

Nama : Yudhistira Kusuma
NPM : 202310067
Kelas : TI-20-PA
Matkul : Teknologi Microservices dan Web Service

Tabel Program Study

Insert:

The screenshot displays a web application interface on the left and a Postman API client on the right. The web application shows a table with columns: id, code, department_id, description, faculty_id, is_active, name, and program_id. The table contains one row with the following data: id: 1, code: 5, department_id: TI-PA, description: Teknologi Informasi, faculty_id: 23, is_active: 1, name: Teknologi Informasi, and program_id: 20. The Postman interface shows a POST request to the endpoint localhost:8080/api/program_study. The request body is a JSON object: {"name": "Teknologi Informasi", "description": "SI", "code": "TI-PA", "program_id": 20, "faculty_id": 23, "department_id": 100, "is_active": 1}. The response is a 200 OK status with a response body: {"id": 6, "name": "Teknologi Informasi", "description": "SI", "code": "TI-PA", "program_id": 20, "faculty_id": 23, "department_id": 100, "is_active": true}.

Update:

The screenshot displays a web application interface on the left and a Postman API client on the right. The web application shows a table with columns: id, code, department_id, description, faculty_id, is_active, name, and program_id. The table contains three rows with the following data: id: 1, code: 5, department_id: TI-PA, description: Teknologi Informasi, faculty_id: 23, is_active: 1, name: Teknologi Informasi, and program_id: 20; id: 2, code: 6, department_id: SI-PA, description: Sistem Informasi, faculty_id: 23, is_active: 1, name: Sistem Informasi, and program_id: 21; id: 3, code: 7, department_id: PR-PA, description: Perilaku, faculty_id: 23, is_active: 1, name: Perilaku, and program_id: 22. The Postman interface shows a PUT request to the endpoint localhost:8080/api/program_study. The request body is a JSON object: {"id": 7, "name": "Perilaku", "description": "SI", "code": "PR-PA", "program_id": 22, "faculty_id": 23, "department_id": 100, "is_active": 1}. The response is a 200 OK status with a response body: {"id": 7, "name": "Perilaku", "description": "SI", "code": "PR-PA", "program_id": 22, "faculty_id": 23, "department_id": 100, "is_active": true}.

Find All:

The screenshot displays a web application interface on the left and a Postman API client on the right. The web application shows a table of programs with columns: id, code, department_id, description, faculty_id, is_active, name, and program_id. The table contains 25 rows of data. The Postman interface shows the 'Find All Programs' endpoint (GET localhost:8080/api/program_study) with a response body containing a list of program details.

id	code	department_id	description	faculty_id	is_active	name	program_id
1	TI-PA	100	1	23	1	Teknologi Informasi	20
2	SI-PA	100	1	23	1	Sistem Informasi	21
3	PA-PA	100	1	23	1	Parkinson	22
4	SI-PA	100	1	23	1	Sistem Informasi	23

Postman Response Body (JSON):

```
{
  "id": 5,
  "name": "Teknologi Informasi",
  "description": "SI",
  "code": "TI-PA",
  "program_id": 20,
  "faculty_id": 23,
  "department_id": 100,
  "is_active": true
}
```

Find By Id:

The screenshot displays a web application interface on the left and a Postman API client on the right. The web application shows a table of programs with columns: id, code, department_id, description, faculty_id, is_active, name, and program_id. The table contains 25 rows of data. The Postman interface shows the 'Find Programs By Id' endpoint (GET localhost:8080/api/program_study/6) with a response body containing a single program detail.

id	code	department_id	description	faculty_id	is_active	name	program_id
1	TI-PA	100	1	23	1	Teknologi Informasi	20
2	SI-PA	100	1	23	1	Sistem Informasi	21
3	PA-PA	100	1	23	1	Parkinson	22
4	SI-PA	100	1	23	1	Sistem Informasi	23

Postman Response Body (JSON):

```
{
  "id": 6,
  "name": "Sistem Informasi",
  "description": "SI",
  "code": "SI-PA",
  "program_id": 21,
  "faculty_id": 23,
  "department_id": 100,
  "is_active": true
}
```

Delete:

The screenshot displays a web application interface on the left and a Postman API client on the right. The web application shows a table with columns: id, code, department_id, description, faculty_id, is_active, name, and program_id. The table contains three rows of data. The Postman interface shows a DELETE request to the endpoint `localhost:8080/api/program_study/8`. The response is a 200 OK status with a response time of 15 ms and a body of `{}`.

Tabel Student

Insert:

The screenshot displays a web application interface on the left and a Postman API client on the right. The web application shows a table with columns: id, department_id, first_name, last_name, middle_name, npm, and program_id. The table contains one row of data. The Postman interface shows a POST request to the endpoint `localhost:8080/api/student`. The request body is a JSON object: `{ "npm": "202310067", "first_name": "Yudhistira", "middle_name": "Kusuma", "last_name": "Hermawan", "program_id": 20, "department_id": 100 }`. The response is a 200 OK status with a response time of 94 ms and a body of `{ "id": 1, "npm": "202310067", "first_name": "Yudhistira", "middle_name": "Kusuma", "last_name": "Hermawan", "program_id": 20, "department_id": 100 }`.

Update:

The screenshot displays a web application on the left and the Postman API client on the right. The web application shows a table of students with columns: id, department_id, first_name, last_name, middle_name, nym, and program_id. The table contains three rows of data. Below the table, there are options to 'Post', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'. The Postman interface shows a PUT request to 'localhost:8080/api/student/' with a JSON body containing student details. The response is a 200 OK status with a JSON body.

id	department_id	first_name	last_name	middle_name	nym	program_id
1	100	Yudhistira	Aja	Kusuma	202310067	22
2	100	Michael	Rusman	Merin	202310068	21
3	100	Irvan	Atanayah	Ricky	202310069	22

```
PUT localhost:8080/api/student/
{
  "id": 1,
  "npe": "202310067",
  "firstname": "Yudhistira",
  "middlename": "Kusuma",
  "lastname": "Aja",
  "program_id": 22,
  "department_id": 100
}
```

```
200 OK 42 ms 290 B
{
  "id": 1,
  "npe": "202310067",
  "firstname": "Yudhistira",
  "middlename": "Kusuma",
  "lastname": "Aja",
  "program_id": 22,
  "department_id": 100
}
```

Find All:

The screenshot displays the same web application on the left and the Postman API client on the right. The web application shows the same table of students. The Postman interface shows a GET request to 'localhost:8080/api/student' with a response of 200 OK and a JSON body containing the same student details as the previous screenshot.

id	department_id	first_name	last_name	middle_name	nym	program_id
1	100	Yudhistira	Aja	Kusuma	202310067	22
2	100	Michael	Rusman	Merin	202310068	21
3	100	Irvan	Atanayah	Ricky	202310069	22

```
GET localhost:8080/api/student
{
  "id": 1,
  "npe": "202310067",
  "firstname": "Yudhistira",
  "middlename": "Kusuma",
  "lastname": "Aja",
  "program_id": 22,
  "department_id": 100
}
```

```
200 OK 82 ms 546 B
{
  "id": 1,
  "npe": "202310067",
  "firstname": "Yudhistira",
  "middlename": "Kusuma",
  "lastname": "Aja",
  "program_id": 22,
  "department_id": 100
}
```

Find By Id:

The screenshot displays a web application on the left and the Postman API client on the right. The web application shows a table of students with columns: id, department_id, first_name, last_name, middle_name, nym, and program_id. The table contains three rows of data. The Postman interface shows a GET request to `localhost:8080/api/student/2` with a status of 200 OK. The response body is a JSON object representing a student with ID 2.

id	department_id	first_name	last_name	middle_name	nym	program_id
1	100	Yulvindra	Alpa	Kusuma	202310067	22
2	100	Michael	Rusewan	Merwin	202310068	21
3	100	Irvan	Adhanyah	Ricky	202310069	22

```
{  "id": 2,  "npe": "282318968",  "firstname": "Michael",  "middlename": "Merwin",  "lastname": "Rusewan",  "program_id": 21,  "department_id": 100}
```

Delete:

The screenshot displays the same web application on the left and the Postman API client on the right. The web application table now shows only two rows of data, indicating that the student with ID 2 has been deleted. The Postman interface shows a DELETE request to `localhost:8080/api/student/3` with a status of 200 OK. The response body is empty, indicating a successful deletion.

id	department_id	first_name	last_name	middle_name	nym	program_id
1	100	Yulvindra	Alpa	Kusuma	202310067	22
2	100	Michael	Rusewan	Merwin	202310068	21