

# Survei Strategi Pengujian Software Menggunakan Metode Systematic Literature Review

Akhmad Murtado<sup>1\*</sup>, Erlangga Firmansyah<sup>2</sup>, M.Ryzha Fadillah<sup>3</sup>, Puspita Maula<sup>4</sup>, Aries Saifudin<sup>5</sup>

<sup>1,2,3</sup>Teknik Informatika, Universitas Pamulang, Taggerang Selatan, Indonesia

<sup>1,2,3</sup>Jl. Surya Kencana No.1, Pamulang Bar., Kec. Pamulang, Kota Tangerang Selatan, Banten 15417

email: <sup>1</sup>akhmadmurtado0@gmail.com, <sup>2</sup>erlanggafs05@gmail.com, <sup>3</sup>ryzha.fadillah@gmail.com, <sup>4</sup>puspitamaula@gmai.com, <sup>5</sup>aries.saifudin@unpam.ac.id

**Abstract** – Software testing is a crucial process to ensure the quality and reliability of software. This study employs a Systematic Literature Review (SLR) approach to identify and review various software testing strategies published in the scientific literature. The results identify four main strategies: Unit Testing, Acceptance/Validation Testing, Integration Testing, and System Testing. Each strategy has its own advantages and disadvantages that need to be considered in its application. Unit Testing excels at detecting early bugs but is limited in testing interactions between components. Acceptance/Validation Testing validates user requirements but is complex and requires end-user participation. Integration Testing tests interactions between components but depends on the availability of integrated components. System Testing evaluates the entire system but requires more time and resources. Choosing the right strategy can enhance the quality of the software produced. This study contributes to understanding and improving software testing strategies, assisting developers and researchers in selecting the most suitable methods for their project needs, thus creating more reliable software that meets user expectations.

**Abstrak** – Pengujian perangkat lunak adalah proses penting dalam memastikan kualitas dan kehandalan perangkat lunak. Studi ini menggunakan pendekatan Systematic Literature Review (SLR) untuk mengidentifikasi dan meninjau berbagai strategi pengujian perangkat lunak yang dipublikasikan dalam literatur ilmiah. Hasil penelitian mengidentifikasi empat strategi utama: Unit Testing, Acceptance/Validation Testing, Integration Testing, dan System Testing. Setiap strategi memiliki kelebihan dan kekurangan yang harus dipertimbangkan dalam penerapannya. Unit Testing unggul dalam mendeteksi bug dini namun terbatas dalam menguji interaksi antar komponen. Acceptance/Validation Testing memvalidasi kebutuhan pengguna, tetapi kompleks dan memerlukan partisipasi pengguna akhir. Integration Testing menguji interaksi antar komponen namun tergantung pada ketersediaan komponen yang terintegrasi. System Testing menguji seluruh sistem namun membutuhkan biaya dan waktu lebih banyak. Pemilihan strategi yang tepat dapat meningkatkan kualitas perangkat lunak yang dihasilkan. Studi ini memberikan kontribusi dalam memahami dan meningkatkan strategi pengujian perangkat lunak, serta membantu para pengembang dan peneliti dalam memilih metode yang paling sesuai dengan kebutuhan proyek mereka, sehingga menciptakan perangkat lunak yang lebih handal dan sesuai dengan harapan pengguna.

**Kata Kunci** – Pengujian Perangkat Lunak, Systematic Literature Review, Unit Testing, Acceptance/Validation Testing, Integration Testing, System Testing

## I. PENDAHULUAN

Pengujian perangkat lunak dibutuhkan dalam melakukan konfirmasi dan validasi dari hasil pengembangan suatu perangkat lunak. Pengujian dilakukan secara sistematis dengan menggunakan sejumlah strategi pengujian perangkat lunak. Pengujian perangkat lunak digunakan untuk mencegah peluang kesalahan yang dibuat oleh manusia dalam suatu sistem atau dengan kata lain untuk menghindari perangkat lunak yang ada dari cacat. Pengujian perangkat lunak adalah metode untuk mengkonfirmasi apakah perangkat lunak sesuai dengan persyaratan yang ada, sekaligus untuk memastikan bahwa perangkat lunak bebas dari error. Strategi pengujian perangkat lunak memastikan kelancaran pengembangan perangkat lunak, memastikan agar perangkat lunak yang ada sesuai dengan persyaratan dan juga memastikan agar perangkat lunak tersebut bebas dari cacat. Strategi Pengujian Software merupakan salah satu hal penting dalam pengembangan aplikasi software development life cycle [1].

Kita akan mengklasifikasikan Strategi pengujian software berdasarkan kelebihan dan kekurangan. Sehingga, diharapkan dapat melakukan proses pengujian software secara optimal dan dapat menghasilkan software sesuai yang diharapkan. Oleh karena itu penelitian ini dilakukan dengan cara mengidentifikasi dan meninjau macam macam strategi pengujian software dengan menggunakan metode Systematic Literature Review (SLR). Dari survei yang dilakukan, secara garis besar strategi pengujian software menjadi 4 tipe, yang terdiri dari Unit testing, Acceptance / Validation testing, Integration testing, dan System testing [2]. Salah dalam strategi pengujian dapat mengakibatkan kegagalan dalam mendeteksi bug atau cacat perangkat lunak. Hal ini dapat menyebabkan penurunan kualitas perangkat lunak secara keseluruhan dan membuatnya tidak dapat diandalkan. Peluncuran pun akan tertunda dan Keterlambatan ini dapat mengakibatkan kerugian finansial dan reputasi bagi perusahaan. Biaya yang akan dikeluarkan juga akan semakin besar dan juga kita akan kehilangan peluang bisnis yang berharga[3].

Survei ini menggunakan metode penelitian Systematic Literature Review (SLR). Systematic literature review / SLR atau yang sering disebut sebagai tinjauan pustaka sistematis dalam bahasa indonesia adalah metode literature review yang mengenali, menilai, dan menginterpretasi seluruh temuan-temuan yang dikumpulkan pada satu topik penelitian, untuk menjawab pertanyaan penelitian / research question yang telah ditetapkan pada awal penelitian. Secara umum, terdapat 3

tahapan besar dalam melakukan SLR, diantaranya adalah Planning, Conducting dan Reporting [4].

Research Question (RQ) adalah bagian awal dan dasar berjalannya SLR. RQ memandu proses pencarian dan ekstraksi dari literatur yang akan dikumpulkan. Tahapan conducting adalah tahapan pelaksanaan SLR, dimana seharusnya sesuai dengan protokol SLR yang telah ditentukan, dimulai dengan penentuan keyword dari pencarian literatur (search string) yang basisnya adalah PICOC yang telah didesain sebelumnya. Reporting adalah tahapan penulisan hasil SLR yang dilakukan [5].

\*) **penulis korespondensi:** M. Ryzha Fadillah  
Email: ryzha.fadillah@gmail.com

## II. METODOLOGI

Metodologi penelitian ini mengikuti pendekatan Systematic Literature Review (SLR) untuk mengidentifikasi dan meninjau berbagai strategi pengujian software yang telah dipublikasikan dalam literatur ilmiah. Tahapan-tahapan utama dalam pelaksanaan SLR, yaitu Planning, Conducting, dan Reporting, diikuti dengan seksama untuk memastikan keberhasilan dan keakuratan penelitian ini.

### A. Perencanaan (Planning)

Tahap perencanaan dimulai dengan menetapkan tujuan penelitian dan merumuskan pertanyaan penelitian (Research Question/RQ). RQ menjadi pedoman dalam melakukan pencarian literatur yang relevan. Selain itu, dalam tahap ini juga dilakukan penentuan kriteria inklusi dan eksklusi, serta penyusunan strategi pencarian yang efektif.

#### Pertanyaan Penelitian (RQ):

RQ1: Apa saja strategi pengujian software yang telah dikemukakan dalam literatur ilmiah?

RQ2: Apa kelebihan dan kekurangan dari setiap strategi pengujian software yang telah diajukan?

#### Kriteria Inklusi:

Publikasi dalam jurnal ilmiah atau konferensi terkemuka.

Membahas tentang strategi pengujian software.

Ditulis dalam bahasa Inggris.

Diterbitkan dalam rentang waktu tertentu, misalnya tahun 2010 hingga 2022.

#### Kriteria Eksklusi:

Publikasi yang tidak relevan dengan topik penelitian.

Publikasi yang tidak tersedia dalam akses publik atau berlangganan.

Publikasi yang tidak berada dalam bahasa Inggris.

### B. Pelaksanaan (Conducting)

Tahap pelaksanaan melibatkan pencarian literatur yang sesuai dengan kriteria inklusi yang telah ditetapkan. Pencarian literatur dilakukan melalui basis data akademik seperti IEEE Xplore, ACM Digital Library, Google Scholar, dan database lainnya. Setelah itu, dilakukan proses seleksi dan penilaian kualitas literatur yang relevan.

### C. Pelaporan (Reporting)

Tahap pelaporan melibatkan analisis dan sintesis dari literatur yang telah dikumpulkan. Hasil dari analisis literatur

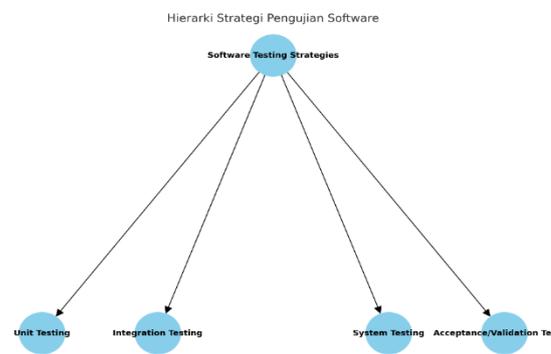
kemudian dirangkum dan dipresentasikan secara sistematis sesuai dengan struktur penulisan artikel ilmiah yang baik dan benar. Selain itu, kesimpulan dan temuan-temuan penting juga disampaikan dengan jelas untuk memberikan pemahaman yang komprehensif kepada pembaca.

Dengan mengikuti ketiga tahapan tersebut secara cermat dan terstruktur, diharapkan penelitian ini dapat memberikan kontribusi yang berharga dalam pemahaman dan peningkatan strategi pengujian software yang ada.

## III. HASIL DAN PEMBAHASAN

### A. Strategi Pengujian Software yang Teridentifikasi

Berdasarkan penelitian yang dilakukan melalui Systematic Literature Review (SLR), berbagai strategi pengujian software telah teridentifikasi dari literatur ilmiah yang relevan. Strategi-strategi ini mencakup:



Gambar 1 Hierarki strategi pengujian software

#### Hierarki Strategi Pengujian Software

- Unit Testing
- Acceptance / Validation Testing
- Integration Testing
- System Testing

Setiap strategi pengujian memiliki ciri khas, ruang lingkup, dan tujuan tersendiri dalam menguji perangkat lunak. Unit Testing, misalnya, fokus pada pengujian komponen perangkat lunak secara terpisah, sementara Integration Testing menguji interaksi antara komponen-komponen tersebut. Sementara itu, System Testing mengevaluasi keseluruhan sistem perangkat lunak untuk memastikan bahwa fungsi-fungsi utama berjalan dengan baik dalam berbagai kondisi.

### B. Kelebihan dan Kekurangan Setiap Strategi Pengujian

Pada tahap ini, dilakukan analisis terhadap kelebihan dan kekurangan dari masing-masing strategi pengujian software yang telah teridentifikasi. Dengan pemahaman yang mendalam terhadap karakteristik setiap strategi, kita dapat mengaplikasikan strategi pengujian yang paling sesuai dengan kebutuhan dan kondisi proyek perangkat lunak.

- **Unit Testing:** Kelebihannya termasuk kemampuan untuk mendeteksi bug secara dini dan memfasilitasi pengujian otomatis. Namun, kelemahannya adalah keterbatasan dalam menguji interaksi antar komponen.
- **Acceptance / Validation Testing:** Strategi ini memungkinkan validasi terhadap persyaratan pengguna dan kebutuhan bisnis. Namun, prosesnya cenderung

kompleks dan memerlukan partisipasi aktif dari pengguna akhir.

- **Integration Testing:** Kelebihannya adalah mampu menemukan bug yang terkait dengan interaksi antara komponen. Namun, kekurangannya adalah ketergantungan pada ketersediaan komponen-komponen yang terintegrasi.
- **System Testing:** Strategi ini menguji keseluruhan sistem dari awal hingga akhir, memastikan bahwa keseluruhan aplikasi berfungsi dengan baik. Namun, kekurangannya adalah biaya dan waktu yang dibutuhkan untuk melaksanakan pengujian yang menyeluruh.

Dengan memahami kelebihan dan kekurangan dari masing-masing strategi pengujian, kita dapat mengembangkan rencana pengujian yang efektif dan efisien untuk memastikan kualitas perangkat lunak yang optimal.

### C. Temuan dan Implikasi

Temuan dari penelitian ini memberikan pemahaman yang lebih mendalam tentang berbagai strategi pengujian software yang tersedia, serta kelebihan dan kekurangannya masing-masing. Implikasi dari temuan ini adalah pentingnya pemilihan strategi pengujian yang tepat sesuai dengan kebutuhan proyek perangkat lunak. Dengan demikian, diharapkan dapat meningkatkan kualitas dan kehandalan perangkat lunak yang dihasilkan.

## IV. KESIMPULAN

Studi ini bertujuan untuk mengidentifikasi dan meninjau berbagai strategi pengujian perangkat lunak yang telah dipublikasikan dalam literatur ilmiah melalui pendekatan Systematic Literature Review (SLR). Dari hasil penelitian, empat strategi pengujian perangkat lunak utama teridentifikasi, yaitu Unit Testing, Acceptance/Validation Testing, Integration Testing, dan System Testing. Setiap strategi memiliki kelebihan dan kekurangan masing-masing yang perlu dipertimbangkan dalam pemilihan dan pelaksanaannya.

Unit Testing memiliki kelebihan dalam mendeteksi bug secara dini dan memfasilitasi pengujian otomatis, namun memiliki keterbatasan dalam menguji interaksi antar komponen. Acceptance/Validation Testing memungkinkan validasi terhadap persyaratan pengguna dan kebutuhan bisnis, namun cenderung kompleks dan memerlukan partisipasi aktif pengguna akhir. Integration Testing mampu menemukan bug yang terkait dengan interaksi antara komponen, tetapi tergantung pada ketersediaan komponen yang terintegrasi. Sementara System Testing menguji keseluruhan sistem untuk memastikan keseluruhan aplikasi berfungsi dengan baik, namun membutuhkan biaya dan waktu yang lebih besar.

Temuan dari studi ini memberikan pemahaman yang lebih mendalam tentang strategi pengujian perangkat lunak yang tersedia, serta kelebihan dan kekurangannya. Implikasi dari temuan ini adalah pentingnya pemilihan strategi pengujian yang tepat sesuai dengan kebutuhan proyek perangkat lunak untuk meningkatkan kualitas dan kehandalan perangkat lunak yang dihasilkan. Dengan demikian, studi ini memberikan kontribusi yang berharga dalam pemahaman dan peningkatan strategi pengujian perangkat lunak.

## DAFTAR PUSTAKA

- [1] Amam, & Rusdiana, S. (2022, februari). Ulasan dengan Metode Systematic Literature Review (SLR). *Jurnal Peternakan*, Vol.19, No.1, (9-21).
- [2] Anuraga, G., Indrasetyaningih, A., & Athoillah, M. (2021). Pelatihan Pengujian Hipotesis Statistika Dasar dengan Software R. *JURNAL PENGABDIAN MASYARAKAT*, Vol 3, No 2, (327-334).
- [3] Dhaifullah, I. R., H. M. M., Salsabila, A. A., & Yakin, M. A. (2022, Mei). Survei Teknik Pengujian Software. *Journal Automation Computer Information System*, Vol.2, No.1, (31-38).
- [4] Hernandez, D. (2023). Implementation of PHP Unit Testing in Web Development Projects. *Jurnal Ilmu Komputer (JILKOM)*, Vol. 2 No. 4.
- [5] Huda, M., Yaqin, M. A., Kurniawan, R. F., & Choiri, M. W. (2022, April). Survei Strategi Pengujian Software Menggunakan Metode Systematic. *Journal of Computer Science and Applied Informatics*, Vol. 4, No. 1, (116-133).
- [6] Nugraha, B. F., Aditama, F., Arrofi, M., Ahmad, S. U., & Yulianti. (2020). Pengujian Black Box pada Aplikasi Penghitungan Parkir Swalayan ADA. *Jurnal Informatika Universitas Pamulang*, Vol. 5, No. 2, (146-151).
- [7] Wahyudin, Y., & Rahayu, D. N. (2020, Oktober). Volume 15 Nomor 03 Bulan Oktober-Tahun 202. *Jurnal Interkom*, Vol.15, No.03, (119-133).