Informatech: Jurnal Ilmiah Informatika dan Komputer
Volume 1, Nomor 2, Desember 2024, hlm 112-118 **E-ISSN: 3047-4752**

Pengujian Strategi Perangkat Lunak: Tinjauan Literatur Sistematis

Luthfi Nur Syihab[1]*, Janniethia[2], Yuni Sugiarti[3]
UIN Syarif Hidayatullah , Jl. Ir H. Juanda No.95 Ciputat Kota Tangerang Selatan 15412
*[1]*email: luthfi.nursyihab22@mhs.uinjkt.ac.id*
[2]*email: jannie.thia22@mhs.uinjkt.ac.id*
[3]*email: yuni.sugiarti@uinjkt.ac.id*

(Naskah diterima: *4 Mei 2024*; Naskah direvisi: *25 Juni 2024*; Naskah diterbitkan: *1 Desember 2024*)

ABSTRAK *– Pengujian perangkat lunak merupakan langkah penting dalam siklus hidup pengembangan perangkat lunak untuk memastikan kualitas dan keandalan perangkat lunak yang dihasilkan. Penelitian ini menggunakan metode Systematic Literature Review (SLR) sesuai dengan tahapan-tahapan dalam metode SLR, yaitu dengan mengumpulkan, mensurvei, dan mengulas literatur yang berkaitan dengan strategi pengujian perangkat lunak. Pada penelitian ini dilakukan pendekatan secara komprehensif dalam mengulas literatur terbaru terkait strategi pengujian perangkat lunak, serta pemetaan teknik-teknik pengujian berdasarkan efektivitasnya dalam berbagai konteks pengembangan perangkat lunak. Tujuan dari penelitian ini adalah untuk menemukan berbagai teknik pengujian perangkat lunak yang tersedia, menganalisis dan membandingkannya, serta menentukan efektivitas dari masing-masing teknik. Data yang digunakan dalam penelitian ini berasal dari artikel-artikel ilmiah yang telah diseleksi berdasarkan kriteria inklusi dan eksklusi tertentu. Hasil dan pembahasan menunjukkan bahwa terdapat empat kategori utama strategi pengujian perangkat lunak, yang masing-masing terdiri dari beberapa tipe teknik pengujian. Temuan ini memberikan wawasan bahwa proses pengujian dapat dilakukan dengan lebih efisien, sehingga mendukung pengembangan perangkat lunak yang berkualitas.*

Kata Kunci *– Perangkat Lunak, Pengujian, Systematic Literature Review*

Strategy Testing Software: Systematic Literature Review

ABSTRACT *– Software testing is a crucial step in the software development lifecycle to ensure the quality and reliability of the resulting product. This study employs the Systematic Literature Review (SLR) method, following its established stages by collecting, surveying, and reviewing literature related to software testing strategies. In this research, a comprehensive approach is taken to review the latest literature on software testing strategies, as well as to map testing techniques based on their effectiveness in various software development contexts. The objective of this study is to identify the available software testing techniques, analyze and compare them, and determine the effectiveness of each technique. The data used in this study is derived from scholarly articles that have been selected based on specific inclusion and exclusion criteria. The results and discussion reveal four main categories of software testing strategies, each comprising several types of testing techniques. These findings provide insights into how the testing process can be carried out more efficiently, thereby supporting the development of high-quality software.*

Keywords *– Software, Software Testing, Systematic Literature Review*

1. PENDAHULUAN

Verifikasi dan validasi pada sebuah hasil pengembangan perangkat lunak memerlukan proses pengujian. Pendekatan sistematis dengan beragam teknik pengujian diterapkan untuk mengidentifikasi sekaligus mencegah berbagai macam potensi berupa beberapa kesalahan yang dilakukan oleh manusia dalam suatu sistem. Tujuan utamanya adalah memastikan kualitas sebuah perangkat lunak dengan meminimalkan kemungkinan adanya sebuah cacat atau bug [1]. Evaluasi properti tertentu pada perangkat lunak dilaksanakan melalui proses pengujian, baik dengan menggunakan metode manual maupun dengan menggunakan sebuah bantuan teknik

otomatisasi. Kerangka menyeluruh proses pengujian perangkat lunak tercakup dalam sebuah strategi pengujiannya. Pendekatan ini memberikan gambaran komprehensif tentang cara melakukan pengujian selama siklus hidup pengembangan perangkat lunak. Strategi ini mencakup berbagai aspek penting, termasuk sasaran pengujian, alokasi waktu, sumber daya yang diperlukan, serta lingkungan pengujian yang digunakan. [2]. Fungsi utama strategi pengujian perangkat lunak adalah menjamin kelancaran pada sebuah proses pengembangan, memverifikasi kesesuaian dengan persyaratan, dan memastikan perangkat lunak bebas dari cacat. Strategi ini merupakan elemen krusial dalam siklus hidup pengembangan aplikasi atau perangkat lunak. Namun, saat ini pelaksanaan strategi pengujian sering kali tidak terstruktur dan kurang efisien, sebagian besar disebabkan oleh ketidaktahuan seorang pengembang tentang pendekatan pengujian yang paling optimal. [3]. Jadi, untuk mencapai efektivitas dan efisiensi dalam pengujian perangkat lunak, para pengembang perlu memiliki pemahaman mendalam tentang strategi pengujian yang paling ampuh serta dianggap optimal bagi. Pengujian unit, penerimaan/validasi, integrasi, dan pengujian sistem adalah empat jenis pengujian software yang biasa digunakan, menurut penelitian [4]. Survei ini menghasilkan informasi tentang berbagai strategi pengujian perangkat lunak. Oleh karena itu, pengkajian harus diklasifikasikan. Meskipun literatur yang sudah dikumpulkan tidak mencantumkan klasifikasi ini, Pemahaman menyeluruh tentang inti setiap fase dalam strategi pengujian perangkat lunak dapat dicapai dengan lebih mudah oleh pembaca. Penelitian ini bertujuan untuk mengklasifikasikan strategi pengujian software berdasarkan tinjauan jurnal. Oleh karena itu, metode *Systematic Literature Review* (SLR) digunakan untuk menemukan dan mengevaluasi berbagai metode pengujian software [5].

2. TINJAUAN PUSTAKA

Di antara tahapan pengembangan sistem yang paling umum adalah pengujian unit, integrasi, validasi, dan sistem. Berikut adalah penjelasan singkat tentang masing-masing tahapan pengujian tersebut.

Pengujian unit adalah tahap dalam pengujian perangkat lunak yang dapat dilakukan untuk dapat memverifikasi setiap unit atau beberapa komponen sistem. Unit dalam hal ini dapat berupa fungsi, metode, kelas, atau modul kecil dari sistem yang sedang dikembangkan. Pengujian unit memiliki beberapa tujuan untuk memastikan bahwa setiap unit selalu beroperasi dengan benar dan sesuai dengan spesifikasi dan persyaratan yang telah ditetapkan. Ada banyak teknik pengujian yang dapat digunakan untuk mencapai tujuan ini. [6].

Pengujian struktural adalah suatu bentuk pengujian perangkat lunak yang berfokus pada struktur internal program; lebih spesifik, pengujian ini dilaksanakan oleh personel yang memiliki pengetahuan mendalam tentang proses pengembangan sebuah perangkat lunak tersebut. Sebuah tes struktural dilakukan untuk memastikan bahwa sebuah sistem atau aplikasi pengembangan program dan program itu sendiri supaya berjalan dengan baik. Untuk memastikan bahwa desain produk berfungsi dengan baik dan memiliki struktur yang kuat, tujuannya adalah untuk melakukannya. [7].

Pengujian integrasi adalah tahapan pengujian perangkat lunak di mana modul perangkat lunak individual digabungkan dan diuji dalam kelompok. Setelah dan sebelum pengujian sistem untuk mengevaluasi hubungan antara dua atau lebih komponen yang mengirimkan data dari satu area ke area lain [8].

Pengujian sistem dapat dilakukan oleh tim pengembang untuk memastikan kelengkapan teknis dan fungsionalitas sistem. Ini berusaha untuk mengetahui terdapat keraguan apakah komponen-komponen terpisah akan beroperasi sesuai rencana, mengingat adanya kesenjangan antara kinerja sistem yang sebenarnya dan ekspektasi yang telah ditetapkan. modul diskrit bekerja [9].

Fase final dalam rangkaian pengujian, yang dilakukan sebelum sistem disetujui untuk implementasi operasional, dikenal sebagai pengujian penerimaan. Pengujian penerimaan memiliki potensi untuk mengungkap perilaku sistem yang berbeda, mengingat sistem atau aplikasi dapat beroperasi secara distingtif ketika menggunakan data aktual dan dibandingkan dengan data simulasi yang digunakan dalam pengujian. Menunjukkan sebuah kesalahan dan melakukan penghapusan definisi persyaratan sistem karena sistem diuji dengan data yang diberikan oleh pelanggan sistem.[10].

Pengujian sistem manual adalah sebuah jenis pengujian yang dilakukan secara manual tanpa menggunakan alat otomatisasi. Di sini, penguji perangkat lunak berfungsi sebagai pengguna akhir untuk memeriksa apakah perangkat lunak berfungsi dengan baik. Pengujian manual melibatkan penulisan kasus uji secara manual, yang kemudian dapat memeriksa kesalahan dan

melaporkannya melalui alat atau instrumen pelaporan bug. [11].

Pengujian otomatis adalah metode pengujian yang menggunakan alat dan skrip otomatis. Skrip ini diprogram untuk mengotomatisasi tugas-tugas pengujian, termasuk pengujian fungsional dan pengujian lainnya. Alat pengujian otomatis dapat merekam tindakan pengguna, mengumpulkan data, dan secara otomatis memeriksa keluaran.

3. METODE PENELITIAN

Naskah Metode penelitian *Systematic Literature Review* (SLR) digunakan dalam survei ini. SLR, juga dikenal sebagai tinjauan pustaka sistematis dalam bahasa Indonesia, Suatu pendekatan penelitian yang melibatkan proses identifikasi, evaluasi, dan analisis komprehensif terhadap seluruh temuan penelitian yang berkaitan dengan suatu topik spesifik untuk menjawab pertanyaan penelitian atau pertanyaan penelitian. SLR umumnya terdiri dari tiga tahap utama: persiapan, pelaksanaan, dan laporan.

Tahapan persiapan yakni menentukan *Research Question* Pertanyaan Penelitian (*Research Question*/RQ) merupakan elemen kunci dan awal dalam pelaksanaan Tinjauan Sistematis Literatur (*Systematic Literature Review*/SLR). RQ berfungsi sebagai panduan dalam proses pencarian dan pemilihan literatur yang relevan. Dalam konteks ini, pertanyaan penelitian yang diajukan adalah:

a) Sumber-sumber literatur dan jurnal mana yang paling signifikan dalam pembahasan strategi pengujian perangkat lunak?
b) Bagaimana mekanisme kerja dari berbagai metode pengujian perangkat lunak yang ada?

Tahapan pelaksanaan yakni Identifikasi literatur yang akan digunakan pada penelitian dari keyword, sumber dan kriteria inklusi dan eksekusi. Pada tahapan pelaksanaan metode SLR, seseorang mulai mencari keyword melalui pencarian literatur dan harus sesuai dengan protokol SLR yang telah ditentukan. Selain itu, keakuratan literatur yang akan dicari akan dipengaruhi oleh pemahaman yang dimiliki tentang sinonim dan alternatif kata yang digunakan. tentukan sumber literatur yang kami cari menggunakan google scholar sebagai sumber penelitian digital. Sumber-sumber ini termasuk *ScienceDirect*. Untuk memenuhi beberapa kriteria referensi, paper ini harus memenuhi beberapa persyaratan. Salah satunya adalah bahwa makalah ini membahas strategi pengujian software, tetapi tidak membahas topik yang cakupannya terlalu luas akan dihindari; sumber literatur akan dibatasi pada yang ditulis dalam bahasa Inggris atau Indonesia; fokus akan diberikan pada literatur yang menyajikan uraian tentang penerapan praktis strategi pengujian perangkat lunak.

Tahapan laporan yakni tahapan yang dapat menentukan literatur yang akan digunakan setelah tahap pelaksanaan. Penulisan hasil SLR dilakukan. Setelah literatur dikumpulkan, kriteria eksklusi dan inklusi digunakan untuk memilih literatur saat ini.

4. HASIL DAN PEMBAHASAN

Hasil dari proses pencarian dan kriteria inklusi dan eksklusi hingga tahun 2022, 15 artikel telah melalui proses seleksi dengan kriteria data.

Tabel 1. Data Artikel

Tahun	Penulis	Klasifikasi	Keterangan
2019	Narayan C. Debnath, Carlos Humberto Salgado, Mario Peralta, Daniel Eduardo Riesco, Luis Roque, Germán Antonio Montejano, Mouna Mazzi	Pengujian penerimaan	Dalam jurnal ini, penelitian yang dilakukan termasuk tes program yang diarahkan oleh kasus penggunaan. [12]
2022	Zhiqiang Lu	Pengujian unit	Dalam jurnal ini, bagian penting dari proses pengujian strategi periklanan online dibahas untuk menghindari masalah serius sebelum diterapkan.[13]
2022	Fatma Molu	Pengujian penerimaan	Dalam jurnal ini, membahas apakah produk dikembangkan sesuai dengan standar dan

Tahun	Penulis	Klasifikasi	Keterangan
			kriteria rinci dan memenuhi semua persyaratan pengguna.[14]
2018	Adam Roman	Pengujian fungsional	Dalam jurnal ini, menemukan cara untuk membantu menemukan perbedaan dari perilaku yang diharapkan dan memastikan bahwa perangkat lunak berjalan dengan benar.[15]
2017	Rakesh Kumar	Pengujian sistem	Jurnal ini membahas pengujian yang mengevaluasi perangkat lunak terintegrasi lengkap untuk mengetahui apakah memenuhi persyaratan yang ditentukan.[16]
2013	Anubha Chauhan, Naveen Hemrajani	Pengujian integrasi	Berbicara tentang pengujian integrasi untuk memastikan bahwa pemberdayaan transaksi bisnis antara aplikasi dan sistem berhasil.[17]
2006	Mats Grindal, Asa G. Dahlstedt, Jeff Offutt, Jonas melin	Pengujian unit	Keseluruhan proses pengujian yang dijelaskan dalam jurnal terutama dalam hal memilih strategi kombinasi yang tepat dan menangani konflik dalam model parameter input.[18]
2002	Kai-Yuan Cai	Pengujian unit	Mengidentifikasi dan memperbaiki kesalahan di awal siklus pengembangan dengan menguji unit secara independen adalah subjek dari jurnal ini.[19]
2017	Hanyu Pei, Beibei Yin, Min Xie, Kai-Yuan Cai	Pengujian unit	Dalam jurnal ini, setiap komponen atau unit perangkat lunak secara terpisah diuji untuk memastikan bahwa mereka beroperasi dengan benar.[20]
2013	Simon Poulding	Pengujian struktural	Dalam jurnal ini, pengujian kotak putih dibahas, yang melibatkan menilai struktur internal kode perangkat lunak untuk merancang kasus uji yang memastikan bahwa semua pernyataan, cabang, dan jalur dilaksanakan.[21]
2016	Kevin Taylor-Sakyi	Pengujian struktural	Dengan memverifikasi bahwa struktur internal kode selaras dengan perilaku dan persyaratan yang diharapkan, strategi

Tahun	Penulis	Klasifikasi	Keterangan
			pengujian keseluruhan meningkatkan keandalan sistem perangkat lunak.[22]
2015	Edward J.Waller	Pengujian unit	Solusi perangkat keras responden pertama untuk triase personel yang terpapar secara internal di lapangan dibahas dalam jurnal ini. Solusi ini diuji di lapangan dengan unit medis militer dan ditemukan mudah dipelajari dan digunakan.[23]
2021	Italo Santos, Allan Mori, Simone R.S. Souza	Pengujian fungsional	Dalam paper ini, analisis survei digunakan untuk mengevaluasi dampak pengujian perangkat lunak pengajaran dengan strategi pengujian tambahan. [24]
2014	Fatma Molu	Pengujian fungsional	Fokus artikel ini adalah untuk membantu menemukan penyimpangan dari perilaku yang diharapkan serta memastikan bahwa sistem memenuhi kebutuhan pengguna.(Molu, 2014)
2012	K.Ajay Babu, K.Madhuri, M.Suman	Pengujian penerimaan	Dalam paper ini, evaluasi kepatuhan sistem dengan persyaratan bisnis dan kesiapannya untuk diterapkan dibahas.[26]

Hasil RQ 1 : Sumber-sumber literatur dan jurnal mana yang paling signifikan dalam pembahasan strategi pengujian perangkat lunak? Strategi pengujian perangkat lunak dan Jurnal yang relevan dicari melalui termasuk *Science Direct*.

Hasil RQ 2 : Bagaimana mekanisme kerja dari berbagai metode pengujian perangkat lunak yang ada?

Pengujian unit (*Unit Testing*) dapat menguji unit perangkat lunak secara terpisah. Cara kerjanya adalah dengan membuat kasus uji untuk setiap unit, yang menguji fungsionalitas dasar mereka. Menguji kasus pada unit. Pengembang atau penguji memeriksa hasil yang sebenarnya sesuai fakta dan membandingkannya dengan hasil yang diharapkan. Analisis hasil dan laporkan masalah atau bug.

Pengujian integrasi (*Integration Testing*) memiliki tugas untuk memeriksa bagaimana berbagai modul atau unit berinteraksi satu sama lain. Cara kerjanya adalah dengan membuat kasus uji yang mensimulasikan interaksi antar modul dalam skenario yang realistis. Menyelesaikan kasus uji perangkat lunak. Penguji memeriksa apakah modul berfungsi dengan benar dan apakah data tidak hilang atau korup. Analisis hasil dan laporkan masalah atau bug.

Sistem pengujian menguji seluruh sistem perangkat lunak untuk memastikan bahwa itu memenuhi semua persyaratan dan tujuan. Cara kerjanya adalah sebagai berikut: Membuat kasus uji untuk menguji semua fungsi sistem. Kasus uji ini mencakup semua skenario penggunaan utama serta persyaratan non-fungsional seperti kinerja dan keamanan. Kasus pengujian ini dijalankan dalam kondisi lingkungan yang realistis. Penguji memantau kinerja sistem, stabilitas, dan kegunaan. Analisis hasil dan laporkan masalah atau bug.

Pengujian fungsional menunjukkan bahwa kemampuan perangkat lunak untuk menjalankan fungsi-fungsi yang telah ditetapkan dan memenuhi spesifikasi fungsionalnya dapat diverifikasi. Dengan perspektif lain, proses pengujian ini bertujuan untuk mengkonfirmasi bahwa perangkat lunak tersebut dapat melakukan apa yang seharusnya dilakukan. Cara kerjanya adalah sebagai berikut: Kasus uji dibuat berdasarkan spesifikasi fungsional, yang menentukan fungsionalitas yang harus dimiliki

perangkat lunak. Menyelesaikan kasus uji perangkat lunak. Untuk memastikan bahwa perangkat lunak menghasilkan hasil yang tepat, penguji memberikan input dan mengamati outputnya. Analisis hasil dan laporkan masalah atau bug.

Pengujian struktural memastikan bahwa struktur internal perangkat lunak benar dan sesuai dengan desain; dengan kata lain, mereka memastikan bahwa perangkat lunak dibangun dengan baik dan tidak memiliki kesalahan struktural. Cara kerjanya: Kasus uji dibuat sesuai dengan struktur kode program dan dirancang untuk memastikan bahwa setiap bagian kode dieksekusi sedikitnya sekali dan untuk menguji semua aliran kontrol yang mungkin selalu terjadi. Kemudian, kasus uji ini dijalankan pada perangkat lunak. Penguji mengamati aliran program dan variabel internal untuk menemukan kemungkinan bug atau masalah. Mereka kemudian menganalisis hasil dan melaporkan bug atau masalah tersebut.

Pengujian penerimaan (*acceptance*) adalah proses yang dilakukan oleh pengguna atau pemangku kepentingan untuk memastikan bahwa perangkat lunak memenuhi kebutuhan dan siap digunakan. Cara kerjanya adalah sebagai berikut: Pengguna atau pemangku kepentingan mendefinisikan kriteria penerimaan. Kriteria Smart harus spesifik, terukur, dapat dicapai, relevan, dan berjangka waktu. Pengguna atau pemangku kepentingan harus menjalankan kasus uji berdasarkan kriteria ini, mencatat hasilnya, dan memberikan umpan balik kepada pengembang. Pengembang (*developer*) juga akan memperbaiki kesalahan atau masalah yang ditemukan, dan mengulangi kegiatan pengujian jika diperlukan.

5. SIMPULAN

Pengujian adalah bagian penting dan dapat diandalkan dari pengembangan atau pembuatan perangkat lunak yang berkualitas tinggi. Strategi pengujian yang tepat dapat membantu perusahaan mengurangi beberapa kemungkinan kegagalan proyek, dapat meningkatkan kepuasan pengguna, dan meningkatkan profitabilitas perusahaan. Strategi pengujian software biasanya dibagi menjadi empat kategori antara lain seperti pengujian unit, pengujian penerimaan/validasi, pengujian integrasi, dan pengujian sistem. Hasil penelitian ini membuktikan bahwa salah satu strategi pengujian perangkat lunak (*software*) yang efektif dan paling sering digunakan oleh pengembang (*developer*) perangkat lunak yaitu kegiatan pengujian unit untuk menemukan kesalahan pada tingkat modul atau komponen. Oleh karena itu, strategi pengujian ini diperlukan untuk menjaga kualitas perangkat lunak.

DAFTAR PUSTAKA

[1] I. Jovanic, *Comperehensive study of software testing : categories, levels, techniques, and types*.

[2] and P. . C. A. A. Sawant, P. H. Bari, "Software Testing Techniques and Strategies," 2012. [Online]. Available: https://www.researchgate.net/publication/316510706_Software_Testing_Techniques_and_Strategies

[3] S. J. Putra, Y. Sugiarti, B. Y. Prayoga, D. W. Samudera, and D. Khairani, "Analysis of Strengths and Weaknesses of Software Testing Strategies: Systematic Literature Review," *2023 11th Int. Conf. Cyber IT Serv. Manag. CITSM2023*, 2023, doi: 10.1109/CITSM60085.2023.10455226.

[4] and P. C. S. Thakare, S. Chavan, "'Software Testing Strategies and Techniques,'" vol. 2, no. 4, pp. 682–687, 2012, [Online]. Available: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Software+Testing+Strategies+and+Te chniques#0.

[5] S. F. Arief and Y. Sugiarti, "Literature Review: Analisis Metode Perancangan Sistem Informasi Akademik Berbasis Web," *J. Ilm. Ilmu Komput.*, vol. 8, no. 2, pp. 87–93, 2022, doi: 10.35329/jiik.v8i2.229.

[6] M. I. T. M.Kom, M. J., Ph.D, R. G. A., M. Kom, & Ph.D, P., "Pengujian dan Implementasi Sistem Informasi.," Deepublish.

[7] J. Simarmata, "Rekayasa Perangkat Lunak," Andi.

[8] S. ST. M.Acc, A. N. I., "Buku Ajar Audit Sistem Informasi.," Nas Media Pustaka.

[9] E. Y. Cahyono, R. Y., Wulandari, H. M., Hartati, S., & Anggraeni, "Sistem Informasi Manajemen," NEM, 2023.

[10] Erlangga, *Software Engineering*.

[11] S. K. M.Kom, M. A. S., "Perancangan dan Pengembangan Sistem Informasi Berbasis Scrum.," Deepublish, 2024.

[12] N. Debnath *et al.*, "A Software Testing Strategy Based on a Software Product Quality Model," *Learn. Anal. Intell. Syst.*, vol. 7, no. February, pp. 248–259, 2020, doi: 10.1007/978-3-030-36778-7_27.

[13] Z. Lu, "An intelligent software testing method for online advertising strategy," *2022 IEEE Int. Conf. Electr. Eng. Big Data Algorithms, EEBDA*

2022, 2022, doi: 10.1109/EEBDA53927.2022.9744981.

[14] F. Molu, "Software Testing Strategy," *Int. J. E-Services Mob. Appl.*, vol. 6, no. 2, pp. 23–36, 2014.

[15] A. Roman, "Testing Strategies: How to Become a Better Tester?," *Thinking-Driven Test.*, pp. 53–97, 2018, doi: 10.1007/978-3-319-73195-7_2.

[16] R. Kumar, "Software Testing Strategies to Improve Software Quality," vol. 04, no. 17, 2017.

[17] A. Chauhan, "Smart Meter Implementation Program," vol. 2, no. 5, pp. 1447–1462, 2013.

[18] J. Grindal, Mats Dahlstedt, Åsa G.Offutt, Jeff Offutt, "Using Combination Strategies for Software Testing in Practice : A Proof-of-Concept".

[19] K. Y. Cai, "Optimal software testing and adaptive software testing in the context of software cybernetics," *Inf. Softw. Technol.*, vol. 44, no. 14, pp. 841–855, 2002, doi: 10.1016/S0950-5849(02)00108-8.

[20] B. Pei, Hanyu Yin and K. Y. Xie, Min Cai, "A cloud-based dynamic random software testing strategy," *IEEE Int. Conf. Ind. Eng. Eng. Manag.*, vol. 2017-Decem, pp. 509–513, 2017, doi: 10.1109/IEEM.2017.8289943.

[21] S. Poulding, "The Use of Automated Search in Deriving Software Testing Strategies," no. July, 2013.

[22] K. Taylor-Sakyi, "Reliability Testing Strategy - Reliability in Software Engineering," 2016.

[23] E. J. Waller, "Operational testing of a combined hardware-software strategy for triage of radiologically-contaminated persons," vol. 109, no. 2, pp. S176–S185, 2015, doi: 10.1097/HP.0000000000000316.

[24] I. Santos, A. Mori, and S. R. S. Souza, "Using an Incremental Testing Strategy to Improve Students' Perception of Software Quality," pp. 181–190, 2021, doi: 10.5753/WEI.2021.15909.

[25] F. Molu, "Software Testing Strategy: In Conversion of Complex Financial Systems," *Int. J. E-Services Mob. Appl.*, vol. 6, no. 2, pp. 23–36, 2014, doi: 10.4018/IJESMA.2014040103.

[26] K. Ajay Babu, K. Madhuri, and M. Suman, "An Evaluation Scheme of Software Testing Strategy," *Behav. Comput. Model. Anal. Min. Decis.*, pp. 353–361, 2012, doi: 10.1007/978-1-4471-2969-1_23.