

## PENGUJIAN BLACKBOX PADA WEBSITE WORKER'S MENGGUNAKAN METODE EQUIVALENCE PARTITIONING

Oleh:

Derick Triady<sup>1</sup>, Izmy Alwiah Musdar<sup>2\*</sup>, Hendra Surasa<sup>3</sup>

<sup>1,2,3</sup>Teknik Informatika, STMIK Kharisma Makassar

e-mail: <sup>1</sup>[dericktriady\\_19@kharisma.ac.id](mailto:dericktriady_19@kharisma.ac.id) <sup>2</sup>[izmyalwiah@kharisma.ac.id](mailto:izmyalwiah@kharisma.ac.id)

<sup>3</sup>[hendrasurasa@kharisma.ac.id](mailto:hendrasurasa@kharisma.ac.id)

**Abstrak:** Pengujian software dilakukan untuk mendapatkan kesalahan yang ada pada software sebelum ditemukan oleh pengguna. Penelitian ini menggunakan metode black box pada website worker's, yang merupakan website penyedia informasi mengenai lowongan pekerjaan. Metode pengujian yang digunakan yaitu pengujian black box dengan metode equivalence partitioning sebagai teknik perancangan test case dengan menggunakan teknik observasi untuk mengumpulkan data dan teknik analisis data kualitatif untuk melakukan pengolahan data. Equivalence partitioning merupakan metode yang membagi data mengenai domain masukan menjadi partisi-partisi, dari partisi-partisi tersebut dapat menghasilkan test case. Hasil pengujian menunjukkan bahwa pengujian black box dengan metode equivalence partitioning dapat digunakan untuk menemukan kesalahan pada software. Terdapat 29 test case dimana 68% test case tersebut berhasil atau pass dan 32% test case tersebut gagal atau fail.

**Kata kunci:** Black box testing, Pengujian perangkat lunak, Website, Test case, Equivalence partitioning.

**Abstract:** Software testing is carried out to find error that exist in the software before they are found by the user. This research uses the black box method on the worker's website, which is a website that provides information about job vacancies. The testing method used is black box testing with the equivalence partitioning method as a test case design technique by using observation techniques to collect data and qualitative data analysis techniques to process data. The equivalence partitioning is a method that divides data regarding input domain into partitions, from these partitions can produce test cases. The test result show that black box testing with the equivalence partitioning method can be used to find error in software. There are 29 test case where 68% of the test case successful or passed and 32% of the test case failed.

**Keywords:** Black box testing, Software testing, Website, Test case, Equivalence partitioning.

### 1. PENDAHULUAN

Pengujian software adalah proses untuk mengevaluasi kualitas software [1]. Pengujian merupakan proses menjalankan software untuk menemukan kesalahan, dimana pengujian tersebut sebaiknya menemukan kesalahan yang tidak disengaja dan pengujian akan dibilang sukses apabila berhasil menemukan serta memperbaiki masalah tersebut. Serta pengujian juga dilakukan untuk menunjukkan bahwa fungsi – fungsi perangkat lunak tersebut telah jalan

\* Corresponding author : Izmy Alwiah Musdar (izmyalwiah@kharisma.ac.id)

sesuai dengan spesifikasinya. Perangkat lunak akan dikatakan gagal, apabila perangkat lunak tersebut tidak memenuhi spesifikasi [2].

Pengujian *software* penting dilakukan agar tidak menghasilkan kerugian seperti kejadian pada *Knight Capital Group* yang mengalami kerugian sebesar 440 Juta *dollar* dimana terjadi karena kesalahan teknis dari *software* yang digunakan oleh *Knight Capital Group* sehingga mengalami kerugian akibat *software* membeli saham dengan harga tinggi [3]. Apabila *software* yang dikembangkan tidak bekerja sesuai dengan seharusnya, maka dapat berakibat fatal yang dapat merugikan *user* maupun perusahaan. Dalam melakukan pengujian *software* terdapat 2 metode pengujian yang biasa digunakan yaitu pengujian *Black Box* dan pengujian *White Box*.

Dalam penelitian ini, metode yang digunakan merupakan pengujian *Black Box*. Pengujian *Black Box* merupakan pengujian yang berfokus terutama kepada fungsionalitas *software* [4]. Pengujian *black box* biasanya dilakukan tanpa melihat kode dari *software* yang ingin diuji [5]. Sedangkan pengujian *White Box* merupakan pengujian yang berfokus terutama kepada kode *software* tanpa melihat tampilan *interface* *software* tersebut [6]. Pengujian *Black Box* dapat menemukan kesalahan seperti fungsi yang salah atau hilang, kesalahan antarmuka, kesalahan dalam struktur data atau akses database eksternal, kesalahan kinerja, inisialisasi dan kesalahan terminasi, validasi fungsional, kesensitifan sistem terhadap nilai input tertentu, dan batasan suatu data [7]. Pengujian *black box* memiliki keuntungan seperti pengujian tidak perlu memiliki pengetahuan tentang pemrograman, pengujian ini dilakukan dalam sudut pandang pengguna, hal ini dapat membantu mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan, programmer dan tester saling bergantung satu dengan yang lain [8].

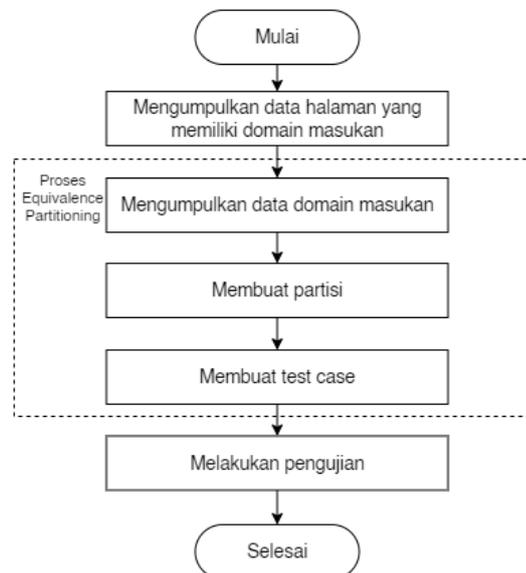
Pengujian *Black box* memiliki beberapa jenis metode yaitu *Equivalence Partitioning* [1], *Boundary Value Analysis* [4], *Fuzzing Testing* [9], *Cause Effect Graph* [10], *Orthogonal Array Testing* [11], *All Pair Testing* [12], dan *State Transition Testing* [13]. Metode pengujian *black box* yang digunakan dalam penelitian ini yaitu *Equivalence Partitioning*. *Equivalence Partitioning* merupakan metode pengujian yang dilakukan dengan cara membagi domain masukan menjadi kelas-kelas data sehingga dapat menghasilkan *test case* [14]. *Test Case* pada *equivalence partitioning* dibuat berdasarkan evaluasi partisi atau kelas ekuivalensi pada setiap kondisi input yang valid maupun tidak valid [14]. Metode ini memiliki kelebihan seperti dapat membuat *test case* yang lebih spesifik, bisa mendapatkan kekurangan dari sistem berdasarkan *test case* yang diuji [15], dapat mengurangi jumlah *test case* dimana pada penelitian ini dilakukan perbandingan antara dua metode dan mendapatkan hasil bahwa *test case* yang perlu dibuat lebih sedikit dibandingkan dengan *boundary value analysis* yang membuat *test case* hampir dua kali dari jumlah *test case equivalence partitioning*[16].

Worker's (<https://ccd-workers.com/>) merupakan *website* penyedia informasi lowongan pekerjaan terbaru bagi *user*. *Website* ini dirancang agar *user* dapat menggunakan *software* ini untuk menghemat waktu pencarian lowongan dan mendapatkan informasi yang valid terkait lowongan pekerjaan. Dalam penelitian ini dilakukan pengujian *black box* pada Worker's untuk menguji *website* worker's karena terdapat error yang dilaporkan oleh pengguna mengenai

validasi pada *input* pada *website worker's*. Selain itu, *website worker's* belum pernah dilakukan pengujian validasi input untuk memastikan semua data inputan menghasilkan *output* yang diharapkan. Pengujian *black box* ini menggunakan metode *equivalence partitioning* untuk menguji domain masukan yang terdapat pada *worker's*, dimana pengujian ini dilakukan pada tahap *testing* dalam siklus *SDLC*, tahapan *testing* dilakukan setelah tahap implementasi.

Teknik perancangan *test case* yang digunakan adalah *equivalence partitioning*. Pengujian dilakukan sehingga dapat mengetahui apakah *worker's* memiliki *error*, kesalahan, dan ketidaksesuaian pada fungsionalitasnya. Sejumlah studi terkait mengenai pengujian *black box* dengan metode *equivalence partitioning* telah dilakukan, diantaranya penelitian dengan judul Implementasi Teknik *Equivalence Partitioning* pada Pengujian Aplikasi *E-learning* Berbasis Web yang bertujuan untuk melakukan pengujian pada bagian validasi data pada Aplikasi *E-Learning* untuk mencegah terjadinya *overload* pada sistem [1], dan pada penelitian dengan judul Pengujian *Software* Aplikasi Perawatan Barang Milik Negara Menggunakan Metode *Black Box Testing Equivalence Partitions*, yang bertujuan melakukan pengujian untuk mengetahui tingkat efektifitas metode *equivalence partitioning* [8]. Berdasarkan kedua penelitian tersebut metode *equivalence partitioning* efektif digunakan untuk melakukan pengujian pada aplikasi. Dalam penelitian ini, dilakukan pengujian *black box* dengan metode *equivalence partitioning* terhadap *website Worker's*.

## 2. METODE PENELITIAN



Gambar 1. Alur Tahap Penelitian

Pengujian ini dilakukan oleh *developer* sendiri, agar dapat mendeteksi penyebab kesalahan yang terjadi saat pengujian dan mengurangi kemungkinan melakukan pengujian yang sama. Tahapan penelitian yang dilakukan dapat dilihat pada Gambar 1, penelitian dimulai dengan melakukan pengumpulan data mengenai halaman yang memiliki domain masukan yang terdapat pada *website worker's*, domain masukan merupakan semua *input* yang terdapat

pada suatu *software* [17]. Pengumpulan data ini dilakukan dengan tujuan menentukan halaman yang akan dilakukan pengujian. Selanjutnya akan dikumpulkan data mengenai domain masukan yang terdapat pada *website worker's*, dimana data yang dikumpulkan berupa kondisi dari domain masukan yang ada dapat berupa kondisi valid atau tidak valid dari domain masukan tersebut. Kemudian, akan dibuat partisi–partisi berdasarkan kondisi yang dimiliki oleh domain masukan, dimana partisi merupakan gabungan dari kelas ekuivalensi [17]. Pembuatan partisi lalu dijadikan acuan dalam pembentukan *test case* untuk pengujian dengan menggabungkan partisi–partisi yang memiliki kondisi yang mirip atau sama, *test case* merupakan pasangan antara data *input* dan hasil *output* [17]. Pada *equivalence partitioning* terdapat panduan dalam pembuatan partisi – partisi yaitu [14]:

- a. Jika kondisi *input* menentukan rentang, satu kelas ekuivalensi valid dan dua tidak valid ditentukan
- b. Jika kondisi *input* membutuhkan nilai tertentu, satu kelas ekuivalensi valid dan dua tidak valid ditentukan
- c. Jika kondisi *input* menentukan anggota himpunan, satu kelas ekuivalensi yang valid dan satu tidak valid ditentukan
- d. Jika kondisi *input* adalah *Boolean*, satu kelas valid dan satu kelas tidak valid ditentukan

Dengan menerapkan panduan untuk mendapatkan kelas ekuivalensi, *test case* untuk setiap domain input dapat dikembangkan dan dijalankan. *Test case* dipilih agar jumlah atribut terbanyak dari kelas ekuivalensi dapat dijalankan sekaligus [14]. Berikut adalah tahap–tahap yang digunakan untuk memudahkan pembuatan partisi/kelas pada *equivalence partitioning* [17]:

- a. Mengidentifikasi domain input  
Tahap pertama dalam melakukan *equivalence partitioning* yaitu mengidentifikasi domain masukan, dimana dilakukan identifikasi terhadap semua variabel *input* dan *output*, tipenya, dan kondisi dari input serta outputnya.
- b. Pembagian kelas ekuivalensi  
Partisi dari himpunan nilai setiap variabel ke dalam subhimpunan yang terpisah–pisah. Dimana setiap subhimpunan merupakan kelas ekuivalensi. Partisi domain input menggunakan nilai dari variabel yang didasarkan dari hasil yang diharapkan. Nilai–nilai yang berperilaku sama akan dikelompokkan bersama.
- c. Menggabungkan kelas – kelas ekuivalensi  
Tahap ini jarang digunakan karena dapat menyebabkan jumlah kelas ekuivalensi yang besar dan tidak terkelola. Pada tahap ini dengan menggabungkan kelas ekuivalensi berdasarkan kondisi – kondisi dari variabel yang dapat digunakan untuk menentukan *test case*.
- d. Mengidentifikasi kelas ekuivalensi yang tidak mungkin  
Pada kelas ekuivalensi terdapat kombinasi data masukan yang tidak dapat dibuat saat melakukan pengujian. Kelas–kelas tersebut disebabkan beberapa alasan seperti data

masukannya dibatasi oleh *GUI (Graphical User Interface)*, dimana *GUI* dapat membatasi input yang dapat diterima dengan merekomendasikan *input* yang valid.

## 2.1 Jenis Data & Sumber Data

Pada penelitian ini, digunakan jenis data kualitatif dimana data tersebut didapatkan dari hasil observasi *website worker's*. Observasi dilakukan dengan mengakses setiap halaman web dari *website* untuk menentukan halaman yang memiliki domain masukan. Dimana hasil observasi berupa kualitas *website* yang diperoleh dari persentasi *test* yang *pass* dan *fail*. Sumber data pada penelitian ini yaitu sumber data primer berupa data dari *website worker's* yang diperoleh dari teknik observasi. Data didapatkan berupa halaman yang memiliki domain masukan.

## 2.2 Teknik Pengumpulan Data

Teknik yang digunakan dalam penelitian ini adalah observasi. Data-data yang dikumpulkan dengan teknik observasi yaitu data-data mengenai halaman pada *website worker's* yang memiliki domain masukan serta data mengenai domain masukan tersebut.

## 2.3 Teknik Pengolahan Data

Penelitian ini menggunakan teknik analisis data kualitatif untuk pengolahan datanya. Analisis data kualitatif merupakan teknik yang dilakukan dengan mereduksi data agar dapat membuat kesimpulan dari data, dimana data-data tersebut dibedakan berdasarkan konsep tertentu, kategori tertentu, dan tema tertentu [18].

Hasil yang didapatkan dari reduksi data tersebut kemudian diolah agar dapat ditampilkan datanya secara keseluruhan dimana dapat berbentuk sketsa, sinopsis, matriks, dan bentuk lainnya. Hal tersebut diperlukan agar memudahkan saat menguraikan dan menjelaskan kesimpulan [18]. Pada penelitian ini, analisis data kualitatif dilakukan pada data hasil pengujian untuk memperoleh kesimpulan jumlah *test case* yang berhasil dan gagal.

## 2.4 Penarikan Kesimpulan

Penarikan kesimpulan dalam penelitian ini diambil dari hasil pengujian *test case* dengan metode *equivalence partitioning*. Dari data domain masukan, diperoleh partisi-partisi yang dijadikan acuan untuk *test case*. Kemudian dilakukan pengujian terhadap *test case* tersebut agar dapat mengetahui apakah terdapat kesalahan pada *website worker's*.

## 3. HASIL DAN PEMBAHASAN

### 3.1 Pengumpulan Data Halaman dan Domain Masukan *Website Worker's*

Data yang digunakan dalam penelitian ini yaitu data mengenai domain masukan yang terdapat pada halaman *worker's*. Data tersebut kemudian dijadikan acuan untuk melakukan pengujian *black box testing* dengan metode *equivalence partitioning*. Pada *worker's* terdapat 10 halaman yang memiliki domain masukan, tetapi hanya menggunakan 8 halaman karena terdapat 4 halaman yang memiliki domain masukan yang sama.

Tabel 1: Data Halaman dan Domain Masukan

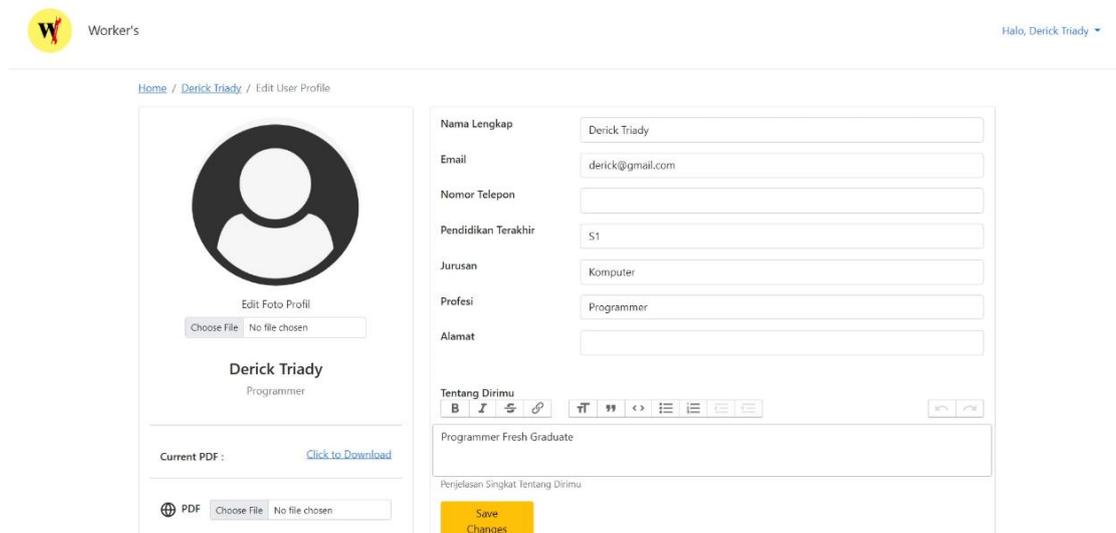
No	Halaman	Domain Masukan
1	Halaman <i>Login</i>	<i>Email, Password</i>
2	Halaman <i>Home</i>	<i>Search</i>
3	Halaman Register <i>JobSeeker</i>	Nama Lengkap, Jenis Kelamin, <i>Username, Email, Password, Ulang Password, Gambar Profile, Profesi, Pendidikan Terakhir, Jurusan, Tentang Kamu</i>
4	Halaman <i>Edit Profile JobSeeker</i>	Nama Lengkap, Alamat, Nomor Telepon, Profesi, Jurusan, Pendidikan Terakhir, Profesi, Keterangan, Gambar <i>Profile, CV</i>
5	Halaman Register <i>Company</i>	Nama Usaha, Kategori Perusahaan, <i>Username, Email, Password, Ulang Password, Gambar Perusahaan, Keterangan Perusahaan</i>
6	Halaman <i>Edit Profile Company</i>	Nama Usaha, <i>Email, Kategori Perusahaan, Alamat, Nomor Telepon, Website, Twitter, Instagram, Facebook, Google Maps, Keterangan Perusahaan, Gambar Perusahaan</i>
7	Halaman <i>Job Vacancy</i>	Nama Lowongan, Alamat, Bidang, Jurusan, Jenis Kerja, Minimal Pendidikan, Batas Post, Gaji Minimal, Gaji Maksimal, Deskripsi
8	Halaman <i>Interview</i>	Lowongan, Jadwal <i>Interview, Tipe Interview, Lokasi Interview, Catatan</i>

Berdasarkan data pada Tabel 1, maka dihasilkan partisi berdasarkan domain masukan yang ada. Partisi dibuat berdasarkan kondisi yang dimiliki oleh domain masukan.

### 3.2 Pembuatan Partisi dan Test Case

Pengujian dimulai setelah membagi partisi dari domain masukan pada halaman yang telah ditentukan sesuai dengan data pada Tabel 1, dari partisi – partisi tersebut kemudian digunakan untuk menjadi acuan pembuatan *test case*.

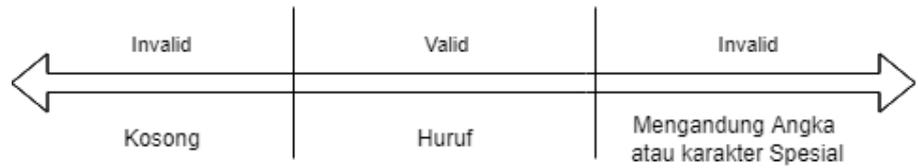
#### 1. Halaman *Edit Profile JobSeeker*



Gambar 2. Halaman *Edit Profile JobSeeker*

Halaman ini berfungsi sebagai tempat *edit profile* bagi *user jobseeker*. Pada halaman ini terdapat 10 domain masukan. Partisi yang dapat dibuat berdasarkan domain masukan tersebut yaitu:

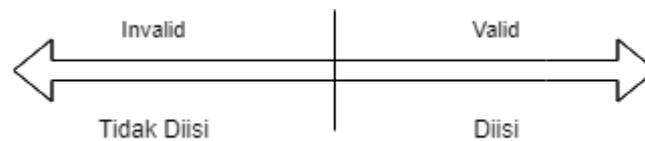
a. Nama Lengkap



Gambar 3. Partisi Nama Lengkap

Pada Gambar 3, nama lengkap memiliki 3 partisi di mana nama lengkap akan valid jika menggunakan input huruf, dan invalid jika kosong atau mengandung angka/karakter spesial.

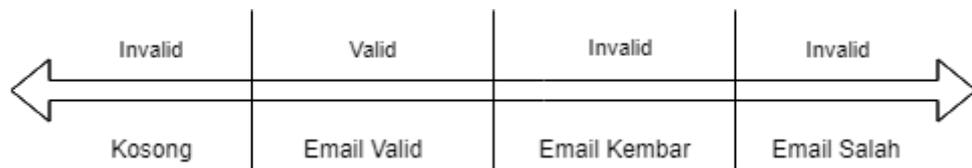
b. Alamat



Gambar 4. Partisi Alamat

Pada Gambar 4, alamat memiliki 2 partisi di mana alamat akan valid apabila diisi, dan alamat invalid apabila kosong.

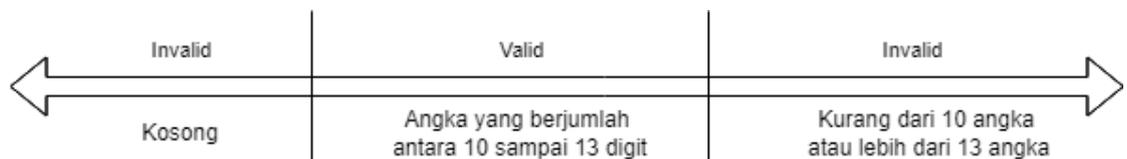
c. Email



Gambar 5. Partisi Email

Pada Gambar 5, email memiliki 4 partisi di mana email akan valid jika menggunakan format email yang valid, dan invalid apabila kosong, email sudah digunakan, dan jika format emailnya salah.

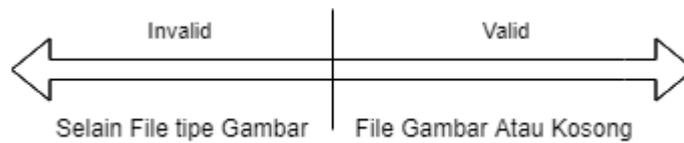
d. Nomor Telepon



Gambar 6. Partisi Nomor Telepon

Pada Gambar 6, nomor telepon memiliki 3 partisi di mana nomor telepon akan valid jika menggunakan angka dengan 10-13 digit, dan invalid jika kosong. Pada partisi ini tidak dapat memasukkan nilai huruf atau karakter special karena telah dibatasi sehingga hanya dapat menginput angka dan terdapat pesan error apabila diinput selain angka.

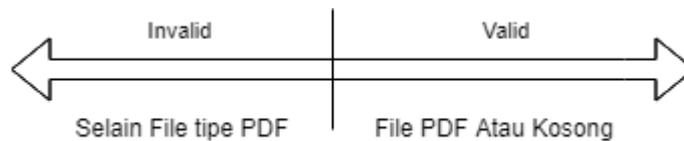
e. Gambar *Profile*



Gambar 7. Partisi Gambar *Profile*

Pada Gambar 7, gambar *profile* memiliki 2 partisi di mana gambar *profile* akan valid jika menggunakan *file* gambar atau kosong, dan gambar *profile* invalid apabila menggunakan selain file format gambar.

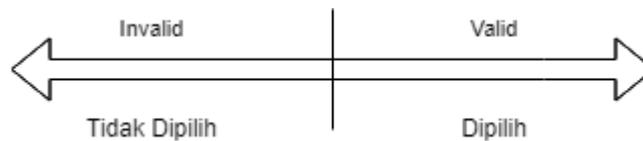
f. CV



Gambar 8. Partisi CV

Pada Gambar 8, CV memiliki 2 partisi. CV akan valid jika menggunakan *file* PDF atau kosong, dan invalid jika menggunakan selain file format PDF.

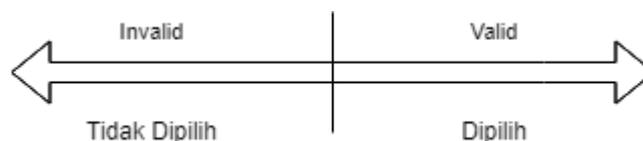
g. Profesi



Gambar 9. Partisi Profesi

Pada Gambar 9, profesi memiliki 2 partisi dimana, profesi akan valid jika memilih profesi, dan invalid jika tidak memilih atau memilih “Profesi Kamu”.

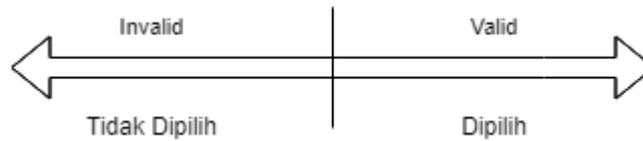
h. Pendidikan Terakhir



Gambar 10. Partisi Pendidikan Terakhir

Pada Gambar 10, pendidikan terakhir memiliki 2 partisi di mana pendidikan terakhir akan valid jika memilih pendidikan terakhir, dan invalid jika tidak memilih atau memilih “Pendidikan Terakhir Kamu”.

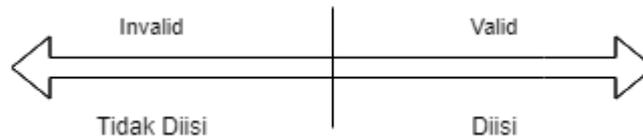
i. Jurusan



Gambar 11. Partisi Jurusan

Pada Gambar 11, jurusan memiliki 2 partisi di mana jurusan akan valid jika memilih jurusan, dan invalid jika tidak memilih atau memilih "Jurusan Kamu".

j. Tentang Kamu



Gambar 12. Partisi Tentang Kamu

Pada Gambar 12 Tentang Kamu memiliki 2 partisi di mana tentang kamu akan valid apabila diisi, dan tentang kamu invalid apabila kosong.

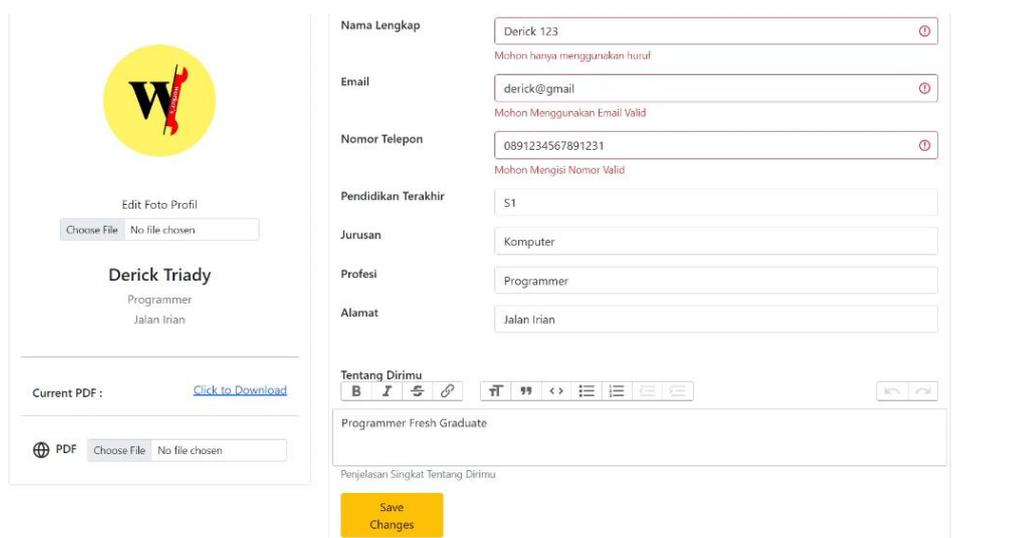
Berdasarkan partisi Gambar 3 sampai Gambar 12 yang telah dibuat dihasilkan *test case* yang dapat dilihat pada Tabel 2. Berdasarkan *test case* tersebut maka akan dilakukan pengujian, dan hasil dari *test case* halaman *edit profile jobseeker* dapat dilihat pada pada Tabel 2.

Tabel 2: *Test Case* dan Hasil Pengujian Halaman Edit *Profile JobSeeker*

Test Case ID	Test Case	Test Steps	Test Data	Expected Result	Actual Results	Pass/Fail
W-PJ_01	Mengisi semua input di form	1. Membuka Halaman Profil 2. Menekan Tombol Edit 3. Mengisi Inputan yang kosong pada Halaman 4. Menekan Save Changes	1. Nama Lengkap: Derick Triady 2. Email: derick@gmail.com 3. Nomor Telepon: 089123456789 4. Pendidikan Terakhir: S1 5. Jurusan: Komputer 6. Profesi: Programmer 7. Alamat: Jalan Irian 8. Foto Profil: Logo.png 9. CV: C.V.pdf 10. Tentang Dirimu: Programmer Fresh Graduate	Berhasil Mengedit <i>Profile</i>	Berhasil mengedit <i>profile</i>	Pass
W-PJ_02	Mengubah atau mengisi input dengan input yang salah atau tidak valid	1. Membuka Halaman Profil 2. Menekan Tombol Edit 3. Mengisi atau mengganti Inputan menjadi yang invalid pada Halaman 4. Menekan Save Changes	1. Nama Lengkap: Derick 123 2. Email: derick@gmail 3. Nomor Telepon: 0891234567891231 4. Pendidikan Terakhir: S1 5. Jurusan: Komputer 6. Profesi: Programmer 7. Alamat: Jalan Irian 8. Foto Profil: FlowChart.pdf 9. CV: 150529.png 10. Tentang Dirimu: Programmer Fresh Graduate	Menolak Edit, muncul pesan mengenai input yang salah atau invalid	Tidak Berhasil Edit dan muncul pesan pada Nama, email dan nomor bahwa input perlu input yang valid tetapi tidak muncul pesan <i>error</i>	Fail

Test Case ID	Test Case	Test Steps	Test Data	Expected Result	Actual Results	Pass/Fail
					pada Gambar profile dan CV	
W-PJ_03	Mengosongkan Data Profile Sebelum Menekan Simpan	<ol style="list-style-type: none"> <li>1. Membuka Halaman Profil</li> <li>2. Menekan Tombol Edit</li> <li>3. Mengosongkan Inputan pada Halaman</li> <li>4. Menekan Save Changes</li> </ol>	<ol style="list-style-type: none"> <li>1. Nama Lengkap: Kosong</li> <li>2. Email: Kosong</li> <li>3. Nomor Telepon: Kosong</li> <li>4. Pendidikan Terakhir: Pendidikan Terakhir Kamu</li> <li>5. Jurusan: Jurusan Kamu</li> <li>6. Profesi: Profesi Kamu</li> <li>7. Alamat: Kosong</li> <li>8. Foto Profil: Kosong</li> <li>9. CV: Kosong</li> <li>10. Tentang Dirimu: Kosong</li> </ol>	Menolak Edit, muncul pesan bahwa input tidak boleh kosong	Tidak Berhasil Edit dan muncul pesan untuk mengisi inputan	Pass
W-PJ_04	Mencoba untuk menggunakan email lain yang sudah dipakai	<ol style="list-style-type: none"> <li>1. Membuka Halaman Profil</li> <li>2. Menekan Tombol Edit</li> <li>3. Mengganti Email menjadi email lain yang sudah dipakai</li> <li>4. Menekan Save Changes</li> </ol>	<ol style="list-style-type: none"> <li>1. Nama Lengkap: Derick Triady</li> <li>2. Email: taman@gmail.com</li> <li>3. Nomor Telepon: 089123456789</li> <li>4. Pendidikan Terakhir: S1</li> <li>5. Jurusan: Komputer</li> <li>6. Profesi: Programmer</li> <li>7. Alamat: Jalan Irian</li> <li>8. Foto Profil: Kosong</li> <li>9. CV: Kosong</li> <li>10. Tentang Dirimu: Programmer Fresh Graduate</li> </ol>	Tidak Berhasil Edit Muncul Pesan bahwa email telah digunakan	Tidak Berhasil Edit Muncul Pesan Bahwa Email Sudah Terdaftar	Pass

Berdasarkan pengujian yang telah dilakukan, dari total 4 test case yang ada, terdapat 3 test case yang pass dan 1 test case yang fail. Test case dinyatakan pass apabila actual result sesuai dengan expected result, dan dinyatakan fail apabila actual result tidak sesuai dengan expected result. Jika test case fail, maka akan dilakukan perbaikan.



Gambar 13. Hasil Pengujian Test Case W-PJ\_02

The screenshot shows a user profile interface. On the left, there is a profile picture placeholder for 'Derick Triady', a programmer from 'Jalan Irian'. Below it, there are options to edit the profile picture and upload a PDF CV. The main part of the page contains a form with the following fields:
 

- Nama Lengkap:** Derick 123 (with a note: 'Mohon hanya menggunakan huruf')
- Email:** derick@gmail (with a note: 'Mohon Menggunakan Email Valid')
- Nomor Telepon:** 0891234567891231 (with a note: 'Mohon Mengisi Nomor Valid')
- Pendidikan Terakhir:** S1
- Jurusan:** Komputer
- Profesi:** Programmer
- Alamat:** Jalan Irian

 At the bottom, there is a 'Tentang Dirimu' section with a rich text editor containing the text 'Programmer Fresh Graduate' and a 'Save Changes' button.

Gambar 14. Hasil Perbaikan *Test Case* W-PJ\_02

Berdasarkan hasil pengujian pada Gambar 13, dapat dilihat *test case* W-PJ\_02 dinyatakan *fail* karena tidak menampilkan semua pesan *error* yang seharusnya ditampilkan. Perbaikan pada *test case* W-PJ\_02 dapat dilihat pada Gambar 14 dimana semua pesan *error* berhasil tampil.

Setelah melakukan pengujian terhadap semua *test case*, dibuat ringkasan hasil *test case* secara keseluruhan pada Tabel 3.

Tabel 3: Ringkasan Keseluruhan *Test Case*

Test Case ID	Test Case	Expected Result	Actual Results	Pass/Fail
W-L_01	Menguji <i>Login Worker's</i> dengan <i>Email</i> dan <i>Password Valid</i>	Berhasil <i>Login</i> dan Masuk ke <i>Home Worker's</i>	Berhasil <i>Login</i> Dan membuka halaman <i>home</i> pada <i>Worker's</i>	Pass
W-L_02	Menguji <i>Login Worker's</i> dengan <i>Email</i> dan <i>Password Salah</i>	Gagal <i>Login</i> , menampilkan pesan <i>Email</i> atau <i>Password</i> salah	Tidak berhasil <i>login</i> dan menampilkan pesan <i>Email</i> atau <i>Password</i> Salah	Pass
W-L_03	Menekan <i>Login</i> Tanpa Mengisi Form	Menampilkan pesan untuk mengisi <i>Email</i> dan <i>Password</i>	Tidak berhasil <i>login</i> dan menampilkan pesan untuk mengisi <i>Email</i> dan <i>Password</i>	Pass
W-S_01	Menguji Search <i>Worker's</i> dengan mengisi nama lowongan	Berhasil menemukan lowongan	Lowongan dengan nama <i>Programmer</i> berhasil didapat	Pass
W-S_02	Menguji Search <i>Worker's</i> dengan mengisi nama lowongan yang salah atau lowongan yang tidak ada	Tidak mendapatkan lowongan	Tidak Menampilkan Lowongan Apapun	Pass
W-S_03	Menguji Search <i>Worker's</i> dengan tanpa mengisi nama lowongan	Menampilkan semua lowongan	Menampilkan Semua lowongan yang ada	Pass
...	...	...	...	...
W-CJ_01	Mengisi semua input valid pada input kosong yang diperlukan pada form	Berhasil membuat Lowongan, dan di <i>redirect</i> kembali ke halaman <i>dashboard</i> lowongan	Lowongan Berhasil Dibuat dan Di- <i>redirect</i> Ke halaman <i>dashboard</i> lowongan, Tetapi tidak menampilkan alamat di <i>dashboard</i>	Fail

Test Case ID	Test Case	Expected Result	Actual Results	Pass/ Fail
W-CJ_02	Mengisi semua input menjadi invalid pada input yang ada pada form	Gagal untuk membuat lowongan serta muncul pesan <i>error</i> mengenai inputan yang salah	Lowongan Gagal Dibuat dan muncul pesan <i>error</i> pada inputan yang salah atau invalid	Pass
W-CJ_03	Menekan Tombol Save tanpa mengisi	Tidak bisa membuat Lowongan dan muncul pesan untuk mengisi inputan	Tidak dapat membuat lowongan dan muncul pesan pada inputan yang diperlukan untuk diisi	Pass
W-CJ_04	Mengisi gaji minimal diatas gaji maksimal dan gaji maksimal dibawah gaji minimal	Gagal tersimpan Muncul pesan bahwa gaji minimal harus dibawa gaji maksimal dan gaji maksimal harus diatas gaji minimal	Lowongan tidak berhasil dibuat dan muncul pesan untuk mengisi Gaji minimal diatas gaji maksimal dan sebaliknya	Pass
W-I_01	Mengisi semua input valid pada input kosong yang diperlukan pada form	Berhasil Membuat Jadwal <i>interview</i> dan diarahkan kembali ke <i>Dashboard</i> halaman <i>interview</i>	Jadwal <i>interview</i> Berhasil Dibuat dan diarahkan ke <i>Dashboard</i> Halaman <i>interview</i>	Pass
W-I_02	Mengisi input invalid pada input Tanggal yang diperlukan pada form	Tidak Berhasil Membuat Jadwal <i>interview</i> , muncul pesan <i>error</i> pada input yang invalid	Jadwal <i>interview</i> tidak berhasil dibuat, muncul pesan <i>error</i> bahwa tanggal harus lewat dari tanggal hari ini	Pass
W-I_03	Mencoba membuat <i>interview</i> tanpa mengisi input pada halaman	Tidak Berhasil Membuat Jadwal <i>interview</i> , Muncul Pesan Bahwa Harus mengisi form	Jadwal <i>interview</i> tidak terbuat, muncul pesan bahwa harus mengisi input - input berikut	Pass

Berdasarkan keseluruhan hasil pengujian yang telah dilakukan pada Tabel 3 yang lengkapnya dapat dilihat pada lampiran, terdapat total 29 *test case* yang dihasilkan dari 54 partisi. Sehingga menghasilkan hasil seperti pada Tabel 4. Apabila menggunakan pengujian menggunakan *exhaustive testing* untuk inputan nomor telepon dengan panjang 10 digit, maka dapat menghasilkan jumlah *test case* sebesar  $10^{10} = 10.000.000.000$  *test case*. Berdasarkan contoh tersebut maka dapat dikatakan bahwa *test case* yang dihasilkan dari metode *equivalence partitioning* jauh lebih sedikit.

Tabel 4: Keterangan Hasil Pengujian

Keterangan	Jumlah
Pass	20
Fail	9



Gambar 15. Hasil Pengujian

Berdasarkan data dari Tabel 4 dan Gambar 15, dapat dilihat hasil pengujian terhadap *website worker's*. Dari total 29 *test case*, *test case* yang *pass* sebesar 68% dan *test case* yang *fail* sebesar 32%. *Test case* yang *fail* berjumlah 9 *test case*. *Test case* ini merupakan *test case* yang mengalami kegagalan saat melakukan pengujian, di mana 7 *test case* dinyatakan *fail* karena tidak menampilkan pesan *error* dan sisanya karena tidak menampilkan data yang seharusnya ditampilkan.

Apabila 9 *test case* tersebut tidak diperbaiki maka jika terjadi kesalahan, user tidak mengetahui kesalahan pada penginputan data, *website* tidak menampilkan gambar profile dan alamat lowongan. Untuk menangani hal tersebut, dilakukan perbaikan dan pengujian ulang terhadap *test case* tersebut. Seperti yang dapat dilihat pada Gambar 13 di mana *test case* W-PJ\_02 yang memiliki kesalahan pada saat pengujian telah diperbaiki. Hasil perbaikan dapat dilihat pada Gambar 14 dimana semua pesan *error* telah berhasil ditampilkan.

#### 4. KESIMPULAN

Pengujian black box dengan metode *equivalence partitioning* terhadap *website worker's* menggunakan metode *equivalence partitioning* dapat menghasilkan *test case* yang mampu mendapatkan *error* atau kesalahan pada fungsional *website worker's*. Dengan menggunakan data domain masukan dari halaman yang diuji, diperoleh 54 partisi, yang kemudian menghasilkan 29 *test case*. Setelah melakukan pengujian, terdapat 20 *test case* yang *pass* dan 9 *test case* yang *fail*. Oleh karena itu, dapat dikatakan bahwa terdapat 9 celah *error* pada *website worker's* yang perlu diperbaiki.

#### 5. SARAN

Penelitian ini dapat dikembangkan lebih lanjut dengan mengkombinasikan metode *equivalence partitioning* dengan metode *black box* lainnya, dan juga pada hasil fitur pencarian lowongan yang tidak ada disarankan untuk menampilkan pesan apabila tidak mendapatkan lowongan yang dicari. Apabila *software* memiliki domain masukan dan

kombinasi *output* dengan jumlah yang banyak dapat mempertimbangkan untuk menggunakan metode *equivalence partitioning* dalam merancang *test case* pengujian *black box*.

#### DAFTAR PUSTAKA

- [1] A. Agustian, I. Andryani, S. Khoerunisa, A. Pangestu, and A. Saifudin, "Implementasi Teknik Equivalence Partitioning pada Pengujian Aplikasi E-learning Berbasis Web," *J. Teknol. Sist. Inf. dan Apl.*, vol. 3, no. 3, p. 178, 2020, doi: 10.32493/jtsi.v3i3.5371.
- [2] M. Komarudin, "Pengujian Perangkat Lunak Metode Black-Box Berbasis Equivalence Partitions pada Aplikasi Sistem Informasi di Sekolah," *J. Mikrotik*, vol. 06, no. 3, pp. 02–16, 2016.
- [3] Wahyono, "10 Kesalahan Sistem Komputer Paling Fatal di Dunia, Apa Saja?," 2020. <https://tekno.sindonews.com/read/87476/207/10-kesalahan-sistem-komputer-paling-fatal-di-dunia-apa-saja-1593605272?showpage=all>.
- [4] M. Nurudin, W. Jayanti, R. D. Saputro, M. P. Saputra, and Y. Yulianti, "Pengujian Black Box pada Aplikasi Penjualan Berbasis Web Menggunakan Teknik Boundary Value Analysis," *J. Inform. Univ. Pamulang*, vol. 4, no. 4, p. 143, 2019, doi: 10.32493/informatika.v4i4.3841.
- [5] A. Amalia, S. W. Putri Hamidah, and T. Kristanto, "Pengujian Black Box Menggunakan Teknik Equivalence Partitions Pada Aplikasi E-Learning Berbasis Web," *Build. Informatics, Technol. Sci.*, vol. 3, no. 3, pp. 269–274, 2021, doi: 10.47065/bits.v3i3.1062.
- [6] E. S. Eriana, "Pengujian Sistem Informasi Aplikasi Perpustakaan Berbasis Web Dengan White Box Testing," *J. Teknol. Inf. ESIT*, vol. XV, no. 10, pp. 28–33, 2020.
- [7] D. Widhyaestoeti, S. Iqram, S. N. Mutiyah, and Y. Khairunnisa, "Black Box Testing Equivalence Partitions Untuk Pengujian Front-End Pada Sistem Akademik Sitoda," *J. Ilm. Teknol. Infomasi Terap.*, vol. 7, no. 3, pp. 211–216, 2021, doi: 10.33197/jitter.vol7.iss3.2021.626.
- [8] N. W. Rahadi and C. Vikasari, "Pengujian Software Aplikasi Perawatan Barang Milik Negara Menggunakan Metode Black Box Testing Equivalence Partitions," *Infotekmesin*, vol. 11, no. 1, pp. 57–61, 2020, doi: 10.35970/infotekmesin.v11i1.124.
- [9] F. S. Kristara, G. Kanuraga, Rohmat, D. Yansah, A. Saifudin, and Yulianti, "Pengujian Kualitas Aplikasi Web E-Learning Universitas Pamulang Menggunakan Metode Black Box," *J. Inform. Univ. Pamulang*, vol. 6, no. 2, pp. 225–231, 2021.
- [10] Ardiyansah, I. Munandar, D. Wantara, M. Sugianto, and U. Ardiansyah, "PENGUJIAN BLACKBOX DENGAN METODE CAUSE EFFECT GRAPHING PADA SISTEM

- INFORMASI PERPUSTAKAAN,” vol. 3, pp. 36–44, 2022.
- [11] L. Lazic and N. Mastorakis, “Orthogonal Array application for optimal combination of software defect detection techniques choices,” *WSEAS Trans. Comput.*, vol. 7, no. 8, pp. 1319–1336, 2008.
- [12] Hidayatulhidayah, S. Widowati, and D. Richasdy, “Modifikasi pada Pembangkit Kasus Uji untuk Pengujian Kombinasi Modification on Test Case Generator for Pairwise Testing,” vol. 8, no. 5, pp. 10717–10727, 2021.
- [13] A. R. Baktiar, D. Mulainsyah, E. C. Sasgoro, and E. Sumiati, “Pengujian Menggunakan Black Box Testing dengan Teknik State Transition Testing Pada Perpustakaan Yayasan Pendidikan Islam Pakualam Berbasis Web,” *J. Kreat. Mhs. Inform.*, vol. 2, no. 1, pp. 142–145, 2021.
- [14] S. Pressman, Roger and B. Maxim, *Software Engineering: A Practitioner’s Approach*, 8th ed. McGraw Hill Education, 2014.
- [15] R. Pramudita, “Pengujian Black Box pada Aplikasi Ecampus Menggunakan Metode Equivalence Partitioning,” *INFORMATICS Educ. Prof. J. Informatics*, vol. 4, no. 2, p. 193, 2020, doi: 10.51211/itbi.v4i2.1347.
- [16] N. Kumawat, Y. Sharma, and U. Pilania, “Comparative Study between Equivalence Class Partitioning and Boundary Value Analysis Testing Methods,” vol. 2, no. 10, pp. 74–78, 2016.
- [17] S. Mathur, Aditya, *Foundations of software testing*, 2nd ed. Pearson, 2014.
- [18] A. Rijali, “Analisis Data Kualitatif,” vol. 17, no. 33, pp. 81–95, 2018.

## LAMPIRAN

### [Tabel 3. Ringkasan Keseluruhan Test Case](#)