

Studi Literatur Kekurangan dan Kelebihan Pengujian Black Box

LITERATURE STUDY OF THE LACK AND EXCESS OF TESTING THE BLACK BOX

Rizky Parlika^{*1}, Tasya Ardhan Nisaa², Shavira Maya Ningrum³, Berlianda Adha Haque⁴

^{1,2} UPN "Veteran" Jawa Timur: jl. Rungkut Madya 1 Gn. Anyar, Surabaya, Indonesia

^{1,2} Jurusan Informatika: UPN Veteran Jawa Timur

e-mail: ^{*1}rizkyparlika.if@upnjatim.ac.id, ²18081010049@student.upnjatim.ac.id,

³18081010050@student.upnjatim.ac.id, ⁴18081010051@student.upnjatim.ac.id

Abstrak

Suatu perangkat lunak tidak akan bisa sempurna dan selalu saja memerlukan pengembangan. Pengembangan perangkat lunak terdiri dari beberapa tahapan, salah satunya yaitu tahapan pengujian. Tahapan ini bertujuan untuk mencari kesalahan pada perangkat lunak ketika melakukan pengembangan perangkat lunak. Tahapan pengujian sendiri dibagi menjadi beberapa teknik, salah satunya yaitu pengujian black box. Pengujian black box adalah pengujian yang hanya menguji bagian luar dari perangkat lunak, contohnya seperti desain antarmuka. Hal tersebut merupakan salah satu alasan pengujian ini layak digunakan untuk menguji luaran suatu perangkat lunak. Makalah ini disusun untuk mencari kekurangan dan kelebihan pengujian black box, persamaan isi, dan perbedaan isi dengan cara mengamati literatur yang telah ada dengan metode pengelompokan dari ketiga hal tersebut. Dilihat dari beberapa literatur yang masuk dalam pengamatan, pengujian black box memiliki banyak kelebihan dan beberapa kekurangan, beberapa isi yang sama, dan beberapa isi yang berbeda.

Kata Kunci — pengujian kotak hitam, kelebihan, kekurangan, persamaan, perbedaan.

Abstract

A software cannot be perfect and always requires development. Software development consists of several stages, one of which is the testing stage. This stage aims to find fault with the software when doing software development. Stages of testing itself is divided into several techniques, one of which is black box testing. Black box testing is testing that only tests the outside of the software, for example, such as interface design. This is one of the reasons this test is appropriate to be used to test the software output. This paper is structured to look for the strengths and weaknesses of black box testing, content similarities, and differences in content by observing existing literature by the method of grouping these three things. Judging from some of the literature included in the observation, black box testing has many advantages and some drawbacks, some of the same contents, and several different contents.

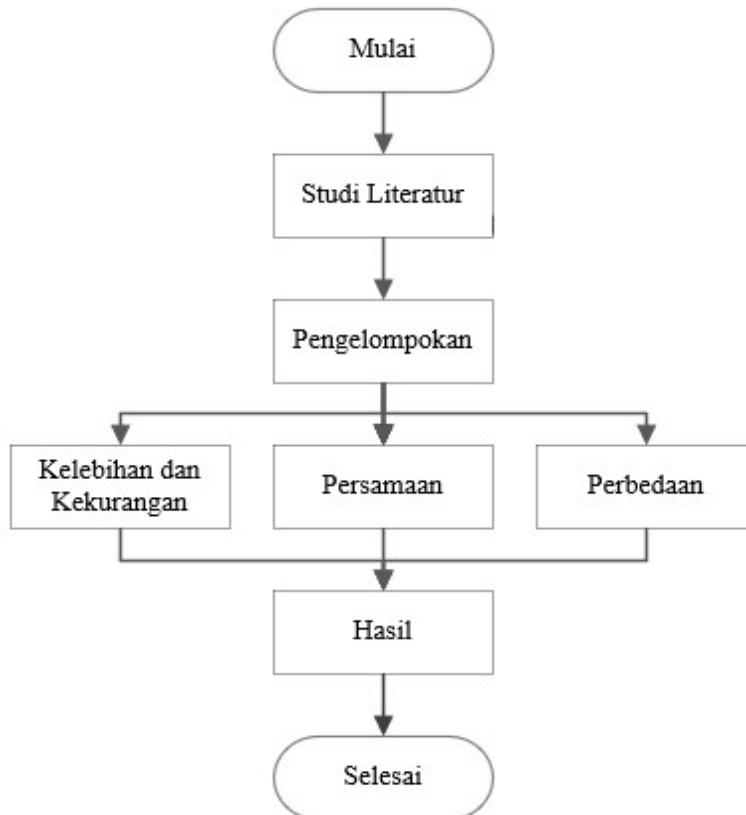
Keywords — black box testing, strengths, weaknesses, similarities, differences.

1. PENDAHULUAN

Siklus hidup pengembangan perangkat lunak atau yang biasa disebut SDLC mempunyai aktivitas yang signifikan yaitu pengujian perangkat lunak [1]. Pengujian perangkat lunak digunakan untuk mendeteksi kegagalan sehingga cacat dapat ditemukan dan diperbaiki, serta dapat juga menguji perangkat lunak dalam segi kualitas [2]. Salah satu pengujian yang sering digunakan adalah pengujian black box [3]. Pengujian black box memiliki banyak kelebihan dan kekurangan. Pada makalah ini, metode perbandingan digunakan sebagai media untuk mengetahui lebih dalam kelebihan dan kekurangan pengujian black box [1][2][3][4][5][6][7]. Dalam pembuatan makalah ini, penulis melakukan pengelompokan menjadi tiga bagian, yaitu kekurangan dan kelebihan black box, persamaan black box, dan perbedaan black box dari literatur tentang pengujian black box yang sudah ada. Dan ketiga bagian tersebut merupakan hasil, yang dimana hasil tersebut adalah tujuan dari disusunnya makalah ini.

2. METODE PENELITIAN

Penulis menggunakan studi literature dalam penelitian ini, penulis juga melakukan pengelompokan terhadap literatur guna mengetahui kekurangan dan kelebihan dari seluruh literatur, serta penulis juga dapat mengetahui apakah ada persamaan dan perbedaan dalam beberapa literatur tersebut.



Gambar 1. Diagram Alir Alur Pembuatan Paper

3. HASIL DAN PEMBAHASAN

Tahapan pengujian suatu perangkat lunak seperti sistem informasi sangatlah penting, sebelum akhirnya perangkat lunak digunakan oleh pengguna [3][8][9]. Ada beberapa teknik penting dalam pengujian yaitu pengujian black box, pengujian white box, dan pengujian gray box. Pengujian black box adalah teknik pengujian tanpa mengacu pada struktur internal dari komponen atau sistem. Pengujian white box adalah teknik pengujian yang didasarkan pada struktur internal dari komponen atau sistem. Pengujian gray box adalah gabungan dari pengujian black box dan white box [2][4].

Pengujian black box adalah salah satu pengujian yang sering digunakan karena pengujian ini tidak perlu mengetahui apa isinya, cukup melakukan pengujian bagian luarnya [3][7][8][9][10][11][12][13][14][15]. Pengujian black box hanya melibatkan antara input dan output [28]. Pengujian ini menangani kebutuhan pelanggan dari input yang valid maupun tidak valid [5].

II-A. Pengujian Black Box

Pengujian black box memiliki peran penting dalam pengujian perangkat lunak yaitu untuk memvalidasi fungsi keseluruhan sistem apakah telah bekerja dengan baik [1][16]. Pengujian black box bersifat dinamis [29].

Seseorang yang menguji perangkat lunak dengan metode pengujian black box tidak perlu memiliki pengetahuan pemrograman / struktur dalam perangkat lunak [9]. Penguji yang menggunakan pengujian black box tidak memiliki akses untuk mengetahui kode sumber dan arsitektur sistem, hanya melalui antarmuka dengan memberikan input dan memeriksa output tanpa mengetahui bagaimana input dioperasikan hingga menjadi sebuah output [6]. Pemodelan black box tergantung pada akurasi yang diinginkan serta struktur opsional dipilih untuk memetakan data yang diukur dari sistem termasuk input dan output [26].

Tingkat keberhasilan suatu pengujian dapat dilihat dari hasil akhir suatu perangkat lunak yang sudah sesuai mulai dari spesifikasi kebutuhan untuk kepuasan pengguna, skenario, dan rancangan [17]. Langkah pertama dalam pengembangan suatu perangkat lunak sebaiknya pilih metode pengujian yang tepat [18]. Maka dari itu perencanaan pengujian dapat dimulai sejak awal proses perangkat lunak dikembangkan [8].

Pengujian black box harus membuat kasus uji dengan dua perbandingan antara benar atau salah. Contoh pada saat pengguna masuk pada aplikasi (perangkat lunak), maka uji kasusnya:

- a) Jika pengguna memasukkan username dan password yang benar.
- b) Jika pengguna memasukkan username yang salah dan password yang benar, atau sebaliknya [12].

II-B. Tujuan Black Box Testing adalah untuk mencari:

1. Fungsi yang salah atau hilang.
2. Kesalahan antarmuka.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.
6. Validasi fungsional.
7. Kesensitifan sistem terhadap nilai input tertentu.
8. Batasan suatu data [2][4][11][12][16][19][30].

Menurut Roger Pressman, kesalahan ditemukan pada 5 poin pertama diatas [19]. Tujuan utama pengujian adalah untuk mencari kesalahan dalam perangkat lunak untuk menghindari suatu kegagalan, atau jika sudah terlanjur gagal maka dapat diperbaiki [20]. Karena itu pengujian black box memiliki langkah pertama untuk memecahkan suatu masalah [27].

II-C. Metode Pengujian Black Box

a) *Equivalence Partitioning*

Equivalence Partitioning adalah sebuah metode pengujian berdasarkan masukan data pada setiap form yang ada pada sistem aplikasi informasi data kinerja, dimana setiap menu masukan akan dilakukan pengujian dan juga dikelompokkan berdasarkan fungsinya baik itu bernilai valid maupun tidak valid [21]. Pengujian *equivalence partitioning* untuk perangkat lunak dapat dirancang dengan memeriksa keluaran dan masukan data [30].

b) *Boundary Value Analysis*

Metode *Boundary Value Analysis* merupakan pengujian yang berfokus pada batas, dimana batas nilai – nilai ekstrim dipilih [22]. *Boundary Value Analysis* adalah metode yang menguji jumlah limit maksimal dan jumlah limit minimal untuk menghasilkan nilai yang valid, yang dinilai cukup relevan [23].

Dasar sejumlah teori yang ada pada *Boundary Value Analysis* :

1. Pada masukan banyak kesalahan yang terjadi.
2. Pengujian batasan nilai masukan diijinkan oleh *Boundary Value Analysis* untuk menyeleksi pada uji kasus.
3. *Boundary Value Analysis* lebih memilih elemen-elemen di dalam kelas ekivalen pada bagian sisi batas dari kelas. *Boundary Value Analysis* merupakan pelengkap dari *Equivalence Partitioning* [23].
4. Contoh:
 - a. Uji $(a-1)$, a , $(a+1)$, $(b-1)$, b , $(b+1)$ untuk rentang yang dibatasi a dan b .
 - b. Uji dengan sejumlah $(n-1)$, n dan $(n+1)$ nilai, jika kondisi masukkan sejumlah n nilai.
 - c. Gunakan dua poin sebelumnya pada kondisi keluaran (buat tabel pengujian yang hasil keluarannya untuk nilai maksimal dan minimal).
 - d. Jika struktur data internal dari program memiliki cakupan (misal: batas array) gunakan data masukan yang menguji batas cakupan [24].

Bentuk dasar dari implementasi *Boundary Value Analysis* adalah untuk menjaga agar satu variabel berada pada nilai nominal dan dapat mengijinkan variabel lain diisikan dengan nilai ekstrimnya [24].

c) *Cause Effect Graph*

Metode *Cause Effect Graphing* adalah metode pengujian yang membantu dalam membangkitkan kasus uji berdasarkan pada hubungan antar *causes (input)* dan *effect (output)* yang terdapat pada spesifikasi kebutuhan perangkat lunak. Metode *Cause Effect Graphing* ini relatif lebih unggul dibandingkan metode lainnya dikarenakan metode ini memperhatikan integrasi antar kombinasi input dan output dan dapat mereduksi kasus uji [22].

Tahapan yang dilakukan dalam membuat kasus uji dengan metode *Cause Effect Graphing*, diantaranya :

1. Mengidentifikasi *cause* dan *effect* dari spesifikasi yang ada. *Cause* merupakan kondisi input dan *effect* merupakan kondisi output atau transformasi sebuah sistem.
2. Menggambarkan hubungan antar *cause* dan *effect* menggunakan grafik *cause* dan *effect*.
3. Mengidentifikasi *constraint* untuk menjelaskan kombinasi antar *cause* dan antar *effect* yang tidak mungkin karena adanya batasan lingkungan.
4. Mengubah grafik *cause* dan *effect* menjadi *decision table* [22].

d) *Random Data Selection*

Memasukkan suatu nilai acak, kemudian dibuat sebuah tabel yang isinya validasi sebuah keluaran [2].

e) *Feature Test*

Melakukan pengujian terhadap spesifikasi yang ada di dalam perangkat lunak. Contohnya seperti sistem informasi (perangkat lunak) akademik, dilakukan pengecekan apakah terdapat fitur untuk melakukan pengisian nilai, pengisian data mahasiswa, dan pengisian data guru [2].

f) *All-Pair Testing*

All-Pair Testing adalah teknik desain uji black box di mana pengujian ini dirancang untuk melaksanakan kombinasi diskrit dari setiap pasang parameter masukan. Tujuan utama pengujian pasangan adalah untuk memiliki satu set kasus uji yang

mencakup semua pasangan. Seperti ada parameter 'p', maka *set test case* akan mencakup $(p-1) + (p-2) + \dots = p(p-1)/2$ pasangan [25].

g) *Fuzzing*

Fuzzing digunakan sebagai teknik pengujian perangkat lunak black box, yang digunakan untuk menemukan bug implementasi menggunakan injeksi data salah bentuk / setengah cacat dalam suatu otomatisasi. *Fuzzing* juga digunakan untuk menguji masalah keamanan dalam perangkat lunak.

Fuzzing juga dapat menyarankan bagian mana dari program yang harus mendapat perhatian khusus. Pengujian *fuzz* digunakan untuk menemukan bug seperti kegagalan pernyataan dan kebocoran memori.

Salah satu keuntungan *fuzzing* adalah desain pengujinya sangat luar biasa, mudah dipahami dan bebas dari konsepsi awal tentang perilaku sistem [25].

h) *Orthogonal Array Testing*

Orthogonal Array Testing atau OAT adalah teknik pengujian black box yang dapat diterapkan untuk masalah di mana masukan domain relatif besar mengakomodasi pengujian yang lengkap [25]. Pengujian ini juga sangat berguna dalam menemukan wilayah kesalahan dengan logika yang salah [30]. *Orthogonal Array Testing* diterapkan dalam pengujian sistem pengujian antarmuka pengguna, pengujian regresi, pengujian konfigurasi, dan pengujian kinerja [25].

Ada manfaat tertentu dari OAT seperti mengurangi waktu siklus pengujian, menyediakan secara seragam cakupan distribusi dari domain uji, dan membuat lebih sedikit kasus uji yang dihasilkan karena set uji singkat [25].

Hasil dari pengelompokan terhadap seluruh literatur, menyebutkan kekurangan dan kelebihan literatur pengujian black box yang tercantum sebagai berikut.

Tabel 1. Kekurangan dan Kelebihan Literatur Pengujian Black Box

Kekurangan	Kelebihan	Sumber Literatur
1. Hasil uji dibesar besarkan dan sering ditaksir terlalu tinggi. 2. Tidak semua produk perangkat lunak dapat diuji. 3. Sebab kegagalan tidak ditemukan. 4. Uji kasus sulit dan juga menantang untuk dirancang jika persyaratan tidak jelas dan singkat. 5. Cakupan kode terbatas. 6. Uji <i>authoring</i> tidak efisien. 7. Tidak bisa mengendalikan dan menargetkan segmen kode.	1. Memproduksi tes black box. 2. Upaya investasi dapat digunakan beberapa kali. 3. Tes dilakukan dari sudut pandang pengguna atau bisa sesuai dengan persyaratan dari pelanggan. 4. Cocok dan efisien digunakan pada sistem atau segmen yang lebih besar. 5. Kode akses tidak diperlukan. 6. Uji kasus dapat dirancang secara cepat dan spesifik.	[1],[2],[3],[4],[5],[6],[7]

Kekurangan	Kelebihan	Sumber Literatur
<p>8. Tes redundan dapat dibentuk jika perangkat lunak perancang / pengembang telah menjalankan <i>case test</i>.</p> <p>9. Untuk menguji segmen kompleks kode pengujian ini tidak efisien.</p>	<p>7. <i>Tester</i> tidak perlu tahu bahasa pemrograman atau bagaimana pelaksanaan program.</p> <p>8. Membantu untuk mengeksplosi setiap ambiguitas atau inkonsistensi dalam persyaratan spesifikasi.</p> <p>9. Mudah dieksekusi.</p> <p>10. Hasil dari pengujian black box dapat memperjelas kontradiksi ataupun kerancuan yang mungkin timbul pada saat eksekusi.</p>	

Hasil dari pengelompokan terhadap seluruh literatur, beberapa dari literatur tersebut memiliki persamaan yang tercantum sebagai berikut.

Tabel 2. Persamaan Literatur Pengujian Black Box

Persamaan	Pada Literatur	Penjelasan
Menyelesaikan masalah dengan cara memisahkan perspektif melalui pemisahan <i>quality assurance</i> .	[4],[6]	Disebutkan pada literatur [4] bahwa “Rekomendasi Pengujian: jelas memisahkan perspektif pengguna dari mengembangkan perspektif melalui pemisahan QA (<i>quality assurance</i>) dan tanggung jawab pembangunan” dan juga pada literatur [6] bahwa “ Pengujian yang Tidak Cocok - dengan jelas memisahkan perspektif pengguna dari perspektif pengembang melalui pemisahan tanggung jawab QA (<i>quality assurance</i>) dan Pengembangan.”
Pengujian cocok dan efisien untuk sistem atau segmen yang besar	[4],[5],[6]	Disebutkan pada literatur [4] bahwa “Efisien Pengujian: Cocok dan efisien untuk segmen kode besar Efisien Pengujian: Cocok dan efisien untuk segmen kode besar atau unit. ” , pada literatur [5] bahwa “ Pengujian efisien ketika digunakan pada sistem yang lebih besar.” dan juga pada literatur [6] bahwa “ Pengujian Efisien - Sangat cocok dan efisien untuk segmen atau unit kode besar.”
Pengujian black box bisa bukan orang teknis dari bidang pemrograman.	[2],[3],[5]	Disebutkan pada literatur [2] bahwa “ Pada pengujian black box tidak perlu untuk tester untuk memiliki pengetahuan pemrograman

Persamaan	Pada Literatur	Penjelasan
		yang baik, karena hanya mengkaji aspek fundamental dari sistem tanpa masuk ke detail ” , pada literatur [3] bahwa “ Anggota tim tester tidak harus dari seseorang yang memiliki kemampuan teknis di bidang pemrograman. ” dan juga pada literatur [5] bahwa “ Pengujii bisa bukan orang teknis. Pemrograman dan pengetahuan implementasi tidak diperlukan untuk teknik pengujian ini. ”
Cakupan kode terbatas	[4],[6]	Disebutkan pada literatur [4] bahwa “Cakupan kode terbatas.” dan juga pada literatur [6] bahwa “ Pengujian Lokal - Cakupan jalur kode terbatas karena hanya sejumlah kecil input pengujian yang benar-benar diuji.”
Uji kasus sulit untuk dirancang	[2],[5]	Disebutkan pada literatur [2] bahwa “Uji kasus sulit dan juga menantang untuk dirancang.” dan juga pada literatur [5] bahwa “Sangat sulit untuk merancang kasus uji jika persyaratannya tidak jelas dan singkat.”
Tidak dapat menargetkan segmen kode	[4],[6]	Disebutkan pada literatur [4] bahwa “Tidak bisa mengendalikan dan menargetkan segmen kode.” dan juga pada literatur [6] bahwa “Cakupan Buta - tidak dapat menargetkan jalur kode atau mengontrol segmen yang mungkin lebih rentan kesalahan daripada yang lain.”
Kode akses tidak diperlukan	[4],[6]	Disebutkan pada literatur [4] bahwa “Kode akses tidak diperlukan.” dan juga pada literatur [6] bahwa “ <i>Non intrusif</i> - akses kode tidak diperlukan.”

Hasil dari pengelompokan terhadap seluruh literatur, beberapa dari literatur tersebut memiliki perbedaan yang tercantum sebagai berikut.

Tabel 3. Perbedaan Literatur Pengujian Black Box

Perbedaan	Pada Literatur	Penjelasan
Cara penyelesaian masalah.	[3],[10]	Disebutkan pada literatur [3] bahwa “metode yang digunakan yaitu dengan menggunakan checklist (angket).” sedangkan pada literatur [10] bahwa “model pendekatan dan pengembangan sistem ini menggunakan model prototype.”

Perbedaan	Pada Literatur	Penjelasan
Sudut pandang pengujian.	[2],[5]	Disebutkan pada literatur [2] bahwa “Tes dilakukan dari sudut pengguna pandang” sedangkan pada literatur [5] bahwa “Pengujian dilakukan sesuai dengan sudut pandang pelanggan persyaratan.”

4. KESIMPULAN

Pengujian perangkat lunak menggunakan teknik black box adalah salah satu hal yang penting, karena sebelum perangkat lunak benar-benar digunakan, pengujii harus melakukan pengecekan apakah ada kesalahan atau kecacatan pada perangkat lunak guna menghindari suatu kegagalan. Pada makalah ini, penulis memaparkan hasil pengamatan dari 30 literatur. Yang pertama dapat dilihat dari tabel 1 menyatakan bahwa ada 9 kekurangan dan 10 kelebihan black box yang bersumber dari 7 literatur, maka terbukti lebih banyak kelebihan dari black box daripada kekurangannya yang menyebabkan pengujian black box sangat layak dan sangat diperlukan untuk menguji luaran suatu perangkat lunak. Yang kedua pada tabel 2 yaitu berisi persamaan literatur yang menyatakan bahwa ada 2 literatur yang mempunyai banyak persamaan di bagian menyelesaikan masalah dengan cara memisahkan perspektif melalui pemisahan *quality assurance*, pengujian cocok dan efisien untuk sistem atau segmen yang besar, cakupan kode terbatas, tidak dapat menargetkan segmen kode, dan kode akses tidak diperlukan. Dan yang terakhir pada tabel 3 berisi perbedaan literatur yang menyatakan bahwa ada 2 bagian yang mempunyai perbedaan yaitu pada bagian cara penyelesaian masalah dan sudut pandang pengujian.

UCAPAN TERIMA KASIH

Ucapan terima kasih penulis sampaikan untuk informatika dan seluruh peserta mata kuliah Rekayasa Perangkat Lunak baik dosen maupun mahasiswa, serta kepada editorial Teknomatika yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] S. Nidhra, “Black Box and White Box Testing Techniques - A Literature Review,” *Int. J. Embed. Syst. Appl.*, vol. 2, no. 2, pp. 29–50, 2012, doi: 10.5121/ijesa.2012.2204.
- [2] M. Kumar, A. Professor, S. Kumar Singh, R. K. Dwivedi, and A. Professor, “A Comparative Study of Black Box Testing and White Box Testing Techniques,” *Int. J. Adv. Res. Comput. Sci. Manag. Stud.*, vol. 3, no. 10, pp. 32–44, 2015.
- [3] M. Fatkhurrokhman, “Analisis Pengujian Sistem Informasi Akademik STMIK El Rahma Yogyakarta menggunakan International Organization for Standardization (ISO 9126),” *Dipetik Februari*, vol. 6, no. Iso 9126, p. 2016, 2014.
- [4] S. Acharya and V. Pandya, “Bridge between Black Box and White Box - Gray Box Testing Technique,” *Int. J. Electron. Comput. Sci. Eng.*, vol. 2, no. 1, pp. 175–85, 2013.
- [5] A. Verma, A. Khatana, and S. Chaudhary, “A Comparative Study of Black Box Testing and White Box Testing,” *Int. J. Comput. Sci. Eng.*, vol. 5, no. 12, pp. 301–304, 2017, doi: 10.26438/ijcse/v5i12.301304.
- [6] B. M. Yunus, R. Mallal, and C. Networks, “SOA Testing using Black , White and Gray Box Techniques I . Introduction Key Concepts.”

- [7] I. T. Sevandri, R. K. Dewi, and M. T. Ananta, "Implementasi Algoritma Topsis Pada Sistem Rekomendasi Pencarian Lokasi Gym Berbasis Android (Studi Kasus : Kota Malang)," vol. 3, no. 4, pp. 3182–3190, 2019.
- [8] Herdiana, "Don't No Title No Title," *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2013, doi: 10.1017/CBO9781107415324.004.
- [9] U. Menampilkan *et al.*, "Rancang Bangun Basis Data Perguruan Tinggi Untuk Menampilkan Jadwal Kuliah," *Semin. Nas. Teknol. Terap. Berbas. Kearifan Lokal*, pp. 359–366, 2015.
- [10] P. Handoko and A. A. Setiawan, "Pengembangan Aplikasi Analisis Penampang Beton Bertulang Berbasis Android," *Din. Rekayasa*, vol. 13, no. 2, p. 69, 2017, doi: 10.20884/1.dr.2017.13.2.182.
- [11] R. Khoirunnisa, R. R. Isnanto, and K. T. Martono, "Pembuatan Aplikasi Web Manajemen Laundry dan Integrasi Data dengan Web Service," *J. Teknol. dan Sist. Komput.*, vol. 4, no. 1, p. 93, 2016, doi: 10.14710/jtsiskom.4.1.2016.93-101.
- [12] Y. B. Utomo, Herlawati, and E. G. Sihombing, "Animasi Interaktif Pengenalan Pakaian Adat Tradisional Pada Smp Pgri 1 Bekasi," *Tek. Komput. Amik Bsi*, vol. 1, no. 1, p. 66, 2015.
- [13] N. Astiani, D. Andreswari, and Y. Setiawan, "Aplikasi Sistem Pendukung Keputusan Tanaman Obat Herbal Untuk Berbagai Penyakit Dengan Metode Roc (Rank Order Centroid) Dan Metode Oreste Berbasis Mobile Web," *J. Inform.*, vol. 12, no. 2, 2016, doi: 10.21460/inf.2016.122.486.
- [14] A. Kholis and Herlawati, "Animasi Interaktif Pembelajaran Tajwid Pada Taman Qur'an Anak (Tqa) Al Washilah Cirebon," *J. Tek. Komput. Amik Bsi*, no. 2, pp. 159–170, 2015.
- [15] P. Pahwa and R. Miglani, "Test Case Design using Black Box Testing Techniques for Data Mart," *Int. J. Comput. Appl.*, vol. 109, no. 3, pp. 18–22, 2015, doi: 10.5120/19169-0636.
- [16] T. Wahyuningrum and D. Dwi Januarita, "Implementasi dan Pengujian Web E-commerce untuk Produk Unggulan Desa," *J. Komput. Terap.*, vol. 1, no. 1, pp. 57–66, 2015.
- [17] P. Widyaningrum, R. R. Isnanto, and K. T. Martono, "Pengembangan Simulasi Peternakan Sapi Perah dengan Game Maker Berbasis Windows," *J. Teknol. dan Sist. Komput.*, vol. 1, no. 4, pp. 162–168, 2013, doi: 10.14710/JTSISKOM.1.4.2013.162-168.
- [18] S. Beydeda, "Hitam putih-Box Self-pengujian COTS Komponen."
- [19] E. Auliasin, D. S. Rusdianto, and A. A. Soebroto, "Pengembangan Aplikasi Diagnosis Gejala Depresi pada Mahasiswa Fakultas Ilmu Komputer (Studi Kasus : Fakultas Ilmu Komputer Universitas Brawijaya)No Title," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 9, pp. 8823–8830, 2019.
- [20] H. Bhasin, E. Khanna, and S. Sudha, "Black Box Testing based on Requirement Analysis and Design Specifications," *Int. J. Comput. Appl.*, vol. 87, no. 18, pp. 36–40, 2014, doi: 10.5120/15311-4024.
- [21] A. R. Sinulingga, M. Zuhri, R. B. Mukti, Z. Syifa, and A. Saifudin, "Pengujian Black Box pada Sistem Aplikasi Informasi Data Kinerja Menggunakan Teknik Equivalence Partitions," *J. Teknol. Sist. Inf. dan Apl.*, vol. 3, no. 1, p. 9, 2020, doi: 10.32493/jtsi.v3i1.4303.
- [22] M. Nurudin, W. Jayanti, R. D. Saputro, M. P. Saputra, and Y. Yulianti, "Pengujian Black Box pada Aplikasi Penjualan Berbasis Web Menggunakan Teknik Boundary Value Analysis," *J. Inform. Univ. Pamulang*, vol. 4, no. 4, p. 143, 2019, doi: 10.32493/informatika.v4i4.3841.
- [23] Yasuhito Ueda *et al.*, "3. A case of intrabronchial hamartoma with recurrent obstructive pneumonia (The 23rd Annual Meeting of the Japanese Society of Respiratory Endoscopy Chugoku-Shikoku Branch)," *J. Japan Soc. Respir. Endosc.*, vol. 37, no. 3, p. 343, 2015,

- doi: 10.18907/jjsre.37.3_343_4.
- [24] 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.
 - [25] M. E. Khan, “Different approaches to white box testing technique for finding errors,” *Int. J. Softw. Eng. its Appl.*, vol. 5, no. 3, pp. 1–14, 2011, doi: 10.5121/ijsea.2011.2404.
 - [26] B. Zaker, G. B. Gharehpetian, M. Mirsalim, and N. Moaddabi, “PMU-based linear and nonlinear black-box modelling of power systems,” *2013 21st Iran. Conf. Electr. Eng. ICEE 2013*, 2013, doi: 10.1109/IranianCEE.2013.6599578.
 - [27] M. C. S. Reddy, K. B. Reddy, P. A. K. Reddy, and E. Venkataramana, “Black box for vehicles,” vol. 1, no. 7, pp. 6–12, 2016.
 - [28] P. Michael *et al.*, “Membuka ‘ Black Box ’ dari ‘ Black Box ’: The Metafora dari ‘ Black Box ’ dan penggunaannya dalam STS,” pp. 1–32, 2007
 - [29] S. Arlt, E. Ermis, S. Feo-Arenis, and A. Podelski, “Verification of GUI applications: A black-box approach,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8802, pp. 236–252, 2014, doi: 10.1007/978-3-662-45234-9_17.
 - [30] V. Chandra, “Teori Fuzzy di Black Box Testing,” vol. 2, no. Ijarcst, pp. 289–291, 2014.
-