**A. Basic Programming**

1. Type Data Operasi Aritmatika

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type Data of Operand |  |  |  |  |  |
|  | **SMALLINT** | **INTEGER** | **DECIMAL** | **SMALLFLT** | **FLOAT** |
| **SMALLINT** | INTEGER | INTEGER | DECIMAL | SMALLFLT | FLOAT |
| **INTEGER** | INTEGER | INTEGER | DECIMAL | FLOAT | FLOAT |
| **DECIMAL** | DECIMAL | DECIMAL | DECIMAL | FLOAT | FLOAT |
| **SMALLFLT** | SMALLFLT | FLOAT | FLOAT | SMALLFLT | FLOAT |
| **FLOAT** | FLOAT | FLOAT | FLOAT | FLOAT | FLOAT |

1. Type Data Word and Character

|  |
| --- |
| Type Data of Word and Character |
| **STRING** |
| **CHAR** |
| **CHARACTER** |

1. Output of code below

**int** a = 100 + 3 \* 5 - 10 / 5;

[System](http://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+system).out.println(a);

Adalah 113

1. Output of code below

**int** a = 0;

**for** (**int** i = 0; i < 6; ++i) {

a = a + 2;

}

[System](http://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+system).out.println(a/6);

Adalah 2

1. Output of code below

[String](http://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+string) name, address;

name = "Jack Donovan";

address = "3566 Golden Street Miami";

[System](http://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+system).out.println(address.replace(" ", ""));

Adalah 3566GoldenStreetMiami

**B. Flaky Test**

1. Flaky Test adalah tes yang lolos dan gagal secara berkala tanpa perubahan kode apa pun. Pengujian yang tidak stabil pasti akan mengganggu, tetapi juga bisa sangat mahal karena sering kali mengharuskan teknisi untuk melakukan Retrigger seluruh build di CI dan sering kali membuang banyak waktu menunggu build baru selesai dengan sukses.
2. Yang saya lakukan untuk menguji sebuah elemen dalam sebuah website menggunakan ajax / javascript untuk menampilkan data (diproses secara asynchronous) adalah

Ketika kita memeriksa kasus seperti menggunakan Selenium, kita harus mempertimbangkan fakta bahwa pemberitahuan "AJAX" tidak akan segera muncul di halaman dan yang terpenting kita tidak tahu kapan tepatnya pemberitahuan itu akan muncul. Dengan demikian, skrip pengujian di Selenium harus menunggu beberapa elemen berubah alih kehalaman baru. Di sisi lain, pengujian seharusnya tidak berlangsung lama. Kita dapat mengimplementasikan menunggu dengan Thread.sleep (),WebDriver implicit waits, dan explicit waits in Selenium-based.

1. Halaman situs web hanya dapat diakses dengan CAPTCHA. Bagaimana Anda menguji halaman itu?

Bagaimana mengkonfigurasi alat pengujian beban untuk membaca teks yang ditampilkan. Inti dari menambahkan keamanan captcha adalah untuk mencegah alat otomatis mengakses situs web, jadi jika itu dengan mudah dilewati oleh alat pengujian beban, maka pelaku spam juga dapat menggunakan teknik yang sama untuk mengakses situs web kita.

Ada tiga opsi yang bisa saya lihat:

* Nonaktifkan captcha sepenuhnya untuk tujuan pengujian
* Konfigurasikan perangkat lunak captcha untuk menerima apa pun yang diketikkan sebagai valid
* Konfigurasikan captcha untuk menampilkan nilai sebenarnya pada gambar di suatu tempat di halaman web yang dapat diurai oleh Load Tester. (Jelas ini harus dihapus sebelum pengiriman.)

**C. ESSAY QUESTION TOPIC**

1. – **Login**

WebDriver driver;

@Given("^open chrome and start application$")

**public** **void** open\_chrome\_and\_start\_application() **throws** Throwable {

System.*setProperty*("webdriver.chrome.driver", "C:\\ChromeDriver\\chromedriver.exe");

driver = **new** ChromeDriver();

driver.manage().window().maximize();

driver.get("https://www.bukalapak.com/");

}

@When("^I enter valid username and valid password$")

**public** **void** I\_enter\_valid\_username\_and\_valid\_password() **throws** Throwable {

driver.findElement(By.*id*("loginform-username")).sendKeys("ameliamanurung07@gmail.com");

driver.findElement(By.*id*("loginform-password")).sendKeys("Manurung-107"); Thread.*sleep*(3000);

}

@Then("^I can login successfully$")

**public** **void** I\_can\_login\_successfully() **throws** Throwable {

driver.findElement(By.*cssSelector*(".btn")).click();

}

* **Search Product**

@When("^I Enter valid item in SearchItem$")

**public** **void** I\_Enter\_valid\_item\_in\_SearchItem() **throws** Throwable {

driver.findElement(By.*cssSelector*("#container > header > div > div > div.\_2KctL > div > div > div.zYZEU > div > form > fieldset > div > input[type=text]")).sendKeys("Handphone");

Thread.*sleep*(3000);

}

@Then("^I can see objects by item$")

**public** **void** I\_can\_see\_objects\_by\_item() **throws** Throwable {

driver.findElement(By.*cssSelector*("#container > header > div > div > div.\_2KctL > div > div > div.\_3b3oR > span > svg")).click();

Thread.*sleep*(3000);

}

1. **Login**

@Given("^open chrome and start application$")

**public** **void** open\_chrome\_and\_start\_application() **throws** Throwable {

System.*setProperty*("webdriver.chrome.driver", "C:\\ChromeDriver\\chromedriver.exe");

driver = **new** ChromeDriver();

driver.manage().window().maximize();

driver.get("https://apkpure.com/sample-android-app-login-test/com.loginmodule.learning/download?from=details");

}

@When("^I enter valid username and valid password$")

**public** **void** I\_enter\_valid\_username\_and\_valid\_password() **throws** Throwable {

driver.findElement(By.*id*("loginform-username")).sendKeys("ameliamanurung07@gmail.com");

driver.findElement(By.*id*("loginform-password")).sendKeys("Manurung-107"); Thread.*sleep*(3000);

}

@When("^I enter invalid username$")

**public** **void** I\_enter\_valid\_username() **throws** Throwable {

driver.findElement(By.*cssSelector*("#container > header > div > div > div.\_14lZ9.\_110yh > button > span")).click();

Thread.*sleep*(3000);

driver.findElement(By.*xpath*("/html/body/div[2]/div/div/div/button[2]/span")).click();

Thread.*sleep*(3000);

driver.findElement(By.*name*("loginform-username")).sendKeys("amelia");

Thread.*sleep*(3000);

}

@When("^I enter valid username and invalid password$")

**public** **void** I\_enter\_valid\_username\_and\_invalid\_password() **throws** Throwable {

driver.findElement(By.*cssSelector*("#container > header > div > div > div.\_14lZ9.\_110yh > button > span")).click();

Thread.*sleep*(3000);

driver.findElement(By.*xpath*("/html/body/div[2]/div/div/div/button[2]/span")).click();

Thread.*sleep*(3000);

driver.findElement(By.*name*("loginform-username")).sendKeys("ameliamanurung07@gmail.com");

Thread.*sleep*(3000);

driver.findElement(By.*xpath*("/html/body/div[2]/div/div/form/div/button/span")).click();

Thread.*sleep*(3000);

driver.findElement(By.*name*("password")).sendKeys("ppmpl");

Thread.*sleep*(3000);

}

@Then("^I can login successfully$")

**public** **void** I\_can\_login\_successfully() **throws** Throwable {

driver.findElement(By.*xpath*("/html/body/div[2]/div/div/form/div/button/span")).click();

Thread.*sleep*(3000);

}

@Then("^I cant login successfully$")

**public** **void** I\_cant\_login\_successfully() **throws** Throwable {

driver.findElement(By.*xpath*("/html/body/div[2]/div/div/form/div/button/span")).click();

Thread.*sleep*(3000);

}

@Then("^I cant login successfully$")

**public** **void** I\_cant\_login\_successfully() **throws** Throwable {

driver.findElement(By.*xpath*("/html/body/div[2]/div/div/form/div/button/span")).click();

Thread.*sleep*(3000);

}