Soal B

Berdasarkan Soal A, buatlah CRUD dengan memiliki ketentuan sebagai berikut : 1. Fitur CRUD untuk tabel "users", "courses" dan "userCourse". 2. Memiliki 2 akses login administrator maupun user biasa. Dengan ketentuan sebagai berikut : a. Jika login sebagai admin menampilkan data keseluruhan user sesuai dengan ketentuan masing - masing pada Soal A. b. Jika login sebagai user menampilkan data sesuai dengan user yang melakukan login. 3. Buatlah grafik yang ditampilkan dalam dashboard admin sesuai dengan query / perintah Soal A. point 7 dan 8. 4. Buatlah API yang menampilkan response sesuai dengan query / perintah Soal A. pada point 5,6,7 dan 8. 5. Buatlah Contoh File UnitTest dengan ketentuan sebagai berikut : a. UnitTest untuk melakukan validasi akun yang melakukan login b. UnitTest untuk melakukan validasi input data user.

1. API terkait soal A point 5

Script di python :

from flask import Flask, request, jsonify

import mysql.connector

app = Flask(\_\_name\_\_)

def get\_connection():

connection = mysql.connector.connect(

host='localhost',

user='pythonuser',

password='Python12345',

database='sekolahku'

)

return connection

@app.route('/get\_data5', methods=['POST'])

def get\_user():

username = request.json['username']

password = request.json['password']

connection = get\_connection()

cursor = connection.cursor()

if username == 'Admin' :

query = f"""

SELECT

u.id AS id,

u.username AS username,

c.course AS course,

c.mentor AS mentor,

c.title AS title

FROM

usercourse uc

INNER JOIN

users u ON uc.id\_user = u.id

INNER JOIN

courses c ON uc.id\_course = c.id

where c.title in ('S.T.','S.Kom')

order by u.id asc;"""

cursor.execute(query,)

else :

query = f"""

SELECT

u.id AS id,

u.username AS username,

c.course AS course,

c.mentor AS mentor,

c.title AS title

FROM

usercourse uc

INNER JOIN

users u ON uc.id\_user = u.id

INNER JOIN

courses c ON uc.id\_course = c.id

where u.username = %s AND u.password =%s

AND c.title in ('S.T.','S.Kom')

order by u.id asc; """

cursor.execute(query, (username, password,))

result = cursor.fetchall()

cursor.close()

connection.close()

if result:

return jsonify(result), 200

else:

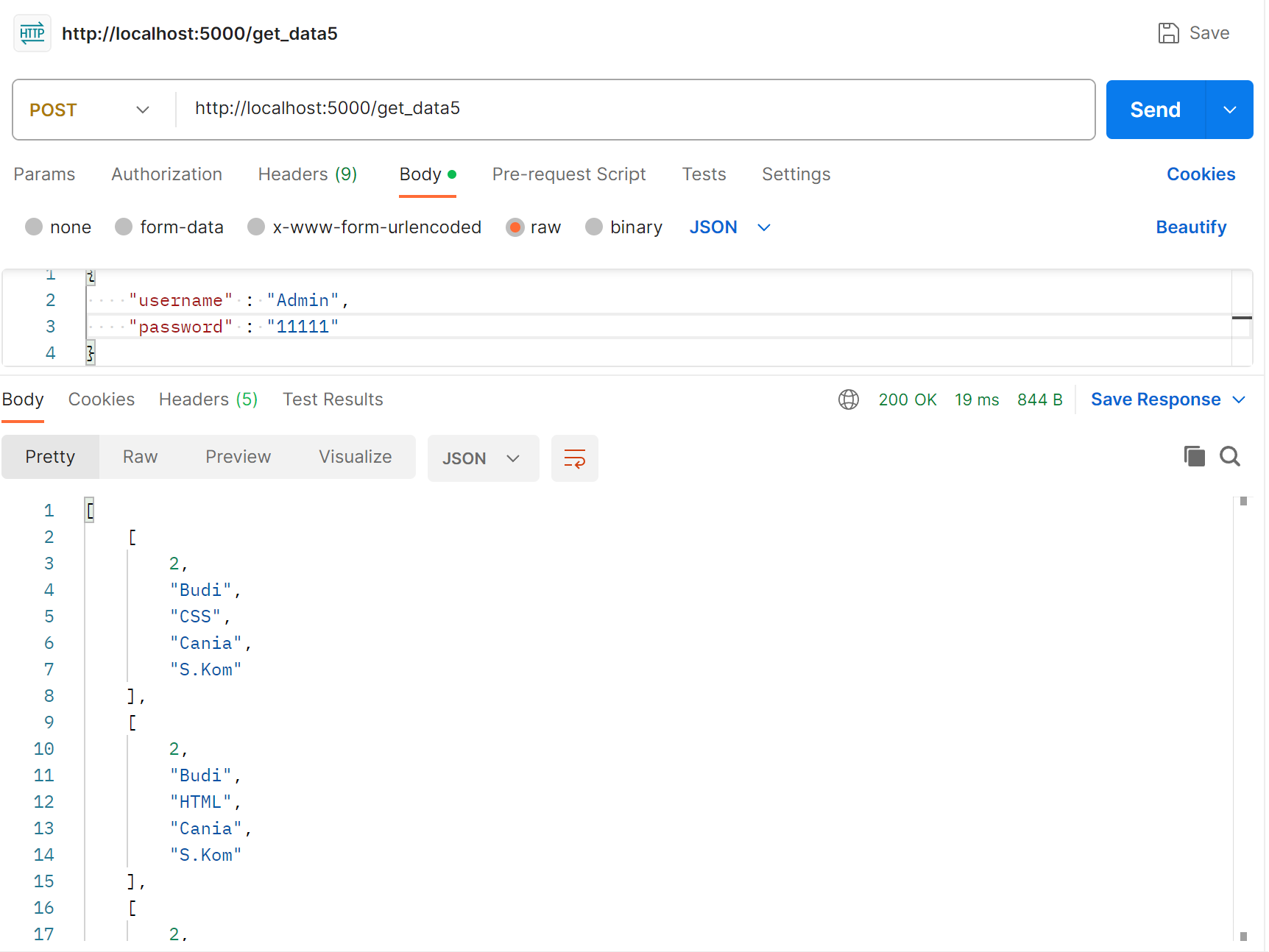
return jsonify({"error": "Authentication failed"}), 404

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

============================hasil di postman=========================

Menggunakan user Admin



Hasil plain text :

[

[

2,

"Budi",

"CSS",

"Cania",

"S.Kom"

],

[

2,

"Budi",

"HTML",

"Cania",

"S.Kom"

],

[

2,

"Budi",

"Javascript",

"Cania",

"S.Kom"

],

[

3,

"Caca",

"Python",

"Barry",

"S.T."

],

[

3,

"Caca",

"Micropython",

"Barry",

"S.T."

],

[

4,

"Deni",

"HTML",

"Cania",

"S.Kom"

],

[

5,

"Euis",

"CSS",

"Cania",

"S.Kom"

],

[

5,

"Euis",

"Javascript",

"Cania",

"S.Kom"

],

[

6,

"Fafa",

"Python",

"Barry",

"S.T."

],

[

6,

"Fafa",

"Micropython",

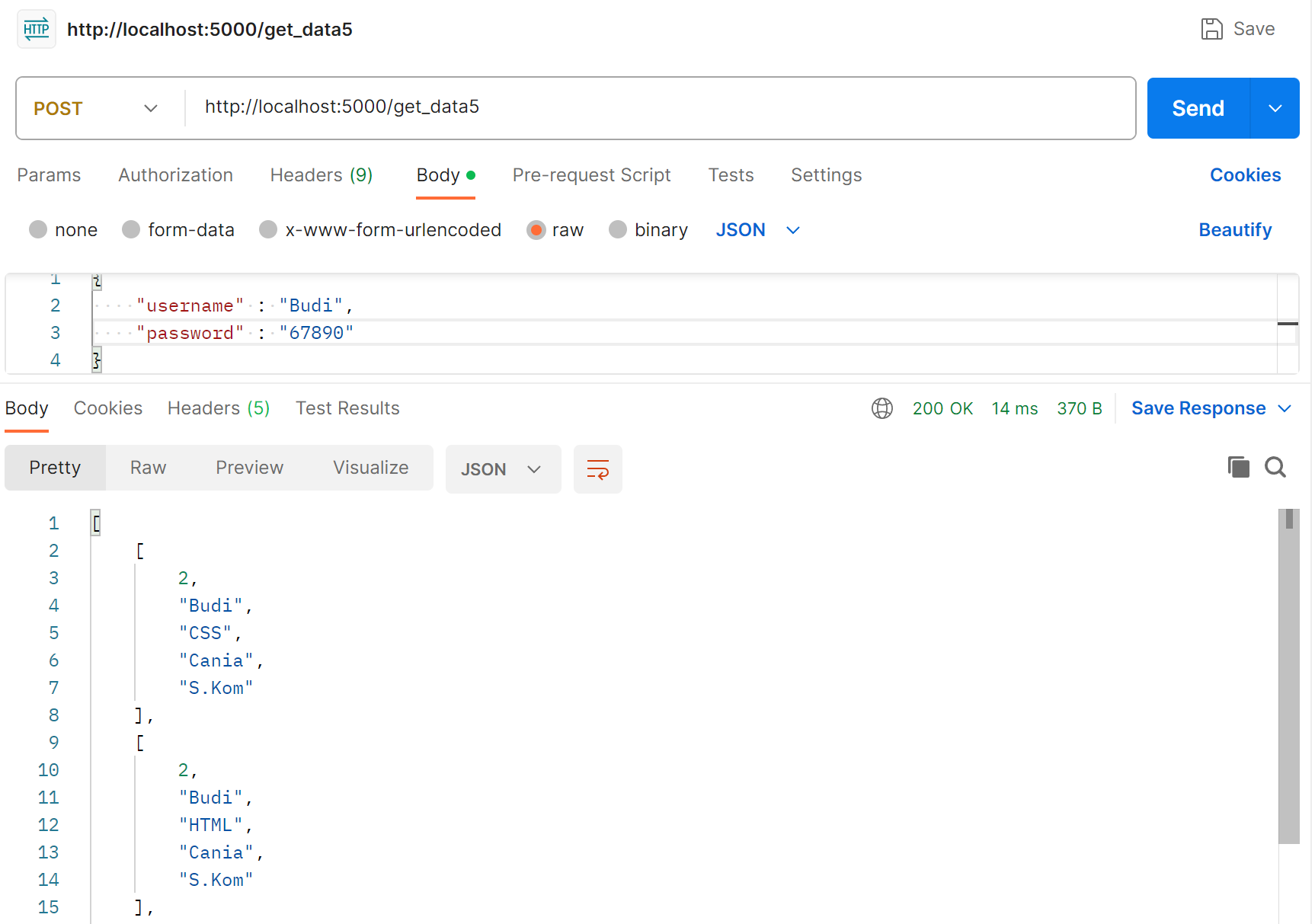
"Barry",

"S.T."

]

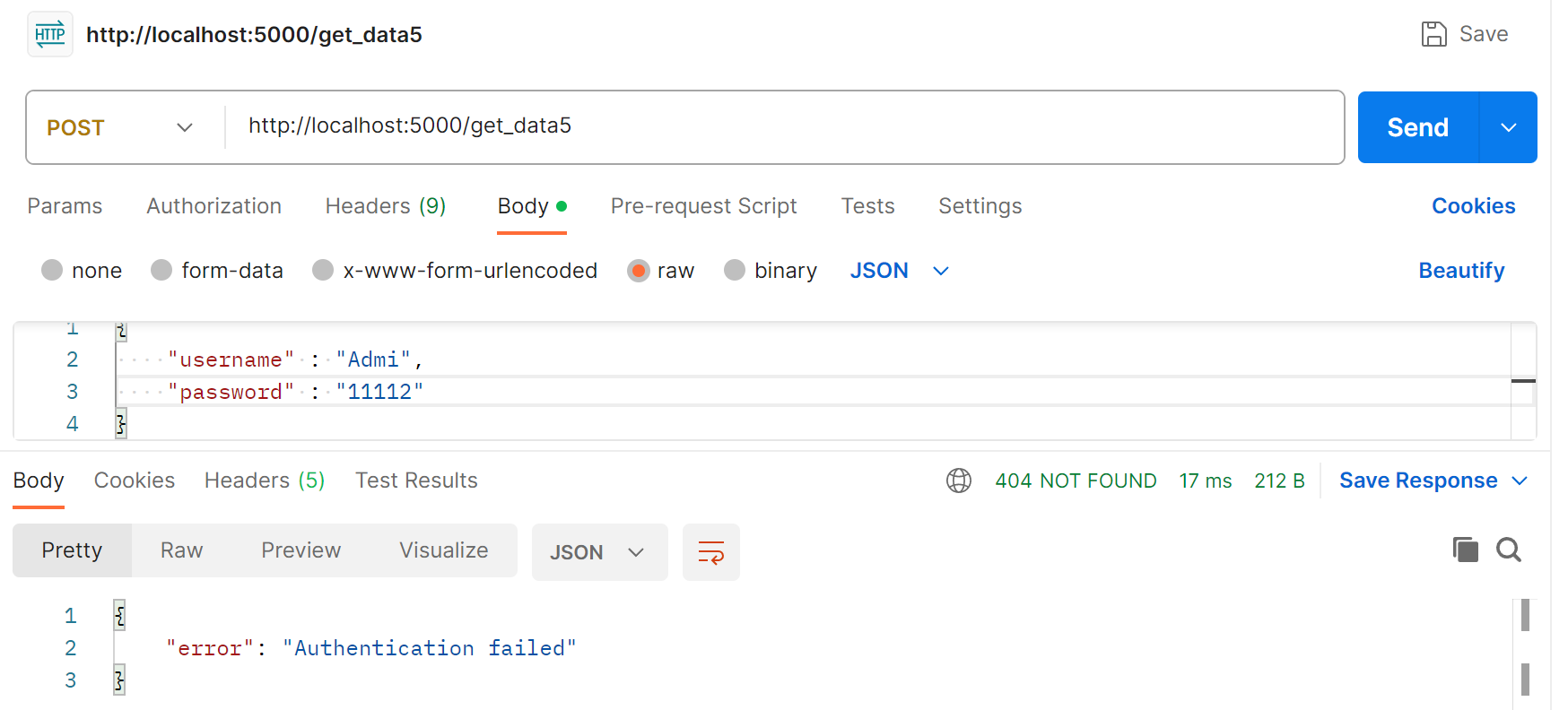
]

Menggunakan user Budi





Menggunakan username atau password selain Admin dan user terdaftar

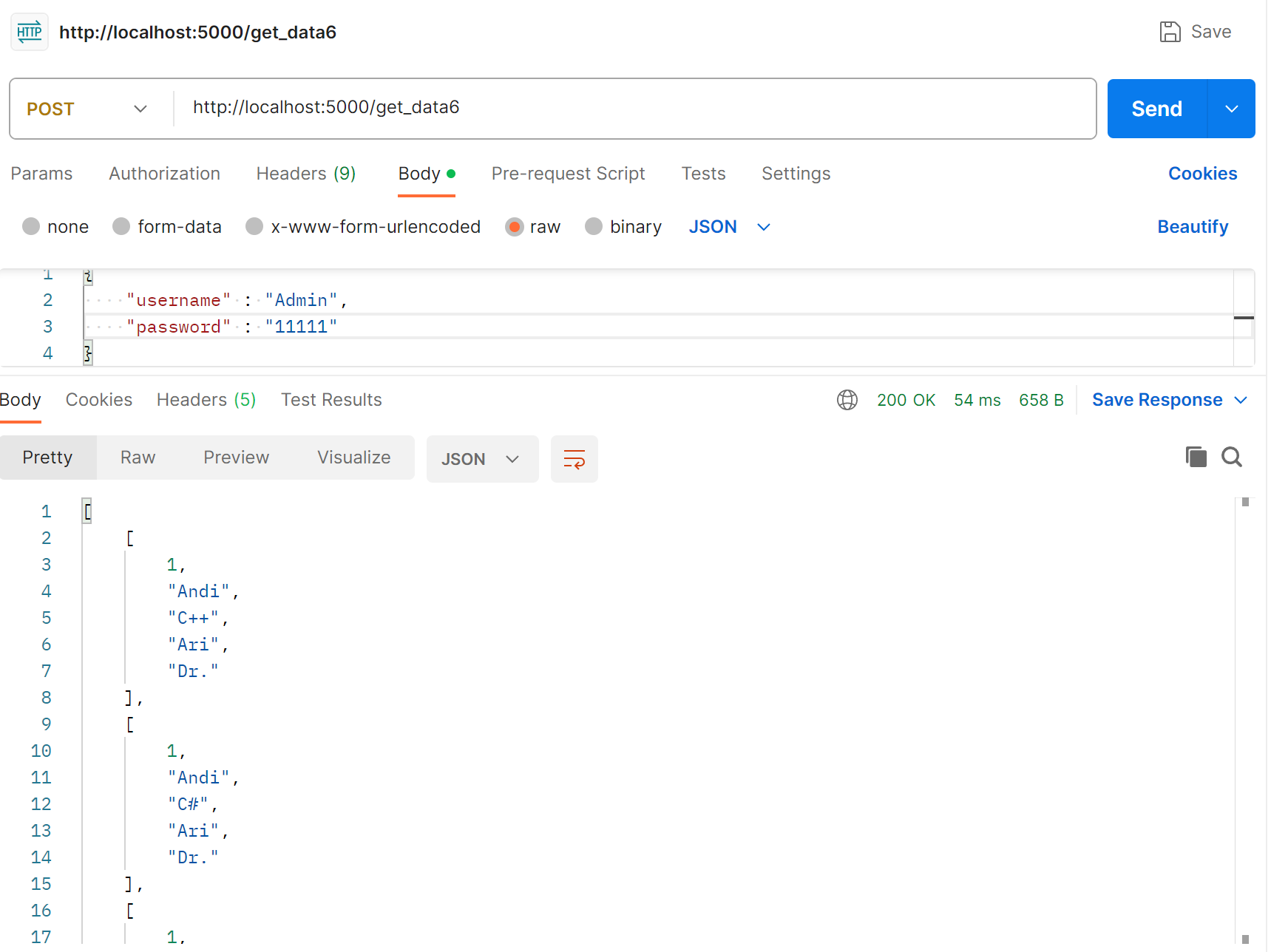


1. API terkait soal B point 6

Python code :

from flask import Flask, request, jsonify  
import mysql.connector  
  
app = Flask(\_\_name\_\_)  
  
def get\_connection():  
 connection = mysql.connector.connect(  
 host='localhost',  
 user='pythonuser',  
 password='Python12345',  
 database='sekolahku'  
 )  
 return connection  
  
@app.route('/get\_data6', methods=['POST'])  
def get\_user():  
 username = request.json['username']  
 password = request.json['password']  
  
 connection = get\_connection()  
 cursor = connection.cursor()  
  
 if username == 'Admin' :  
 query = f"""  
SELECT   
 u.id AS id,  
 u.username AS username,  
 c.course AS course,  
 c.mentor AS mentor,  
 c.title AS title  
FROM   
 usercourse uc  
INNER JOIN   
 users u ON uc.id\_user = u.id  
INNER JOIN   
 courses c ON uc.id\_course = c.id  
where c.title not in ('S.T.','S.Kom')  
order by u.id asc;"""  
 cursor.execute(query,)  
 else :  
 query = f"""  
SELECT   
 u.id AS id,  
 u.username AS username,  
 c.course AS course,  
 c.mentor AS mentor,  
 c.title AS title  
FROM   
 usercourse uc  
INNER JOIN   
 users u ON uc.id\_user = u.id  
INNER JOIN   
 courses c ON uc.id\_course = c.id  
where u.username = %s AND u.password =%s   
AND c.title not in ('S.T.','S.Kom')  
order by u.id asc; """  
 cursor.execute(query, (username, password,))  
  
 result = cursor.fetchall()  
 cursor.close()  
 connection.close()  
  
 if result:  
 return jsonify(result), 200  
 else:  
 return jsonify({"error": "Authentication failed"}), 404  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=True)

Unit Test di Postman menggunakan user admin



[

[

1,

"Andi",

"C++",

"Ari",

"Dr."

],

[

1,

"Andi",

"C#",

"Ari",

"Dr."

],

[

1,

"Andi",

"C++",

"Ari",

"Dr."

],

[

3,

"Caca",

"Java",

"Darren",

"M.T."

],

[

4,

"Deni",

"C++",

"Ari",

"Dr."

],

[

4,

"Deni",

"C++",

"Ari",

"Dr."

],

[

5,

"Euis",

"C#",

"Ari",

"Dr."

],

[

6,

"Fafa",

"Java",

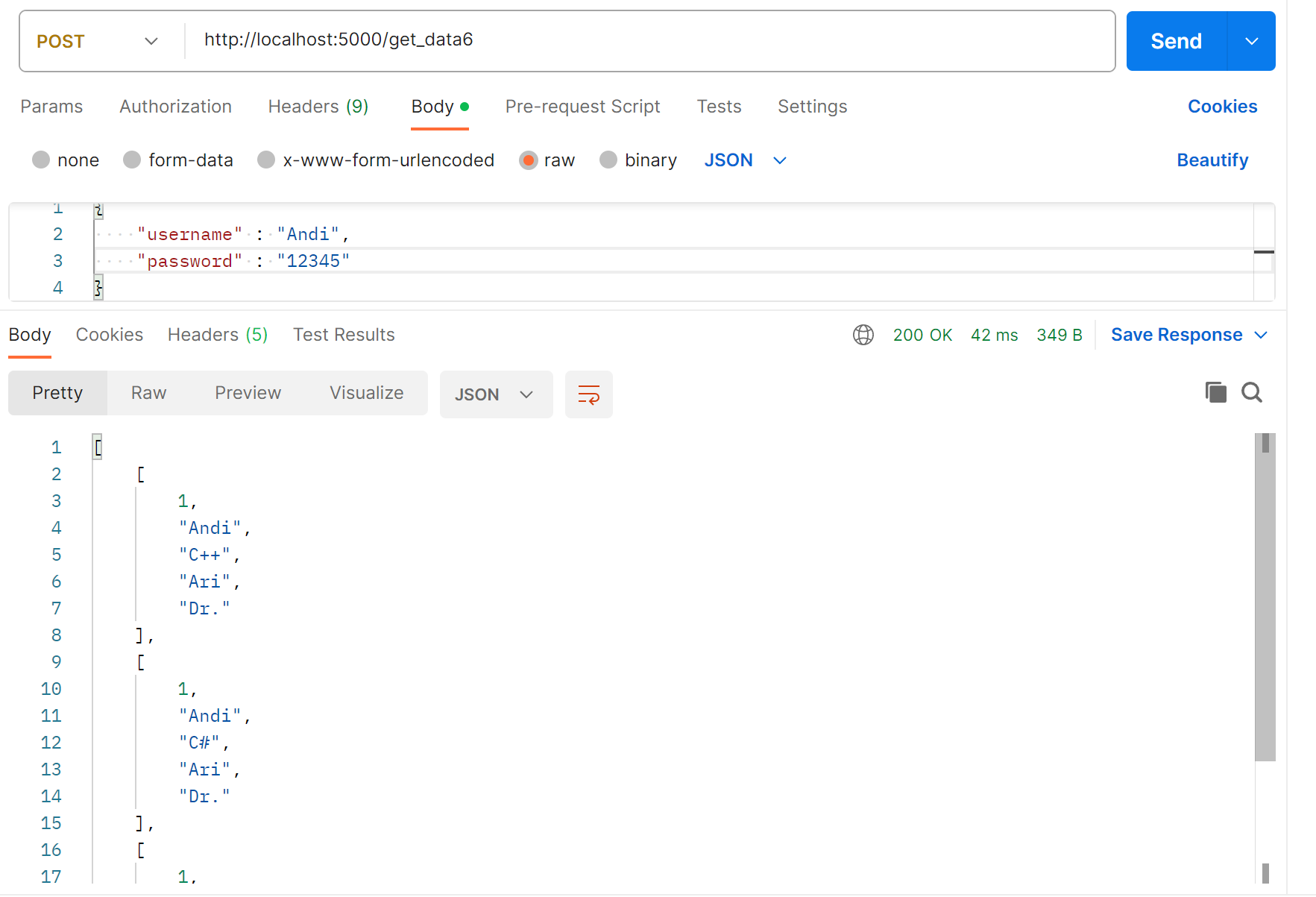
"Darren",

"M.T."

]

]

Unit Test di Postman menggunakan user yang datanya ada di dalam data



[

[

1,

"Andi",

"C++",

"Ari",

"Dr."

],

[

1,

"Andi",

"C#",

"Ari",

"Dr."

],

[

1,

"Andi",

"C++",

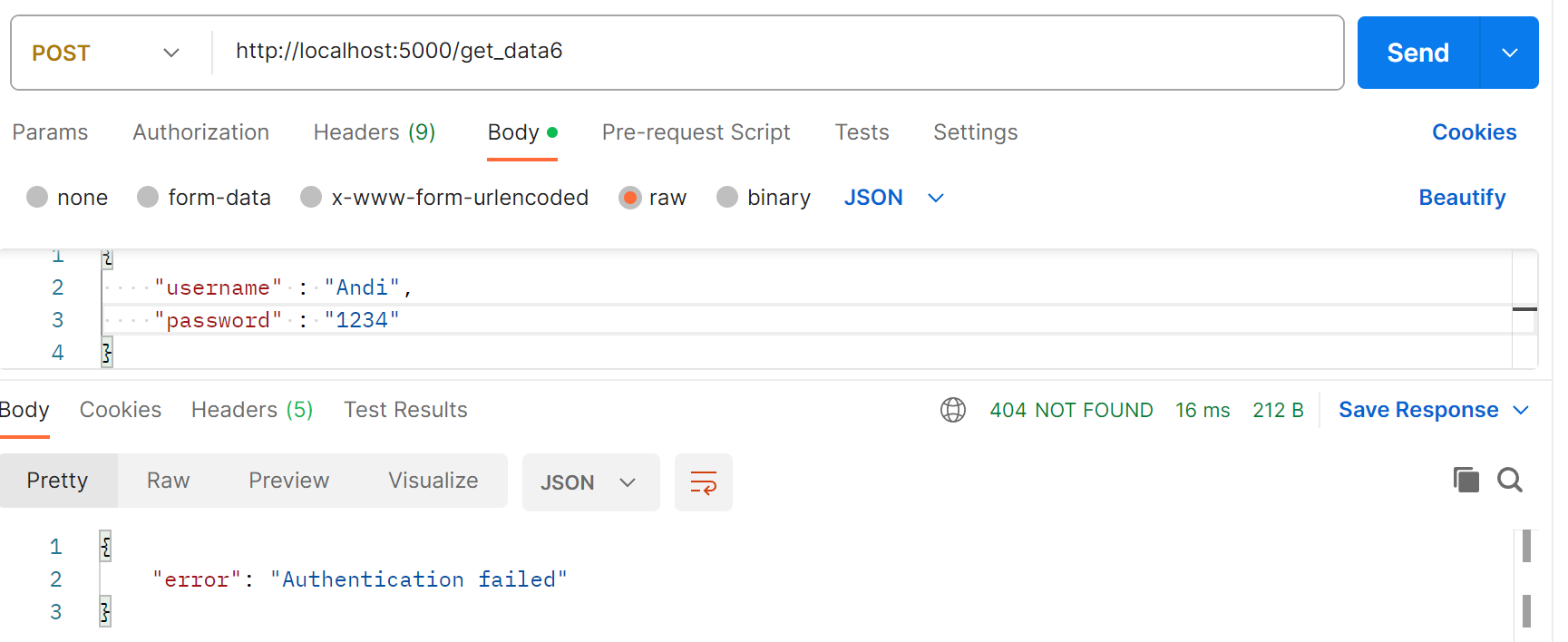
"Ari",

"Dr."

]

]

Unit Test menggunakan username password yang tidak sesuai

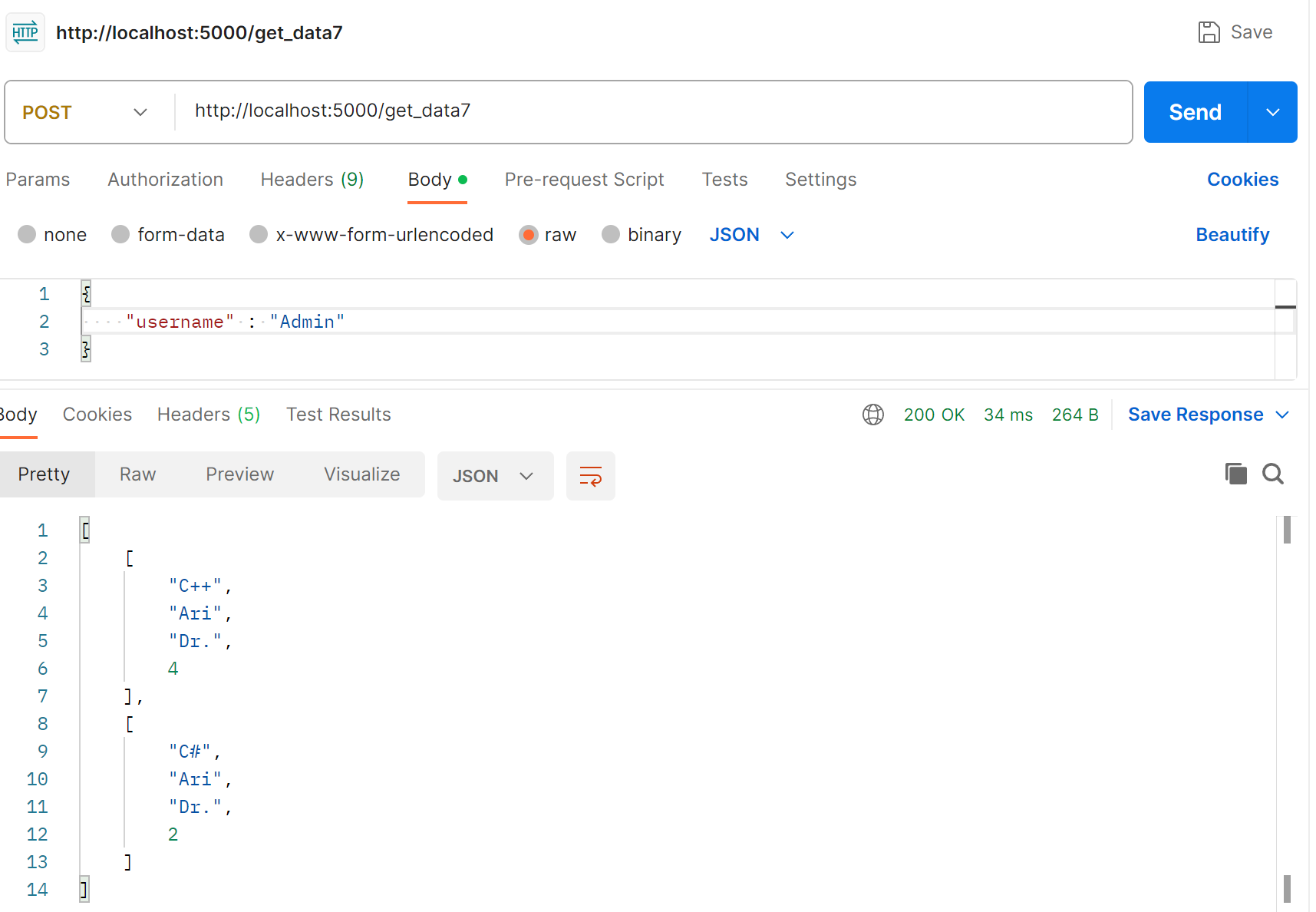


1. API terkait soal B point 7

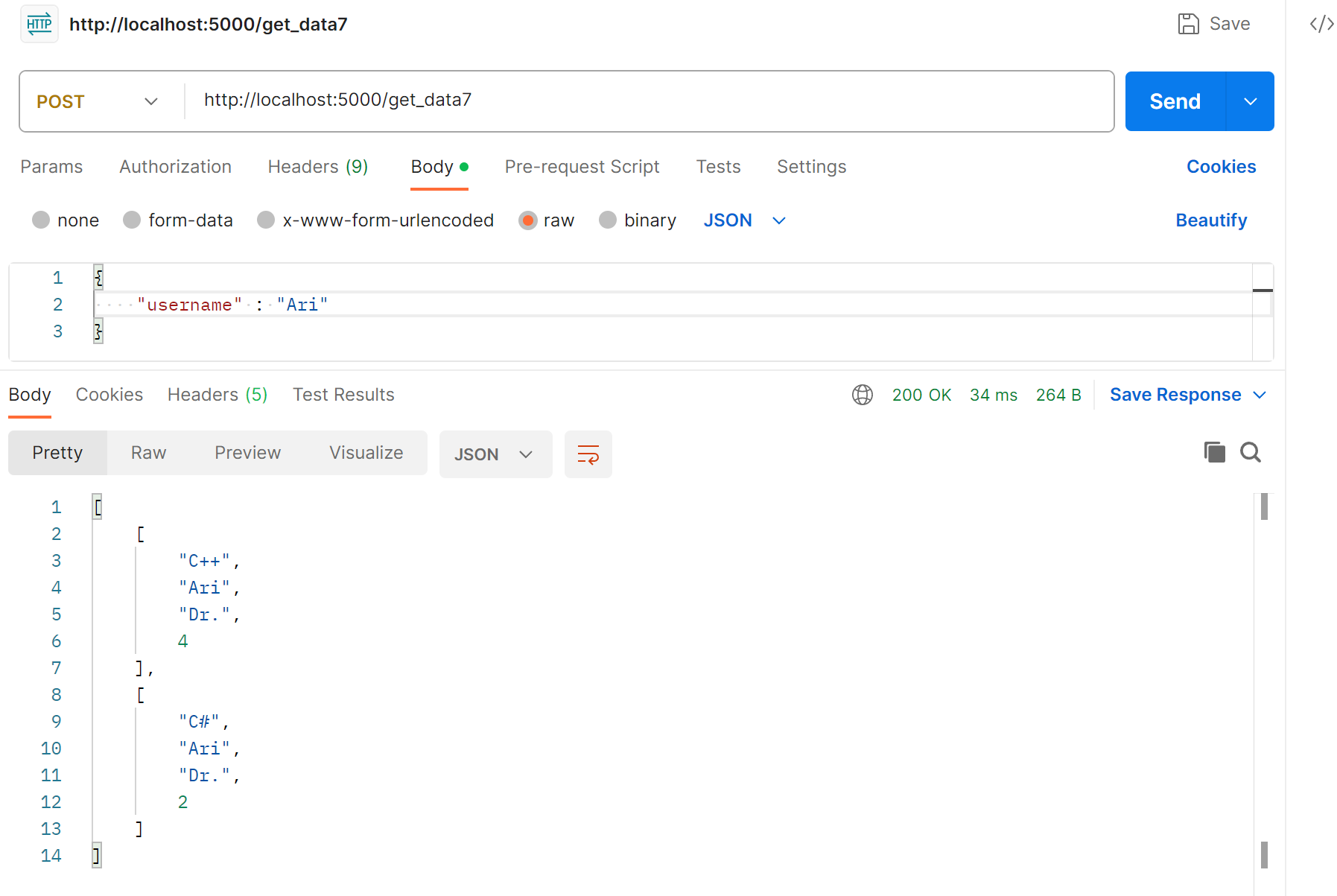
Python

from flask import Flask, request, jsonify  
import mysql.connector  
  
app = Flask(\_\_name\_\_)  
  
def get\_connection():  
 connection = mysql.connector.connect(  
 host='localhost',  
 user='pythonuser',  
 password='Python12345',  
 database='sekolahku'  
 )  
 return connection  
  
@app.route('/get\_data7', methods=['POST'])  
def get\_user():  
 username = request.json['username']  
  
 connection = get\_connection()  
 cursor = connection.cursor()  
  
 if username == 'Admin' :  
 query = f"""  
SELECT   
 c.course AS course,  
 c.mentor AS mentor,  
 c.title AS title,  
 COUNT(u.username) AS jumlahpeserta  
FROM   
 usercourse uc  
INNER JOIN   
 users u ON uc.id\_user = u.id  
INNER JOIN   
 courses c ON uc.id\_course = c.id  
GROUP BY   
 c.course,  
 c.mentor,  
 c.title  
ORDER BY   
 mentor ASC;"""  
 cursor.execute(query,)  
 else :  
 query = f"""  
SELECT   
 c.course AS course,  
 c.mentor AS mentor,  
 c.title AS title,  
 COUNT(u.username) AS jumlahpeserta  
FROM   
 usercourse uc  
INNER JOIN   
 users u ON uc.id\_user = u.id  
INNER JOIN   
 courses c ON uc.id\_course = c.id  
where c.mentor = %s  
GROUP BY   
 c.course,  
 c.mentor,  
 c.title  
order by mentor ASC;"""  
 cursor.execute(query, (username,))  
  
 result = cursor.fetchall()  
 cursor.close()  
 connection.close()  
  
 if result:  
 return jsonify(result), 200  
 else:  
 return jsonify({"error": "Authentication failed"}), 404  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=True)

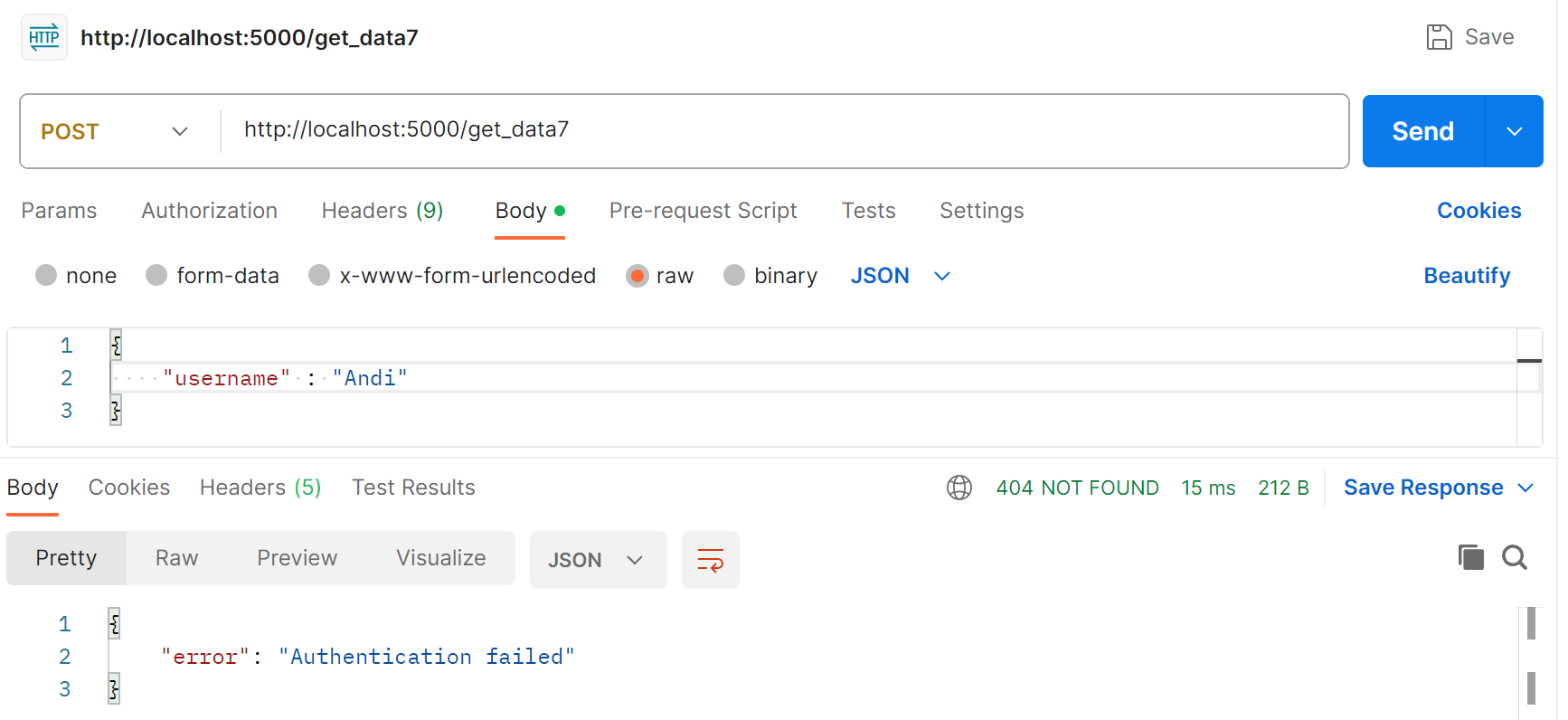
Unit Test menggunakan username Admin



Unit Test menggunakan username mentor



Unit test selain menggunakan username mentor

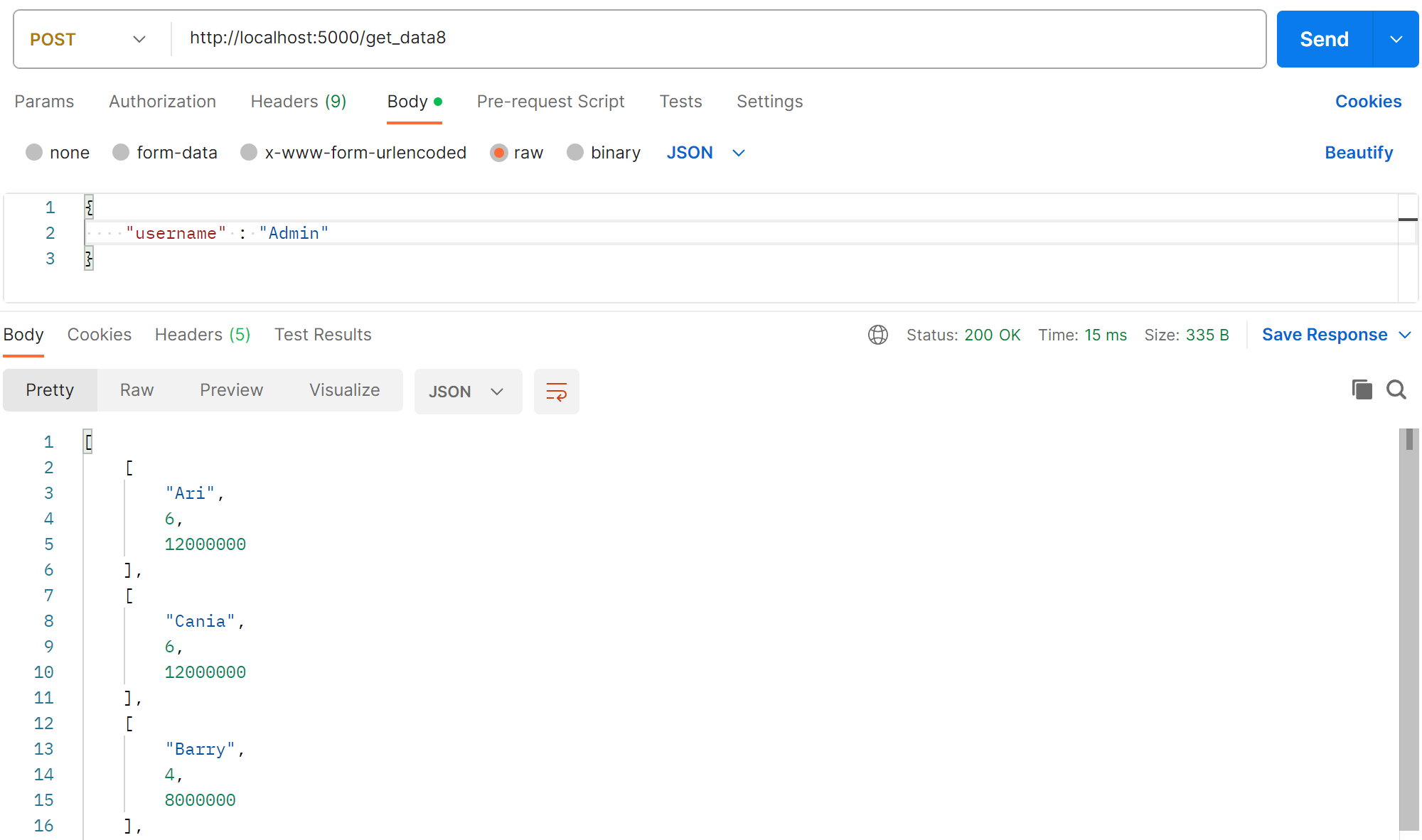


1. API terkait soal B point 8

Python

from flask import Flask, request, jsonify  
import mysql.connector  
  
app = Flask(\_\_name\_\_)  
  
def get\_connection():  
 connection = mysql.connector.connect(  
 host='localhost',  
 user='pythonuser',  
 password='Python12345',  
 database='sekolahku'  
 )  
 return connection  
  
@app.route('/get\_data8', methods=['POST'])  
def get\_user():  
 username = request.json['username']  
  
 connection = get\_connection()  
 cursor = connection.cursor()  
  
 if username == 'Admin' :  
 query = f"""  
SELECT   
 c.mentor AS mentor,  
 COUNT(u.username) AS jumlahpeserta,  
 COUNT(u.username)\*2000000 AS total\_fee  
FROM   
 usercourse uc  
INNER JOIN   
 users u ON uc.id\_user = u.id  
INNER JOIN   
 courses c ON uc.id\_course = c.id  
GROUP BY   
 c.mentor  
ORDER BY   
 jumlahpeserta DESC;"""  
 cursor.execute(query,)  
 else :  
 query = f"""  
SELECT   
 c.mentor AS mentor,  
 COUNT(u.username) AS jumlahpeserta,  
 COUNT(u.username)\*2000000 AS total\_fee  
FROM   
 usercourse uc  
INNER JOIN   
 users u ON uc.id\_user = u.id  
INNER JOIN   
 courses c ON uc.id\_course = c.id  
 where c.mentor = %s  
GROUP BY   
 c.mentor  
ORDER BY   
 jumlahpeserta DESC;  
"""  
 cursor.execute(query, (username,))  
  
 result = cursor.fetchall()  
 cursor.close()  
 connection.close()  
  
 if result:  
 return jsonify(result), 200  
 else:  
 return jsonify({"error": "Authentication failed"}), 404  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=True)

Unit test menggunakan user admin



[

[

"Ari",

6,

12000000

],

[

"Cania",

6,

12000000

],

[

"Barry",

4,

8000000

],

[

"Darren",

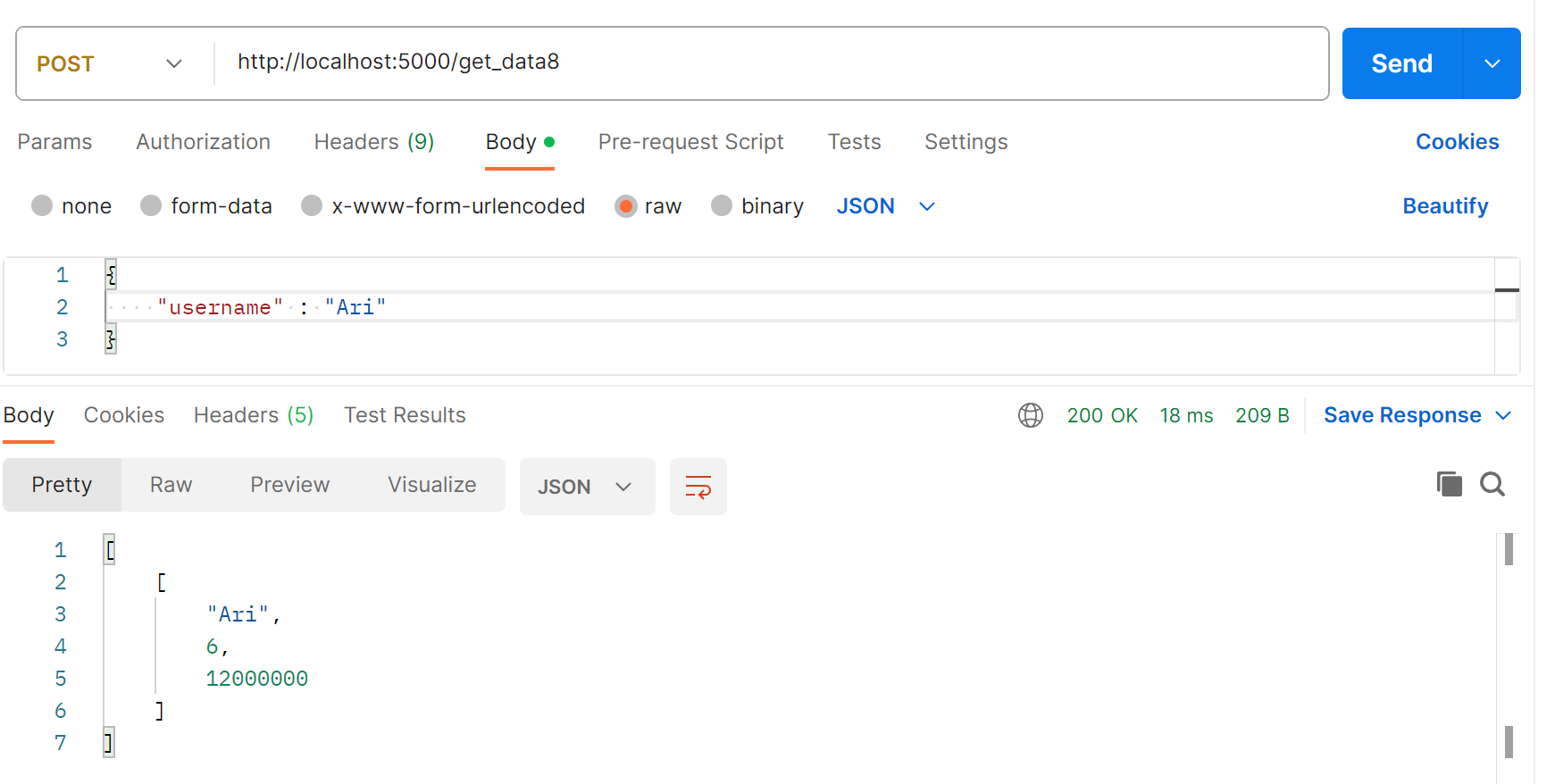
2,

4000000

]

]

Unit Test menggunakan user mentor



Unit Test menggunakan user selain mentor dan admin