

# Pengujian Aplikasi Presensi dengan Black box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis

A. Rifqi Yarzuq Arfani<sup>1</sup>, Patmi Kasih<sup>2</sup>, Danar Putra Pamungkas<sup>3</sup>

<sup>1,2,3</sup>Teknik Informatika, Fakultas Teknik, Universitas Nusantara PGRI Kediri

E-mail: <sup>\*</sup>[aryarfani@gmail.com](mailto:aryarfani@gmail.com), <sup>2</sup>[fatkasih@gmail.com](mailto:fatkasih@gmail.com), <sup>3</sup>[danar@unpkediri.ac.id](mailto:danar@unpkediri.ac.id)

**Abstrak** – Presensi perkuliahan adalah hal penting dalam dunia perkuliahan, karena kehadiran mahasiswa merupakan salah satu faktor dari penilaian suatu mata kuliah. Tetapi presensi manual dengan menggunakan media tulis memiliki kekurangan seperti rekapitulasi manual dan transparansi data. Memanfaatkan fitur pada gawai seperti gps dibuatlah sebuah aplikasi presensi untuk mengatasi masalah diatas. Melihat pentingnya data presensi, maka wajib dilakukannya pengujian perangkat lunak pada aplikasi tersebut, yang mana hasil pengujian bisa digunakan sebagai pedoman pengembang dalam mencari kesalahan, memastikan kelayakan serta kesesuaian dengan kebutuhan pengguna. Metode yang dipakai pada pengujian perangkat lunak ini adalah metode black box, serta menjelaskan alur untuk mempersiapkan perancangan skenario pengujian yang memakai teknik equivalence partitioning dan boundary value analysis. Hasil yang diperoleh dari penelitian ini adalah kesimpulan untuk memastikan terpenuhinya kebutuhan yang telah ditentukan pada perangkat lunak yang diuji dan fungsionalitas bisa berjalan dengan baik. Berdasarkan kesimpulan pengujian perangkat lunak yang telah dilakukan, maka hasil uji yang didapatkan bisa bermanfaat sebagai bahan koreksi dimasa mendatang.

**Kata Kunci** — Presensi, Pengujian Perangkat Lunak, Black-Box Testing, Equivalence Partitioning, Boundary Value Analysis

## 1. PENDAHULUAN

Presensi perkuliahan adalah sebuah catatan atas hadir atau tidaknya seorang mahasiswa dalam kegiatan perkuliahannya. Dalam kegiatan perkuliahan daftar hadir seorang mahasiswa adalah poin penting yang sangat diperhatikan dan diawasi oleh dosen pengampu[1]. Karena hasil rekap atas daftar hadir mahasiswa tersebut dapat berpengaruh terhadap penilaian masing masing mata kuliah dan dapat sebagai syarat mahasiswa dalam mengikuti UAS. Daftar hadir yang dilakukan dengan cara manual[2], yaitu dengan memberikan tanda tangan mahasiswa masing-masing dalam setiap mata kuliah.

Presensi manual menggunakan media tulis adalah sistem yang sering diterapkan sebagai alat presensi saat ini seperti di perkuliahan teknik informatika. Sistem presensi dengan media alat tulis dapat membantu dosen dalam pencatatan data kehadiran. Tetapi presensi manual atau dengan menggunakan media tulis ini memiliki kekurangan[3], yaitu data rekapitulasi dari presensi bersifat transparan yang artinya disini dosen pengampu masing-masing mata kuliah tidak mengetahui kebenarannya.

Aplikasi presensi merupakan jawaban dari masalah diatas, dosen atau admin dapat mengisi area yang diperbolehkan untuk mahasiswa melakukan presensi, dan mahasiswa akan menggunakan aplikasi tersebut untuk mengisi presensi. Memanfaatkan fitur gps yang ada pada perangkat gawai dan algoritma haversine[4], aplikasi ini memastikan bahwa mahasiswa dapat mengisi presensi jika berada pada lokasi yang telah ditentukan sebelumnya. Selain presensi mahasiswa juga dapat mengisi izin maupun

cuti perkuliahan yang bisa dilihat pada web dashboard administrator.

Pengujian perangkat lunak adalah hal penting untuk menjamin kualitas perangkat lunak yang menyajikan bagian utama atau dasar atas desain, spesifikasi, dan pengkodean[5]. Pengujian perangkat lunak dilaksanakan dengan tujuan supaya memastikan bahwa kualitas dari perangkat lunak yang digunakan telah memenuhi syarat[6]. Pengujian perangkat lunak ini dapat dilakukan dengan pengekseskuan program mencari adanya kesalahan sintaks program[7], lalu mengadakan verifikasi program guna menyesuaikan perangkat dengan kebutuhan pengguna. Pengujian yang digunakan dalam penelitian ini menggunakan metode black box testing.

Metode black box merupakan sebuah pengujian yang dilakukan dengan melihat dan mengamati hasil dari pengekseskuan program serta memeriksa fungsi perangkat lunak. Dapat dikatakan black box adalah sebagai kotak hitam yang dimana kita hanya melihat bagian luar tanpa mengetahui isi dari kotak hitam tersebut. Hal ini memiliki kesamaan dalam melakukan pengujian black box, dimana melakukan evaluasi dengan mengamati bagian luar atau tampilannya saja tanpa melihat proses detail yang terjadi didalamnya[8].

## 2. METODE PENELITIAN

Tahapan yang dilakukan pada penilitan pengujian aplikasi presensi dengan black-box testing menggunakan metode equivalence partitioning dan boundary value analysis sebagai berikut[9]:

- a. Identifikasi Permasalahan: Mengidentifikasi permasalahan dan menganalisa supaya meraih jalan keluar atau pemecahan dari permasalahan yang ditemukan.
- b. Analisa Kebutuhan Pengujian: Menganalisa kebutuhan yang berguna untuk mempersiapkan pengujian perangkat lunak, seperti mencari metode pengujian yang tepat, dilanjutkan dengan mempelajari kebutuhan sistem diambil dengan sudut pandang penguji untuk menganalisa persyaratan pengujian perangkat lunak tersebut.
- c. Perancangan Test Case: Merancang skenario pengujian atau test case berdasarkan analisa spesifikasi dan kebutuhan yang telah dianalisa.
- d. Pelaksanaan Pengujian: Melaksanakan pengujian dengan menjalankan skenario pengujian yang telah dibuat. Hasil dari pengujian bisa menjadi bahan penentu kesimpulan dari pengujian yang telah dilakukan.
- e. Hasil dan Kesimpulan Pengujian: Tahapan terakhir dari penelitian ini adalah menyimpulkan hasil dari pengujian yang bisa digunakan sebagai bahan koreksi untuk aplikasi tersebut dan peningkatan pengujian selanjutnya. Serta terdapat rekomendasi perbaikan untuk perangkat lunak presensi untuk meningkatkan kualitas aplikasi tersebut.

Pengujian *black box* adalah sebuah program tanpa memperhatikan struktur logika internal perangkat lunak. Metode *black box* memiliki fungsi untuk mengamati kebenaran hasil dari perangkat lunak. Metode *black box* adalah sebuah rancangan pengujian data berdasarkan spesifikasi perangkat lunak. Pengujian data dalam perangkat lunak ini dilakukan eksekusi lalu diamati dan disesuaikan dengan harapan pengguna[10].

Metode *black box* dalam melakukan pengujian data berfungsi untuk menemukan penyimpangan atau kesalahan dengan berbagai kategori dengan sebagai berikut [11]:

- a. Adanya berbagai fungsi yang salah atau ada sesuatu yang hilang.
- b. Adanya kesalahan pada bagian luar (tampilan).
- c. Adanya kesalahan pada database eksternal.
- d. Adanya kesalahan pada saat kinerja pengujian.
- e. Adanya kesalahan yang teminasi

Metode *black box* tidak sama seperti metode *white box* yang menguji secara internal. Metode *black box* ini tidak memerlukan pengetahuan tentang struktur atau alur dalam pengujian perangkat lunak[12]. Maka metode *black box* memudahkan penulis dalam membuat masukan yang dapat menguji semua persyaratan fungsional perangkat lunak.

*Black box* memiliki tujuan yaitu dengan tidak memperdulikan cara kinerja dari struktur sistem. Metode *black box* sama seperti kotak hitam, sehingga pengujian menggunakan metode ini memiliki fokus

pada pencarian kesalahan atau penyimpangan spesifikasi sistem pada data awal.

Metode *equivalence partitioning* merupakan terbagi menjadi domain *input* sistem kedalam kelas data. Metode ini menguji kelas yang terjadi penyimpangan atau kesalahan, sehingga hal ini dapat mengurangi kasus pengujian. Metode *equivalence partitioning* menjelaskan mengenai valid dan tidak validnya saat di *input*. Data yang dimasukkan dapat terdiri atas harga numerik, harga terkait dan rentang harga, dan kondisi boolean[10].

Penentuan kelas ekuivalensi sebagai berikut :

- a. Jika kondisi *input* menentukan suatu range nilai, maka akan ada satu kelas valid dan dua kelas yang invalid.
- b. Jika kondisi menentukan anggota suatu himpunan, maka buat kelas ekivalensi valid untuk semua nilai himpunan dan dua kelas invalid.
- c. Jika suatu kondisi *input* berupa *boolean*, maka akan ada kelas untuk nilai *true* dan *false*.

Berdasarkan penjelasan diatas dapat dibuat sebuah contoh seperti berikut: *inputan password* dengan ketentuan minimal 6 karakter dan maksimal 10 karakter, maka dapat dibuatlah bagian – bagian partisi 0-5, 6-10, 11-14 karakter sebagai nilai yang sepadan. Dari contoh diatas terciptalah satu partisi valid 6-10 dan dua partisi tidak valid 0-5, 11-14.

Tabel 1. Kelas Partisi Input Password

No	Kelas	Hasil diharapkan
1	Masukan 0 sampai 5 karakter	Sistem menolak
2	Masukan 6 sampai 10 karakter	Sistem menerima
3	Masukan 11 sampai 14 karakter	Sistem menolak

Tabel 1, memperlihatkan contoh kelas partisi dari metode *equivalence partitioning* pada *inputan password*.

*Boundary value analysis* merupakan suatu metode pengujian *black box* yang melaksanakan pengujian di batas atas dan batas bawah pada nilai atau data uji yang dimasukkan pada suatu *input* aplikasi. *Boundary value analysis* adalah teknik desain proses bertujuan untuk melengkapi metode *equivalence partitioning* dimana hanya berfokus pada nilai input atau data uji saja, sementara teknik ini juga memerperhatikan nilai output. Metode *boundary value analysis* juga bisa digunakan sebagai acuan bahwa data uji yang melebihi batas yang sudah ditentukan tidak bisa diproses pada database atau sistem dan mampu mengeluarkan peringatan, serta hanya memasukkan data uji yang kurang dari batas atas[8]. Beberapa prinsip mendasar pada *boundary value analysis* yaitu[13]:

- Untuk suatu *input range* yang dibatasi a dan b, untuk data uji adalah (a-1), a, (a+1), (b-1), b, (b+1).
- Apabila *input* mensyaratkan suatu bilangan nilai n, untuk data uji adalah sejumlah (n-1), n dan (n+1) nilai.
- Terapkan dua aturan diatas pada syarat *output*
- Apabila struktur data input dari aplikasi mempunyai cakupan (misal: ukuran *buffer*, batas *array*) gunakan data uji yang berada pada batas cakupan.

Tabel 2. Aturan Masukan Input Password

No	Masukan	Status
1	Teks dengan 5 karakter	Invalid
2	Teks dengan 6 karakter	Valid
3	Teks dengan 10 karakter	Valid
4	Teks dengan 11 karakter	Invalid

Tabel 2, memperlihatkan contoh *input* atau masukan menggunakan metode *boundary value analysis* dengan menerapkan kelas partisi yang dihasilkan dari metode *equivalence partitioning* pada *inputan* password.

### 3. HASIL DAN PEMBAHASAN

Pada bagian ini akan dijelaskan tentang penjelasan alur pengujian untuk setiap fungsionalitas yang akan diuji, penjelasan dilengkapi dengan analisa kebutuhan, menentukan partisi dan data uji dengan metode *equivalence partitioning* dan *boundary value analysis*, hasil pengujian dan skenario pengujian. Fungsionalitas yang akan diuji pada penelitian ini adalah fungsi registrasi.



Gambar 1. Form registrasi mahasiswa

Pada Gambar 1, disajikan tampilan form registrasi dimana terdapat beberapa input dimana pengujian

akan dilakukan yaitu: gambar, nama, npm, email, alamat, phone, password, dan konfirmasi password.

Langkah pertama dalam pembuatan skenario pengujian atau *test case* diawali dengan pembuatan partisi menggunakan metode *equivalence partitioning*.

Tujuan dari *equivalence partitioning* adalah membagi kelas data uji menjadi partisi yang sepadan yaitu kelas valid dan kelas invalid. Metode *equivalence partitioning* digunakan untuk meningkatkan efektifitas pengujian dengan membuat skenario pengujian yang berisi data uji dengan minimal satu kali pada setiap kelas partisi.

Tabel 3. *Equivalence Partitioning* Registrasi Valid

Nama Data	Kelas	Status
Profile Image	File Image ukuran 1 – 1000kb	Valid
Nama	[A-Z a-z .] 5 – 40 karakter	Valid
NPM	[0-9 .] 15 karakter	Valid
Email	[A-Z a-z 0- 9 @ .] 5 – 40 karakter	Valid
Alamat	[A-Z a-z 0-9 .] 5 – 40 karakter	Valid
Phone	[0-9] 5 – 40 karakter	Valid
Password	[A-Z a-z 0- 9 @ .] 5 – 40 karakter	Valid
Konfirmasi Password	[A-Z a-z 0- 9 @ .] 5 – 40 karakter	Valid

Tabel 3, menampilkan kelas partisi registrasi valid, dimana terdapat kelas-kelas dari input registrasi yang sistem dapat memproses.

Tabel 4. *Equivalence Partitioning* Registrasi Valid

Nama Data	Kelas	Status
Profile Image	ukuran > 1000 kb	Invalid
Nama	< 5 karakter > 40 karakter	Invalid
NPM	< 15 karakter > 15 karakter	Invalid
Email	< 5 karakter > 40 karakter	Invalid
Alamat	< 5 karakter > 40 karakter	Invalid
Phone	< 5 karakter > 40 karakter	Invalid
Password	< 5 karakter > 40 karakter	Invalid
Konfirmasi Password	< 5 karakter > 40 karakter	Invalid

Tabel 4, menampilkan kelas partisi registrasi invalid, dimana terdapat kelas-kelas dari input registrasi yang sistem tidak dapat memproses.

Setelah kelas *equivalence partitioning* ditemukan maka selanjutnya adalah tahapan *boundary value analysis*. *Boundary value analysis* bertujuan untuk mengujian ambang batas data masukan pada kelas ekuivalensi atau padanan yang telah ditentukan

sebelumnya pada metode *equivalence partitioning*. Dengan kata lain metode *boundary value analysis* adalah pelengkap dari metode *equivalence partitioning*.

Tabel 5. *Boundary Value Analysis* Registrasi Valid

Nama Data	Kelas	Status
Profile Image	File dengan ukuran 1000 kb	Valid
Nama	Teks dengan 5 karakter	Valid
	Teks dengan 40 karakter	Valid
NPM	Teks dengan 15 karakter	Valid
Email	Teks dengan 5 karakter	Valid
	Teks dengan 40 karakter	Valid
Alamat	Teks dengan 5 karakter	Valid
	Teks dengan 40 karakter	Valid
Phone	Angka dengan 5 karakter	Valid
	Angka dengan 40 karakter	Valid
Password	Angka dengan 5 karakter	Valid
	Angka dengan 40 karakter	Valid
Konfirmasi Password	Angka dengan 5 karakter	Valid
	Angka dengan 40 karakter	Valid

Tabel 5, menampilkan ketentuan data uji valid dari setiap input registrasi yang akan digunakan dalam skenario pengujian.

Tabel 6. *Boundary Value Analysis* Registrasi Invalid

Nama Data	Kelas	Status
Profile Image	File dengan ukuran 1001kb	Invalid
Nama	Teks dengan 4 karakter	Invalid
	Teks dengan 41 karakter	Invalid
NPM	Teks dengan 14 karakter	Invalid
	Teks dengan 16 karakter	Invalid
Email	Teks dengan 4 karakter	Invalid
	Teks dengan 41 karakter	Invalid
Alamat	Teks dengan 4 karakter	Invalid
	Teks dengan 41 karakter	Invalid
Phone	Angka dengan 4 karakter	Invalid
	Angka dengan 41 karakter	Invalid
Password	Angka dengan 4 karakter	Invalid
	Angka dengan 41 karakter	Invalid
Konfirmasi Password	Angka dengan 4 karakter	Invalid
	Angka dengan 41 karakter	Invalid

Tabel 6, menampilkan ketentuan data uji invalid dari setiap input registrasi yang akan digunakan dalam skenario pengujian.

Setelah batas – batas data masukan valid dan invalid ditentukan, maka dibuatlah kasus uji yang berisi masukan atau data uji berdasarkan kelas – kelas pada ambang batas bawah dan batas atas yang terbuat menggunakan metode *boundary value analysis* akan diuji pada aplikasi presensi. Dibawah ini merupakan rancangan skenario pengujian untuk form registrasi.

Tabel 7. Skenario Pengujian Registrasi

<b>Nama Uji Kasus</b>	Registrasi Mahasiswa	
<b>Deskripsi</b>	Mahasiswa melakukan registrasi akun, dan data tersebut akan digunakan untuk login aplikasi	
<b>Kondisi Awal</b>	Aplikasi menampilkan halaman login	
<b>Skenario</b>		
<ol style="list-style-type: none"> <li>Masuk ke halaman registrasi.</li> <li>Masukkan data gambar, nama, NPM, email, alamat, phone, password, konfirmasi password.</li> <li>Tekan tombol registrasi.</li> </ol>		
<b>Hasil</b>		
<b>Yang Diharapkan</b>	<b>Kesimpulan</b>	
	<b>PASS</b>	<b>FAIL</b>
1. Halaman registrasi tampil.	✓	
2. Dapat mengambil <i>input</i> gambar dari penyimpanan gawai.	✓	
3. Dapat mengambil <i>input</i> gambar dari kamera gawai.	✓	
4. <i>Input</i> profile image dapat terisi file ukuran kurang dari 1000 kb	✓	
5. <i>Input</i> nama dapat terisi dengan 5 sampai 40 karakter	✓	
6. <i>Input</i> NPM dapat terisi dengan 5 sampai 40 karakter	✓	
7. <i>Input</i> email dapat terisi dengan 5 sampai 40 karakter	✓	
8. <i>Input</i> alamat dapat terisi dengan 5 sampai 40 karakter	✓	
9. <i>Input</i> phone dapat terisi dengan 5 sampai 40 karakter	✓	
10. <i>Input</i> password dapat terisi dengan 5 sampai 40 karakter	✓	
11. <i>Input</i> konfirmasi password dapat terisi dengan 5 sampai 40 karakter	✓	
12. Tampil pesan berhasil disimpan.	✓	
13. Kembali ke halaman dashboard.	✓	
<b>Alternatif</b>		
Kondisi : Menggunakan Data Uji Registrasi A01 Invalid		
<ol style="list-style-type: none"> <li>Masuk ke halaman registrasi.</li> <li>Masukkan data invalid gambar, nama, NPM, email, alamat, phone, password, konfirmasi password.</li> <li>Tekan tombol registrasi.</li> </ol>		

Hasil		
Yang Diharapkan	Kesimpulan	
	PASS	FAIL
1. <i>Input</i> profile image tidak dapat terisi file ukuran lebih dari 1000 kb	✓	
2. <i>Input</i> nama tidak dapat terisi dengan kurang dari 5 dan lebih dari 40 karakter	✓	
3. <i>Input</i> NPM tidak dapat terisi dengan kurang dari 5 dan lebih dari 40 karakter	✓	
4. <i>Input</i> email tidak dapat terisi dengan kurang dari 5 dan lebih dari 40 karakter	✓	
5. <i>Input</i> alamat tidak dapat terisi dengan kurang dari 5 dan lebih dari 40 karakter	✓	
6. <i>Input</i> phone tidak dapat terisi dengan kurang dari 5 dan lebih dari 40 karakter	✓	
7. <i>Input</i> password tidak dapat terisi dengan kurang dari 5 dan lebih dari 40 karakter	✓	
8. <i>Input</i> konfirmasi tidak dapat terisi dengan kurang dari 5 dan lebih dari 40 karakter	✓	
9. Aplikasi tidak menyimpan masukan	✓	
10. Tampil pesan kesalahan.	✓	

Tabel 7 menunjukkan skenario pengujian aplikasi presensi dari fungsionalitas registrasi. Hasil dari pengujian skenario tersebut dari 23 poin skenario pengujian yang diuji, 23 poin berstatus berhasil dan 0 poin berstatus gagal.

#### 4. SIMPULAN

Berdasarkan pengujian pada aplikasi presensi menggunakan teknik *black box* dengan metode *equivalence partitioning* dan *boundary value analysis*, dapat disimpulkan bahwa aplikasi presensi khususnya pada fungsi registrasi telah sesuai dengan kebutuhan dan persyaratan yang ditentukan. Dengan demikian aplikasi presensi mempunyai kualitas bagus dan layak untuk digunakan.

#### 5. SARAN

Saran untuk penelitian mendatang, pengujian dapat dilakukan menggunakan teknik *black box* yang lain seperti *error guessing* atau *regression testing* untuk mengetahui tingkat efektivitas suatu teknik dengan teknik lain, pada akhirnya akan membuat kualitas aplikasi menjadi lebih baik.

#### DAFTAR PUSTAKA

- [1] Madhiyono, S. Kosasi, and David, "Aplikasi Absensi Fingerprint Berbasis Website Menggunakan JWT dan Algoritma Haversine," *J. SISFOKOM (Sistem Inf. dan Komputer)*, vol. 10, no. 3, pp. 328–333, 2021.
- [2] N. L. Khoiriyah, F. Marisa, and I. D. Wijaya, "Rancang Bangun Sistem Presensi Online Berbasis Granted Validitas Data," *J I M P - J. Inform. Merdeka Pasuruan*, vol. 3, no. 1, pp. 53–61, 2018, doi: 10.37438/jimp.v3i1.89.
- [3] D. Hardiyanto and D. Anggun Sartika, "Optimalisasi Metode Deteksi Wajah berbasis Pengolahan Citra untuk Aplikasi Identifikasi Wajah pada Presensi Digital," *Setrum Sist. Kendali-Tenaga-elektronika-telekomunikasi-komputer*, vol. 7, no. 1, p. 107, 2018, doi: 10.36055/setrum.v7i1.3367.
- [4] A. P. Aldya, A. Rahmatulloh, and M. Fachurroji, "Haversine Formula Untuk Membatasi Jarak Pada Aplikasi Presensi Online," *J. Instek Inform. Sains Dan Teknol.*, vol. 4, no. 2, pp. 171–180, 2019.
- [5] D. Ahrizal, M. K. Miftah, R. Kurniawan, T. Zaelani, and Y. Yulianti, "Pengujian Perangkat Lunak Sistem Informasi Peminjaman PlayStation dengan Teknik Boundary Value Analysis Menggunakan Metode Black Box Testing," *J. Inform. Univ. Pamulang*, vol. 5, no. 1, p. 73, 2020, doi: 10.32493/informatika.v5i1.4338.
- [6] I. M. S. Ardana, "Pengujian Software Menggunakan Metode Boundary Value Analysis dan Decision Table Testing," *J. Teknol. Inf. ESIT*, vol. 14, no. 11, pp. 40–47, 2019.
- [7] E. P. Mandyartha, "Pembangkitan Data Uji Menggunakan Algoritma Genetika Multi-populasi Fuzzy Adaptif," *INTEGER J. Inf. Technol.*, vol. 2, no. 1, pp. 60–69, 2017, [Online]. Available: <https://ejournal.itats.ac.id/integer/article/view/98%0Ahttps://ejournal.itats.ac.id/integer/article/download/98/133>
- [8] M. H. Reza, Sugiarto, and A. L. Nurlaili, "Pengujian Menggunakan Black Box Boundary Value Analysis Pada Aplikasi Voucher dan Receipt PT. Samudera Agencies Indonesia," *J. Inform. dan Sist. Inf.*, vol. 2, no. 2, pp. 181–189, 2021.
- [9] 2019 Kiki Hardiansyah, "PEMANTAUAN KEGIATAN SISWA ( Studi Kasus : Madrasah Ibtidaiyah Cahaya Cimahi )," 2019.
- [10] W. E. Perry, *Effective Methods for Software Testing Third Edition*, vol. 3. 2006.

- [11] N. Catur, R. Utama, S. Hariani, E. Wulandari, and E. Rahmawati, "JSIKA Vol. 9, No. 2 Tahun 2020 ISSN 2338 - 137X," vol. 9, no. 2, 2020.
- [12] P. Astuti, "Penggunaan Metode Black Box Testing (Boundary Value Analysis) Pada Sistem Akademik (Sma/Smk)," *Fakt. Exacta*, vol. 11, no. 2, p. 186, 2018, doi: 10.30998/faktorexacta.v11i2.2510.
- [13] M. S. Mustaqbal, R. F. Firdaus, and H. Rahmadi, "PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS (Studi Kasus: Aplikasi Prediksi Kelulusan SNMPTN)," vol. I, no. 3, pp. 31–36, 2015.