

Materi 3

Program

Create Program

The screenshot shows the Postman interface for the 'Create Program' endpoint. The left sidebar displays the 'My Workspace' with a tree view of collections: 'Acarain', 'IBIK API', and 'Program'. The 'Program' collection is expanded, showing endpoints: 'POST Create Program', 'GET Get All Program', 'GET Get Program By ID', 'POST Update Program', and 'DEL Delete Program By ID'. The main panel shows the 'POST Create Program' endpoint with the URL 'localhost:8080/api/programs'. The request body is set to 'JSON' and contains the following JSON:

```
{  "name": "Novi",  "description": "21",  "is_active": 1}
```

The response is displayed in the 'Pretty' view, showing a 200 OK status with a 269 ms response time and 203 B of data. The response body is:

```
{  "id": 4,  "name": "Novi",  "is_active": true}
```

Get All Programs

The screenshot shows the Postman interface for the 'Get All Programs' endpoint. The left sidebar displays the 'My Workspace' with a tree view of collections: 'Acarain', 'IBIK API', and 'Program'. The 'Program' collection is expanded, showing endpoints: 'POST Create Program', 'GET Get All Program', 'GET Get Program By ID', 'POST Update Program', and 'DEL Delete Program By ID'. The main panel shows the 'GET Get All Program' endpoint with the URL 'localhost:8080/api/programs'. The request is a simple GET request. The response is displayed in the 'Pretty' view, showing a 200 OK status with a 512 ms response time and 279 B of data. The response body is:

```
{  "id": 2,  "name": null,  "is_active": true}, {  "id": 3,  "name": "",  "is_active": true}, {  "id": 4,  "name": "Novi",  "is_active": true}
```

Get Program By Id

The screenshot shows the Postman interface with the 'Get Program By ID' endpoint selected. The endpoint is a GET request to `localhost:8080/api/programs/2`. The response is a 200 OK status with a response time of 136 ms and a body of `{ "id": 2, "name": null, "is_active": true }`.

Endpoint: GET `localhost:8080/api/programs/2`

Params: Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body: `{ "id": 2, "name": null, "is_active": true }`

Status: 200 OK, 136 ms, 201 B

Update Program

The screenshot shows the Postman interface with the 'Update Program' endpoint selected. The endpoint is a PUT request to `localhost:8080/api/programs/`. The request body is a JSON object: `{ "id": 2, "name": "novi widia d", "description": "202310017", "is_active": 1 }`. The response is a 200 OK status with a response time of 61 ms and a body of `{ "id": 2, "name": "novi widia d", "is_active": true }`.

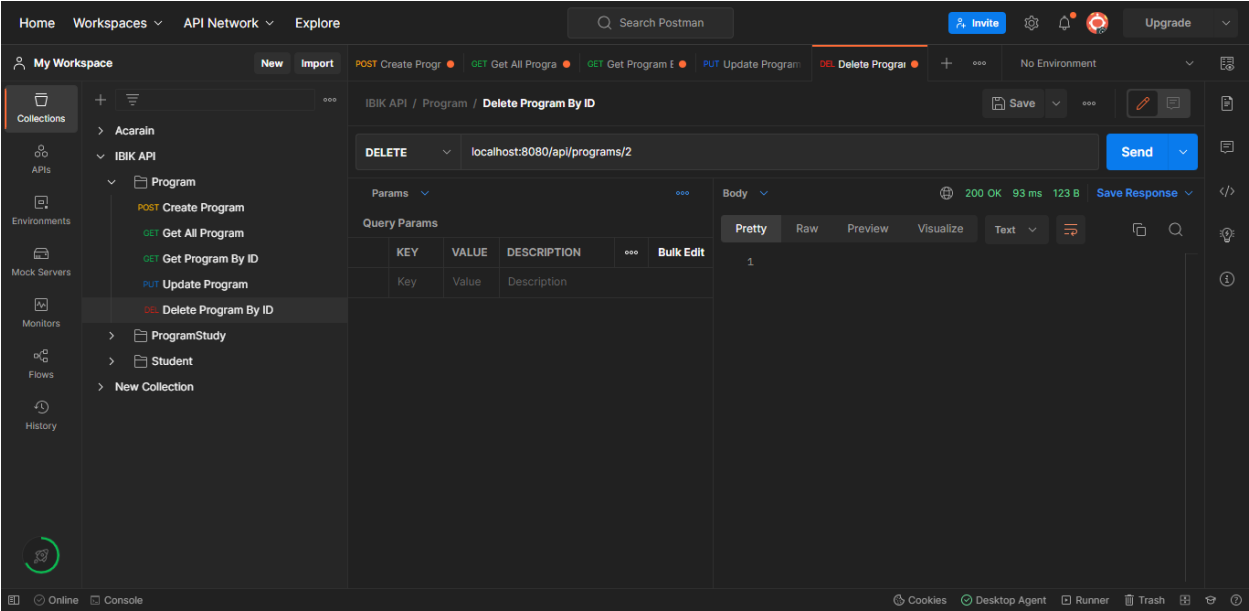
Endpoint: PUT `localhost:8080/api/programs/`

Body: `{ "id": 2, "name": "novi widia d", "description": "202310017", "is_active": 1 }`

Status: 200 OK, 61 ms, 211 B

Response Body: `{ "id": 2, "name": "novi widia d", "is_active": true }`

Delete Program By Id



ProgramStudy

Create ProgramStudy

Home Workspaces API Network Explore

My Workspace

IBIK API / ProgramStudy / Create ProgramStudy

POST localhost:8080/api/programstudy

Body JSON Beautify Pretty Raw Preview Visualize JSON

200 OK 83 ms 298 B Save Response

```
1 {
2   "name": "teknologi informasi",
3   "description": "ti",
4   "code": "006",
5   "program_id": 2,
6   "faculty_id": 1,
7   "department_id": 2,
8   "is_active": true
9 }
```

```
1 {
2   "id": 2,
3   "name": "teknologi informasi",
4   "description": "ti",
5   "code": "006",
6   "program_id": 2,
7   "faculty_id": 1,
8   "department_id": 2,
9   "is_active": true
10 }
```

Online Console Cookies Desktop Agent Runner Trash

Get All ProgramStudy

Home Workspaces API Network Explore

My Workspace

IBIK API / ProgramStudy / Get All ProgramStudy

GET localhost:8080/api/programstudy

Params Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

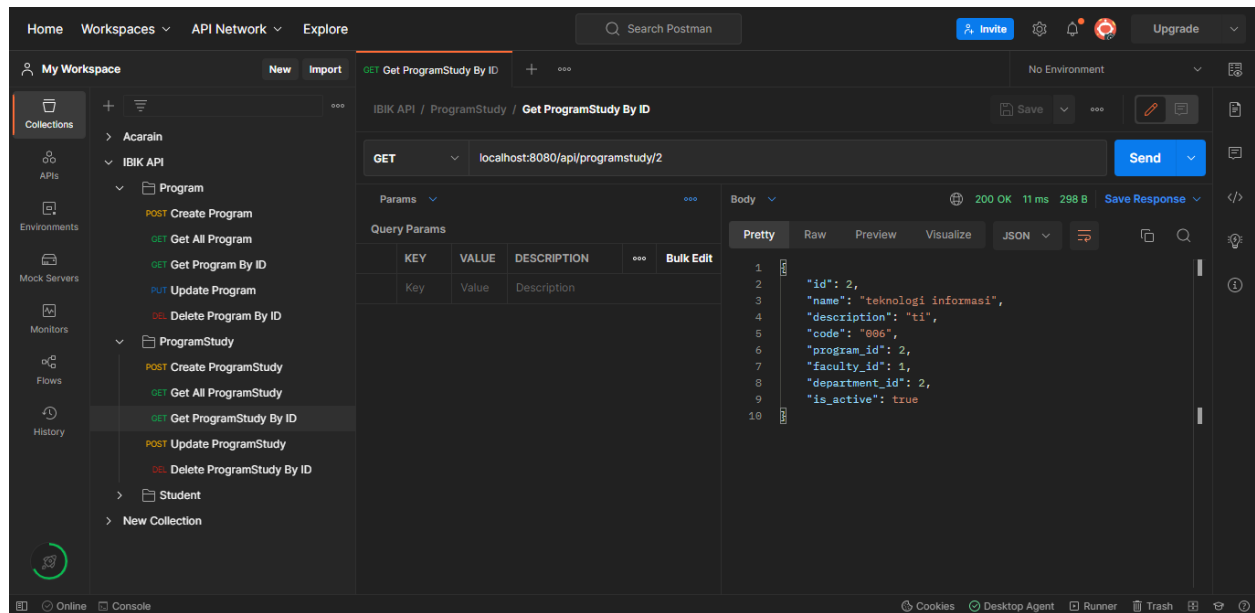
Body Pretty Raw Preview Visualize JSON

200 OK 15 ms 419 B Save Response

```
2 {
3   "id": 1,
4   "name": "novi",
5   "description": "21",
6   "code": null,
7   "program_id": 0,
8   "faculty_id": 0,
9   "department_id": 0,
10  "is_active": true
11 },
12 {
13   "id": 2,
14   "name": "teknologi informasi",
15   "description": "ti",
16   "code": "006",
17   "program_id": 2,
18   "faculty_id": 1,
19   "department_id": 2,
20   "is_active": true
21 }
```

Online Console Cookies Desktop Agent Runner Trash

Get ProgramStudy By Id

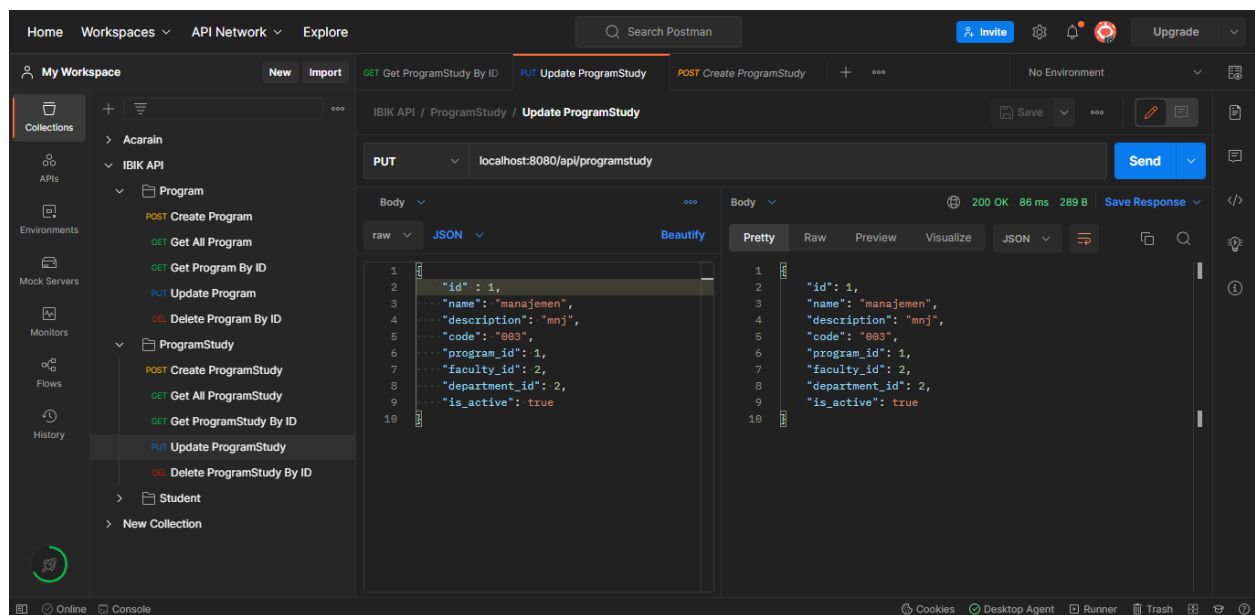


The screenshot shows the Postman interface with a GET request to `localhost:8080/api/programstudy/2`. The response is a JSON object with the following structure:

```
1 {
2   "id": 2,
3   "name": "teknologi informasi",
4   "description": "ti",
5   "code": "006",
6   "program_id": 2,
7   "faculty_id": 1,
8   "department_id": 2,
9   "is_active": true
10 }
```

The status bar indicates a 200 OK response with 11 ms latency and 298 B of data.

Update ProgramStudy



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

```
1 {
2   "id": 1,
3   "name": "manajemen",
4   "description": "mnj",
5   "code": "003",
6   "program_id": 1,
7   "faculty_id": 2,
8   "department_id": 2,
9   "is_active": true
10 }
```

The status bar indicates a 200 OK response with 86 ms latency and 289 B of data.

Delete ProgramStudy By Id

The image shows the Postman application interface. The top navigation bar includes 'Home', 'Workspaces', 'API Network', and 'Explore'. A search bar labeled 'Search Postman' is present. On the right, there are buttons for 'Invite', 'Upgrade', and a notification bell.

The left sidebar, titled 'My Workspace', shows a tree view of collections. The 'IBIK API' collection is expanded, showing sub-collections like 'Program' and 'ProgramStudy'. The 'Delete ProgramStudy By ID' endpoint is selected.

The main workspace displays the details of the selected endpoint:

- Method: DELETE
- URL: localhost:8080/apl/programstudy/1
- Params: Empty
- Body: Empty
- Response: 200 OK, 43 ms, 123 B

Below the main workspace, there is a 'Query Params' table with columns: KEY, VALUE, DESCRIPTION, and Bulk Edit. The table is currently empty.

The bottom status bar shows 'Online' and 'Console' tabs. On the right, there are icons for 'Cookies', 'Desktop Agent', 'Runner', 'Trash', and a help icon.

Student

Create Student

The screenshot shows the Postman interface with a workspace named "My Workspace". The "API Network" tab is selected, showing a collection of API endpoints. The "POST Create Student" endpoint is selected, and the request is configured with the URL "localhost:8080/api/student". The request body is a JSON object with the following fields: "npm" (202310017), "firstname" (novi), "middlename" (Widia), "lastname" (Damayanti), "program_id" (1), and "department_id" (1). The response is a 200 OK status with a response time of 86 ms and a body containing a JSON object with "result": true, "message": [], and "data" containing an array of student objects.

```
POST localhost:8080/api/student
```

```
{  "npm": "202310017",  "firstname": "novi",  "middlename": "Widia",  "lastname": "Damayanti",  "program_id": 1,  "department_id": 1}
```

```
{  "result": true,  "message": [],  "data": [    {      "id": 6,      "npm": "202310017",      "firstname": "novi",      "middlename": "Widia",      "program_id": 1,      "department_id": 1,      "lastname": "Damayanti"    }  ]}
```

Get All Student

The screenshot shows the Postman interface with a workspace named "My Workspace". The "API Network" tab is selected, showing a collection of API endpoints. The "GET All Student" endpoint is selected, and the request is configured with the URL "localhost:8080/api/student". The response is a 200 OK status with a response time of 17 ms and a body containing an array of student objects.

```
GET localhost:8080/api/student
```

```
{  "id": 1,  "npm": "",  "firstname": "",  "middlename": "Widia",  "program_id": 1,  "department_id": 1,  "lastname": null}, {  "id": 2,  "npm": null,  "firstname": null,  "middlename": "Widia",  "program_id": 1,  "department_id": 1,  "lastname": null}, {  "id": 3,  "npm": null,  "firstname": null,  "middlename": "Widia",  "program_id": 1,  "department_id": 1,  "lastname": null}, {  "id": 4,  "npm": null,  "firstname": null,  "middlename": "Widia",  "program_id": 1,  "department_id": 1,  "lastname": null}, {  "id": 5,  "npm": null,  "firstname": null,  "middlename": "Widia",  "program_id": 1,  "department_id": 1,  "lastname": null}
```

Get Student By Id

The screenshot shows the Postman interface with a GET request to `localhost:8080/api/student/1`. The response is a JSON object with the following structure:

```
{
  "id": 1,
  "npm": "",
  "firstname": "",
  "middlename": "Widia",
  "program_id": 1,
  "department_id": 1,
  "lastname": null
}
```

The status bar indicates a 200 OK response with 18 ms latency and 265 B of data.

Update Student

The screenshot shows the Postman interface with a POST request to `localhost:8080/api/student`. The request body is a JSON object with the following structure:

```
{
  "id": 1,
  "npm": "202310017",
  "firstname": "Novi",
  "middlename": "Widia",
  "lastname": "Damayanti",
  "program_id": 1,
  "department_id": 1
}
```

The response is a JSON object with the following structure:

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

The status bar indicates a 200 OK response with 37 ms latency and 321 B of data.

Delete Student By Id

The screenshot displays the Postman application interface. At the top, the navigation bar includes 'Home', 'Workspaces', 'API Network', and 'Explore'. A search bar labeled 'Search Postman' is present. On the right, there are buttons for 'Invite', settings, notifications, and 'Upgrade'.

The left sidebar, titled 'My Workspace', shows a tree view of API collections. The 'Student' collection is expanded, listing several endpoints. The 'DELETE Student By ID' endpoint is selected.

The main workspace area shows the details of the selected endpoint:

- Method:** DELETE
- URL:** localhost:8080/api/student/1
- Params:** A table for Query Params is visible, with columns for KEY, VALUE, and DESCRIPTION. The table is currently empty.
- Body:** The body tab is selected, showing a single line of text: '1'.
- Response:** The response status is 200 OK, with a response time of 49 ms and a size of 123 B. The response is displayed in the 'Pretty' format.

At the bottom of the interface, there is a status bar with icons for 'Online', 'Console', 'Cookies', 'Desktop Agent', 'Runner', 'Trash', and a help icon.