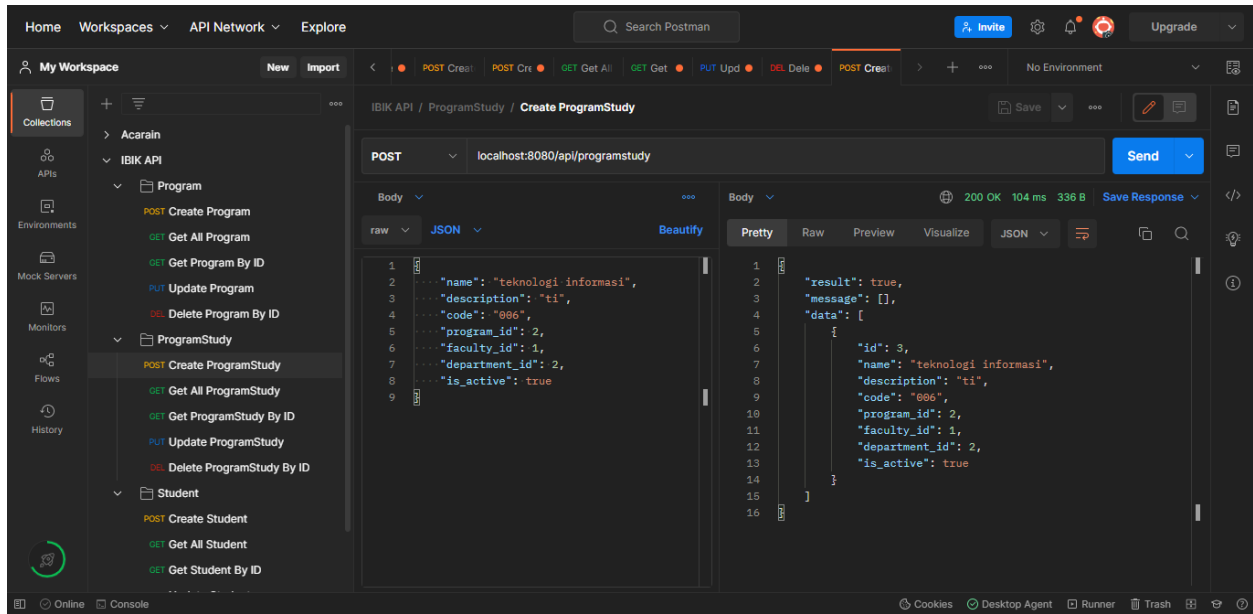


Delete Program By Id – Error

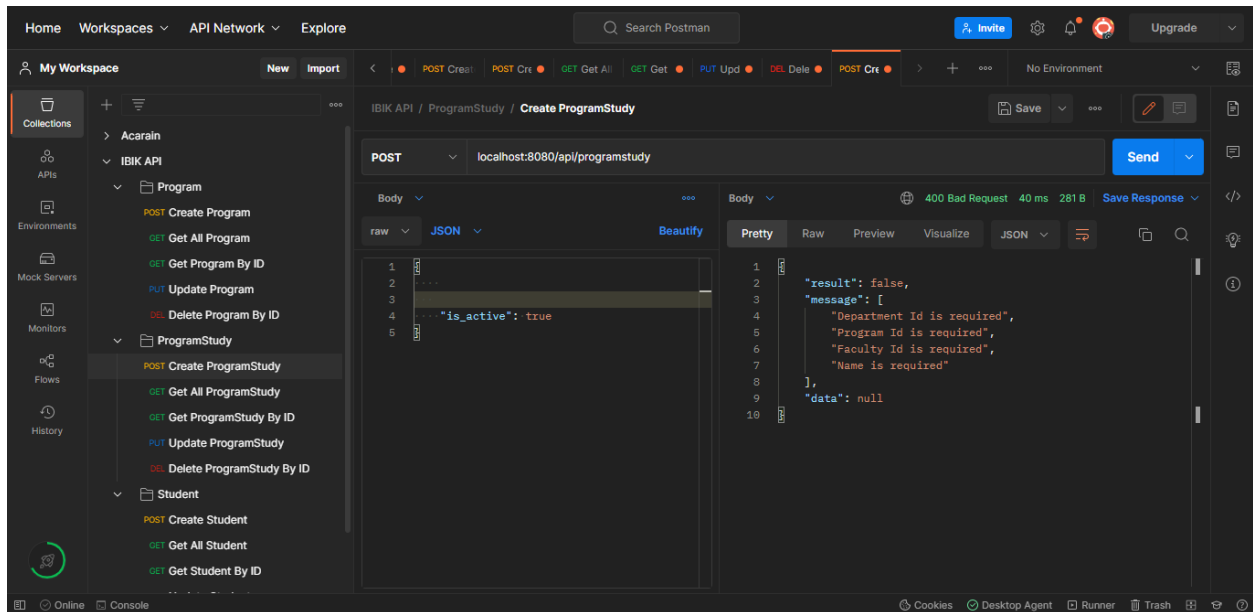


ProgramStudy

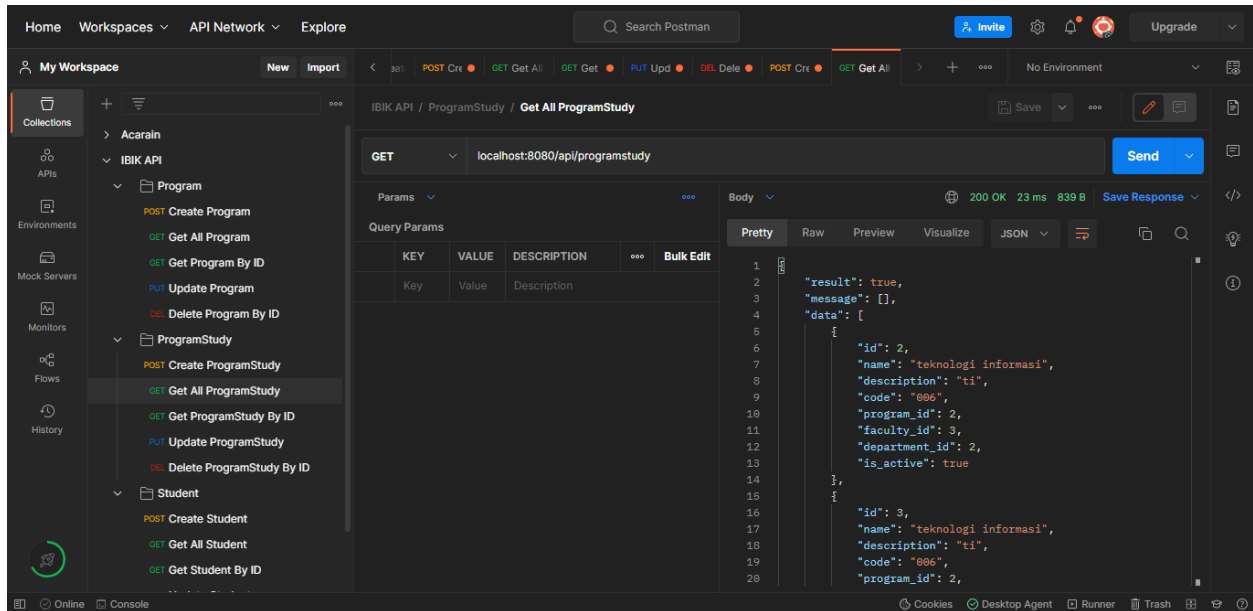
Create ProgramStudy – Success



Create ProgramStudy – Error



Get All ProgramStudy



Postman interface showing a successful GET request to `localhost:8080/api/programstudy`. The response is a 200 OK status with a JSON body containing an array of two program study objects.

Request Details:

- Method: GET
- URL: `localhost:8080/api/programstudy`
- Params: None
- Body: None

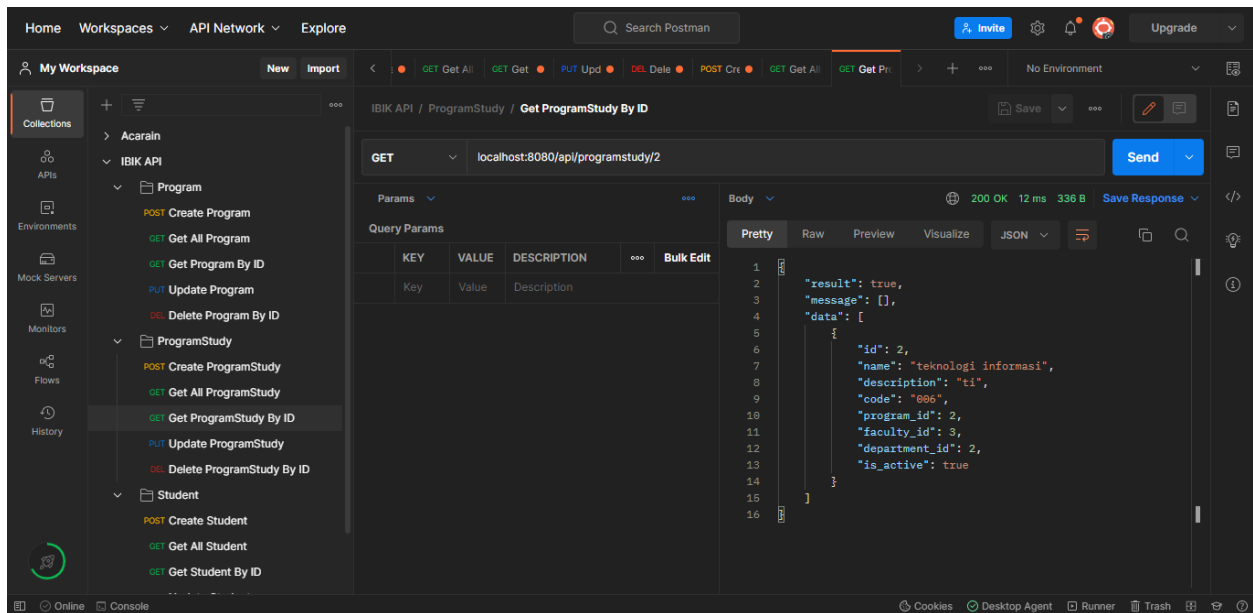
Response Details:

- Status: 200 OK
- Time: 23 ms
- Size: 839 B

Response Body (Pretty):

```
{
  "result": true,
  "message": [],
  "data": [
    {
      "id": 2,
      "name": "teknologi informasi",
      "description": "ti",
      "code": "006",
      "program_id": 2,
      "faculty_id": 3,
      "department_id": 2,
      "is_active": true
    },
    {
      "id": 3,
      "name": "teknologi informasi",
      "description": "ti",
      "code": "006",
      "program_id": 2,
      "faculty_id": 3,
      "department_id": 2,
      "is_active": true
    }
  ]
}
```

Get ProgramStudy By Id – Success



Postman interface showing a successful GET request to `localhost:8080/api/programstudy/2`. The response is a 200 OK status with a JSON body containing a single program study object.

Request Details:

- Method: GET
- URL: `localhost:8080/api/programstudy/2`
- Params: None
- Body: None

Response Details:

- Status: 200 OK
- Time: 12 ms
- Size: 336 B

Response Body (Pretty):

```
{
  "result": true,
  "message": [],
  "data": [
    {
      "id": 2,
      "name": "teknologi informasi",
      "description": "ti",
      "code": "006",
      "program_id": 2,
      "faculty_id": 3,
      "department_id": 2,
      "is_active": true
    }
  ]
}
```

Get ProgramStudy By Id – Error

The screenshot shows the Postman interface with a GET request to `localhost:8080/api/programstudy/7`. The response is a 400 Bad Request with the following JSON body:

```
{  "result": false,  "message": [    "No value present"  ],  "data": null}
```

The left sidebar shows the API collection structure:

- Acacain
 - IBIK API
 - Program
 - POST Create Program
 - GET Get All Program
 - GET Get Program By ID
 - PUT Update Program
 - DEL Delete Program By ID
 - ProgramStudy
 - POST Create ProgramStudy
 - GET Get All ProgramStudy
 - GET Get ProgramStudy By ID
 - PUT Update ProgramStudy
 - DEL Delete ProgramStudy By ID
 - Student
 - POST Create Student
 - GET Get All Student
 - GET Get Student By ID

Update ProgramStudy – Success

The screenshot shows the Postman interface with a PUT request to `localhost:8080/api/programstudy`. The response is a 200 OK with the following JSON body:

```
{  "result": true,  "message": [],  "data": [    {      "id": 7,      "name": "manajemen",      "description": "mene",      "code": "003",      "program_id": 1,      "faculty_id": 2,      "department_id": 2,      "is_active": true    }  ]}
```

The left sidebar shows the API collection structure:

- Acacain
 - IBIK API
 - Program
 - POST Create Program
 - GET Get All Program
 - PUT Update Program
 - DEL Delete Program By ID
 - ProgramStudy
 - POST Create ProgramStudy
 - GET Get All ProgramStudy
 - GET Get ProgramStudy By ID
 - PUT Update ProgramStudy
 - DEL Delete ProgramStudy By ID
 - Student
 - POST Create Student
 - GET Get All Student
 - GET Get Student By ID

Update ProgramStudy – Error

The screenshot shows the Postman interface with a PUT request to `localhost:8080/api/programstudy`. The request body is a JSON object:

```
1 {
2   "description": "mene",
3   "code": "993",
4   "program_id": 1,
5   "faculty_id": 2,
6   "department_id": 2,
7   "is_active": true
8 }
9
```

The response is a 400 Bad Request with the following JSON body:

```
1 {
2   "result": false,
3   "message": [
4     "ID is required"
5   ],
6   "data": null
7 }
```

The status bar at the bottom indicates "400 Bad Request 9 ms 201 B".

Delete ProgramStudy By Id – Success

The screenshot shows the Postman interface with a DELETE request to `localhost:8080/api/programstudy/7`. The response is a 200 OK with the following JSON body:

```
1 {
2   "result": true,
3   "message": [
4     "Successfully Remove"
5   ],
6   "data": null
7 }
```

The status bar at the bottom indicates "200 OK 70 ms 225 B".

Delete ProgramStudy By Id – Error

The screenshot shows the Postman interface with a DELETE request to `localhost:8080/api/programstudy/7`. The response is a 400 Bad Request with the following JSON body:

```
{  "result": false,  "message": [    "No class com.ibik.ledent.ledent.programstudy.ProgramStudy entity with id 7 exists!"  ],  "data": null}
```

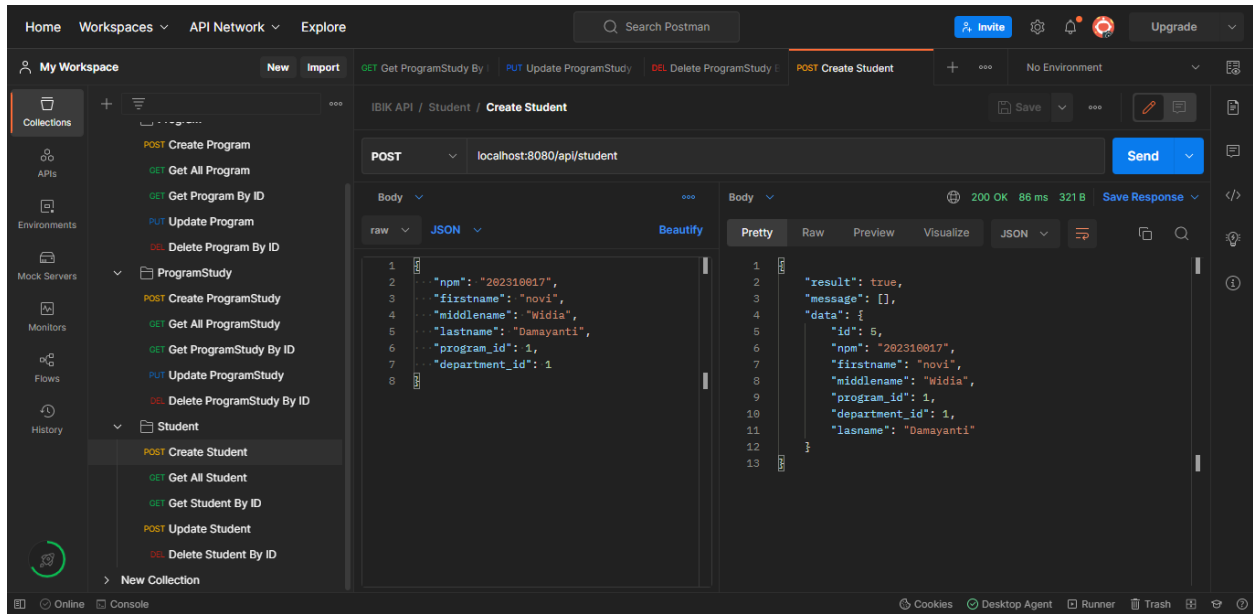
The left sidebar shows the API Explorer with the following endpoints:

- Acacain
 - POST Create Program
 - GET Get All Program
 - GET Get Program By ID
 - PUT Update Program
 - DEL Delete Program By ID
- ProgramStudy
 - POST Create ProgramStudy
 - GET Get All ProgramStudy
 - GET Get ProgramStudy By ID
 - PUT Update ProgramStudy
 - DEL Delete ProgramStudy By ID
- Student
 - POST Create Student
 - GET Get All Student
 - GET Get Student By ID

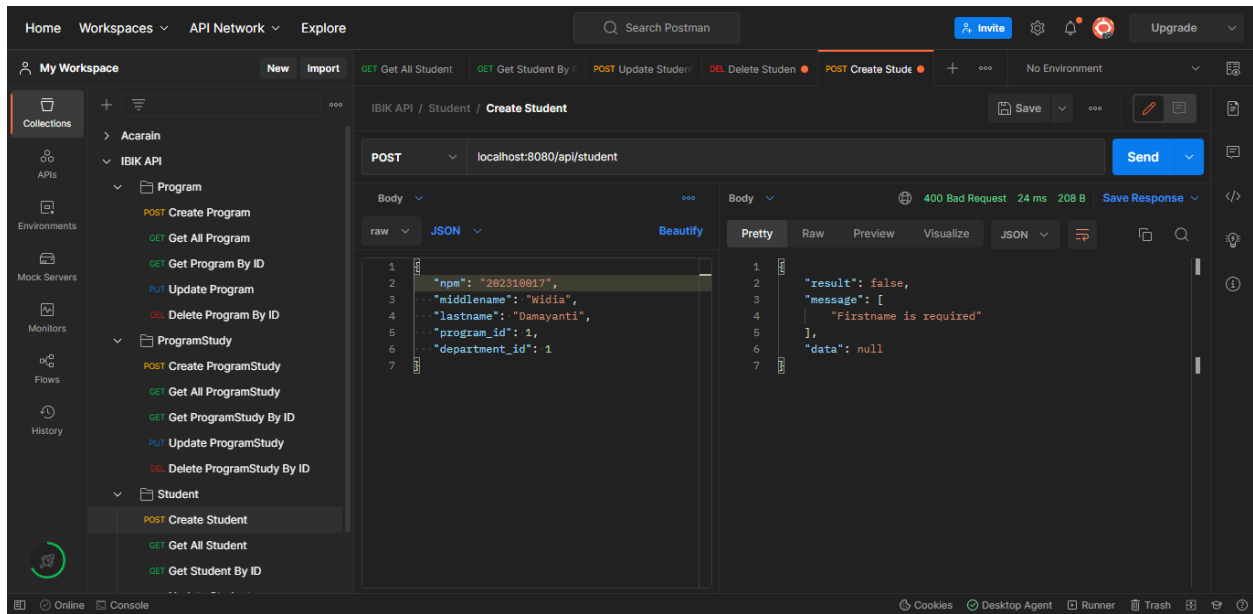
The bottom status bar shows the following icons: Online, Console, Cookies, Desktop Agent, Runner, Trash, and Help.

Student

Create Student - Success



Create Student - Error



Get All Student - Success

The screenshot shows the Postman interface with a workspace named 'My Workspace'. The left sidebar lists collections, including 'AcaraIn', 'IBIK API', 'Program', 'ProgramStudy', and 'Student'. The 'Student' collection is expanded, showing a 'GET Get All Student' request. The main panel displays the details of this request, which is a GET method to 'localhost:8080/api/student'. The response is a 200 OK status with a 32 ms response time and 795 B of data. The response body is shown in the 'Pretty' tab, displaying a JSON array of two student objects.

```
{
  "result": true,
  "message": [],
  "data": [
    {
      "id": 1,
      "npm": null,
      "firstname": null,
      "middlename": "Widia",
      "program_id": 1,
      "department_id": 1,
      "lastname": null
    },
    {
      "id": 3,
      "npm": "202310017",
      "firstname": "novi",
      "middlename": "Widia",
      "program_id": 1,
      "department_id": 1
    }
  ]
}
```

Get Student By Id - Success

The screenshot shows the Postman interface with the same workspace. The 'Student' collection is expanded, showing a 'GET Get Student By ID' request. The main panel displays the details of this request, which is a GET method to 'localhost:8080/api/student/1'. The response is a 200 OK status with a 27 ms response time and 323 B of data. The response body is shown in the 'Pretty' tab, displaying a JSON object for a single student.

```
{
  "result": true,
  "message": [],
  "data": [
    {
      "id": 1,
      "npm": "202310017",
      "firstname": "novi",
      "middlename": "Widia",
      "program_id": 1,
      "department_id": 1,
      "lastname": "Damayanti"
    }
  ]
}
```

Get Student By Id - Error

The screenshot shows the Postman interface with a workspace named "My Workspace". The "IBIK API" collection is expanded, and the "Student" folder is selected. The "Get Student By ID" endpoint is highlighted. The request is a GET method to the URL "localhost:8080/api/student/2". The response is a 400 Bad Request with a status of 400, a time of 11 ms, and a body of 203 B. The response body is displayed in the "Pretty" view, showing a JSON object with "result": false, "message": ["No value present"], and "data": null.

```
{  "result": false,  "message": [    "No value present"  ],  "data": null}
```

Update Student - Success

The screenshot shows the Postman interface with a workspace named "My Workspace". The "IBIK API" collection is expanded, and the "Student" folder is selected. The "Update Student" endpoint is highlighted. The request is a POST method to the URL "localhost:8080/api/student". The request body is a JSON object with the following fields: "id": 1, "npm": "11786286", "firstname": "Novi", "middlename": "Widia", "lastname": "Damayanti", "program_id": 1, and "department_id": 1. The response is a 200 OK with a status of 200, a time of 66 ms, and a body of 322 B. The response body is displayed in the "Pretty" view, showing a JSON object with "result": true, "message": [], and "data": [{"id": 1, "npm": "11786286", "firstname": "Novi", "middlename": "Widia", "program_id": 1, "department_id": 1, "lastname": "Damayanti"}].

```
{  "result": true,  "message": [],  "data": [    {      "id": 1,      "npm": "11786286",      "firstname": "Novi",      "middlename": "Widia",      "program_id": 1,      "department_id": 1,      "lastname": "Damayanti"    }  ]}
```


Update Student - Error

The screenshot shows the Postman interface with a workspace named 'My Workspace'. The 'API Network' tab is active, displaying a collection of API endpoints. The selected endpoint is 'PUT Update Student' with the URL 'localhost:8080/api/student'. The request body is a JSON object:

```
{  "id": "11786286",  "firstname": "Novi",  "middlename": "Widia",  "lastname": "Damayanti",  "program_id": 1,  "department_id": 1}
```

. The response is a 400 Bad Request with a message:

```
{  "result": false,  "message": [    "ID is required"  ],  "data": null}
```

.

IBIK API / Student / Update Student

PUT localhost:8080/api/student

Body

```
1 {
2   "id": "11786286",
3   "firstname": "Novi",
4   "middlename": "Widia",
5   "lastname": "Damayanti",
6   "program_id": 1,
7   "department_id": 1
8 }
```

Body

```
1 {
2   "result": false,
3   "message": [
4     "ID is required"
5   ],
6   "data": null
7 }
```

400 Bad Request 8 ms 201 B Save Response

Delete Student By Id - Success

The screenshot shows the Postman interface with a workspace named 'My Workspace'. The 'API Network' tab is active, displaying a collection of API endpoints. The selected endpoint is 'DELETE Delete Student By ID' with the URL 'localhost:8080/api/student/1'. The response is a 200 OK with a message:

```
{  "result": true,  "message": [    "Successfully Remove"  ],  "data": null}
```

.

IBIK API / Student / Delete Student By ID

DELETE localhost:8080/api/student/1

Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body

```
1 {
2   "result": true,
3   "message": [
4     "Successfully Remove"
5   ],
6   "data": null
7 }
```

200 OK 118 ms 225 B Save Response

Delete Student By Id - Error

The screenshot shows the Postman interface with a workspace named "My Workspace". The selected collection is "IBIK API / Student / Delete Student By ID". The request method is "DELETE" and the URL is "localhost:8080/api/student/1". The response is a 400 Bad Request with a 10 ms response time and 259 B of data. The response body is displayed in the "Pretty" view, showing a JSON object with a "result" of false and a "message" array containing an error message.

IBIK API / Student / Delete Student By ID

DELETE localhost:8080/api/student/1

Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Body

400 Bad Request 10 ms 259 B Save Response

```
1 {
2   "result": false,
3   "message": [
4     "No class com.ibik.ledent.ledent.student.Student
5     entity with id 1 exists!"
6   ],
7   "data": null
}
```