# BAB III ANALISIS DAN PERANCANGAN

## Analisis

### Identifikasi Masalah

Identifikasi masalah yang memiliki kekurangan atau kelemahan sistem saat ini dapat dibagi menjadi pokok-pokok permasalahan antara lain:

1. Aplikasi Space Venturo saat ini belum memiliki sebuah fitur untuk membuat *Test Scenario* sehingga *Quality Assurance Engineer* mencatat skenario yang akan dicoba menggunakan *excel*.
2. Aplikasi Space Venturo saat ini belum memiliki sebuah fitur untuk membuat *Test Run* sehingga hasil run dari skenario yang sudah dibuat masih didokumentasikan menggunakan *excel*.
3. Aplikasi Space Venturo saat ini belum memiliki fitur untuk menampilkan hasil laporan dari testing yang sudah dijalankan *Quality Assurance Engineer* sehingga *Quality Assurance Engineer* masih menggunakan *excel* tanpa adanya visual.

Masalah yang dihadapi tersebut juga dapat diidentifikasikan dan dianalisa menggunakan tabel sebab akibat yang dapat diuraikan pada tabel berikut:

Tabel 3. 1 Tabel Sebab Akibat

|  |  |  |
| --- | --- | --- |
| Permasalahan | Solusi | Manfaat |
| Aplikasi Space Venturo saat ini belum memiliki sebuah fitur untuk membuat *Test Scenario* sehingga *Quality Assurance Engineer* mencatat skenario yang akan dicoba menggunakan excel | Membuat fitur untuk membuat *Test Scenario* | Memudahkan *Quality Assurance Engineer* mencatat skenario sehingga penjaminan mutu produk dapat terjamin. |
| Aplikasi Space Venturo saat ini belum memiliki sebuah fitur untuk membuat *Test Run* sehingga hasil run dari skenario yang sudah dibuat masih didokumentasikan menggunakan excel. | Membuat fitur untuk membuat *Test Run* | Memudahkan *Quality Assurance Engineer* menjalankan *Test Run* dan mendapatkan hasil berdasarkan *Test Scenario* yang sudah dibuat sebelumnya. |
| Aplikasi Space Venturo saat ini belum memiliki fitur untuk menampilkan hasil laporan dari testing yang sudah dijalankan *Quality Assurance Engineer* sehingga *Quality Assurance Engineer* masih menggunakan excel tanpa adanya visual. | Membuat fitur untuk menampilkan laporan dari *Test Run* yang sudah dijalankan. | Memudahkan *Quality Assurance Engineer* untuk menyajikan informasi terkait *Test Run* yang sudah dijalankan sebelumnya. |

### Pemecahan Masalah

Berdasarkan identifikasi yang sudah dijabarkan sebelumnya, solusi dari masalah dapat diturunkan. Dimana diperlukan sebuah Sistem *Test Management System* seperti contohnya *Test Run* dan *Test Scenario* untuk mengolah dan menyimpan informasi terkait bagaimana pengujian penjaminan mutu produk harus dikerjakan agar mempermudah dalam pencatatan dan pelaksanaan penjaminan mutu produk. Serta, diperlukan juga sebuah fitur laporan informasi terkait *Test Run* yang sudah dijalankan sebelumnya agar memudahkan pengguna dalam hal ini *Quality Assurance Engineer* dalam menampilkan informasi tersebut ke atasan dan mempresentasikannya.

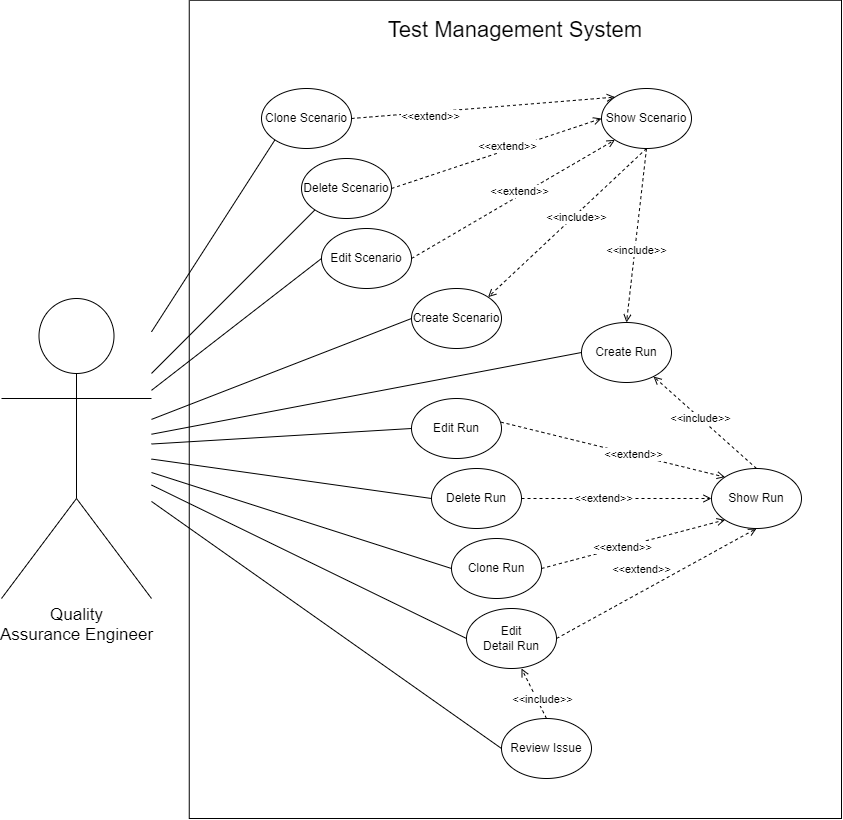
## Perancangan

### Perancangan Sistem

Perancangan sistem yang dilakukan pada saat proses pembuatan akan diimplementasikan pada aplikasi *Test Management System* Space Venturo dan akan dirancang berdasarkan orientasi terhadap proses dan diuraikan pada bagian ini dengan menggunakan perancangan *Use Case Diagram*.

#### **Use Case Diagram**

*Use Case* adalah diagram yang menjelaskan tentang siapa saja aktor yang menggunakan sistem. *Use Case* juga menjelaskan apa saja yang bisa dilakukan aktor tersebut di dalam sistem.



Gambar 3. 1 Use Case Diagram

1. **Definisi Aktor**

Berikut jabaran mengenai aktor-aktor yang terdapat didalam aplikasi *Test Management* System Venturo Space.

Tabel 3. 2 Tabel Definisi Aktor

|  |  |  |
| --- | --- | --- |
| No | Aktor | Deskripsi |
| 1. | *Quality Assurance Engineer* | Aktor yang menggunakan aplikasi dan menggunakan penuh fitur *Test Management System*. |

1. **Use Case Scenario**

Tabel 3. 3 Tabel Use Case Create Test Scenario

|  |  |
| --- | --- |
| **Feature** | *Create Test Scenario* |
| **Pre-conditions** | 1. *User* berada diluar menu *Test Scenario* |
| **Post-conditions** | 1. *List Test Skenario* tampil beserta data baru |
| **Normal Flow** | 1. *User* memilih menu *Test Scenario* 2. *User* mendapatkan *List Test Scenario* 3. *User* menekan tombol *Create Scenario* 4. *User* mengisi *field* yang tersedia. 5. *User* kembali ke *List Test Scenario* dengan tambahan data baru. |
| **Alternative Flows** | Alternatif 1 (Membuat *Scenario* baru dengan field yang kosong)   1. *User* memilih menu *Test Scenario* 2. *User* mendapatkan *List Test Scenario* 3. *User* menekan tombol *Create Scenario* 4. *User* mengisi field dengan data kosong. 5. Muncul peringatan bahwa ada *field* yang kosong. |
| **Includes** | - |
| **Priority** | *High* (diprioritaskan ketika pengembangan sistem) |

Tabel 3. 4 Tabel Use Case Create Test Run

|  |  |
| --- | --- |
| **Feature** | *Create Test Run* |
| **Pre-conditions** | 1. *User* berada diluar menu *Test Run* |
| **Post-conditions** | 1. *List Test Run* tampil dengan data baru |
| **Normal Flow** | 1. *User* memilih menu *Test Run* 2. *User* mendapatkan *List Test Run* 3. *User* menekan tombol *Create Run* 4. *User* mengisi *field* yang tersedia 5. *User* memilih data *Test Scenario* serta *Assignee*. 6. *User* kembali ke *List Test Scenario* dengan tambahan data baru. |
| **Alternative Flows** | Alternatif 1 (Membuat *run* baru dengan field nama, *environment*, dan cases kosong)   1. *User* memilih menu *Test Run* 2. *User* mendapatkan *List Test Run* 3. *User* menekan tombol *Create Run* 4. *User* mengosongkan field nama, environment, dan cases 5. Muncul peringatan bahwa ada *field* yang kosong.   Alternatif 2 (Saat menambah *Test Scenario* mengosongkan *Assignee*)   1. *User* memilih menu *Test Run* 2. *User* mendapatkan *List Test Run* 3. *User* menekan tombol *Create Run* 4. *User* mengisi *field* yang tersedia 5. *User* memilih data *Test Scenario* tanpa memilih *Assignee*. 6. Muncul peringatan bahwa *Assignee* masih kosong. |
| **Includes** | - |
| **Priority** | *High* (diprioritaskan ketika pengembangan sistem) |

Tabel 3. 5 Tabel Use Case Edit Detail Test Run

|  |  |
| --- | --- |
| **Feature** | *Edit Detail Test Run* |
| **Pre-conditions** | 1. *User* berada diluar menu *Test Run* |
| **Post-conditions** | 1. *List Test Run* tampil dengan data detail yang berhasil diubah |
| **Normal Flow** | 1. *User* memilih menu *Test Run* 2. *User* mendapatkan *List Test Run* 3. *User* memilih salah satu *Test Run* 4. *User* mendapatkan detail dari *Test Run* tersebut. 5. *User* memilih salah satu *Scenario* didalam detail. 6. *User* mengganti Status dari *Step* didalam *Scenario*. 7. *User* mengubah semua Status pada tiap *Step* 8. *System* secara otomatis akan memberitahu bahwa *Scenario* tersebut *pass* atau *failed*. |
| **Alternative Flows** | Alternatif 1 (*User* menambah *Issue*)   1. *User* memilih menu *Test Run* 2. *User* mendapatkan *List Test Run* 3. *User* memilih salah satu *Test Run* 4. *User* mendapatkan detail dari *Test Run* tersebut. 5. *User* memilih salah satu Scenario didalam detail. 6. *User* mengganti Status dari *Step* didalam *Scenario*. 7. *User* menekan Status *Failed* pada salah satu *Step* 8. *User* mengisi formulir terkait *Issue* kenapa *Step* tersebut dapat gagal dijalankan. |
| **Includes** | - |
| **Priority** | *High* (diprioritaskan ketika pengembangan sistem) |

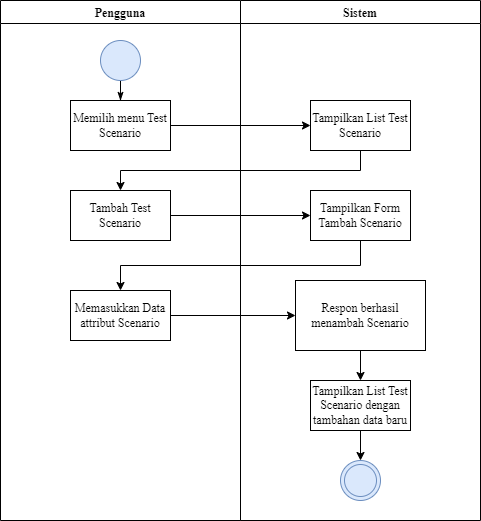
Tabel 3. 6 Tabel Use Case Review Issue

|  |  |
| --- | --- |
| **Feature** | *Review Issue* |
| **Pre-conditions** | 1. *User* berada diluar menu *Test Run* |
| **Post-conditions** | 1. *List Issue* tampil |
| **Normal Flow** | 1. *User* memilih menu *Test Run* 2. *User* mendapatkan *List Test Run* 3. *User* memilih salah satu *Test Run* 4. *User* mendapatkan detail dari *Test Run* tersebut. 5. *User* menekan tombol Lihat Selengkapnya. 6. *User* mendapatkan *List Review Issue.* |
| **Alternative Flows** | Alternatif 1 (*User* mengubah *Issue*)   1. *User* memilih menu *Test Run* 2. *User* mendapatkan *List Test Run* 3. *User* memilih salah satu *Test Run* 4. *User* mendapatkan detail dari *Test Run* tersebut. 5. *User* menekan tombol Lihat Selengkapnya. 6. *User* mendapatkan *List Review Issue.* 7. *User* mengganti *field* dari salah satu *Issue* 8. Muncul *popup* bahwa data berhasil diubah. |
| **Includes** | - |
| **Priority** | *High* (diprioritaskan ketika pengembangan sistem) |

#### **Activity Diagram**

Berikut adalah *activity diagram* dari aplikasi *Test Management System* Venturo Space yang akan digunakan :

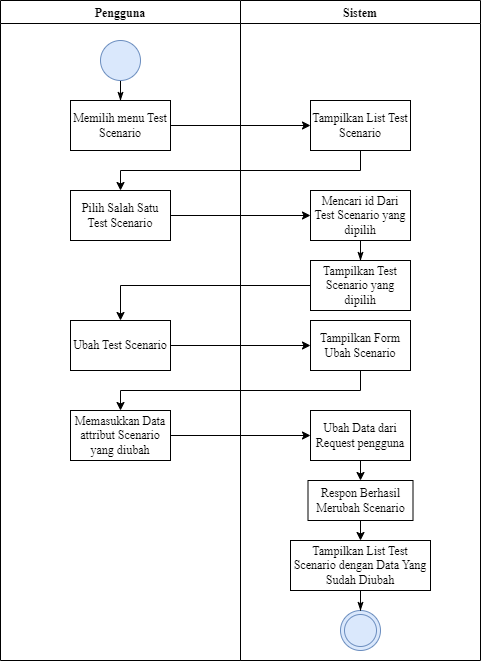
1. Menambah *Test Scenario*



Gambar 3. 2 Menambah Test Scenario

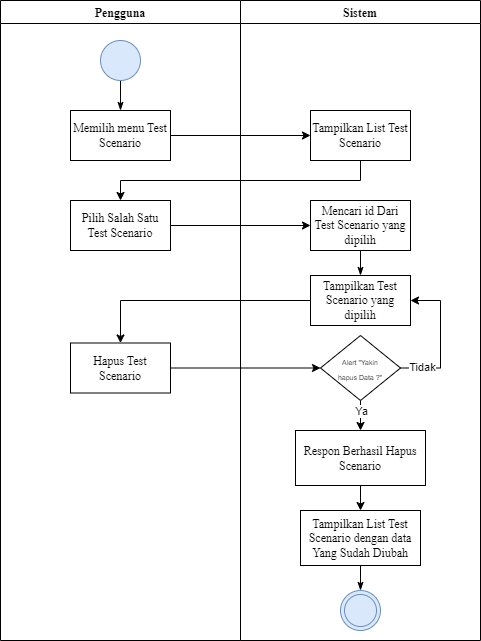
Pada gambar diagram diatas, pengguna yang sebelumnya sudah diarahkan untuk melihat list dari *Test Scenario*, bisa menambahkan *Test Scenario* baru lalu dihadapkan dengan *form* terkait *Test Scenario* apa yang diinginkan pengguna. Setelah selesai, sistem akan menyimpan data tersebut lalu mengarahkan pengguna kembali ke halaman *Test Scenario* dengan menampilkan *List* serta data baru yang sebelumnya sudah disimpan system.

1. Ubah *Test Scenario*



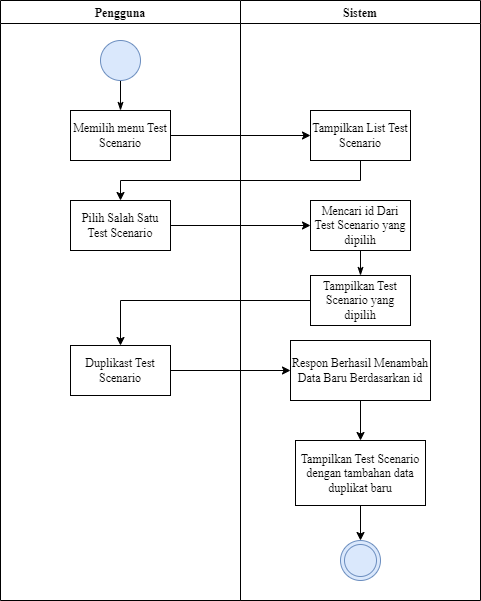
Gambar 3. 3 Menggubah Test Scenario

Pada gambar diagram diatas, pengguna yang sebelumnya sudah diarahkan untuk melihat *list* dari *Test Scenario*, bisa mengubah salah satu *scenario* yang dipilih pengguna. Nantinya, pengguna akan dihadapakan *form* terkait *field* apa saja yang ingin diubah oleh pengguna. Setelah selesai, sistem akan menyimpan data tersebut lalu mengarahkan pengguna kembali ke halaman *Test Scenario* dengan menampilkan *list* serta data yang baru saja diubah pengguna.

1. Hapus *Test Scenario*

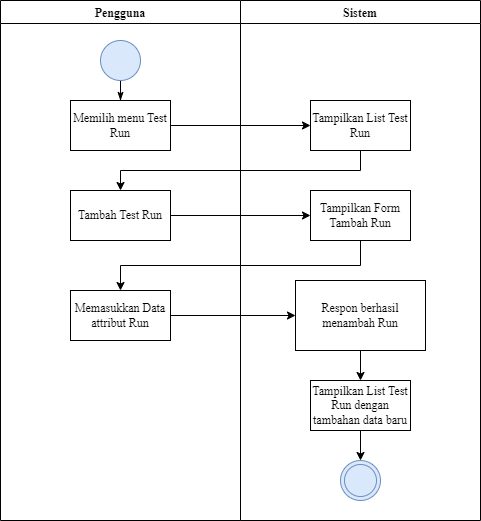
Gambar 3. 4 Menghapus Test Scenario

Pada gambar diagram diatas, pengguna yang sebelumnya sudah diarahkan untuk melihat *list* dari *Test Scenario*, bisa menghapus salah satu *scenario* yang dipilih pengguna. Nantinya, pengguna akan dihadapakan peringatan apakah yakin untuk menghapus data tersebut. Apabila pengguna menjawab tidak maka data tidak jadi dihapus. Dan apabila menjawab ya, sistem akan menghapus data tersebut dan menampilkan peringatan bahwa data berhasil dihapus lalu mengarahkan pengguna kembali ke halaman *Test Scenario* dengan menampilkan *list* data tersebut.

1. Duplikat *Test Scenario*

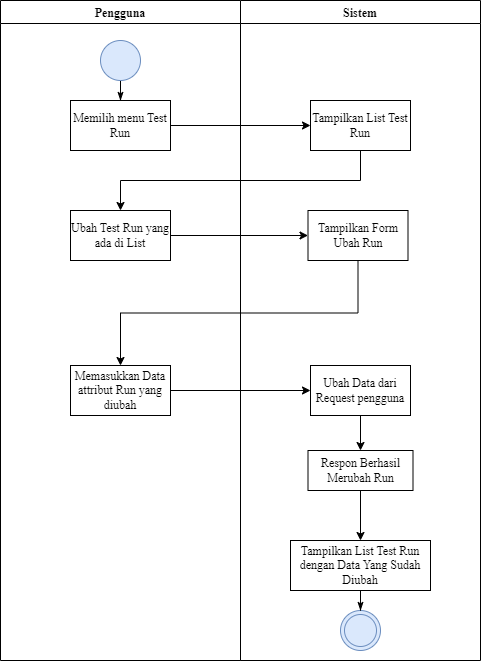
Gambar 3. 5 Duplikat Test Scenario

Pada gambar diagram diatas, pengguna yang sebelumnya sudah diarahkan untuk melihat *list* dari *Test Scenario*, bisa menduplikat salah satu *scenario* yang dipilih pengguna. Nantinya, sistem akan membuat data baru berdasarkan data *Test Scenario* yang dipilih. Setelah selesai, sistem akan menyimpan data tersebut lalu mengarahkan pengguna kembali ke halaman *Test Scenario* dengan menampilkan *list* serta data yang baru saja diduplikat pengguna.

1. Menambah *Test Run*

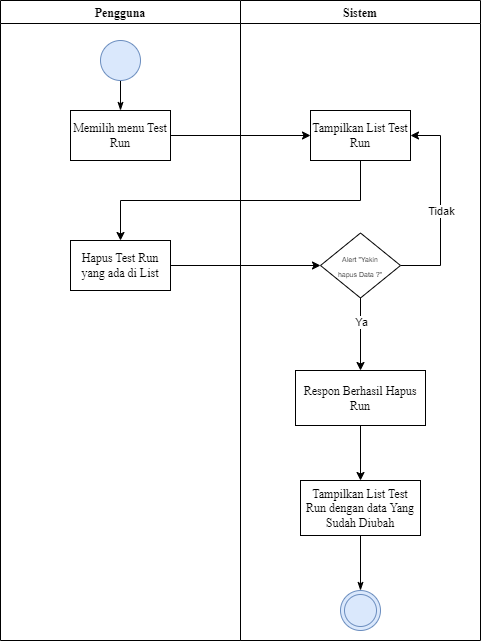
Gambar 3. 6 Menambah Test Run

Pada gambar diagram diatas, pengguna yang sebelumnya sudah diarahkan untuk melihat *list* dari *Test Run*, bisa menambahkan *Test Run* baru lalu dihadapkan dengan *form* terkait *Test Run* apa yang diinginkan pengguna. Setelah selesai, sistem akan menyimpan data tersebut lalu mengarahkan pengguna kembali ke halaman *Test Run* dengan menampilkan *list* serta data baru yang sebelumnya sudah disimpan sistem.

1. Ubah *Test Run*

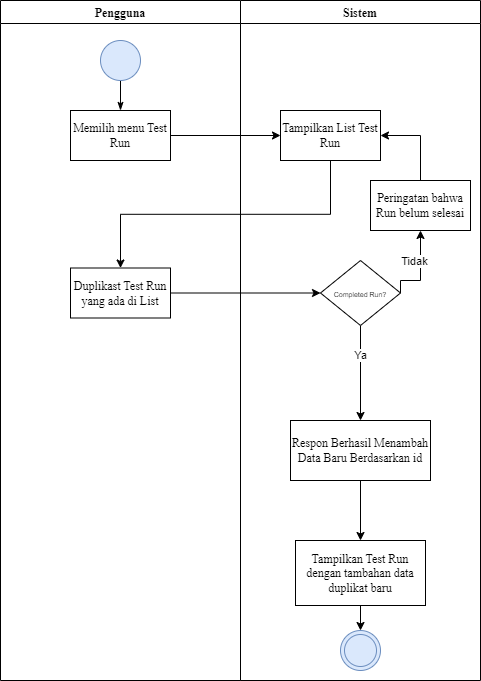
Gambar 3. 7 Menggubah Test Run

Pada gambar diagram diatas, pengguna yang sebelumnya sudah diarahkan untuk melihat *list* dari *Test Run*, bisa mengubah salah satu *run* yang dipilih pengguna. Nantinya, pengguna akan dihadapakan *form* terkait *field* apa saja yang ingin diubah oleh pengguna. Setelah selesai, sistem akan menyimpan data tersebut lalu mengarahkan pengguna kembali ke halaman *Test Run* dengan menampilkan list serta data yang baru saja diubah pengguna.

1. Delete *Test Run*

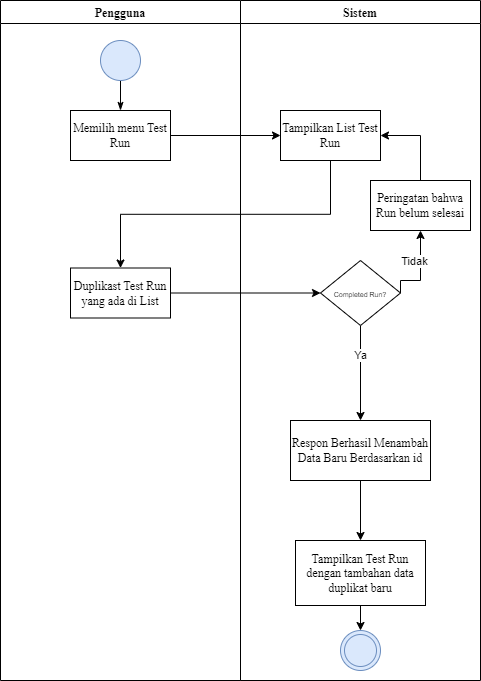
Gambar 3. 8 Menghapus Test Run

Pada gambar diagram diatas, pengguna yang sebelumnya sudah diarahkan untuk melihat *list* dari *Test Run*, bisa menghapus salah satu run yang dipilih pengguna. Nantinya, pengguna akan dihadapakan peringatan apakah yakin untuk menghapus data tersebut. Apabila pengguna menjawab tidak maka data tidak jadi dihapus. Dan apabila menjawab ya, sistem akan menghapus data tersebut dan menampilkan peringatan bahwa data berhasil dihapus lalu mengarahkan pengguna kembali ke halaman *Test Run* dengan menampilkan *list* data tersebut.

1. Duplikat *Test Run*

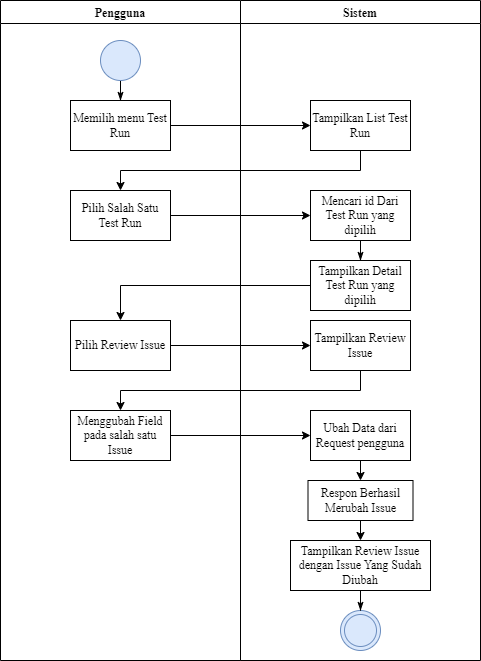
Gambar 3. 9 Duplikat Test Run

Pada gambar diagram diatas, pengguna yang sebelumnya sudah diarahkan untuk melihat *list* dari *Test Run*, bisa menduplikat salah satu run yang dipilih pengguna. Nantinya, sistem akan membuat data baru berdasarkan data *Test Run* yang dipilih. Setelah selesai, sistem akan menyimpan data tersebut lalu mengarahkan pengguna kembali ke halaman *Test Run* dengan menampilkan *list* serta data yang baru saja diduplikat pengguna

1. Ubah *Detail Test Run*

Gambar 3. 10 Ubah Detail Test Run

Pada gambar diagram diatas, pengguna yang sebelumnya sudah diarahkan untuk melihat *list* dari *Test Run*, bisa mengubah detail dari run yang dipilih pengguna. Nantinya, pengguna akan dihadapakan *form* terkait *field* apa saja yang ingin diubah oleh pengguna. Setelah selesai, sistem akan menyimpan data tersebut lalu mengarahkan pengguna kembali ke halaman *Detail Test Run* dengan menampilkan *run* dengan data yang baru saja diubah pengguna.

1. Ubah Review Issue

Gambar 3. 11 Ubah Review Issue

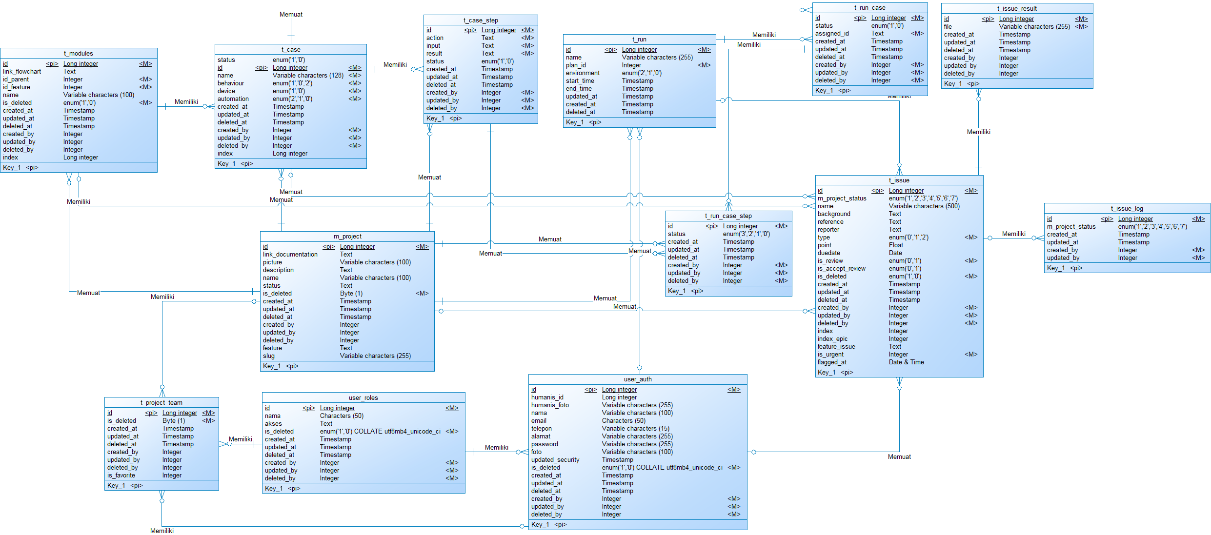
Pada gambar diagram diatas, pengguna yang sebelumnya sudah diarahkan untuk melihat *list* dari *Test Run*, bisa melihat dan mengubah issue dari *Test Run* yang sudah dijalankan. Nantinya, pengguna akan dihadapakan *form* terkait *field* apa saja yang ingin diubah oleh pengguna. Setelah selesai, sistem akan menyimpan data tersebut lalu mengarahkan pengguna kembali ke halaman *Review Issue* dengan menampilkan *list Review Issue* serta data yang baru saja diubah pengguna.

### Perancangan Data

Berikut merupakan gambaran perancangan kerangka *database* yang digunakan dalam pembuatan sistem ini. Data yang dimasukkan pengguna disimpan dalam database melalui *Restful API*. Gambar terlampir apabila gambar kurang jelas.

#### **Conceptual Data Model (CDM)**

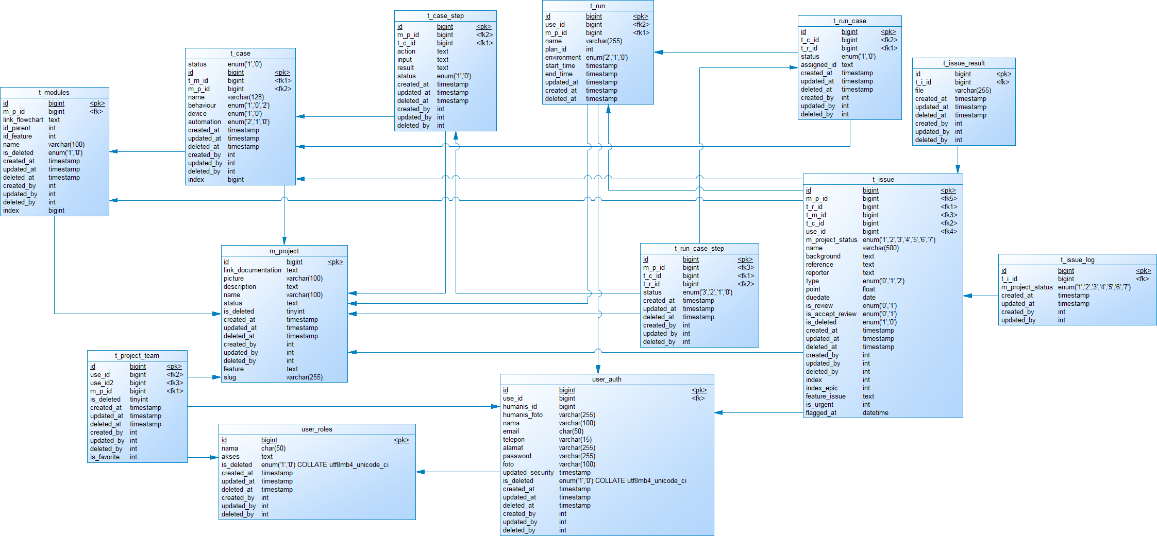
Pada *Conceptual Data Model* seperti yang terdapat pada gambar dibawah terdapat berbagai tabel yang digunakan pada sistem nantinya.



Gambar 3. 12 Conceptual Data Model

#### **Physical Data Model (PDM)**

Pada Physical Data Model dibawah ini terdapat hasil dari *Conceptual Data Model* dengan relasi *one to many*.



Gambar 3. 13 Physical Data Model

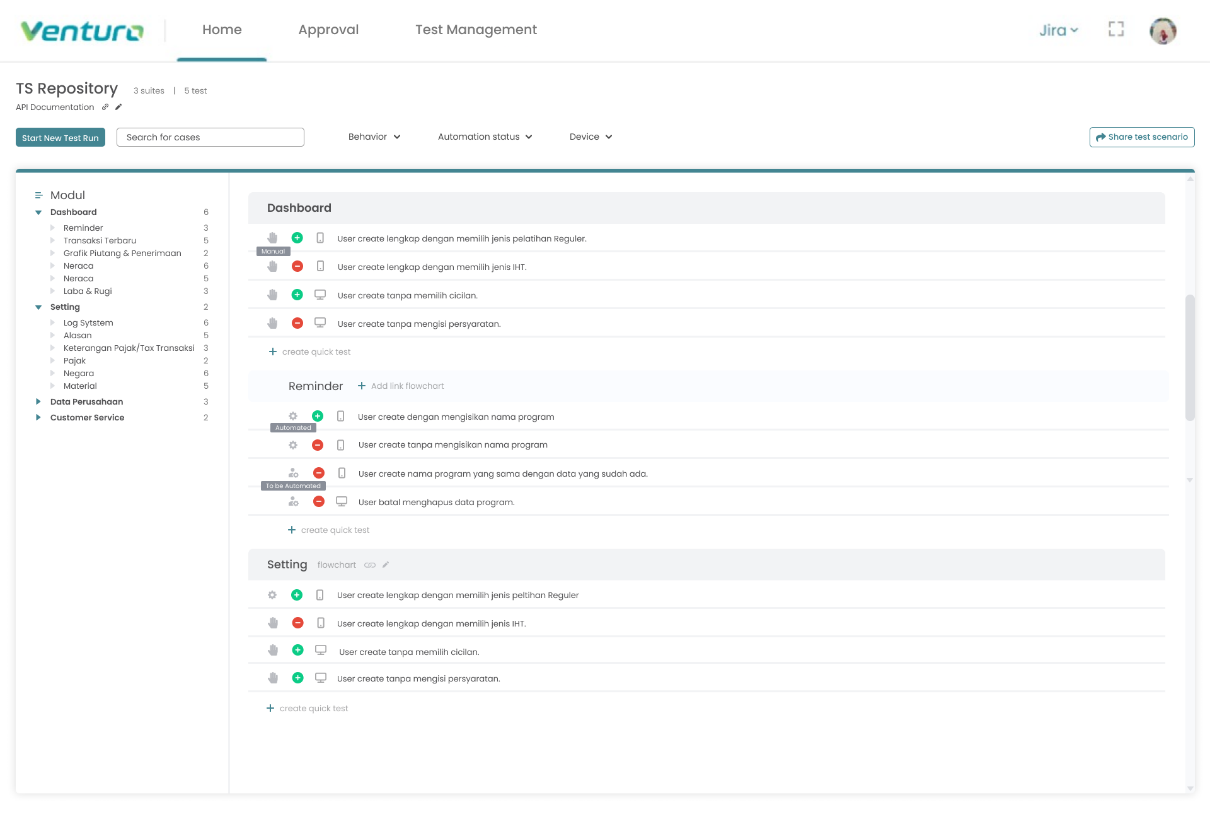
### Perancangan *User Interface* / *Mock-up* aplikasi

Pada bagian ini terdapat beberapa *mockup* dari tampilan yang akan dibuat nanti. *Mockup* dibuat menggunakan aplikasi Figma. Berikut merupakan *link* Figma dari *mockup* yang akan digunakan :

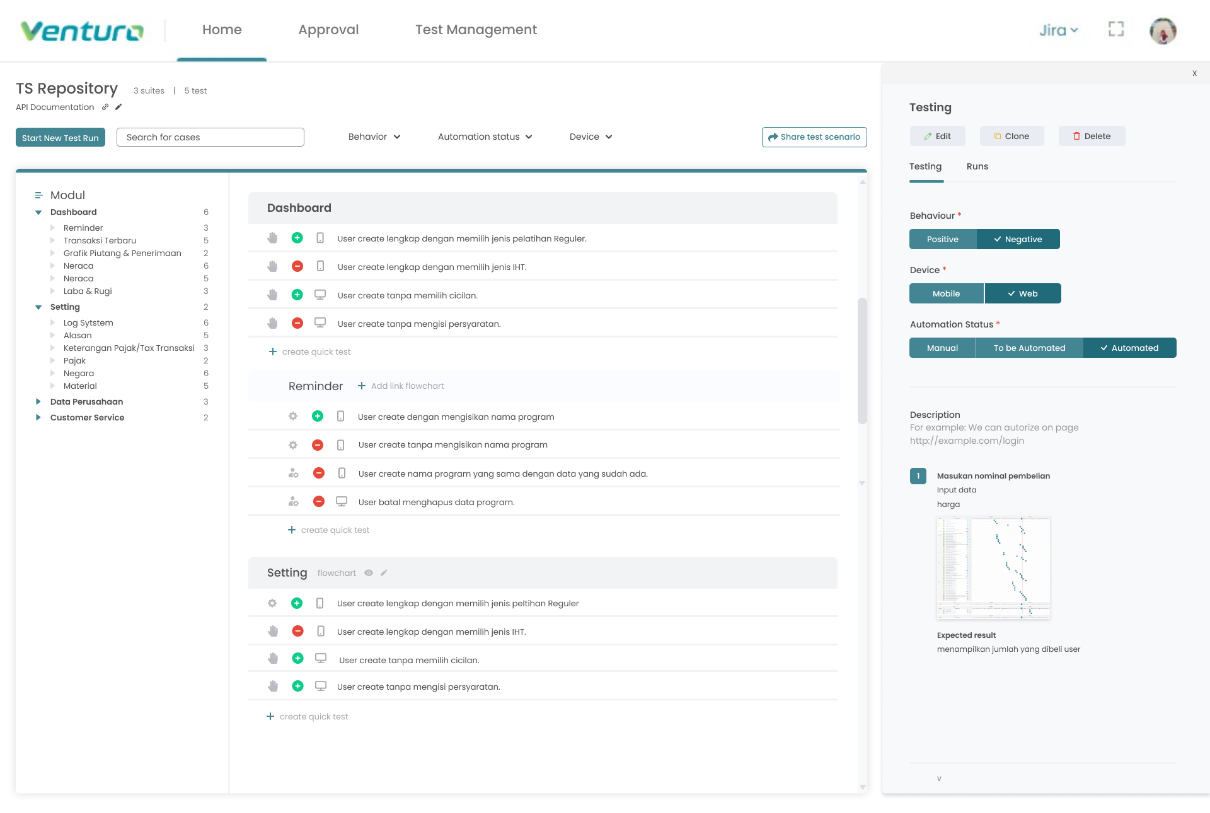
<https://www.figma.com/file/1vbj5KLAEYYByQ16AjxKBB/Test-Management-System?node-id=206%3A10781&t=2OcMMBYTmrr0cOzd-1>



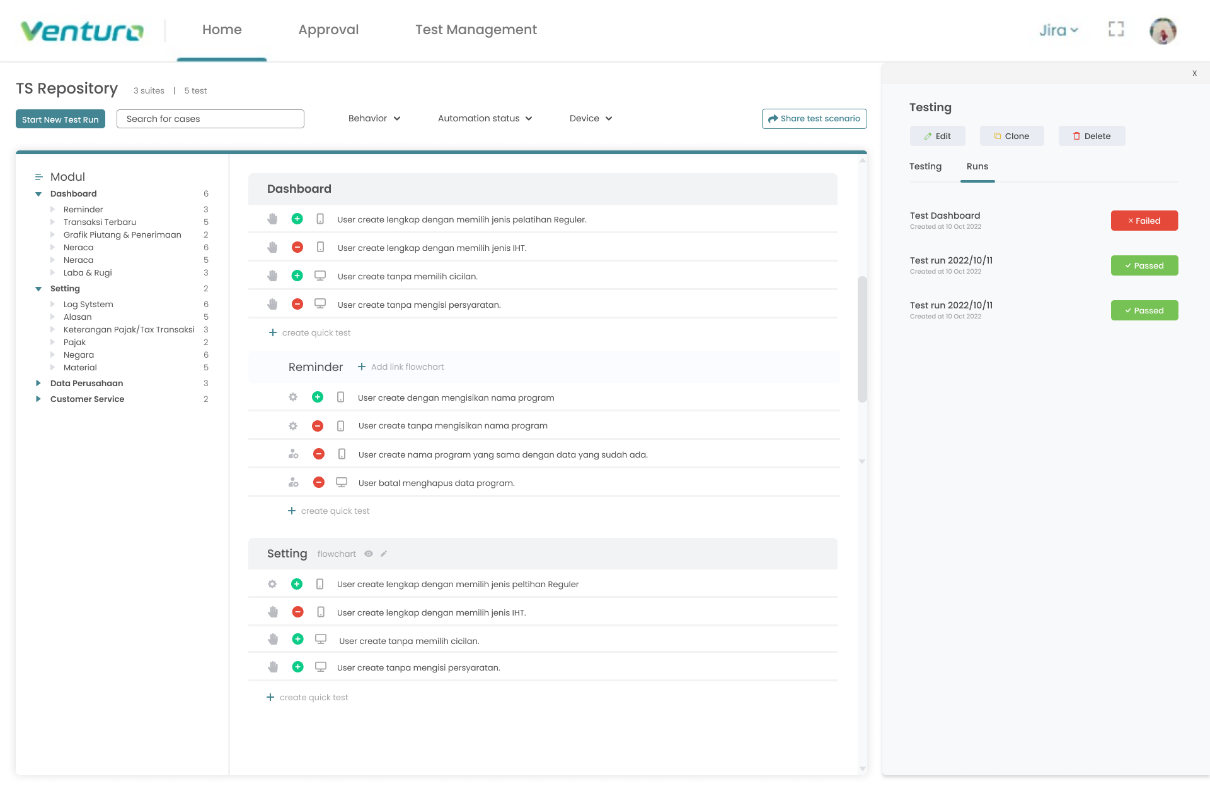
#### **Test Scenario**

Tampilan aplikasi merupakan interaksi antara pengguna dengan sistem. Pada gambar tampilan halaman *Test Scenario*, pengguna akan dihadapkan dengan *list* dari *Test Scenario* yang sudah ditambahkan sebelumnya. Disisi kiri ditampilan *list* modul/fitur yang akan ditambahkan *Test Scenario* nantinya. Disini pengguna dapat menambahkan *Test Scenario* sesuai dengan modul yang diinginkan.

Gambar 3. 14 Menu Test Scenario

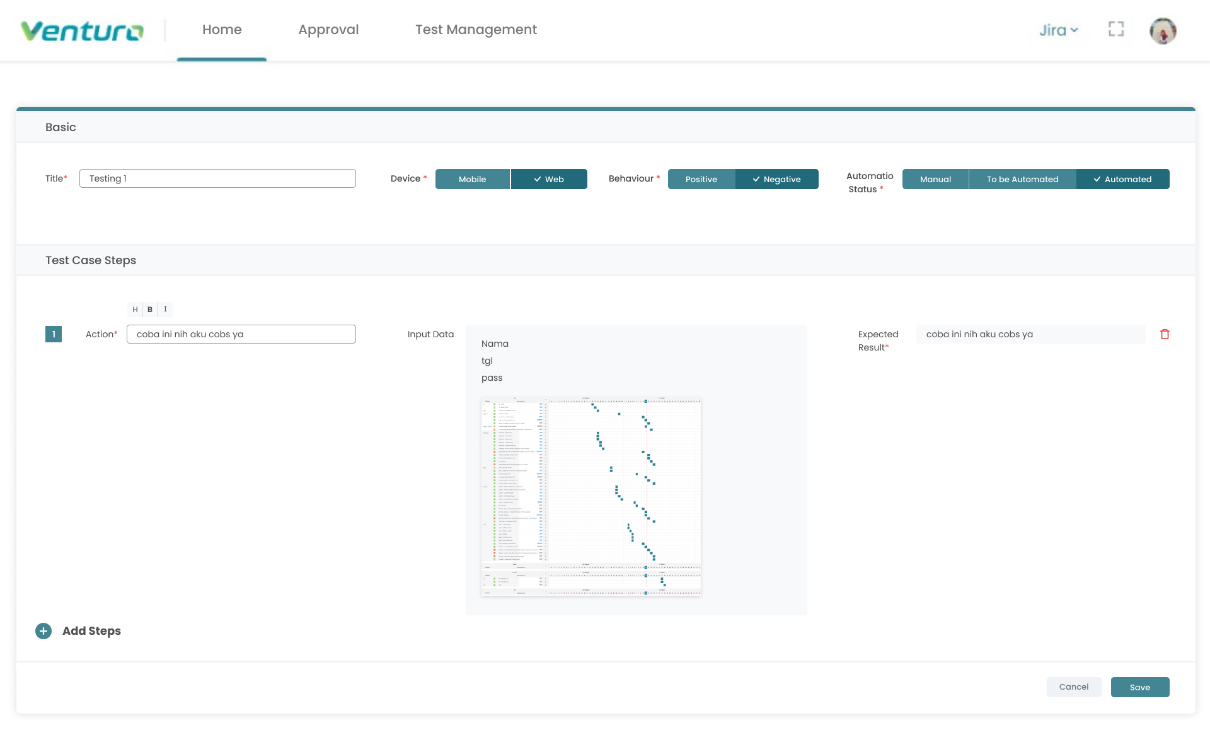
Apabila pengguna menekan salah satu *Test Scenario* maka akan ditampilkan detail dari *scenario* tersebut dalam bentuk sidebar. *Sidebar* ini menampilkan beberapa informasi seperti *behaviour, device, automation status, description*, dan *step – step* *test scenario* yang akan dijalankan.   


Gambar 3. 15 Detail Test Scenario (Testing)

Lalu, Apabila pengguna menekan *‘Runs’* di *sidebar* kanan, maka akan menampilkan detail *Runs* berupa *sidebar* disisi kanan. *Sidebar* ini menampilkan informasi terkait hasil *Run* tiap langkah yang ada didalam tiap *Scenario*.  


Gambar 3. 16 Detail Test Scenario (Run)

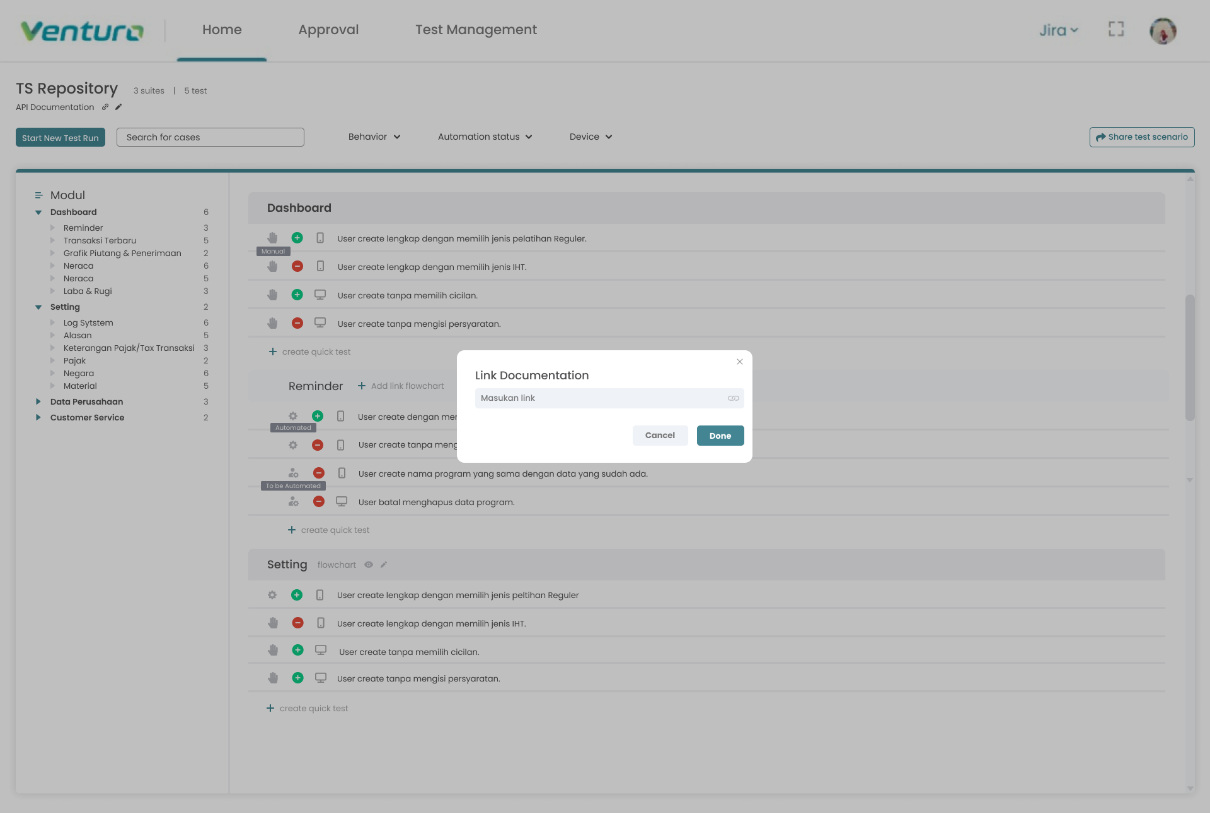
Kemudian Apabila pengguna menekan tombol *‘Edit’* maka akan menampilkan detail dan kendali untuk mengubah informasi terkait *Scenario*. Halaman menampilkan beberapa informasi seperti *behaviour, device, automation status, description*, dan *step – step test scenario* yang akan dijalankan.



Gambar 3. 17 Edit Test Scenario

#### **Add Link Documentation**

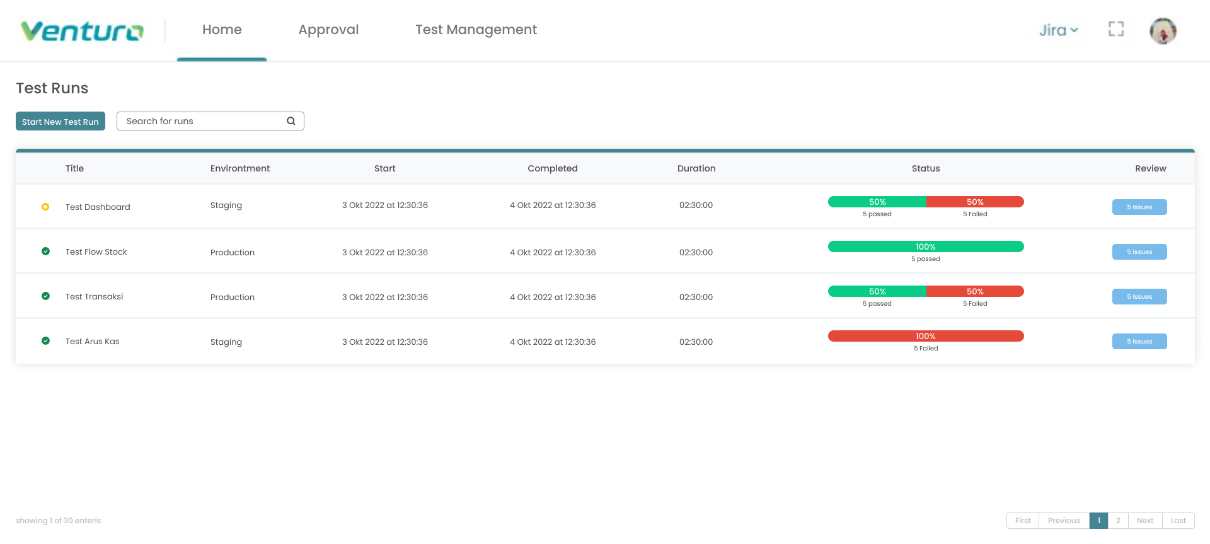
Apabila pengguna menekan tombol *“Add Link Documentation”* maka akan muncul *popup input link* fitur ini berguna yang sudah memasukan *link* dokumentasi dari *Test Run* tersebut. biasanya *link* *berisikan flow, mockup*, dan desain yang akan digunakan pada pengembangan aplikasi kedepannya.



Gambar 3. 18 Add Link Documentation

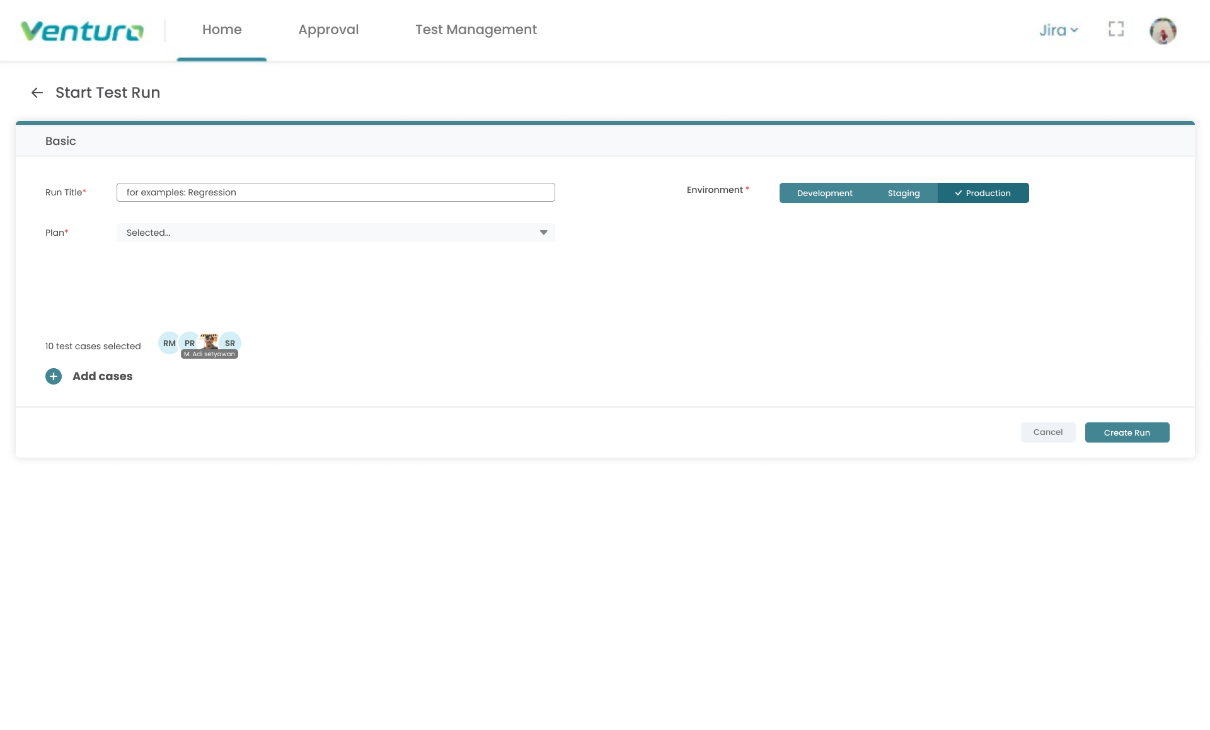
#### **Test Run**

Pada gambar tampilan halaman *Test Run*, pengguna akan dihadapkan dengan *list* dari *test runs* yang sudah ditambahkan sebelumnya. Apabila pengguna menekan salah satu *test runs* maka nantinya akan muncul halamana baru berisi detail dari *Test Run* yang baru saja dipilih.



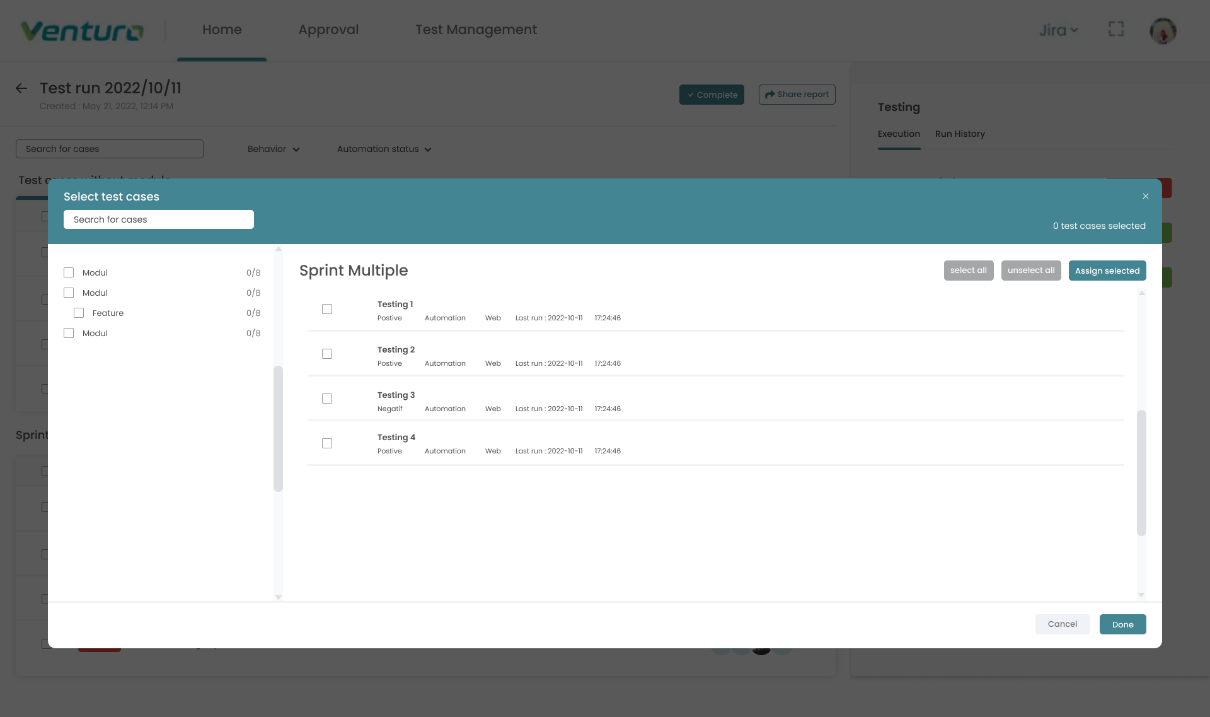
Gambar 3. 19 Menu Test Run

Apabila pengguna menekan tombol “*Start New Test Run”* maka akan muncul *form* baru tambah *Test Run*. Disini pengguna dapat membuat *test run* baru dengan mengisi beberapa informasi seperti nama *run, environment*, dan *test cases*.

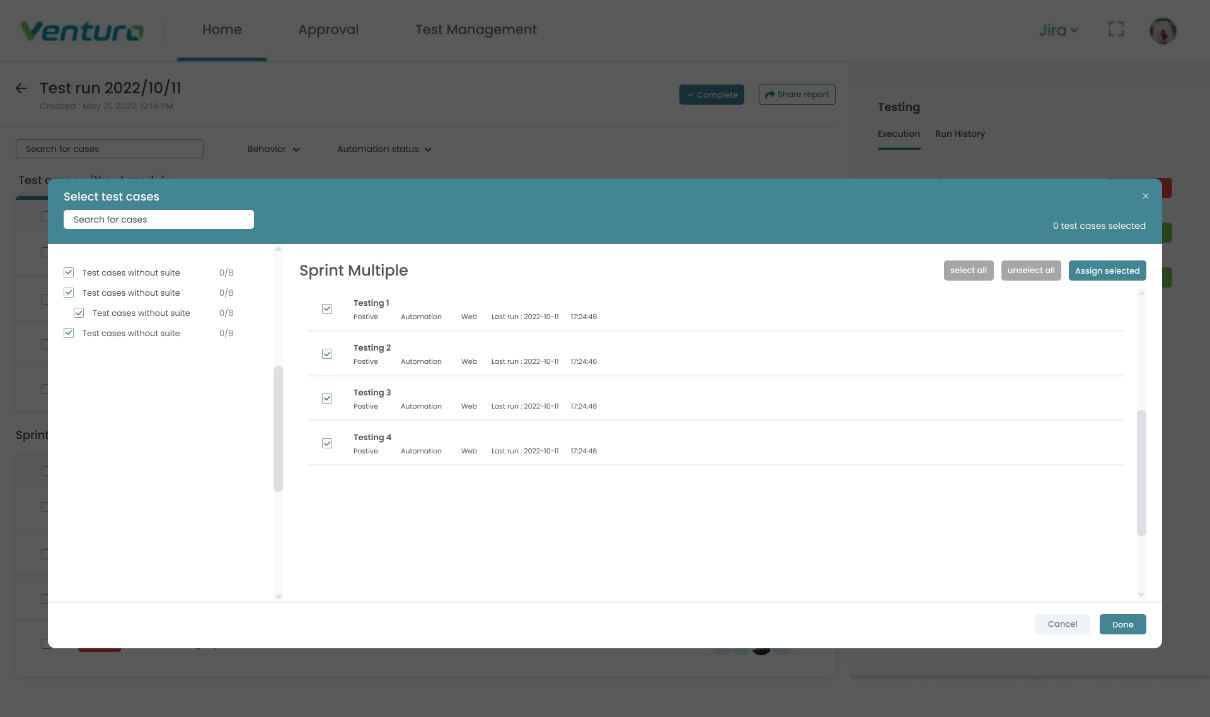


Gambar 3. 20 Create Test Run

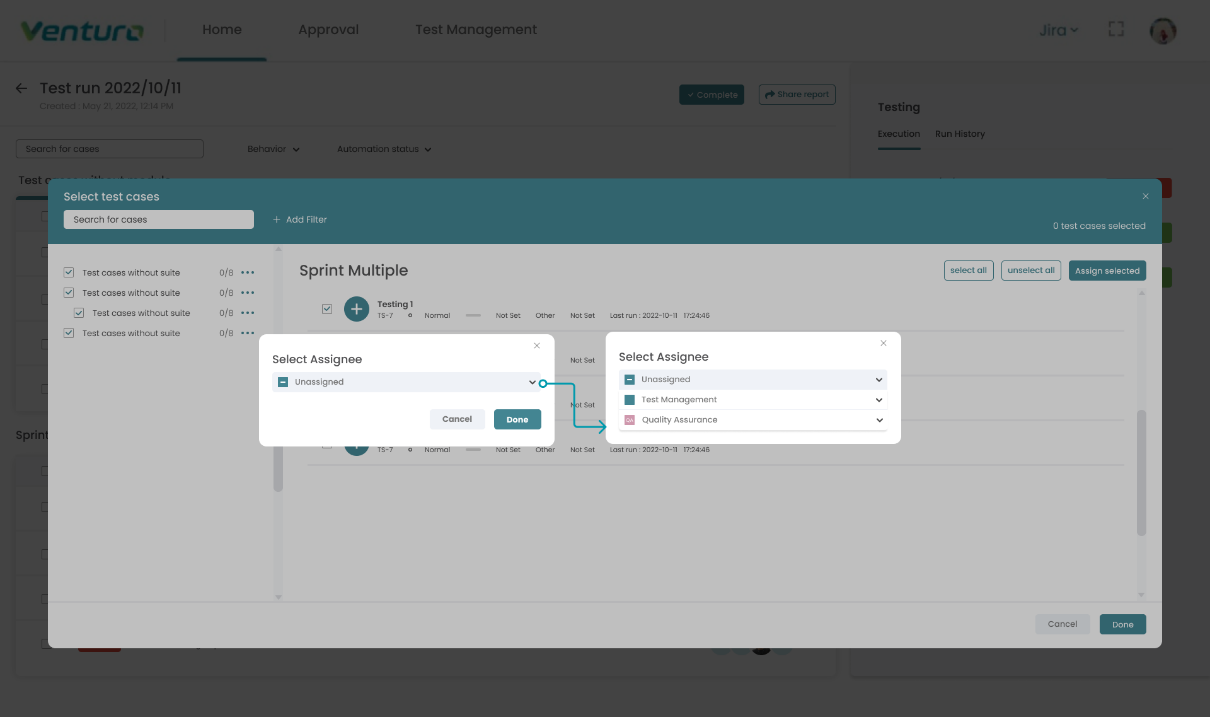
Apabila pengguna menekan *“Add Cases”* maka akan muncul *popup* *list test scenario* yang sudah dibuat sebelumnya. Pada halaman ini, pengguna dapat memilih *test scenario* yang akan dijalankan pada test run kali ini. Nantinya, user dapat memilih *QA Engineer* mana yang bertugas untuk menjalankan *testing* tersebut.



Gambar 3. 21 Create Run (Add Scenario Empty)



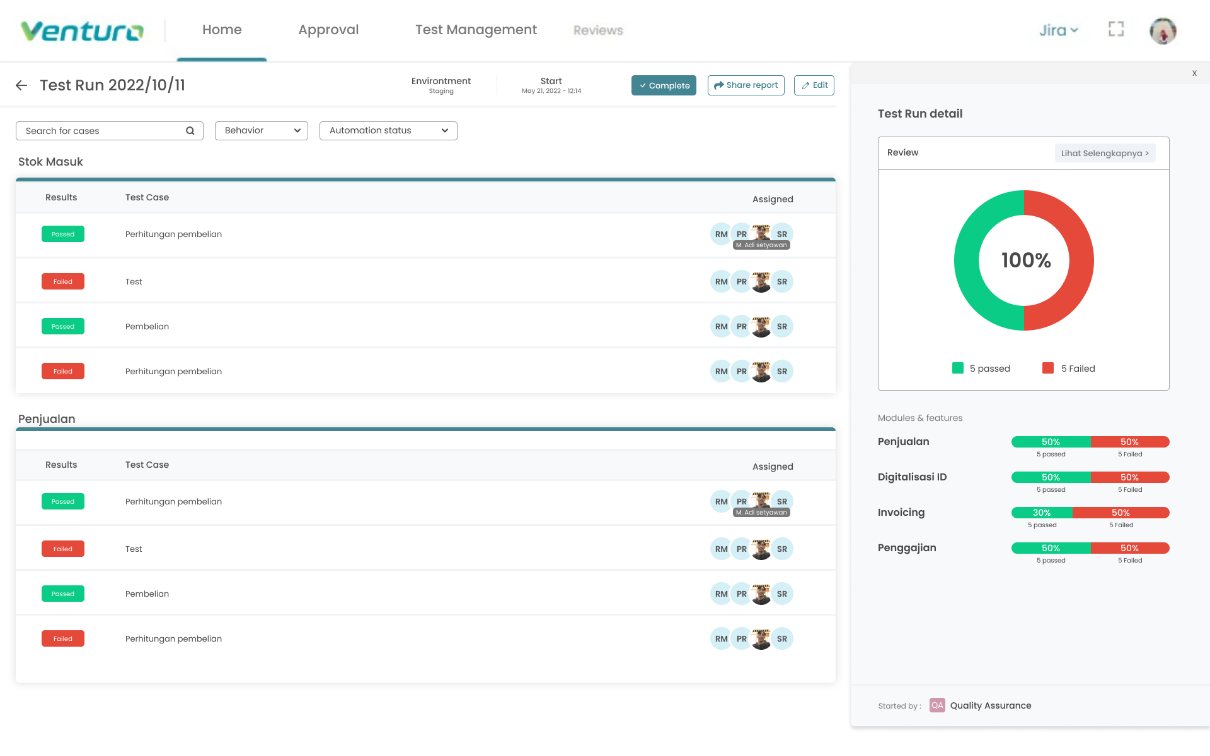
Gambar 3. 22 Create Run (Add Scenario)



Gambar 3. 23 Create Run (Add Assignee)

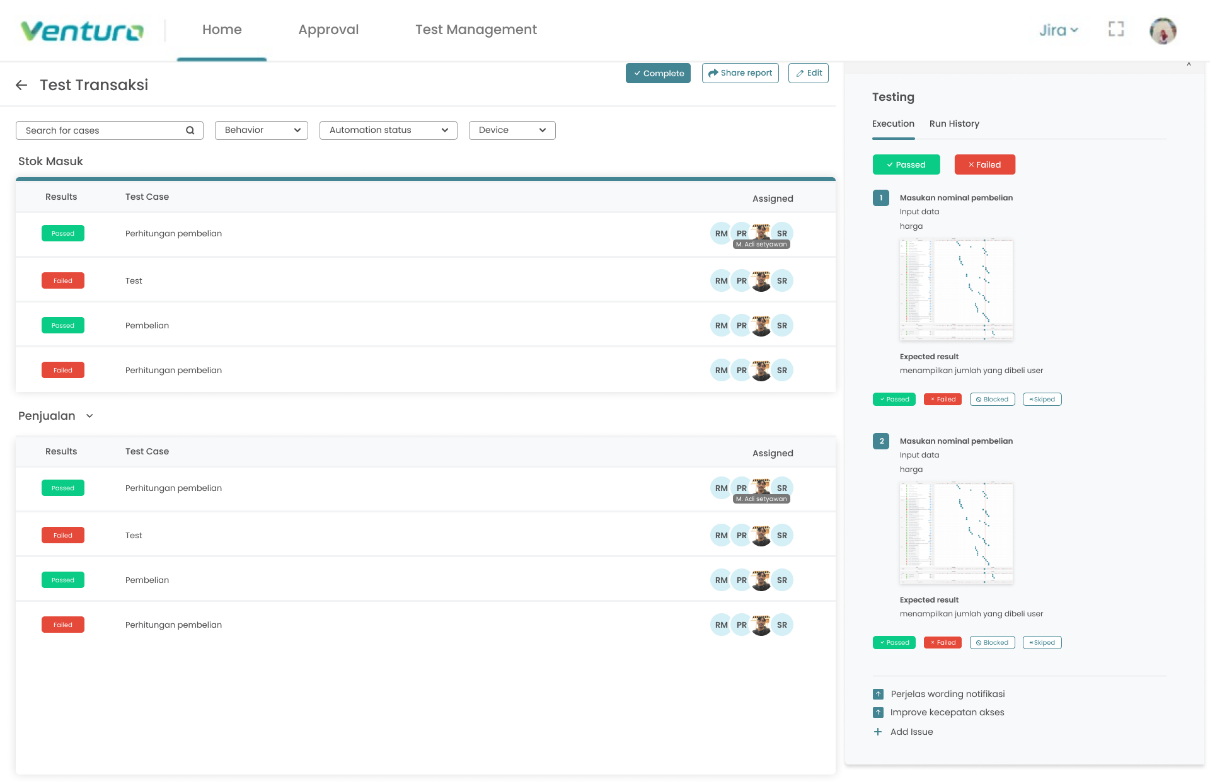
#### **Detail Test Run**

Pada halaman ini, terdapat informasi detail dari *test run* yang sudah dipilih di halaman *Test Run*. Disini pengguna dapat melihat informasi terkait *test run* berdasarkan *test scenario* yang sudah dibuat tadi. Pada sisi kanan, terdapat *sidebar* informasi terkait presentasi dari *test run* yang sudah dijalankan.



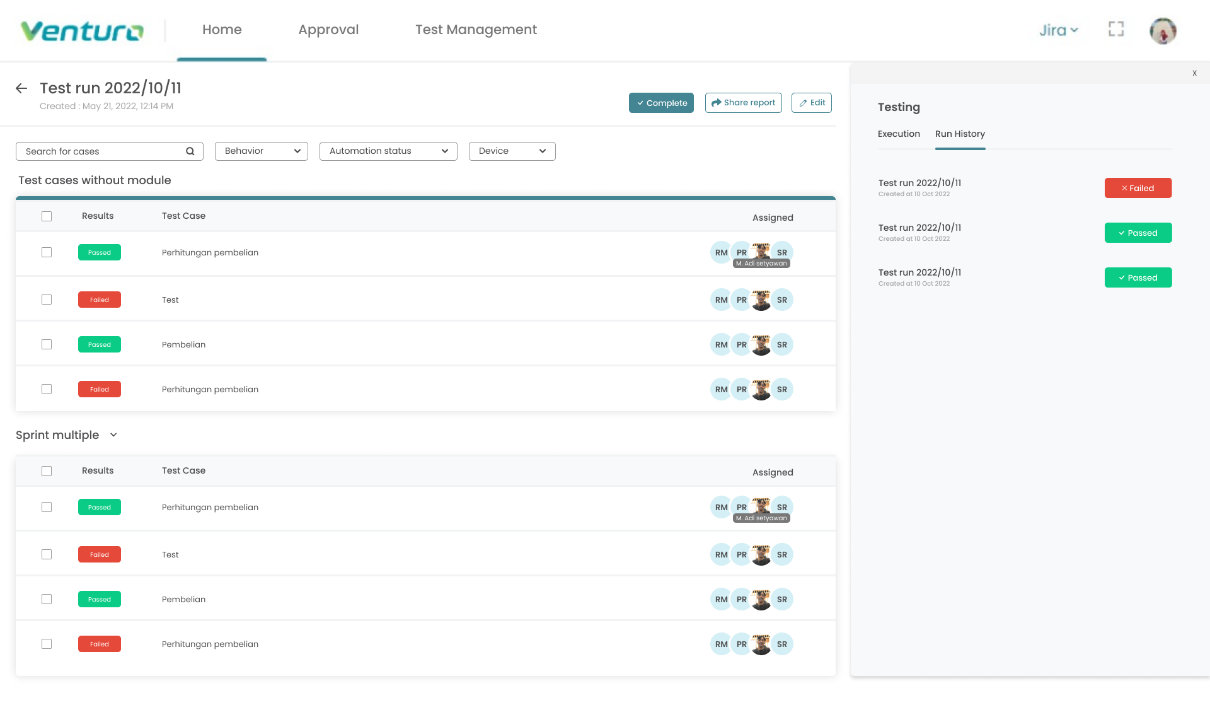
Gambar 3. 24 Detail Test Run

Apabila pengguna menekan salah satu dari *cases* maka akan menampilkan detail dari *cases* tersebut di *sidebar* kanan. Pada halaman “*Detail Execution”*, pengguna akan melihat *step – step* dari *cases* yang akan dijalankan serta memberikan informasi terkait hasil dari *case* tersebut *step by step*.



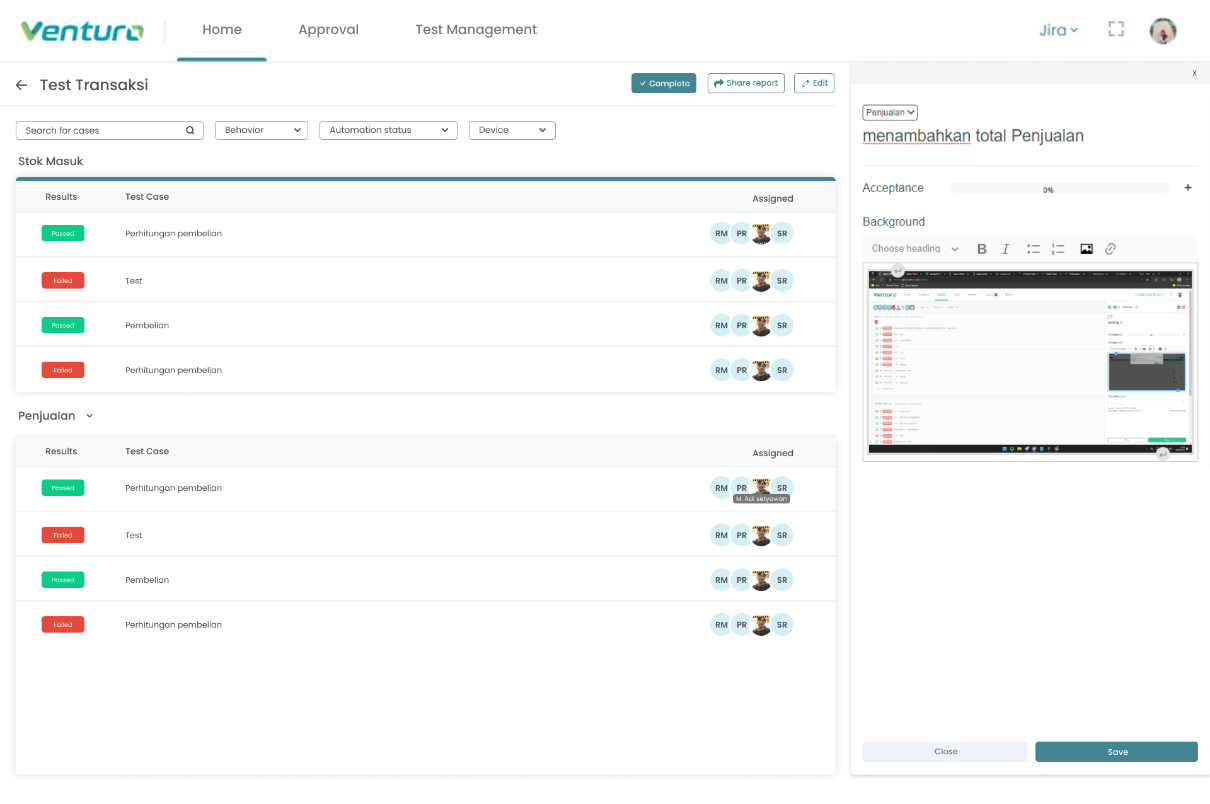
Gambar 3. 25 Detail Test Run (Detail Scenario Execution)

Lalu, ada *Detail Run History*. Pada halaman ini, pengguna dapat melihat riwayat dari *cases* yang sudah dijalankan sebelumnya.



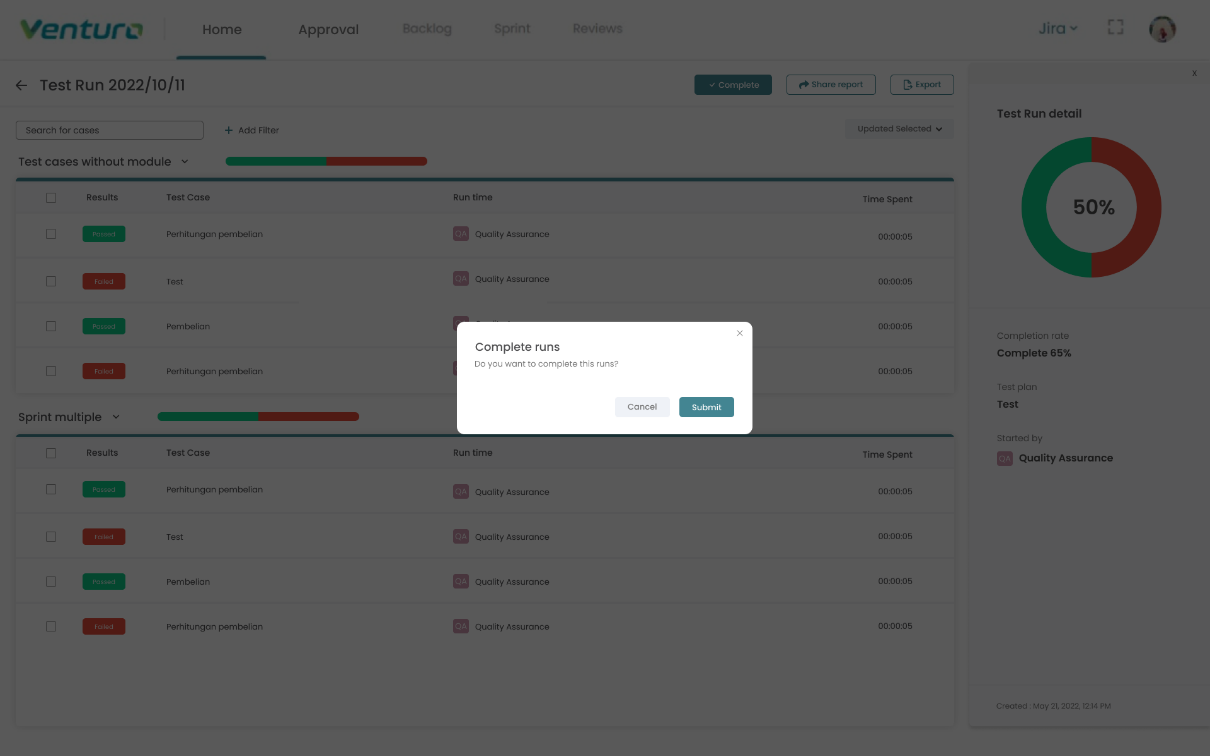
Gambar 3. 26 Detail Test Run (Detail Scenario Run History)

Disaat masuk menu *Execution* dan pengguna menemukan salah satu langkah gagal, maka pengguna akan pindah ke halaman *“Add Issue”* dan diharuskan menambahkan informasi terkait masalah apa yang dapat menyebabkan hasil langkah tersebut gagal.

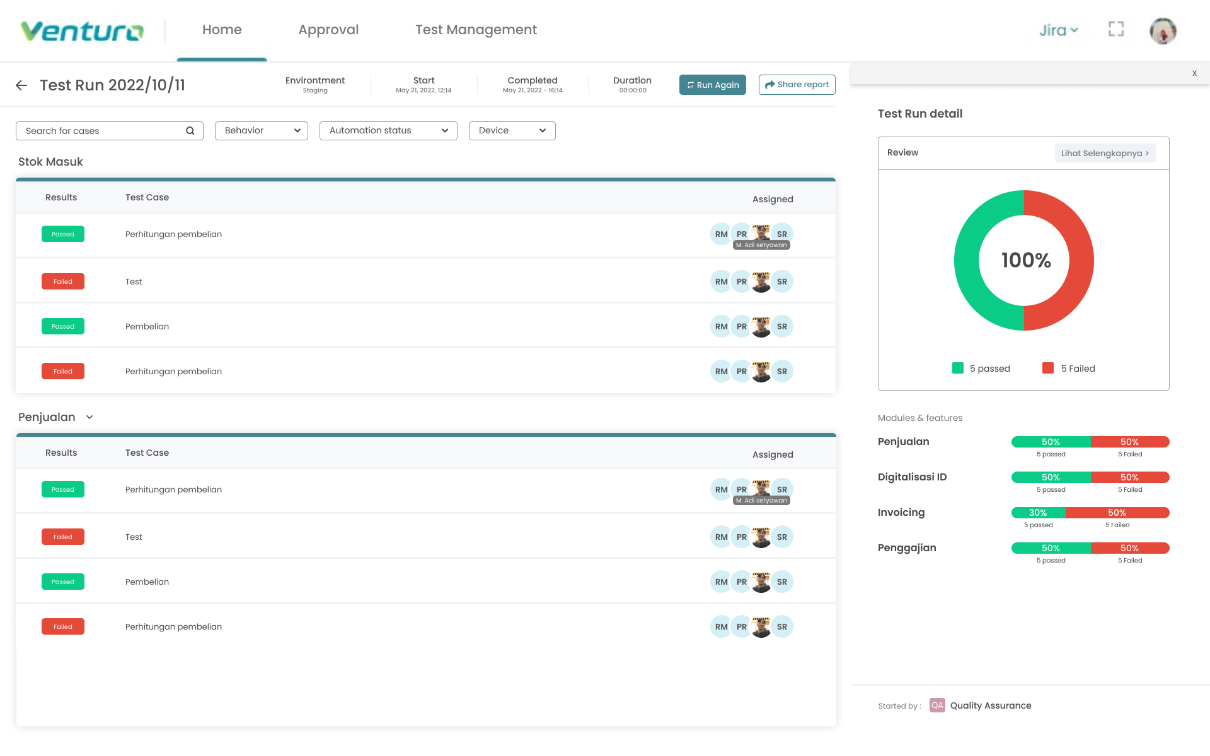


Gambar 3. 27 Detail Test Run (Add Issue)

Apabila pengguna merasa *run* tersebut sudah dijalankan semua, maka pengguna dapat menekan tombol *“Completed Run”* untuk menyatakan bahwa run tersebut sudah selesai.



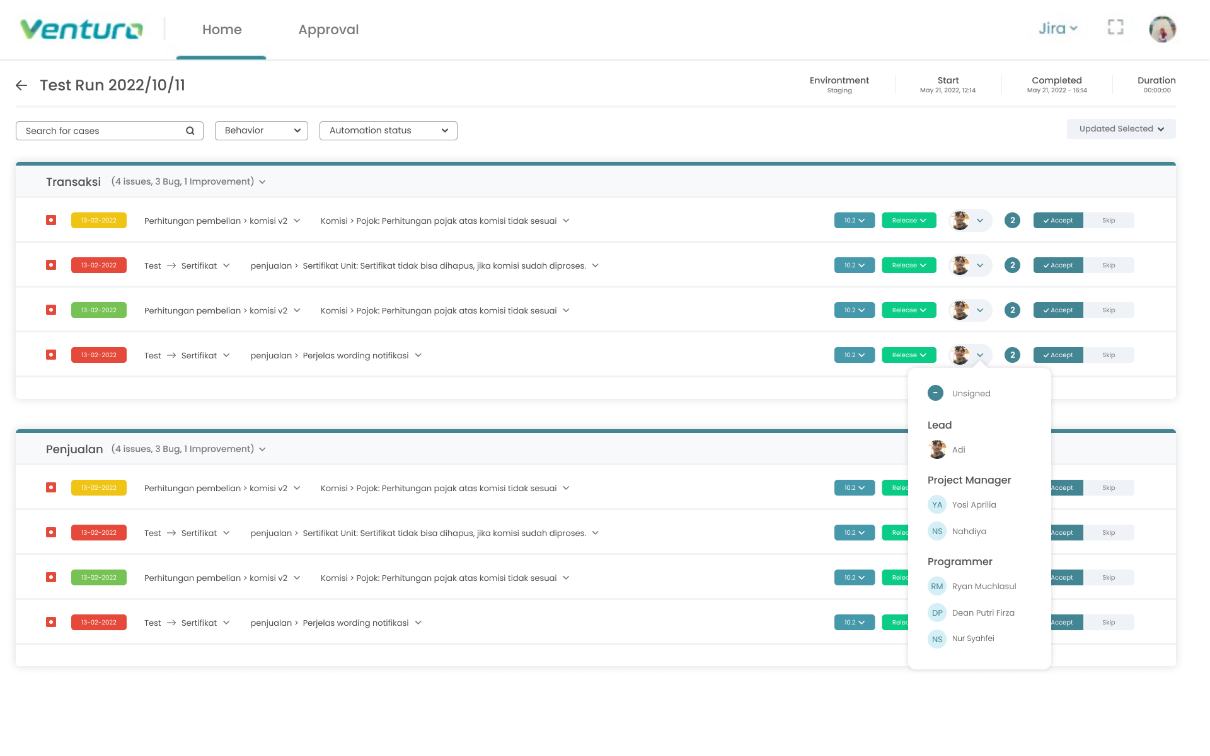
Gambar 3. 28 Alert Complete Run



Gambar 3. 29 Completed Run

#### **Reviews Issues**

Pada halaman ini, pengguna dapat menampilkan masalah yang dihadapi ketika menjalankan *Run* kali ini. Disini pengguna juga dapat mengganti status terkait bagaimana masalah tersebut ditangani.



Gambar 3. 30 Review Issue

## Rancangan Pengujian

Untuk penelitian ini, metodologi pengujian yang digunakan dalam mengembangkan aplikasi ini adalah pengujian *black box*. Pengujian *black box* atau pengujian fungsional adalah pengujian yang dijalankan hanya dengan menggunakan data pengujian dan mengamati hasil eksekusi untuk *verifikasi* fungsionalitas aplikasi dan apakah sistem bekerja dengan baik.

*Testing* ini dirancang untuk menguji sistem dari sisi pengguna. Pengujian ini juga berlaku untuk tampilan *user interface* sebagai sarana untuk membuat aplikasi mudah dipahami dan digunakan dari sudut pandang pengguna. Pengujian *black box* sangat membantu peneliti ketika pengguna menggunakan suatu aplikasi.

Dari pernyataan yang sudah dijabarkan diatas, peneliti memutuskan untuk menggunakan pengujian *black box*, selain implementasinya yang mudah dan biaya teknis yang rendah, peneliti berharap aplikasi yang dibuat mampu dimanfaatkan dan digunakan secara optimal oleh pengguna.

Tabel 3. 7 Tabel Rancangan Pengujian

|  |  |  |  |
| --- | --- | --- | --- |
| **Fitur** | **Skenario** | **Kasus** | **Hasil yang diharapkan** |
| *Test Scenario* | Menambahkan *scenario* baru | Menambahkan judul skenario | Data *scenario* berhasil tersimpan |
| Menambahkan judul kosong | Muncul peringatan bahwa judul masih kosong |
| Ubah *scenario* | Merubah atrribut didalam skenario seperti *Automation, Behaviour*, dan *Device.* | Attribut berhasil diubah. |
| Merubah atrribut dengan data kosong | Muncul peringatan bahwa ada attribut yang masih kosong |
| Menambah *step* baru pada skenario | Data *step* berhasil ditambahkan |
| Menghapus *step* pada skenario | Data *step* berhasil dihapus |
| Menghapus skenario | Menghapus skenario dengan *soft delete*. | Data skenario tidak tampil lagi walapun data masih ada di *database*. |
| Duplikat skenario | Membuat data skenario baru berdasarkan data skenario lain. | Data skenario berhasil diduplikat. |
| *Test Run* | Menambahkan run baru | Membuat *run* tanpa atrribut lengkap (Judul,  *Environment*, dll) | Gagal membuat skenario |
| Memilih skenario tanpa *assigned* | Gagal memilih skenario |
| Memilih skenario dengan *assigned* | Berhasil menmilih skenario |
| Membuat *run* dengan atrribut lengkap (Judul,  *Environment,* dll) | Data skenario berhasil tersimpan |
| Menghapus run | Menghapus *run* dengan *soft delete.* | Data *run* tidak tampil lagi walapun data masih ada di database. |
| Duplikat *run* | Duplikat *run* yang belum *complete* | Gagal dan muncul peringatan bahwa *run belum complete.* |
| Duplikat *run* yang sudah complete | Data *run* berhasil diduplikat. |
| Ubah *run* | Mengubah judul *run* dengan data kosong | Gagal dan peringatan judul harus diisi |
| Menggubah *assignee* pada test scenario dengan data kosng | Gagal dan peringatan *assignee* harus diisi |
| Menggubah *assignee* pada test skenario | Berhasil menggubah *assignee* |
| Menambah *test scenario* | Berhasil menambah *test scenario* |
| Menghapus *test scenario* | Berhasil menghapus *test scenario* |
| Ubah *detail run* | Ubah status *step* pada skenario | Berhasil menggubah status *step* |
| Menambah *issue step* pada skenario | Berhasil menambahkan *issue step* |
| Ubah status *run* menjadi *complete* | Berhasil menggubah status *run* |
| Review Issue | Ubah issue | Ubah *status issue* pada *step skenario* | Berhasil menggubah status *issue* |