

## Pengujian Perangkat Lunak Sistem Informasi Penilaian Mahasiswa dengan Teknik *Boundary Value Analysis* Menggunakan Metode Black Box Testing

Bagas Putra Pratama<sup>1</sup>, Ilham Bagus Vitriadi Ristiano<sup>2</sup>, Ismail Adi Prayogo<sup>3</sup>, Nasrullah<sup>4</sup>, Aries Saifudin<sup>5</sup>

Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia

e-mail: <sup>1</sup>bagasputrapratama18@gmail.com, <sup>2</sup>ilhambagusbanget@gmail.com, <sup>3</sup>ismailadi306@gmail.com, <sup>4</sup>nasanasz@gmail.com, <sup>5</sup>aries.saifudin@unpam.ac.id

### Abstract

*In this student assessment application, if there is an error program, there will be losses for students and universities. Losses for students, if this application error occurs, then the assessment received by students will experience delays due to the difficulty in the process of calculating student grades and students will be late to receive an assessment. Losses for the university, if there is an error program on this application, the university will suffer losses if students want a rapid decline in grades but the university cannot calculate quickly and accurately. The method used in testing this application uses the black box method with boundary value analysis techniques. Boundary value analysis techniques test the quality of the application by showing that there are still some errors when entering data to be tested in the column to determine whether the input value is valid or not with the lower limit and the upper limit that has been determined. So the problems that occur can cause data stored in the database does not match the expected data. The test is carried out in the Student Rating form by testing the upper and lower limit values through several predetermined stages for each column contained in the form. Test results can be used as input to improve the application. The results of this test can provide a reference for testing the quality of student assessment applications by analyzing the functionality of the program that has been made in accordance with the expected results.*

*Keywords: information system, assessment, testing, black box, boundary value analysis*

### Abstrak

Pada aplikasi penilaian mahasiswa ini, jika terdapat program *error* maka akan terjadi kerugian bagi mahasiswa dan universitas. Kerugian bagi mahasiswa, jika aplikasi ini terjadi *error* maka penilaian yang diterima mahasiswa akan mengalami keterlambatan dikarenakan sulit dalam proses penghitungan nilai mahasiswa dan mahasiswa akan terlambat menerima penilaian. Kerugian bagi universitas, jika terjadi program *error* pada aplikasi ini, universitas akan mengalami kerugian jika mahasiswa menginginkan penurunan penilaian dengan cepat tapi universitas tidak bisa menghitung dengan cepat dan tepat. Metode yang digunakan pada pengujian aplikasi ini menggunakan metode *black box* dengan teknik *boundary value analysis*. Teknik *boundary value analysis* menguji kualitas aplikasi dengan menunjukkan bahwa masih terdapat beberapa kesalahan pada saat melakukan pemasukan data yang akan diuji pada kolom untuk menentukan nilai masukan tersebut telah valid atau tidak dengan batas bawah dan batas atas yang sudah ditentukan. Sehingga masalah yang terjadi dapat menyebabkan data yang disimpan pada *database* tidak sesuai dengan data yang diharapkan. Pengujian dilakukan pada form Penilaian Mahasiswa dengan menguji nilai batas atas dan nilai batas bawah melalui beberapa tahapan yang telah ditentukan untuk masing-masing kolom yang terdapat di dalam *form* tersebut. Hasil pengujian dapat dijadikan masukan untuk memperbaiki aplikasi. Hasil dari pengujian ini dapat memberikan referensi atas pengujian kualitas aplikasi penilaian mahasiswa dengan menganalisa fungsionalitas dari program yang sudah dibuat sesuai dengan hasil yang diharapkan.

*Kata Kunci: Sistem informasi, penilaian, pengujian, black box, boundary value analysis*

## 1. Pendahuluan

Pengujian perangkat lunak merupakan sebuah proses pengujian program yang dimaksudkan untuk mencari kesalahan pada software. Pengujian juga bertujuan untuk memastikan bahwa software memiliki kualitas yang baik. Kualitas software yang baik adalah software memenuhi kriteria yang diinginkan dan memberikan produktivitas yang tinggi (MZ, 2016, p. 3). Pengujian perangkat lunak penting adanya untuk melakukan uji kualitas perangkat lunak agar menekan peluang terjadinya kesalahan pada manusia dan menutupi kekurangan manusia yang kurang mampu melakukan komunikasi dengan sempurna, hal ini mengakibatkan pengembangan perangkat lunak terhambat dan menjadi bergantung pada jaminan kualitas software (Liana, 2015, p. 8).

Pengujian perangkat lunak memiliki peranan penting dalam suatu sistem informasi, dengan pengujian ini dapat diketahui galat atau error yang akan muncul pada perangkat lunak. Dengan pengujian perangkat lunak diharapkan dapat meminimalisir kesalahan dan cacat pada sebuah software dan sebagai pengukuran kualitas dari software tersebut (Kurniawan, 2017, p. 28). Sehingga sangat perlu melakukan pengujian untuk mengurangi terjadinya kesalahan yang merugikan tersebut (Ningrum, Suherman, Aryanti, Prasetya, & Saifudin, 2019).

Perangkat Lunak yang akan diuji adalah sebuah aplikasi penilaian mahasiswa. Aplikasi Sistem Informasi Penilaian Mahasiswa merupakan sistem untuk memasukan data penilaian mahasiswa yang masuk ke *database*. Aplikasi ini dapat menghitung nilai keseluruhan dari nilai kehadiran, nilai tugas, nilai uts, dan nilai uas.

Bagi mahasiswa, jika aplikasi ini terjadi *error* maka penilaian yang diterima mahasiswa akan mengalami keterlambatan dikarenakan sulit dalam proses penghitungan nilai mahasiswa dan mahasiswa akan terlambat menerima penilaian. Kerugian bagi universitas, jika terjadi program *error* pada aplikasi ini, universitas akan mengalami kerugian jika mahasiswa menginginkan penurunan penilaian dengan cepat tapi universitas tidak bisa menghitung dengan cepat dan tepat.

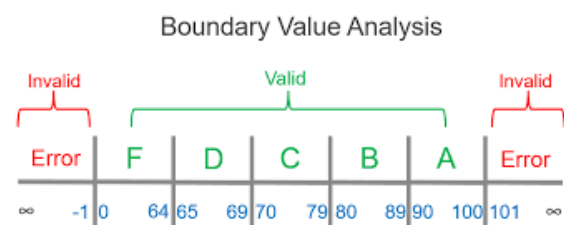
Masalah yang ada pada sistem ini yaitu terdapat kesalahan pada saat memasukan nilai. Sistem ini tetap dapat menghitung dan menampilkan nilai yang lebih dari 100 atau kurang dari 0. Aplikasi ini tidak dapat mengubah data yang sudah dimasukkan ke dalam database.

Metode usulan yang digunakan untuk menguji kualitas aplikasi ini adalah metode *black box* dengan teknik *boundary value analysis*. Teknik *boundary value analysis* digunakan untuk menentukan nilai batas bawah dan batas atas dari data yang ingin diuji (Jaya, 2018, p. 46).

Berdasarkan metode pengujian yang diusulkan, maka dapat dibuat tahapan-tahapan pengujian perangkat lunak ini. Tahapan-tahapan atau rencana dalam melakukan pengujian ini yaitu dengan menentukan nilai batas atas dan nilai batas bawah melalui beberapa tahapan yang telah ditentukan untuk masing-masing kolom dan membangun kasus untuk digunakan dalam pengujian (Wahyu, 2018, p. 207).

## 2. Metodologi

*Boundary value analysis* adalah teknik pengujian perangkat lunak dimana tes dirancang untuk mencakup perwakilan dari nilai batas dalam kisaran. Idenya berasal dari batas. Mengingat bahwa kita memiliki satu set vektor uji untuk menguji sistem, topologi dapat didefinisikan di set itu. Teknik *boundary value analysis* digunakan untuk menentukan nilai batas bawah dan batas atas dari data yang ingin diuji dengan dengan menguji nilai batas atas dan nilai batas bawah melalui beberapa tahapan yang telah ditentukan untuk masing-masing *field* dan membangun kasus untuk digunakan dalam pengujian (Astuti, 2018, p. 188).



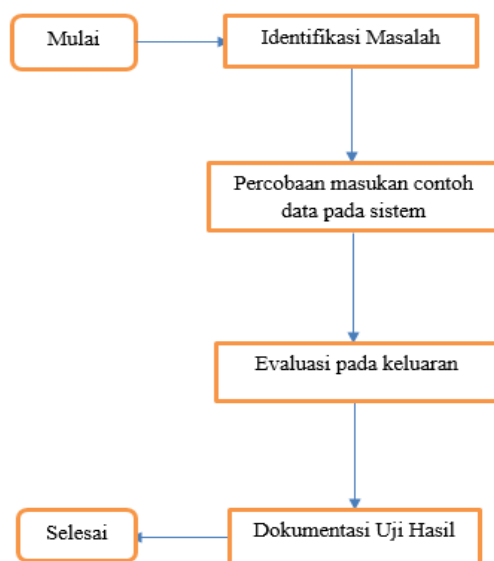
Gambar 1 Desain Teknik *Boundary Value Analysis*

Analisis nilai batas atau disebut juga *Boundary value analysis*, dirancang untuk mencakup perwakilan dari nilai batas dalam kisaran nilai 0 sampai 100, tidak dapat menerima nilai kurang dari 0, dan tidak dapat menerima nilai lebih dari 100. Beberapa prinsip yang mendasari pada *Boundary Value Analysis* yaitu:

1. Terdapat beberapa kesalahan yang terjadi pada masukan.
2. *Boundary Value Analysis* mengijinkan untuk menentukan kasus uji yang menguji batasan nilai masukan.

3. *Boundary Value Analysis* merupakan komplemen dari *equivalence partitioning*. Lebih berfokus pada menentukan elemen-elemen di dalam kelas ekuivalen pada bagian sisi batas dari kelas (Nurudin, Jayanti, Saputro, Saputra, & Yulianti, 2019).

Selain itu suatu proses testing membutuhkan tujuan akhir yang dapat dinilai sehingga pihak tester bisa berhenti melakukan suatu testing ketika tujuan-tujuan itu tercapai (Krismadi, et al., 2019). Pada penelitian ini, ada beberapa tahapan yang dilakukan. Alur penelitian dapat dilihat pada diagram Gambar 2 di bawah ini.



Gambar 2 Alur Penelitian *Boundary Value Analysis*

Penjelasan setiap tahapan dalam diagram Gambar 2 adalah sebagai berikut:

- Identifikasi masalah: menyeleksi fungsionalitas yang akan diuji, merancang tahapan-tahapan pengujian, menentukan data yang akan diuji.
- Percobaan masukan contoh data ke dalam sistem: menentukan nilai batas atas dan batas bawah sesuai dengan ketentuan yang dibuat.
- Proses pengujian evaluasi pada keluaran: melakukan pengujian aplikasi.
- Dokumentasi hasil uji: membuat kesimpulan dari pengujian aplikasi dan mendokumentasi hasil pengujian aplikasi yang sudah dilakukan (Zuriati, 2018, p. 88).

### 3. Hasil dan Pembahasan

Berdasarkan penjelasan sebelumnya, dicoba menerapkan teknik *black box* untuk proses

pengujian aplikasi penilaian mahasiswa. Pengujian dilakukan dengan mengikuti deskripsi pengujian.

Hasil dari pengujian tersebut kemudian dicatat ke dalam kolom hasil yang uji. Berdasarkan dari hasil pengujian kemudian dapat menentukan kesimpulan dengan menyesuaikan antara hasil yang diuji dan hasil yang diharapkan. Jika hasil yang diuji sesuai dengan hasil yang diharapkan maka kesimpulannya adalah berhasil. Jika hasil diuji tidak sesuai dengan hasil yang diharapkan maka kesimpulannya adalah gagal.

The screenshot shows a web application titled "Aplikasi Penilaian Mahasiswa". It features a blue header and a white content area. On the left, there is a vertical list of input fields for "Nama", "NIM", "Nilai Kehadiran", "Nilai Tugas", "Nilai UTS", "Nilai UAS", "Total Nilai", and "Grade". Below these fields are three buttons: "Hitung", "Hapus", and "Simpan". On the right side of the form, there is a large empty white box, likely for displaying the results or a message.

Gambar 3 Form Penilaian Mahasiswa

Dari *form* pada Gambar 3 terdapat beberapa masukan. Masukan Nama menerima masukan berupa huruf, berdasarkan teknik *boundary value analysis*, contoh nilai masukan yang valid antara a-z & A-Z sedangkan yang tidak valid 0-9. Masukan NIM menerima masukan berupa angka, berdasarkan teknik *boundary value analysis*, contoh nilai masukan yang valid antara 0-9 sedangkan yang tidak valid a-z & A-Z. Masukan Nilai Kehadiran menerima masukan berupa angka, berdasarkan teknik *boundary value analysis*, contoh nilai masukan yang valid antara 0-9 sedangkan yang tidak valid a-z & A-Z.

Masukan Nilai Tugas menerima masukan berupa angka, berdasarkan teknik *boundary value analysis*, contoh nilai masukan yang valid antara 0-9 sedangkan yang tidak valid a-z & A-Z. Masukan Nilai UTS menerima masukan berupa angka, berdasarkan teknik *boundary value analysis*, contoh nilai masukan yang valid antara 0-9 sedangkan yang tidak valid a-z & A-Z.

Masukan Nilai UAS menerima masukan berupa angka, berdasarkan teknik *boundary value analysis*, contoh nilai masukan yang valid antara 0-9 sedangkan yang tidak valid a-z & A-Z. Karena

pada kolom Total Nilai dan Grade bukan merupakan kolom masukkan, jadi tidak ada pengujian masukkan nilai.

Dari *form* pada Gambar 4 merupakan hasil dari penambahan data nilai yang telah dimasukkan oleh user untuk data penilaian mahasiswa yang terdiri dari Nama, NIM, Nilai Kehadiran, Nilai Tugas, Nilai UTS, Nilai UAS, dan ada aksi untuk menghitung Total Nilai dan Grade serta aksi untuk menyimpan dan menghapus data penilaian mahasiswa.



Gambar 4 Hasil Tambah Data Mahasiswa

Table 1 Tabel Test Case

Test ID	Deskripsi	Hasil yang Diharapkan	Hasil Pengujian	Status
CS1	Masukan kolom Nama, NIM, Nilai Kehadiran, Nilai Tugas, Nilai UTS, Nilai UAS lalu tekan tombol Hitung.	Sistem akan menghitung dan menampilkan data yang sudah diisi pada form penilaian mahasiswa.	Sistem menghitung dan menampilkan data.	Berhasil
CS2	Mengosongkan salah satu kolom pada form lalu tekan tombol Hitung.	Sistem tidak akan menghitung dan menampilkan data yang sudah diisi pada form penilaian mahasiswa dan akan menampilkan pesan "Ada Data Yang Belum Diisi".	Sistem tidak menghitung dan menampilkan data dan menampilkan pesan "Ada Data Yang Belum Diisi".	Berhasil
CS3	Tidak memasukkan kolom Nama, NIM, Nilai Kehadiran, Nilai Tugas, Nilai UTS, Nilai UAS lalu tekan tombol Hitung.	Sistem tidak akan menghitung dan menampilkan data yang sudah diisi pada form penilaian mahasiswa.	Sistem tidak menghitung dan menampilkan data.	Berhasil
CS4	Masukan kolom Nama, NIM, Nilai Kehadiran, Nilai Tugas, Nilai UTS, Nilai UAS lalu tekan tombol Simpan.	Sistem akan menyimpan data yang sudah dihitung pada form penilaian mahasiswa ke dalam database.	Sistem menyimpan data yang sudah dihitung ke dalam database.	Berhasil
CS5	Mengosongkan salah satu kolom pada form lalu tekan tombol Simpan.	Sistem tidak akan menyimpan data yang sudah dihitung pada form penilaian mahasiswa ke dalam database.	Sistem tidak menyimpan data yang sudah dihitung ke dalam database.	Berhasil
CS6	Masukan kolom Nama, NIM, Nilai Kehadiran, Nilai Tugas, Nilai UTS, Nilai UAS lalu tekan tombol Hapus.	Sistem akan menghapus data yang sudah diisi pada form penilaian mahasiswa.	Sistem mengosongkan semua kolom pada form.	Berhasil

#### 4. Kesimpulan

Berdasarkan pada pengujian yang telah dibahas, dapat diambil kesimpulan bahwa pengujian aplikasi yang menggunakan metode black box hanya berfokus terhadap masukan dan keluaran yang menginformasikan kesesuaian aplikasi yang dikembangkan dengan spesifikasi yang ditetapkan. Hasil pengujian menunjukkan bahwa tingkat keberhasilan aplikasi hanya berjalan sekitar 80%. Sistem tetap mampu menghitung nilai dimana keadaan salah satu kolom dikosongkan. Aplikasi ini perlu diperbaiki lagi untuk meningkatkan kualitas aplikasi dalam menghitung nilai dari 0-100 sebagaimana fungsinya.

#### 5. Saran

Pengujian Aplikasi Penilaian Mahasiswa ini diharapkan dapat menjadi sumber referensi untuk pengujian selanjutnya. Pada penelitian selanjutnya disarankan untuk tidak hanya menggunakan satu metode pengujian, sehingga hasil yang didapatkan bisa lebih valid lagi.

#### Referensi

- Astuti, P. (2018). Penggunaan Black Box Testing (Boundary Value Analysis) Pada Sistem Akademik (SMA/SMK). 188-189.
- Jaya, T. S. (2018). Pengujian Aplikasi dengan Metode BlackBox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung). *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, Vol.03, No.02, 45-48.
- 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.
- Kurniawan, T. A. (2017). Pengujian Perangkat Lunak. *Pengujian Struktur Program Dengan Pengujian Jalur Dasar (Basic Path Testing) : Teori dan Aplikasi*, 24-32.
- Liana, L. (2015). Pengujian Perangkat Lunak. *Pengujian Perangkat Lunak*, 1-15.
- MZ, M. K. (2016). Pengujian Perangkat Lunak Metode Black-Box Berbasis Equivalence Partitions Pada Aplikasi Sistem Informasi Sekolah. *Mikrotik*, 2-7.
- 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.

- Wahyu, Y. S. (2018). Pengujian Black Box Testing Pada Aplikasi Action & Strategi Berbasis Andriod Dengan Teknologi Phonegap. *Jurnal String Vol. 3 No.2*, 206-208.
- Zuriati, D. K. (2018). Teknik Pengujian Boundary Value Analysis Pada Aplikasi Learning Management System Polinela. *Jurnal TAM (Technology Acceptance Model) Volume 9*, 87-88.