### Nama : Muhammad Amin Iqbaal Alam

### Posisi : QA\_Engineer

**Nomor 1**

### Test Scenario

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Scenario ID:** | TS001 | **Process ID:** | Faktorial001 | |
| **Environment:** | Functional Factorial | **Machine tested:** | ASUS GL503GE | |
| **Test Scenario Description:** | Pengujian Fungsional Faktorial | | | |
| **Objective:** | Mengetahui fungsionalitas faktorial agar dapat berjalan sesuai kebutuhan | | | |
| **Assumptions/ Constraints:** | Input yang dimasukkan berisi number | | | |
| **Author:** | Iqbaal | **Last Modified:** | 18 Maret 2023 | |
| **Reviewed by:** | Iqbaal | **Reviewed Date:** | 18 Maret 2023 | |
| **Executed by:** | Iqbaal | **Execution Date:** | 18 Maret 2023 | |
|  | | | | |
| **Test Steps:** | | | | |
| **Step #** | **Description** | **Expected Result** | | **Actual Result** |
|  | Aktor mengisi form kalkulator faktorial dengan *input ‘Masukkan Angka’ : ‘1’* | Faktorial dari 1 adalah: 1 | | Faktorial dari 1 adalah: 1 |
|  | Aktor mengisi form kalkulator faktorial dengan *input ‘M*asukkan Angka’ : ‘1,5’ | Please enter an integer | | Please enter an integer |
|  | Aktor mengisi form kalkulator faktorial dengan input ‘Masukkan Angka’ : ‘30’ | Faktorial dari 30 adalah: 265252859812191058636308480000000 | | Faktorial dari 30 adalah: 2.652528598121911e+32 |

### Test Case *Factorial*

Tabel 6.48 merupakan hasil pengujian *factorial*.

Tabel 1 Test Case *Factorial*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kode Kebutuhan | | F-001 | | |
| *Test case* Number | | TC-1 | | |
| *Test case* *Name* | | *Factorial* | | |
| No. | Prosedur Uji | Input | *Expected result* | Status |
| 1. | Aktor mengisi form kalkulator faktorial dengan *input ‘M*asukkan Angka’ | *‘Masukkan Angka’ : ‘1’* | Faktorial dari 1 adalah: 1 | *Passed* |
| 2. | Aktor mengisi form kalkulator faktorial dengan *input ‘M*asukkan Angka’ | *‘Masukkan Angka’ : ‘1,5’* | Please enter an integer | *Passed* |
| 3. | Aktor mengisi form kalkulator faktorial dengan *input ‘M*asukkan Angka’ | *‘Masukkan Angka’ : ‘30’* | Faktorial dari 30 adalah: 2.652528598121911e+32 | *Passed* |

### Bug System

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama Bug | Result | Expected Result |
| 1. | Tidak bisa input selain integer, namun tidak ada perintah agar input dengan tipe data integer | Please enter an integer | - |
|  | Sistem tidak dapat menampilkan hasil yang sesuai dengan perhitungan manual, seperti pada 30 faktorial | Faktorial dari 30 adalah: 2.652528598121911e+32 | Faktorial dari 30 adalah: 265252859812191058636308480000000 |

### Saran Perbaikan

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Bug | Bugfixed | Type |
| 1. | Tidak bisa input selain integer, namun tidak ada perintah agar input dengan tipe data integer | Menambahkan perintah agar user menginputkan tipe data integer (sesuai yang diinginkan) agar tidak terjadi Bug pada system | Mayor |
|  | Sistem tidak dapat menampilkan hasil yang sesuai dengan perhitungan manual, seperti pada 30 faktorial | Menambahkan hasil perhitungan yang lebih akurat | Mayor |

**Nomor 2**

Link Github : *<https://github.com/Aminiqbaal/QA_Sevima.git>*

**Nomor 3**

1. $i <= count($balls) - 1 bisa diubah $i < count($balls) - 1

lebih efisien karena ketika $i = 7 tidak perlu masuk ke increment $i++, sehingga tidak perlu ada if untuk mengecek $i < 8

1. Dengan $i < count($balls) - 1

kita hanya perlu 7x looping, sedangkan $i <= count($balls) - 1 kita masih butuh looping 8x

1. break ketika timbangan tidak sama

for ($i = 0; $i < count($balls) - 1; $i++) {

if ($balls[$i] != $balls[$i + 1]) {

break;

}

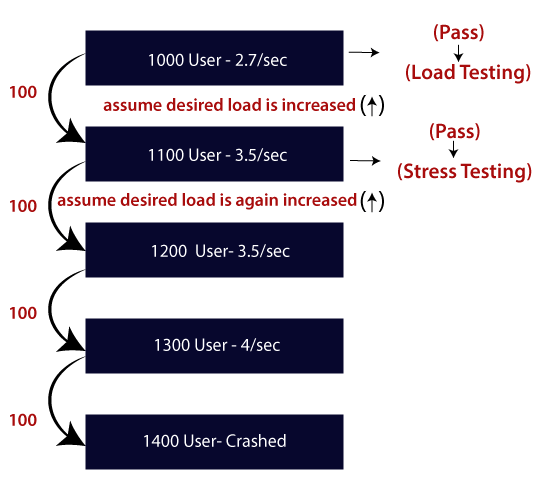
$timbang++;

}

**Nomor 4**

Menggunakan Performance Testing. Bisa kita ambil satu contoh di mana kita akan menguji perilaku aplikasi di mana beban yang diinginkan kurang dari 1000 atau sama dengan 1000 pengguna.

* Skenario1: Ketika kami memiliki 1000 pengguna sebagai beban yang diinginkan, dan 2,7/detik adalah waktu sasaran, skenario ini akan lolos saat melakukan uji beban karena dalam pengujian beban, kami akan berkonsentrasi pada no. pengguna, dan sesuai kebutuhan sama dengan 1000 pengguna.
* Skenario2: Dalam skenario berikutnya, kami akan meningkatkan beban yang diinginkan sebanyak 100 pengguna, dan waktu sasaran akan naik menjadi 3,5\dtk. Skenario ini akan lolos jika kita melakukan stress testing karena disini beban aktual lebih besar dari (1100) beban yang diinginkan (1000).
* Skenario3: Dalam hal ini, jika kita meningkatkan beban yang diinginkan tiga kali lipat  
  **1200** → 3,5\detik: [tidak kurang dari atau sama dengan beban yang diinginkan karena itu akan **Gagal**]  
  **1300** → 4\detik: [tidak kurang dari atau sama dengan beban yang diinginkan. yaitu, **Gagal**]  
  **1400** → **Hancur**



**Nomor 5**

Link

https://intip.in/nomor5