

BDD

PPL PRAK

SOFTWARE TESTING

Pengujian perangkat lunak sesuai dengan BDD menggunakan format Behat Cheat Sheet.
Pengujian menggunakan selenium webdriver.

Dr. Indra Kharisma Raharjana, S.Kom., M.T.

INSTALL SELENIUM & CHROME DRIVER

INSTALL SELENIUM

Pada command prompt masuk ke folder project lalu install selenium menggunakan command berikut :

pip install selenium

INSTALL CHROME DRIVER

download chrome driver sesuai dengan versi chrome yang telah terinstall di komputer. Berikut link untuk mendownload chrome driver :

<https://chromedriver.chromium.org/>

USER STORY & USER STORY SCENARIO

Untuk menguji apakah aplikasi dapat berjalan sesuai dengan kebutuhan maka dapat dilakukan penyusunan user story scenario. Semua skenario yang mungkin dari jalannya aplikasi dilakukan pengujian. Format untuk menulis user story scenario mengikuti format penulisan **Behat Cheat Sheet** berikut ini.

Web apps (Mink)	
Surf:	Assertions:
I am on "url"	I should see "content"
I go to "url"	the response should contain "content"
I reload the page	I should not see "content"
I move backward one page	the response should not contain "content"
I move forward one page	the "form_element" field should contain "value"
I press "button"	the "form_element" field should not contain "value"
I follow "link"	the "form_checkbox" checkbox should be checked
	the "form_checkbox" checkbox should not be checked
Forms:	I should be on "page"
I fill in "form_element" with "value"	the url should match "url"
I fill in "value" for "form_element"	the "num_position" element should contain "value"
I fill in the following	I should see "value" in the "element" element
I select "form_option" from "form_select"	I should see an "element" element
I additionally select "form_option" from "form_select"	I should not see an "element" element
I check "form_checkbox"	I should see number "element" elements
I uncheck "form_checkbox"	the response status code should be code
I attach the file "/path/file.file" to "form_file"	

CONTOH USER STORY & USER STORY SCENARIO

User Story :

As a Systems Analyst, I want to extract user stories from the owned news text, so that I can obtain user story aspects from the news text.

User Story Scenario :

SCENARIO 1 : Extraction of user story aspects using the 'WHO' method, specifically employing the 'Lexical Names From WordNet' technique, with manually inputted news text.

GIVEN I am on

"http://127.0.0.1:8000/manual/extractionuserstory/"

WHEN I fill in "News Text" with "Earthquakes hit Indonesia, including 7.5 magnitude tremor near Palu"

AND I select "WHO Aspect" from "Aspect"

AND I select "Lexical Names From WordNet" from "WHO Aspect"

AND I press "extract"

THEN I should see "Who aspect" in the "Result News Text Extraction"

SCENARIO 2 : Extraction of user story aspects using the 'WHY' method, specifically employing the 'Syntactic Dependency Parsing' technique, with manually inputted news text.

GIVEN I am on

"http://127.0.0.1:8000/manual/extractionuserstory/"

WHEN I fill in "News Text" with "Earthquakes hit Indonesia, including 7.5 magnitude tremor near Palu"

AND I select "WHY Aspect" from "Aspect"

AND I select "Syntactic Dependency Parsing" from "WHY Aspect"

AND I press "extract"

THEN I should see "Who aspect" in the "Result News Text Extraction"

IMPORT MODUL SELENIUM AND NAMING OF CLASSES AND FUNCTIONS

Setelah membuat file test, selanjutnya modul selenium diimport secara otomatis sebelum menjalankan skenario user story untuk testing. Berikut modul yang biasanya dibutuhkan

- **import unittest**
- **from selenium import webdriver**
- **from selenium.webdriver.common.keys import Keys**

Nama kelas diambil dari nama fitur atau nama user story yang telah ditentukan.

- **Contoh : Class UserStoryExtractionManual ()**

Function yang digunakan berasal dari nama masing-masing skenario. Hal ini berarti setiap skenario mewakili satu fungsi pengujian.

- **Contoh : def test_extraction_who (), def test_extraction_what ()**
-

CONTOH TRANSLATE SCENARIO TO CODE

Scenario Part	Condition	Code Result
Given	If the URL is defined.	<code>selenium.get({URL})</code>
When	If the action is either input or fill and element ID is defined.	<code>variableName = selenium.find_element_by_id({id_element})</code> <code>variableName.send_keys({input})</code>
	If the action is click and element ID is defined.	<code>variableName = selenium.find_element_by_id({id_element})</code> <code>variableName.click()</code>
	If the action is click, a text and element ID is defined.	<code>variableName = selenium.find_element_by_id({id_element})</code> <code>for option in</code> <code>variableName.find_elements_by_tag_name("option"):</code> <code>if option.text == "text":</code> <code>option.click()</code>
	If the action is go and URL is defined.	<code>selenium.get({URL})</code>
	If the action is either see or contain and text is defined.	<code>assert "{text}" in selenium.page_source</code>
	If the action is either see or contain and the element ID is defined.	<code>if selenium.find_element_by_id("{id_element}"):</code> <code>assert True</code> <code>else:</code> <code>assert False</code> <code>assertEqual(selenium.find_element_by_id({id_element})</code>
	If the action is on and the URL is defined.	<code>assertEqual(selenium.current_url, {URL})</code>

Selenium menyediakan beberapa metode untuk menemukan elemen pada halaman website, diantaranya sebagai berikut :

- `find_element_by_id`
- `find_element_by_name`
- `find_element_by_tag_name`
- `find_element_by_xpath`
- `find_element_by_link_text`
- `find_element_by_partial_link_text`
- `find_element_by_class_name`
- `find_element_by_css_selector`

Selenium menyediakan assert yang digunakan untuk memverifikasi apakah kondisi tertentu benar atau tidak selama eksekusi pengujian

- `assertEqual(a, b)`
- `assertNotEqual(a, b)`
- `assertTrue(x)`
- `assertFalse(x)`
- `assertIn(a, b)`
- `assertNotIn(a, b)`
- `assertIs(a, b)`
- `assertIsNot(a, b)`
- `assertRaises(exception, callable, *args, **kwargs)`

BDD**PPL PRAK**

TRANSLATE SCENARIO TO CODE

Scenario	Code
GIVEN I am on "http://127.0.0.1:8000/manual/extractionuserstory/"	<ul style="list-style-type: none">• self.driver = webdriver.Chrome()• self.base_url = "http://127.0.0.1:8000/manual/extractionuserstory/"• self.driver.get(self.base_url)
WHEN I fill in "teks_berita" with "Earthquakes hit Indonesia, including 7.5 magnitude tremor near Palu"	<ul style="list-style-type: none">• teks_berita = "Earthquakes hit Indonesia, including 7.5 magnitude tremor near Palu"• self.driver.find_element(By.NAME,'teks_berita').send_keys(teks_berita)
<ul style="list-style-type: none">• AND I select "WHO Aspect" from "'id_aspect"• AND I select "Lexical Names From WordNet" from "WHO Aspect"	<ul style="list-style-type: none">• aspect_select = Select(self.driver.find_element(By.ID,'id_aspect'))• aspect_select.select_by_value('1')
AND I press "btnExtract"	<ul style="list-style-type: none">• extract_button = self.driver.find_element(By.ID,'btnExtract')• extract_button.click()
THEN I should see "Who aspect" in the "Result News Text Extraction"	<ul style="list-style-type: none">• result_text = self.driver.find_element("id", "berita")• self.assertIsNotNone(result_text.text)

BDD

PPL PRAK

RUNNING TEST

SELURUH TEST

python manage.py test

SEBAGIAN TEST

python manage.py test namafolder.namafile

```
DevTools listening on ws://127.0.0.1:54358/devtools/browser/9e98975a-e210-4398-bf76-2b25192d8081
Hasil Ekstraksi Aspek ALL: <selenium.webdriver.remote.webelement.WebElement (session="ce297076ee6f2d9d704d082c0e72a197", element="AC3B058ADBC93E91575723B6C7D3DFEE_element_24")>
.
DevTools listening on ws://127.0.0.1:54391/devtools/browser/f4c205c0-30d4-4a24-a9c3-35ffeacca7fe
Hasil Ekstraksi Aspek WHAT: <selenium.webdriver.remote.webelement.WebElement (session="d216c40f4a1cee6272589ebf1cf46aae", element="3152EF60D46B75BE0C8DA93547D81421_element_22")>
.
DevTools listening on ws://127.0.0.1:54421/devtools/browser/49ff18ef-f5a6-4150-ac66-e95d456e076
Hasil Ekstraksi Aspek WHO: <selenium.webdriver.remote.webelement.WebElement (session="d55fcf926a87292d550f484e341ccbcf", element="2486404AA7A65A787301079B6A840AAE_element_22")>
.
DevTools listening on ws://127.0.0.1:54454/devtools/browser/6ca47013-895a-4508-9472-dafe7716289f
Hasil Ekstraksi Aspek WHY: <selenium.webdriver.remote.webelement.WebElement (session="639cb01d7b5b09ab442fdbf6b39242ef", element="F0B28F658B514091F3579467FC2C4340_element_22")>
.
-----
Ran 4 tests in 109.649s

OK
```

Gambar disamping menunjukkan hasil running test, dimana pada hasil test menunjukkan

- Running 4 test sesuai dengan skenario testing yang terdiri dari 4 skenario
- Masing-masing skenario jika berhasil akan menampilkan mengeksekusi skenario selanjutnya. Di sini sebagai penanda bahwa skenario berhasil menggunakan print "Hasil Ekstraksi Aspek (sesuai aspek)"
- Di bagian akhir akan menampilkan "OK" yang menunjukkan bahwa testing berhasil.

BDD

PPL PRAK

THANK YOU