

DUPL-01

DOKUMEN UJI PERANGKAT LUNAK

MyDoc

untuk:

Instansi Kesehatan

Dipersiapkan oleh:

Akmal Muhamad Firdaus - 1301204188

Muhammad Naufal Zaidan - 1301204338

Adinda Arwa Salsabil - 1301204011

Dilia Fadilah Mutmainah - 1301204383

Prodi Teknik Informatika - Universitas Telkom

	Prodi Teknik Informatika Universitas Telkom	Nomor Dokumen		Halaman
		<i>DUPL-MyDoc <001></i>		<#>/<jml #
		Revisi	1	Tgl: 2 Januari 2023

Daftar Isi

1 PERENCANAAN Pengujian	3
1.1 Tujuan Pengujian	3
1.2 Overview Aplikasi	3
1.3 Fungsionalitas yang akan diuji	3
1.4 Jadwal Pengujian dan PIC	3
1.5 Metode Pengujian	4
2 PELAKSANAAN Pengujian	4
2.1 Tahapan Pengujian	4
2.1.1 Metode Pengujian : White Box Method	5
2.1.1.1 Pengujian DUPL-01 Menambahkan instansi kesehatan (Role Admin)	5
2.1.1.2 Pengujian DUPL-02 Menambahkan dokter (Role Instansi Kesehatan)	6
2.1.1.3 Pengujian DUPL-03 Get Dashboard (Role Pasien)	7
2.1.1.4 Pengujian DUPL-04 Menampilkan list dokter (Role Pasien)	8
2.1.1.5 Pengujian DUPL-05 Mendaftar Berobat (Role Pasien)	8
2.1.1.6 Pengujian DUPL-06 Invoice (Role Pasien)	9
2.1.1.7 Pengujian DUPL-07 Memberikan Ulasan (Role Pasien)	10
3 PELAPORAN Pengujian	12
3.1 Hasil pengujian Fitur	12
3.2 Summary Pengujian / Bug Reporting	13
Lampiran	14

1 PERENCANAAN Pengujian

1.1 Tujuan Pengujian

- Memastikan aplikasi MyDoc dapat digunakan dengan baik dan sesuai dengan spesifikasi yang diinginkan.
- Memastikan bahwa fungsi-fungsi yang ditentukan dapat digunakan dengan baik dan benar.
- Memastikan bahwa aplikasi tidak mengalami kegagalan atau kesalahan dalam operasi normal.

1.2 Overview Aplikasi

MyDoc adalah aplikasi kesehatan yang bertujuan untuk mempermudah proses mengakses layanan kesehatan, khususnya melalui pendaftaran janji online. Pasien dapat mendaftar janji tanpa perlu pergi ke fasilitas dan mengantri secara fisik. Aplikasi ini juga mencatat informasi pasien dan pendaftarannya dalam sebuah database, yang dapat dilacak dan dikelola dengan mudah dengan bantuan institusi kesehatan. Secara keseluruhan, aplikasi ini bertujuan untuk membuat proses mengakses kesehatan lebih efisien bagi pasien dan penyedia layanan kesehatan.

1.3 Fungsionalitas yang akan diuji

- DUPL-01 Menambahkan instansi kesehatan (Role Admin)
Fungsionalitas ini adalah untuk Role admin dapat menambahkan akun instansi kesehatan yang berafiliasi dengan MyDoc.
- DUPL-02 Menambahkan dokter (Role Instansi Kesehatan)
Setelah pembuatan Role Instansi Kesehatan, akun tersebut dapat langsung menambah dokter baru di instansi tersebut.
- DUPL-03 Get Dashboard
Digunakan untuk pasien sebagai overview data data pasien/akun tersebut.
- DUPL-04 Menampilkan list dokter (Role Pasien)
Sebelum melakukan checkout, pasien dapat melihat list dokter yang ada di aplikasi MyDoc.
- DUPL-05 Mendaftar Berobat (Role Pasien)
Setelah menemukan dokter yang cocok, pasien dapat menghubungi dokter tersebut dan melakukan pendaftaran berobat.
- DUPL-06 Invoice (Role Pasien)
Detail invoice jika pasien sudah melakukan mendaftar berobat.
- DUPL-07 Memberikan Ulasan (Role Pasien)
Pasien dapat memberikan ulasan setelah melakukan berobat di dokter tersebut.

1.4 Jadwal Pengujian dan PIC

Fungsionalitas	PIC	Jadwal Pengujian
DUPL-01 Menambahkan instansi kesehatan (Role Admin)	Akmal Muhamad Firdaus	20 Desember 2022

DUPL-02 Menambahkan dokter (Role Instansi Kesehatan)	Muhammad Naufal Zaidan	20 Desember 2022
DUPL-03 Get Dashboard	Muhammad Naufal Zaidan	10 Januari 2023
DUPL-04 Menampilkan list dokter (Role Pasien)	Dilia Fadilah Mutmainah	10 Januari 2023
DUPL-05 Mendaftar Berobat (Role Pasien)	Adinda Arwa Salsabil	10 Januari 2023
DUPL-06 Invoice (Role Pasien)	Adinda Arwa Salsabil	10 Januari 2023
DUPL-07 Memberikan Ulasan (Role Pasien)	Akmal Muhamad Firdaus	20 Desember 2022

1.5 Metode Pengujian

- Jenis Testing: **UNIT**
- Alat yang digunakan: **PHPUnit Testing**

2 PELAKSANAAN Pengujian

2.1 Tahapan Pengujian

- Review kode sumber aplikasi
- Pembuatan skenario pengujian
- Pembuatan Codingan untuk pengujian sesuai skenario
- Pelaksanaan pengujian berdasarkan skenario yang telah ditentukan
- Analisis hasil pengujian

Fungsionalitas	Tahapan Pengujian
DUPL-01 Menambahkan instansi kesehatan (Role Admin)	1. Login 2. Mengisi data-data instansi kesehatan 3. Tambah instansi kesehatan
DUPL-02 Menambahkan dokter (Role Instansi Kesehatan)	1. Login 2. Mengisi data-data dokter 3. Tambah dokter
DUPL-03 Get Dashboard	1. Register 2. Get Dashboard
DUPL-04 Menampilkan list dokter (Role Pasien)	1. Login 2. Get dokter
DUPL-05 Mendaftar Berobat (Role Pasien)	1. Login 2. Memilih dokter 3. Checkout
DUPL-06 Invoice (Role Pasien)	1. Login 2. Detail invoice
DUPL-07 Memberikan Ulasan (Role Pasien)	1. Login 2. Mendaftar berobat 3. Melakukan pembayaran 4. Konfirmasi antrian by Instansi Kesehatan 5. Memberikan ulasan

ID TestCase	Nama TestCase	Inputan
TC001	Login	email : String password : String fcm_token : String

TC002	Register	fullname : String email : String birthday : String address: String password1 : String password2 : String
TC003	Login As Admin	login : String password : String
TC004	Login As Instansi Kesehatan	login : String password : String

2.1.1 Metode Pengujian : White Box Method

- Metode ini digunakan untuk menguji aplikasi dengan melihat kode sumbernya.
- Fokus pada struktur kode, algoritma, dan logika yang digunakan dalam aplikasi.

2.1.1.1 Pengujian DUPL-01 Menambahkan instansi kesehatan (Role Admin)

Pengujian terdiri dari login menggunakan akun admin yang telah disediakan, lalu akan menambahkan instansi kesehatan :

Data Normal

Tabel 2.1 Menambahkan instansi kesehatan (Role Admin)

Kasus dan Hasil Uji (Data normal)				
Test Case Yang Digunakan	Test Case/test data	Expected Result	Actual Result	Kesimpulan
TC003	Username: admin_123 Password: 12345678	Form menampilkan data sukses diambil dari data session "success" dan akan terdapat data pada session "jwt_mydoc"	Terdapat data session "success" dan "jwt_mydoc"	[x] Diterima [] Ditolak
DUPL-01	nama : Nama Instansi username : username_instansi(random number) email:	Sistem memberikan data session "success"	Terdapat data session "success"	[x] Diterima [] Ditolak

	testing_instansi(r andom number)@gmail. com birthday: 2002-06-24 address : Lorem ipsum password1: 12345678 password2: 12345678			
--	---	--	--	--

2.1.1.2 Pengujian DUPL-02 Menambahkan dokter (Role Instansi Kesehatan)

Pengujian terdiri dari login menggunakan akun instansi kesehatan yang telah disediakan, lalu akan menambahkan dokter dari instansi kesehatan tersebut :

Data Normal

Tabel 2.2 Menambahkan dokter (Role Instansi Kesehatan)

Kasus dan Hasil Uji (Data normal)				
Test Case Yang Digunakan	Test Case/test data	Expected Result	Actual Result	Kesimpulan
TC004	Username: instansi_kesehat an Password: 12345678	Form menampilkan data sukses diambil dari data session "success" dan akan terdapat data pada session "jwt_mydoc"	Terdapat data session "success" dan "jwt_mydoc"	[x] Diterima [] Ditolak
DUPL-02	nama: Name Testing profession: Testing Profession price: rand(100000,500 000)	Sistem memberikan data session "success"	Terdapat data session "success"	[x] Diterima [] Ditolak

2.1.1.3 Pengujian DUPL-03 Get Dashboard (Role Pasien)

Pengujian terdiri dari login pada aplikasi mobile menggunakan akun pasien yang telah disediakan, lalu akan melakukan request GET Dashboard :

Data Normal

Tabel 2.3 Get Dashboard (Role Pasien)

Kasus dan Hasil Uji (Data normal)				
TestCase Yang Digunakan	Test Case/test data	Expected Result	Actual Result	Kesimpulan
TC002	fullname: Testing name email: testing_email(rand(0,100))@gmail.com birthday: 2002-01-01 address:Lorem Ipsum password1: 12345678 password2:1234 5678	Http Code 201 dan memberikan return json pada message "Berhasil register!"	Http Code 201 dan diberikan return json "Berhasil register!"	[x] Diterima [] Ditolak
TC001	email: testing_email(rand(0,100))@gmail.com password: 12345678 fcm_token: AnyToken	Http Code 200 dan terdapat email yang diloginkan pada data->user-> email serta mengembalik an JWT	Http Code diberikan 200 dan terdapat data email tersebut dan diapatkan JWT	[x] Diterima [] Ditolak
DUPL-03	test_token: JWT	Http Code 200 dan mengembalik an json "Berhasil mendapatkan	Http Code 201 dan diberikan return json "Berhasil mendapatkan	[x] Diterima [] Ditolak

		data!"	data!"	
--	--	--------	--------	--

2.1.1.4 Pengujian DUPL-04 Menampilkan list dokter (Role Pasien)

Pengujian terdiri dari login pada aplikasi mobile menggunakan akun pasien yang telah disediakan, lalu akan memanggil request ListDokter :

Data Normal

Tabel 2.4 Menampilkan list dokter (Role Pasien)

Kasus dan Hasil Uji (Data normal)				
Test Case Yang Digunakan	Test Case/test data	Expected Result	Actual Result	Kesimpulan
TC001	email: akmalmf007@g mail.com password: 12345678 fcm_token: AnyToken	Http Code 200 dan terdapat email yang diloginakan pada data->user->email serta mengembalik an JWT	Http Code diberikan 200 dan terdapat data email tersebut dan didapatkan JWT	[x] Diterima [] Ditolak
DUPL-04	test_token: JWT	Http Code 200 dan mengembalik an json "Berhasil mendapatkan data!"	Http Code 200 dan diberikan return json "Berhasil mendapatkan data!"	[x] Diterima [] Ditolak

2.1.1.5 Pengujian DUPL-05 Mendaftar Berobat (Role Pasien)

Pengujian terdiri dari login pada aplikasi mobile menggunakan akun pasien yang telah disediakan, lalu akan mendaftar berobat sesuai dengan dokter dan instansi kesehatan tertentu :

Data Normal

Tabel 2.5 Mendaftar Berobat (Role Pasien)

Kasus dan Hasil Uji (Data normal)				
Test Case Yang Digunakan	Test Case/test data	Expected Result	Actual Result	Kesimpulan
TC001	email: akmalmf007@g	Http Code 200 dan	Http Code diberikan 200	[x] Diterima [] Ditolak

	mail.com password: 12345678 fcm_token: AnyToken	terdapat email yang diloginakan pada data->user->email serta mengembalik an JWT	dan terdapat data email tersebut dan didapatkan JWT	
DUPL-04	test_token: JWT	Http Code 200 dan mengembalik an json "Berhasil mendapatkan data!" dan diberikan list id dari dokter	Http Code 200 dan diberikan return json "Berhasil mendapatkan data!" dan mendapatkan id dokter	
DUPL-05	test_token: JWT dokter_id: 1 time:12:30 payment_metho d: VA	Http Code 201 dan mengembalik an json "Berhasil melakukan checkout!"	Http Code 201 dan diberikan return json "Berhasil melakukan checkout!"	[x] Diterima [] Ditolak

2.1.1.6 Pengujian DUPL-06 Invoice (Role Pasien)

Pengujian terdiri dari login pada aplikasi mobile menggunakan akun pasien yang telah disediakan, lalu akan melihat detail invoice berdasarkan ID invoice pada DUPL-05:

Data Normal

Tabel 2.6 Invoice (Role Pasien)

Kasus dan Hasil Uji (Data normal)				
TestCase Yang Digunakan	Test Case/test data	Expected Result	Actual Result	Kesimpulan
TC001	email: akmalmf007@g mail.com password: 12345678 fcm_token: AnyToken	Http Code 200 dan terdapat email yang diloginakan pada data->user->	Http Code 200 dan terdapat data email tersebut dan didapatkan JWT	[x] Diterima [] Ditolak

		email serta mengembalik an JWT		
DUPL-06	test_token: JWT	Http Code 200 dan mengembalik an json "Berhasil mendapatkan data!"	Http Code 200 dan diberikan return json "Berhasil mendapatkan data!"	[x] Diterima [] Ditolak

2.1.1.7 Pengujian DUPL-07 Memberikan Ulasan (Role Pasien)

Pengujian terdiri dari login pada aplikasi mobile menggunakan akun pasien yang telah disediakan, lalu akan memberikan ulasan terhadap dokter yang ada pada invoice tersebut :

Data Abnormal

Tabel 2.7 Memberikan Ulasan (Role Pasien)

Kasus dan Hasil Uji (Data normal)				
TestCase Yang Digunakan	Test Case/test data	Expected Result	Actual Result	Kesimpulan
TC001	email: akmalmf007@g mail.com password: 12345678 fcm_token: AnyToken	Http Code 200 dan terdapat email yang diloginkan pada data->user-> email serta mengembalik an JWT	Http Code diberikan 200 dan terdapat data email tersebut dan diapatkan JWT	[x] Diterima [] Ditolak
DUPL-05	test_token: JWT dokter_id: 1 time:12:30 payment_metho d: VA	Http Code 201 dan mengembalik an json "Berhasil melakukan checkout!"	Http Code 201 dan diberikan return json "Berhasil melakukan checkout!"	[x] Diterima [] Ditolak
DUPL-07	invoice_id: invoice_id description: Lorem Ipsum	Http Code 403 dan menampilkan json "Sesi	Http code 201 dan menampilkan json "Berhasil	[] Diterima [x] Ditolak

	Dolor Sit Amet star: 5 test_token: JWT	dengan dokter belum dimulai”	memberikan ulasan!”	
--	--	------------------------------------	------------------------	--

Data Normal

Tabel 2.8 Memberikan Ulasan (Role Pasien)

Kasus dan Hasil Uji (Data normal)				
TestCase Yang Digunakan	Test Case/test data	Expected Result	Actual Result	Kesimpulan
TC001	email: akmalmf007@g mail.com password: 12345678 fcm_token: AnyToken	Http Code 200 dan terdapat email yang diloginkan pada data->user-> email serta mengembalik an JWT	Http Code diberikan 200 dan terdapat data email tersebut dan diapatkan JWT	[x] Diterima [] Ditolak
DUPL-05	test_token: JWT dokter_id: 1 time:12:30 payment_metho d: VA	Http Code 201 dan mengembalik an json “Berhasil melakukan checkout!”	Http Code 201 dan diberikan return json “Berhasil melakukan checkout!”	[x] Diterima [] Ditolak
TC005	id_invocie: id_invoice	Pengubahan status invoice menjadi 2	Status invoice berubah menjadi 2	[x] Diterima [] Ditolak
DUPL-07	invoice_id: invoice_id description: Lorem Ipsum Dolor Sit Amet star: 5 test_token: JWT	Http Code 201 dan menampilkan json “Berhasil memberikan ulasan!”	Http code 201 dan menampilkan json “Berhasil memberikan ulasan!”	[x] Diterima [] Ditolak

3 PELAPORAN Pengujian

3.1 Hasil pengujian Fitur

Fungsionalitas Uji	Modul Uji	Kesimpulan Pengujian	Deskripsi
DUPL-01 Menambahkan instansi kesehatan (Role Admin)	TC003	Berhasil	
	DUPL-01	Berhasil	
DUPL-02 Menambahkan dokter (Role Instansi Kesehatan)	TC004	Berhasil	
	DUPL-02	Berhasil	
DUPL-03 Get Dashboard (Role Pasien)	TC002	Berhasil	
	TC001	Berhasil	
	DUPL-03	Berhasil	
DUPL-04 Menampilkan list dokter (Role Pasien)	TC001	Berhasil	
	DUPL-04	Berhasil	
DUPL-05 Mendaftar Berobat (Role Pasien)	TC001	Berhasil	
	DUPL-04	Berhasil	
	DUPL-05	Berhasil	
DUPL-06 Invoice (Role Pasien)	TC001	Berhasil	
	DUPL-06	Berhasil	
DUPL-07 Memberikan Ulasan (Role Pasien) Abnormal Data	TC001	Berhasil	
	DUPL-05	Berhasil	
	DUPL-07	Gagal	Ulasan tetap dapat diberikan walaupun sesi dokter belum dimulai
DUPL-07 Memberikan Ulasan (Role Pasien) Normal Data	TC001	Berhasil	
	DUPL-05	Berhasil	
	TC005	Berhasil	
	DUPL-07	Berhasil	

3.2 Summary Pengujian / Bug Reporting

Fungsionalitas Uji	Hasil Pengujian	Deskripsi	Prioritas Perbaikan
DUPL-07 Memberikan Ulasan (Role Pasien) Abnormal Data	Terdapat bug pada saat memberikan ulasan	Ulasan tetap dapat diberikan walaupun sesi dokter belum dimulai	Tidak mendesak dan bukan hal yang krusial

Kesimpulan dari pengujian ini adalah bahwa aplikasi MyDoc dapat digunakan dengan baik dan sesuai dengan spesifikasi yang diinginkan. Source Code yang digunakan dalam aplikasi sesuai dengan standar yang ditentukan. Terdapat satu minor bug pada spesifikasi DUPL-07, tetapi hal tersebut bukan merupakan hal yang krusial dan tidak mendesak untuk dibenarkan. Pengujian ini menunjukkan bahwa aplikasi MyDoc siap untuk digunakan oleh pengguna.

Lampiran

A. Code

```
private function testCase001($email_testing, $password_testing)
{
    if(is_null($email_testing) || is_null($password_testing)){
        $email_testing = "akmalmf007@gmail.com";
        $password_testing = "12345678";
    }

    // Validate the response if login with correct email and
    password
    $response = $this->call('post', '/api/v1/auth/login', [
        'email' => $email_testing,
        'password' => $password_testing,
        'fcm_token' => "AnyToken"
    ]);
    //expecting data is not null and status code to be 200 /
    Ok
    $response->assertJSONFragment(['data' => ['user' =>
    ['email' => $email_testing]]]);
    $response->assertStatus(200);
    $json = json_decode($response->getJSON(), true);

    return $json['data']['token'];
}

private function testCase002()
{
    $fullname_testing = "Testing Name";
    $email_testing = "testing_email".rand(0,100)."@gmail.com";
    $birthday_testing = "2002-01-01";
    $addres_testing = "Lorem Ipsum";
    $password_testing = "12345678";
    // Validate the response if login with correct email and
    password
    $response = $this->call('post', '/api/v1/auth/register', [
        'fullname' => $fullname_testing,
        'email' => $email_testing,
        'birthday' => $birthday_testing,
        'address' => $addres_testing,
        'password1' => $password_testing,
```

```

        'password2' => $password_testing
    });
    //expecting data is not null and status code to be 201 /
Created
    $response->assertJSONFragment(['message' => "Berhasil
register!"]);
    $response->assertStatus(201);
    return [$email_testing, $password_testing];
}

// $this->expectOutputString('foo');

private function testCase003() {
    $username_testing = "admin_123";
    $password_testing = "12345678";

    $response = $this->call('post', '/auth/login', [
        'login' => $username_testing,
        'password' => $password_testing
    ]);
    //getenv("JWT_WEB_NAME")
    $jwt = session(getenv("JWT_WEB_NAME"));
    $this->assertTrue($jwt != null);
    return $jwt;
}

private function testCase004() {
    $username_testing = "instansi_kesehatan";
    $password_testing = "12345678";

    $response = $this->call('post', '/auth/login', [
        'login' => $username_testing,
        'password' => $password_testing
    ]);
    //getenv("JWT_WEB_NAME")
    $jwt = session(getenv("JWT_WEB_NAME"));
    $this->assertTrue($jwt != null);
    return $jwt;
}

// Route Add Instansi Kesehatan: /admin/instansi/add
public function testDUPL_01(){
    $jwt = $this->testCase003();
    if($jwt != null){

```

```

        $response = $this->withSession([
            getenv("JWT_WEB_NAME") => $jwt
        ])->call('post', '/admin/instansi/add', [
            'nama' => 'Nama Instansi',
            'username' => 'username_instansi'.rand(0,100),
            'email' =>
'testing_instansi'.rand(0,100).'@gmail.com',
            'birthday' => '2002-01-01',
            'address' => 'Jl Instansi',
            'password' => '12345678',
            'password2' => '12345678'
        ]);
        $isSuccess = session("success");
        $this->assertTrue($isSuccess != null);
    }
}

// Route Add Dokter: /admin/dokter/add
public function testDUPL_02(){
    $jwt = $this->testCase004();
    if($jwt != null){
        $response = $this->withSession([
            getenv("JWT_WEB_NAME") => $jwt
        ])->call('post', '/admin/dokter/add', [
            'nama' => 'Name Testing',
            'profession' => 'Testing Profession',
            'price' => rand(100000,500000)
        ]);
        $isSuccess = session("success");
        $this->assertTrue($isSuccess != null);
        return json_decode($response->getJSON(), true);
    }
}

public function testDUPL_03(){
    $u = $this->testCase002();
    if($u != null){
        $jwt = $this->testCase001($u[0], $u[1]);
        if($jwt != null){
            $response = $this->call('get',
'/api/v1/master/dashboard',[
                'test_token' => 'Bearer '.$jwt
            ]);

```



```

        $response->assertJSONFragment(['message' =>
"Berhasil mendapatkan data!"]);
        $response->assertStatus(200);
    }
}

public function testDUPL_04(){
    $jwt = $this->testCase001(null, null);
    if($jwt != null){
        $response = $this->call('get',
'/api/v1/master/dokter',[
        'test_token' => 'Bearer '.$jwt
    ]);

        $response->assertJSONFragment(['message' => "Berhasil
mendapatkan data!"]);
        $response->assertStatus(200);
    }
}

public function testDUPL_05(){
    $jwt = $this->testCase001(null, null);
    if(!is_null($jwt)){
        $dokter = 1;
        $response = $this->call('post',
'/api/v1/transaction/checkout', [
        'dokter_id' => $dokter,
        'time' => "12:30",
        'payment_method' => "VA",
        'test_token' => 'Bearer '.$jwt
    ]);
        //expecting data is not null and status code to be 201
/ Created
        $response->assertJSONFragment(['message' => "Berhasil
melakukan checkout"]);
        $response->assertStatus(201);
        return json_decode($response->getJSON(),
true)['data'];
    }
}

public function testDUPL_06(){
    $jwt = $this->testCase001(null, null);

```

```

        if($jwt != null){
            $invoice_id = 31;
            $response = $this->call('get',
'/api/v1/transaction/invoice/' . $invoice_id, [
                'test_token' => 'Bearer ' . $jwt
            ]);

            $response->assertJSONFragment(['message' => "Berhasil
Mendapatkan data!"]);
            $response->assertJSONFragment(['data' => ['id' =>
$invoice_id]]);
            $response->assertStatus(200);
        }
    }

    public function testDUPL_07_negative(){
        $jwt = $this->testCase001(null, null);
        if(!is_null($jwt)){
            $invoice_id = $this->testDUPL_05();
            $response = $this->call('post',
'/api/v1/transaction/give_rating', [
                'invoice_id' => $invoice_id,
                'description' => "Lorem Ipsum Dolor Sit Amet",
                'star' => 5,
                'test_token' => 'Bearer ' . $jwt
            ]);
            //expecting data is not null and status code to be 403
/ Forbidden
            $response->assertStatus(403);
        }
    }

    public function testDUPL_07_positive(){
        $jwt = $this->testCase001(null, null);
        if(!is_null($jwt)){
            $invoice_id = $this->testDUPL_05();
            $data = array(
                'status' => 2
            );
            $this->db->table('invoice')->where('id',
$invoice_id)->update($data);

            $response = $this->call('post',
'/api/v1/transaction/give_rating', [

```

```

        'invoice_id' => $invoice_id,
        'description' => "Lorem Ipsum Dolor Sit Amet",
        'star' => 5,
        'test_token' => 'Bearer '.$jwt
    ]);
    //expecting data is not null and status code to be 201
/ Created
    $response->assertStatus(201);
}
}

```

B. Hasil Testing

```

E:\Kuliah\Semester 5\RPL_IMPAL\MyDoc\website\vendor\codeigniter4\framework\system\Test\phpunit
PHPUnit 9.5.25 #StandWithUkraine

Warning:      No code coverage driver available

.....F.                                     8 / 8 (100%)

Time: 00:07.458, Memory: 14.00 MB

There was 1 failure:

1) MyDocTest::testDUPL_07_negative
Failed asserting that 201 is identical to 403.

E:\Kuliah\Semester 5\RPL_IMPAL\MyDoc\website\vendor\codeigniter4\framework\system\Test\TestResponse.php:145
E:\Kuliah\Semester 5\RPL_IMPAL\MyDoc\website\tests\unit\MyDocTest.php:210
phpvfscomposer://E:\Kuliah\Semester 5\RPL_IMPAL\MyDoc\website\vendor\phpunit\phpunit\phpunit:97

FAILURES!
Tests: 8, Assertions: 53, Failures: 1.

E:\Kuliah\Semester 5\RPL_IMPAL\MyDoc\website>

```

1 Dari 8 Testing DUPL terdapat 1 yang failure, yaitu DUPL-07 dengan Abnormal Data