

Teknik Boundary Value Analysis pada Blackbox Testing untuk Aplikasi Buku Catatan Harian

Rayhan Putra Fajar

Universitas Muhammadiyah Malang
rayhanputra2706@webmail.umm.ac.id

Abstrak

Di era digital saat ini, aplikasi buku harian digunakan sebagai alat untuk mencatat pengalaman sehari-hari bagi sebagian orang. Namun, untuk memastikan kualitas dan fungsionalitas aplikasi tersebut, diperlukan pengujian yang tepat. Pengujian dengan metode Black Box Testing yang menerapkan teknik Boundary Value Analysis untuk menguji aplikasi buku harian. Boundary Value Analysis adalah teknik yang efektif dalam mengidentifikasi kesalahan dan kerentanan pada batas-batas masukan aplikasi. Pendekatan ini bertujuan untuk mengidentifikasi batas atas dan bawah yang mungkin muncul dalam aplikasi, seperti jumlah karakter maksimal dalam entri atau format masukan yang tidak sesuai. Dengan menganalisis hasil dari pengujian, artikel ini memberikan pengetahuan tentang sejauh mana aplikasi buku harian mampu mengatasi kondisi ekstrem dan menjamin keandalan, sehingga pengguna dapat mengandalkan aplikasi tersebut dalam mencatat pengalaman sehari-hari mereka.

Kata Kunci: Black Box, Boundary Value Analysis, Pengujian

Abstract

In today's digital age, diary apps are used as a tool to record daily experiences for some people. However, to ensure the quality and functionality of the application, proper testing is required. Testing with the Black Box Testing method that applies the Boundary Value Analysis technique to test the diary application. Boundary Value Analysis is an effective technique in identifying errors and vulnerabilities at the input boundaries of the application. This approach aims to identify upper and lower limits that may appear in the application, such as the maximum number of characters in an entry or an inappropriate input format. By analyzing the results from the tests, this article provides knowledge on the extent to which the diary app is able to cope with extreme conditions and guarantee reliability, so that users can rely on the app in recording their daily experiences.

Keywords: Black Box, Boundary Value Analysis, Testing

1. Pendahuluan

Dalam setiap metode *Software Development Life Cycle* setelah tahap implementation/development, langkah selanjutnya ialah Testing/Pengujian [1]. Pengujian sangat penting dalam metode *Software Development Life Cycle* dimana pengujian mengacu pada kualitas dari sebuah sistem. Pengujian adalah suatu rangkaian aktifitas yang terencana dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan. Aktifitas pengujian terdiri dari satu atau sekumpulan tahap dimana dapat menempatkan desain kasus uji yang spesifik [2].

Sebuah perangkat lunak harus diuji untuk mencari berapa banyak kesalahan yang terjadi di dalam perangkat lunak tersebut. Dalam pengujian diperlukan sebuah sesi yang difokuskan untuk menguji dan menemukan kesalahan yang belum ditemukan kemudian diperbaiki agar kualitas dari aplikasi tetap terjaga dan pengujian dapat dikatakan berhasil [3],[4].

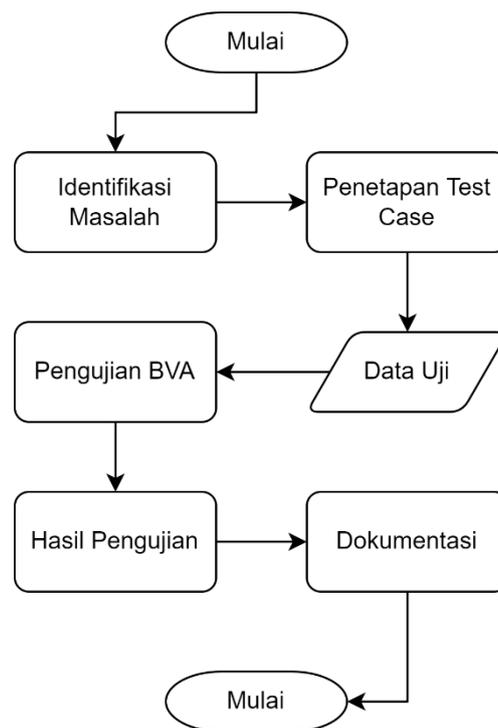
Perangkat lunak yang dilakukan pengujian adalah aplikasi buku harian berbasis android yang dirancang untuk mencatat keseharian pengguna. Aplikasi tersebut merupakan hasil perancangan dari tugas akhir matakuliah pemrograman mobile dengan menggunakan flutter sebagai *framework* yang digunakan untuk mengembangkan aplikasi *mobile cross-platform* (IOS, Android). Pada aplikasi ini terdapat beberapa fitur seperti mengelola buku catatan harian (melihat, menambah, mengubah, menghapus), dan menampilkan catatan harian berdasarkan tanggal.

Metode Pengujian yang akan digunakan pada aplikasi ini adalah *Black Box Testing*. *Black-Box Testing* merupakan Teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak [5]. Dengan menggunakan metode *Black Box*, pengujian dilakukan dengan meninjau nilai input dan output dari sistem perangkat lunak tanpa harus mengetahui struktur kode dari perangkat lunak. Kelebihan dari menggunakan metode pengujian *Black Box* dalam pengujian perangkat lunak adalah tidak harus memiliki pengetahuan tentang suatu bahasa pemrograman dan sudut pandang pengguna dibutuhkan agar dapat menemukan inkonsistensi dalam perangkat lunak. Pengujian *Black Box* dilakukan untuk memastikan bahwa tiap proses sudah berfungsi sesuai dengan kebutuhan yang diharapkan [6].

Metode pengujian *Black Box* memiliki beberapa teknik, di antaranya *Equivalence Partitioning*, *Boundary Value Analysis*, *All pair Testing*, *State Transition* dan *Cause-effect*. Dari teknik pengujian tersebut, teknik *Boundary Value Analysis* diterapkan untuk menguji aplikasi Buku Catatan Harian. Teknik *Boundary Value Analysis* dilakukan untuk menetapkan nilai batas atas dan bawah dari data-data yang diinputkan pada aplikasi yang diuji berdasarkan skenario pengujian yang sudah ditetapkan [7].

2. Metode Penelitian

Untuk menguji kelayakan dari sistem aplikasi buku catatan harian ini, *Black Box Testing* dengan menggunakan teknik *Boundary Value Analysis* yang terdiri atas beberapa tahapan yang dilakukan. Pengujian dimulai dengan mengidentifikasi masalah terlebih dahulu, kemudian menetapkan *test case*-nya, kemudian memasukkan setiap elemen *test case* yang diuji ke dalam aplikasi, pengujian dengan teknik *Boundary Value Analysis*, mendokumentasikan hasil uji [8][9].



Gambar 1. Alur Penelitian

Berdasarkan alur dari penelitian yang digambarkan pada *Gambar 1*, alur pengujian dijabarkan menjadi:

1. Mulai: Penentuan metode dan teknik yang akan digunakan. Pengujian ini menggunakan metode *Black Box Testing* dan Teknik *Boundary Value Analysis*.
2. Identifikasi masalah: Permasalahan yang akan dilakukan pengujian adalah validitas pada aplikasi buku catatan harian yang tidak sesuai dengan data masukan yang tersimpan dalam *database*.

3. Penetapan test case: Data uji yang akan dipakai yaitu halaman login, halaman pendaftaran, halaman tambah catatan, dan halaman mengubah catatan.
4. Data uji: Berupa skenario kasus yang akan dilakukan pengujian berdasarkan *Black Box Testing* berbasis teknik *Boundary Value Analysis* pada *test case* yang telah ditentukan.
5. Hasil pengujian: menunjukkan dokumentasi dari kelayakan aplikasi tersebut sebagai acuan untuk pelaporan pengujian yang dilakukan [10].

3. Hasil Penelitian dan Pembahasan

Berdasarkan penjelasan yang telah dijabarkan sebelumnya, dengan menerapkan teknik BVA dalam proses pengujian aplikasi buku catatan harian untuk menemukan kesalahan pada fungsional aplikasi. Proses pengujian difokuskan pada fungsionalitas dari fitur yang telah ditetapkan untuk dilakukan pengujian. Fitur tersebut adalah fitur *login*, fitur pendaftaran, fitur tambah catatan, fitur ubah catatan. Dalam fitur tersebut memerlukan fungsi masukan/inputan pada setiap kolom/field masukan untuk mendapatkan respons balik dari sistem, seperti pada Tabel 1.

Tabel 1. ID Test Pada Masing-Masing Halaman

| ID Test Page | Nama Halaman |
|--------------|------------------------|
| P1 | Halaman Sign In |
| P2 | Halaman Sign Up |
| P3 | Halaman Tambah Catatan |
| P4 | Halaman Ubah Catatan |



Hello There!

Email

Password

[Sign In](#)

[Don't have an Account? Sign Up](#)

Gambar 2. Halaman Sign In

Pada Gambar 2 halaman login terdapat dua *field* yang dapat diisi yaitu *email* dan *password*. Tombol *Sign In* berfungsi agar pengguna dapat *login* ke dalam aplikasi dengan akun yang telah terdaftar di *database*.

Field email bernilai valid apabila diisi dengan format *email* yang benar, tidak menggunakan simbol selain (`_`, `-`, `.`) contohnya `john.smith@example.com`. *Field password* bernilai valid apabila diisi dengan lebih dari 8 digit kombinasi huruf angka, dan simbol yang diperbolehkan. Dengan

menggunakan teknik *Boundary Value Analysis*, skenario pada halaman sign in dijelaskan pada Tabel 2.

Tabel 2. Skenario Pengujian Halaman Sign In

| ID | Deskripsi | Hasil yang diharapkan |
|----------|--|----------------------------------|
| P1-TC001 | Tidak mengisi <i>email</i> dan <i>password</i> lalu menekan tombol <i>Sign In</i> . | Tampil pesan kesalahan |
| P1-TC002 | Mengosongkan <i>email</i> dan <i>password</i> diisi dengan "password123", lalu menekan tombol <i>Sign In</i> . | Tampil pesan kesalahan |
| P1-TC003 | Mengisi <i>email</i> dengan "email@example.com" dan mengosongkan <i>password</i> , lalu menekan tombol <i>Sign In</i> . | Tampil pesan kesalahan |
| P1-TC004 | Mengisi <i>email</i> dengan format yang salah "email.example.com" dan mengisi <i>password</i> dengan "password123", lalu menekan tombol <i>Sign In</i> . | Tampil pesan kesalahan |
| P1-TC005 | Mengisi <i>email</i> dengan "email@example.com" dan mengisi <i>password</i> dengan data yang salah "password12345", lalu menekan tombol <i>Sign In</i> . | Tampil pesan kesalahan |
| P1-TC006 | Mengisi <i>email</i> dan <i>password</i> dengan data yang telah terdaftar, lalu menekan tombol <i>Sign In</i> . | Berhasil masuk ke dalam aplikasi |

Kemudian berdasarkan Tabel 2, selanjutnya dilakukan pengujian pada halaman sign in dan hasil dari pengujian dirangkum pada Tabel 3.

Tabel 3. Hasil Pengujian Halaman Sign In

| ID | Deskripsi | Hasil Pengujian | Kesimpulan |
|----------|--|----------------------------------|------------|
| P1-TC001 | Tidak mengisi <i>email</i> dan <i>password</i> lalu menekan tombol <i>Sign In</i> . | Tampil pesan kesalahan | Berhasil |
| P1-TC002 | Mengosongkan <i>email</i> dan <i>password</i> diisi dengan "password123", lalu menekan tombol <i>Sign In</i> . | Tampil pesan kesalahan | Berhasil |
| P1-TC003 | Mengisi <i>email</i> dengan "email@example.com" dan mengosongkan <i>password</i> , lalu menekan tombol <i>Sign In</i> . | Tampil pesan kesalahan | Berhasil |
| P1-TC004 | Mengisi <i>email</i> dengan format yang salah "email.example.com" dan mengisi <i>password</i> dengan "password123", lalu menekan tombol <i>Sign In</i> . | Tampil pesan kesalahan | Berhasil |
| P1-TC005 | Mengisi <i>email</i> dengan "email@example.com" dan mengisi <i>password</i> dengan data yang salah "password12345", lalu menekan tombol <i>Sign In</i> . | Tampil pesan kesalahan | Berhasil |
| P1-TC006 | Mengisi <i>email</i> dan <i>password</i> dengan data yang telah terdaftar, lalu menekan tombol <i>Sign In</i> . | Berhasil masuk ke dalam aplikasi | Berhasil |

Pengujian selanjutnya pada halaman Register yang dicontohkan pada Gambar 3.

Gambar 3. Halaman Sign Up

Pada halaman *sign up* terdapat lima *field* yang terdiri dari *email*, *password*, *name*, *username*, dan *phone number*. Semua data harus terisi supaya pengguna dapat mendaftarkan akun ke *database*.

Field email bernilai valid apabila diisi dengan format *email* yang benar, tidak menggunakan simbol selain (*_*, *-*, *.*) contohnya *john.smith@example.com*. *Field password* bernilai valid apabila diisi dengan lebih dari 8 digit kombinasi huruf angka, dan simbol yang diperbolehkan, kemudian *field data diri username* hanya dapat diisi dengan huruf dan angka, *name* hanya dapat diisi dengan huruf dan angka, dan *phone number* hanya dapat diisi dengan angka. Dengan menggunakan teknik *Boundary Value Analysis*, skenario pada halaman *sign up* dijelaskan pada tabel 4.

Tabel 4. Skenario Halaman Sign Up

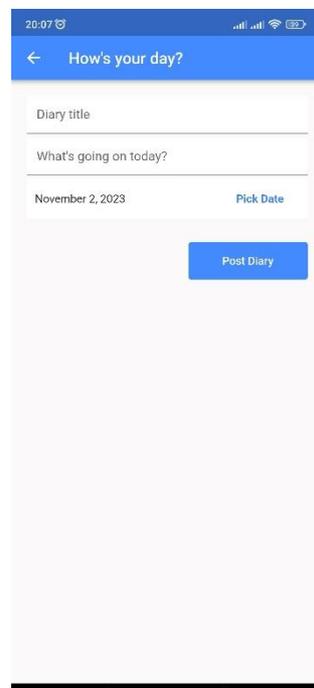
| ID | Deskripsi | Hasil yang diharapkan |
|----------|---|---|
| P2-TC007 | Tidak mengisi semua <i>field</i> lalu menekan tombol <i>Sign Up</i> . | Tampil pesan kesalahan |
| P2-TC008 | Mengosongkan salah satu <i>field</i> , lalu menekan tombol <i>Sign Up</i> . | Tampil pesan kesalahan |
| P2-TC009 | Mengisi semua <i>field</i> dengan data yang benar, namun <i>field phone number</i> diisi dengan huruf, lalu menekan tombol <i>Sign Up</i> . | Tampil pesan kesalahan |
| P2-TC010 | Mengisi semua <i>field</i> dengan data yang benar, lalu menekan tombol <i>Sign Up</i> . | Data pengguna tersimpan dalam <i>database</i> |

Kemudian berdasarkan Tabel 4, selanjutnya dilakukan pengujian pada halaman login dan hasil dari pengujian dirangkum pada Tabel 5.

Tabel 5. Hasil Pengujian Halaman Sign Up

| ID | Deskripsi | Hasil Pengujian | Kesimpulan |
|----------|---|--|------------|
| P2-TC007 | Tidak mengisi semua <i>field</i> lalu menekan tombol <i>Sign Up</i> . | Tampil pesan kesalahan | Berhasil |
| P2-TC008 | Mengosongkan salah satu <i>field</i> , lalu menekan tombol <i>Sign Up</i> . | Tampil pesan kesalahan | Berhasil |
| P2-TC009 | Mengisi semua <i>field</i> dengan data yang benar, namun <i>field phone number</i> diisi dengan huruf, lalu menekan tombol <i>Sign Up</i> . | Tampil pesan kesalahan | Gagal |
| P2-TC010 | Mengisi semua <i>field</i> dengan data yang benar, lalu menekan tombol <i>Sign Up</i> . | Data pengguna tersimpan dalam database | Berhasil |

Selanjutnya pengujian dilakukan pada halaman tambah catatan yang ditunjukkan oleh Gambar 4.



Gambar 4. Halaman Tambah Catatan

Pada halaman tambah catatan terdapat dua *field* yang terdiri dari judul, isi konten, dan pilih tanggal. Semua data harus terisi supaya pengguna dapat menambahkan catatan akun ke *database*.

Field judul hanya dapat berisi 10 kalimat dan tanggal yang dapat dipilih tidak boleh melampaui waktu saat itu. Dengan menggunakan teknik *Boundary Value Analysis*, skenario pada halaman tambah catatan dijelaskan pada Tabel 6.

Tabel 6. Skenario Halaman Tambah Catatan

| ID | Deskripsi | Hasil yang diharapkan |
|----------|--|------------------------|
| P3-TC011 | Tidak mengisi semua <i>field</i> lalu menekan tombol <i>Post</i> . | Tampil pesan kesalahan |
| P3-TC012 | Mengosongkan salah satu <i>field</i> , lalu menekan tombol <i>Post</i> . | Tampil pesan kesalahan |

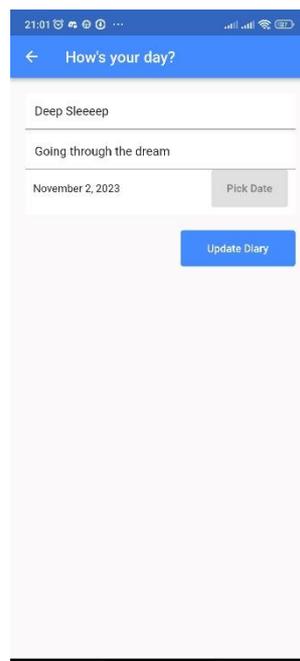
| | | |
|----------|---|--|
| P3-TC013 | Mengisi semua <i>field</i> dengan data yang benar, namun tanggal yang dipilih melampaui waktu saat itu, lalu menekan tombol <i>Post</i> . | Tampil pesan kesalahan |
| P3-TC014 | Mengisi <i>field</i> judul melebihi 10 kalimat, mengisi <i>field</i> isi konten dan tanggal dengan benar, lalu menekan tombol <i>Post</i> . | Tampil pesan kesalahan |
| P3-TC015 | Mengisi semua <i>field</i> dengan data yang benar, lalu menekan tombol <i>Post</i> . | Data catatan pengguna tersimpan dalam database |

Kemudian pengujian dari skenario pada Tabel 6 pada halaman tambah catatan, rangkuman dari pengujian ditunjukkan pada Tabel 7.

Tabel 7 Hasil Pengujian Halaman Tambah Catatan

| ID | Deskripsi | Hasil Pengujian | Kesimpulan |
|----------|---|--|------------|
| P3-TC011 | Tidak mengisi semua <i>field</i> lalu menekan tombol <i>Post</i> . | Tampil pesan kesalahan | Berhasil |
| P3-TC012 | Mengosongkan salah satu <i>field</i> , lalu menekan tombol <i>Post</i> . | Tampil pesan kesalahan | Berhasil |
| P3-TC013 | Mengisi semua <i>field</i> dengan data yang benar, namun tanggal yang dipilih melampaui waktu saat itu, lalu menekan tombol <i>Post</i> . | Tampil pesan kesalahan | Gagal |
| P3-TC014 | Mengisi <i>field</i> judul melebihi 10 kalimat, mengisi <i>field</i> isi konten dan tanggal dengan benar, lalu menekan tombol <i>Post</i> . | Tampil pesan kesalahan | Gagal |
| P3-TC015 | Mengisi semua <i>field</i> dengan data yang benar, lalu menekan tombol <i>Post</i> . | Data catatan pengguna tersimpan dalam database | Berhasil |

Pengujian selanjutnya dilakukan pada halaman ubah catatan yang ditunjukkan pada Gambar 5.



Gambar 5. Halaman Ubah Catatan

Pada halaman ubah catatan terdapat dua *field* yang terdiri dari judul, isi konten, dan tanggal yang tidak dapat diubah. Semua data harus terisi supaya perubahan dari catatan pengguna dapat diperbarui dalam *database*.

Field judul hanya dapat berisi 10 kalimat. Dengan menggunakan teknik *Boundary Value Analysis*, skenario pada halaman ubah catatan dijelaskan pada Tabel 8.

Tabel 8. Skenario Halaman Ubah Catatan

| ID | Deskripsi | Hasil yang diharapkan |
|----------|---|--|
| P4-TC016 | Tidak mengisi semua <i>field</i> lalu menekan tombol <i>Update</i> . | Tampil pesan kesalahan |
| P4-TC017 | Mengosongkan salah satu <i>field</i> , lalu menekan tombol <i>Update</i> . | Tampil pesan kesalahan |
| P4-TC018 | Mengisi <i>field</i> judul melebihi 10 kalimat, mengisi <i>field</i> isi konten dengan benar, lalu menekan tombol <i>Update</i> . | Tampil pesan kesalahan |
| P3-TC019 | Mengisi semua <i>field</i> dengan data yang benar, lalu menekan tombol <i>Update</i> . | Data catatan pengguna tersimpan dalam database |

Kemudian dari skenario pada Tabel 8 pada halaman ubah catatan, rangkuman pengujian dari Tabel 8 ditunjukkan pada Tabel 9.

Tabel 9. Hasil Pengujian Halaman Ubah Catatan

| ID | Deskripsi | Hasil Pengujian | Kesimpulan |
|----------|---|---|------------|
| P4-TC016 | Tidak mengisi semua <i>field</i> lalu menekan tombol <i>Update</i> . | Tampil pesan kesalahan | Berhasil |
| P4-TC017 | Mengosongkan salah satu <i>field</i> , lalu menekan tombol <i>Update</i> . | Tampil pesan kesalahan | Berhasil |
| P4-TC018 | Mengisi <i>field</i> judul melebihi 10 kalimat, mengisi <i>field</i> isi konten dengan benar, lalu menekan tombol <i>Update</i> . | Tampil pesan kesalahan | Gagal |
| P3-TC019 | Mengisi semua <i>field</i> dengan data yang benar, lalu menekan tombol <i>Update</i> . | Data catatan pengguna dalam database diperbarui | Berhasil |

4. Kesimpulan

Berdasarkan penelitian dengan melakukan pengujian sebagai proses lanjutan pengembangan aplikasi buku catatan harian yang merupakan hasil perancangan dari tugas akhir dari mata kuliah pemrograman mobile sebagai penerapan mata kuliah Penjamin Kualitas Perangkat Lunak dengan menerapkan metode Black Box Testing dengan teknik Boundary Value Analysis dapat disimpulkan bahwa hasil dari pengujian yang dilakukan sesuai dengan alur pengujian yang telah dirancang sebelumnya dengan hasil sebagai berikut:

1. Metode Black Box Testing dengan Teknik Boundary Value Analysis merupakan pengujian yang efektif untuk mencari kesalahan pada fungsionalitas dari suatu sistem.
2. Pengujian fungsionalitas dilaksanakan dengan menggunakan beberapa skenario masukan yang diujikan menghasilkan data yang valid untuk kemudian disimpan dalam database, namun masih terdapat fungsionalitas masih dapat berjalan meskipun menerima masukan data yang tidak diharapkan sehingga dapat menyebabkan data yang disimpan kurang valid.
3. Berdasarkan hasil uji dengan teknik Boundary Value Analysis, fungsi masukan data perlu dilengkapi dengan proses validasi data untuk menjamin akurasi masukan data sesuai dengan fungsional yang diharapkan.

Referensi

- [1] Velmourougan, S., Dhavachelvan, P., Baskaran, R., & Ravikumar, B. (2014). Software development life cycle model to improve maintainability of software applications.

- "Proceedings - 2014 4th International Conference on Advances in Computing and Communications", ICACC 2014, 270–273. <https://doi.org/10.1109/ICACC.2014.71>
- [2] W. N. Cholifah, Y. Yulianingsih, and S. M. Sagita, "Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap," *STRING (Satuan Tulisan Ris. dan Inov. Teknol.*, vol. 3, no. 2, p. 206, 2018.
- [3] S. R. Yulistina, T. Nurmala, R. M. A. T. Supriawan, S. H. I. Juni, and A. Saifudin, "Penerapan Teknik Boundary Value Analysis untuk Pengujian Aplikasi Penjualan Menggunakan Metode Black Box Testing," *J. Inform. Univ. Pamulang*, vol. 5, no. 2, p. 129, 2020.
- [4] T. Hidayat and H. D. Putri, "Pengujian Portal Mahasiswa pada Sistem Informasi Akademik (SINA) menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis," *J. Inform. Pengemb. IT*, vol. 7, no. 1, pp. 83–92, 2019.
- [5] S. Roohullah Jan, S. Tauhid Ullah Shah, Z. Ullah Johar, Y. Shah, and F. Khan., 2016. An Innovative Approach to Investigate Various Software Testing Techniques and Strategies. *Int. J. Sci. Res. Sci. Eng. Technol.*, vol. 2, no. 2, pp. 682–689.
- [6] Maulana, B. A., Mawarni, E., Hidayattuloh, M. Y., Suryawijaya, V., & Saifudin, A. (2023). "Pengujian Black Box pada Sistem Informasi Barang Berbasis Web Menggunakan Metode Boundary Value Analysis". *OKTAL : Jurnal Ilmu Komputer dan Science* vol. 2, no. 6, p 1747-1753.
- [7] D. Debiyanti, S. Sutrisna, B. Budrio, A. K. Kamal, and Y. Yulianti, "Pengujian Black Box pada Perangkat Lunak Sistem Penilaian Mahasiswa Menggunakan Teknik Boundary Value Analysis," *J. Inform. Univ. Pamulang*, vol. 5, no. 2, p. 162, 2020
- [8] Sholeh, M., Gisfas, I., Cahiman, & Fauzi, M. A. (2021). Black Box Testing on ukmbantul.com Page with Boundary Value Analysis and Equivalence Partitioning Methods. *Journal of Physics: Conference Series*, 1823(1). <https://doi.org/10.1088/1742-6596/1823/1/012029>
- [9] Megawati, C. D., Miwa, N. D., & Palevi, B. R. P. D. (2023). Black Box Testing of the "Hybrid Engine" Application Using Boundary Value Analysis Technique. *Sinkron*, 8(2), 923–938. <https://doi.org/10.33395/sinkron.v8i2.12278>
- [10] Hidayat, T., & Muttaqin, M. (2018). "Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis". In *Jurnal Teknik Informatika UNIS JUTIS* (Vol. 6, Issue 1). p 25-29 www.ccsenet.org/cis

