

## Validation & Response Data

Nama : Fajar Alfiantino

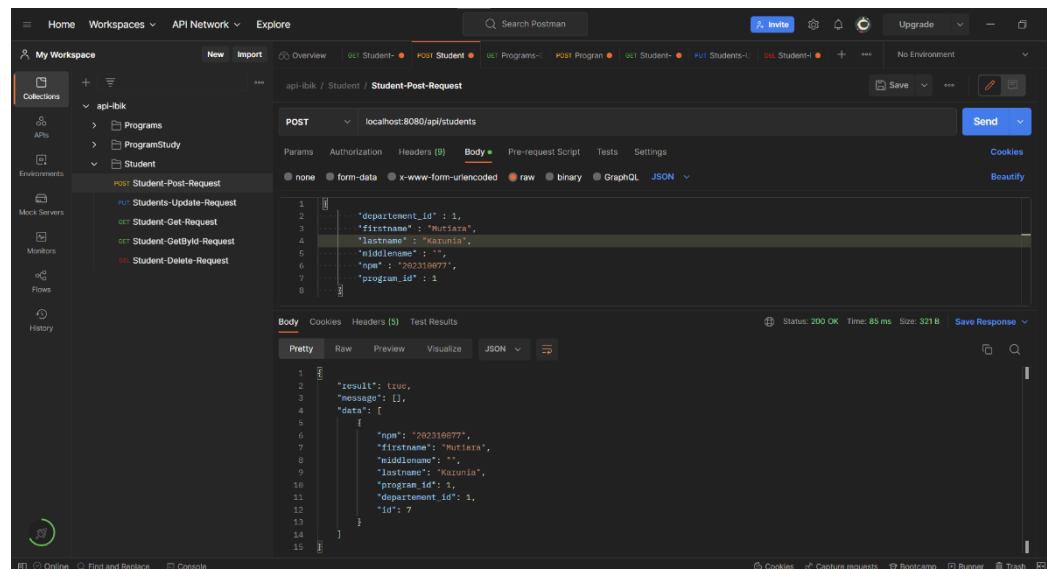
NPM : 202310072

Kelas : TI-20-A Pagi

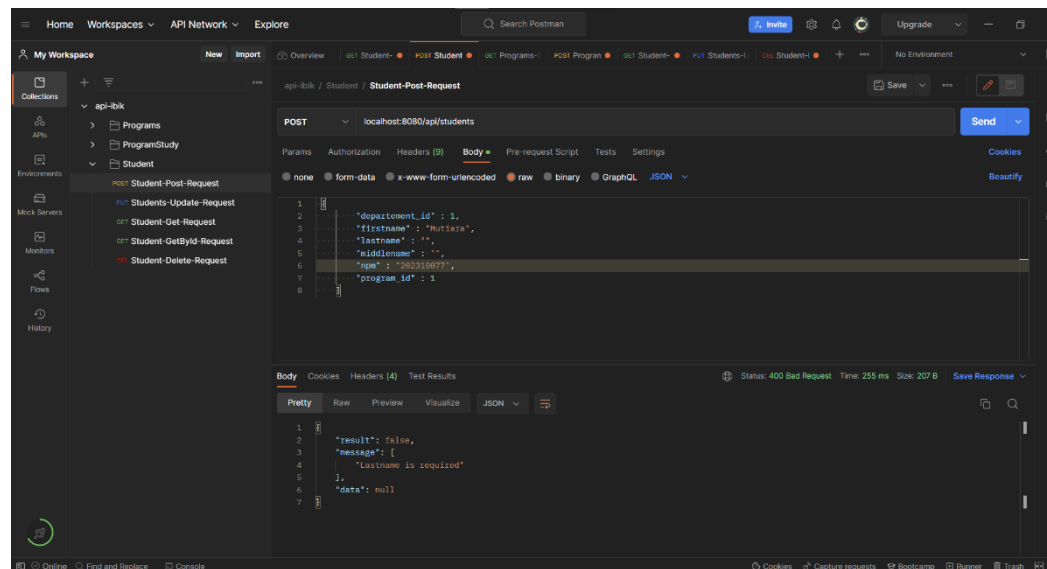
Mata Kuliah : Microservices

### Tabel Student

#### Create: True



#### Create: Error



## Read: True

The screenshot shows the Postman interface with a workspace named 'api-ibk'. The 'Student-Get-Request' is selected in the left sidebar. The request is a GET to 'localhost:8080/api/students'. The response is a 200 OK with a time of 106 ms and a size of 568 B. The JSON body contains two student records.

KEY	VALUE	DESCRIPTION
Key	Value	Description

```
1 {
2   "result": true,
3   "message": [],
4   "data": [
5     {
6       "npe": "262316972",
7       "firstname": "Rajet",
8       "middlename": null,
9       "lastname": "Alliantino",
10      "program_id": 1,
11      "department_id": 1,
12      "id": 1
13    },
14    {
15      "npe": "262316915",
16      "firstname": "Kelisha",
17      "middlename": "Annalis",
18      "lastname": "Salsabila",
19      "program_id": 1,
20      "department_id": 1,
21      "id": 2
22    }
23  ]
24 }
```

## Read ById: True

The screenshot shows the Postman interface with a workspace named 'api-ibk'. The 'Student-GetById-Request' is selected in the left sidebar. The request is a GET to 'localhost:8080/api/students/1'. The response is a 200 OK with a time of 44 ms and a size of 324 B. The JSON body contains one student record.

KEY	VALUE	DESCRIPTION
Key	Value	Description

```
1 {
2   "result": true,
3   "message": [],
4   "data": [
5     {
6       "npe": "262316972",
7       "firstname": "Rajet",
8       "middlename": null,
9       "lastname": "Alliantino",
10      "program_id": 1,
11      "department_id": 1,
12      "id": 1
13    }
14  ]
15 }
```

## Read ById: False

api-ibk / Student / Student-GetById-Request

GET localhost:8080/api/students/99

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (4) Test Results

Status: 400 Bad Request Time: 18 ms Size: 203 B Save Response

Pretty Raw Preview Visualize JSON

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

## Update: True

api-ibk / Student / Students-Update-Request

PUT localhost:8080/api/students/

Params Authorization Headers (5) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 32 ms Size: 318 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": true,
3   "message": [],
4   "data": [
5     {
6       "age": "202310036",
7       "firstname": "Amanda",
8       "middlename": "",
9       "lastname": "Nails",
10      "program_id": 1,
11      "department_id": 1,
12      "id": 5
13    }
14  ]
15 }
```

## Update: Error

The screenshot shows a Postman interface with a PUT request to `localhost:8080/api/students/`. The request body is a JSON object: `{ "departement_id": 1, "firstname": "Amanda", "lastname": "Naila", "age": "20031005", "program_id": 1 }`. The response status is `400 Bad Request` with a time of `44 ms` and size of `201 B`. The response body is a JSON object: `{ "result": false, "message": [ "ID is Required" ], "data": null }`.

```
1 {
2   "departement_id": 1,
3   "firstname": "Amanda",
4   "lastname": "Naila",
5   "age": "20031005",
6   "program_id": 1
7 }
```

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

## Delete: True

The screenshot shows a Postman interface with a DELETE request to `localhost:8080/api/students/6`. The response status is `200 OK` with a time of `90 ms` and size of `225 B`. The response body is a JSON object: `{ "result": true, "message": [ "Successfully Remove" ], "data": null }`.

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

## Delete: Error

The screenshot shows a Postman interface with a workspace named 'My Workspace'. The left sidebar shows a collection 'api-ibik' with a folder 'Programs' containing a 'Student' folder. The 'Student' folder contains several requests, including 'Student-Delete-Request'. The main panel shows the 'Student-Delete-Request' selected, which is a DELETE request to 'localhost:8080/api/students/9'. The response is a 400 Bad Request with a status of 400, time of 77 ms, and size of 273 B. The response body is JSON: { "result": false, "message": [ "No class com.ibik.academic.academicservices.students.Students entity with id 9 exists!" ], "data": null }.

```
DELETE localhost:8080/api/students/9
```

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body: { "result": false, "message": [ "No class com.ibik.academic.academicservices.students.Students entity with id 9 exists!" ], "data": null }

## Tabel Programs

### Create: True

The screenshot shows a Postman interface with a workspace named 'My Workspace'. The left sidebar shows a collection 'api-ibik' with a folder 'Programs' containing a 'Programs-Post-Request'. The main panel shows the 'Programs-Post-Request' selected, which is a POST request to 'localhost:8080/api/programs'. The request body is JSON: { "name": "Doctor", "description": "53", "is\_active": true }. The response is a 200 OK with a status of 200, time of 44 ms, and size of 262 B. The response body is JSON: { "result": true, "message": [], "data": [ { "id": 3, "name": "Doctor", "description": "53", "is\_active": true } ] }.

```
POST localhost:8080/api/programs
```

Body: { "name": "Doctor", "description": "53", "is\_active": true }

Body: { "result": true, "message": [], "data": [ { "id": 3, "name": "Doctor", "description": "53", "is\_active": true } ] }

## Create: Error

The screenshot shows a Postman interface with a workspace named 'My Workspace'. The left sidebar shows a collection 'api-ibk' with a folder 'Programs' containing several requests. The main panel shows a 'POST' request to 'localhost:8080/api/programs'. The request body is a JSON object: 

```
{  "name": "",  "description": "S1",  "is_active": true}
```

. The response status is '400 Bad Request' with a time of '39 ms' and size of '203 B'. The response body is a JSON object: 

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

.

## Read: True

The screenshot shows a Postman interface with a workspace named 'My Workspace'. The left sidebar shows a collection 'api-ibk' with a folder 'Programs' containing several requests. The main panel shows a 'GET' request to 'localhost:8080/api/programs'. The response status is '200 OK' with a time of '19 ms' and size of '400 B'. The response body is a JSON object: 

```
{  "result": true,  "message": [],  "data": [    {      "id": 1,      "name": "Bachelor Degree",      "description": "S1",      "is_active": true    },    {      "id": 2,      "name": "Master Degree",      "description": "S2",      "is_active": true    },    {      "id": 3,      "name": "Doctor",      "description": "S3",      "is_active": true    }  ]}
```

.

## Read ById: True

The screenshot shows the Postman interface with a workspace named 'My Workspace'. The left sidebar shows a collection 'api-ibk' with a sub-collection 'Programs'. The 'Programs-GetById-Request' is selected. The main panel shows the request details for a GET request to 'localhost:8080/api/programs/3'. The status is '200 OK' and the response is a JSON object.

Request Details:

- Method: GET
- URL: localhost:8080/api/programs/3
- Status: 200 OK
- Time: 23 ms
- Size: 262 B

Response Body (Pretty):

```
1 {
2   "result": true,
3   "message": [],
4   "data": {
5     "id": 3,
6     "name": "Doctor",
7     "description": "53",
8     "is_active": true
9   }
10 }
11
12 }
```

## Read ById: Error

The screenshot shows the Postman interface with the same workspace and collection. The 'Programs-GetById-Request' is selected. The main panel shows the request details for a GET request to 'localhost:8080/api/programs/5'. The status is '400 Bad Request' and the response is a JSON object indicating an error.

Request Details:

- Method: GET
- URL: localhost:8080/api/programs/5
- Status: 400 Bad Request
- Time: 24 ms
- Size: 203 B

Response Body (Pretty):

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

## Update: True

The screenshot shows the Postman interface with a PUT request to `localhost:8080/api/programs`. The request body is a JSON object: `{ "id": 3, "name": "Doctor Degree", "description": "53", "is_active": true }`. The response status is `200 OK` with a time of `62 ms` and size of `269 B`. The response body is a JSON object: `{ "result": true, "message": [], "data": { "id": 3, "name": "Doctor Degree", "description": "53", "is_active": true } }`.

## Update: False

The screenshot shows the Postman interface with a PUT request to `localhost:8080/api/programs`. The request body is a JSON object: `{ "name": "Doctor", "description": "53", "is_active": true }`. The response status is `400 Bad Request` with a time of `34 ms` and size of `201 B`. The response body is a JSON object: `{ "result": false, "message": [ { "id is Required" } ], "data": null }`.



## Delete: True

The screenshot shows the Postman interface with a DELETE request to `localhost:8080/api/programs/3`. The response status is 200 OK, and the body contains a JSON object indicating successful deletion.

KEY	VALUE	DESCRIPTION
Key	Value	Description

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

## Delete: False

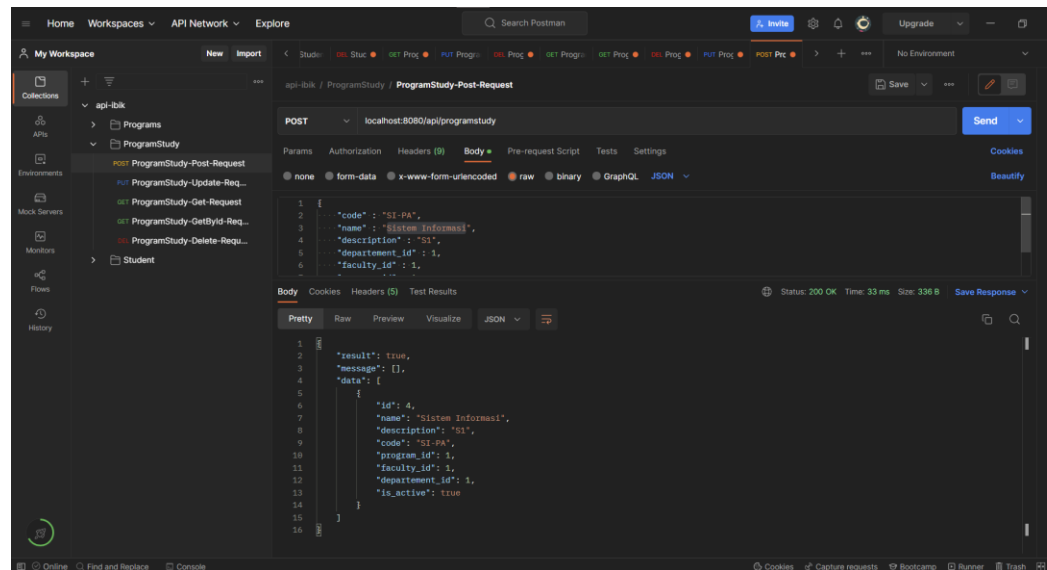
The screenshot shows the Postman interface with a DELETE request to `localhost:8080/api/programs/5`. The response status is 400 Bad Request, and the body contains a JSON object indicating that the entity does not exist.

KEY	VALUE	DESCRIPTION
Key	Value	Description

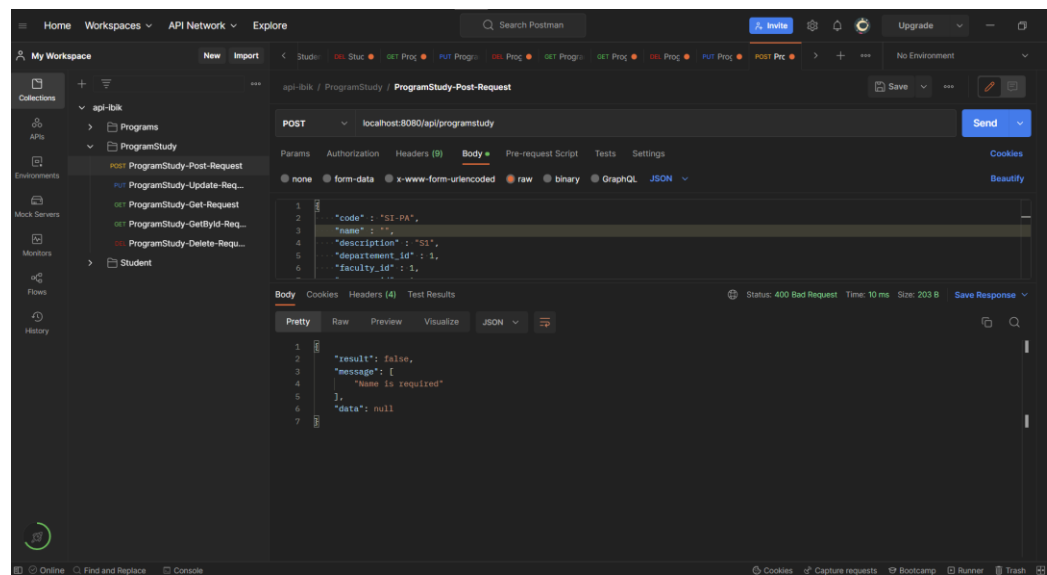
```
1 {
2   "result": false,
3   "message": [
4     "No class com.ibik.academic.academiceservices.programs.Programs entity with id 5 exists:"
5   ],
6   "data": null
7 }
```

## Tabel ProgramStudy

### Create: True



### Create: Error



## Read: True

The screenshot shows the Postman interface with a GET request to `localhost:8080/api/programstudy`. The response status is 200 OK, and the body is displayed in JSON format. The response contains a list of two program entries.

```
1 {
2   "result": true,
3   "message": [],
4   "data": [
5     {
6       "id": 1,
7       "name": "Teknologi Informasi",
8       "description": "SI",
9       "code": "TI-PA",
10      "program_id": 1,
11      "faculty_id": 1,
12      "department_id": 1,
13      "is_active": true
14    },
15    {
16      "id": 4,
17      "name": "Sistem Informasi",
18      "description": "SI",
19      "code": "SI-PA",
20      "program_id": 1,
21      "faculty_id": 1,
22      "department_id": 1,
23      "is_active": true
24    }
25  ]
26 }
```

## Read ById: True

The screenshot shows the Postman interface with a GET request to `localhost:8080/api/programstudy/1`. The response status is 200 OK, and the body is displayed in JSON format. The response contains a single program entry with ID 1.

```
1 {
2   "result": true,
3   "message": [],
4   "data": [
5     {
6       "id": 1,
7       "name": "Teknologi Informasi",
8       "description": "SI",
9       "code": "TI-PA",
10      "program_id": 1,
11      "faculty_id": 1,
12      "department_id": 1,
13      "is_active": true
14    }
15  ]
16 }
```

## Read ById: Error

The screenshot shows a Postman interface with a GET request to `localhost:8080/api/programstudy/2`. The response status is **400 Bad Request** with a time of 20 ms and size of 203 B. The response body is a JSON object:

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

## Update: True

The screenshot shows a Postman interface with a PUT request to `localhost:8080/api/programstudy`. The response status is **200 OK** with a time of 34 ms and size of 339 B. The request body is a JSON object:

```
1 {
2   "id": 1,
3   "code": "TI-PA",
4   "name": "Teknologi Informasi",
5   "description": "SI",
6   "department_id": 1,
7 }
8
```

The response body is a JSON object:

```
1 {
2   "result": true,
3   "message": [],
4   "data": [
5     {
6       "id": 1,
7       "name": "Teknologi Informasi",
8       "description": "SI",
9       "code": "TI-PA",
10      "program_id": 1,
11      "faculty_id": 1,
12      "department_id": 1,
13      "is_active": true
14    }
15  ]
16 }
```

## Update: Error

The screenshot shows a Postman interface with a PUT request to `localhost:8080/api/programstudy`. The request body is a JSON object: `{ "code": "TI-PA", "name": "Teknologi Informasi", "description": "SI", "department_id": 1, "faculty_id": 1 }`. The response status is `400 Bad Request` with a time of `27 ms` and size of `201 B`. The response body is: `{ "result": false, "message": [ { "ID is Required" } ], "data": null }`.

## Delete: True

The screenshot shows a Postman interface with a DELETE request to `localhost:8080/api/programstudy/3`. The response status is `200 OK` with a time of `40 ms` and size of `225 B`. The response body is: `{ "result": true, "message": [ { "Successfully Remove" } ], "data": null }`.

## Delete: False

The screenshot shows the Postman interface with a DELETE request to `localhost:8080/api/programstudy/2`. The response is a 400 Bad Request with the following JSON body:

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

The status bar at the bottom indicates the response is a 400 Bad Request, took 10 ms, and has a size of 281 B.