

# **INTEGRATION TESTING DAN USER TESTING DALAM PERANCANGAN PERANGKAT LUNAK**

**Mata Kuliah: Software Engineering**



**DOSEN: Yudhi Fajar Saputra, S.Kom., M.Sc**

**Pertemuan ke-**

**Topik Bahasan ke-**

**SEMESTER : 3/ TA. 2024-2025**

**KODE MK/SKS: MKP001/3 SKS**

**PRODI INFORMATIKA/ILMU KOMPUTER  
UNIVERSITAS WIDYA GAMA MAHAKAM SAMARINDA**

Nama Mata Kuliah : Software Engineering/Rekayasa Perangkat Lunak  
Kode Mata Kuliah/SKS : MKP \_\_\_\_/3 SKS  
Dosen : Yudhi Fajar Saputra,  
Semester : 3/ 2024  
Hari Pertemuan / Jam : -  
Tempat Pertemuan : Ruang Kelas A.06

**Integration Testing** dan **User Testing** adalah dua jenis pengujian perangkat lunak yang penting dalam siklus pengembangan perangkat lunak. Keduanya memiliki tujuan yang berbeda dalam memastikan kualitas perangkat lunak, dan keduanya melibatkan pengujian dari perspektif yang berbeda pula. Integration testing adalah tahap penting dalam siklus pengujian perangkat lunak yang bertujuan untuk memastikan bahwa komponen-komponen atau modul-modul yang terpisah dapat bekerja secara efektif saat digabungkan, sedangkan User testing, atau pengujian kegunaan, adalah proses untuk mengevaluasi seberapa mudah dan intuitif suatu perangkat lunak digunakan oleh pengguna akhir. Tujuan dari user testing adalah untuk mengidentifikasi potensi masalah dalam antarmuka pengguna dan alur kerja yang dapat mempengaruhi pengalaman pengguna.

## 1. INTEGRATION TESTING

**Integration Testing** adalah tahap pengujian perangkat lunak yang dilakukan untuk menguji interaksi antara komponen-komponen atau modul-modul yang telah terpisah, untuk memastikan bahwa antar komponen bekerja dengan baik ketika digabungkan menjadi satu sistem yang utuh.

### 1) Tujuan dari Integration Testing

beberapa tujuan dari Integration Testing adalah

- a) Menjamin bahwa berbagai modul atau komponen perangkat lunak dapat berinteraksi dengan benar.
- b) Mengidentifikasi masalah yang mungkin timbul ketika komponen-komponen yang sebelumnya diuji secara terpisah digabungkan, seperti kesalahan dalam pertukaran data antar modul.

### 2) Jenis-Jenis Integration Testing

beberapa Jenis Integration Testing: adalah:

- a) **Top-Down Integration Testing:** dalam metode ini, pengujian dimulai dengan modul paling atas dalam hierarki sistem dan kemudian secara bertahap menguji modul-modul yang lebih rendah. Pengujian dilakukan secara bertahap hingga seluruh sistem terintegrasi [1], maka dari itu metode

ini digunakan terhadap sistem yang memiliki struktur hirarki yang jelas.

- b) Bottom-Up Integration Testing: metode ini memulai pengujian dengan modul-modul paling bawah atau dasar dalam sistem, kemudian berlanjut ke modul yang lebih tinggi. Dalam metode ini, driver digunakan untuk menggantikan modul yang lebih tinggi yang belum siap [2].
- c) Big Bang Integration Testing: dalam metode ini, seluruh sistem atau aplikasi diuji sekaligus setelah semua komponen dan modul digabungkan. Pengujian dilakukan tanpa adanya pengujian bertahap sebelumnya, maka dari itu metode ini digunakan dalam proyek yang lebih kecil atau ketika semua komponen sudah siap pada saat yang bersamaan [1].
- d) Incremental Integration Testing: Pengujian dilakukan secara bertahap, dengan menggabungkan dan menguji modul-modul satu per satu, hingga seluruh sistem terintegrasi. Setiap modul diuji dalam urutan tertentu hingga sistem akhir tercapai [1], maka dari itu metode ini sering digunakan dalam proyek yang lebih besar dengan banyak modul yang bergantung satu sama lain.
- e) Sandwich Integration Testing: Metode ini menggabungkan pendekatan top-down dan bottom-up. Pengujian dilakukan dengan cara menggabungkan kedua metode tersebut, yaitu dengan menguji modul atas dan bawah secara bersamaan. metode ini digunakan untuk sistem yang kompleks di mana beberapa bagian dari sistem sudah siap lebih awal dan dapat diuji secara bersamaan [2].

## 2. USER TESTING

**User Testing** (atau **Usability Testing**) adalah jenis pengujian yang dilakukan untuk mengevaluasi pengalaman pengguna dengan perangkat lunak atau aplikasi. Tujuan dari user testing adalah untuk mengidentifikasi masalah dalam penggunaan sistem yang mungkin dihadapi oleh pengguna akhir.

### 1) Tujuan dari User Testing:

beberapa tujuan dari user testing adalah

- a) Menilai seberapa mudah dan intuitif perangkat lunak digunakan oleh pengguna.
- b) Mengidentifikasi kesulitan atau kebingungannya dalam antarmuka pengguna (UI) atau alur kerja aplikasi.
- c) Mendapatkan umpan balik langsung dari pengguna mengenai fitur dan fungsi sistem.

### 2) Jenis-Jenis User Testing

beberapa jenis user testing adalah

- a) **Formative Usability Testing:** Formative usability testing dilakukan selama tahap pengembangan perangkat lunak, metode testing ini sering digunakan untuk sistem prototipe atau versi awal. Tujuan utama dari pengujian ini adalah untuk mengidentifikasi masalah kegunaan (usability) dan memberikan umpan balik yang dapat digunakan untuk meningkatkan desain sebelum produk akhir dirilis [4].
- b) **Summative Usability Testing:** Pengujian yang dilakukan setelah perangkat lunak selesai dikembangkan dan siap digunakan. Pengujian ini mengukur seberapa efektif, efisien, dan memuaskan perangkat lunak digunakan oleh pengguna, untuk itu metode ini digunakan untuk mengevaluasi apakah perangkat lunak sudah siap digunakan secara luas oleh pengguna dan untuk mengukur sejauh mana tujuan kegunaan tercapai [5].
- c) **Exploratory User Testing:** Pengguna diberikan kebebasan untuk mengeksplorasi aplikasi tanpa instruksi yang jelas atau tugas tertentu. Tujuan dari pengujian ini adalah untuk melihat bagaimana pengguna berinteraksi dengan perangkat lunak secara spontan, tanpa bimbingan dari penguji, maka dari itu pengujian ini cocok untuk mengeksplorasi aplikasi yang sangat interaktif atau inovatif, dimana pengujian bebas dapat memberikan wawasan mengenai bagian-bagian yang mungkin membingungkan pengguna atau fitur yang menarik perhatian mereka [4].
- d) **Task-Based Usability Testing:** Dalam jenis pengujian ini, pengguna diminta untuk menyelesaikan tugas tertentu dengan menggunakan aplikasi. Pengujian ini mengukur seberapa mudah dan efisien pengguna dapat menyelesaikan tugas yang diberikan, serta apakah mereka mengalami kesulitan dalam proses tersebut, pengujian ini sering digunakan untuk menguji tugas-tugas khusus yang relevan dengan tujuan utama perangkat lunak, seperti memesan produk, mengirim email, atau menyelesaikan pembelian [6].
- e) **A/B Testing:** A/B testing melibatkan perbandingan dua versi berbeda dari antarmuka pengguna atau fitur perangkat lunak untuk melihat mana yang lebih efektif dalam memenuhi tujuan tertentu. Pengguna dibagi dalam dua kelompok, masing-masing menguji versi A dan B. Pengujian ini diterapkan untuk membandingkan perubahan desain atau fitur tertentu, seperti menguji apakah satu warna tombol lebih menarik atau tidak, atau apakah satu layout halaman lebih efektif dalam menarik klik dibanding dengan yang lain [7].

### 3. DAFTAR REFERENSI

1. Myers, G. J., Sandler, C., & Badgett, T. (2011). *The Art of Software Testing*. Wiley.
2. ISTQB Foundation Level Syllabus, 2018

3. Kaner, Falk, & Nguyen, 1999, *Testing Computer Software*
4. Nielsen, J. (1994). *Usability Engineering*. Academic Press
5. ISO 9241-11:2018 - *Ergonomics of human-system interaction - Part 11: Usability: Definitions and concepts*.
6. Rubin, J. (2012). *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. Wiley
7. Kohavi, R., Longbotham, R., & Tang, D. (2009). *Testing for Usability in Online Systems*. ACM Conference on Human Factors in Computing Systems.

#### 4. Daftar Bacaan

1. Sama seperti pada daftar referensi

#### 5. JADWAL PERKULIAHAN DAN TOPIK BAHASAN

Pertemuan Ke-	TOPIK BAHASAN	BACAAN
1	a. Kontrak Perkuliahan, Perkenalan dan Penjelasan b. Pengenalan Rekayasa Perangkat Lunak	Kontrak Perkuliahan
2	a. Karakteristik perangkat lunak b. Komponen perangkat lunak c. Model perangkat lunak d. Fungsi dan peran dari software engineer	1-6
3	a. Definisi SDLC b. Jenis-jenis SDLC	Idem
4	a. Observasi dan estimasi dalam perencanaan proyek b. Tujuan perencanaan proyek c. Manajemen proyek perangkat lunak yang efektif	Idem
5	a. Proses analisis kebutuhan b. Metode analisis kebutuhan c. Spesifikasi dan validasi kebutuhan	Idem
6	a. Perangkat bantu proses analisis kebutuhan b. Konsep dasar, Konteks, Proses, dan Prinsip Perancangan Perangkat Lunak; c. Isu mendasar dalam perancangan perangkat lunak	Idem
7	a. Alat bantu perancangan (DFD dan UML) b. Macam-macam diagram yang terdapat pada UML (Class Diagram, Use Case Diagram, Activity Diagram, Sequence Diagram)	Idem

<b>8</b>	<b>UTS</b>	
9	a. Konsep dalam User Interface b. Prinsip Desain Antarmuka (user experience, user guidance, user diversity)	Idem
10	a. Perencanaan dalam pengujian b. Proses testing: (black box testing, white box testing) c. Integration testing dan user testing d. Faults, Error dan Failures	Idem
11	Review Teknik Pengujian Perangkat Lunak dari proses testing	Idem
12	a. Pengujian unit b. Pengujian integrasi c. Pengujian sistem d. Debugging dan quality assurance	Idem
13	a. Quality assurance pada perangkat lunak b. Keamanan data akses	Idem
14	a. Definisi pemeliharaan perangkat lunak. b. Konsep Pemeliharaan Perangkat lunak	Idem
15	Teknik pemeliharaan perangkat lunak (Pemeliharaan korektif, pemeliharaan adaptif, pemeliharaan perfektif, pemeliharaan preventif)	Idem
<b>16</b>	<b>UAS</b>	