

## Pengujian *Black Box* pada Sistem Aplikasi Informasi Data Kinerja Menggunakan Teknik *Equivalence Partitions*

Andri Ricat Sinulingga<sup>1</sup>, Muhammad Zuhri<sup>2</sup>, Rizky Budi Mukti<sup>3</sup>, Ziasyifa<sup>4</sup>, Aries Saifudin<sup>5</sup>

Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia  
<sup>1</sup>andrircat86@gmail.com, <sup>2</sup>m.zuhri13@gmail.com, <sup>3</sup>rizkybudi86@gmail.com,  
<sup>4</sup>ziasyifa790@gmail.com <sup>5</sup>aries.saifudin@unpam.ac.id

Submitted Date: January 30<sup>th</sup>, 2020

Reviewed Date: January 30<sup>th</sup>, 2020

Revised Date: January 31<sup>st</sup>, 2020

Accepted Date: January 31<sup>st</sup>, 2020

### **Abstract**

*Software that contains defects (errors) can cause harm. Losses caused can affect all stakeholders. Losses can take the form of financial, time, and others. So before the software is implemented it needs to be tested. Testing on an application aims to check whether a program is running properly or not. In testing this software will use Blackbox Testing. Black Box Testing Method consists of several techniques, including Equivalence Partitions, Boundary Value Analysis, Sample Testing, and so on. Among the many testing techniques, in this study the testing technique was chosen using Equivalence Partitions. Equivalence Partitions technique is a test based on data input on each form in the performance data information application system, each menu input will be tested and grouped according to function whether it is valid or invalid. Testing is done on 4 forms that will be filled with data form added jobdesk, form added work targets, form edit work targets, form approve work targets, and employee performance while in the company. The test results show that there are still many shortcomings when validating data to be entered, so that it can cause data stored in the database to be incompatible with the expected data. The results of this test can be used as input to improve the application of performance data for the future.*

*Keyword: Application, Testing, Black Box, Equivalence Partitions*

### **Abstrak**

Perangkat lunak yang mengandung cacat (kesalahan) dapat menimbulkan kerugian. Kerugian yang diakibatkan dapat berdampak pada semua stakeholder. Kerugian dapat berbentuk finansial, waktu, dan lain-lain. Sehingga sebelum perangkat lunak diimplementasikan perlu dilakukan pengujian. Pengujian pada suatu aplikasi bertujuan untuk memeriksa apakah sebuah program telah berjalan dengan semestinya atau tidak. Pada pengujian perangkat lunak ini akan menggunakan *Blackbox Testing*. Metode *Black Box Testing* terdiri atas beberapa teknik, antara lain *Equivalence Partitions*, *Boundary Value Analysis*, *Sample Testing*, dan sebagainya. Di antara banyak teknik pengujian tersebut, pada penelitian ini dipilih teknik pengujian menggunakan *Equivalence Partitions*. Teknik *Equivalence Partitions* merupakan sebuah pengujian berdasarkan masukan data pada setiap form yang ada pada sistem aplikasi informasi data kinerja, setiap menu masukan akan dilakukan pengujian dan dikelompokkan berdasarkan fungsinya baik itu bernilai valid ataupun tidak valid. Pengujian dilakukan pada 4 form yang akan diisi dengan data form tambah *jobdesk*, form tambah target kerja, form edit target kerja, form approve target kerja, serta kinerja karyawan selama di perusahaan. Hasil pengujian menunjukkan bahwa masih terdapat banyak kekurangan saat melakukan validasi data yang akan dimasukkan, sehingga dapat menyebabkan data yang disimpan pada database tidak sesuai dengan data yang diharapkan. Hasil pengujian ini dapat dijadikan masukan untuk memperbaiki aplikasi data kinerja untuk masa yang akan datang.

**Kata Kunci:** Aplikasi, Pengujian, *Black Box*, *Equivalence Partitions*

## 1. Pendahuluan

Pengujian pada sebuah program sangat penting untuk dilakukan untuk memeriksa semua kesalahan yang ada pada program tersebut agar tidak terjadi kerugian yang akan ditimbulkan dari kesalahan tersebut, sehingga sangat perlu untuk melakukan pengujian untuk mengurangi terjadinya kesalahan yang merugikan tersebut (Ningrum, Suherman, Aryanti, Prasetya, & Saifudin, 2019).

Pada Sistem Aplikasi Informasi Data Kinerja ini bertujuan untuk mengetahui kualitas seorang karyawan. Hasil dari penilaian dalam Sistem Aplikasi Informasi Data Kinerja ini dapat memberikan sebuah pertimbangan bagi manager atau atasan dalam pengambilan keputusan untuk menilai baik dan buruk kinerja karyawan (MZ, 2016).

Metode *Black Box Testing* adalah sebuah metode yang digunakan untuk menguji sebuah perangkat lunak tanpa harus memperhatikan hal detail perangkat lunak. Pengujian ini hanya memeriksa nilai keluaran berdasarkan nilai masukan masing-masing (Hanifa, 2016). Proses *Black Box Testing* dengan cara mencoba memasukan data pada setiap formnya. Pengujian ini sangat diperlukan untuk mengetahui apakah program tersebut berjalan sesuai dengan yang diinginkan atau tidak (Wahyudi, 2016).

Berdasarkan dari observasi yang dilakukan dalam menilai karyawan bekerja dengan baik atau tidak masih menggunakan cara yang manual yang cenderung tidak akurat, khususnya bagi atasan atau manajernya dan perusahaan. Dari permasalahan yang ada, dibutuhkan sebuah aplikasi penunjang keputusan penilaian data kinerja yang diharapkan dengan adanya aplikasi tersebut akan dapat memberikan keuntungan bagi manager dan perusahaan (Krismadi, et al., 2019).

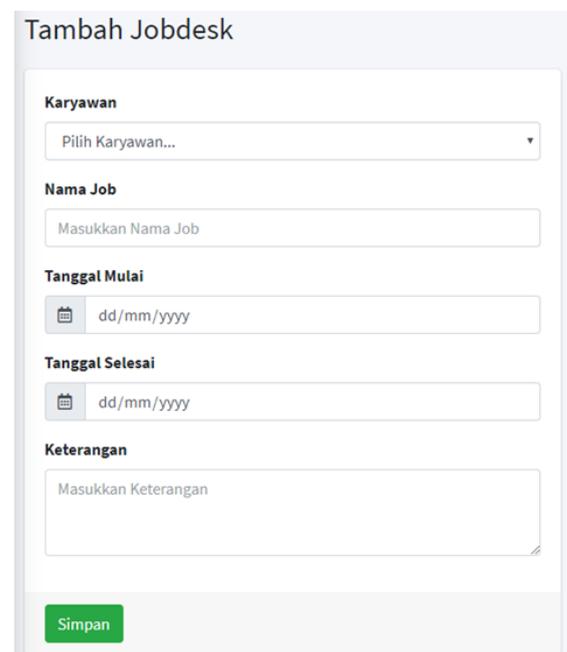
Ada beberapa teknik dalam pengujian menggunakan *Black Box Testing*, salah satunya adalah menggunakan teknik *Equivalence Partitions* (Muttaqin, 2018). *Equivalence Partitions* merupakan sebuah pengujian berdasarkan masukan data pada setiap form yang ada pada sistem aplikasi informasi data kinerja, setiap menu masukan akan dilakukan pengujian dan dikelompokkan berdasarkan fungsinya, baik itu hasilnya valid atau tidak valid (Hidayat, 2018).

## 2. Metodologi

Pengujian yang dilakukan pada aplikasi informasi data kinerja ini menggunakan masukan data acak yang bertujuan untuk memastikan sistem menolak untuk menyimpan data masukan

pada database, sehingga sistem dikatakan layak untuk digunakan (Nurudin, Jayanti, Saputro, Saputra, & Yulianti, 2019). 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 data berupa dokumentasi pengujian dengan metode *Equivalence Partitions* dan nilai tingkat efektifitas metode *Equivalence Partitions* (Jaya, Gumilang, Wati, Andersen, & Desyani, 2019).

Pada hasil pengujian terdapat tabel *Test Case* yang berfungsi untuk menyimpulkan apakah sistem berhasil dalam pengujian tipe tersebut atau tidak (Arochman, 2016). Berikut terdapat penjelasan dari tabel *Test Case*.



Gambar 1 Form Tambah *Jobdesk*

Dari form pada Gambar 1 ada beberapa rencana pengujian. Kolom Karyawan diisi dengan memilih nama karyawan untuk menyelesaikan *jobdesk*, kolom Nama Job diisi sesuai pekerjaan yang ingin diberikan oleh manager atau atasan, kolom Tanggal Mulai diisi sesuai pekerjaan dimulainya tanggal berapa dan tidak boleh kosong, kolom Tanggal Selesai diisi sesuai tanggal selesai yang diberikan oleh manager atau atasan dan tidak boleh kosong, kolom Keterangan diisi sesuai keterangan yang ingin diberikan manager atau atasan dan boleh kosong.

Tabel 1. Rancangan *Test Case* Form Tambah *Jobdesk*

ID	Deskripsi	Hasil yang diharapkan
A01	Memilih “Karyawan” yang akan diberikan <i>jobdesk</i> , mengisi “Nama Job”, menentukan “Tanggal Mulai” dan “Tanggal Selesai”, mengisi “Keterangan”, kemudian klik tombol “Simpan”	Sistem mampu menyimpan data ke dalam database dan halaman list <i>jobdesk</i> menampilkan data yang sudah dimasukan
A02	Tidak memilih “Karyawan” yang akan diberikan <i>jobdesk</i> , tidak mengisi nama job, tidak menentukan “Tanggal Mulai” dan “Tanggal Selesai”, kemudian klik tombol “Simpan”	Sistem menolak untuk menyimpan data dan menampilkan pesan peringatan, mohon data diisi

Dari form pada Gambar 2 ada beberapa rencana pengujian. Kolom *Jobdesk* diisi dengan memilih *jobdesk* untuk mengambil *jobdesk*, kolom Tanggal Mulai diisi sesuai keinginan karyawan kapan untuk memulai *jobdesk* tersebut dan tidak boleh kosong, kolom Keterangan diisi sesuai keterangan yang ingin diberikan karyawan dan boleh kosong.

Gambar 2 Form Tambah Target Kerja

Tabel 2. Rancangan *Test Case* Form Tambah Target Kerja

ID	Deskripsi	Hasil yang diharapkan
B01	Memilih “ <i>Jobdesk</i> ” yang akan dikerjakan, menentukan “Tanggal Mulai”, mengisi “Keterangan”, kemudian klik tombol “Simpan”	Sistem mampu menyimpan data ke dalam database dan list target kerja menampilkan data yang sudah dimasukan
B02	Tidak memilih “ <i>Jobdesk</i> ” yang akan diberikan <i>jobdesk</i> , tidak menentukan “Tanggal Mulai”, kemudian klik tombol “Simpan”	Sistem menolak untuk menyimpan data dan menampilkan pesan peringatan, mohon data diisi

Dari form pada Gambar 3 ada beberapa rencana pengujian. Kolom *Jobdesk* tidak dapat diedit, kolom Tanggal Selesai diisi sesuai kapan karyawan menyelesaikan *jobdesk* tersebut dan tidak boleh kosong, kolom Keterangan diisi sesuai keterangan yang ingin diberikan karyawan dan boleh kosong.

Gambar 3 Form Edit Target Kerja

Tabel 3. Rancangan *Test Case* Form Edit Target Kerja

ID	Deskripsi	Hasil yang diharapkan
C01	Memilih “ <i>Jobdesk</i> ” yang akan dikerjakan, menentukan “Tanggal Selesai”, mengisi “Keterangan”, kemudian klik tombol “Simpan”	Sistem mampu menyimpan data ke dalam database dan list target kerja menampilkan data yang sudah dimasukan
C02	Tidak menentukan “Tanggal Selesai”, kemudian klik tombol “Simpan”	Sistem menolak untuk menyimpan data dan menampilkan pesan peringatan, mohon data diisi

Gambar 4 Form *Approve* Target Kerja

Dari form pada Gambar 4 ada beberapa rencana pengujian. Kolom *Jobdesk* tidak dapat diedit, kolom Tanggal Selesai tidak dapat diedit, kolom Tanggal Mulai tidak dapat diedit, kolom Keterangan tidak dapat diedit, kolom Nilai hanya dapat diisi dengan angka dan tidak boleh kosong, kolom Nilai Kreatifitas hanya dapat diisi dengan angka dan tidak boleh kosong, kolom Nilai Attitude hanya dapat diisi dengan angka dan tidak boleh kosong.

Tabel 4. Rancangan *Test Case* Form *Approve* Target Kerja

ID	Deskripsi	Hasil yang diharapkan
D01	Masukan Nilai “80”, Masukan Nilai Kreatifitas “90”, Masukan Nilai Attitude “100”, kemudian klik tombol “Simpan”	Sistem mampu menyimpan data ke dalam database dan list approve menampilkan nilai karyawan yang sudah dimasukan
D02	Masukan Nilai “AAA”, Masukan Nilai Kreatifitas “BBB”, Masukan Nilai Attitude “CCC”, kemudian klik tombol “Simpan”	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 5.

Tabel 5. Hasil Pengujian

ID	Deskripsi	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
A01	Memilih “Karyawan” yang akan diberikan <i>jobdesk</i> , mengisi “Nama Job”, menentukan “Tanggal Mulai” dan “Tanggal Selesai”, mengisi “Keterangan”, kemudian klik tombol “Simpan”	Sistem mampu menyimpan data ke dalam database dan halaman list <i>jobdesk</i> menampilkan data yang sudah dimasukan	Menampilkan halaman list <i>jobdesk</i> yang sudah dimasukan oleh manajer.	Berhasil
A02	Tidak memilih “Karyawan” yang akan diberikan <i>jobdesk</i> , tidak mengisi nama job, tidak menentukan “Tanggal Mulai” dan “Tanggal Selesai”, kemudian klik tombol “Simpan”	Sistem menolak untuk menyimpan data dan menampilkan pesan peringatan, mohon data diisi	Tidak menyimpan data dan tidak menampilkan halaman <i>jobdesk</i> yang sudah dimasukan oleh manajer.	Berhasil
B01	Memilih “ <i>Jobdesk</i> ” yang akan dikerjakan, menentukan “Tanggal Mulai”, mengisi “Keterangan”, kemudian klik tombol “Simpan”	Sistem mampu menyimpan data ke dalam database dan list target kerja menampilkan data yang sudah dimasukan	Menampilkan hasil masukan dari tanggal mulai <i>jobdesk</i> yang sudah di isi.	Berhasil
B02	Tidak memilih “ <i>Jobdesk</i> ” yang akan diberikan <i>jobdesk</i> , tidak menentukan “Tanggal Mulai”, kemudian klik tombol “Simpan”	Sistem menolak untuk menyimpan data dan menampilkan pesan peringatan, mohon data diisi	Tidak menyimpan data dan tidak menampilkan hasil masukan dari tanggal mulai <i>jobdesk</i>	Berhasil
C01	Memilih “ <i>Jobdesk</i> ” yang akan dikerjakan, menentukan “Tanggal Selesai”, mengisi “Keterangan”, kemudian klik tombol “Simpan”	Sistem mampu menyimpan data ke dalam database dan list target kerja menampilkan data yang sudah dimasukan	Menampilkan halaman data list target kerja yang sudah dimasukan	Berhasil
C02	Tidak menentukan “Tanggal Selesai”, kemudian klik tombol “Simpan”	Sistem menolak untuk menyimpan data dan menampilkan pesan peringatan, mohon data diisi	Tidak menyimpan data dan menampilkan pesan peringatan	Berhasil
D01	Masukan Nilai “80”, Masukan Nilai Kreatifitas “90”, Masukan Nilai Attitude “100”, kemudian klik tombol “Simpan”	Sistem mampu menyimpan data ke dalam database dan list <i>approve</i> menampilkan nilai karyawan yang sudah dimasukan	Menyimpan data dan list <i>approve</i> menampilkan nilai karyawan yang sudah dimasukan	Berhasil
D02	Masukan Nilai “AAA”, Masukan Nilai Kreatifitas “BBB”, Masukan Nilai Attitude “CCC”, kemudian klik tombol “Simpan”	Sistem menolak untuk menyimpan data dan menampilkan pesan peringatan, mohon data diisi dengan angka	Tidak menyimpan data dan tidak menampilkan nilai karyawan yang sudah dimasukkan	Berhasil

Pada sistem aplikasi informasi data kinerja ini ada beberapa jumlah form yang diuji, 4 form dengan form tambah *jobdesk* diuji sebanyak 2 kali, form tambah target kerja diuji sebanyak 2 kali, form edit target kerja diuji sebanyak 2 kali dan form approve target kerja diuji sebanyak 2 kali. Jadi total pengujian ada 8 kali. Jumlah form yang ditemukan adanya error atau bug pada proses pengujian fungsional. Untuk meningkatkan fungsi dari perangkat lunak maka harus memiliki suatu prosedur yang jelas, agar mendapatkan hasil yang valid.

#### 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 kasus pengujian dan menentukan kualitas serta menemukan kesalahan yang ada, dan menjamin aplikasi yang diuji sesuai dengan fungsional yang diinginkan.

#### 5. Saran

Berdasarkan proses pengujian dan hasil pembahasan pengujian perangkat lunak Sistem Aplikasi Informasi Data Kinerja masih perlu untuk dikembangkan. Adapun saran untuk pengembangan aplikasi berikutnya yaitu menggunakan metode selanjutnya dengan metode *White Box Testing* agar dapat mengetahui setiap detail perancangan dari pembuatan struktur perangkat lunak ini, serta dapat mencoba teknik-teknik pengujian *Black Box Testing* lainnya agar mengetahui kualitas perangkat lunak jauh lebih baik untuk menemukan kesalahan.

#### Referensi

Arochman, A. S. (2016). Otomatisasi Desain Test Case Pengujian Perangkat Lunak Metode *Black-Box Testing* Dengan Teknik *Equivalence Partitioning* Menggunakan Algoritma Genetika. *ictech arochmanmk*, 47-52.

Hanifa, R. (2016). Penggunaan Metode *Black Box* Pada Pengujian Sistem Informasi Surat Keluar Masuk. *Jurnal Teknik Informatika*, 34-38.

Hidayat, T. &. (2018). Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online Menggunakan *Black Blox testing* dengan Menggunakan Metode *Equivalence Partitioning* dan *Boundary Value Analysis*. *Jurnal Teknik Informatika UNIS*, 6(1), 25-29.

Jaya, M. S., Gumilang, P., Wati, T., Andersen, Y. P., & Desyani, T. (2019). Pengujian *Black Box* pada Aplikasi Sistem Penunjang Keputusan Seleksi Calon Pegawai Negeri Sipil Menggunakan Teknik *Equivalence Partitions*. *Jurnal Informatika Universitas Pamulang*, 4(4), 131-136.

Krismadi, A., Lestari, A. F., Pitriyah, A., Mardangga, I. W., Astuti, M., & Aries, S. (2019). Pengujian *Black Box* berbasis *Equivalence Partitions* pada Aplikasi Seleksi Promosi Kenaikan Jabatan. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 2(4), 155-161.

Muttaqin, T. (2018). Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online Menggunakan *Black Box Testing* dengan Metode *Equivalence Partitioning* dan *Boundary Value Analysis*. *Jurnal Teknik Informatika UNIS*, 23-27.

MZ, M. K. (2016). Pengujian Perangkat Lunak Metode *Black-Box* Berbasis *Equivalence Partitions* Pada Aplikasi Sistem Informasi Sekolah. *Jurnal Mikrotik*, 2-8.

Ningrum, F. C., Suherman, D., Aryanti, S., Prasetya, H. A., & Saifudin, A. (2019). Pengujian *Black Box* pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik *Equivalence Partitions*. *Jurnal Informatika Universitas Pamulang*, 4(4), 125-130.

Nurudin, M., Jayanti, W., Saputro, R. D., Saputra, M. P., & Yulianti. (2019). Pengujian *Black Box* pada Aplikasi Penjualan Berbasis Web Menggunakan Teknik *Boundary Value Analysis*. *Jurnal Informatika Universitas Pamulang*, 4(4), 143-148.

Wahyudi, R. U. (2016). Sistem Pakar E-Tourism Pada Dinas Pariwisata D.I.Y Menggunakan Metode *Forward Chaining*. *Jurnal Ilmiah DASI*, 67-75.