

## **BAB III**

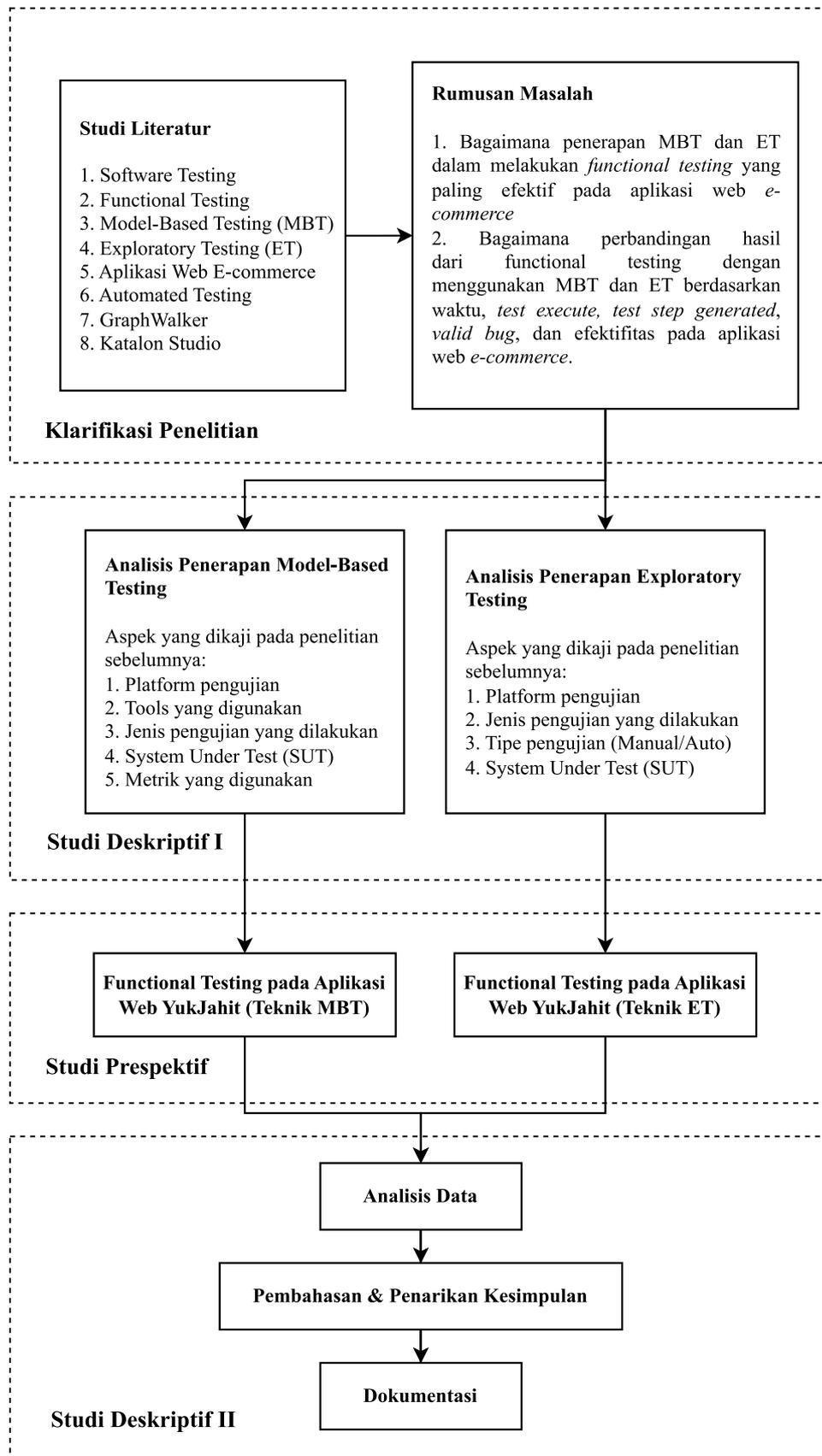
### **METODE PENELITIAN**

#### **1.1 Desain Penelitian**

Penelitian dilakukan dengan menggunakan metode *Design Research Methodology* (DRM). DRM merupakan sebuah metode atau petunjuk untuk melakukan dan merancang sebuah penelitian (Blessing dan Chakrabarti, 2009). DRM ditujukan untuk membuat perancangan penelitian lebih efektif dan efisien. Penerapan DRM memiliki beberapa objektif, yaitu:

1. Membuat kerangka untuk rancangan penelitian,
2. Mengidentifikasi cakupan penelitian, proyek, dan program yang akan berkaitan baik secara akademik maupun praktik,
3. Membuka kemungkinan untuk menggunakan pendekatan dan metode pada penelitian yang bervariasi,
4. Memberikan pedoman untuk perencanaan sistematis penelitian,
5. Untuk menghasilkan pedoman bagi penelitian yang lebih ketat,
6. Untuk mendapatkan sebuah argumen yang kuat,
7. Menerapkan metode baru dan menunjukkan metode yang sudah ada untuk dibawa ke tahapan penelitian,
8. Membantu untuk memilih metode yang tepat atau kombinasi metode-metode,
9. Menjelaskan posisi dari penelitian jika dibandingkan dengan penelitian yang lain,
10. Mendorong refleksi terhadap pendekatan yang digunakan.

Desain penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 1.1 Desain Penelitian

### 1.1.1 Klarifikasi Penelitian

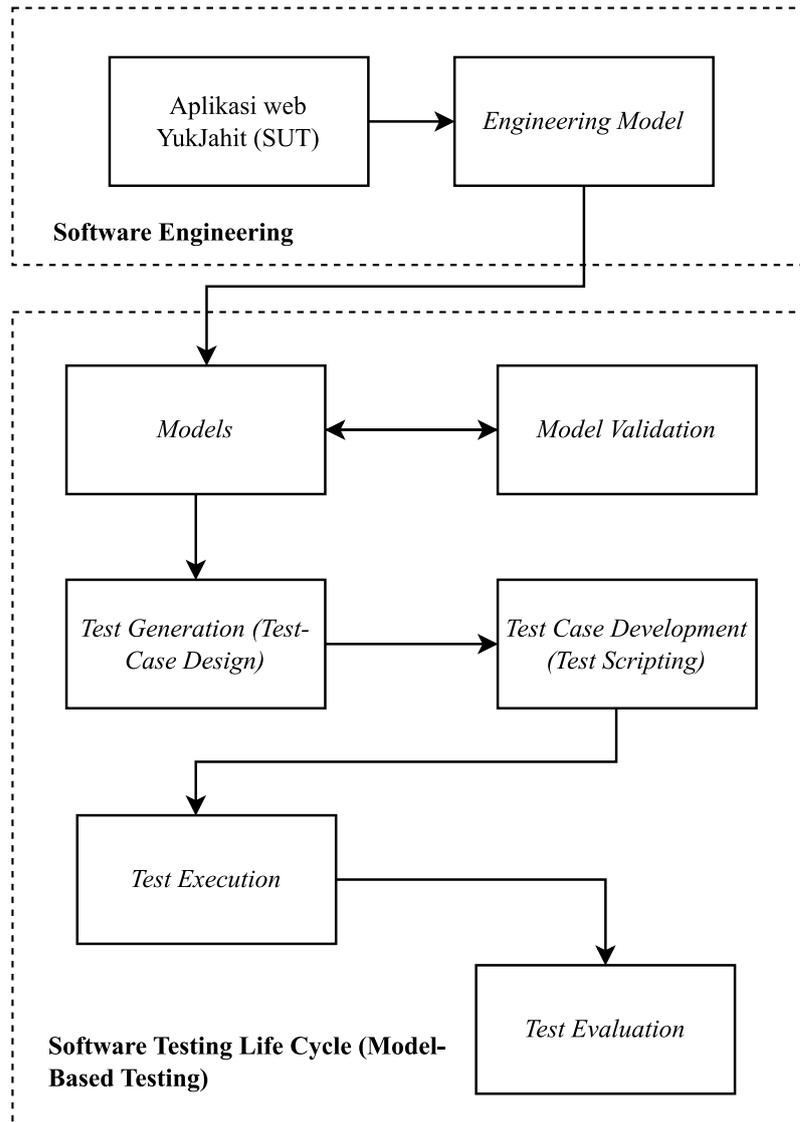
Klarifikasi penelitian merupakan tahapan pertama pada DRM yang bertujuan untuk mengumpulkan dan menganalisa literatur terkait penelitian guna memberikan landasan penelitian. Beberapa kajian literatur tersebut diantaranya adalah *Software Testing*, *Functional Testing*, *Model-Based Testing*, *Exploratory Testing*, Aplikasi Web *E-commerce*, *Automated Testing*, GraphWalker, Selenium IDE dan Katalon Studio. Kajian tersebut dibahas pada BAB II yang didapatkan dari sumber seperti jurnal, artikel, buku, skripsi, tesis, dsb. Dari kajian literatur tersebut, peneliti dapat merumuskan permasalahan dan tujuan yang akan dibahas pada penelitian ini pada BAB I.

### 1.1.2 Studi Deskriptif I

Pada tahapan ini, peneliti mendalami pemahaman terkait dengan topik penelitian untuk mencari tahu dan menjelaskan perbedaan penelitian ini dengan penelitian-penelitian sebelumnya. Topik penelitian yang dimaksud adalah penerapan MBT untuk pengujian perangkat lunak. Dari hal tersebut, terdapat beberapa aspek dari penelitian ini yang akan memiliki perbedaan dengan penelitian-penelitian sebelumnya, yaitu platform pengujian, *tools* yang digunakan, jenis pengujian yang dilakukan, *system under test* (SUT), dan metrik yang digunakan. Terdapat beberapa kajian terkait yang dijadikan sebagai referensi tambahan juga seperti, pengujian pada aplikasi web dan penerapan ET untuk pengujian perangkat lunak. Dari kajian tersebut, peneliti dapat memaparkan *state-of-the-art* dari topik yang diangkat.

### 1.1.3 Studi Prespektif

Pada tahapan *functional testing* dengan menggunakan teknik MBT, *tools* MBT yang akan digunakan pada penelitian ini adalah GraphWalker, Selenium WebDriver, dan JUnit. *Tools* tersebut nantinya akan digabungkan agar terjadinya *automation testing* dengan script yang dibuat pada Selenium WebDriver dan penggunaan UML Finite State Machine pada GraphWalker untuk pengujian aplikasi web *E-commerce* YukJahit. Tahapan pada MBT dapat digambarkan pada Gambar 3.2 sebagai berikut (Garousi dkk., 2018).



Gambar 1.2 Skenario pengujian MBT

Penjelasan dari masing-masing tahapan dapat dijelaskan secara rinci, sebagai berikut:

a. *Engineering Model*

Mencari atau membuat sebuah model dari aplikasi web YukJahit yang nantinya dapat digunakan sebagai referensi yang akan digunakan pada pembuatan model di GraphWalker. Pada penelitian ini, peneliti menggunakan *state machine diagram* yang telah dibuat pada tahapan desain di tahapan SDLC sebelumnya. *State machine diagram* dapat dilihat pada Gambar 3.3.



c. *Model Validation*

Memvalidasi apakah model-model yang telah dibuat bersih dari kesalahan desain dan dapat dijalankan pada GraphWalker. Validasi ini dapat dilakukan secara otomatis dengan bantuan GraphWalker.

d. *Test Generation (Test Case Design)*

Tahapan ini sepenuhnya dijalankan oleh GraphWalker. *Test case* akan otomatis dibuat langsung dari GraphWalker berdasarkan model yang telah kita buat.

e. *Test Case Development*

Mengembangkan *test case* dengan mengintegrasikannya langsung dengan bantuan *test scripting* pada *automated testing tools*. Dalam penelitian ini, *tools* yang dipakai adalah Selenium WebDriver.

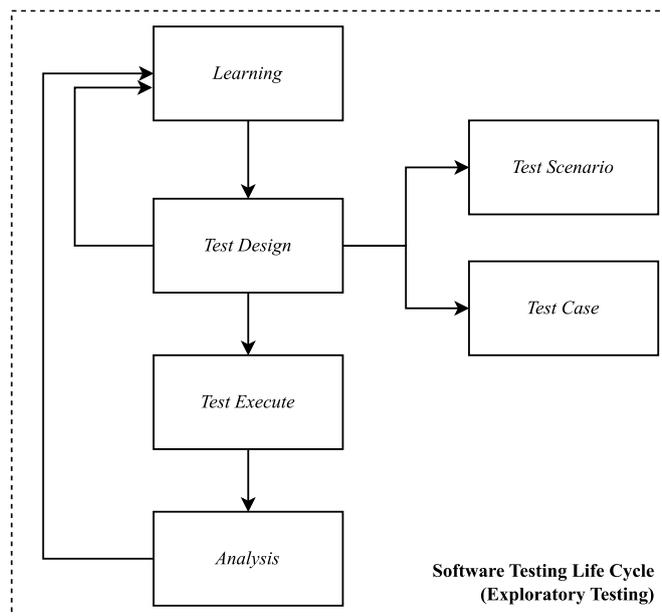
f. *Test Execution*

Menjalankan *test case* yang telah dibuat. GraphWalker akan diintegrasikan dengan Selenium WebDriver sebagai *automated testing tools* dan JUnit library sebagai *assertion tools*.

g. *Test Evaluation*

Melakukan evaluasi terhadap seluruh kegiatan pengujian yang telah dilakukan. Data yang akan dievaluasi adalah waktu, *test execute*, *test steps generated*, dan jumlah *bug* yang ditemukan.

Pada tahapan *functional testing* dengan menggunakan ET terdapat 3 tahapan utama yaitu, *learning*, *test design*, *test execute*. Ketiga tahapan ini dapat berjalan paralel dan dinamis yang akan selalu berlanjut hingga SUT sudah tidak ditemukan adanya *error* ataupun *bug* (Afzal dkk., 2015). Tahapan pada ET dapat digambarkan pada Gambar 3.4 sebagai berikut.



Gambar 1.4 Skenario pengujian ET

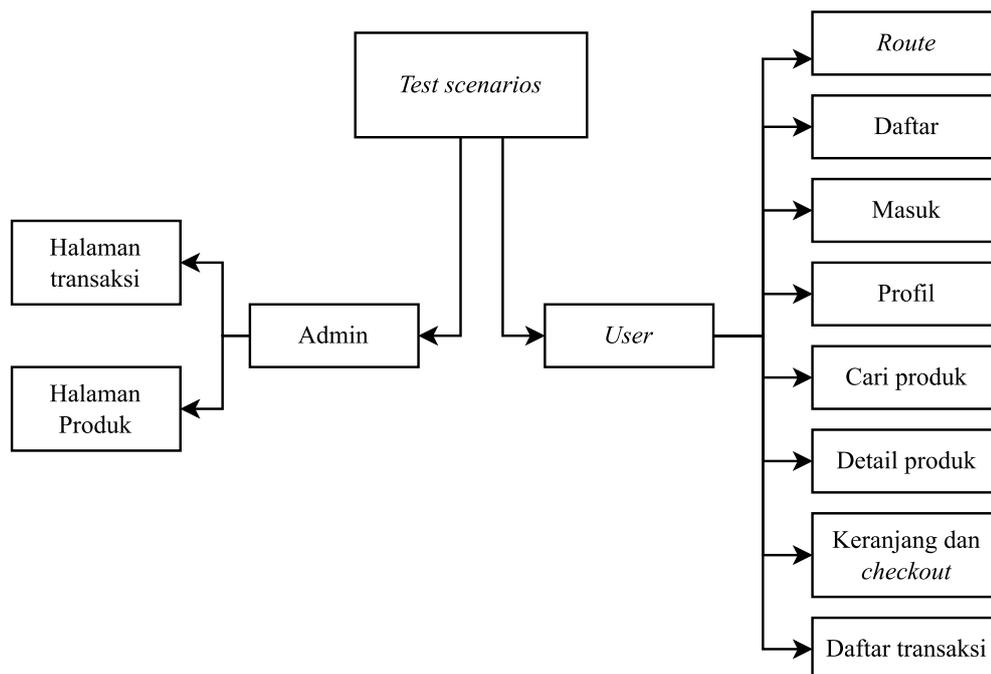
Penjelasan dari masing-masing tahapan dapat dijelaskan secara rinci, sebagai berikut:

a. *Learning*

Mempelajari secara teliti hal-hal yang terkait dengan SUT seperti, spesifikasi, fitur-fitur, alur kerja, dan sebagainya untuk mencari kemungkinan terjadinya *error* atau *bug* yang terdapat pada SUT. Dapat dilakukan secara paralel dengan tahapan lainnya.

b. *Test Design*

Membuat *test scenario* dan *test case* berdasarkan pengetahuan dari peneliti. *Test case* dapat berubah sewaktu-waktu, sebagai contoh misalkan peneliti secara tidak sengaja menemukan sebuah celah yang dapat menggagalkan sistem pada SUT. Maka ini akan dimasukkan kedalam bagian *test case* yang sebelumnya belum pernah dicantumkan. *Test scenario* dibuat menjadi 2 bagian besar, yaitu *user* atau pengguna dan administrator (admin). *User* kemudian dibagi menjadi 8 bagian, yaitu *route*, daftar, masuk, profil, cari produk, detail produk, keranjang dan *checkout*, daftar transaksi. Sedangkan admin dibagi menjadi 2 bagian, yaitu halaman produk dan halaman transaksi. Modul-modul pada *test scenario* digambarkan pada Gambar 3.5.



Gambar 1.5 Modul *test scenario*

Pada masing-masing bagan, maka akan dibuat sebuah *test data* dan *test case* yang akan digunakan untuk mempermudah pengujian pada Katalon Studio. Berikut merupakan hasil *test data* dan *test case* yang akan digunakan pada penelitian ini:

- 1) Prekondisi dari skenario *user*:
  - a. Pengguna terkoneksi ke internet
  - b. Pengguna sudah berada pada halaman utama YukJahit pada <https://yukjahit.vercel.app/>
- 2) Skenario *protected route* dan 404
  - a. Prekondisi : Pengguna belum terautentikasi pada aplikasi.
  - b. *Test data* : Tabel 3.1.

Tabel 1.1  
*Test data protected route dan 404*

<b><i>Route</i></b>	<b><i>expected result</i></b>
/	<i>success</i>
/search, /products; /login; /register; /products:id	<i>success</i>
/cart; /profile; /order; /checkout;	<i>failed</i>
/ruteselainrutediatas	404

Dimana kolom *route* menunjukkan rute dari aplikasi yang akan menghasilkan 3 jenis keluaran (kolom *expected result*), yaitu *success* berarti pengguna dapat mengakses rute tersebut, *failed* berarti pengguna tidak dapat mengakses rute tersebut dan 404 berarti pengguna akan dialihkan ke halaman tidak ditemukan.

c. *Test case* : Tabel 3.2.

Tabel 1.2  
TC00 - *Test case protected route* dan 404

<i>Test Case ID</i>	<i>Feature</i>	<i>Detail Test Case</i>	<i>Test Step</i>	<i>Expected Result</i>
TC00001	<i>Route</i>	<b>Tidak Berhasil.</b> Pengguna mencoba masuk ke rute halaman yang terkunci.	1. Pengguna masuk ke rute yang terkunci.	Pengguna mendapatkan notifikasi <i>toast</i> “Silahkan masuk terlebih dahulu” dan dialihkan ke halaman utama.
TC00002	<i>Route</i>	<b>Berhasil.</b> Pengguna mencoba masuk ke rute halaman yang terbuka.	1. Pengguna masuk ke rute yang terbuka.	Pengguna berhasil masuk ke halaman tersebut.
TC00003	<i>Route</i>	<b>Tidak Berhasil.</b> Pengguna mencoba masuk ke rute yang tidak terdaftar.	1. Pengguna masuk ke rute yang tidak terdaftar.	Pengguna akan dialihkan ke halaman 404.

### 3) Skenario Masuk

- Prekondisi : Pengguna belum terautentikasi pada sistem.
- Test data* : Merupakan data yang dimasukkan pengguna ke *form* masuk, dapat dilihat pada Tabel 3.3.

Tabel 1.3  
*Test data* masuk

<i>Email</i>	<i>password</i>	<i>expected result</i>
Null	Null	<i>fail_null</i>

<a href="mailto:yukjahit@email.com">yukjahit@email.com</a>	Null	<i>fail_null</i>
Null	yukjahit123	<i>fail_null</i>
<a href="mailto:yukjahit@email.com">yukjahit@email.com</a>	wrongpassword	<i>fail_toast</i>
<a href="mailto:yukjahit@wrongemail.com">yukjahit@wrongemail.com</a>	yukjahit123	<i>fail_toast</i>
<a href="mailto:yukjahit@email.com">yukjahit@email.com</a>	yukjahit123	<i>success</i>

c. *Test case* : Tabel 3.4.

Tabel 1.4  
TC01 - *Test case* masuk

<i>Test Case ID</i>	<i>Feature</i>	<i>Detail Test Case</i>	<i>Test Step</i>	<i>Expected Result</i>
<b>TC01001</b>	Masuk	<b>Berhasil.</b> Pegguna masuk dengan <i>email</i> dan <i>password</i> yang sesuai.	<ol style="list-style-type: none"> <li>Pegguna membuka halaman masuk.</li> <li>Pegguna mengisi <i>email</i> yang sudah terdaftar.</li> <li>Pegguna mengisi <i>password</i> yang sesuai dengan <i>email</i>.</li> <li>Klik tombol 'Masuk'.</li> </ol>	Pegguna berhasil masuk ke Aplikasi dan mendapatkan akses token dan akan dialihkan ke halaman utama.
<b>TC01002</b>	Masuk	<b>Tidak Berhasil.</b> Pegguna masuk dengan <i>email</i> atau <i>password</i> yang salah.	<ol style="list-style-type: none"> <li>Pegguna membuka halaman masuk.</li> <li>Pegguna mengisi <i>email</i> dan/atau <i>password</i> yang salah.</li> <li>Klik tombol 'Masuk'.</li> </ol>	Pegguna tidak berhasil masuk dan tidak akan mendapatkan akses token.
<b>TC01003</b>	Masuk	<b>Tidak Berhasil.</b> Pegguna masuk tanpa mengisi field <i>email</i> dan <i>password</i> .	<ol style="list-style-type: none"> <li>Pegguna membuka halaman masuk.</li> <li>Pegguna tidak mengisi <i>email</i> dan <i>password</i>.</li> <li>Klik tombol 'Masuk'.</li> </ol>	Pegguna dapat notifikasi untuk mengisi field yang dikosongkan.
<b>TC01004</b>	Masuk	<b>Berhasil</b> Pegguna mencoba untuk mendaftar akun baru.	<ol style="list-style-type: none"> <li>Pegguna membuka halaman masuk.</li> <li>Pegguna memilih untuk daftar akun baru.</li> </ol>	Pegguna akan dialihkan ke halaman daftar.

## 4) Skenario Daftar

- a. Prekondisi : Pengguna belum terautentikasi pada sistem.
- b. *Test data* : Merupakan data yang dimasukkan pengguna ke *form* daftar, dapat dilihat pada Tabel 3.5.

Tabel 1.5  
*Test data* daftar

<i>nama</i>	<i>email</i>	<i>password</i>	<i>konfirmasi password</i>	<i>no hp</i>	<i>alamat lengkap</i>	<i>expect ed result</i>
yukjahit daftar	yukjahitdaftar@email.com	yukjahit123	yukjahit123	08123456789	Alamat lengkap yukjahit	<i>success</i>
email terdaftar	<u>emailterdaftar@email.com</u>	emailterdaftar	emailterdaftar	08123456789	Alamat email terdaftar	<i>fail</i>
password tidak sama	<u>passwordtidaksama@email.com</u>	Tidak sama	tidak sama 2	08123456789	alamat password tidak sama	<i>fail</i>
password salah format	<u>passwordsalahformat@email.com</u>	password 1234	password1234	08123456789	alamat password salah format	<i>fail</i>
null	null	null	null	Null	null	<i>fail_null</i>

- c. *Test case* : Tabel 3.6

Tabel 1.6  
TC02 - *Test case* daftar

<i>Test Case ID</i>	<i>Feature</i>	<i>Detail Test Case</i>	<i>Test Step</i>	<i>Expected Result</i>
TC02001	Daftar	<b>Berhasil.</b> Pengguna daftar dengan mengisi semua field pada	1. Pengguna membuka halaman daftar. 2. Pengguna mengisi nama lengkap.	Pengguna berhasil daftar dan akan dialihkan ke halaman masuk

		<i>form</i> registrasi dengan benar.	<ol style="list-style-type: none"> <li>3. Pengguna mengisi <i>email</i> yang belum terdaftar.</li> <li>4. Pengguna mengisi <i>password</i> dan konfirmasi <i>password</i> dengan data yang sama.</li> <li>5. Pengguna mengisi nomor HP.</li> <li>6. Pengguna mengisi alamat lengkap.</li> <li>7. Klik tombol ‘Daftar’.</li> </ol>	
<b>TC02002</b>	Daftar	<b>Tidak Berhasil.</b> Pengguna daftar dengan mengisi semua field pada <i>form</i> registrasi dengan benar, tetapi dengan <i>email</i> yang sudah terdaftar.	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman daftar.</li> <li>2. Pengguna mengisi nama lengkap.</li> <li>3. Pengguna mengisi <i>email</i> yang sudah terdaftar.</li> <li>4. Pengguna mengisi <i>password</i> dan konfirmasi <i>password</i> dengan data yang sama.</li> <li>5. Pengguna mengisi nomor HP.</li> <li>6. Pengguna mengisi alamat lengkap.</li> <li>7. Klik tombol ‘Daftar’.</li> </ol>	Pengguna tidak berhasil daftar dan akan muncul sebuah notifikasi ( <i>toast</i> ) ‘Registrasi Gagal’.
<b>TC02003</b>	Daftar	<b>Tidak Berhasil.</b> Pengguna daftar tanpa mengisi field tertentu pada <i>form</i> registrasi.	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman daftar</li> <li>2. Pengguna tidak mengisi nama lengkap.</li> <li>3. Pengguna tidak mengisi <i>email</i>.</li> <li>4. Pengguna mengisi <i>password</i> dan konfirmasi <i>password</i>.</li> <li>5. Pengguna mengisi nomor HP.</li> <li>6. Pengguna mengisi alamat lengkap.</li> <li>7. Klik tombol ‘Daftar’.</li> </ol>	Pengguna tidak berhasil masuk dan akan terdapat notifikasi untuk mengisi field yang dikosongkan.

<b>TC02004</b>	Daftar	<b>Tidak Berhasil.</b> Pegguna daftar dengan mengisi semua field pada <i>form</i> registrasi dengan benar, tetapi dengan data <i>password</i> yang tidak sesuai.	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman daftar.</li> <li>2. Pengguna mengisi nama lengkap .</li> <li>3. Pengguna mengisi <i>email</i>.</li> <li>4. Pengguna mengisi <i>password</i> dan konfirmasi <i>password</i> yang mengandung kata-kata '<i>password</i>'.</li> <li>5. Pengguna mengisi nomor HP.</li> <li>6. Pengguna mengisi alamat lengkap.</li> <li>7. Klik tombol 'Daftar'.</li> </ol>	Pengguna tidak berhasil daftar dan akan muncul sebuah notifikasi ( <i>toast</i> ) 'Registrasi Gagal'.
<b>TC02000</b>	Daftar	<b>Tidak Berhasil.</b> Pegguna daftar dengan mengisi semua field pada <i>form</i> registrasi dengan benar, tetapi dengan data <i>password</i> dan konfirmasi yang tidak sama.	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman daftar.</li> <li>2. Pengguna mengisi nama lengkap .</li> <li>3. Pengguna mengisi <i>email</i>.</li> <li>4. Pengguna mengisi <i>password</i> dan konfirmasi <i>password</i> yang berbeda.</li> <li>5. Pengguna mengisi nomor HP dan alamat lengkap.</li> <li>6. Klik tombol 'Daftar'.</li> </ol>	Pengguna tidak dapat melakukan daftar dan terdapat sebuah notifikasi bahwa <i>password</i> tidak sama.

### 5) Skenario Cari Produk

a. Prekondisi :-

b. *Test data* : Data yang dimasukkan ke *field* pencarian (Tabel 3.7).

Tabel 1.7  
*Test data* cari produk

keyword pencarian	expected result
Kaos	Ditemukan
Kaos putih	Ditemukan
Produk yang tidak ada	tidak ditemukan
Null	semua produk

c. *Test case* : Tabel 3.8

Tabel 1.8  
TC03 - *Test case* cari produk

<i>Test Case ID</i>	<i>Feature</i>	<i>Detail Test Case</i>	<i>Test Step</i>	<i>Expected Result</i>
<b>TC03001</b>	Cari	<b>Berhasil.</b> Pegguna mencari produk yang tersedia pada Database dengan keyword tertentu.	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman utama.</li> <li>2. Pengguna mengisi field. 'search' pada navbar dengan keyword tertentu.</li> <li>3. Pengguna menekan tombol keyboard 'Enter' atau klik ikon 'cari'.</li> <li>4. Aplikasi akan menampilkan list produk hasil pencarian.</li> <li>5. Pengguna memilih salah satu produk hasil pencarian.</li> </ol>	Pengguna akan dialihkan menuju halaman detail produk yang sesuai.
<b>TC03002</b>	Cari	<b>Tidak Berhasil.</b> Pegguna mencari produk yang tidak tersedia pada Database dengan keyword tertentu.	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman utama.</li> <li>2. Pengguna mengisi field 'search' pada navbar dengan keyword tertentu.</li> <li>3. Pengguna menekan tombol keyboard 'Enter' atau klik ikon 'cari'.</li> </ol>	Aplikasi akan menampilkan bahwa produk yang terkait dengan keyword tidak ditemukan.
<b>TC03003</b>	Cari	<b>Tidak Berhasil.</b> Pegguna mencari produk tanpa mengisi field 'search'.	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman utama.</li> <li>2. Pengguna tidak mengisi field 'search' pada navbar.</li> <li>3. Pengguna menekan tombol keyboard 'Enter' atau klik ikon 'cari'.</li> </ol>	Pengguna akan dialihkan ke halaman semua produk.

6) Skenario lihat detail produk

a. Prekondisi : -

b. *Test data* : -

c. *Test case* : Tabel 3.9

Tabel 1.9  
TC04 - Test case lihat detail produk

<i>Test Case ID</i>	<i>Feature</i>	<i>Detail Test Case</i>	<i>Test Step</i>	<i>Expected Result</i>
<b>TC04001</b>	Detail Produk	<b>Berhasil.</b> Pengguna melihat detail produk dari salah satu produk pada halaman hasil pencarian atau halaman utama.	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman utama.</li> <li>2. Pengguna memilih salah satu produk rekomendasi.</li> <li>3. Pengguna akan dialihkan ke halaman detail produk sesuai dengan pilihan.</li> </ol>	Pengguna akan melihat informasi detail produk berupa gambar, nama, deskripsi, harga, kuantitas, dan sebuah tombol untuk menambahkan produk ke keranjang.
<b>TC04002</b>	Detail Produk	<b>Tidak Berhasil.</b> Pengguna menambahkan produk ke dalam keranjang dengan mengisi kuantitas kurang dari 1.	<ol style="list-style-type: none"> <li>1. Pengguna mencari sebuah produk pada field 'search' navbar.</li> <li>2. Aplikasi akan menampilkan list produk hasil pencarian.</li> <li>3. Pengguna memilih salah satu produk hasil pencarian.</li> <li>4. Pengguna menentukan kuantitas produk kurang dari 1.</li> </ol>	Tombol 'tambahkan ke keranjang' akan terlihat tidak aktif ( <i>disabled</i> ) .
<b>TC04003</b>	Detail Produk	<b>Tidak Berhasil.</b> Pengguna mencoba untuk mengganti link detail produk dengan id produk yang tidak terdaftar.	<ol style="list-style-type: none"> <li>1. Pengguna memasukkan produk id yang tidak terdaftar pada halaman '/product/:id'.</li> </ol>	Pengguna akan melihat tampilan 'halaman tidak ditemukan'.

#### 7) Skenario halaman profil

- a. Prekondisi : Pengguna sudah terautentikasi pada sistem.
- b. *Test data* : Merupakan data yang akan diuji kesesuaiannya (*assertion*) pada halaman profil, dapat dilihat pada Tabel 3.10.

Tabel 1.10  
Test data halaman profil

nama	Email	Passwor d	no hp	alamat lengkap
yukjahit	<u>yukjahit@e mail.com</u>	yukjahit1 23	0812345 6789	Alamat lengkap yukjahit
yukjahit2	<u>yukjahit2@ email.com</u>	yukjahit1 23	0812345 6789	Alamat lengkap yukjahit (2)

c. Test case : Tabel 3.11.

Tabel 1.11  
TC05 - Test case halaman profil

Test Case ID	Feature	Detail Test Case	Test Step	Expected Result
TC05001	Profil	<b>Berhasil.</b> Pegguna melihat informasi detail akun.	1. Pengguna membuka halaman profil.	Pegguna akan melihat data-data mengenai akun sesuai dengan data yang didaftarkan.
TC05002	Profil	<b>Berhasil.</b> Pegguna mencoba untuk merubah data.	1. Pengguna membuka halaman profil. 2. Klik tombol 'Ubah data'.	Aplikasi akan menampilkan sebuah modal 'form' yang sudah diisi oleh data yang didaftarkan sebelumnya.
TC05003	Profil	<b>Berhasil</b> Pegguna tidak jadi untuk merubah data.	1. Pengguna membuka halaman profil. 2. Klik tombol 'Ubah data'. 3. Klik icon 'tutup' diatas form.	modal 'form' akan tertutup dan data akun tidak akan mengalami perubahan.

<b>TC05004</b>	Profil	<b>Berhasil.</b> Pegguna mengganti data akun seperti ‘nama, alamat dan/atau nomor hp’.	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman profil.</li> <li>2. Klik tombol ‘Ubah data’.</li> <li>3. Pengguna merubah data ‘nama, alamat lengkap dan nomor hp’.</li> <li>4. Klik tombol ‘Ubah’.</li> </ol>	Aplikasi akan merubah data pada akun dan data-data pada halaman profil akan mengalami perubahan dalam beberapa saat.
<b>TC05005</b>	Profil	<b>Tidak Berhasil.</b> Pegguna mengganti data akun tanpa memberikan perubahan data.	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman profil.</li> <li>2. Klik tombol ‘Ubah data’</li> <li>3. Pengguna tidak merubah data ‘nama, alamat lengkap dan nomor hp’.</li> </ol>	Tombol ‘Ubah’ tidak akan bisa diklik ( <i>disabled</i> ).
<b>TC05006</b>	Profil	<b>Tidak Berhasil.</b> Pegguna mengganti data akun tanpa mengisi seluruh field yang tersedia.	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman profil.</li> <li>2. Klik tombol ‘Ubah data’.</li> <li>3. Pengguna tidak mengisi seluruh field yang tersedia.</li> </ol>	Tombol ‘Ubah’ akan terlihat tidak aktif ( <i>disabled</i> ).
<b>TC05007</b>	Profil	<b>Tidak Berhasil.</b> Pegguna mengganti data akun tanpa mengisi salah satu field yang tersedia.	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman profil.</li> <li>2. Klik tombol ‘Ubah data’</li> <li>3. Pengguna tidak mengisi field ‘nama’.</li> </ol>	Tombol ‘Ubah’ akan terlihat tidak aktif ( <i>disabled</i> ).

## 8) Skenario daftar transaksi

- a. Prekondisi : Pengguna telah terautentikasi pada sistem.
- b. *Test data* : Tabel 3.12.

Tabel 1.12  
*Test data* daftar transaksi

<i>Email</i>	hasOrder
<u>yukjahitOrder@email.com</u>	Yes
<u>yukjahitNoOrder@email.com</u>	No

Dimana kolom email merupakan email dari akun pengguna, dan kolom *hasOrder* merupakan data apakah akun tersebut sudah melakukan transaksi pada sistem.

c. *Test case* : Tabel 3.13

Tabel 1.13  
TC06 - *Test case* daftar transaksi

<i>Test Case ID</i>	<i>Feature</i>	<i>Detail Test Case</i>	<i>Test Step</i>	<i>Expected Result</i>
<b>TC06001</b>	Order	<b>Berhasil.</b> Pengguna melihat daftar transaksi (belum pernah).	1. Pengguna membuka halaman daftar transaksi.	Aplikasi akan menampilkan tulisan “Belum ada transaksi”.
<b>TC06002</b>	Order	<b>Berhasil.</b> Pengguna melihat daftar transaksi (sudah pernah).	1. Pengguna membuka halaman daftar transaksi.	Aplikasi akan menampilkan seluruh transaksi yang telah dilakukan pada akun.
<b>TC06003</b>	Order	<b>Berhasil.</b> Pengguna melihat detail dari salah satu transaksi.	1. Pengguna membuka halaman daftar transaksi 2. Klik tombol ‘Lihat detail transaksi’.	Aplikasi akan menampilkan sebuah modal yang menampilkan transaksi secara detail.
<b>TC06004</b>	Order	<b>Berhasil.</b> Pengguna mencoba untuk membatalkan salah satu transaksi yang dapat dibatalkan.	1. Pengguna membuka halaman daftar transaksi. 2. Klik tombol ‘Batalkan’. 3. Konfirmasi pembatalan.	Akan terdapat notifikasi berupa <i>toast</i> bahwa Transaksi telah dibatalkan.

#### 9) Skenario keranjang dan *checkout*

a. Prekondisi : Pengguna terautentikasi dalam sistem

b. *Test data* : -

c. *Test case* : Tabel 3.14

Tabel 1.14  
TC07 - *Test case* keranjang dan *checkout*

<i>Test Case ID</i>	<i>Feature</i>	<i>Detail Test Case</i>	<i>Test Step (Simplify)</i>	<i>Expected Result</i>
<b>TC07001</b>	Keran- jang	<b>Berhasil.</b> Pegguna melihat halaman keranjang kosong.	1. Pegguna membuka halaman Keranjang.	Pegguna akan melihat tampilan bahwa 'keranjangmu kosong'.
<b>TC07002</b>	Keran- jang	<b>Berhasil.</b> Pegguna menambahkan produk ke dalam keranjang.	1. Pegguna memilih salah satu produk terbaru pada halaman utama. 2. Pegguna menentukan jumlah produk yang akan dimasukkan ke keranjang. 3. Klik tombol 'Tambah ke Keranjang'. 4. Pegguna membuka halaman keranjang.	Pegguna akan mendapatkan sebuah <i>toast</i> bahwa produk berhasil ditambahkan ke keranjang. Kemudian produk tersebut akan muncul pada halaman keranjang dengan jumlah yang telah dimasukkan.
<b>TC07005</b>	Keran- jang	<b>Berhasil.</b> Pegguna melihat halaman keranjang dengan produk.	1. Pegguna membuka halaman Keranjang.	Pegguna akan melihat semua produk yang telah dimasukkan kedalam keranjang, termasuk total harga yang perlu dibayarkan.
<b>TC07004</b>	Keran- jang	<b>Berhasil</b> Pegguna menambahkan produk yang sama dengan yang ada pada keranjang sebelumnya.	1. Pegguna membuka halaman keranjang. 2. Pegguna memilih salah satu produk yang tersedia pada keranjang. 3. Pegguna akan dialihkan ke detail produk.	Pegguna akan mendapatkan <i>toast</i> 'produk xx berhasil ditambahkan ke keranjang'. Kemudian produk tersebut akan muncul pada

			4. Pengguna menentukan kuantitas produk.	halaman keranjang dengan total jumlah yang telah
			5. Klik tombol ‘Tambah ke Keranjang’.	ditambahkan.
			6. Pengguna membuka halaman keranjang.	
<b>TC07005</b>	Keran- jang	<b>Berhasil</b> Pengguna menambahkan produk yang berbeda dengan yang ada pada keranjang sebelumnya.	1. Pengguna mencari produk lain yang berbeda dari produk pada keranjang. 2. Pengguna memilih produk. 3. Pengguna dialihkan ke detail produk. 4. Pengguna menentukan kuantitas produk. 5. Klik tombol ‘Tambah ke Keranjang’. 6. Pengguna membuka halaman keranjang.	Pengguna akan mendapatkan <i>toast</i> ‘produk xx berhasil ditambahkan ke keranjang’. Kemudian akan terdapat dua produk yang akan muncul pada halaman keranjang.
<b>TC07006</b>	Keran- jang	<b>Berhasil.</b> Pengguna menghapus salah satu produk yang ada pada keranjang.	1. Pengguna membuka halaman keranjang. 2. Pengguna menekan tombol ikon ‘hapus’ pada sebelah ujung kanan produk.	Produk tersebut dihapus dari keranjang.
<b>TC07007</b>	Check- out	<b>Berhasil.</b> Pengguna melanjutkan ke checkout dengan mengisi notes dengan benar.	1. Pengguna membuka halaman keranjang. 2. Pengguna mengisi field catatan. 3. Klik tombol ‘Lakukan pembayaran’. 3. Klik tombol ‘Checkout’.	Pengguna akan dialihkan ke halaman checkout yang menampilkan link pembayaran.
<b>TC07008</b>	Check- out	<b>Berhasil.</b> Pengguna melakukan pembayaran dengan	1. Klik link pembayaran pada halaman checkout.	Transaksi tersebut akan dinyatakan selesai pada halaman order.

meng-klik link yang ditampilkan.	<ol style="list-style-type: none"> <li>2. Pengguna akan dialihkan ke halaman midtrans.</li> <li>3. Pengguna memasukkan PIN dan mengonfirmasi pembayaran.</li> <li>4. Pengguna Kembali ke halaman checkout.</li> </ol>
----------------------------------	---

10) Skenario Sistem (*system testing*)

Tabel 1.15  
TC08 - *Test case* untuk *system testing*

<i>Test Case ID</i>	<i>Test Step</i>	<i>Expected Result</i>
<b>TC08</b>	<ol style="list-style-type: none"> <li>1. Pengguna mendaftar pada aplikasi web YukJahit dengan data yang benar.</li> <li>2. Pengguna masuk dengan menggunakan kredensial yang telah didaftarkan.</li> <li>3. Pengguna memilih salah satu produk terbaru pada halaman utama.</li> <li>4. Pengguna menambahkan produk tersebut ke keranjang.</li> <li>5. Pengguna mencoba untuk melakukan pencarian produk yang berbeda.</li> <li>6. Pengguna menambahkan produk tersebut ke keranjang.</li> <li>7. Pengguna mengganti kuantitas produk yang kemudian menambahkannya ke keranjang.</li> <li>8. Pengguna melakukan <i>checkout</i> setelah mengisi catatan.</li> <li>9. Pengguna melakukan pembayaran dengan meng-klik link pembayaran.</li> <li>10. Pengguna melihat status pembayaran pada halaman daftar transaksi.</li> <li>11. Pengguna melihat halaman profil.</li> <li>12. Pengguna keluar dari aplikasi.</li> </ol>	Semua fitur pada aplikasi web dapat dijalankan dengan baik.

## 11) Skenario halaman produk – admin

## a. Prekondisi:

- Terautentikasi dalam sistem menggunakan hak administrator pada halaman <https://yukjahit-admin.vercel.app>.
- Membuka halaman <https://yukjahit-admin.vercel.app/products>.

b. *Test data* :-c. *Test case* : Tabel 3.16.

Tabel 1.16  
TC09 - *Test case* halaman produk (admin)

<i>Test Case ID</i>	<i>Feature</i>	<i>Detail Test Case</i>	<i>Test Step</i>	<i>Expected Result</i>
TC09001	Product	<b>Berhasil.</b> Admin melihat seluruh produk yang ada pada <i>database</i> .	1. Admin melihat seluruh produk dengan informasi seperti gambar, nama produk, harga dengan masing-masing produk memiliki opsi untuk ubah dan hapus.	Admin akan melihat seluruh produk pada <i>database</i> .
TC09002	Product	<b>Berhasil.</b> Admin menambahkan sebuah produk ke <i>database</i> .	1. Admin menekan tombol 'Tambah Produk' pada bagian kanan atas halaman. 2. Admin dialihkan ke halaman ubah produk. 3. Admin menambahkan produk dengan memasukkan data berupa 'nama', 'harga', 'deskripsi', memilih 'kategori', dan mengunggah 'gambar'. 4. Admin menekan tombol 'Tambah'.	Aplikasi akan menampilkan sebuah <i>toast</i> yang menampilkan bahwa produk telah berhasil ditambahkan. Produk kemudian akan tersimpan pada <i>database</i> .
TC09003	Product	<b>Berhasil</b> Admin mengubah data dari salah satu produk pada <i>database</i> .	1. Admin memilih salah satu produk dan menekan tombol 'ubah'. 2. Admin akan dialihkan ke halaman ubah produk.	Aplikasi akan menampilkan sebuah <i>toast</i> yang menampilkan

			3. Admin kemudian memilih data yang akan diubah.	bahwa produk telah berhasil diubah. Produk tersebut akan tersimpan pada <i>database</i> dengan data terbaru.
			4. Admin menekan tombol ‘Ubah’.	
<b>TC09004</b>	Product	<b>Berhasil.</b> Admin menghapus data dari salah satu produk pada <i>database</i> .	5. Admin memilih salah satu produk dan menekan tombol “Hapus”.	Produk terhapus pada <i>database</i> dan tidak akan muncul pada halaman produk.

12) Skenario halaman produk – admin

- a. Prekondisi: Terotentikasi dalam sistem menggunakan hak administrator pada halaman <https://yukjahit-admin.vercel.app>.
- b. *Test data* : -
- c. *Test case* : Tabel 3.17.

Tabel 1.17  
TC10 - *Test case* halaman transaksi (admin)

<i>Test Case ID</i>	<i>Feature</i>	<i>Detail Test Case</i>	<i>Test Step</i>	<i>Expected Result</i>
<b>TC10001</b>	Order	<b>Berhasil.</b> Admin melihat daftar dari seluruh transaksi yang telah dilakukan pada sistem.	1. Admin membuka halaman transaksi 2. Admin melihat daftar dari seluruh transaksi.	Aplikasi akan menampilkan seluruh transaksi yang telah dilakukan dengan data-data berupa ‘waktu’, ‘status’, ‘produk’, dan ‘total harga’.
<b>TC10002</b>	Order	<b>Berhasil.</b> Admin melihat detail dari salah satu transaksi.	1. Admin membuka halaman transaksi. 2. Admin memilih salah satu transaksi.	Aplikasi akan menampilkan sebuah modal detail dari

			3. Admin menekan tombol ‘Lihat Detail Transaksi’.	transaksi dengan data berupa ‘detail transaksi’, ‘info penerima’, ‘produk’ dan ‘metode pembayaran’.
<b>TC10003</b>	Order	<b>Berhasil.</b>	2. Pengguna membuka halaman transaksi 3. Admin menghapus salah satu transaksi dengan menekan tombol ‘Hapus Transaksi’	Aplikasi akan menampilkan sebuah <i>toast</i> bahwa ‘Transaksi telah dihapus’.

#### c. Test Execute

Eksekusi akan dijalankan dengan teknik *automated testing* dengan menggunakan bantuan *automated testing tools* Katalon Studio.

#### d. Analisis

Menganalisis hasil akhir dari pengujian secara keseluruhan. Data yang akan dievaluasi adalah waktu yang dibutuhkan untuk membuat *test case* dan menjalankan *test execution*, *test steps generated*, dan jumlah kesalahan yang ditemukan.

### 1.1.4 Studi Deskriptif II

Guna mengetahui tingkat keberhasilan dari penelitian ini, diperlukan sebuah analisis dan pembahasan yang menunjukkan apakah model-based testing dapat memberikan hasil pengujian yang lebih baik serta lebih efektif dalam penggunaan sumber daya. Hal pertama yang dilakukan adalah melaksanakan *functional testing* pada aplikasi web *e-commerce* YukJahit dengan menggunakan *exploratory testing* (Katalon Studio) dan *model-based testing* (GraphWalker).

Pada pengujiannya (*test execution*), MBT (GraphWalker) akan menggunakan *test path generation* “random(edge\_coverage(100))”, yang berarti model akan dieksekusi secara acak dan akan berhenti ketika semua edge telah dilewati. Hal tersebut membuat pengujian akan menghasilkan data yang beragam. Sehingga pengujian akan dilakukan pengulangan sebanyak 20 (dua puluh) kali untuk

mendapatkan data yang lebih akurat. Sedangkan untuk *exploratory testing* (Katalon Studio), data yang dihasilkan relatif konsisten sehingga membuat pengulangan tidak diperlukan.

#### 1.1.4.1 Analisis Data

Pada tahap ini data yang telah didapat dari kedua metode pengujian akan diolah dan dianalisis. Mengikuti penelitian serupa yang telah dilakukan oleh Garousi dkk., (2021), data/metrik yang akan diambil sendiri merupakan:

a. waktu/usaha

Mengikuti *literature review* yang dilakukan oleh Bluemke dan Malanowska (Bluemke dan Malanowska, 2021), waktu ini dapat diambil sesuai dari waktu pengerjaan yang dibutuhkan oleh *tester* untuk menyelesaikan pembuatan *test case* sekaligus *test scripting* baik dalam satuan bulan/hari/jam/menit/detik. Pada penelitian ini, satuan yang akan digunakan adalah jam. Metrik ini menunjukkan nilai dari usaha yang diperlukan untuk melakukan pengujian.

b. *test execute*

Merupakan waktu yang dibutuhkan setiap teknik/*tools* untuk menjalankan seluruh *test case* atau *test suite* dalam satuan detik (s). Data ini diambil dari hasil yang ditampilkan oleh *tools* (GraphWalker dan Katalon Studio) pada masing-masing teknik ketika keseluruhan *test execution* telah selesai. Metrik ini menunjukkan nilai performa dari *tools* pengujian.

c. *test steps generated*

Merupakan jumlah tahapan atau langkah yang dibutuhkan pada masing-masing teknik/*tools* untuk menjalankan seluruh *test case* atau *test suite*. Data ini diambil dari hasil yang ditampilkan oleh *tools* (GraphWalker dan Katalon Studio) pada masing-masing teknik. Metrik ini menunjukkan nilai komprehensif dari teknik pengujian. Bisa juga disebut sebagai nilai cakupan pengujian atas semua aspek fungsionalitas yang akan diuji.

d. jumlah *bug* yang ditemukan

Merupakan jumlah *bug* yang ditemukan pada keseluruhan *test suite*. Data ini diambil dari hasil yang ditampilkan oleh *tools* (GraphWalker dan Katalon Studio) pada masing-masing teknik. Metrik ini menunjukkan nilai efektivitas

dari masing-masing teknik pengujian. Semakin banyak *bug* yang ditemukan, maka semakin efektif teknik pengujian tersebut.

Data-data yang telah diperoleh tersebut kemudian akan dibandingkan. MBT akan dikatakan lebih efektif jika penggunaan waktu/usaha dan *test execute* lebih rendah dan/atau jika nilai *test steps generated* dan jumlah *bug* yang ditemukan lebih tinggi daripada ET.

#### 1.1.4.2 Penarikan Kesimpulan

Kesimpulan akan diambil berdasarkan hasil dari analisis data. Kesimpulan ini akan menunjukkan apakah benar bahwa MBT dapat memberikan hasil pengujian yang lebih efektif jika dibandingkan dengan pengujian ET menggunakan Katalon Studio untuk *functional testing* aplikasi web YukJahit.

## 1.2 Alat dan Bahan Penelitian

Perangkat keras yang digunakan sebagai alat penelitian dalam pengembangan dan pengujian produk memiliki spesifikasi sebagai berikut:

1. Prosesor Intel Core i5-7200U @2.50GHz
2. Nvidia Geforce 920MX
3. RAM 12GB DDR4
4. SSD 500GB SATA

Sedangkan untuk perangkat lunak, yaitu sebagai berikut:

1. Microsoft Windows 10 Home SL 22H2
2. Visual Studio Code
3. IntelliJ IDEA 2023.1
4. Node JS v18.15.0
5. MongoDB v7.0.2
6. Express JS v4.18.2
7. React.js – Vite v4.3.9
8. GraphWalker v4.3.2
9. Katalon Studio v8.6.0
10. Selenium WebDriver v4.9.1
11. Selenium IDE v3.17.2
12. JUnit v4.13.2
13. Google Chrome

### 1.3 Instrumen Penelitian

Melakukan eksperimen (non-partisipan) *functional testing* pada aplikasi web *e-commerce* YukJahit dengan menggunakan dua teknik yaitu teknik *model-based testing* menggunakan GraphWalker. Dimana GraphWalker sudah terbukti dan layak digunakan oleh banyak peneliti untuk menguji sebuah aplikasi web dan mobile (Garousi dkk., 2021; Kilincceker dkk., 2021; Zafar dkk., 2021). Dilanjutkan menggunakan teknik *exploratory testing* dengan menggunakan Katalon Studio yang juga sudah terbukti untuk layak digunakan pada pengujian aplikasi web oleh Tirza Tempomona (2022) dan Zulianto dkk. (2021).