Nama : Khalish Muhammad Fattah

Basic QE Kelas D

**Soal API Testing Eksplorasi**

1. Lakukan pengujian pada aplikasi RESTful API berikut dengan menggunakan Postman:

| **No** | **Endpoint** | **Method** | **Deskripsi** |
| --- | --- | --- | --- |
| 1 | <https://jsonplaceholder.typicode.com/posts> | GET | Mendapatkan semua data post. |
| 2 | <https://jsonplaceholder.typicode.com/posts> | POST | Membuat data post baru. |
| 3 | <https://jsonplaceholder.typicode.com/posts/1> | GET | Mendapatkan data post berdasarkan ID. |
| 4 | <https://jsonplaceholder.typicode.com/posts/1> | PUT | Mengubah data post berdasarkan ID. |
| 5 | <https://jsonplaceholder.typicode.com/posts/1> | DELETE | Menghapus data post berdasarkan ID. |

Kriteria dari pembuatan test case adalah sebagai berikut:

1. Terdapat case positif dan negatif.

Jawaban:

# 

# Endpoint 1

## Script

pm.test("Status code is 200", function () {

pm.response.to.have.status(200);

});

pm.test("Response is an array", function () {

pm.expect(pm.response.json()).to.be.an('array');

});

pm.test("Response array contains at least one post", function () {

pm.expect(pm.response.json().length).to.be.above(0);

});

pm.test("Each item in the response array has correct keys", function () {

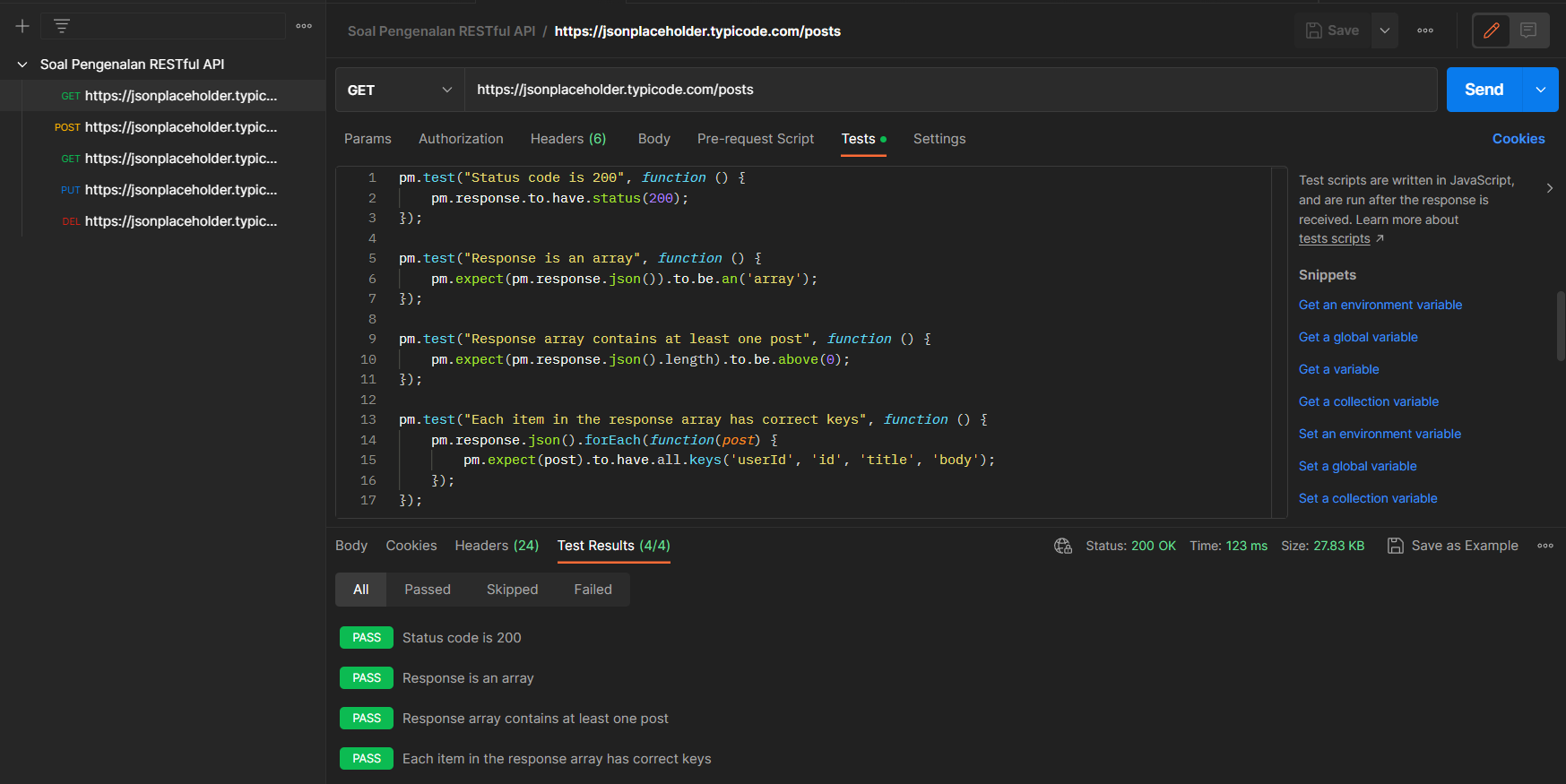
pm.response.json().forEach(function(post) {

pm.expect(post).to.have.all.keys('userId', 'id', 'title', 'body');

});

});

## Screenshot



* Penjelasan

Test case pertama "Status code is 200" memastikan bahwa status code yang diterima oleh Postman setelah melakukan request ke server adalah 200, yang artinya request berhasil dan response dari server diterima oleh Postman dengan baik.

Test case kedua "Response is an array" memastikan bahwa response yang diterima oleh Postman merupakan sebuah array, dan bukan tipe data lain seperti string atau object.

Test case ketiga "Response array contains at least one post" memastikan bahwa array response yang diterima oleh Postman mengandung minimal satu post.

Test case keempat "Each item in the response array has correct keys" memastikan bahwa setiap item dalam array response mengandung keys 'userId', 'id', 'title', dan 'body' dengan nilai yang sesuai dengan ekspektasi.

# 

# Endpoint 2

## Script

// Test case positif 2: Membuat data post baru dengan benar

pm.sendRequest({

url: "https://jsonplaceholder.typicode.com/posts",

method: "POST",

header: {

"Content-type": "application/json; charset=UTF-8"

},

body: {

mode: "raw",

raw: JSON.stringify({

"title": "testing post baru",

"body": "ini adalah body dari post baru",

"userId": 1

})

}

}, function (err, res) {

if (err) {

console.log(err);

} else {

pm.test("Response status harus 201 Created", function () {

pm.response.to.have.status(201);

});

console.log(res.json()); // Tambahkan ini untuk memeriksa isi response dari server

pm.test("Data post yang dibuat harus sesuai dengan yang diharapkan", function () {

var jsonData = res.json();

pm.expect(jsonData.title).to.eql("testing post baru");

pm.expect(jsonData.body).to.eql("ini adalah body dari post baru");

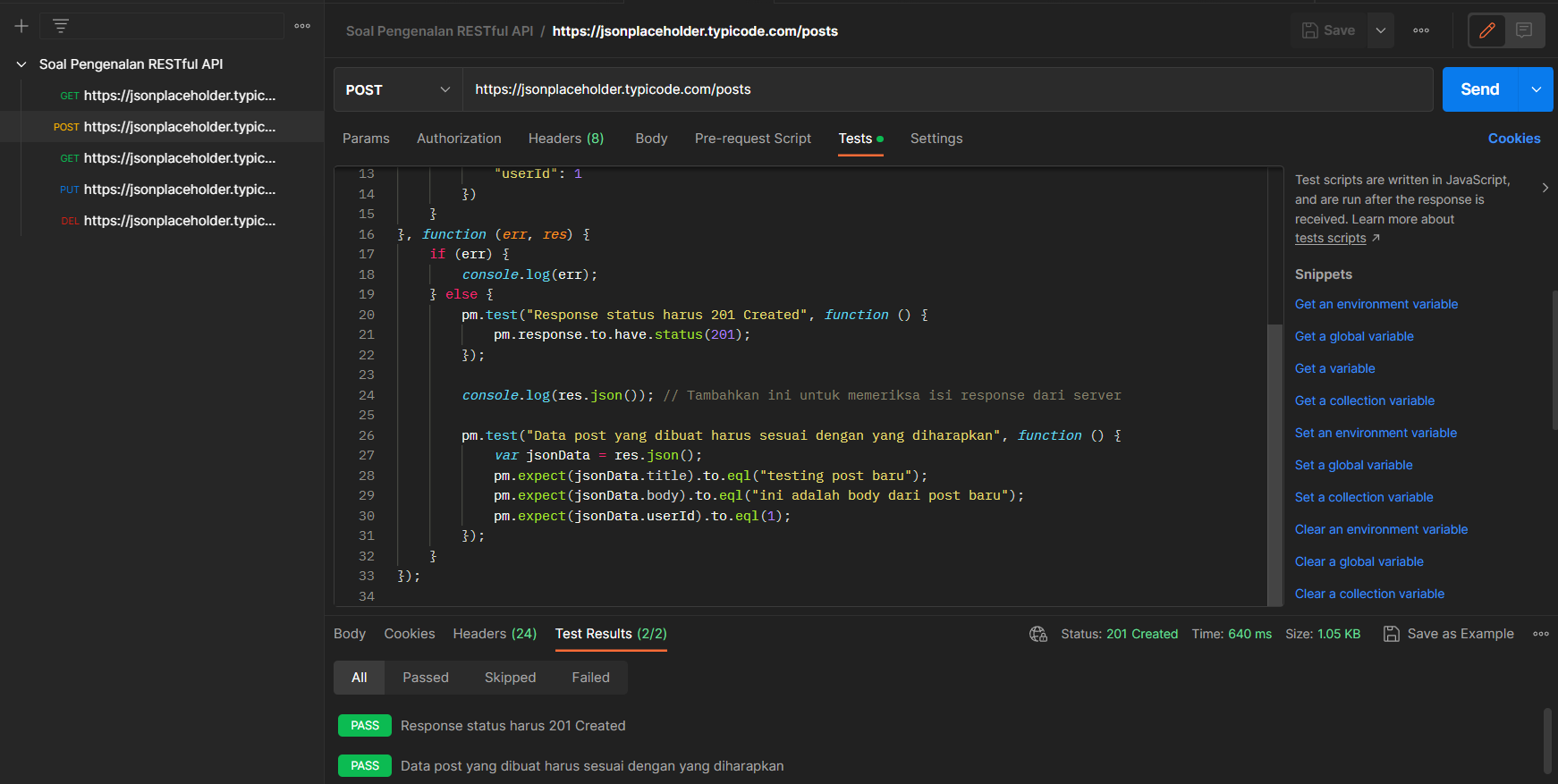
pm.expect(jsonData.userId).to.eql(1);

});

}

});

## Screenshot



* Penjelasan

Pada test case tersebut, terdapat penggunaan method POST dengan mengirimkan data dalam format JSON pada body request. Terdapat pula pengecekan pada response yang dihasilkan oleh server.

Pada bagian callback function setelah pengiriman request, terdapat dua buah test yang dilakukan dengan menggunakan pm.test(). Test pertama dilakukan untuk mengecek apakah response status dari server adalah 201 Created, sedangkan test kedua dilakukan untuk memastikan apakah data post yang telah dibuat oleh server sesuai dengan data yang diharapkan.

Pada test kedua, terdapat beberapa asserstion yang dilakukan menggunakan pm.expect(). Assertion tersebut meliputi pengecekan terhadap title, body, dan userId dari data post yang telah dibuat. Jika semua asserstion berhasil dilakukan, maka dapat dipastikan bahwa pembuatan data post baru pada API endpoint tersebut telah berhasil dilakukan dengan benar.

# 

# Endpoint 3

## Script

pm.test("Status code is 200", function () {

pm.response.to.have.status(200);

});

pm.test("Response has required keys", function () {

pm.response.to.be.ok;

pm.response.to.be.withBody;

pm.response.to.have.jsonBody();

pm.expect(pm.response.json()).to.have.all.keys('userId', 'id', 'title', 'body');

});

pm.test("Response contains correct values", function () {

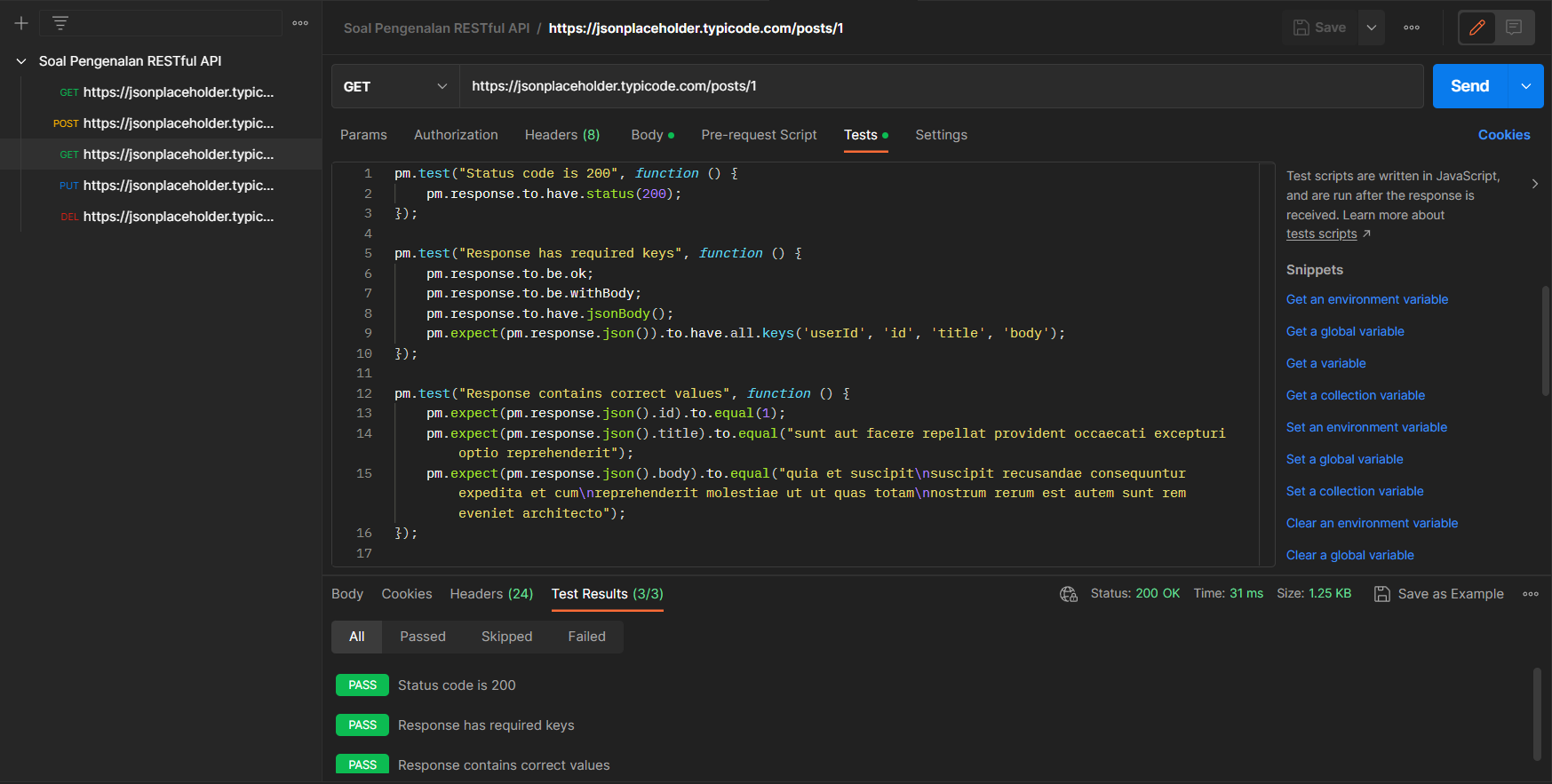
pm.expect(pm.response.json().id).to.equal(1);

pm.expect(pm.response.json().title).to.equal("sunt aut facere repellat provident occaecati excepturi optio reprehenderit");

pm.expect(pm.response.json().body).to.equal("quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto");

});

## Screenshot



* Penjelasan

Pertama-tama, Postman akan menjalankan tiga buah test case. Test case pertama akan memeriksa apakah status respons dari server adalah 200 OK. Test case tersebut menggunakan assertion pm.response.to.have.status(200) untuk memeriksa status respons. Jika status respons bukan 200, maka test case akan gagal dan Postman akan mencetak pesan error.

Test case kedua akan memeriksa apakah respons dari server memiliki semua keys yang dibutuhkan. Test case tersebut menggunakan beberapa assertion untuk memeriksa apakah respons berupa objek JSON (pm.response.to.have.jsonBody()) dan memiliki keys yang dibutuhkan (pm.expect(pm.response.json()).to.have.all.keys('userId', 'id', 'title', 'body')). Jika respons tidak memiliki salah satu keys yang dibutuhkan, maka test case akan gagal dan Postman akan mencetak pesan error.

Test case ketiga akan memeriksa apakah data post yang didapatkan dari respons sudah sesuai dengan yang diharapkan. Test case tersebut menggunakan assertion pm.expect() untuk membandingkan nilai dari beberapa keys di dalam respons dengan nilai yang diharapkan. Jika salah satu assertion gagal, maka test case akan gagal dan Postman akan mencetak pesan error.

Dengan menjalankan testing script tersebut, kita dapat memastikan bahwa API endpoint pada JSONPlaceholder untuk mendapatkan data post dengan ID 1 bekerja dengan benar dan menghasilkan respons yang sesuai dengan yang diharapkan.

# 

# Endpoint 4

## Script

// Test case positif 4: Mengubah data post dengan ID 1 dengan benar

pm.sendRequest({

url: "https://jsonplaceholder.typicode.com/posts/1",

method: "PUT",

header: { "Content-Type": "application/json" },

body: {

mode: "raw",

raw: JSON.stringify({

title: "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",

body: "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto",

userId: 1

})

}

}, function (err, res) {

if (err) {

console.log(err);

} else {

pm.test("Response status harus 200 OK", function () {

pm.response.to.have.status(200);

});

pm.test("Data post yang diubah harus sesuai dengan yang diharapkan", function () {

var jsonData = res.json();

pm.expect(jsonData.title).to.eql("sunt aut facere repellat provident occaecati excepturi optio reprehenderit");

pm.expect(jsonData.body).to.eql("quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto");

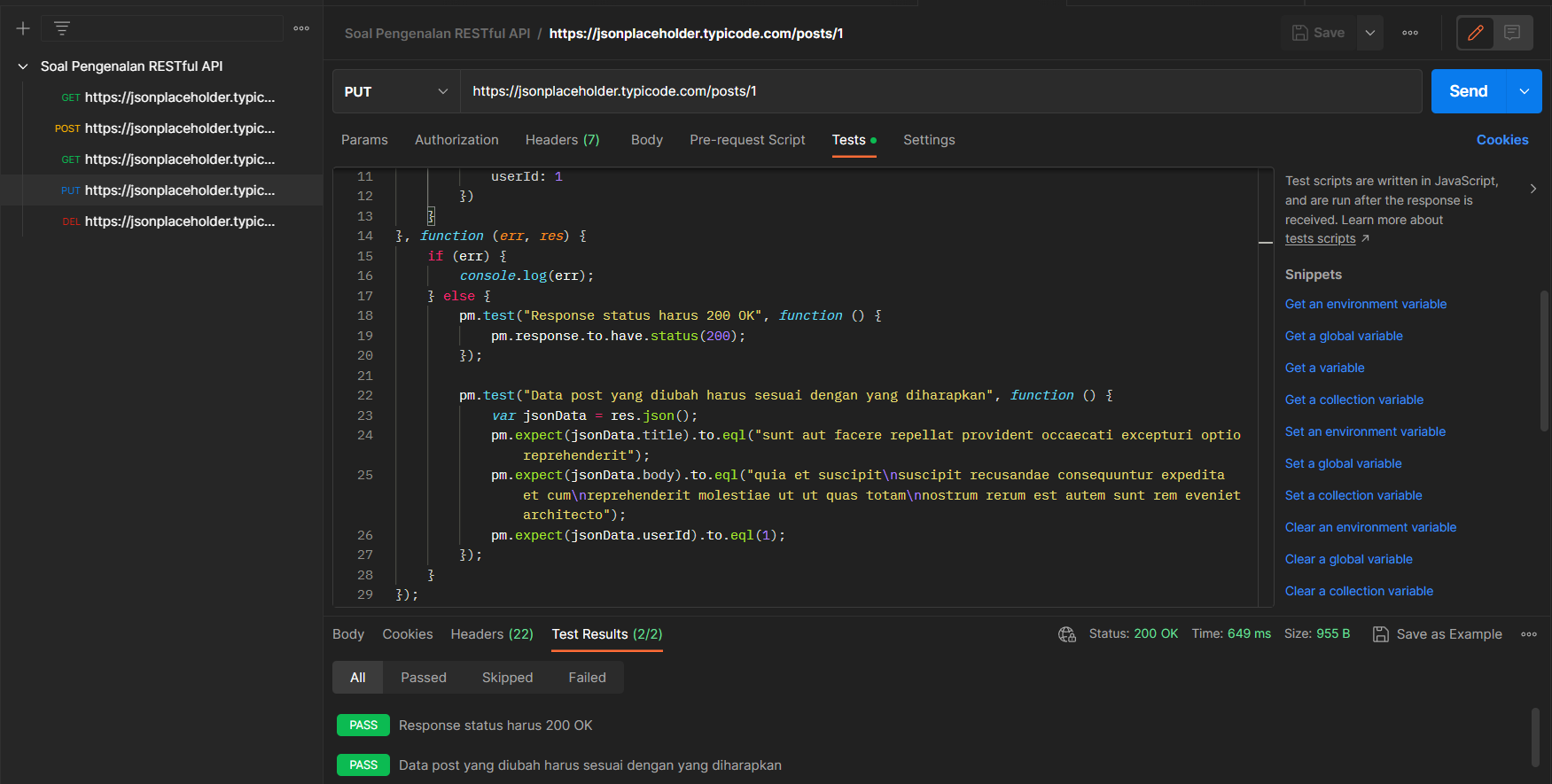
pm.expect(jsonData.userId).to.eql(1);

});

}

});

## Screenshot



* Penjelasan

Pertama-tama, Postman akan mengirimkan sebuah request PUT dengan mengakses URL "https://jsonplaceholder.typicode.com/posts/1" untuk mengubah data post dengan ID 1. Request tersebut juga akan mengirimkan data post yang baru melalui body request dengan menggunakan header "Content-Type" sebagai "application/json".

Setelah request dikirimkan, Postman akan menunggu respons dari server. Setelah mendapatkan respons dari server, Postman akan mengevaluasi apakah ada kesalahan (error) yang terjadi pada saat request dan respons. Jika ada kesalahan, Postman akan mencetak pesan error ke konsol.

Jika tidak ada kesalahan, Postman akan menjalankan dua buah test case. Test case pertama akan memeriksa apakah status respons dari server adalah 200 OK. Test case tersebut menggunakan assertion pm.response.to.have.status(200) untuk memeriksa status respons. Jika status respons bukan 200, maka test case akan gagal dan Postman akan mencetak pesan error.

Test case kedua akan memeriksa apakah data post yang diubah sudah sesuai dengan yang diharapkan. Test case tersebut menggunakan beberapa assertion untuk memeriksa apakah judul (title), isi (body), dan user ID dari data post sudah sesuai dengan yang diharapkan. Jika salah satu atau lebih assertion gagal, maka test case akan gagal dan Postman akan mencetak pesan error.

Dengan menjalankan testing script tersebut, kita dapat memastikan bahwa API endpoint untuk mengubah data post pada JSONPlaceholder bekerja dengan benar dan sesuai dengan yang diharapkan.

# 

# Endpoint 5

## Script

// Test case positif 5: Menghapus data post dengan ID 1 dengan benar

pm.sendRequest({

url: "https://jsonplaceholder.typicode.com/posts/1",

method: "DELETE"

}, function (err, res) {

if (err) {

console.log(err);

} else {

pm.test("Response status harus 200 OK", function () {

pm.response.to.have.status(200);

});

pm.test("Data post dengan ID 1 harus berhasil dihapus", function () {

var jsonData = res.json();

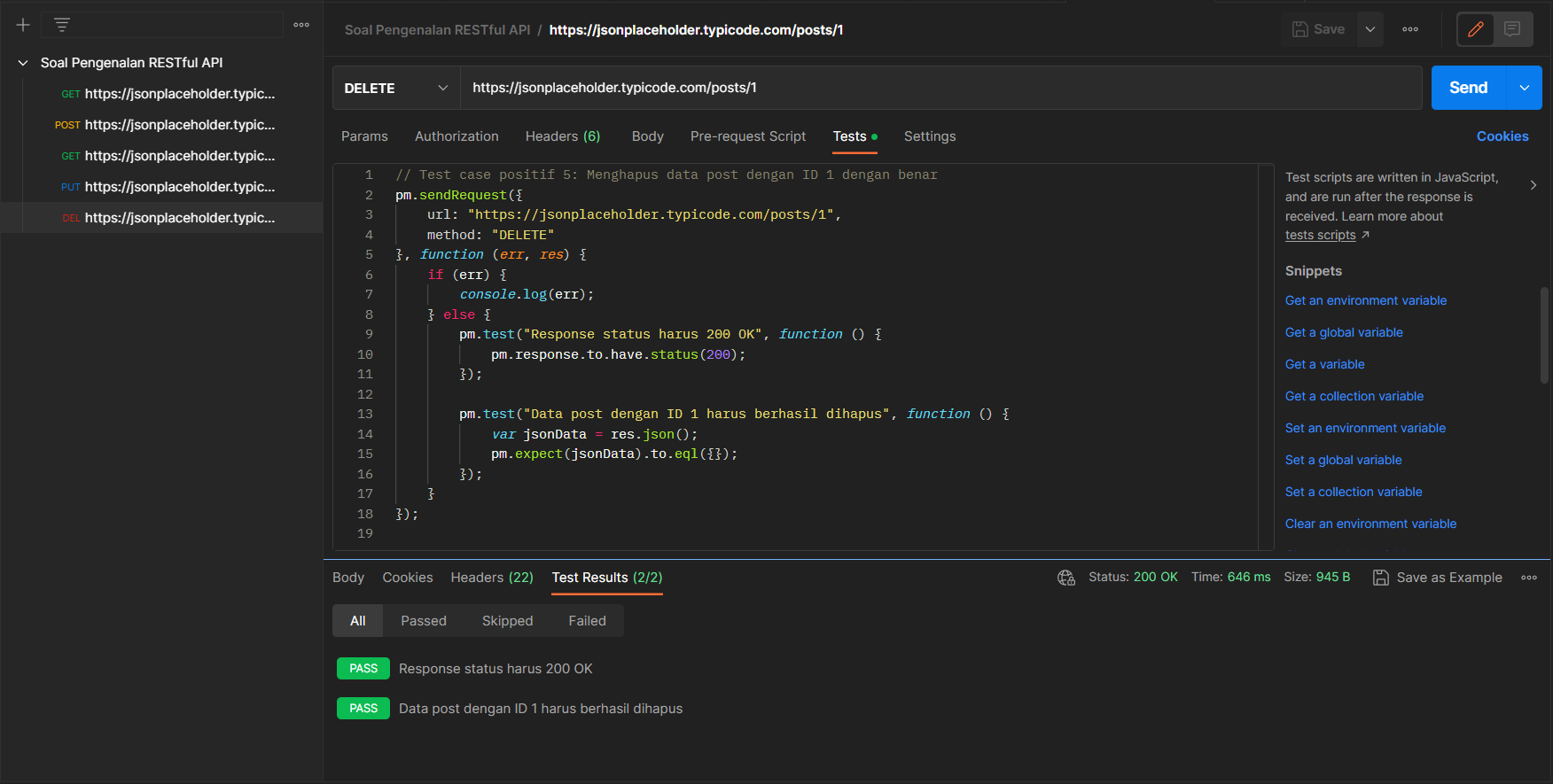
pm.expect(jsonData).to.eql({});

});

}

});

## Screenshot



* Penjelasan

Pertama-tama, Postman akan mengirimkan sebuah request DELETE dengan mengakses URL "https://jsonplaceholder.typicode.com/posts/1" untuk menghapus data post dengan ID 1. Setelah request dikirimkan, Postman akan menunggu respons dari server.

Setelah mendapatkan respons dari server, Postman akan mengevaluasi apakah ada kesalahan (error) yang terjadi pada saat request dan respons. Jika ada kesalahan, Postman akan mencetak pesan error ke konsol. Jika tidak ada kesalahan, Postman akan menjalankan dua buah test case.

Test case pertama akan memeriksa apakah status respons dari server adalah 200 OK. Test case tersebut menggunakan assertion pm.response.to.have.status(200) untuk memeriksa status respons. Jika status respons bukan 200, maka test case akan gagal dan Postman akan mencetak pesan error.

Test case kedua akan memeriksa apakah data post dengan ID 1 berhasil dihapus. Test case tersebut akan mengambil data respons dari server dalam bentuk JSON menggunakan variabel jsonData. Lalu, test case akan memeriksa apakah jsonData sama dengan {} atau tidak menggunakan assertion pm.expect(jsonData).to.eql({}). Jika jsonData tidak sama dengan {}, maka test case akan gagal dan Postman akan mencetak pesan error.

Dengan menjalankan testing script tersebut, kita dapat memastikan bahwa API endpoint untuk menghapus data post pada JSONPlaceholder bekerja dengan benar dan sesuai dengan yang diharapkan.