

# Mini Project

Fery Hafidz Prasetyo - QE

# Data Diri

**Nama : Fery Hafidz Prasetyo**

**Asal : Bekasi**

**Kampus : Telkom University**



# MUP

Test Case Management Tools:

- a. Qase.io

Tools yang digunakan untuk pengujian:

- a. Web Testing : BDD with Cucumber & JMeter
- b. API Testing : REST-assured
- c. Mobile Testing : Katalon Studio & Appium



# Test Case



## Testing WEB-Mini Porject

6 test cases | 4 suites | No active runs



## RESTful API Testing-Mini Project

22 test cases | 14 suites | No active runs



## Mobile Testing - Mini Project

4 test cases | 0 suites | No active runs



# Web Testing

Test Case setiap fitur pada testing Web Mini Book :

**Update book list** + 🖊️ 📁 🗑️

- ↑ ⚡ WEB-5 Updated the book list successfully
- ↑ ⚡ WEB-6 Update book list with invalid data

+ Create quick test

**Create a book list** + 🖊️ 📁 🗑️

- ↑ ⚡ WEB-1 Create New Book List
- ↑ ⚡ WEB-2 Add a new book list with empty data

+ Create quick test

**Show Info Book List** + 🖊️ 📁 🗑️

- ↑ ⚡ WEB-4 View the book list that has been created

+ Create quick test

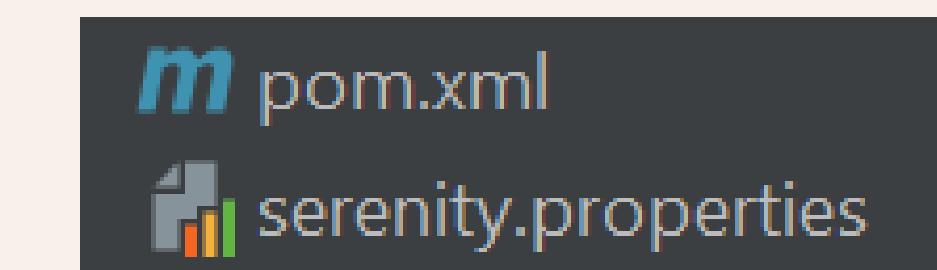
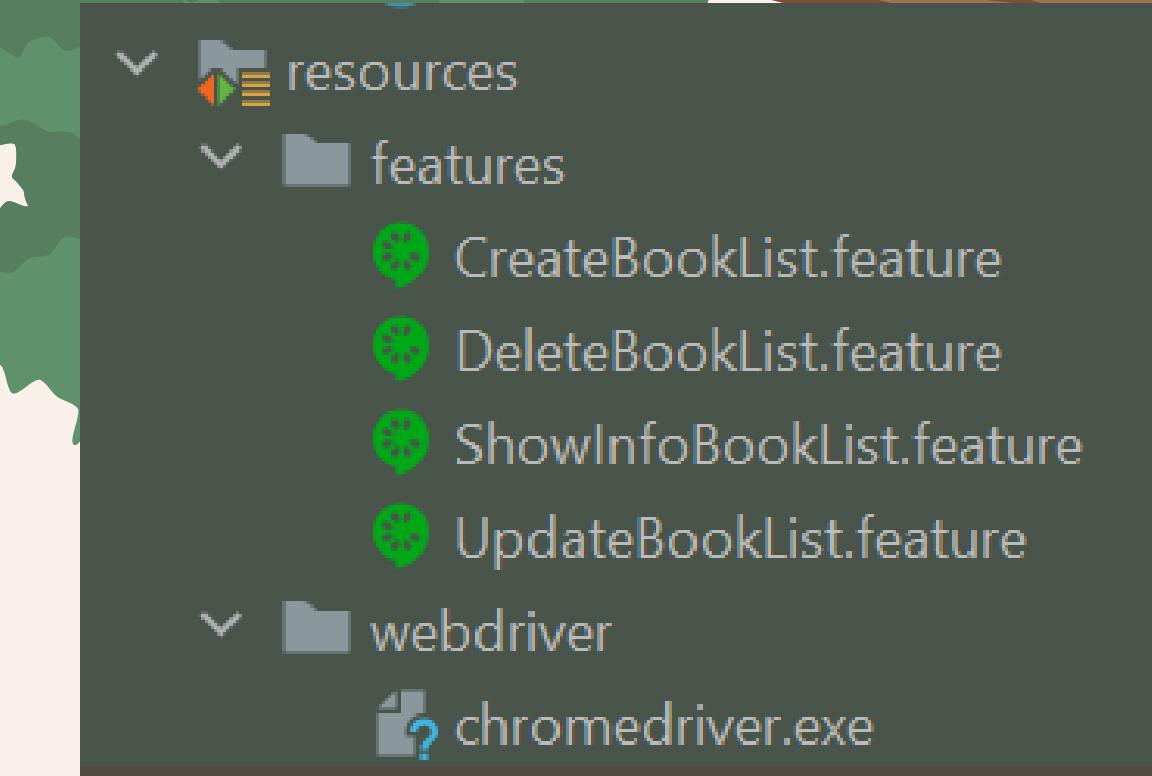
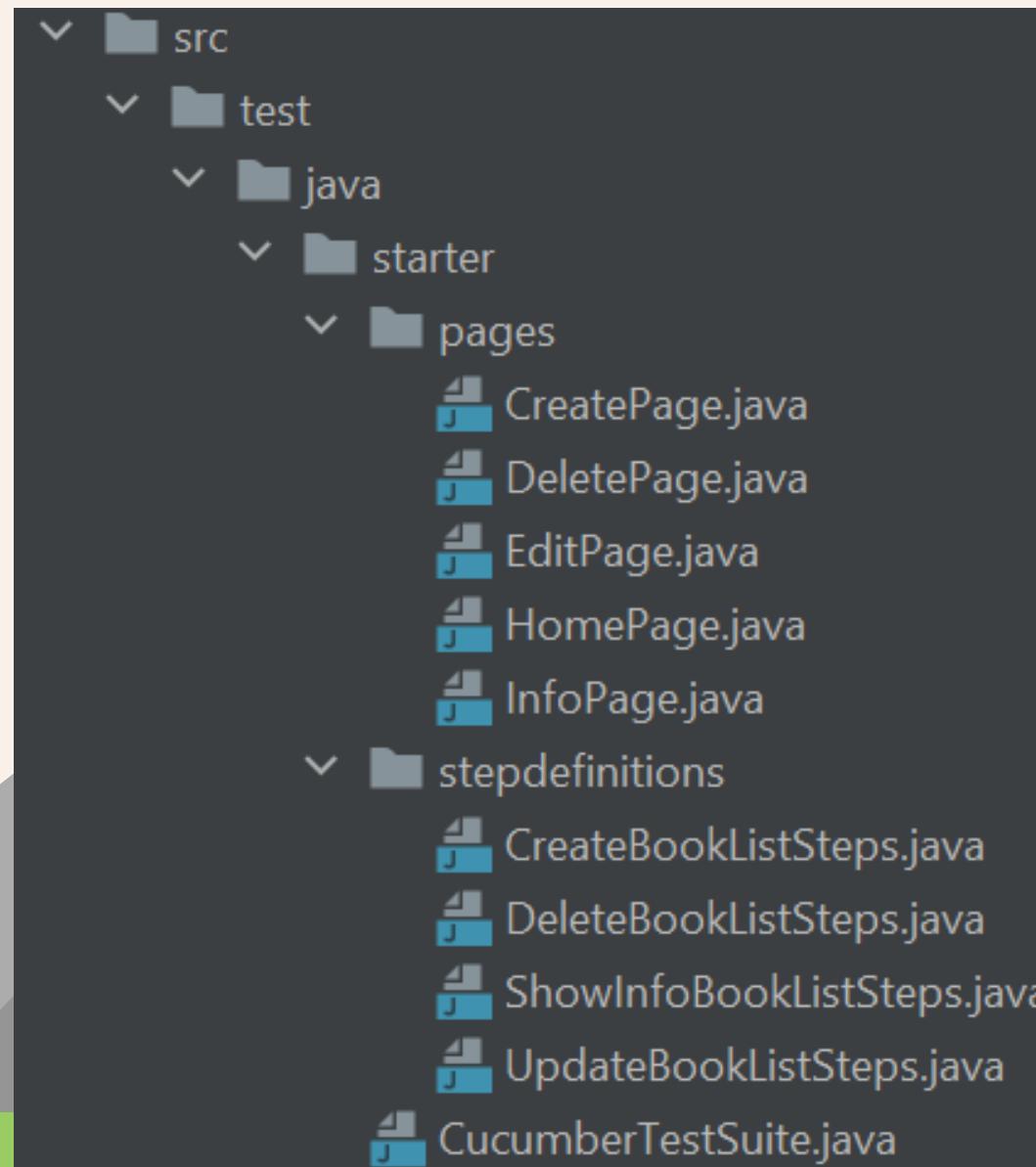
**Delete a Book List** + 🖊️ 📁 🗑️

- ↑ ⚡ WEB-7 Delete the book list

+ Create quick test

# Struktur Folder

Berikut struktur folder WEB Testing:



# Flow/Step

Berikut flow/step dalam WEB Testing:

1

```
list.feature.x  
Feature: Create a Book List  
As a user  
I want to Create New Book List  
So that I can access products page  
  
Scenario: As a user I can create New Book List  
Given I am on the landing page  
When I click the add icon  
And I see create book page  
And I input Title with valid input  
And I input Author with valid input  
And I input Publish Year with valid input  
And I click Save Book button  
And I back to landing page  
Then I saw a message Book Created Successfully
```

2

```
ListSteps.java.x  
@Steps  
HomePage homePage;  
  
@Steps  
CreatePage createPage;  
  
@Given("I am on the landing page")  
public void onTheLandingPage(){  
    homePage.openUrl(absoluteUrl: "https://mini-book-store-fe.vercel.app/")  
    Assertions.assertTrue(homePage.validateOnLandingPage());  
}  
  
@When("I click the add icon")  
public void ClickAddIcon(){ homePage.clickAddNewButton(); }  
@And("I see create book page")  
public void SeeCreateBookPage(){ Assertions.assertTrue(createPage.OnCreateBookPage()); }  
  
@And("I input Title with valid input")  
public void InputTitleCreate(){  
    createPage.ValidInputTitle("Menanam Jagung"); }  
  
@And("I input Author with valid input")  
public void InputAuthorCreate(){  
    createPage.ValidInputAuthor("Herman Gunawan"); }  
  
@And("I input Publish Year with valid input")  
public void InputYearCreate(){  
    createPage.ValidInputYear("2023"); }  
  
@And("I click Save Book button")  
public void ClickSaveBook(){  
    createPage.ClickSaveBook(); }  
  
@Then("I back to landing page")  
public void BackToLandingPage(){  
    homePage.ValidateOnLandingPage(); }  
  
@Then("I saw a message Book Created Successfully")  
public void SuccessMessage(){  
    Assertions.assertTrue(homePage.SuccessMessage()); }
```

3

```
HomePage.java.x  
import net.thucydides.core.pages.PageObject;  
import org.openqa.selenium.By;  
  
public class HomePage extends PageObject {  
    @Step  
    private By title() { return new By.ByXPath(xpathExpression: "//*[@id='root']/div/div[2]/h1"); }  
    @Step  
    private By addNewBookButton() { return new By.ByXPath(xpathExpression: "//*[@id='root']/div/div[2]/a"); }  
    @Step  
    private By infoBookButton() { return new By.ByXPath(xpathExpression: "//*[@id='root']/div/table/tbody/tr"); }  
    @Step  
    private By deleteBookButton() { return new By.ByXPath(xpathExpression: "//*[@id='root']/div/table/tbody/tr"); }  
    @Step  
    private By editBookButton() { return new By.ByXPath(xpathExpression: "//*[@id='root']/div/table/tbody/tr"); }  
    @Step  
    private By SuccessCreatedMessage() { return new By.ByXPath(xpathExpression: "//*[@id='notistack-snackbar']/div"); }  
    @Step  
    private By SuccessDeletedMessage() { return new By.ByXPath(xpathExpression: "//*[@id='root']/div[2]"); }  
  
    @Step  
    public boolean validateOnLandingPage() {  
        try {  
            Thread.sleep(millis: 2100);  
        } catch (InterruptedException e) {  
            e.printStackTrace();  
        }  
        return true;  
    }  
}
```

4

```
webdriver  
chromedriver.exe
```

5

```
properties.x  
webdriver.driver=chrome  
webdriver.chrome.driver=src/test/resources/webdriver/chromedriver.exe  
webdriver.base.url=https://mini-book-store-fe.vercel.app/  
serenity.browser.maximized=true
```



# Testing Web with JMeter

HTTP Request

Name: Mini Book Store

Comments:

Basic Advanced

Web Server

Protocol [http]: https Server Name or IP: mini-book-store-fe.vercel.app Port Number:

HTTP Request

GET Path: / Content encoding:

Redirect Automatically  Follow Redirects  Use KeepAlive  Use multipart/form-data  Browser-compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?
-------	-------	-------------	--------------	-----------------

Thread Group

Name: Thread Group

Comments:

Action to be taken after a Sampler error

Continue  Start Next Thread Loop  Stop Thread  Stop Test  Stop Test Now

Thread Properties

Number of Threads (users): 900

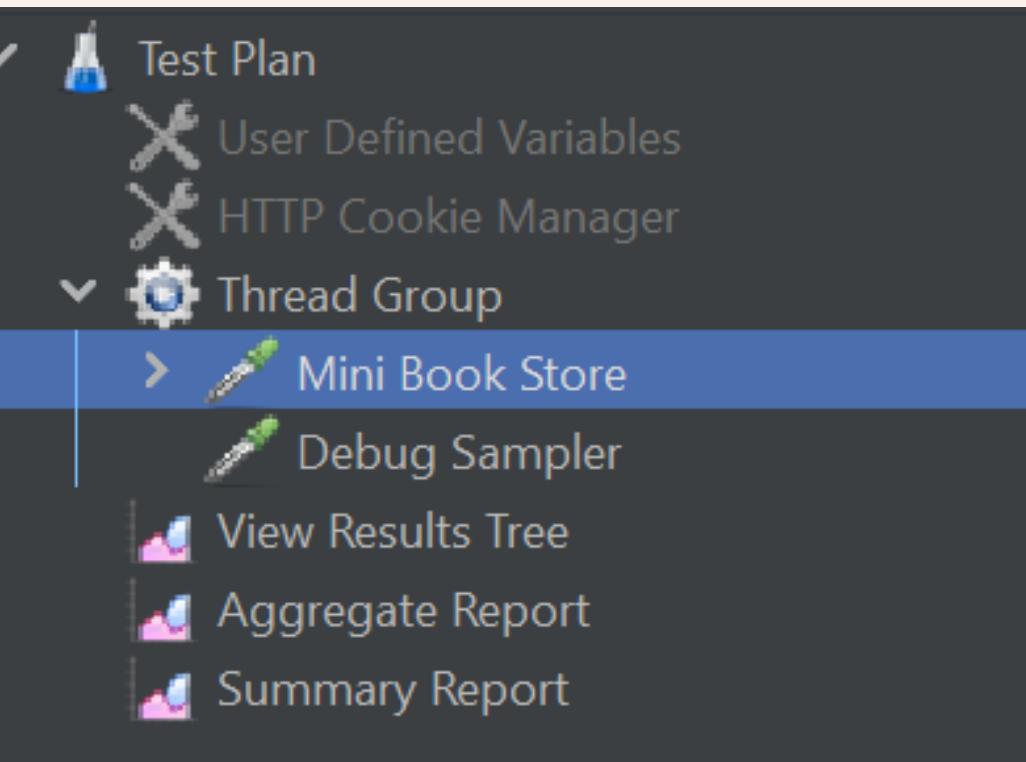
Ramp-up period (seconds): 2

Loop Count:  Infinite 1

Same user on each iteration

Delay Thread creation until needed

Specify Thread lifetime



# API Testing

Test Case setiap fitur pada testing API Alta-shop :

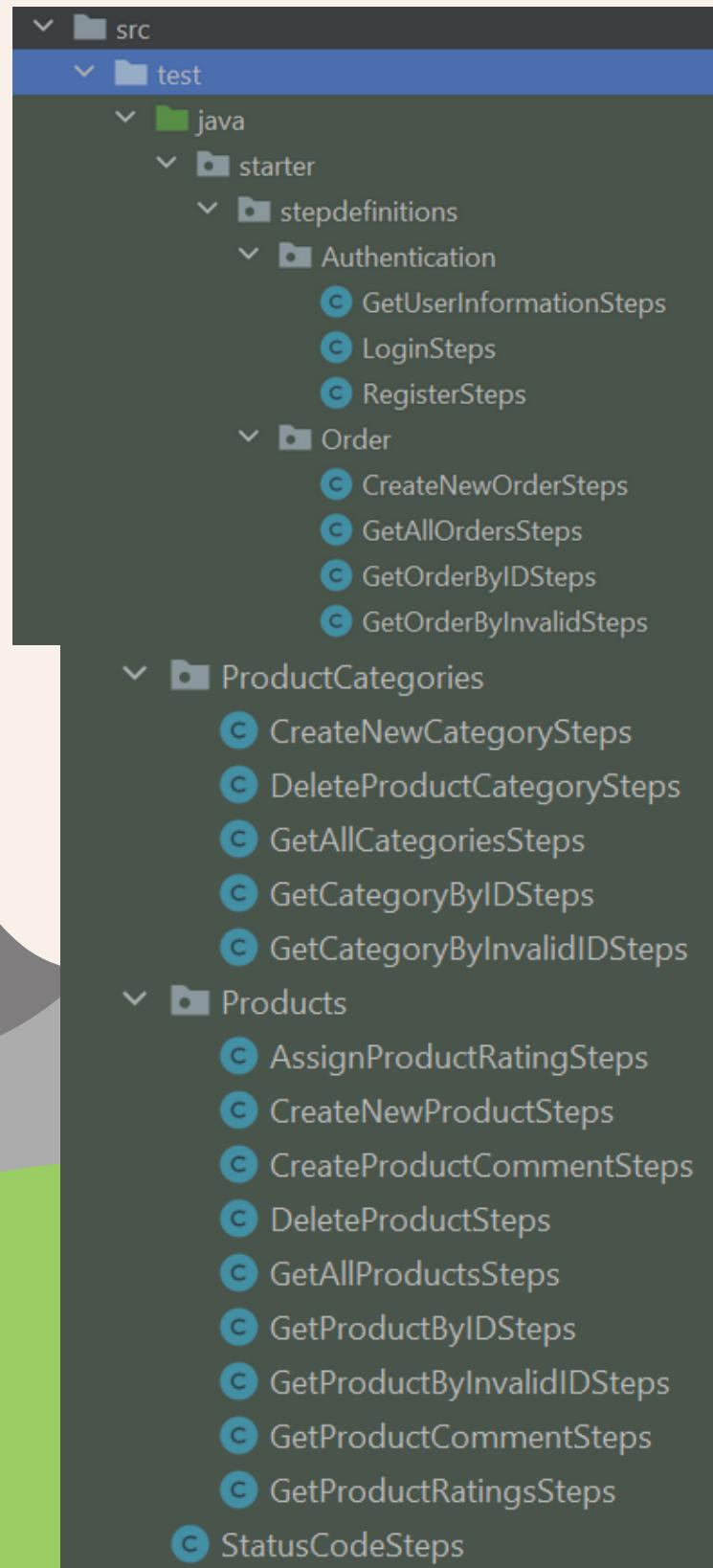
Product	+
	+ Create quick test
GET-Product	+
↑  API-1 Get all products	
↑  API-5 Admin get data product by ID	
↑  API-13 Admin get product by invalid ID	
○  API-14 Admin can get product ratings	
○  API-15 Admin can get products comments	
	+ Create quick test
DEL-Product	+
↑  API-3 Admin can delete product	
	+ Create quick test
POST-Product	+
↑  API-2 Admin successfully created a new product	
↑  API-16 Admin can assign a product rating	
—  API-17 Admin can create a comment for product	
	+ Create quick test

Product Categories	+
	+ Create quick test
POST-PC	+
↑  API-4 Admin can create new product category	
	+ Create quick test
GET-PC	+
↑  API-6 GET Category by ID	
↑  API-18 Admin can get product category by invalid ID	
↑  API-19 Admin can get all categories	
	+ Create quick test
DEL-PC	+
↑  API-7 Delete a category	
	+ Create quick test

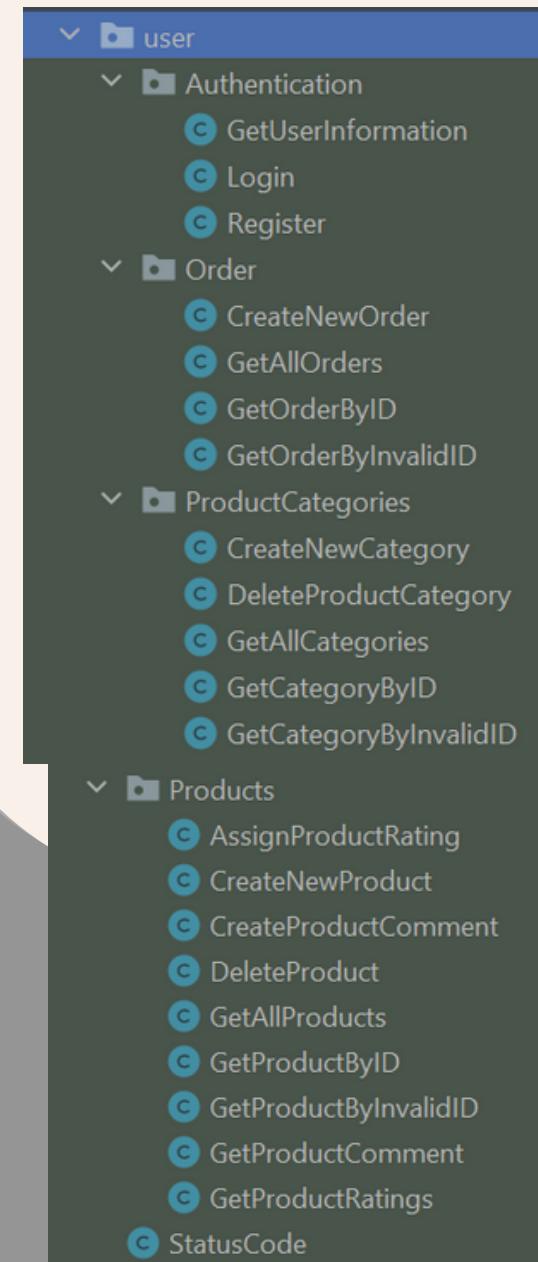
Authentication	+
	+ Create quick test
GET-AUTH	+
—  API-10 Admin get user information	
	+ Create quick test
POST-AUTH	+
↑  API-8 Admin can register account	
↑  API-9 Admin can log in with a registered account	
—  API-20 Admin can register with invalid data (field blank)	
	+ Create quick test
Orders	+
	+ Create quick test
POST-Orders	+
↑  API-11 Admin can create new order	
	+ Create quick test
GET-Orders	+
—  API-12 Admin can get order by ID	
↑  API-21 Admin can get order by invalid ID	
↑  API-22 Admin can get all order	
	+ Create quick test

# Struktur Folder

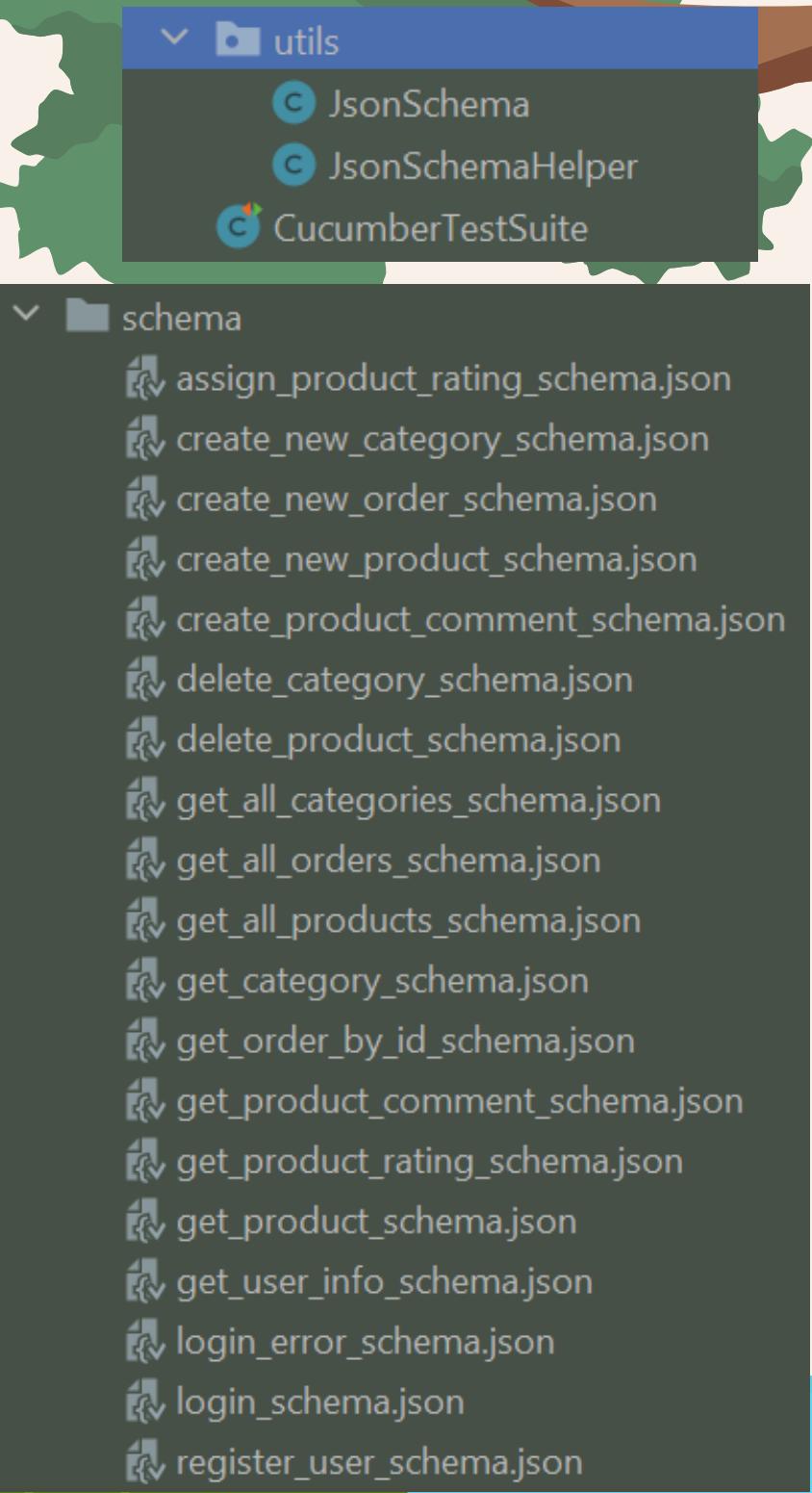
Berikut struktur folder API Testing:



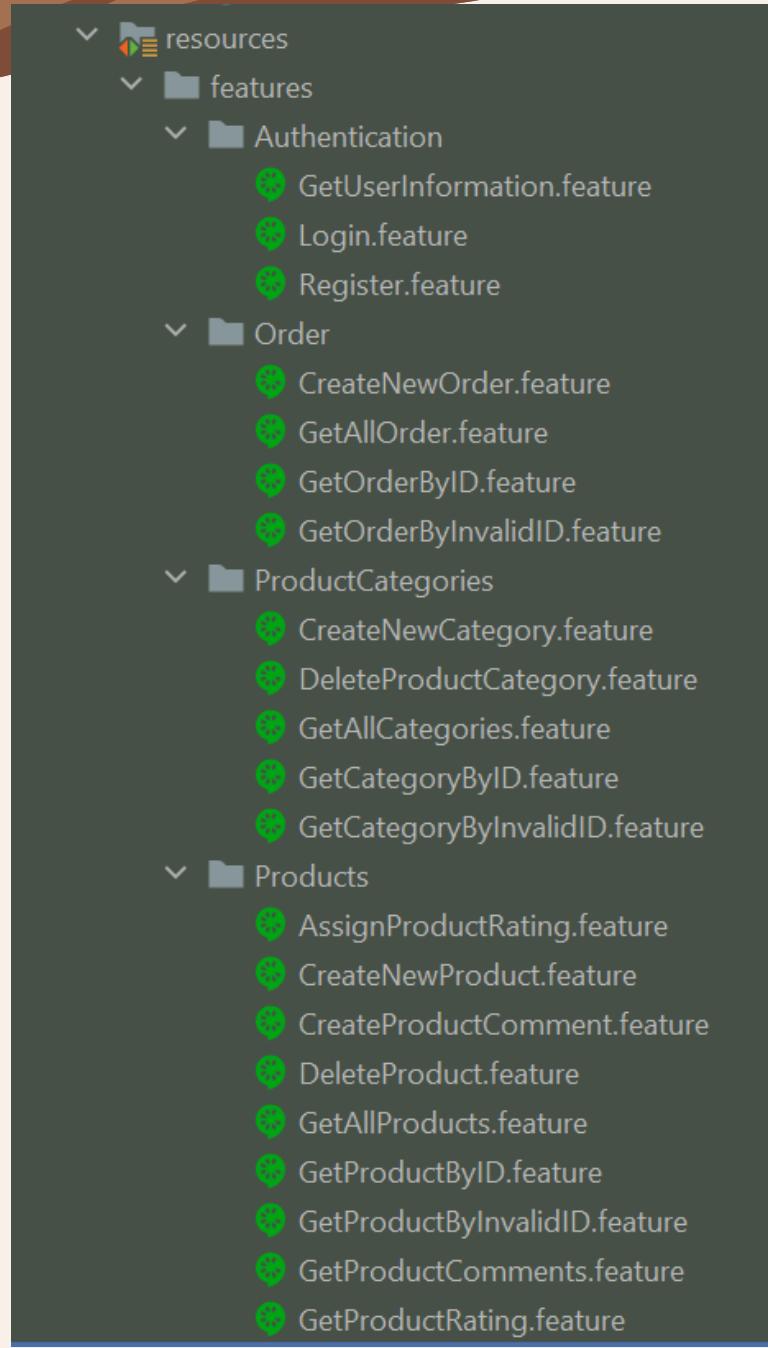
```
src
└── test
    ├── java
    │   └── starter
    │       ├── stepdefinitions
    │       │   ├── Authentication
    │       │       ├── GetUserInformationSteps
    │       │       ├── LoginSteps
    │       │       ├── RegisterSteps
    │       └── Order
    │           ├── CreateNewOrderSteps
    │           ├── GetAllOrdersSteps
    │           ├── GetOrderByIDSteps
    │           └── GetOrderByInvalidSteps
    └── ProductCategories
        ├── CreateNewCategorySteps
        ├── DeleteProductCategorySteps
        ├── GetAllCategoriesSteps
        ├── GetCategoryByIDSteps
        └── GetCategoryByInvalidIDSteps
    └── Products
        ├── AssignProductRatingSteps
        ├── CreateNewProductSteps
        ├── CreateProductCommentSteps
        ├── DeleteProductSteps
        ├── GetAllProductsSteps
        ├── GetProductByIDSteps
        ├── GetProductByInvalidIDSteps
        ├── GetProductCommentSteps
        └── GetProductRatingsSteps
    └── StatusCodeSteps
```



```
user
└── Authentication
    ├── GetUserInformation
    ├── Login
    └── Register
└── Order
    ├── CreateNewOrder
    ├── GetAllOrders
    ├── GetOrderByID
    └── GetOrderByInvalidID
└── ProductCategories
    ├── CreateNewCategory
    ├── DeleteProductCategory
    ├── GetAllCategories
    ├── GetCategoryByID
    └── GetCategoryByInvalidID
└── Products
    ├── AssignProductRating
    ├── CreateNewProduct
    ├── CreateProductComment
    ├── DeleteProduct
    ├── GetAllProducts
    ├── GetProductByID
    ├── GetProductByInvalidID
    ├── GetProductComment
    └── GetProductRatings
    └── StatusCode
```



```
utils
└── JsonSchema
    ├── JsonSchemaHelper
    └── CucumberTestSuite
schema
└── assign_product_rating_schema.json
    ├── create_new_category_schema.json
    ├── create_new_order_schema.json
    ├── create_new_product_schema.json
    ├── create_product_comment_schema.json
    ├── delete_category_schema.json
    ├── delete_product_schema.json
    ├── get_all_categories_schema.json
    ├── get_all_orders_schema.json
    ├── get_all_products_schema.json
    ├── get_category_schema.json
    ├── get_order_by_id_schema.json
    ├── get_product_comment_schema.json
    ├── get_product_rating_schema.json
    ├── get_product_schema.json
    ├── get_user_info_schema.json
    ├── login_error_schema.json
    ├── login_schema.json
    └── register_user_schema.json
```



```
resources
└── features
    ├── Authentication
    │   ├── GetUserInformation.feature
    │   ├── Login.feature
    │   └── Register.feature
    └── Order
        ├── CreateNewOrder.feature
        ├── GetAllOrder.feature
        ├── GetOrderByID.feature
        └── GetOrderByInvalidID.feature
    └── ProductCategories
        ├── CreateNewCategory.feature
        ├── DeleteProductCategory.feature
        ├── GetAllCategories.feature
        ├── GetCategoryByID.feature
        └── GetCategoryByInvalidID.feature
    └── Products
        ├── AssignProductRating.feature
        ├── CreateNewProduct.feature
        ├── CreateProductComment.feature
        ├── DeleteProduct.feature
        ├── GetAllProducts.feature
        ├── GetProductByID.feature
        └── GetProductByInvalidID.feature
    └── StatusCodes
        ├── GetProductComments.feature
        └── GetProductRating.feature
```

# Flow/Step

Berikut flow/step dalam API Testing:

1

```
signProductRating.feature x
@AssignProductRating
Feature: Assign a Product Rating
As an admin
I want to assign a product rating
So that the product rating is saved
Scenario: As an admin I can assign a product rating
  Given I set ratings API endpoint
  When I send POST request body for assign a product rating
  Then I receive status code 200 OK response
```

2

```
AssignProductRatingSteps.java x
package starter.stepdefinitions.Products;

import ...

public class AssignProductRatingSteps {
    @Steps
    AssignProductRating assignProductRating;

    @Given("I set ratings API endpoint")
    public void setApiNewRate() { assignProductRating.setApiRating(); }

    @When("I send POST request body for assign a product rating")
    public void sendRating() { assignProductRating.sendAssignRating(); }

    @And("I receive detailed products with ratings that have been assigned")
    public void receiveRating() { assignProductRating.receiveRating(); }
}
```

3

```
AssignProductRating.java x
package starter.user.Products;

import ...

public class AssignProductRating {
    private static String url = "https://altashop-api.fly.dev/api/";

    @Step("I set ratings API endpoint")
    public String setApiRating() { return url + "products/88701/ratings"; }

    @Step("I send POST request body for assign a product rating")
    public void sendAssignProductRating() {
        JSONObject requestBody = new JSONObject();
        requestBody.put("count", 4);

        String token = "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJGdWxsbmFtZSI";
        SerenityRest.given()
    }
}
```

4

```
get_product_rating_schema.json x
{
  "$schema": "http://json-schema.org/draft-06/schema#",
  "$ref": "#/definitions/Welcome",
  "definitions": {
    "Welcome": {
      "type": "object",
      "additionalProperties": false,
      "properties": {
        "data": {
          "$ref": "#/definitions/Data"
        },
        "required": [
          "data"
        ],
        "title": "Welcome"
      }
    }
  }
}
```

5

```
JsonSchema.java x
package starter.utils;

public class JsonSchema {

    public static final String GET_ALL_PRODUCT_SCHEMA = "schema/get_all_products_schema.json";
    public static final String GET_PRODUCT_RATINGS_SCHEMA = "schema/get_product_rating_schema.json";
    public static final String REGISTER_USER_SCHEMA = "schema/register_user_schema.json";
    public static final String CREATE_NEW_CATEGORY_SCHEMA = "schema/create_new_category_schema.json";
    public static final String GET_PRODUCT_CATEGORY_BY_ID_SCHEMA = "schema/get_category_schema.json";
    public static final String GET_ALL_CATEGORIES_SCHEMA = "schema/get_all_categories_schema.json";
    public static final String GET_ALL_ORDERS_SCHEMA = "schema/get_all_orders_schema.json";
    public static final String CREATE_NEW_ORDER_SCHEMA = "schema/create_new_order_schema.json";
    public static final String GET_PRODUCT_COMMENTS_SCHEMA = "schema/get_product_comment_schema.json";
    public static final String GET_PRODUCT_SCHEMA = "schema/get_product_schema.json";
    public static final String CREATE_NEW_PRODUCT_SCHEMA = "schema/create_new_product_schema.json";
    public static final String ASSIGN_RATING_SCHEMA = "schema/assign_product_rating_schema.json";
    public static final String CREATE_COMMENT_SCHEMA = "schema/create_product_comment_schema.json";
    public static final String GET_USER_INFO_SCHEMA = "schema/get_user_info_schema.json";
    public static final String LOGIN_SCHEMA = "schema/login_schema.json";
    public static final String GET_ORDER_BY_ID_SCHEMA = "schema/get_order_by_id_schema.json";
    public static final String DELETE_CATEGORY_SCHEMA = "schema/delete_category_schema.json";
}
```



# Mobile Testing

Test Case setiap fitur pada testing Mobile Shopping List App :

MblTesting repository 4 cases (4) | 1 suite (1)

+ Suite + Case Search Add filter

Suites

Add New Item 4 ...

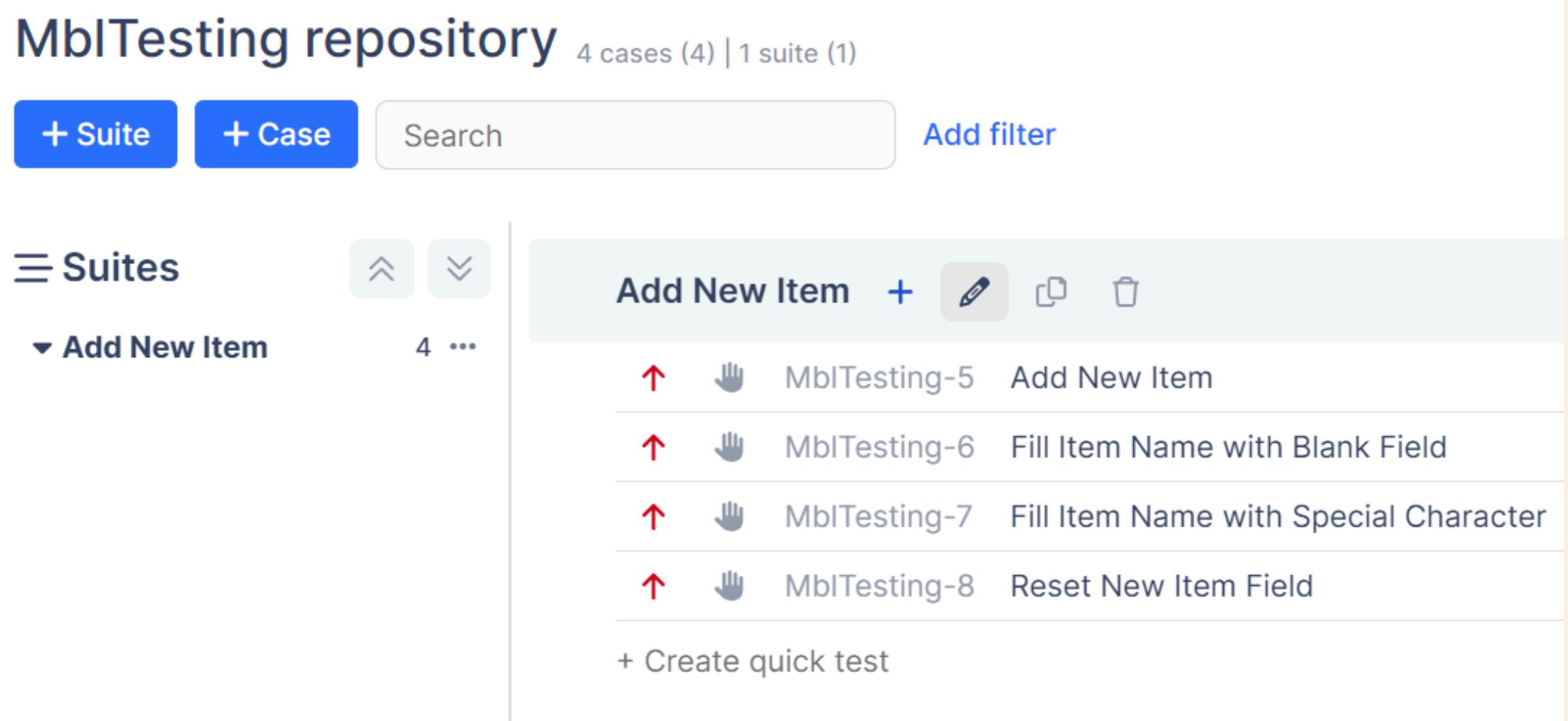
Add New Item + ↗ ⌂ MblTesting-5 Add New Item

↗ ⌂ MblTesting-6 Fill Item Name with Blank Field

↗ ⌂ MblTesting-7 Fill Item Name with Special Character

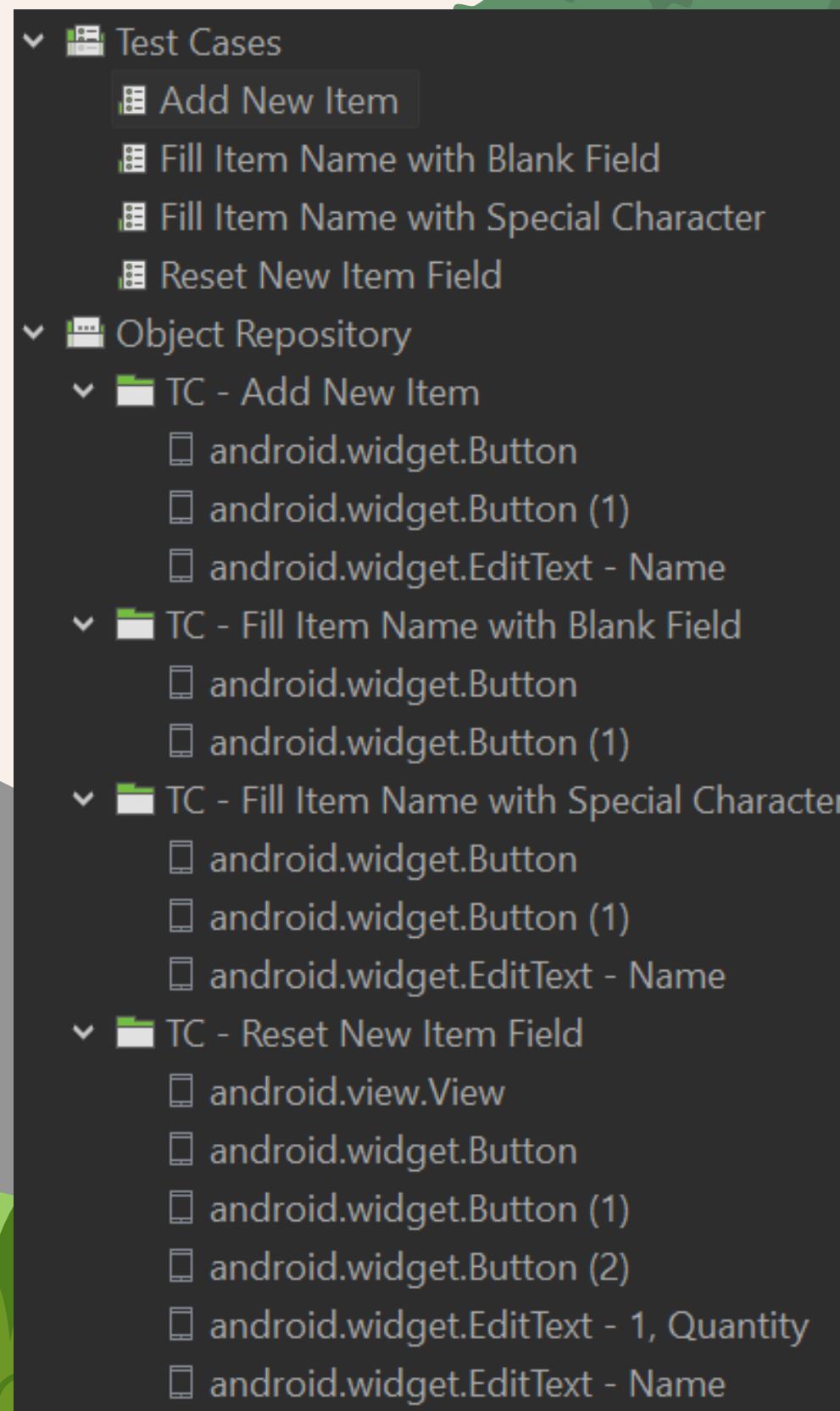
↗ ⌂ MblTesting-8 Reset New Item Field

+ Create quick test



# Struktur Folder

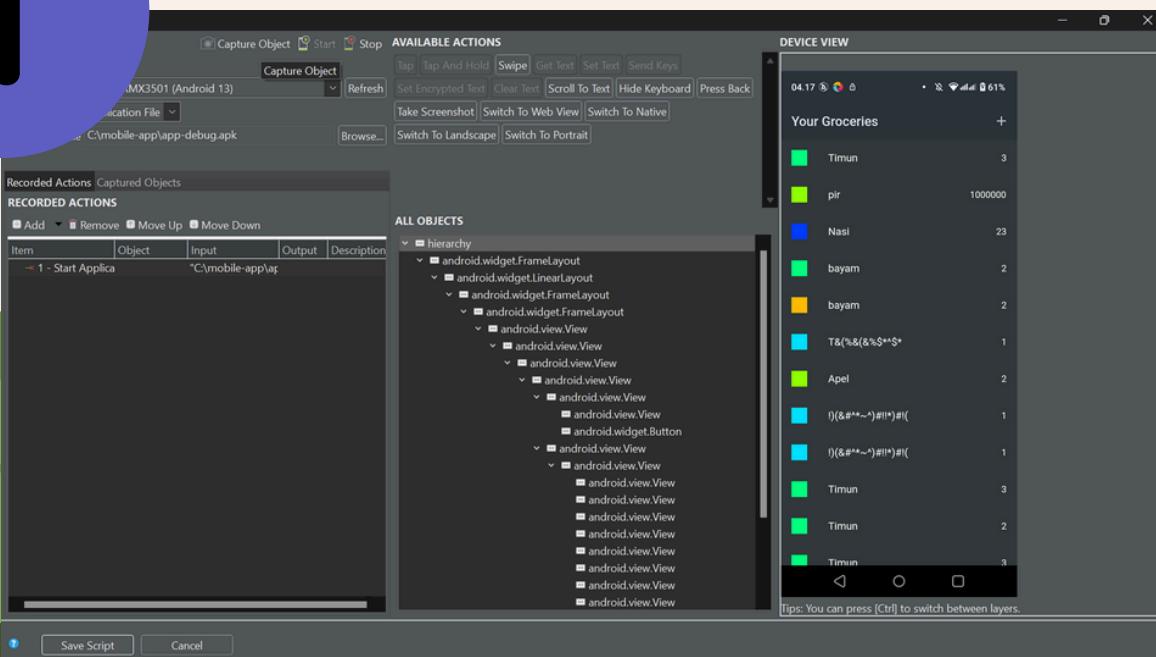
Berikut struktur folder Mobile Testing:



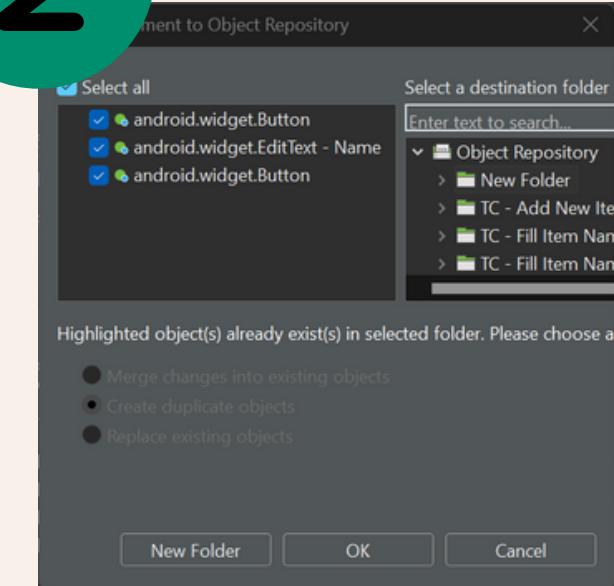
# Flow/Step

Berikut flow/step dalam Mobile Testing:

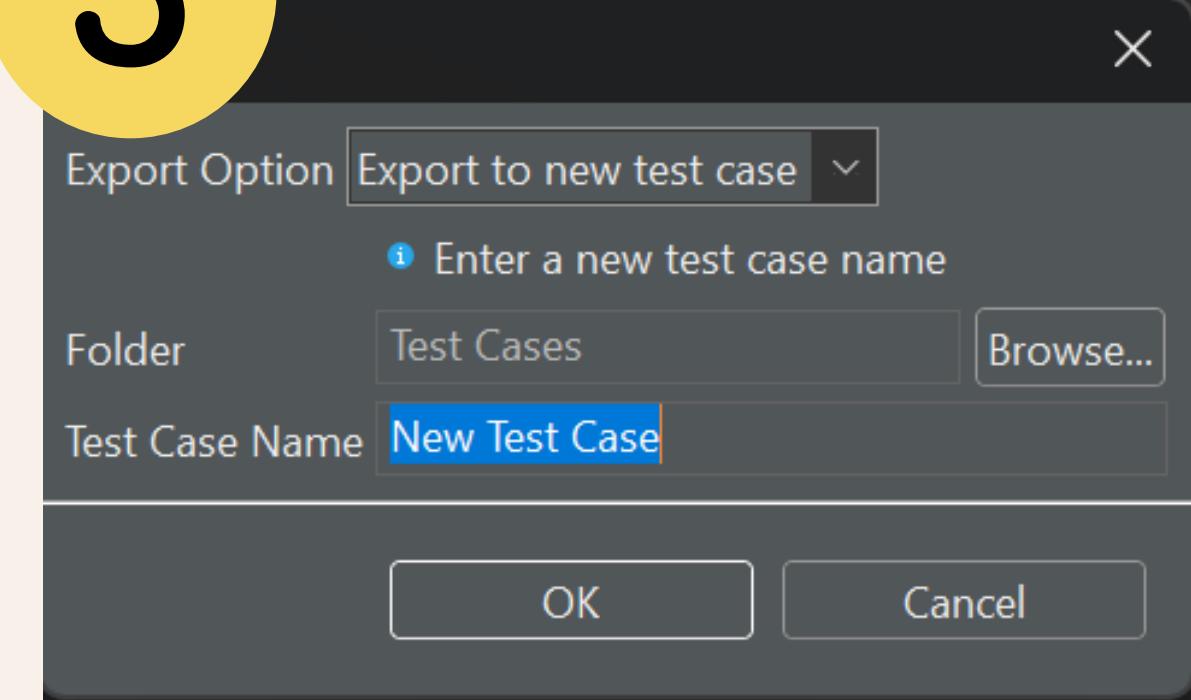
1



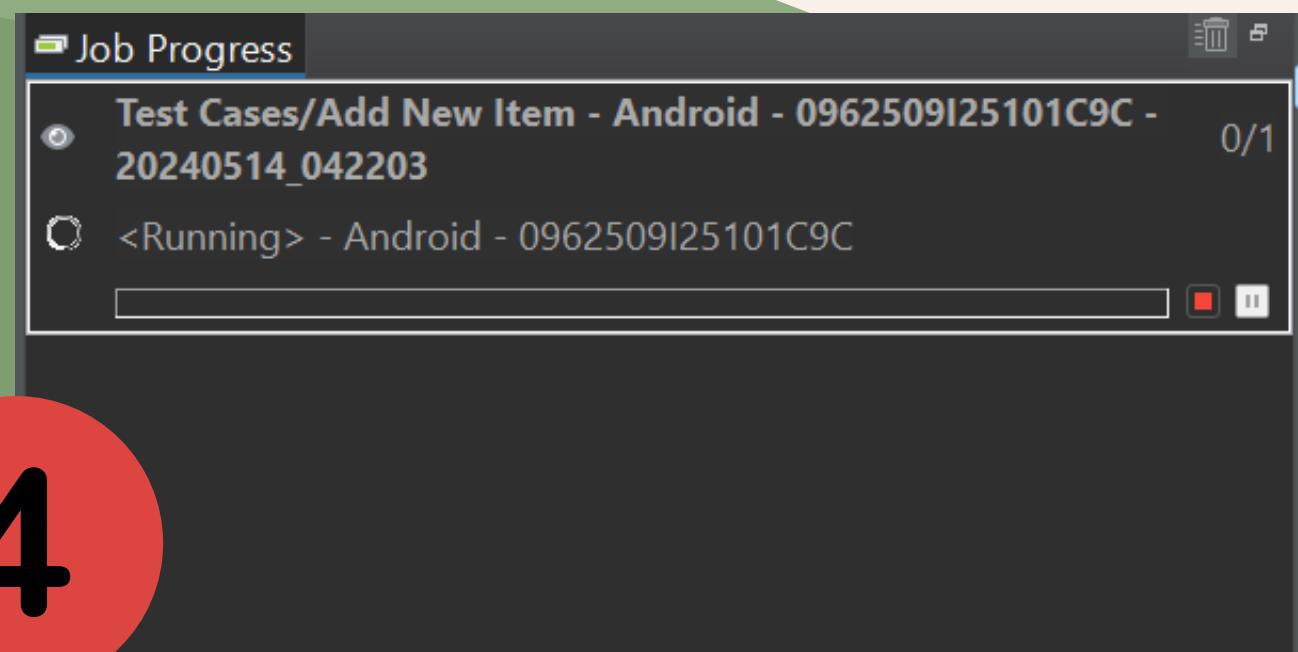
2



3



4



# Hasil Web Testing

## Serenity BDD

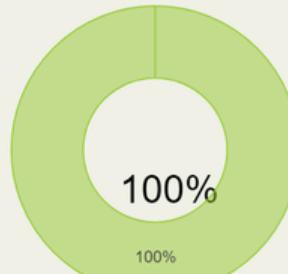


### Test Results: All Tests

6 tests

Summary Test Results

Overview

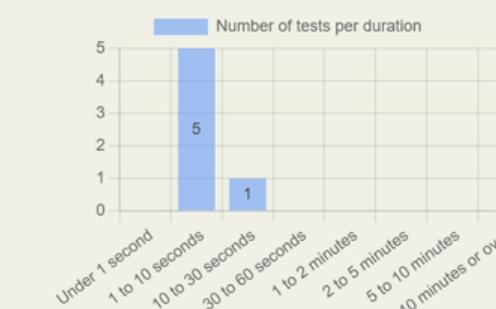


Passing Test Cases  
Pending Test Cases  
Ignored Test Cases  
Aborted Test Cases  
Broken Test Cases  
Compromised Test Cases  
Undefined Test Cases

Test Outcomes



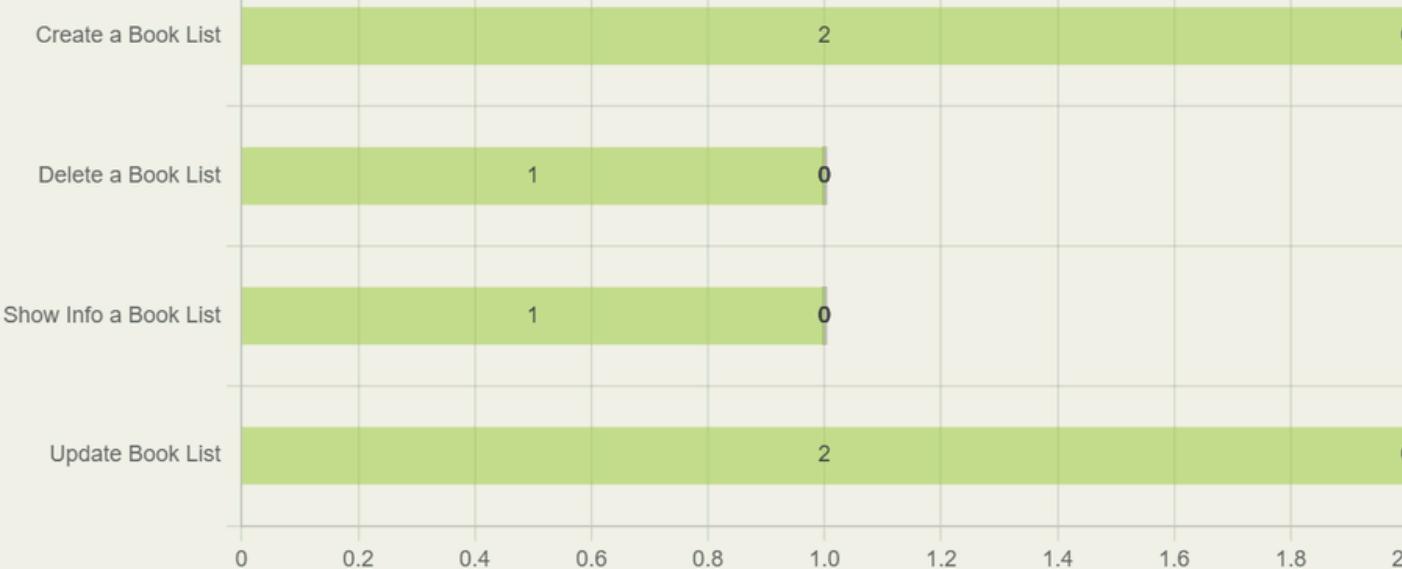
Test Performance



### Functional Coverage Overview

#### Features

Passing Tests Pending Tests Ignored Tests Skipped Tests Aborted Tests  
Failed Tests Errors Compromised Tests



# Hasil Web Testing

JMeter

**View Results Tree**

Name: View Results Tree  
Comments:  
- Write results to file / Read from file  
Filename  Browse... Log/Display Only:  Errors  Successes Configure  
Search:  Case sensitive  Regular exp. Search Reset  
Text Sampler result Request Response data

Sampler result Request Response data

Thread Name: Thread Group 1-773  
Sample Start: 2024-05-14 04:32:09 WIB  
Load time: 1523  
Connect Time: 1311  
Latency: 1523  
Size in bytes: 953  
Sent bytes: 131  
Headers size in bytes: 494  
Body size in bytes: 459  
Sample Count: 1  
Error Count: 0  
Data type ("text"|"bin"|""): text  
Response code: 200  
Response message: OK

**Summary Report**

Name: Summary Report  
Comments:  
- Write results to file / Read from file  
Filename  Browse... Log/Display Only:  Errors  Successes Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Through...	Received...	Sent KB/...	Avg. Bytes
Mini Bo...	900	2546	84	18943	2515.26	0.00%	47.5/sec	44.19	6.08	952.8
Debug S...	900	3	0	180	15.29	0.00%	53.9/sec	162.46	0.00	3086.7
TOTAL	1800	1274	0	18943	2186.54	0.00%	95.0/sec	187.34	6.08	2019.8

**Aggregate Report**

Name: Aggregate Report  
Comments:  
- Write results to file / Read from file  
Filename  Browse... Log/Display Only:  Errors  Successes Configure

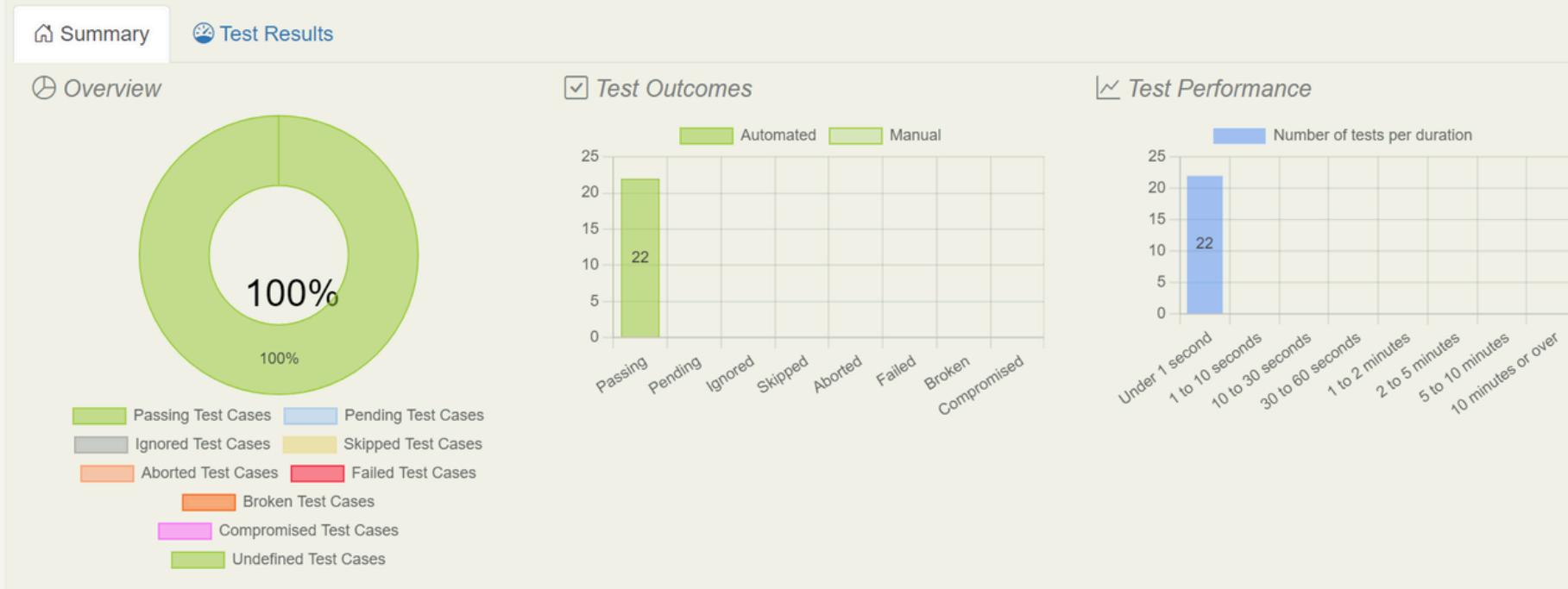
Label	# Sam...	Average	Median	90% Li...	95% Li...	99% Li...	Min	Maxim...	Error %	Throu...	Receiv...	Sent K...
ok Store	900	2546	1818	4828	6276	14300	84	18943	0.00%	47.5/sec	44.19	6.08
Debug...	900	3	1	3	6	41	0	180	0.00%	53.9/sec	162.46	0.00
TOTAL	1800	1274	112	3375	4828	13265	0	18943	0.00%	95.0/sec	187.34	6.08

# Hasil API Testing



## Test Results: All Tests

22 tests

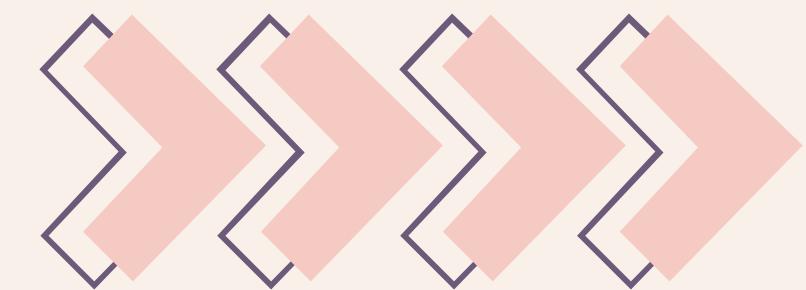
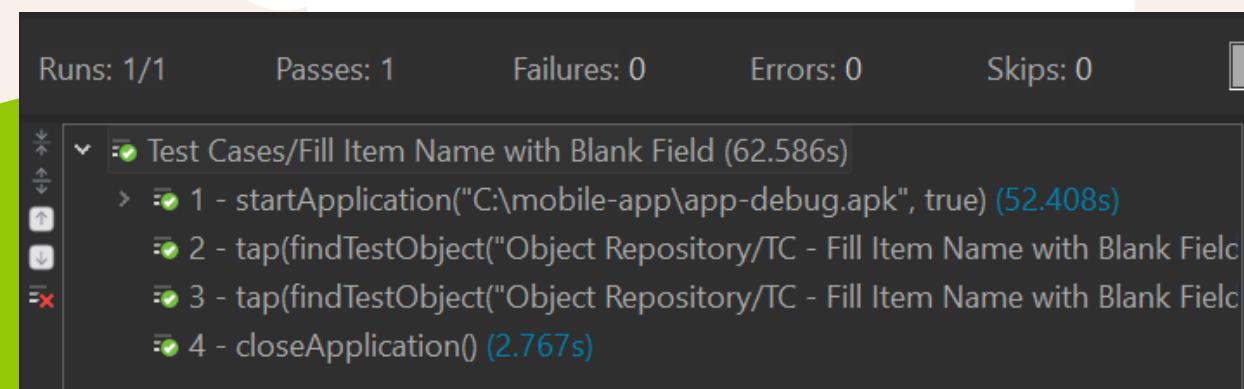
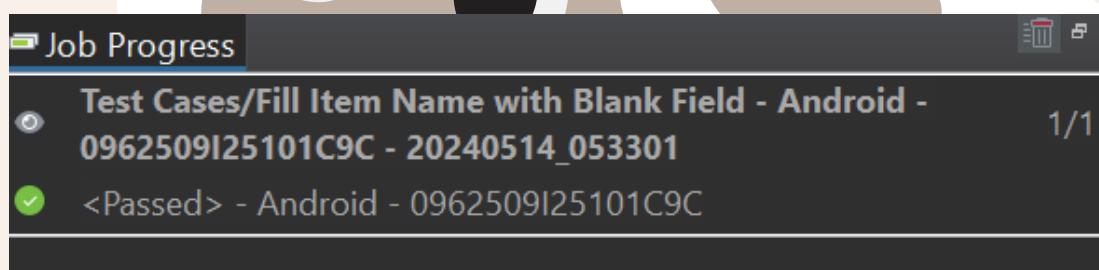
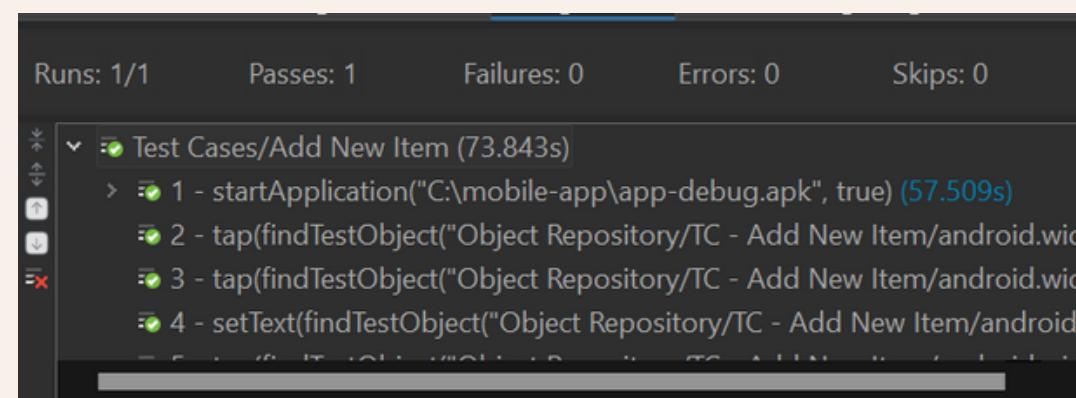
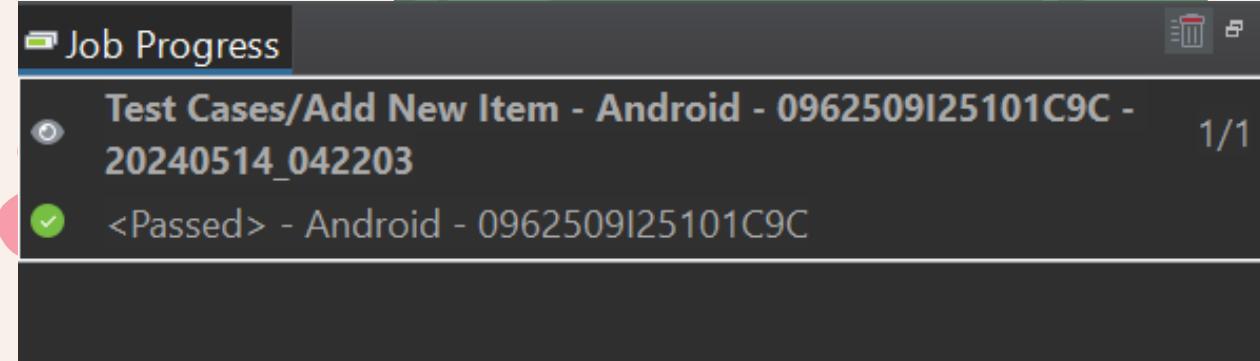


## II. Functional Coverage Overview

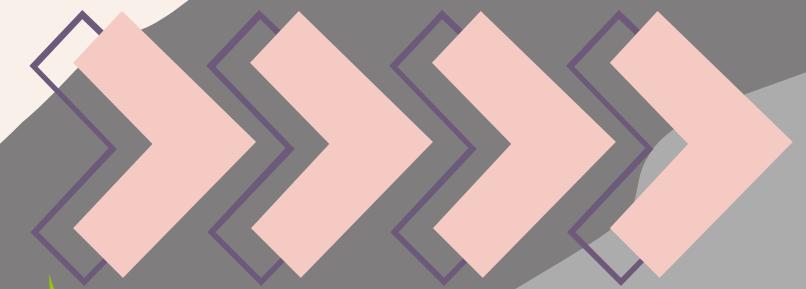
### Features



# Hasil Mobile Testing

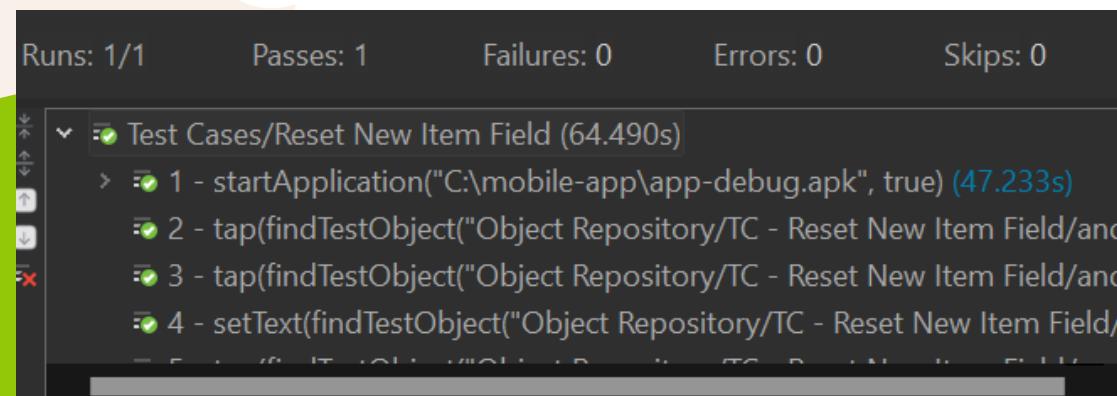
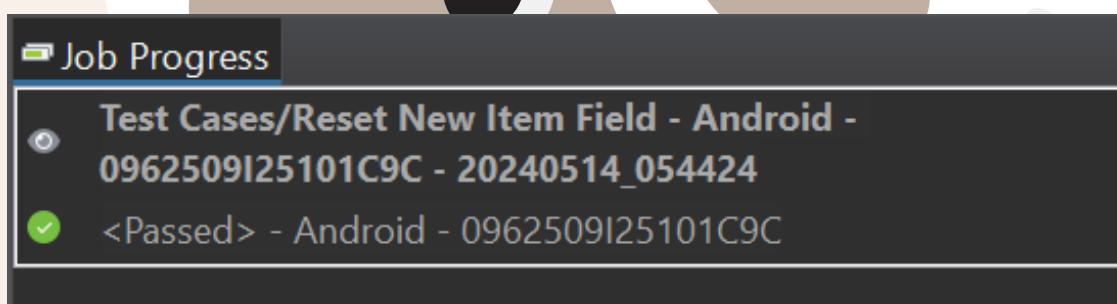
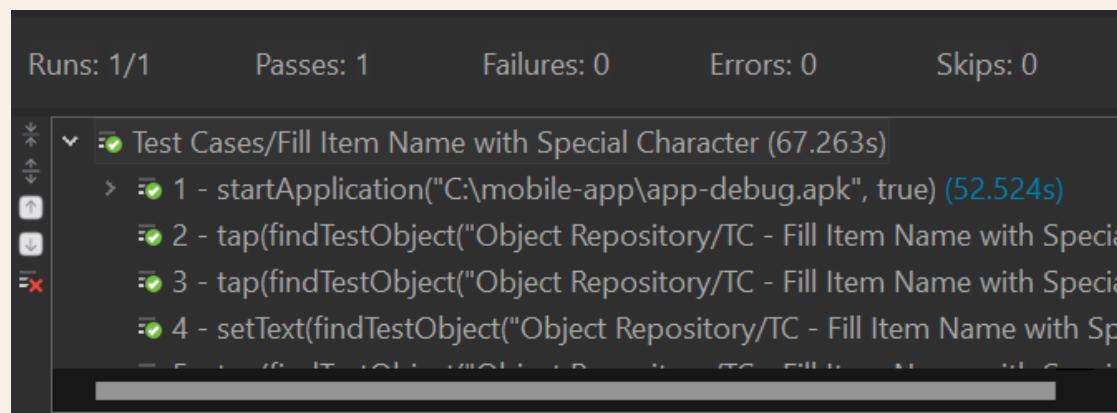
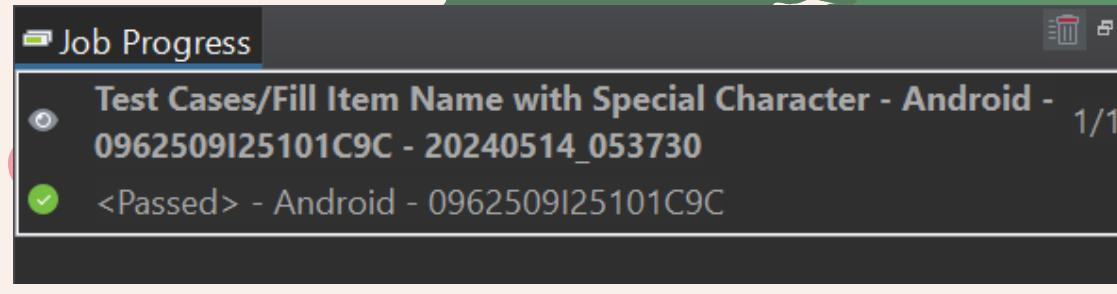


Add New Item



Fill Item Name  
with Blank Field

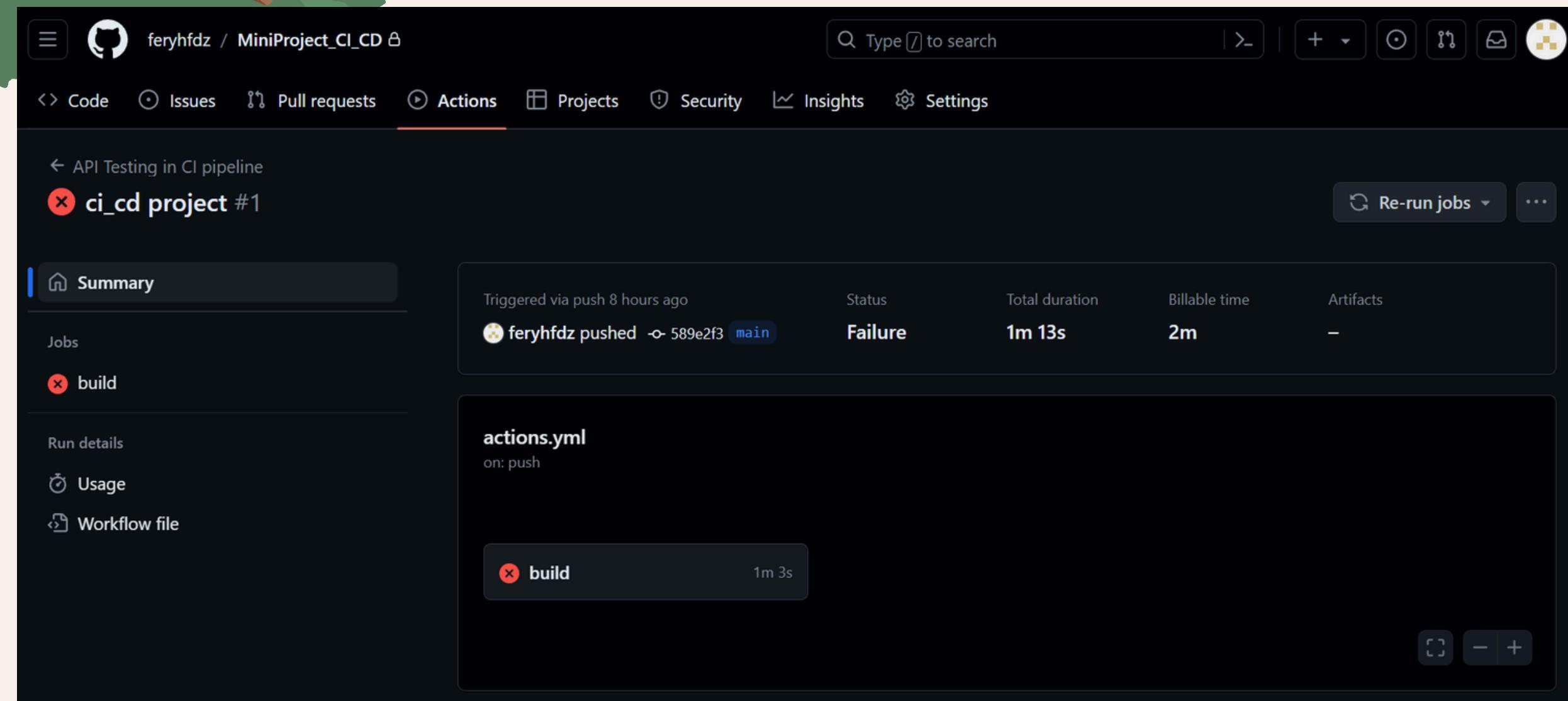
# Hasil Mobile Testing



Fill Item Name with  
Special Character

Reset New Item  
Field

# Hasil CI/CD Testing



Terdapat kesalahan  
ketika API dijalankan



# AI Implementation



Anonim

create a positive negative test case on the web <https://mini-book-store-fe.vercel.app/>

## Positive Test Case

**Title:** Successfully Adding a Book to the Cart

### Steps:

1. Navigate to the [Mini Book Store](#).
2. Browse the available books.
3. Select a book by clicking on its title or image.
4. Click the "Add to Cart" button.

### Expected Result:

- The selected book is added to the cart.
- A confirmation message or notification appears indicating the book has been added.
- The cart icon or counter updates to reflect the new item.

## Negative Test Case

**Title:** Adding a Book to the Cart Without Selection

### Steps:

1. Navigate to the [Mini Book Store](#).
2. Attempt to add a book to the cart without selecting any book (if possible via URL manipulation or by trying to trigger the action through other means).

### Expected Result:

- An error message is displayed, indicating that no book was selected.
- No book is added to the cart.
- The cart icon or counter does not update.

By following these test cases, you can ensure that the critical functionalities of the Mini Book Store are working correctly and handle error scenarios gracefully.

# Terima Kasih

