

“TUGAS 6 Mikroservices”



Dibuat Oleh :

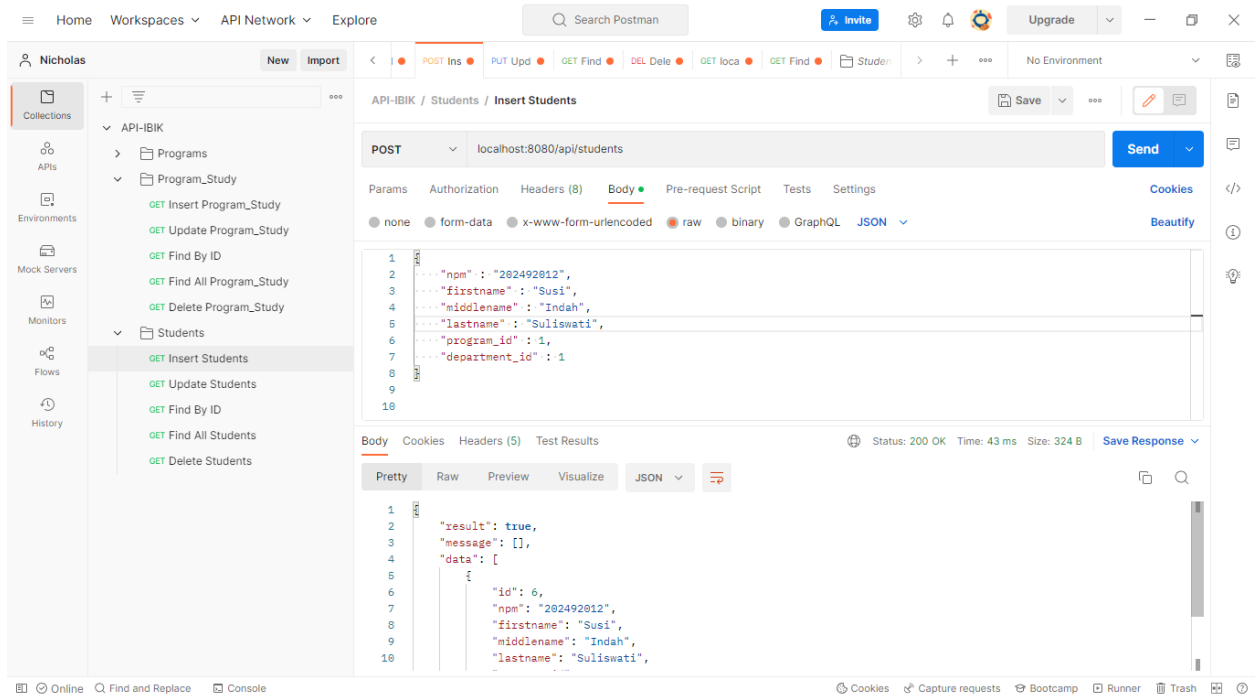
Nicholas Fico (202310077)

TI – 20 – PA

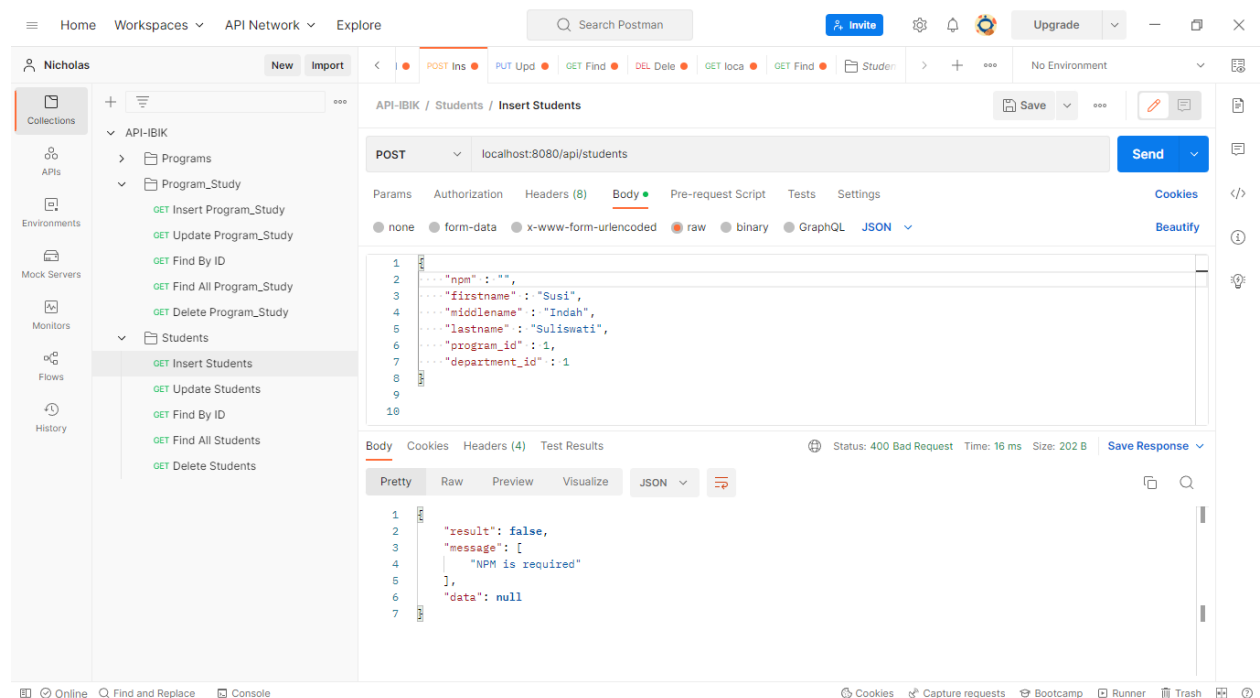
Hasil Screenshoot :

1. Students

- Insert Success



Error



- Update Success

The screenshot shows the Postman interface with a PUT request to `localhost:8080/api/students`. The request body is a JSON object representing a student. The response status is 200 OK, and the response body is a JSON object indicating success.

Request:

```
PUT localhost:8080/api/students
```

Body (JSON):

```
{  "id": 6,  "npm": "20231000",  "firstname": "Susi",  "middlename": "Indah",  "lastname": "Sekali",  "program_id": 2,  "department_id": 2}
```

Response:

```
{  "result": true,  "message": [],  "data": [    {      "id": 6,      "npm": "20231000",      "firstname": "Susi",      "middlename": "Indah",      "lastname": "Sekali"    }  ]}
```

Error

The screenshot shows the Postman interface with a PUT request to `localhost:8080/api/students`. The request body is a JSON object representing a student. The response status is 400 Bad Request, and the response body is a JSON object indicating an error.

Request:

```
PUT localhost:8080/api/students
```

Body (JSON):

```
{  "npm": "20231000",  "firstname": "Susi",  "middlename": "Indah",  "lastname": "Sekali",  "program_id": 2,  "department_id": 2}
```

Response:

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

- FindByID

Success

The screenshot shows the Postman interface with a collection named 'API-IBIK' containing a folder 'Students'. The 'Find By ID' endpoint is selected. The request is a GET to 'localhost:8080/api/students/6'. The response is a JSON object with a 'result' of true, an empty 'message' array, and a 'data' array containing a single student object with ID 6, name 'Susi Indah Sekali', and a timestamp '20231000'.

KEY	VALUE	DESCRIPTION
Key	Value	Description

```
1 {
2   "result": true,
3   "message": [],
4   "data": [
5     {
6       "id": 6,
7       "npm": "20231000",
8       "firstname": "Susi",
9       "middlename": "Indah",
10      "lastname": "Sekali",
11    }
12  ]
13 }
```

Error

The screenshot shows the Postman interface with the same 'Find By ID' endpoint. The response is a 500 Internal Server Error. The response body contains a stack trace and a message: 'No value present'.

```
(AbstractProcessorLight.java:65)\r\n\tat org.apache.coyote.AbstractProtocol$ConnectionHandler.process
(AbstractProtocol.java:890)\r\n\tat org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.
java:1789)\r\n\tat org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:49)\r\n\tat
org.apache.tomcat.util.threads.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1191)\r\n\tat org.apache.
tomcat.util.threads.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:659)\r\n\tat org.apache.tomcat.util.
threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)\r\n\tat java.base/java.lang.Thread.run(Thread.
java:1589)\r\n",
"message": "No value present",
"path": "/api/students/0"
```

- Find All

Success

The screenshot shows the Postman interface with a collection named 'API-IBIK' and a folder 'Students'. The selected endpoint is 'GET Find All Students' with the URL 'localhost:8080/api/students/'. The request is a GET method. The response is a 200 OK status with a response time of 27 ms and a size of 731 B. The response body is in JSON format, showing a successful result with a list of student data.

API-IBIK / Students / Find All Students

GET localhost:8080/api/students/

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

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (5) Test Results Status: 200 OK Time: 27 ms Size: 731 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": true,
3   "message": [],
4   "data": [
5     {
6       "id": 1,
7       "npm": "202310077",
8       "firstname": "Nicholas",
9       "middlename": "Zord",
10      "lastname": "Fico",
11    }
12  ]
13 }
```

Error

The screenshot shows the Postman interface with the same collection and folder. The selected endpoint is 'GET Find All Students' with the URL 'localhost:8080/api/students/9'. The request is a GET method. The response is a 400 Bad Request status with a response time of 143 ms and a size of 203 B. The response body is in JSON format, showing an error result with a message 'No value present' and null data.

API-IBIK / Students / Find All Students

GET localhost:8080/api/students/9

Params Authorization Headers (6) 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: 143 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 }
```

- Delete Success

The screenshot shows the Postman interface with a workspace named 'Nicholas'. The left sidebar shows a collection of API endpoints under 'API-IBIK' > 'Programs' > 'Program_Study'. The selected endpoint is 'DELETE /api/students/6'. The request is configured with the method 'DELETE' and the URL 'localhost:8080/api/students/6'. The response is a 200 OK status with a response time of 49 ms and a size of 225 B. The response body is displayed in the 'Pretty' view, showing a JSON object with 'result': true, 'message': ['Successfully Remove'], and 'data': null.

API-IBIK / Students / Delete Students

DELETE localhost:8080/api/students/6

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

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (5) Test Results Status: 200 OK Time: 49 ms Size: 225 B Save Response

Pretty Raw Preview Visualize JSON

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

Error

The screenshot shows the Postman interface with a workspace named 'Nicholas'. The left sidebar shows a collection of API endpoints under 'API-IBIK' > 'Programs' > 'Program_Study'. The selected endpoint is 'DELETE /api/students/8'. The request is configured with the method 'DELETE' and the URL 'localhost:8080/api/students/8'. The response is a 400 Bad Request status with a response time of 127 ms and a size of 273 B. The response body is displayed in the 'Pretty' view, showing a JSON object with 'result': false, 'message': ['No class com.ibik.academic.academicservices.Students.Students entity with id 8 exists!'], and 'data': null.

API-IBIK / Students / Delete Students

DELETE localhost:8080/api/students/8

Params Authorization Headers (6) 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: 127 ms Size: 273 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": false,
3   "message": [
4     "No class com.ibik.academic.academicservices.Students.Students entity with id 8 exists!"
5   ],
6   "data": null
7 }
```

2. Programs

- Insert
Success

The screenshot shows the Postman interface with a collection named 'API-IBIK' containing a folder 'Programs'. Inside 'Programs', there are several GET requests and two folders: 'Program_Study' and 'Students'. The 'POST Insert Programs' request is selected. The request is a POST to 'localhost:8080/api/programs' with a JSON body:

```
{  "id": 1,  "name": "Doctor",  "description": "S3",  "is_active": 1}
```

. The response status is 200 OK, with a time of 757 ms and a size of 262 B. The response body is a JSON object:

```
{  "result": true,  "message": [],  "data": [    {      "id": 7,      "name": "Doctor",      "description": "S3",      "is_active": true    }  ]}
```

Error

The screenshot shows the same Postman interface, but the response status is 400 Bad Request, with a time of 56 ms and a size of 203 B. The response body is a JSON object:

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

- Update Success

The screenshot shows the Postman interface with a PUT request to `localhost:8080/api/programs`. The request body is a JSON object: `{ "id": 4, "name": "Bachelor Degree", "description": "S1", "is_active": true }`. The response status is `200 OK` with a time of `281 ms` and size of `271 B`. The response body is a JSON object: `{ "result": true, "message": [], "data": { "id": 4, "name": "Bachelor Degree", "description": "S1", "is_active": true } }`.

Error

The screenshot shows the Postman interface with a PUT request to `localhost:8080/api/programs`. The request body is a JSON object: `{ "name": "Bachelor Degree", "description": "S1", "is_active": true }`. The response status is `400 Bad Request` with a time of `33 ms` and size of `201 B`. The response body is a JSON object: `{ "result": false, "message": ["ID is required"], "data": null }`.

- Find By ID Success

The screenshot shows the Postman interface with a workspace named 'Nicholas'. The left sidebar shows a collection 'API-IBIK' with a sub-collection 'Programs'. The 'Find By ID' endpoint is selected. The request is a GET to 'localhost:8080/api/programs/7'. The response is a 200 OK with a JSON body:

```
1 {
2   "result": true,
3   "message": null,
4   "data": [
5     {
6       "id": 7,
7       "name": "Doctor",
8       "description": "S3",
9       "is_active": true
10    }
11  ]
12 }
```

The status bar at the bottom shows 'Status: 200 OK Time: 123 ms Size: 264 B'.

Error

The screenshot shows the Postman interface with the same workspace 'Nicholas'. The 'Find By ID' endpoint is selected, but the request is to 'localhost:8080/api/programs/5'. The response is a 400 Bad Request with a JSON body:

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

The status bar at the bottom shows 'Status: 400 Bad Request Time: 22 ms Size: 203 B'.

- Find All Success

The screenshot shows the Postman interface with a collection named 'API-IBIK' expanded to show a folder 'Programs'. Inside 'Programs', the request 'GET Find All Programs' is selected. The request is a GET method to 'localhost:8080/api/programs'. The response status is '200 OK' with a time of '103 ms' and a size of '545 B'. The response body is displayed in JSON format:

```
1 {
2   "result": true,
3   "message": null,
4   "data": [
5     {
6       "id": 2,
7       "name": "Doctoralll Degree",
8       "description": "S2++",
9       "is_active": true
10    }
11  ]
12 }
```

Error

The screenshot shows the Postman interface with the same collection 'API-IBIK' and folder 'Programs'. The request 'GET Find All Programs' is selected, but the URL is changed to 'localhost:8080/api/programs/5'. The response status is '400 Bad Request' with a time of '28 ms' and a size of '203 B'. The response body is displayed in JSON format:

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

- Delete Success

The screenshot shows the Postman interface with a DELETE request to `localhost:8080/api/programs/2`. The request is successful, returning a 200 OK status. The response body is displayed in JSON format:

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

The interface includes a sidebar with collections, a top navigation bar, and a bottom status bar.

Error

The screenshot shows the same DELETE request to `localhost:8080/api/programs/2`, but it has failed with a 400 Bad Request status. The response body is displayed in JSON format:

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

The interface includes a sidebar with collections, a top navigation bar, and a bottom status bar.

3. Program_Study

- Insert
Success

The screenshot shows the Postman interface with a collection named 'API-IBIK' containing a folder 'Program_Study'. Inside this folder, there is a POST request named 'Insert Program_Study'. The request is configured with the URL 'localhost:8080/api/program_study' and the 'Body' tab is selected. The body content is a JSON object:

```
1 {
2   "id": 4,
3   "name": "Pensaran Digital",
4   "description": "S1",
5   "code": "TIPD",
6   "program_id": 2,
7   "faculty_id": 1,
8   "department_id": 1,
9   "is_active": 1
10 }
```

The response is a 200 OK status with a response time of 211 ms and a size of 334 B. The response body is a JSON object:

```
1 {
2   "result": true,
3   "message": [],
4   "data": [
5     {
6       "id": 5,
7       "name": "Pensaran Digital",
8       "description": "S1",
9       "code": "TIPD",
10      "program_id": 2,
11      "faculty_id": 1,
12      "department_id": 1,
13      "is_active": 1
14     }
15   ]
16 }
```

Error

The screenshot shows the Postman interface with the same collection and request as the previous image. However, the response is a 400 Bad Request status with a response time of 14 ms and a size of 203 B. The response body is a JSON object:

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

- Update Success

The screenshot shows the Postman interface with a PUT request to `localhost:8080/api/program_study`. The request body is a JSON object with the following fields: `id` (1), `name` ("Pemograman Perangkat Bergerak"), `description` ("S1"), `code` ("TIPPB"), `program_id` (1), `faculty_id` (1), `department_id` (1), and `is_active` (1). The response status is 200 OK, and the response body is a JSON object with `result` (true), `message` ([]), and `data` (an object with the same fields as the request body).

Error

The screenshot shows the Postman interface with a PUT request to `localhost:8080/api/program_study`. The request body is a JSON object with the following fields: `name` ("Pemograman Perangkat Bergerak"), `description` ("S1"), `code` ("TIPPB"), `program_id` (1), `faculty_id` (1), `department_id` (1), and `is_active` (1). The response status is 400 Bad Request, and the response body is a JSON object with `result` (false), `message` (["ID is required"]), and `data` (null).

- Find By ID
Success

The screenshot shows the Postman interface with a collection named 'API-IBIK' expanded to 'Program_Study'. The selected item is 'GET Find By ID'. The request is a GET to 'localhost:8080/api/program_study/1'. The response status is '200 OK' with a time of '37 ms' and size of '350 B'. The response body is JSON:

```
1 {
2   "result": true,
3   "message": null,
4   "data": [
5     {
6       "id": 1,
7       "name": "Pemograman Perangkat Bergerak",
8       "description": "S1",
9       "code": "TIPPB",
10      "program_id": 1,
11    }
12  ]
13 }
```

Error

The screenshot shows the Postman interface with the same collection and item. The request is now a GET to 'localhost:8080/api/program_study/7'. The response status is '400 Bad Request' with a time of '37 ms' and size of '203 B'. The response body is JSON:

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

- Find All Success

API-IBIK / Program_Study / Find All Program_Study

GET localhost:8080/api/program_study

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

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (5) Test Results Status: 200 OK Time: 111 ms Size: 958 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": true,
3   "message": null,
4   "data": [
5     {
6       "id": 1,
7       "name": "Pemograman Perangkat Bergerak",
8       "description": "SI",
9       "code": "TIPPB",
10      "program_id": 1,
11    }
12  ]
13 }
```

Error

API-IBIK / Program_Study / Find All Program_Study

GET localhost:8080/api/program_study/7

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

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 50 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 }
```

- Delete Success

The screenshot shows the Postman interface with a DELETE request to `localhost:8080/api/program_study/4`. The request is successful, returning a 200 OK status. The response body is displayed in JSON format:

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

Error

The screenshot shows the Postman interface with a DELETE request to `localhost:8080/api/program_study/5`. The request fails with a 400 Bad Request status. The response body is displayed in JSON format:

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