

## Pengujian *Black Box* pada Aplikasi Sistem Penunjang Keputusan Seleksi Calon Pegawai Negeri Sipil Menggunakan Teknik *Equivalence Partitions*

Misbah Syah Anwar Kesuma Jaya<sup>1</sup>, Pandu Gumilang<sup>2</sup>, Tresnawati<sup>3</sup>,  
Yohanes Philipus Andersen<sup>4</sup>, dan Teti Desyani<sup>5</sup>

Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Banten, Indonesia  
E-mail: <sup>1</sup>misbah.masat03@gmail.com, <sup>2</sup>pandugumilang44@gmail.com,  
<sup>3</sup>tresnawatidc28@gmail.com, <sup>4</sup>yohanesandersen3321@gmail.com, <sup>5</sup>dosen00839@unpam.ac.id

Submitted Date: December 21<sup>st</sup>, 2019  
Revised Date: January 9<sup>th</sup>, 2020

Reviewed Date: December 30<sup>th</sup>, 2019  
Accepted Date: January 9<sup>th</sup>, 2020

### Abstract

*Support System for Decision on the Selection of Prospective Civil Servants is still an error when making changes to the data that has been entered, so that it can cause data can not be updated. So it needs to be tested to find problems with this software. Tests conducted on this software use the Black Box method based on Equivalence Partition as a whole with regard to the use, benefits, and results found from the use of the software. Equivalence Partition-based Black Box method tests the quality of the application that will be carried out by software testing documentation with the discovery of errors in each form that is divided into three error models, including errors in Functions, Data Structures and Interfaces. This test provides results where we will find out whether the software This has an error or not, and this test can guarantee the quality of this software.*

*Keywords: DSS, Black Box, Equivalence Partition, Testing*

### Abstrak

Sistem Penunjang Keputusan Seleksi Calon Pegawai Negeri Sipil ini masih terdapat kesalahan saat melakukan proses perubahan data yang telah dimasukkan, sehingga dapat menyebabkan data tidak dapat diperbarui. Sehingga perlu dilakukan pengujian untuk menemukan masalah pada perangkat lunak ini. Pengujian yang akan dilakukan pada perangkat lunak ini menggunakan metode *Black Box* berbasis *Equivalence Partition* secara menyeluruh berkenaan dengan penggunaan, manfaat, dan hasil yang didapati dari pemanfaatan perangkat lunak tersebut. Teknik *Equivalence Partition* merupakan sebuah pengujian berdasarkan masukkan data pada setiap *form* yang ada pada Sistem Seleksi Calon Pegawai Negeri Sipil, juga menguji kualitas aplikasi yang akan dilakukan dokumentasi pengujian perangkat lunak dengan ditemukannya kesalahan pada setiap form yang dibagi menjadi tiga model kesalahan, di antaranya kesalahan pada Fungsi, Struktur Data dan *Interface* Pengujian ini memberikan hasil dimana kita akan mengetahui apakah perangkat lunak ini memiliki kesalahan atau tidak, serta pengujian ini dapat menjamin kualitas dari perangkat lunak ini.

Kata Kunci: SPK, *Black Box*, *Equivalence Partition*, Pengujian

### 1. Pendahuluan

Pengujian merupakan bagian dari aktifitas pengembangan sistem yang terencana dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan dari *Software* yang kita kembangkan. Aktifitas pengujian mencakup pembuatan dan penerapan sekumpulan kasus uji (*test case*) yang spesifik untuk menjamin kualitas perangkat lunak bergantung kepada kepuasan

pelanggan (Cholifah, Yulianingsih, & Sagita, 2018).

Pengujian perangkat lunak ini merupakan proses penting untuk menilai *Software* dalam menentukan kualitasnya. Karena pengujian juga memberikan pandangan mengenai perangkat lunak secara obyektif dan independen, dan juga untuk melakukan ini dapat menghabiskan 40-50% dari upaya pengembangan, ini adalah bagian penting dari rekayasa perangkat lunak (Lu, 2014).

Pada aplikasi Sistem Penunjang Keputusan Seleksi Calon Pegawai Negeri Sipil ini merupakan cara penghitungan untuk menentukan calon pegawai yang terpilih. Jadi, Sistem Penunjang Keputusan yang dikembangkan bertujuan untuk mengeluarkan hasil yang dapat memberikan pertimbangan kepada Manajemen untuk menentukan calon pegawai yang terpilih ini menggunakan metode *Simple Additive Weighting* (SAW). Hasil penghitungan aplikasi Sistem Pemilihan Calon Pegawai Negeri Sipil ini tidak berbeda dengan hasil penghitungan matematis menggunakan aplikasi lainnya dan Database *Dekstop* sehingga aplikasi Sistem Pemilihan Calon Pegawai Negeri Sipil ini layak digunakan sebagai bahan pertimbangan dalam pengambilan keputusan. (Myers, Badgett, & Sandler, 2012)

Metode *Black box Testing* merupakan sebuah metode yang dipakai untuk menguji sebuah software tanpa harus memperhatikan detail *software*. Pada pengujian *black box*, estimasi banyaknya data uji dapat dihitung melalui banyaknya field data masukan yang akan diuji, aturan masukan yang harus dipenuhi serta batas masukan, baik batas atas maupun batas bawah yang memenuhi spesifikasi. Tidak ada upaya untuk mengetahui kode program apa yang output pakai. (Jaya, Pengujian Aplikasi dengan Metode Black Box)

*Equivalence Partitioning* merupakan sebuah pengujian berdasarkan masukkan data pada setiap *form*, metode pengujian yang memecah domain masukan dari program ke dalam kelas-kelas data sehingga *Test Case* dapat diperoleh. Perancangan *Test Case Equivalence* untuk kondisi masukan yang menggambarkan kumpulan keadaan yang valid atau tidak. (Maturidi, 2014). *Equivalence Partitioning* berdasarkan pada premis masukan dan keluaran dari suatu komponen yang dipartisi ke dalam kelas-kelas, menurut spesifikasi dari komponen tersebut, yang akan diperlakukan sama (ekivalen) oleh komponen tersebut. (Asnawati, 2015)

## 2. Metodologi

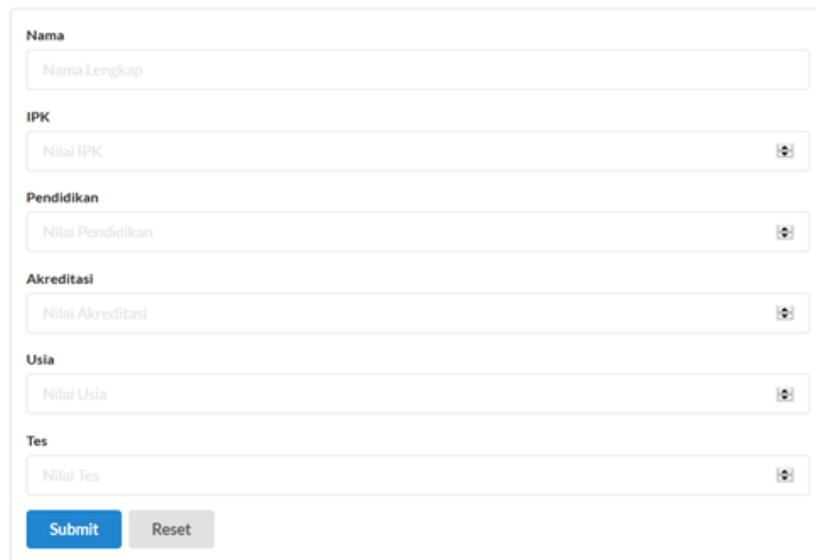
Pengujian software merupakan tahapan pengembangan yang sangat penting untuk memastikan *software* yang sudah/sedang dikembangkan dapat berjalan sesuai dengan kebutuhan yang telah ditetapkan. pengujian *software* untuk melakukan verifikasi dan validasi bahwa software yang telah dibuat sesuai dengan kebutuhan dari perusahaan. Pengujian merupakan bagian dari tahapan yang tidak dapat dipisahkan dari siklus hidup pengembangan software seperti halnya analisis, desain, dan pengkodean. (Shi, 2010)

Pengujian perangkat lunak sangat penting karena memiliki implikasi terhadap kualitas *software*. Pengujian tidak dapat dihindari karena pada aktivitas produksi memiliki peluang terjadinya kesalahan yang dibuat manusia sangat besar dan ketidakmampuan manusia untuk melakukan komunikasi dengan sempurna. Sehingga pengembangan perangkat lunak harus diiringi dengan aktivitas jaminan kualitas (Liana, 2015)

Pengujian menggunakan masukan data acak bertujuan untuk memastikan sistem menolak untuk menyimpan data masukan pada database, sehingga sistem dikatakan layak untuk digunakan. Dalam penelitian ini akan dilakukan beberapa tahapan. Pada tahapan pertama diawali dengan menentukan *Test Case* perangkat lunak yang akan diuji dengan metode *Equivalence Partitions* kemudian menginisialisasi standar grade partition masukan dan keluaran. Hal ini dilakukan untuk mendapatkan dataset berupa dok umentasi pengujian dengan metode *Equivalence Partitions* dan nilai tingkat efektifitas metode *Equivalence Partitions*.

Pada hasil pengujian terdapat tabel *Test Case* yang berfungsi untuk menyimpulkan apakah sistem berhasil dalam pengujian tipe tersebut atau tidak. Berikut terdapat penjelasan dari tabel *Test Case*.

### Input Master Data Calon Pegawai



Gambar 1. Form Tambah Data Calon Pegawai Negeri Sipil

Dari form pada Gambar 1 ada beberapa rencana pengujian. Form *Nama* diisi dengan huruf dan angka dan tidak boleh kosong, form *IPK* diisi dengan angka 1-100 dan tidak boleh kosong, form *Pendidikan* diisi dengan angka 1-100 dan tidak boleh kosong, form *Akreditasi* diisi dengan angka 1-100 dan tidak boleh kosong, form *Usia* diisi

dengan angka 1-100 dan tidak boleh kosong, form *Tes* diisi dengan angka 1-100 dan tidak boleh kosong, apabila data form *IPK* sampai *Tes* diisi dengan huruf akan muncul pesan “Mohon diisi dengan angka” begitupun dengan form *Nama* apabila tidak diisi maka data tidak dapat tersimpan.

Tabel 1. Rancangan Test Case Form Tambah Data Calon Pegawai Negeri Sipil

ID	Deskripsi Pengujian	Hasil yang diharapkan
TC01	Mengisi <i>Nama</i> dengan “Johan”, <i>IPK</i> diisi dengan “60”, <i>Pendidikan</i> diisi dengan “70”, <i>Akreditasi</i> diisi dengan “80”, <i>Usia</i> diisi dengan “90”, dan <i>Tes</i> diisi dengan “100”. Klik tombol Submit.	Sistem mampu menyimpan data ke dalam database dan halaman utama menampilkan data yang sudah dimasukkan
TC02	Mengisi <i>Nama</i> dengan “Johan”, <i>IPK</i> diisi dengan “ab”, <i>Pendidikan</i> diisi dengan “cd”, <i>Akreditasi</i> diisi dengan “ef”, <i>Usia</i> diisi dengan “gh”, dan <i>Tes</i> diisi dengan “ij” atau tidak diisi selanjutnya klik tombol Submit.	Sistem menolak untuk menyimpan data dan menampilkan pesan peringatan mohon data diisi dengan angka

### Edit Master Data Calon Pegawai

Gambar 2. Form Edit Calon Pegawai Negeri Sipil

Dari form pada Gambar 2 ada beberapa rencana pengujian. Form *Nama* diisi dengan huruf dan angka dan tidak boleh kosong, form *IPK* diisi dengan angka 1-100 dan tidak boleh kosong, form *Pendidikan* diisi dengan angka 1-100 dan tidak boleh kosong, form *Akreditasi* diisi dengan angka 1-100 dan tidak boleh kosong,

form *Usia* diisi dengan angka 1-100 dan tidak boleh kosong, form *Tes* diisi dengan angka 1-100 dan tidak boleh kosong, apabila data form *IPK* sampai *Tes* diisi dengan huruf akan muncul pesan “Mohon diisi dengan angka” begitupun dengan form *Nama* apabila tidak diisi maka data tidak dapat tersimpan.

Tabel 2. Rancangan Test Case Tabel Edit Calon Pegawai Negeri Sipil

ID	Deskripsi Pengujian	Hasil yang diharapkan
TD01	Mengisi <i>Nama</i> dengan “Mbah”, <i>IPK</i> diisi dengan “100”, <i>Pendidikan</i> diisi dengan “90”, <i>Akreditasi</i> diisi dengan “80”, <i>Usia</i> diisi dengan “70”, dan <i>Tes</i> diisi dengan “60”. Klik tombol Submit.	Sistem mampu menyimpan data ke dalam database dan mengalami perubahan tampilan pada halaman utama
TD02	Mengisi <i>Nama</i> dengan “Mbah” atau tidak diisi, <i>IPK</i> diisi dengan “ab”, <i>Pendidikan</i> diisi dengan “cd”, <i>Akreditasi</i> diisi dengan “ef”, <i>Usia</i> diisi dengan “gh”, dan <i>Tes</i> diisi dengan “ij” atau tidak diisi selanjutnya klik tombol Submit.	Sistem menolak untuk menyimpan data dan menampilkan pesan peringatan mohon data diisi dengan angka

### 3. Hasil dan Pembahasan

Setelah membuat rancangan pengujian, selanjutnya diterapkan dengan melaksanakan pengujian. Pengujian dilaksanakan dengan mengikuti deskripsi pengujian. Hasilnya kemudian dicatat di kolom hasil pengujian. Kesimpulan dibuat berdasarkan penilaian kesesuaian antara hasil pengujian dengan hasil

yang diharapkan. Jika hasil pengujian sesuai harapan maka disimpulkan Sesuai/Berhasil. Jika hasil pengujian tidak sesuai hasil yang diharapkan maka disimpulkan Salah/Gagal. Hasil pengujian yang telah dilakukan ditunjukkan pada Tabel 3.

Tabel 3 Hasil Pengujian

ID	Deskripsi Pengujian	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
TC01	Mengisi <i>Nama</i> dengan “Johan”, <i>IPK</i> diisi dengan “60”, <i>Pendidikan</i> diisi dengan “70”, <i>Akreditasi</i> diisi dengan “80”, <i>Usia</i> diisi dengan “90”, dan <i>Tes</i> diisi dengan “100”. Klik tombol Submit.	Sistem mampu menyimpan data ke dalam database dan halaman utama menampilkan data yang sudah dimasukkan	Tampilan halaman utama dengan data yang sudah dimasukkan	Sesuai
TC02	Mengisi <i>Nama</i> dengan “Johan”, <i>IPK</i> diisi dengan “ab”, <i>Pendidikan</i> diisi dengan “cd”, <i>Akreditasi</i> diisi dengan “ef”, <i>Usia</i> diisi dengan “gh”, dan <i>Tes</i> diisi dengan “ij” atau tidak diisi selanjutnya klik tombol Submit.	Sistem menolak untuk menyimpan data dan menampilkan pesan peringatan mohon data diisi dengan angka	Tampilan pesan peringatan mohon data diisi dengan angka	Sesuai
TD01	Mengisi <i>Nama</i> dengan “Mbah”, <i>IPK</i> diisi dengan “100”, <i>Pendidikan</i> diisi dengan “90”, <i>Akreditasi</i> diisi dengan “80”, <i>Usia</i> diisi dengan “70”, dan <i>Tes</i> diisi dengan “60”. Klik tombol Submit.	Sistem mampu menyimpan data ke dalam database dan mengalami perubahan tampilan pada halaman utama	Tampilan halaman utama dengan data yang sudah dimasukkan	Sesuai
TD02	Mengisi <i>Nama</i> dengan “Mbah” atau tidak diisi, <i>IPK</i> diisi dengan “ab”, <i>Pendidikan</i> diisi dengan “cd”, <i>Akreditasi</i> diisi dengan “ef”, <i>Usia</i> diisi dengan “gh”, dan <i>Tes</i> diisi dengan “ij” atau tidak diisi selanjutnya klik tombol Submit.	Sistem menolak untuk menyimpan data dan menampilkan pesan peringatan mohon data diisi dengan angka	Tampilan pesan peringatan mohon data diisi dengan angka	Sesuai

#### 4. Kesimpulan

Berdasarkan pada pembahasan yang telah dibahas sebelumnya, dapat diambil kesimpulan bahwa pengujian perangkat lunak menggunakan metode *Black Box* berfokus pada masukan dan keluaran dapat menginformasikan kesesuaian perangkat lunak yang dikembangkan dengan spesifikasi sistem yang ditetapkan. Teknik *Equivalence Partitions* dapat membantu proses pembuatan case pengujian dan menentukan kualitas serta menemukan kesalahan yang ada, selain itu setelah melakukan pengujian diketahui bahwa aplikasi dapat berjalan dengan baik meskipun data yang dimasukkan sudah diubah, akan tetapi perubahan data belum ternormalisasi bobotnya. Perlu dilakukan 2 langkah untuk data dapat dinormalisasi.

#### 5. Saran

Berdasarkan proses pengujian dan hasil pembahasan pengujian perangkat lunak Sistem Penunjang Keputusan Pemilihan Calon Pegawai Negeri Sipil diharapkan melakukan pembuatan spesifikasi secara detail dan didokumentasikan

secara tertulis dalam bentuk laporan serta pemberian keterangan setiap *input* yang ada. Laporan tersebut dapat menjadi bahan penelitian serta pengembangan sistem ke depannya.

#### Referensi

- Asnawati, F. H. (2015). *Rekayasa Perangkat lunak*. Yogyakarta: Deepublish.
- Cholifah, W. N., Yulianingsih, & Sagita, S. M. (2018). *Pengujian Black Box Testing Pada Aplikasi Action & Strategy Berbasis Android Dengan Teknologi PhoneGap*. Jakarta Selatan: Jurnal String.
- Jaya, T. S. (n.d.). *Pengujian Aplikasi dengan Metode Black Box*.
- Liana, L. (2015). Pengujian Perangkat Lunak. *Jurnal Rekayasa Perangkat Lunak*, 2.
- Lu, L. (2014). *Software Testing Techniques*. USA: Carnegie Mellon University.
- Maturidi, A. D. (2014). *Metode penelitian Teknik Informatika*. Yogyakarta: Deepublish.
- Myers, G. J., Badgett, T., & Sandler, C. (2012). *The Art of Software Testing*. Hoboken: John Wiley & Sons, Inc.

Shi, M. (2010). Software Functional Testing from the Perspective of Business Practice. *Journal Computer and Information Science*, 3.