

PEMBANGUNAN PENGUJIAN PADA NEO-CLI



Azzam Syawqi Aziz

155150207111132



Agenda

- Permasalahan & Rumusan Masalah
- Landasan Kepustakaan
- Metodologi Penelitian
- Hasil & Pembahasan
- Kesimpulan & Keberlanjutan

1

Permasalahan







(Wong, Debroy, & Restrepo, 2009)

6,404,200,000,000.00

(Wong, Debroy, & Restrepo, 2009)



Prediksi kegagalan dini



Kualitas PL

(Burnstein, 2006)



Pengujian PL



Rumusan Masalah & Tujuan

Bagaimana perancangan dan implementasi pengujian dengan teknik :

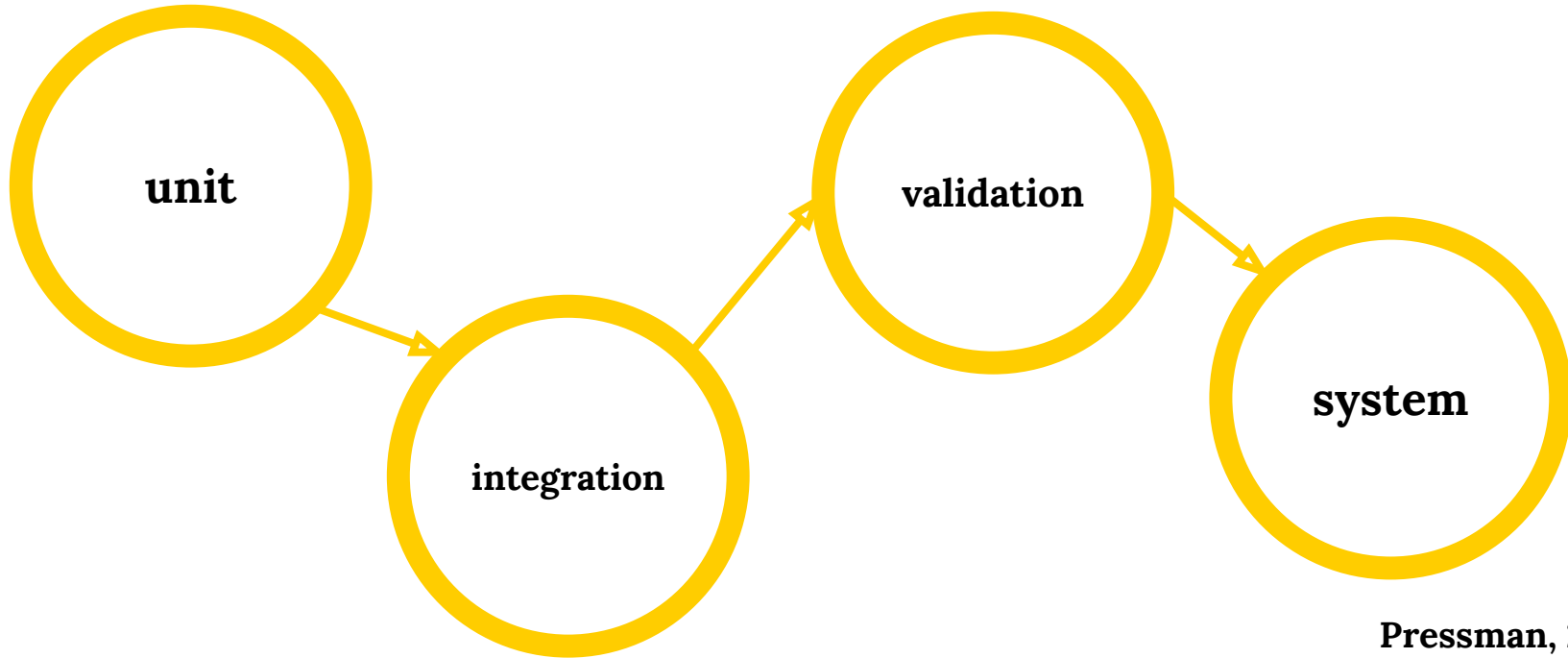
- Basis Path Testing
- Equivalence Partitioning
- Boundary Value Analysis

2

Landasan Kepustakaan



Tahapan Pengujian



Pressman, 2010



Metode Pengujian

White-box Testing

Menguji internal PL

(Myers, Sandler, & Badge , 2011)

Tidak dapat menemukan kebutuhan yang belum diimplementasikan (Dijkstra, 1970).

Black-box Testing

Menguji bagian luar PL

Tidak dapat menemukan bagian yang belum diuji (Savenkov, 2008)

Rigorous testing ?
Exhaustive testing (C^∞) ?

(Pressman, 2011)



Teknik Pengujian

Basis Path Testing

Batas atas untuk membuat kasus–uji (Gregory, 2007).

Equivalence Partitioning

Membagi masukan kepada kelas–kelas data (Pressman, 2010).



Teknik Pengujian

Boundary Value Analysis

Nilai pada boundary memiliki peluang galat tinggi (Pressman, 2010).

Automated Testing

Pengujian menyita 50% dari pengembangan (Brooks Jr, 1995).



Metodologi Penelitian





Analisis Kebutuhan Pengujian

11 Daftar *method* yang akan diuji



Method do_login

No	do_login
1	procedure do_login()
2	TRY (1)
3	username = get_username() (2)
4	password = get_password()
5	generate_session(username, password)
6	return true
7	EXCEPT
8	return false (3)
9	ENDTRY (4)
10	end

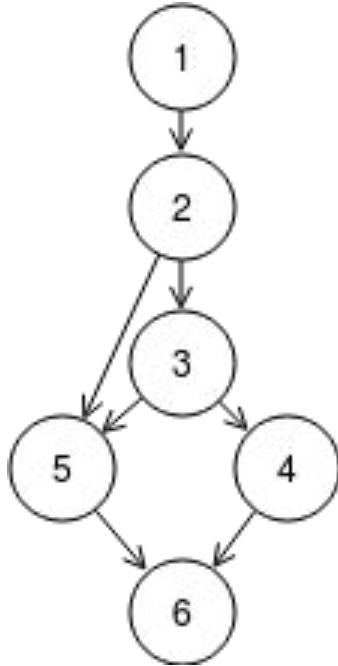


Pengujian Unit get_username

No	get_username
1	procedure get_username()
2	prompt username_input (1)
3	IF username_input <= 5 and (2)
4	username_input >= 255 (3)
5	return username_input (4)
6	ELSE
7	return false (5)
8	ENDIF (6)
9	end



Flow Graph get_username



$V(G) = 3$ regions = 3.

Independent Path

1. Jalur 1 : 1 - 2 - 5 - 6
2. Jalur 1 : 1 - 2 - 3 - 5 - 6
3. Jalur 1 : 1 - 2 - 3 - 4 - 6



Kasus Uji get_username

Jalur	Prosedur Uji	<i>Expected Result</i>	<i>Result</i>	Status
1	Memberikan nilai <i>false</i> pada kondisi "username_input <= 5"	Sistem mengembalikan nilai <i>false</i>	As <i>expe- cted</i>	Valid

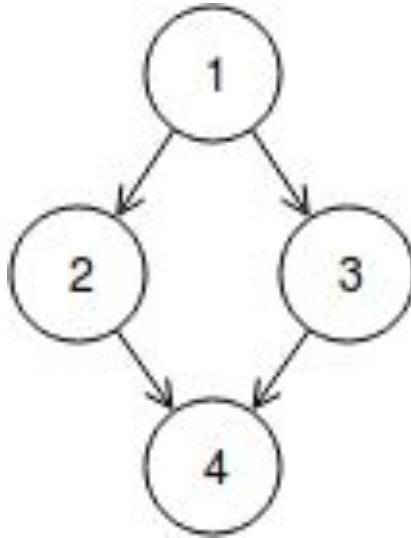


Pengujian Integrasi do_login

No	do_login
1	procedure do_login()
2	TRY (1)
3	username = get_username() (2)
4	password = get_password()
5	generate_session(username, password)
6	return true
7	EXCEPT
8	return false (3)
9	ENDTRY (4)
10	end



Flow Graph do_login



$V(G) = 2 \text{ regions} = 2$

Independent Path

1. Jalur 1 : 1 - 2 - 4

2. Jalur 2 : 1 - 3 - 4

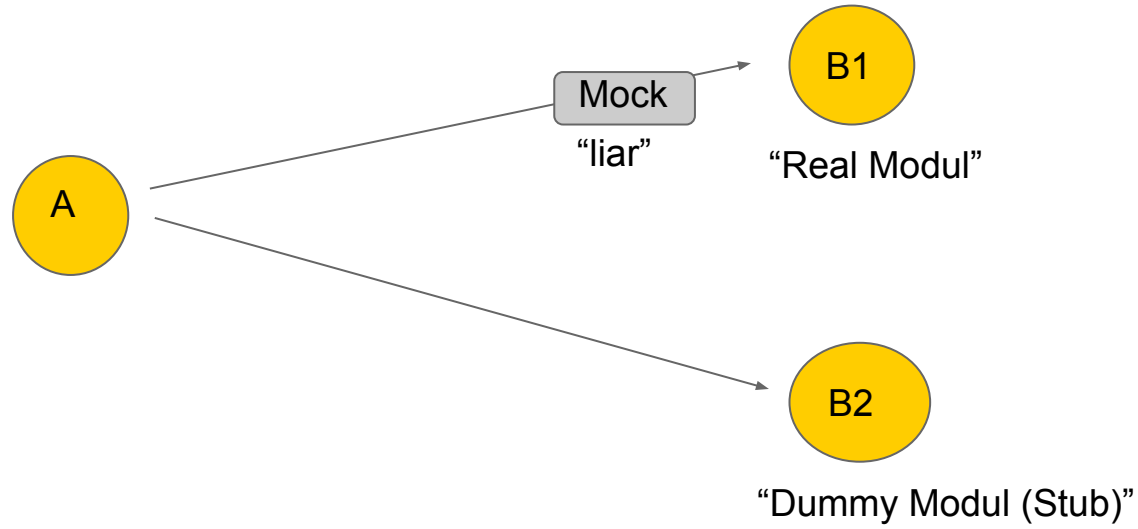


Kasus Uji do_login

Jalur	Prosedur Uji	<i>Expected Result</i>	<i>Result</i>	Status
1	Memberikan nilai <i>raise exception</i>	Sistem mengautentikasi pengguna	As <i>expe- cted</i>	Valid



Isolasi pengujian





Pengujian Input do_login (EP)

Input	<i>Valid Class</i>	<i>Invalid Class</i>
Nilai <i>username</i>	Segala karakter dengan jumlah batas minimum 5 dan maksimum 255	Segala karakter dengan jumlah kurang dari 5 dan lebih dari 255



Kasus Uji EP

- valid : ‘azzamsa’ [7]
- invalid : [karakter berjumlah 2],
[karakter berjumlah 260]



Pengujian Input do_login (BVA)

Input	<i>Boundary Value</i>
Nilai <i>username</i>	Jumlah karakter satu tingkat dibawah batas minimum dan satu tingkat diatas maksimum



Kasus Uji BVA

- Satu dibawah minimum : [karakter berjumlah 4]
- Satu diatas maksimum : [karakter berjumlah 6]

No	test_do_login
----	---------------

1	<code>@pytest.mark.run(order=0)</code>
2	<code>def test_do_login(self, monkeypatch):</code>
3	<code> login.load_env_file()</code>
4	<code> username = os.environ.get('OS_USERNAME')</code>
5	<code> passwd = os.environ.get('OS_PASSWORD')</code>
6	<code> # give value to input() prompt</code>
7	<code> monkeypatch.setattr('builtins.input', lambda x:</code>
	<code> ↪ username)</code>
8	<code> monkeypatch.setattr('getpass.getpass', lambda x:</code>
	<code> ↪ passwd)</code>
9	<code> # return True is login succeed</code>
10	<code> output = login.do_login()</code>
11	<code> assert output == True</code>



Automated Testing



Automated Testing

 BiznetGIO / neo-cli 

Current Branches Build History Pull Requests > Build #109

✓ **devel** upgrade keystoneauth1

🔗 #109 passed

🔗 Commit 8c7e729 [↗](#)

🕒 Ran for 2 min 1 sec

🔗 Compare ebf8039..8c7e729 [↗](#)

📅 3 months ago

🔗 Branch devel [↗](#)

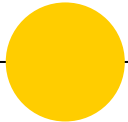
 azzamsa



Hasil & Cakupan Pengujian

tests/test_auth.py::TestAuth::test_do_login	PASSED	[10%]
tests/test_create.py::TestCreate::test_do_create	PASSED	[20%]
tests/test_update.py::TestUpdate::test_do_update	PASSED	[30%]
tests/test_attach.py::TestAttach::test_attach_vm	PASSED	[40%]
tests/test_help.py::TestHelp::test_returns_usage_information	PASSED	[50%]
tests/test_help.py::TestVersion::test_returns_version_information	PASSED	[60%]
tests/test_ls.py::TestLs::test_ls_stack	PASSED	[70%]
tests/test_ls.py::TestLs::test_ls_net	PASSED	[80%]
tests/test_remove.py::TestRemove::test_do_delete_vm	PASSED	[90%]
tests/test_auth.py::TestAuth::test_do_logout	PASSED	[100%]

Kesimpulan & Keberlanjutan





Kesimpulan

- Nilai cakupan tidak dapat dipatok secara umum.
- Modul dengan $V(G)$ tinggi, membutuhkan refactoring.
- Automated Testing mempercepat proses pengujian



Keberlanjutan

- Penggunaan “real resource” patut dihindari. Sebaiknya gunakan mocking.
- Perlunya standar tata pelaksanaan pada proses pull request. Seperti code style (PEP 8)



Terima Kasih!

Pertanyaan ?



Credits

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival
- Photographs by Unsplash