

Pembangkitan Kasus Uji Otomatis Pengujian Aplikasi Web Menggunakan UML Model Based

Rahma Ainun Nisa¹, Sri Widowati², Donni Richasdy³

^{1,2,3}Fakultas Informatika, Universitas Telkom, Bandung

¹rahmaainisa@students.telkomuniversity.ac.id, ²sriwidowati@telkomuniversity.ac.id,

³donnir@telkomuniversity.ac.id

Abstrak

Pengujian perangkat lunak merupakan peran yang penting selama proses aplikasi dikembangkan. Lebih dari 50% bagian dari pengembangan perangkat lunak dihabiskan untuk pengujian. Pengujian dilakukan untuk menemukan kesalahan/defect pada perangkat lunak. Pembangkitan kasus uji merupakan proses penting yang dilakukan pertama kali pada proses pengujian yang dapat dilakukan dari desain kode program perangkat lunak/code based atau desain dari spesifikasi kebutuhan/model based menggunakan Unified Model Language (UML) behavioral diagram. Pembangkitan kasus uji dari spesifikasi kebutuhan/model based lebih efisien dari waktu dan usahanya. Statechart diagram digunakan sebagai model untuk membangkitkan kasus uji yang merepresentasikan aspek dinamis sistem melalui flowchart antara state. Maka pada penelitian ini akan melakukan pembuatan sistem untuk pembangkitan kasus uji otomatis menggunakan model statechart diagram dengan algoritma Depth First Search (DFS). Untuk pengujian aplikasi web menggunakan tools Selenium pada website aplikasi kepegawaian dan menerapkan metode equivalence partitioning untuk memvalidasi antara spesifikasi software. Hasil dari pembangkitan kasus uji secara automasi mendapat hasil kasus uji yang sama dengan spesifikasi dan untuk pengujian juga dihasilkan 100%.

Kata kunci : Depth First Search (DFS), model based testing, software testing, statechart diagram, xmi

Abstract

Software testing is an important role during the application development process. More than 50% part of software development is spent on testing. Tests are carried out to find errors / defects in the software. Test case generation is an important process that is carried out the first time in the testing process which can be done from the design of the software program code / code based or the design of the requirements specification / model based using the Unified Model Language (UML) behavioral diagram. Generating test cases from requirement specifications/model based is more efficient than time and effort. Statechart diagrams are used as models to generate test cases that represent dynamic aspects of the system through flowcharts between states. So in this study, we will build a system for automatic test case generation using a statechart diagram model with the Depth First Search (DFS) algorithm. For testing web applications using Selenium tools on the personnel application website and applying the equivalence partitioning method to validate between software specifications. The results of the automatic generation of test cases get the same test case results as the specifications and 100% of the test cases are also generated.

Keywords: Depth First Search (DFS), model based testing, software testing, statechart diagram, xmi

1. Pendahuluan

Latar Belakang

Pengujian perangkat lunak menjadi hal yang penting dalam menentukan kualitas perangkat lunak. Lebih dari 50% bagian dari pengembangan perangkat lunak dihabiskan untuk pengujian [1] [2]. Pengujian dilakukan untuk menemukan kesalahan/defect pada perangkat lunak. Penguji perlu untuk merencanakan strategi pengujian, mulai dari pembuatan kasus uji, eksekusi dari kasus uji dan juga harus dicek apakah expected result dengan actual result sudah sesuai yang mana apabila dilakukan secara manual akan memakan waktu dan memerlukan kontribusi manusia [3]. Maka pembuatan kasus uji merupakan proses penting yang dilakukan pertama kali pada proses pengujian [4]. Pembuatan kasus uji dapat dilakukan secara manual ataupun otomatis.

Kasus uji dapat dihasilkan dengan dua pendekatan yaitu, dari desain kode program perangkat lunak/code based atau desain spesifikasi kebutuhan/model based menggunakan Unified Model Language (UML) behavioral diagram. Pembangkitan kasus uji dari spesifikasi kebutuhan/model based lebih efisien dari waktu dan usahanya, dibandingkan code based [4]. Terdapat beberapa model diagram pada UML yang umum digunakan untuk membangkitkan kasus uji seperti statechart diagram, sequence diagram, activity diagram [5].

Pada penelitian yang dilakukan hashim [5], model statechart diagram lebih baik dibandingkan model diagram lainnya karena diagram ini menunjukkan perilaku sistem dan perubahan yang terjadi antara event dan transisi. Statechart diagram digunakan sebagai model untuk membangkitkan kasus uji. Maka pada penelitian ini akan melakukan pembangunan sistem untuk pembangkitan kasus uji otomatis menggunakan model statechart

diagram dengan algoritma *Depth First Search* (DFS) dan menerapkan metode *equivalence partitioning* untuk menvalidasi antara spesifikasi hasil kasus uji.

Topik

Pengujian diperlukan untuk menjamin kualitas suatu perangkat lunak. Pembangunan kasus uji dengan model statechart diagram diperlukan validasi agar mendapatkan kasus uji yang benar, maka diperlukan pendefinisian kasus uji untuk memudahkan proses pembangunan kasus uji untuk melakukan pengujian perangkat lunak. Sehingga dapat dirumuskan pada rumusan masalah sebagai berikut

1. Bagaimana membangkitkan kasus uji berbasis *statechart* diagram?
2. Bagaimana kesesuaian antara kasus uji yang dibangkitkan?

Batasan Masalah

Adapun batasan masalah pada penelitian ini sebagai berikut:

1. Pembangkitan kasus uji pada 5 fitur aplikasi web kepegawaian yaitu *login*, *view profile*, *change password*, *change photo*, dan *view absen*, karena hanya ada 5 fitur yang tersedia pada aplikasi web.
2. Untuk menganalisis kasus uji yang dihasilkan sistem menggunakan selenium.

Tujuan

Tujuan dari penelitian ini adalah melakukan pembangunan sistem pembangkitan kasus uji otomatis pengujian perangkat lunak dengan menerapkan model *statechart* diagram.

Organisasi Tulisan

Pada bagian selanjutnya, bagian 2 mengenai studi pustaka tentang teori dan metode terkait penelitian, bagian 3 mengenai rancangan dari sistem yang akan dibangun pada penelitian, bagian 4 mengenai hasil dan analisis dari sistem, dan pada bagian 5 merupakan kesimpulan dari penelitian.

2. Studi Terkait

2.1 Software Testing

Konsep dari *software testing* adalah untuk menemukan *bug*, memastikan aplikasi berfungsi dengan seharusnya [6]. *Testing* adalah komponen penting dari proses *software engineering* untuk menghasilkan produk *software* dengan kualitas yang baik [7]. *Software testing* ini berperan penting untuk keberhasilan *software development* [7].

Software testing dapat dilakukan secara manual ataupun *automation*. *Manual testing* adalah pendekatan di mana penguji yang berperan sebagai *end-user* dan menguji aplikasi apakah fungsionalitas aplikasi sesuai dengan *requirement* atau tidak [3]. Penguji perlu untuk merencanakan strategi pengujian, mulai dari pembuatan *test case*, *execution* dari *test case* juga harus dicek apakah sesuai dengan *expected result* yang dilakukan secara manual [3]. *Automation testing* adalah proses perancangan dan eksekusi *test scripts*, dan menggunakan *automation tools* [8]. Umumnya, *automated testing* terbagi menjadi empat cara dasar, antara lain [6]

- *Prepare/creating test plan*
- *Selection of testing tools*
- *Writing/generating the test script*
- *Performing the test using automated test scripts*

2.2 Model Based Testing

Model based testing (MBT) adalah pendekatan yang menggunakan model dari spesifikasi sistem yang telah dirancang yang bertujuan untuk mengatasi masalah-masalah seperti desain kasus uji secara otomatis dari sistem yang dapat mengurangi biaya desain [9]. Model yang dimaksud adalah model abstrak yang berasal dari perilaku *system under test* (SUT) dan digunakan untuk membangkitkan kasus uji secara otomatis maupun manual [9]. Pada beberapa kasus ini menggunakan UML sebagai dasar desain untuk pembuatan kasus uji [10].

2.3 UML State diagram

Unified Modelling Language (UML) adalah bahasa pemodelan standar untuk merancang perangkat lunak [2]. Terdapat 2 jenis utama UML yaitu *structural diagrams* and *behavioral system*. *Structural diagram* merupakan struktur statik dari sistem yang merepresentasikan konsep dari sistem termasuk

konsep abstrak dan implementasinya [2]. Sedangkan *Behavioral system* merupakan perilaku dinamis dari objek pada sistem. Statechart adalah salah satu bagian dari UML *behavioral system* yang menggambarkan perilaku sistem [11]. Statechart diagram terdiri dari kumpulan *states* dan *transitions* [11]. Gambar 2 di bawah ini menunjukkan statechart diagram mengenai sistem ATM yang sederhana. Terdapat beberapa simbol yang digunakan untuk membuat statechart diagram, yaitu

- States*, digambarkan dengan persegi Panjang dengan sudut bulat yang merupakan perilaku objek di waktu tertentu dan dengan nama sesuai kondisinya. *State* memiliki dua jenis, yaitu *simple state* yang tidak mengandung *sub-states* dan *composite states* yang mengandung *sub-states*. *Composite state* digunakan untuk menyederhanakan *state* yang kompleks.
- Initial state*, digambarkan dengan lingkaran berwarna hitam solid yang merupakan kondisi awal objek sebelum ada perubahan.
- Final state*, digambarkan dengan lingkaran berwarna hitam di tengah yang merupakan kondisi akhir dari sistem.
- Transition*, digambarkan dengan panah yang merupakan perubahan kondisi objek yang disebabkan perubahan sebuah *event*. *Transition* terjadi karena dipicu oleh *event* yang terjadi di antar *state source* dengan *destination*.
- Events*, *event* terjadi jika objek bergerak dari satu *state* ke *state* lainnya.

2.4 XML Metadata Interchange (XMI) [12]

XML *Metadata Interchange* (XMI) adalah standar untuk bertukar informasi metadata melalui *Extensible Markup Language* (XML). XMI bertujuan untuk bertukar informasi mengenai metadata yang dapat membantu *developer* memodelkan model UML. Dengan memodelkan *state* diagram untuk setiap kemungkinan yang terjadi, terdapat dua tag utama dan tag penyusun yaitu,

- Tag `<subvertex>`, merupakan *node* atau *edge*, pada *node* dapat berupa *initial state*, *final state*, dll.
- Tag `<transition>`, merupakan informasi transisi (relasi antara *state*)

2.5 Depth First Search

Algoritma *Depth First Search* (DFS) adalah algoritma yang digunakan untuk pencarian simpul yang memiliki prioritas dilevel terdalam terlebih dahulu, jika jalur sudah sampai dilevel terdalam maka algoritma akan kembali ke *node* awal untuk melakukan proses pencarian sampai semua simpul sudah dikunjungi. Contoh algoritma DFS dapat dilihat pada gambar di bawah ini

```

procedure DFS(input v:integer)
{
  Mengunjungi seluruh simpul graf dengan algoritma
  pencarian DFS
  Masukan: v adalah simpul awal kunjungan
  Keluaran: semua simpul yang dikunjungi ditulis ke
  layar
}

Deklarasi
w : integer

Algoritma
write(v)
dikunjungi[v] <- true
for w <- 1 to n do
  if A[v,w]=1 then
    {simpul v dan simpul w bertetangga }
    if not dikunjungi[w] then
      DFS(w)
    endif
  endfor
endfor

```

Gambar 1 Algoritma DFS

2.6 Selenium

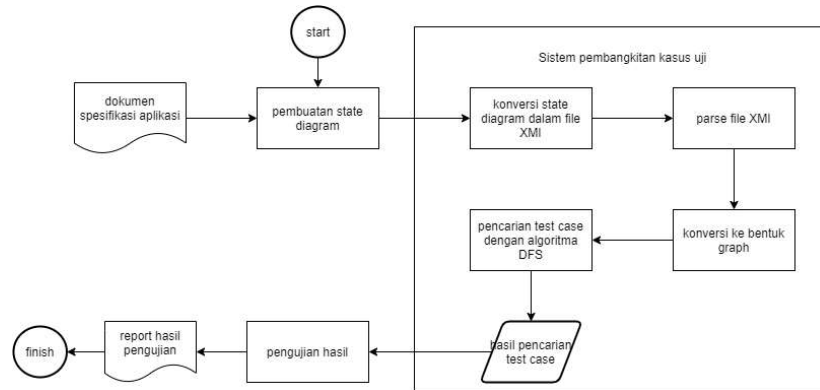
Selenium adalah *framework* yang fokus utamanya terhadap *testing web-based applications* yang sifatnya *open-source* [3]. *Selenium* yang dikembangkan oleh *ThoughtWorks* untuk masalah *automated testing* untuk *web-based application* [13]. *Selenium* terdiri dari empat *tools*, antara lain *Selenium IDE*, *Selenium RC*, *Selenium Webdriver*, dan *Selenium Grid* [6]. Masing-masing komponen memiliki fitur tergantung kepada kebutuhannya masing-masing [3].

Selenium webdriver adalah salah satu komponen *tools* dari *selenium*. Berbeda dari komponen lainnya seperti di *selenium RC* server tidak perlu dimulai untuk eksekusi *test case* di *selenium webdriver* yang langsung berkomunikasi langsung dengan *browser* [3]. Tujuan utama dari *selenium webdriver* yaitu

untuk membantu mengenai masalah *web application testing* [8]. *Selenium webdriver* API lebih sederhana dibandingkan dengan *selenium RC* [8]. Maka *selenium webdriver* dianggap cepat diantara seluruh komponen dari *selenium* [3]. Interaksi antara *selenium webdriver* dengan komponen halaman *website* lebih nyata, jadi hasil pengujian dapat lebih akurat dan nyata [8].

3. Sistem yang Dibangun

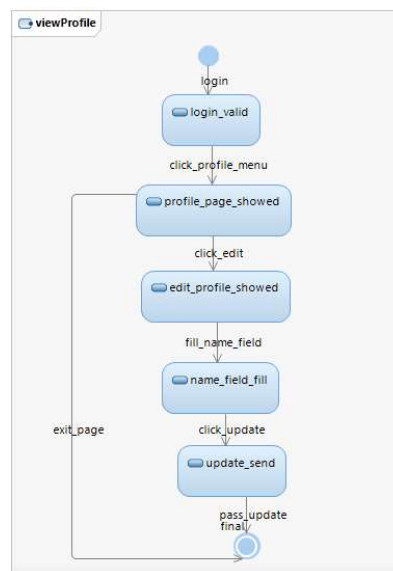
3.1 Alur Penelitian



Gambar 2 Alur penelitian

Penelitian yang dilakukan terbagi menjadi 3 tahap yaitu,

1. Tahap pertama yaitu, pembuatan *statechart* diagram menggunakan *software IBM Rational Software Architecture (RSA)*. Pembuatan diagram ini berdasarkan dokumen spesifikasi aplikasi web. Contohnya pada salah satu fungsi *view profile* pada aplikasi web kepegawaian seperti gambar di bawah ini



Gambar 3 *Statechart* diagram fungsi *view profile*

2. Tahap kedua yaitu, pembangunan sistem pembangkit kasus uji. Berikut penjelasannya,
 - a. *Statechart* diagram dalam file XML
Model *statechart* diagram fungsi login dikonversi ke bentuk XML, model ini digunakan sebagai acuan untuk pembangkitan kasus uji, dapat dilihat pada gambar di bawah ini

```

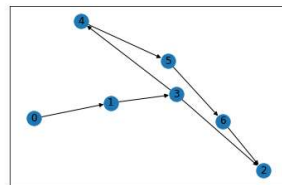
<?xml version="1.0" encoding="UTF-8"?>
<uml:Package xmi:version="2.1" xmlns:xmi="http://schema.omg.org/spec/XMI/2.1" xmlns:xsi="
"http://www.w3.org/2001/XMLSchema-instance" xmlns:ecore="http://www.eclipse.org/emf/2002/Ecore" xmlns:uml=
"http://schema.omg.org/spec/UML/2.2" xsi:schemaLocation="http://schema.omg.org/spec/UML/2.2
http://www.eclipse.org/uml2/3.0.0/UML" xmi:id="rEgKhBLREeyp8_OYHNWmBA" name="model6">
  <packageImport xmi:type="uml:PackageImport" xmi:id="rEgKhRLREeyp8_OYHNWmBA">
    <importPackage xmi:type="uml:Model" href="http://schema.omg.org/spec/UML/2.2/uml.xml#_0"/>
  </packageImport>
  <packageElement xmi:type="uml:StateMachine" xmi:id="rEgKhhlREeyp8_OYHNWmBA" name="viewProfile">
    <region xmi:type="uml:Region" xmi:id="rEgKhxlREeyp8_OYHNWmBA" name="Region1">
      <subvertex xmi:type="uml:Pseudostate" xmi:id="rEgKiBLREeyp8_OYHNWmBA" name="initial"/>
      <subvertex xmi:type="uml:State" xmi:id="rEgKiRLREeyp8_OYHNWmBA" name="login_valid"/>
      <subvertex xmi:type="uml:FinalState" xmi:id="rEgKiHLREeyp8_OYHNWmBA" name="final"/>
      <subvertex xmi:type="uml:State" xmi:id="rEgKixLREeyp8_OYHNWmBA" name="profile_page_showed"/>
      <subvertex xmi:type="uml:State" xmi:id="rEgKjBLREeyp8_OYHNWmBA" name="edit_profile_showed"/>
      <subvertex xmi:type="uml:State" xmi:id="rEgKjRLREeyp8_OYHNWmBA" name="name_field_fill"/>
      <subvertex xmi:type="uml:State" xmi:id="rEgKjhLREeyp8_OYHNWmBA" name="update_send"/>
      <transition xmi:type="uml:Transition" xmi:id="rEgKjxLREeyp8_OYHNWmBA" name="login" source=
        "rEgKiBLREeyp8_OYHNWmBA" target="rEgKiRLREeyp8_OYHNWmBA"/>
      <transition xmi:type="uml:Transition" xmi:id="rEgKkBLREeyp8_OYHNWmBA" name="click_profile_menu"
        source="rEgKiRLREeyp8_OYHNWmBA" target="rEgKixLREeyp8_OYHNWmBA"/>
      <transition xmi:type="uml:Transition" xmi:id="rEgKjRLREeyp8_OYHNWmBA" name="click_edit" source=
        "rEgKixLREeyp8_OYHNWmBA" target="rEgKjBLREeyp8_OYHNWmBA"/>
      <transition xmi:type="uml:Transition" xmi:id="rEgKkHLREeyp8_OYHNWmBA" name="fill_name_field" source=
        "rEgKjBLREeyp8_OYHNWmBA" target="rEgKjRLREeyp8_OYHNWmBA"/>
      <transition xmi:type="uml:Transition" xmi:id="rEgKkxLREeyp8_OYHNWmBA" name="click_update" source=
        "rEgKjRLREeyp8_OYHNWmBA" target="rEgKjhLREeyp8_OYHNWmBA"/>
      <transition xmi:type="uml:Transition" xmi:id="rEgKjLREeyp8_OYHNWmBA" name="exit_page" source=
        "rEgKixLREeyp8_OYHNWmBA" target="rEgKiHLREeyp8_OYHNWmBA"/>
      <transition xmi:type="uml:Transition" xmi:id="rEgKRLREeyp8_OYHNWmBA" name="pass_update" source=
        "rEgKjhLREeyp8_OYHNWmBA" target="rEgKiHLREeyp8_OYHNWmBA"/>
    </region>
  </packageElement>

```

Gambar 4 File dalam format XMI pada view profile

b. Parse file XMI

Model yang sudah dikonversi ke bentuk graf yang merepresentasikan model *statechart* diagram dan bagaimana sistem berjalan. Graf dapat dilihat pada gambar di bawah ini



Gambar 5 Graf dari fungsi view profile

c. Proses pencarian kasus uji

Untuk mencari *test case* dilakukan menggunakan algoritma DFS dan menghasilkan 2 kasus uji seperti yang ditunjukkan pada tabel di bawah

Table 1 Daftar node dari fungsi view profile

Id Node	Nama Node
0	Start (Initial State)
1	Login_valid
2	Final
3	Profile_page_showed
4	Edit_profile_showed
5	Nama_field_fill
6	Update_send

Table 2 Test case dari fungsi view profile

ID	Path	Node
1	0, 1, 3, 2	Initial, login_valid, Profile_page_showed, Final
2	0, 1, 3, 4, 5, 6, 2	Initial, login_valid, Profile_page_showed, edit_profile_showed, nama_field_fill, update_send, Final

3. Tahap ketiga yaitu, pengujian hasil, hasil kasus uji yang telah didapat dari sistem akan diuji menggunakan *selenium webdriver* seperti pada tabel di bawah ini. Status *pass* dan *fail* didapatkan dari hasil *expected result* dan *actual result* yang sama. Status *pass* merupakan *test data* yang dimasukkan berhasil, sedangkan status *fail* terjadi karena *actual result* tidak sesuai yang artinya terjadi kesalahan dalam aplikasi web.

Table 3 Hasil pengujian selenium

<i>Test case id</i>	<i>Test case description</i>	<i>Test step</i>	<i>Test data</i>	<i>Expected result</i>	<i>Actual result</i>	Status
TC2	Menampilkan halaman <i>profile</i> pengguna	1. login 2. choose <i>profile</i> menu	-	Menampilkan halaman <i>profile</i> pengguna	halaman <i>profile</i> ditampilkan	Pass
TC3	Pengguna mengubah <i>profile</i>	1. login 2. choose <i>profile</i> menu 3. klik tombol <i>edit</i> 4. isi <i>field</i> yang ingin diubah	Nama pengguna : Rahma	Nama pengguna berubah menjadi 'Rahma'	Nama pengguna berubah menjadi 'Rahma'	Pass
			Nama pengguna : * kosong	Nama pengguna gagal diubah	Nama pengguna berubah menjadi "/kosong	Fail

4. Evaluasi

4.1 Skenario Pengujian 1

Skenario pengujian ini dilakukan untuk menghasilkan sekumpulan kasus uji yang sudah sesuai dengan skenario pada dokumen spesifikasi pada aplikasi web kepegawaian Yayasan Kesehatan Pertamina Jakarta menggunakan model *statechart* diagram. Pada skenario ini melakukan disemua fungsi yang terdiri dari *login*, *view profile*, *change photo*, *change password*, dan *view absen*. Hasil pembangkitan kasus uji dapat dilihat pada tabel di bawah ini yang sudah sesuai dengan skenario spesifikasi yang ada.

Table 4 Analisis hasil kasus uji

No	Fungsi	ID Scenario	Test Path	Node
1.	<i>Login</i>	TC1	0, 1, 2, 3	<i>Initial, Input_nip_pass, Validation, Final</i>
2.	<i>View Profile</i>	TC2	0, 1, 3, 2	<i>Initial, login_valid, Profile_page_showed, Final</i>
		TC3	0, 1, 3, 4, 5, 6, 2	<i>Initial, login_valid, Profile_page_showed, edit_profile_showed, nama_field_fill, update_send, Final</i>
3.	<i>Change Photo</i>	TC4	0, 1, 2, 5	<i>Initial, login_valid, photo_page_showed, Final</i>
		TC5	0, 1, 2, 3, 4, 5	<i>Initial, login_valid, photo_page_showed, choose_photo_from_device, click_save, Final</i>
4.	<i>Change Password</i>	TC6	0, 1, 2, 3, 4	<i>Initial, login_valid, Change_pass_page_showed, Final</i>
		TC7	0, 1, 2, 4	<i>Initial, login_valid, Change_pass_page_showed, click_change_password, Final</i>
5.	<i>View Absen</i>	TC8	0, 1, 2, 4	<i>Initial, login_valid, Absen_page_showed, Final</i>
		TC9	0, 1, 2, 3, 4	<i>Initial, login_valid, absen_page_showed, downloaded, final</i>

Table 5 Analisa hasil pengujian dengan selenium

No	Fungsi	Expected result	Actual result	Status
1.	<i>Login</i>	Menampilkan halaman <i>Dashboard</i> pengguna	<i>Dashboard</i> pengguna sesuai dengan data <i>login</i>	✓
2.	<i>View profile</i>	Menampilkan halaman <i>profile</i> pengguna	<i>Profile</i> yang ditampilkan sesuai dengan data <i>login</i> pengguna	✓

		Salah satu kolom pengguna ter- <i>update</i> dan muncul <i>alert</i> seperti 'sukses'	Kolom pengguna ter- <i>update</i>	✓
3.	<i>Change photo</i>	Menampilkan halaman ubah foto pengguna	Halaman menu ubah foto ditampilkan	✓
		Foto <i>profile</i> pengguna berubah dan muncul <i>alert</i> seperti 'sukses'	Foto pengguna ter- <i>update</i>	✓
4.	<i>Change password</i>	Menampilkan halaman ubah <i>password</i> pengguna	Halaman menu ubah <i>password</i> ditampilkan	✓
		Password pengguna berubah dan muncul <i>alert</i> seperti 'sukses'	<i>Password</i> pengguna ter- <i>update</i>	✓
5.	<i>View absen</i>	Menampilkan halaman absen pengguna	Halaman menu absen ditampilkan	✓
		<i>File</i> berhasil diunggah	<i>File</i> berhasil diunggah	✓

Simbol ✓ pada tabel di atas merupakan keterangan bahwa pengujian menggunakan selenium berhasil menghasilkan sesuai dengan expected result. Hal ini berarti tools pembangkitan kasus uji sudah mampu menghasilkan kasus uji sesuai dengan *statechart* diagram.

4.2 Skenario Pengujian 2

Skenario pengujian yang kedua ini dilakukan menggunakan metode *equivalence partitioning* yaitu membagi data ke dalam dua jenis yaitu data *valid* dan *invalid*. Pada pengujian ini data *valid* didapatkan dari data yang sesuai dengan *statechart* diagram, sedangkan untuk data *invalid* didapatkan dari modifikasi data *valid*. Data *invalid* nantinya akan menghasilkan kasus uji yang terdapat proses sengaja dibuat gagal. Data ini digunakan pada fungsi yang diuji pada aplikasi web kepegawaian. Kemudian, dari dua data tersebut menghasilkan kasus uji yang akan dieksekusi menggunakan selenium. Tabel di bawah ini menunjukkan hasil data yang sengaja dibuat salah

Table 6 Hasil pengujian fungsi change photo

<i>Test case id</i>	<i>Test case description</i>	<i>Test step</i>	<i>Test data</i>	<i>Expected result</i>	<i>Actual result</i>	Status
TC4	Menampilkan halaman <i>change photo</i>	1. login 2. choose <i>change photo menu</i> "	-	menampilkan halaman <i>change photo</i>	halaman <i>change photo</i> ditampilkan	Pass
TC5	<i>Choose img file</i>	1. Choose <i>photo menu</i> 2. Click <i>choose file button</i> 3. Choose <i>img file from device (jpg/png/jpeg)</i> 4. Click on <i>save buton</i> "	file logo.png	foto <i>profile</i> berhasil diubah dan muncul <i>Alert message 'your photo has been updated'</i>	<i>Alert message 'your photo has been updated'</i>	Pass
			file logobesar.png yaitu <i>file > 500 kb</i>	foto <i>profile</i> gagal diubah dan muncul <i>alert message 'The file you are attempting to upload is larger than the permitted</i>	foto <i>profile</i> gagal diubah dan muncul <i>alert message 'The file you are attempting to upload is larger than</i>	Pass

				size.'	the permitted size.'	
			fileSK.pdf	foto profile gagal diuba dan muncul alert message 'the filetype you are attempting to upload is not allowed'	alert message 'the filetype you are attempting to upload is not allowed'	Pass

Dari hasil kasus uji yang didapatkan dari masing-masing spesifikasi data *valid* dan *invalid*, kemudian melakukan pengujian menggunakan *selenium*. Pengujian *selenium* bertujuan untuk mengetahui hasil dari data *valid* dan *invalid* sesuai dengan sistem pembangkit kasus uji yang telah dibangun. Data ini diharapkan mendapatkan hasil *pass* sesuai ekspektasi. Pada *statechart* diagram *invalid* terdapat proses yang *valid*, hanya dimodifikasi ke proses *invalid*. Jumlah kasus uji yang dihasilkan dari data *invalid* sama dengan jumlah kasus uji pada data *valid*, hal ini karena terdapat modifikasi pada data *valid*. Tabel di bawah ini menunjukkan hasil pengujian yang dilakukan menggunakan *selenium*

Table 7 Hasil pengujian menggunakan metode equivalence partitioning

Spesifikasi	Fungsi	Test case	Hasil pengujian		Total test case
			Pass	Fail	
<i>Valid</i>	<i>login</i>	1	1	0	9
	<i>View profile</i>	2	2	0	
	<i>Change photo</i>	2	2	0	
	<i>Change password</i>	2	2	0	
	<i>View absen</i>	2	1	0	
<i>Invalid</i>	<i>login</i>	1	1	0	9
	<i>View profile</i>	2	2	0	
	<i>Change photo</i>	2	2	0	
	<i>Change password</i>	2	2	0	
	<i>View absen</i>	2	1	0	

Tabel 6 di bawah ini menunjukkan hasil data *valid* dan *invalid* yang dilakukan setelah pengujian menggunakan *selenium*.

Table 8 Perhitungan kasus uji data valid

Fungsi	Test case	Pengujian <i>pass/data valid</i>	Persentase kesesuaian (%)
<i>login</i>	1	1/1	100%
<i>View profile</i>	2	2/2	100%
<i>Change photo</i>	2	2/2	100%
<i>Change password</i>	2	2/2	100%
<i>View absen</i>	2	2/2	100%

Table 9 Perhitungan kasus uji data invalid

Fungsi	Test case	Pengujian <i>pass/data invalid</i>	Persentase kesesuaian (%)
<i>login</i>	1	1/1	100%
<i>View profile</i>	2	2/2	100%
<i>Change photo</i>	2	2/2	100%

<i>Change password</i>	2	2/2	100%
<i>View absen</i>	2	0/0	100%

5. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, maka didapatkan beberapa poin kesimpulan, yaitu sebagai berikut

1. Pada tugas akhir ini, peneliti membangun sistem pembangkit kasus uji berdasarkan model *statechart* diagram yang sesuai dengan skenario aplikasi web kepegawaian.
2. Sistem yang dibangun dapat menghasilkan kasus uji yang memiliki persentase kesesuaian sebesar 100% berdasarkan pengujian menggunakan *selenium*. Hal ini membuktikan fungsi yang dibangun sesuai dengan skenario pada spesifikasi aplikasi web kepegawaian.
3. Dari lima fungsi yang diuji terdapat beberapa kasus yang menghasilkan status *fail* yaitu pada fungsi *view profile*, maka artinya pada fungsi ini terdapat kesalahan/*error*. Hal ini berarti sistem yang dibangun telah menghasilkan kasus uji yang benar dan sesuai spesifikasi dan model diagram.

Berikut ini beberapa saran yang dapat digunakan untuk penelitian selanjutnya

1. Pengujian dapat menggunakan model diagram lain seperti *sequence diagram*, *usecase diagram*, atau diagram lainnya agar kasus uji yang dihasilkan lebih beragam
2. Pengujian dapat dilakukan pada aplikasi yang lebih kompleks

Daftar Pustaka

- [1] M. Aggarwal and S. Sabharwal, "Test Case Generation from UML State Machine Diagram: A Survey," *Third International Conference on Computer and Communication Technology*, pp. 133-140, 2012.
- [2] V. Panthi, A. Gardizy, R. K. Mohapatra and D. P. Mohapatra, "Functionality Testing of Object-Oriented Software Using UML State Machine Diagram," *2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*, pp. 1-7, 2018.
- [3] P. Ramya, V. Sindhura and P. V. Sagar, "Testing using Selenium Web Driver," *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pp. 1-7, 2017.
- [4] Meiliana, I. Septian, R. S. Alianto, Daniel and F. L. Gaol, "Automated Test Case Generation from UML Activity Diagram and Sequence Diagram using Depth First Search Algorithm," *2nd International Conference on Computer Science and Computational Intelligence*, pp. 629-637, 2017.
- [5] N. L. Hashim and Y. Dawood, "Test Case Minimization Applying Firefly Algorithm," *International Journal on Advanced Science Engineering and Information Technology*, vol. 8, pp. 1777-1783, 2018.
- [6] N. Islam, "A Comparative Study of Automated Software Testing Tools," *Culminating Projects in Computer Science and Information Technology*, pp. 1-92, 2016.
- [7] D. Absharina, F. and F. Agustina, "SURVEY PAPER: SOFTWARE AUTOMATED TESTING TOOL USING SYSTEMATIC LITERATURE REVIEW METHOD," *Jurnal Pilar Nusa Mandiri*, vol. 16, pp. 175-182, 2020.
- [8] S. Gojare, R. Joshi and D. Gaigaware, "Analysis and Design of Selenium WebDriver Automation Testing Framework," *International Symposium on Big Data and Cloud Computing (ISBCC)*, pp. 341-346, 2015.
- [9] H. G. Gurbuz and B. Tekinerdogan, "Model-based testing for software safety: a systematic mapping study," *Software Qual J*, pp. 1327-1372, 2018.
- [10] R. S. Pressman, *Software Engineering A Practioners Approach*, SEVENTH EDITION, New York: McGraw-Hill, a business unit of The McGraw-Hill Companies, 2010.
- [11] M. Javani, B. S. Neysiani and S. M. Babamir, "Formal Verification of UML Statecharts using LOTUS Formal Language," *International Conference on Knowledge-based Engineering and Innovation (KBEI)*, pp. 755-761, 2015.
- [12] S. L. Jim, "From UML diagrams to behavioural source code," pp. 1-69, 2006.
- [13] A. Holmes and M. Kellog, "Automating Functional Tests Using Selenium," *Proceedings of AGILE 2006 Conference (AGILE'06)*, pp. 1-6, 2006.

Lampiran

Website aplikasi kepegawaian adalah sistem informasi bagi seluruh karyawan mengenai kepegawaian di Yayasan Kesehatan Pertamina yang dikelola oleh divisi SDM. Dengan tujuan untuk memudahkan pekerja dalam mengakses data diri, dapat memperoleh informasi yang baru dan cepat, website ini menyediakan berbagai fitur yang lengkap. Berikut fungsi yang dilakukan pengujian pada website aplikasi ini,

a. Fungsi login

Fungsi ini digunakan pegawai dengan cara memasukkan NIP dan *password*. Di bawah ini merupakan tabel usecase skenario

Nama Usecase	Login	
Aktor	Pegawai	
Deskripsi	Pegawai melakukan login dengan memasukkan NIP dan password yang sudah terdaftar	
Pre-Kondisi	Pengguna akan menggunakan website ini	
Post-Kondisi	Pengguna berhasil masuk sebagai pegawai	
Primary Flow	Pengguna	Sistem
	1. Pengguna melakukan input NIP dan password 2. Pengguna mengklik login	3. Sistem mengecek validasi data yang diinputkan oleh pengguna 4. Sistem menampilkan dashboard
Alternate Flow	Pengguna	Sistem
	1. Pengguna memasukkan NIP salah 2. Pengguna mengklik login	3. Sistem mengecek validasi data yang diinputkan oleh pengguna 4. Sistem menampilkan alert 'Sorry, your nip have not been registered'
	1. Pengguna memasukkan password salah 2. Pengguna mengklik login	3. Sistem mengecek validasi data yang diinputkan oleh pengguna 4. Sistem menampilkan alert 'Wrong password!'

b. Fungsi view profile

Fungsi ini digunakan pegawai untuk melihat profile dan dapat mengubahnya.

Nama Usecase	Lihat profile	
Aktor	Pegawai	
Deskripsi	Pegawai melihat profile masing-masing	
Pre-Kondisi	Pegawai berhasil login dan berada di halaman profile pengguna	
Post-Kondisi	Profile pegawai ter-update	
Primary Flow	User	Sistem
	1. Pengguna melakukan login 4. Pengguna mengklik tombol edit	2. Sistem mengecek login sebagai login valid 3. Sistem menampilkan halaman profile sesuai login pengguna 5. Sistem menampilkan halaman edit

	6. Pengguna mengisi field yang ingin diubah, misal field nama 7. Pengguna mengklik tombol update	8. Sistem mengecek validasi data yang diinputkan oleh pengguna 9. Sistem menampilkan alert 'data berhasil diubah'
Alternate Flow	User	Sistem
	1. Pengguna mengisi field nama pegawai ''/kosong 2. Pengguna men-klik tombol update	3. Sistem mengecek validasi data yang diinputkan oleh pengguna 4. Sistem menampilkan alert 'data tidak boleh kosong'

c. Fungsi change password

Fungsi ini digunakan pegawai untuk mengubah *password*.

Nama Usecase	Change password	
Aktor	Pegawai	
Deskripsi	Pegawai melakukan ubah password	
Pre-Kondisi	Pegawai berhasil login	
Post-Kondisi	Pegawai berhasil mengganti password	
Primary Flow	User	Sistem
	1. Pengguna melakukan login 3. Pengguna memilih menu change password 5. Pengguna memasukkan password baru 6. Pengguna memasukkan Kembali password baru yang sama 7. Pengguna mengklik tombol change password	2. Sistem mengecek sebagai login valid 4. Sistem menampilkan halaman change password 8. Sistem mengecek validasi data yang diinputkan oleh pengguna 9. Sistem menampilkan alert 'password berhasil diubah'
Alternate Flow	User	Sistem
	1. Pengguna memasukkan password baru 2. Pengguna memasukkan Kembali password baru yang berbeda 3. Pengguna mengklik tombol change password	4. Sistem mengecek validasi data

	1. Pengguna memasukkan password baru kurang dari 6 karakter 2. Pengguna memasukkan Kembali password baru yang kurang dari 6 karakter 3. Pengguna mengklik tombol change password	yang diinputkan oleh pengguna 5. Sistem menampilkan alert new password tidak sama dengan repeat password 4. Sistem mengecek validasi data yang diinputkan oleh pengguna 5. Sistem menampilkan alert 'minimum new password must 6 characters'
--	--	---

d. Fitur change photo

Fungsi ini digunakan pegawai untuk mengubah foto.

Nama Usecase	Change foto	
Aktor	Pegawai	
Deskripsi	Pegawai melakukan ubah foto profile	
Pre-Kondisi	Pegawai berhasil login	
Post-Kondisi	Sistem menampilkan foto pengguna	
Primary Flow	User	Sistem
	1. Pengguna melakukan login 3. Pengguna memilih menu photo 5. Pengguna memilih photo profile dari device pribadi 6. Pengguna mengklik ok untuk pemilihan foto 7. Pengguna mengklik save	2. Sistem mengecek sebagai login valid 4. Sistem menampilkan halaman halaman photo 8. Sistem memproses data 9. Sistem menampilkan alert 'your photo has been updated'
Alternate Flow	User	Sistem
	1. Pengguna memasukkan file format pdf untuk mengganti photo profile 1. Pengguna memasukkan file dengan ukuran >500kb untuk mengganti photo profile	2. Sistem mengecek validasi data yang diinputkan oleh pengguna 3. Sistem menampilkan alert 'the file type your attempting is not allowed' 2. Sistem mengecek validasi data yang diinputkan oleh pengguna 3. Sistem menampilkan alert 'the file

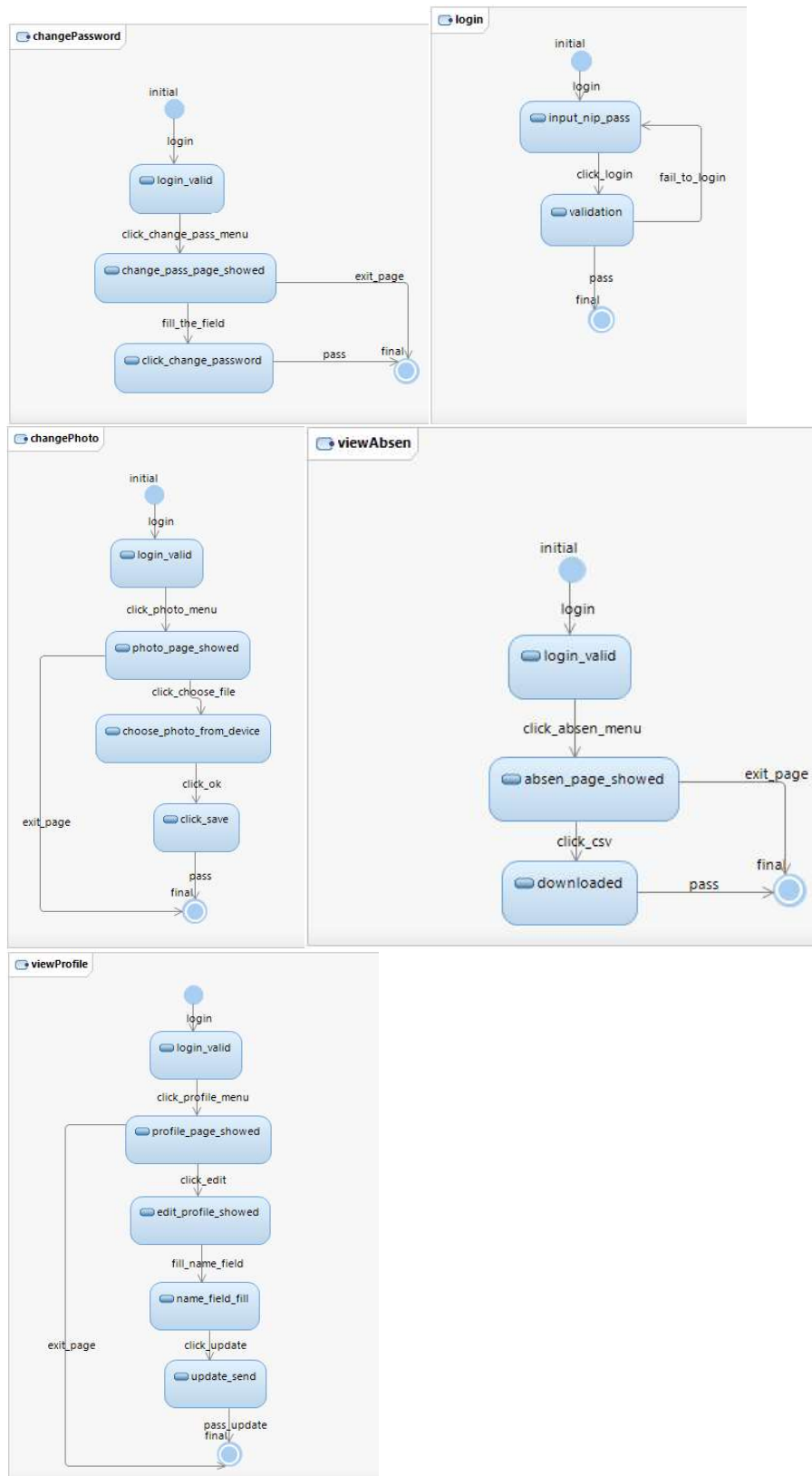
		your attempting to upload is larger than the permitted size'
--	--	--

e. Fitur view absen

Fungsi ini digunakan pegawai untuk melihat daftar absen.

Nama Usecase	Lihat absen	
Aktor	Pegawai	
Deskripsi	Pegawai melakukan lihat absen	
Pre-Kondisi	Pegawai berhasil login	
Post-Kondisi		
Primary Flow	User	Sistem
	1. Pengguna melakukan login 3. Pengguna memilih menu absen 6. Pengguna mengklik tombol csv untuk menyetak absen	2. Sistem mengecek sebagai login valid 4. Sistem menampilkan menu absen 7. Sistem memproses data 8. Sistem mengunduh file
Alternate Flow	-	

Berikut ini merupakan *statechart* diagram dari setiap fungsi



Berikut ini hasil kasus uji dan pengujian selenium yang dihasilkan

Test case id	Test case description	Test step	Test data	Expected result	Actual result	Status
--------------	-----------------------	-----------	-----------	-----------------	---------------	--------

TC1	Enter a valid NIP & password	1. Enter valid NIP and password 2. Click on login button	NIP: 1301172733 Password: 123456 (min6 length)	menampilkan halaman dashboard pengguna	dashboard pengguna sesuai dengan data login	Pass
	Enter an invalid NIP & valid password	1. Enter an invalid NIP and valid password 2. Click on login button	NIP: 123456789 (nip not registered in db) Password: 1234567	menampilkan alert message 'Sorry your nip have not been registered'	gagal login aplikasi dan muncul alert message 'Sorry your nip have not been registered'	Pass
	Enter a valid NIP & an invalid password	1. Enter valid NIP and an invalid password 2. Click on login button	NIP: 1301172733 Password: abcdefg	menampilkan alert message 'wrong password'	gagal login aplikasi dan muncul alert message 'wrong password'	Pass
	Enter a valid nip & pass < 6		NIP: 1301172733 Password: 123	menampilkan alert message 'The Password field must be at least 6 characters in length'	gagal login aplikasi dan muncul alert message 'The Password field must be at least 6 characters in length'	Pass

Test case id	Test case description	Test step	Test data	Expected result	Actual result	Status
TC2	Menampilkan halaman profile pengguna	1. login 2. choose profile menu	-	Menampilkan halaman profile pengguna	halaman profile ditampilkan	Pass
TC3	Pengguna mengubah profile	1. login 2. choose profile menu 3. klik tombol edit 4. isi field yang ingin diubah	Nama pengguna : Rahma	Nama pengguna berubah menjadi 'Rahma'	Nama pengguna berubah menjadi 'Rahma'	Pass
			Nama pengguna : * kosong	Nama pengguna gagal diubah	Nama pengguna berubah menjadi "/kosong	Fail

Test case id	Test case description	Test step	Test data	Expected result	Actual result	Status
TC4	Menampilkan halaman	1. login 2. choose	-	menampilkan halaman	halaman change	Pass

	<i>change photo</i>	<i>change photo menu"</i>		<i>change photo</i>	<i>photo ditampilkan</i>	
TC5	<i>Choose img file</i>	1. Choose photo menu 2. Click choose file button 3. Choose img file from device (jpg/png/jpeg) 4. Click on save buton"	file logo.png	foto profile berhasil diubah dan muncul Alert message 'your photo has been updated'	Alert message 'your photo has been updated'	Pass
			file logobesar.png yaitu file > 500 kb	foto profile gagal diubah dan muncul alert message 'The file you are attempting to upload is larger than the permitted size.'	foto profile gagal diubah dan muncul alert message 'The file you are attempting to upload is larger than the permitted size.'	Pass
			fileSK.pdf	foto profile gagal diuba dan muncul alert message 'the filetype you are attempting to upload is not allowed'	alert message 'the filetype you are attempting to upload is not allowed'	Pass

<i>Test case id</i>	<i>Test case description</i>	<i>Test step</i>	<i>Test data</i>	<i>Expected result</i>	<i>Actual result</i>	<i>Status</i>
TC6	Menampilkan halaman change password	1. login 2. choose change password menu	-	menampilkan halaman change password	halaman change password ditampilkan	Pass
TC7	<i>Enter a valid new password</i>	1. Choose password menu 2. Enter a valid new password 3. Re-enter a valid new password 4. Click on save buton	*(min 6 length) New password: abcdefg Re-enter new password: abcdefg	password berhasil diubah	password berhasil diubah dan muncul Alert message 'password changes'	Pass
			*(min 6 length) New	password gagal diubah	password gagal diubah	Pass

			password: abcdefg Re-enter new password: 123456		dan muncul alert message 'Re-enter your new password'	
			*(min 6 length) New password: abc Re-enter new password: lal	password gagal diubah	password gagal diubah dan muncul alert 'minimum new password must 6 character'	Pass

Test case id	Test case description	Test step	Test data	Expected result	Actual result	Status
TC8	Absen page displayed	1. login 2. choose absen menu	-	Absen page displayed	Absen page displayed	Pass
TC9	Absen page displayed dan menyetak daftar absen	1. Choose Absen menu 2. click csv to download file	-	File Csv berhasil didownload	file csv berhasil didownload dan tersimpan didevice	Pass