

**UAS TEKNOLOGI MICROSERVICES
SOFT COPY PROYEK ACARAIN
DOKUMENTASI HASIL PROYEK**

PROJECT AKHIR

Diajukan Untuk Memenuhi Tugas Mata Kuliah
Teknologi Microservices / Web Services



Disusun Oleh:

Novi Widya Damayanti	202310017
Ryan Roshandy Sachio	202310055
Thesya Marcella	202310020

5-TI-20-PA

**INSTITUT BISNIS DAN INFORMATIKA KESATUAN
FAKULTAS INFORMATIKA DAN PARIWISATA
TEKNOLOGI INFORMASI
BOGOR
2022**

A. Events

Insert Events

The screenshot shows the Postman interface with the following details:

- Collection:** Acarain
- API:** Acarain_API
- Request Type:** POST
- URL:** localhost:8080/api/events/
- Body (JSON):**

```
1 "category": "Horor",
2 "dateStart": "2023-2-4",
3 "description": "horor ini merupakan file yang terinspirasi",
4 "location": "Bogor",
5 "organized": "MT Entertainment",
6 "tag": "#BravePeople",
7 "timeEnd": "12:00:00",
8 "timeStart": "10:00:00",
9 "title": "Kuntilanak",
10 "type": "Movie File"
```
- Status:** 200 OK
- Body Response (Pretty):**

```
1 {
2   "result": true,
3   "message": [],
4   "data": [
5     {
6       "id": 3,
7       "title": "Kuntilanak",
8       "organized": "MT Entertainment",
9       "tag": "#BravePeople",
10      "type": "Movie File",
11      "location": "Bogor",
12      "dateStart": "2023-2-4",
13      "timeEnd": "12:00:00",
14      "timeStart": "10:00:00",
15      "description": "horor ini merupakan file yang terinspirasi",
16      "user": null,
17      "id": 3,
18      "category": "Horor"
19   ]
20 }
```

Insert Success

The screenshot shows the Postman interface with the following details:

- Collection:** Acarain
- API:** Acarain_API
- Request Type:** POST
- URL:** localhost:8080/api/events/
- Body (JSON):**

```
1 "category": "Horor",
2 "dateStart": "2023-2-4",
3 "description": "horor ini merupakan file yang terinspirasi",
4 "location": "Bogor",
5 "organized": "MT Entertainment",
6 "tag": "#BravePeople",
7 "timeEnd": "12:00:00",
8 "timeStart": "10:00:00",
9 "title": "Kuntilanak",
10 "type": "Movie File"
```
- Status:** 400 Bad Request
- Body Response (Pretty):**

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

Insert Error

Find All Events

Acarin API / Events / Get All Events

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
```
{
 "result": true,
 "message": null,
 "data": [
 {
 "title": "Kuntilanak",
 "organizer": "HT Entertainment",
 "tag": "#HorrorMovie",
 "type": "Movie",
 "location": "Bogor",
 "dateStart": "2023-02-04",
 "timeStart": "10:00:00",
 "dateEnd": "2023-02-06",
 "timeEnd": "12:00:00",
 "description": "Film horor ini merupakan film yang terinspirasi",
 "user": null,
 "id": 3,
 "category": "Horror"
 },
 {
 "title": "Kuntilanak",
 "organizer": "HT Entertainment",
 "tag": "null",
 "type": "Movie",
 "location": "Bogor",
 "dateStart": "2023-02-04",
 "timeStart": "10:00:00",
 "dateEnd": "2023-02-06",
 "timeEnd": "12:00:00",
 "description": "Film horor ini merupakan film yang terinspirasi",
 "user": null,
 "id": 3,
 "category": "Horror"
 }
]
}
```

```

Find All Success

Update Events

Acarin API / Events / New Request

```

PUT
localhost:8080/api/events/1
```
{
 "category": "Comedy",
 "dateStart": "2023-2-5",
 "dateEnd": "2023-2-6",
 "description": "Film comedy memberikan adegan adegan yang mengundang gelak tawa penonton",
 "location": "Jakarta",
 "organizer": "Fun Entertainment",
 "tag": "#MAMAKA",
 "timeStart": "12:00:00",
 "timeEnd": "13:00:00",
 "title": "WOKWOK",
 "type": "Movie"
}
```

```

Update Success

The screenshot shows a Postman interface with the following details:

- Collection:** Acarin
- Request Type:** PUT
- URL:** localhost:8080/api/events/1
- Body:** JSON (shown below)
- Response Status:** 404 Not Found
- Response Body:**

```

1 "status": "404",
2 "error": "Not Found",
3 "message": "No message available",
4 "path": "/api/events"
    
```

Update Error

Find By ID Events

The screenshot shows a Postman interface with the following details:

- Collection:** Acarin
- Request Type:** GET
- URL:** localhost:8080/api/events/3
- Headers:** (empty)
- Query Params:** (empty)
- Response Status:** 200 OK
- Response Body:**

```

1 "result": true,
2 "message": null,
3 "data": [
4     {
5         "id": 1,
6         "title": "WokWok",
7         "organizer": "Fun Entertainment",
8         "tag": "#Makanan",
9         "type": "Food",
10        "location": "Jakarta",
11        "dateStart": "2023-02-05",
12        "timeStart": "12:00:00",
13        "dateEnd": "2023-02-06",
14        "timeEnd": "12:00:00",
15        "description": "Film comedy memberikan adegan adegan yang mengundang gelak tawa penonton",
16        "user": null,
17        "id": 1,
18        "category": "Comedy"
19    }
20 ]
    
```

Find by Id Success

The screenshot shows the Postman application interface. On the left, the sidebar displays collections like 'Acarin' and 'Acarin API', environments, mock servers, flows, and history. The main workspace shows a request for 'GET /events/2'. The response status is 200 OK, with a JSON body containing:

```
1   {
2     "result": false,
3     "error": "No value present"
4   },
5   "data": null
```

The bottom status bar indicates: Status: 200 OK, Time: 9ms, Size: 203 B, and Save Response.

Find by Id error

Delete Events

The screenshot shows the Postman application interface. The left sidebar displays the 'Acarain' workspace with various collections like 'Acarain API', 'Events', 'Orders', 'Transactions', 'User', and 'Email'. The 'Events' collection is currently selected, showing requests for 'Post Events', 'Get All Events', 'Get by ID', and 'Delete Events by Id'. The main workspace shows a DELETE request to `localhost:8080/api/events/{id}`. The 'Body' tab is selected, showing a JSON response with a timestamp, a success message, and a null data field. The status bar at the bottom indicates a successful 200 OK response with a time of 18 ms and a size of 226 B.

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

The screenshot shows the Postman interface with the following details:

- Collection:** Acarain
- API:** Acarin_API
- Request Type:** GET
- URL:** localhost:8080/api/events/
- Body:** JSON response (Pretty)

```
1  "result": true,
2  "message": null,
3  "data": [
4    {
5      "title": "Muntulawak",
6      "organizer": "HT Entertainment",
7      "tag": null,
8      "type": "Movie Film",
9      "location": "Popri",
10     "dateStart": "2023-02-04",
11     "timeStart": "19:00:00",
12     "dateEnd": "2023-02-04",
13     "timeEnd": "19:00:00",
14     "description": "Film horor ini merupakan film yang terinspirasi",
15     "user": null,
16     "id": 1,
17     "category": "Horor"
18   }
19 ]
```

Delete Success

The screenshot shows the Postman interface with the following details:

- Collection:** Acarain
- API:** Acarin_API
- Request Type:** DELETE
- URL:** localhost:8080/api/events/2
- Body:** JSON response (Pretty)

```
1  "result": false,
2  "message": [
3    "No class com.acarin.services.acarinServices.events.Events entity with id 2 exists!"
4  ],
5  "data": null
```

Delete Error

B. Auth

Login

The screenshot shows the Postman interface with a successful API call. The URL is `localhost:8080/api/auth/login`. The request method is POST. The body contains the following JSON:

```
1 "email": "syaroshandy@gmail.com",
2 "password": "123456"
```

The response status is 200 OK, and the response body is a long token string.

Token Success

The screenshot shows the Postman interface with an error response. The URL is `localhost:8080/api/auth/login`. The request method is POST. The body contains the following JSON:

```
1 "email": "syaroshandy@gmail.com",
2 "password": ""
```

The response status is 400 Bad Request, and the response body is: `{"result": false, "data": '', "message": 'Cannot invoke \u0022com.acarainservices.acarainservices.Auth.Auth.getEmail()\u0022 because \u0022values\u0022 is null'}`.

Token Error

C. User

Insert User

The screenshot shows the Postman application interface. The left sidebar has a tree structure with 'Acarain' selected, which contains 'Acarain_API' and 'Events'. Under 'Acarain_API', there are several requests: 'POST Post Events', 'PUT New Request', 'GET Get All Events', 'GET Get by ID', 'DELETE Delete Events by Id', 'Orders', 'Transaction', and 'User'. The 'User' node is expanded, showing 'POST Post User Data', 'GET Get All', 'GET New Request', 'Email', 'Auth', 'GET User Auth Data', 'GET Token Auth Generator', and 'GET! New Request'. The 'POST Post User Data' request is selected and highlighted with a red border. The main panel shows a 'POST' request to 'localhost:8080/api/user'. The 'Body' tab is selected, showing a JSON payload:

```
1 "password": "12345678",
2 "username": "Ryan Roshandy S",
3 "email": "ryanroshandy@gmail.com",
4 "fullname": "Ryan Roshandy S",
5 "username": "ryanroshandy"
```

The response status is 'Status: 200 OK' with a 'Time: 21ms' and a 'Size: 316 B'. The response body is:

```
1 {
2   "result": true,
3   "message": [],
4   "data": [
5     {
6       "id": 1,
7       "password": "12345678",
8       "username": "Ryan Roshandy S",
9       "fullname": "Ryan Roshandy S",
10      "email": "ryanroshandy@gmail.com"
11    }
12  ]
13 }
```

At the bottom, the taskbar shows various icons and the system tray indicates it's 10:31 AM, 28°C, Berawan, and V13/03/2023.

Insert Success

This screenshot shows the same Postman interface as the previous one, but the 'User' node in the sidebar is collapsed. The 'POST Post User Data' request is selected. The response status is 'Status: 400 Bad Request' with a 'Time: 5 ms' and a 'Size: 207 B'. The response body is:

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

At the bottom, the taskbar shows various icons and the system tray indicates it's 10:32 AM, 28°C, Berawan, and V13/03/2023.

Insert Error

Find All User

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Acarin' selected, showing collections, APIs, environments, monitors, flows, and history. Under 'APIs', 'Events' is expanded, and 'User' is selected. A 'GET Get All User' request is highlighted. The main panel shows a 'GET' request to 'localhost:8080/api/user'. The 'Body' tab displays a JSON response:

```
1 {
2     "result": true,
3     "message": null,
4     "data": [
5         {
6             "id": 1,
7             "password": "23456789",
8             "username": "ryan ryan",
9             "fullname": "Ryan Roshandy S",
10            "email": "tyaroshandy@gmail.com"
11        },
12        {
13            "id": 2,
14            "password": "23456789",
15            "username": "ryans",
16            "fullname": "Ryan S",
17            "email": "tyars@gmail.com"
18        }
19    ]
20}
```

The status bar at the bottom indicates 'Status: 200 OK Time: 8 ms Size: 414 B'.

Find All Success

Update User

The screenshot shows the Postman application interface. The sidebar is identical to the previous one. A 'PUT New Request' is selected under 'User'. The main panel shows a 'PUT' request to 'localhost:8080/api/user'. The 'Body' tab displays a JSON payload:

```
1 {
2     "id": 1,
3     "password": "23456789",
4     "username": "ryan ryan",
5     "email": "tyaroshandy@gmail.com",
6     "username": "sil ryan"
7 }
```

The status bar at the bottom indicates 'Status: 200 OK Time: 19 ms Size: 300 B'.

Update Success

The screenshot shows the Postman interface with a collection named 'Acarain'. A PUT request is being made to 'localhost:8080/api/user' with the following JSON body:

```

1 "id": 2,
2 "password": "24216846",
3 "full_name": "Ryan RS",
4 "email": "",
5 "username": ""

```

The response status is 400 Bad Request, indicating validation errors. The error message in the response body is:

```

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

```

Update Error

Find By ID User

The screenshot shows the Postman interface with a collection named 'Acarain'. A GET request is being made to 'localhost:8080/api/user/1' with the following query parameters:

KEY	VALUE	DESCRIPTION
Key	Value	Description

The response status is 200 OK, and the response body is:

```

1 {
2   "result": true,
3   "message": null,
4   "data": [
5     {
6       "id": 1,
7       "password": "12345678",
8       "username": "Rho Alz",
9       "full_name": "Ryan Roshandy S",
10      "email": "tyarrozhandy@gmail.com"
11    }
12  ]
}

```

Find by Id Success

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections like 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', 'Plans', and 'History'. The main area displays a collection named 'Acarain' with a sub-collection 'Acarain_API'. Under 'Acarain_API', there are several requests: 'POST Post Events', 'PUT New Request', 'GET Get All Events', 'GET Get by ID', 'DELETE Delete Events by Id', 'POST Orders', 'POST Transaction', and 'User'. Under 'User', there are 'POST Post User Data', 'GET Get All User', and 'PUT New Request'. A specific request, 'GET Get User By Id', is selected and expanded. The request details show a 'GET' method with the URL 'localhost:8080/api/user/4'. The 'Body' tab is selected, showing the JSON response:

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

The status bar at the bottom indicates 'Status: 400 Bad Request'.

Find by Id error/null value

D. Order

Insert Order

The screenshot shows the Postman interface with the following details:

- Environment:** Acarain
- Request Type:** POST
- URL:** localhost:8080/api/orders
- Body (JSON):**

```
1 {
2     "jumlahTicket": 10,
3     "total": 12,
4     "user": {
5         "id": 1
6     }
7 }
```
- Response Status:** 200 OK
- Response Body (JSON):**

```
1 {
2     "result": true,
3     "message": [],
4     "data": [
5         {
6             "idUserOrder": 2,
7             "jumlahTicket": "10",
8             "user": {
9                 "id": 1,
10                "password": null,
11                "username": null,
12                "fullname": null,
13                "email": null,
14            },
15            "ticket": null,
16            "total": "12"
17        }
18    ]
19 }
```

Insert Success

Find All Order

The screenshot shows the Postman interface with the following details:

- Environment:** Acarain
- Request Type:** GET
- URL:** localhost:8080/api/orders/
- Query Params:** None
- Response Status:** 200 OK
- Response Body (JSON):**

```
1 {
2     "result": true,
3     "message": null,
4     "data": [
5         {
6             "idUserOrder": 1,
7             "jumlahTicket": "10",
8             "user": {
9                 "id": 1,
10                "password": "24215546",
11                "username": "si gans",
12                "fullname": "Ryan RS",
13                "email": "ryand@gmail.com",
14            },
15            "ticket": null,
16            "total": "12"
17        },
18        {
19             "idUserOrder": 2,
20             "jumlahTicket": "10",
21             "user": {
22                 "id": 1,
23                 "password": "12345678",
24                 "username": null,
25                 "fullname": null,
26                 "email": null
27             }
28         }
29     ]
30 }
```

Find All Success

Update Order

The screenshot shows the Postman application interface. A collection named 'Acarain' is selected. In the 'Orders' section, the 'PUT Update Order' request is highlighted. The URL is set to 'localhost:8080/api/orders/'. The request body contains the following JSON:

```
1 "id": "1",
2 "jumlahTicket": 129,
3 "total": 32,
4 "user": {
5   "id": 1
6 }
```

The response status is 200 OK, Time: 13 ms, Size: 372 B. The response body is:

```
1 {
2   "result": true,
3   "message": [],
4   "data": [
5     {
6       "idUserOrder": 1,
7       "jumlahTicket": "28",
8       "user": {
9         "id": 2,
10        "password": "12345665",
11        "username": "si_goku",
12        "fullname": "Ryan RS",
13        "email": "ryans@gmail.com"
14      },
15      "ticket": null,
16      "total": "32"
17    }
18 ]
```

Update Success

The screenshot shows the Postman application interface. A collection named 'Acarain' is selected. In the 'Orders' section, the 'PUT Update Order' request is highlighted. The URL is set to 'localhost:8080/api/orders/'. The request body contains the following JSON:

```
1 "jumlahTicket": 29,
2 "total": 32,
3 "user": {
4   "id": 1
5 }
```

The response status is 400 Bad Request, Time: 9 ms, Size: 201 B. The response body is:

```
1 {
2   "result": false,
3   "message": [
4     {
5       "ID is required"
6     }
7   ],
8   "data": null
9 }
```

Update Error

Find By ID Order

The screenshot shows the Postman interface with a successful API call. The URL is `localhost:8080/api/orders/2`. The response body is:

```
1 "result": true,
2 "message": null,
3 "data": [
4     {
5         "id": "1",
6         "julalTicket": "18",
7         "user": {
8             "id": "1",
9             "password": "12345678",
10            "username": "aku alay",
11            "fullname": "Ryan Roshandy S",
12            "email": "ryannroshandy@gmail.com"
13        },
14        "ticket": null,
15        "total": "12"
16    }
17]
```

Find by Id Success

The screenshot shows the Postman interface with an error response. The URL is `localhost:8080/api/orders/4`. The response body is:

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

Find by Id error/null value

Delete Order

The screenshot shows the Postman application interface. The left sidebar displays the 'Acarain' workspace with various API collections like 'Events' and 'Orders'. The main panel shows a 'DELETE' request to 'localhost:8080/api/orders/2'. The 'Body' tab contains a JSON payload with 'Key': 'Value' pairs. The response status is 200 OK, with a body containing a success message and a null data object.

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

Delete Success

The screenshot shows the Postman application interface. The left sidebar displays the 'Acarain' workspace with various API collections like 'Events' and 'Orders'. The main panel shows a 'DELETE' request to 'localhost:8080/api/orders/2'. The response status is 400 Bad Request, indicating that the entity with id 2 does not exist.

```
1 "result": false,
2 "message": [
3     "No class con.acarainservices.acarainservices.Orders entity with id 2 exists!"
4 ],
5 "data": null
```

Delete Error

E. Ticket

Find All Ticket

The screenshot shows the Postman interface with the following details:

- Header:** GET Find All Ticket
- URL:** localhost:8080/api/tickets
- Response Status:** 200 OK (31 ms, 3.12 KB)
- Body (Pretty):**

```
1.  "events": [
2.     {
3.         "title": "crowdIT",
4.         "organizer": "HIMA TI IBIK",
5.         "tag": "IT",
6.         "type": "Lomba",
7.         "location": "Bogor",
8.         "dateStart": "2023-01-02",
9.         "timeStart": null,
10.        "dateEnd": "2023-01-03",
11.        "timeEnd": null,
12.        "description": "create your own portfolio",
13.        "user": [
14.            {
15.                "id": 1,
16.                "password": "thesya23",
17.                "username": "thesyamarcella",
18.                "fullName": "Thesya Marcella"
19.            }
20.        ]
21.    }
22. ]
```

Find All Success

Find By ID Ticket

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace
- Request:** GET /api/tickets/{id}
- URL:** localhost:8080/api/tickets/1
- Response Status:** 200 OK (28 ms, 871 B)
- Body (Pretty):**

```
1 {  
2   "result": true,  
3   "message": null,  
4   "data": [  
5     {  
6       "id": 1,  
7       "name": "Workshop CROWD IT",  
8       "kapasitas": "300",  
9       "deskripsi": "Workshop yang dilakukan sebagai  
10      rangkaian dari CROWD IT",  
11      "harga": 8,  
12      "startDate": "2023-01-02",  
13      "endDate": "2023-01-02",  
14      "maxOrders": 300,  
15      "events": [  
16        {  
17          "title": "CrowdIT",  
18          "organizer": "HIMA TI IBIK",  
19        }  
20      ]  
21    }  
22  ]
```

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace
- Request:** GET /api/tickets/events/{id}
- URL:** localhost:8080/api/tickets/events/1
- Status:** 200 OK (Time: 34 ms, Size: 1.4 KB)
- Body (Pretty):**

```
1 {  
2   "result": true,  
3   "message": null,  
4   "data": [  
5     {  
6       "id": 1,  
7       "name": "Workshop CROWD IT",  
8       "kapasitas": "300",  
9       "deskripsi": "Workshop yang dilakukan sebagai rangkaian dari CROWD IT",  
10      "harga": 8,  
11      "startDate": "2023-01-02",  
12      "endDate": "2023-01-02",  
13      "maxOrders": 300,  
14      "events": [  
15        {  
16          "title": "CrowdIT",  
17          "organizer": "HIMA TI IBIK",  
18        }  
19      ]  
20    }  
21  ]
```

Find by Id Success

F. Email Sender

12:21 20.6KB/s 85%



Acarain.com External Inbox



ryanroshandy6@gm... 12/17/2022
to me ▾



Halo User

Terima Kasih Telah Mengunjungi Website Kami

Datang lagi yaaa

Reply

Reply all

Forward

Available add-ons:



Email Received