

# Survei Teknik Pengujian Software

## Software Testing Engineering Survey

<sup>1</sup>Ilham Rafif Dhaifullah, <sup>2</sup>M Muttanifudin H, <sup>3</sup>Aulia Ananda Salsabila, <sup>4</sup>Muhammad Ainul Yakin  
<sup>1,2,3</sup> Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana  
Malik Ibrahim Malang  
e-mail: <sup>1</sup> 18650006@student.uin-malang.ac.id; <sup>2</sup> 18650014@student.uin-malang.ac.id; <sup>3</sup>  
18650032@student.uin-malang.ac.id; <sup>4</sup> yaqinov@ti.uin-malang.ac.id

### Abstrak

Salah satu tahapan dalam rekayasa perangkat lunak adalah tahap pengujian. Tahap pengujian tidak sepenuhnya menjamin kualitas, akan tetapi pengujian perangkat lunak dapat memberikan kepercayaan lebih terhadap pengguna perangkat lunak. Developer perangkat lunak pemula terkadang tidak melakukan pengujian secara detail. Studi ini bertujuan mengetahui teknik pengujian yang digunakan oleh pengembang perangkat lunak. Selain itu, untuk mengetahui faktor-faktor pendukung dan penghambat dalam pengujian. Dalam studi ini dalam melakukan pengujian perangkat lunak menggunakan tiga aspek pengujian yakni black-box testing, white-box testing, dan grey-box testing. Hasil dari penelitian ini dapat memberikan pertimbangan untuk melakukan pemilihan dan penggunaan metodologi yang tepat berdasarkan kebutuhan, kelebihan dan kelemahan, juga faktor-faktor penilaian yang lain seperti keakraban dengan teknologi, kompleksitas sistem, keandalan sistem, waktu yang singkat dan tepat, hingga mereferensi beberapa jurnal ilmiah

**Katakunci:** Teknik, pengujian, black-box testing, white-box testing, grey-box testing

### Abstrack

One of the stages in software engineering is the testing stage. The testing phase does not fully guarantee quality, but software testing can give more confidence to software users. Beginner software developers sometimes don't do detailed testing. This study aims to determine the testing techniques used by software developers. In addition, to determine the supporting and inhibiting factors in the test. In this study, software testing uses three aspects of testing, namely black-box testing, white-box testing, and gray-box testing. The results of this research can provide consideration for selecting and using the right methodology based on needs, strengths and weaknesses, as well as other assessment factors such as familiarity with technology, system complexity, system reliability, short and appropriate time, to referencing several journals. scientific

**Keyword:** Engineering, testing, black-box testing, white-box testing, gray-box testing

## 1. PENDAHULUAN

Secara definisi survei adalah penelitian kuantitatif melalui pertanyaan terstruktur yang kepada setiap orang yang kemudian semua jawaban tersebut akan dicatat, kemudian diolah, dan pada akhirnya akan dianalisis oleh peneliti[1]. Istilah survei biasanya dirancukan dengan istilah observasi dalam pengertian sehari-hari. Padahal kedua istilah tersebut mempunyai pengertian yang berbeda, walaupun keduanya merupakan kegiatan yang saling berhubungan. Menurut kamus Webster, pengertian survei adalah suatu kondisi tertentu yang menghendaki kepastian informasi, terutama bagi orang – orang yang bertanggung jawab atau yang tertarik[2]. Sehingga dapat di simpulkan bahwa survei merupakan suatu aktivitas atau kegiatan penelitian yang dilakukan untuk mendapatkan suatu kepastian informasi (seperti : jumlah orang, persepsi atau pesan-pesan tertentu), dengan cara mengambil sampel dari satu populasi dan menggunakan kuesioner sebagai alat pengumpul data yang pokok.

Software atau perangkat lunak merupakan suatu istilah umum yang berfungsi untuk menunjukkan informasi data dalam komputer serta instruksi yang terorganisir[3]. Dalam hal ini,

History of article:

Received: Juni, 2022 : Accepted: Juni, 2022

software memiliki tanggung jawab untuk mengendalikan, mengelola, serta mengintegrasikan komponen hardware dari suatu sistem komputer dengan tujuan untuk menyelesaikan tugas atau perintah tertentu.

Pengujian software merupakan suatu proses menjalankan program dengan maksud menemukan kesalahan[4]. Definisi tersebut menyangkut kegiatan mulai dari cek kode yang dilakukan oleh seorang pemimpin tim untuk menjalankan percobaan dari perangkat lunak yang dilakukan oleh seorang rekan, serta tes yang dilakukan oleh suatu unit pengujian, semua bisa dianggap kegiatan pengujian. Dari beberapa survey yang telah peneliti lakukan, secara garis besar teknik pengujian survey dibagi menjadi 3 yakni white-box testing, black-box testing, dan grey-box testing. Dalam white-box testing teknik pengujiannya dibagi lagi menjadi beberapa macam yakni Control Flow Testing[5], Statement and Branch Coverage Testing[6], Path Testing[7], Data Flow Testing[8], dan Loop Testing[9]. Kemudian untuk black-box testing dibagi menjadi 6 metode yakni equivalence partitioning dan boundary value analysis[10], cause effect graphing[11], fuzzy testing[12], model based testing[13]. Selanjutnya untuk grey-box testing dibagi menjadi 4 metode yakni orthogonal array testing, automatic test pattern, matrix test, dan regression test[11]. Masing-masing tes teknik pengujian software memiliki perbedaan dan persamaan. Jadi, dalam survey ini akan dijelaskan mengenai perbedaan dan persamaan teknik pengujian software yang kemudian dibandingkan untuk mendapatkan hasil dari teknik pengujian software yang paling unggul.

## 2. METODE PENELITIAN

*Software Testing Life Cycle* (STLC) adalah tahap-tahap proses pengujian yang dilaksanakan secara sistematis dan terencana. Dalam proses STLC, berbagai kegiatan dilakukan untuk meningkatkan kualitas produk. STLC mengacu pada tahap-tahap yang spesifik dalam proses pengujian untuk memastikan kualitas produk. Setiap tahap-tahap dari STLC memiliki *Entry Criteria* (ketentuan yang harus dimiliki sebelum pengujian dimulai), *Exit Criteria* (ketentuan yang harus diselesaikan sebelum menyelesaikan pengujian), *Activities* dan *Deliverable* (hasil yang didapatkan setelah menyelesaikan pengujian) yang terkait.



Gambar 1. Tahap STLC (*System Testing Life Cycle*)

### Requirement Analysis

*Requirement analysis* adalah tahap pertama dalam *software testing life cycle* atau STLC. *Requirement software* yang sudah ada dari *stakeholder* (*system analyst, product manager, etc*)

kemudian dianalisa oleh tim *QA* untuk mengetahui detail *software*, *module*, fitur dan fungsi yang akan dibangun, *review* dan melakukan validasi jika masih ada kekurangan dan atau untuk melengkapi kejelasan *requirement* tersebut, menganalisa apa saja yang dapat diuji (*testable*) secara manual dan *automated* serta menganalisa cakupan fungsi atau fitur mana saja yang akan diuji secara *functional* dan *non-functional*.

### Test Planning

Tahap *planning* adalah tahap kedua dari proses *STLC* yang juga dapat disebut dengan *test planning* atau *test strategy*. Pada tahap ini tim *QA* mempersiapkan rencana untuk melakukan pengujian berdasarkan *requirement analysis*. Pada tahap ini akan ditentukan *tools* yang akan digunakan untuk menguji dan hal hal yang di *deliver* setelah pengujian, estimasi waktu dan sumber daya, pembagian peran serta tanggung jawab pada tim *QA*. Design

### Test Case Development

Pada tahap selanjutnya adalah *development*. Tahap ini berisi hal-hal yang akan menjadi acuan dalam pengujian yang melibatkan untuk membuat *test case*, membuat *test data*, membuat *script automation test* berdasarkan *test case*. *Test Case* adalah sekumpulan skenario yang disusun oleh *QA* agar sistem yang akan dites dapat memenuhi ketentuan, standar tertentu serta dapat berfungsi dengan baik.

### Test Environment Setup

Pada tahap ini proses yang dilakukan adalah memastikan *environment test* dapat berjalan dengan baik dan sesuai harapan. Pada dasarnya *environment test* menentukan syarat *software* yang diujikan mulai melakukan *smoke test* untuk memastikan *environment test* apakah sudah benar-benar siap. *Smoke test* adalah testing yang dilakukan untuk memastikan fitur atau fungsi penting dari suatu program berjalan dengan baik sebelum dilakukannya tes fungsional atau regression.

### Test Execution

Tahap selanjutnya yaitu *test execution*, setelah *software* siap uji dan persiapan pengujian siap. Pada tahap ini pengujian dilakukan berdasarkan *test plan* dan *test case* yang disepakati di tahap sebelumnya. Fitur yang berjalan sesuai dengan *test requirement*, status fitur tersebut adalah *pass* atau berhasil, dan siap untuk masuk ke tahap *deployment*. Jika fungsi tidak berjalan sesuai dengan *requirement*, maka tergantung dari kategori *error* atau *bug* yang terjadi. *Error* yang ditemukan tersebut dimasukkan ke dalam *test report* tim *QA* untuk disampaikan ke tim *developer* untuk diperbaiki, dan kembali diuji oleh tim *QA*. Pada tahap ini juga dilakukan *sanity testing* dan *regression testing* ketika akan melakukan *deployment program*.

### Test Cycle Closure

Pada tahap akhir dilakukan *meeting* anggota tim untuk mengidentifikasi dan menganalisis rencana yang akan dilakukan di masa depan dan evaluasi terhadap *STLC* yang dijalankan saat ini. Tujuannya adalah untuk meminimalkan hambatan saat proses pengujian dan meningkatkan kualitas dalam menjalankan *STLC* di masa depan.

Pada penelitian ini, penulis menggunakan pengujian perangkat lunak dengan menggunakan teknik pengujian *white-box*, *black-box*, dan *grey-box*. *White box testing* merupakan pengujian yang difokuskan pada internal sistem yaitu source code program. Tujuan dari pengujian *white box* digunakan sebagai alat uji kompleksitas dari code program. Bagi programmer, *white box* sangat penting untuk menentukan kompleksitas dari suatu code. Pengujian *white box* juga dapat digunakan sebagai validasi apakah source code mengikuti desain; apakah source code sesuai dengan kebutuhan fungsional; apakah source code memiliki kerentanan.

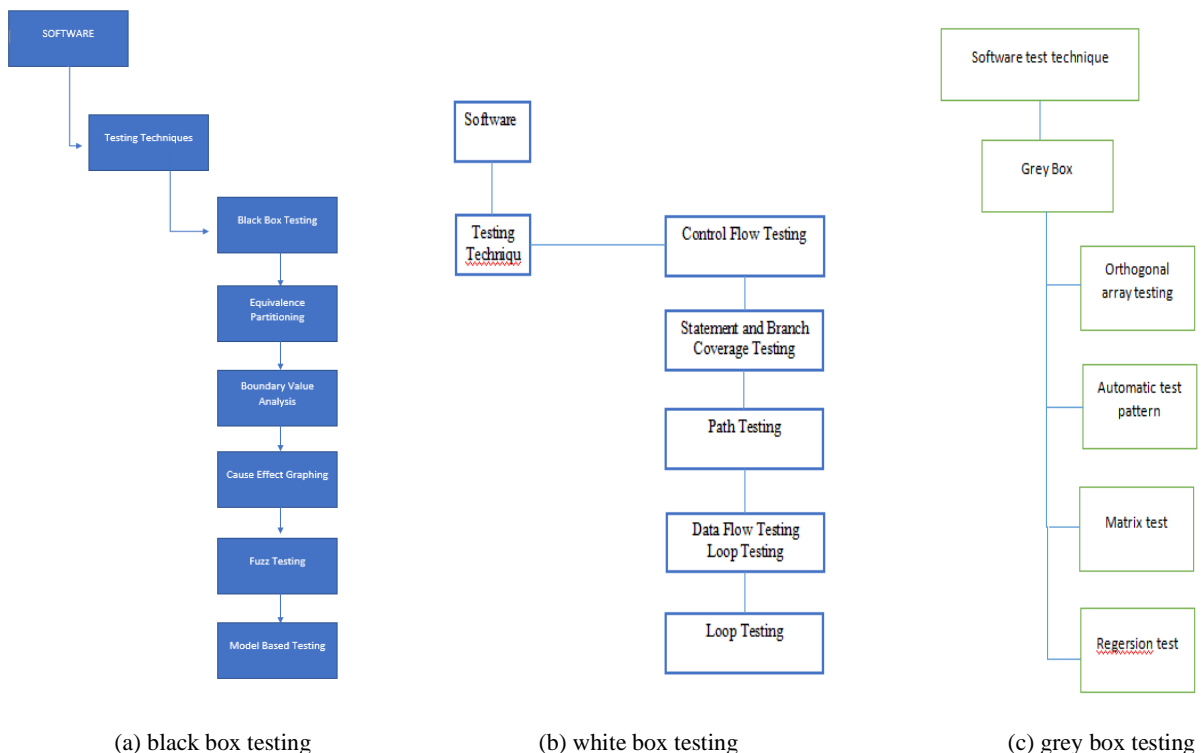
Pengujian *black box* biasanya digunakan untuk menguji pekerjaan internal aplikasi tanpa pengetahuan pemrograman. Pengujian *black box* digunakan untuk menguji

fungsional maupun input output pada aplikasi. Teknik pengujian ini ditujukan kepada para penguji yang tidak memiliki pemahaman dalam pemrograman. Biasanya, pengujian black box diterapkan di semua level pengujian perangkat lunak seperti: unit testing, integration testing, functional testing dan acceptance testing. Fokus utama dalam pengujian black box adalah mengetahui input dalam sistem; luaran yang diharapkan; dan hasil nyata berdasarkan input dari program.

Sedangkan pengujian grey box merupakan sebuah kombinasi pengujian antara white-box dan black-box. pengujian ini menguji software berdasarkan spesifikasi tetapi menggunakan cara kerja dari dalam.

### 3. HASIL DAN PEMBAHASAN

Secara garis besar, teknik-teknik pengujian software ada 3 macam yakni *White-Box Testing*, *Black-box Testing* dan *Grey-Box Testing*. Gambar 2 taksonomi pengujian software.



**Gambar 2.** Taksonomi pengujian software

#### White box testing

White box testing merupakan pengujian yang difokuskan pada internal sistem yaitu *source code* program. Tujuan dari pengujian white box digunakan sebagai alat uji kompleksitas dari code program. Bagi programmer, white box sangat penting untuk menentukan kompleksitas dari suatu code. Pengujian white box juga dapat digunakan sebagai validasi apakah source code mengikuti desain; apakah source code sesuai dengan kebutuhan fungsional; apakah source code memiliki kerentanan. Tabel 1 adalah klasifikasi hasil persamaan pengujian secara white box

**Table 1.** klasifikasi persamaan metode teknik pengujian white box

Perbedaan	Kompleksitas Algoritma	Ukuran Kompleksitas	Struktur alur Algoritma
-----------	------------------------	---------------------	-------------------------

Control Flow	Ya	Tidak	Tidak
Basic Path	Ya	Ya	Tidak
Data Flow	Tidak	Tidak	Ya
Cyclomatic complexity	Ya	Ya	Tidak

### Black box testing

Black Box adalah pengujian tanpa sepengetahuan kerja internal aplikasi yang sedang diuji (AUT). Juga dikenal sebagai pengujian fungsional atau pengujian input driven. Teknik pengujian perangkat lunak di mana cara kerja internal dari item yang diuji tidak diketahui oleh tester. Pengujian secara black box dapat dilakukan dengan 6 cara, yaitu equivalence partitioning, boundary value analysis, cause effect graphing, fuzzy testing, dan model based testing. Table 2 merupakan persamaan dan perbedaan cara pada black box testing.

**Table 2.** klasifikasi persamaan metode teknik pengujian black box

Perbedaan	Kesalahan Kinerja	Kesalahan Interface	Fungsionalnya
Equivalence Partitioning	Ya	Tidak	Ya
Boundary Value Analysis	Ya	Tidak	Ya
Cause Effect Graphing	Ya	Ya	Ya
Fuzz Testing	Ya	Tidak	Ya
Model Based Testing	Ya	Ya	Ya

### Grey box testing

Grey Box Testing adalah sebuah metodologi kombinasi dari Black Box dan White Box Testing, menguji software berdasarkan spesifikasi tetapi menggunakan cara kerja dari dalam. Grey Box dapat di gunakan dengan baik dalam pengujian tim. Klasifikasi pengujian gray box testing menurut persamaan dan perbedaannya pun terdapat pada Tabel 3.

**Table 3.** Persamaan pengujian gray box

Perbedaan	Interface Program	Status Program	Alur Program	Fungsional
Matrix Test	Tidak	Ya	Tidak	Ya
Regression Test	Tidak	Ya	Ya	Tidak
Pattern Test	Ya	Tidak	Ya	Ya
Orthogonal Array Test	Ya	Ya	Ya	Ya

### Klasifikasi dari Implementasi Teknik Pengujian Software

Pada bagian ini penulis akan mengklasifikasikan beberapa jurnal yang telah dikumpulkan, berdasarkan dengan metode yang telah dilakukan. Table 4 merupakan rangkuman akan pengujian white box

**Tabel 4.** White box testing

Judul Jurnal	Penulis	Klasifikasi	Alasan
Implementasi Proses Uji Sistem Informasi Admisi Pascasarjana Universitas Sam Ratulangi	Eko Pandara, Stanley Karouw, Meicsy Najoon	Path Testing	Adanya hubungan pengimplementasian path testing
Testing dan Implementasi Sistem Edisi Pertama	Romeo, S.T.	Loop Testing	Adanya penjelasan parameter dalam setiap kelas pada loop testing
Perancangan Dan Pembuatan Case Tool Software Testing Menggunakan Metode Static Data Flow Analysis (Studi Kasus Pogram Berbasis C++)	Yudhi Purwananto, Arif Bramantoro, Luluk Harini	Control Flow Testing	Hubungan antara control flow testing dan hasil yang diinginkan tergantung pada satu atau lebih parameter
Object-Based Data Flow Testing of Web Applications	Chien-Hung Liu, David C. Kung, Pei Hsia, Chih-Tung Hsu	Data Flow Testing	Hubungan data flow testing dengan pengimplementasiannya pada sebuah software
Dynamic Partitioning and Additional Branch Coverage for Test Case Selection	Arnaldo Marulitua Sinaga, Arie Satia Dharma, Oscar Hutajulu, Anita Ginting and Gustina Simanjuntak	Statement and Branch Coverage Testing	Jurnal ini membahas tentang branch coverage testing dan percobaan-percobaan yang dilakukan menggunakan branch coverage testing

Sedangkan rangkuman mengenai implementasi Teknik pengujian black box terangkum pada Tabel 5.

**Tabel 5.** Black box testing

Judul Jurnal	Penulis	Klasifikasi	Alasan
Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis	Tri Snadhika Jaya	Boundary Value Analysis	Menggunakan klasifikasi kode yang memuat data
Spesification-based testing using cause-effect graphs	Amit Paradkar, K.C. Tai and M.A. Vouk	Cause-Effect Graph Testing	Pengujian yang menggunakan notasi Boolean dalam menentukan hasil dari testing.
A Comparative Study of Software Testing Techniques Viz. White Box Testing, Black Box Testing, Grey Box Testing	Taraq Hussain and Dr. Satyaveer Singh	Fuzz Testing	Uji software dengan memasukkan data random atau <i>Fuzz</i> yang mana akan menelusuri program yang error serta bug

Bridge between Black Box and White Box-Gray Box Testing Techniques	Shivani Acharya and Vidhi Pandya	Model-based Testing	Prosedur Test yang menggunakan model dari system requirement software tersebut
--	----------------------------------	---------------------	--

Sedangkan rangkuman mengenai implementasi Teknik pengujian grey box terangkum pada Tabel 6.

**Tabel 6.** Grey box testing

Judul Jurnal	Penulis	Klasifikasi	Alasan
Regressiontesting minimization,selection andprioritization: asurvey	S. Yoo*,† and M. Harman	Regression Test	Karena jurnal ini berisi tentang penggunaan regression test
Orthogonal Array application for optimal combination of software defect detection techniques choices	Ljubomir Lazic, Nikos Mastorakis	Orthogonal Array Test	Karena Berisi tentang implementasi dari orthogonal array test
Two-Sample Covariance Matrix Testing And Support Recovery	Tony Cai, Weidong Liu and Yin Xia	Matrix Test	Karena pada jurnal ini berisi banyak penjelasan serta contoh tentang matrix test
Random Pattern Testing Versus Deterministic Testing of RAM's	Rene David, Antoine Fuentes	Pattern Test	Pada jurnal ini dijelaskan dengan secara detil dari fungsi serta contoh dari pattern test

#### 4. KESIMPULAN

Pengujian Grey Box ini cocok untuk Aplikasi WEB. Aplikasi Web telah didistribusikan jaringan atau system, karena tidak adanya sumber kode atau binary, tidak mungkin untuk menggunakan pengujian, white-box. Pengujian Black-box juga tidak digunakan karena hanya melakukan kontrak Antara pelanggan dan pengembang, sehingga lebih efisien menggunakan Grey-box testing sebagai informasi penting yang tersedia dalam Web Service Definition Language (WSDL)

#### DAFTAR PUSTAKA

- [1] A. A. Setiyaki *et al.*, "Pengadaan Pentas Seni dan Budaya oleh Unit Kegiatan Mahasiswa Institute Teknologi Bandung pada Masa Pandemi Covid-19," *Sublim Pendidik. Seni Budaya dan Pedagog.*, vol. 1, no. 1, 2021.
- [2] T. Masyitah and H. Malau, "Pelaksanaan Pengawasan Inspektorat Terhadap Distribusi Bantuan Covid-19 di Kota Bukittinggi Provinsi Sumatera Barat," *J. Pendidik. Tambusai*, vol. 5, no. 2, pp. 4291–4296, 2021.
- [3] D. L. Kaligis and R. R. Fatri, "Pengembangan Tampilan Antarmuka Aplikasi Survei Berbasis Web Dengan Metode User Centered Design," *JUST IT J. Sist. Informasi, Teknol. Inf. dan Komput.*, vol. 10, no. 2, p. 106, 2020, doi: 10.24853/justit.10.2.106-114.
- [4] A. N. Hasibuan and T. Dirgahayu, "Pengujian dengan Unit Testing dan Test case pada Proyek Pengembangan Modul Manajemen Pengguna," *Automata*, vol. 2, no. 1, 2021, [Online]. Available: <https://journal.uir.ac.id/AUTOMATA/article/view/17367>.

- [5] A. Pamuji, "Strategi Perbaikan Uji Coba Struktural Perangkat Lunak Pada Metode White-Box," *J. Inform.*, vol. 5, no. 1, pp. 112–118, 2018, doi: 10.31311/ji.v5i1.2893.
- [6] T. Afzal, A. Nadeem, M. Sindhu, and Q. Uz Zaman, "Test case prioritization based on path complexity," *Proc. - 2019 Int. Conf. Front. Inf. Technol. FIT 2019*, no. 1, pp. 363–368, 2019, doi: 10.1109/FIT47737.2019.00074.
- [7] M. F. Londjo, "Implementasi White Box Testing Dengan Teknik Basis Path Pada Pengujian Form Login," *J. Siliwaangi*, vol. 7, no. 2, pp. 35–40, 2021, [Online]. Available: <http://jurnal.unsil.ac.id/index.php/jssainstek/article/view/4086/1929>.
- [8] B. D. Saputra, M. H. Subagja, M. Aldiansyah, W. Setiawan, Y. Jovanka, and J. Riyanto, "Pengujian White Box berbasis Data Flow Testing pada Program Penghitungan Luas Segitiga," *Sci. Sacra J. Sains, Teknol. dan Masy.*, vol. 1, no. 3, pp. 116–120, 2021.
- [9] M. Y. Rafi, I. Y. Arifin, D. Safutri, D. Fadilah, and J. Riyanto, "Pengujian White Box Testing Menggunakan Teknik Loop Testing pada Aplikasi Sistem Informasi Perpustakaan (Studi Kasus SMKN 3 Kota Tangerang Selatan)," *J. Sains, Teknol. dan Masy.*, vol. 1, no. 3, pp. 214–221, 2021.
- [10] T. Hidayat and M. Muttaqin, "Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis," *J. Tek. Inform. UNIS JUTIS*, vol. 6, no. 1, pp. 2252–5351, 2018, [Online]. Available: [www.ccsenet.org/cis](http://www.ccsenet.org/cis).
- [11] R. Parlika, T. A. Nisaa, S. M. Ningrum, and B. A. Haque, "Studi Literatur Kekurangan dan Kelebihan Pengujian Black Box," *Teknomatika*, vol. 10, no. 02, pp. 131–140, 2020, [Online]. Available: <http://ojs.palcomtech.com/index.php/teknomatika/article/view/490>.
- [12] E. Viglianisi, M. Dallago, and M. Ceccato, "RESTTESTGEN: Automated Black-Box Testing of RESTful APIs," *Proc. - 2020 IEEE 13th Int. Conf. Softw. Testing, Verif. Validation, ICST 2020*, pp. 142–152, 2020, doi: 10.1109/ICST46399.2020.00024.
- [13] R. Prada *et al.*, "Agent-based Testing of Extended Reality Systems," *Proc. - 2020 IEEE 13th Int. Conf. Softw. Testing, Verif. Validation, ICST 2020*, pp. 414–417, 2020, doi: 10.1109/ICST46399.2020.00051.