

**PERANGKAT BANTU PROSES**  
**ANALISIS KEBUTUHAN**  
**Mata Kuliah: Software Engineering**



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

**Pertemuan ke-6**

**Topik Bahasan ke-13**

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

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

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

Nama Mata Kuliah	: Software Engineeing/Rekayasa Perangkat Lunak
Kode Mata Kuliah/SKS	: MKP ____/3 SKS
Dosen	: <b>Yudhi Fajar Saputra,</b>
Semester	: <b>3/ 2024</b>
Hari Pertemuan / Jam	: -
Tempat Pertemuan	: <b>Ruang Kelas A.06</b>

Proses analisis kebutuhan perangkat lunak melibatkan berbagai tahap, mulai dari pengumpulan kebutuhan hingga validasi dan dokumentasi. Untuk mendukung proses ini, ada berbagai perangkat bantu yang dirancang untuk memfasilitasi analisis kebutuhan. Berikut adalah beberapa perangkat bantu yang umum digunakan dalam proses analisis kebutuhan perangkat lunak:.

## **1. PERANGKAT UNTUK PENGUMPULAN DAN MANAJEMEN KEBUTUHAN**

Perangkat untuk pengumpulan dan manajemen kebutuhan adalah alat-alat yang digunakan untuk mengumpulkan, mendokumentasikan, mengelola, dan melacak kebutuhan perangkat lunak dari berbagai pemangku kepentingan dan pengguna potensial selama proses pengembangan. Berikut adalah beberapa perangkat populer yang sering digunakan untuk tujuan ini:

### **a. JIRA**

JIRA adalah alat manajemen proyek yang digunakan dalam pengumpulan dan manajemen kebutuhan dalam bentuk *user stories*, *epics*, dan tugas yang bisa dipecah menjadi unit kerja yang lebih kecil. JIRA sering digunakan dalam metodologi Agile <sup>[1]</sup>, JIRA menyediakan fitur pelacakan, pelaporan, dan manajemen backlog.

### **b. IBM Engineering Requirements Management DOORS**

DOORS adalah alat manajemen kebutuhan yang sangat kuat, terutama untuk proyek besar dan kompleks yang memerlukan pelacakan kebutuhan yang ketat dari definisi hingga implementasi, terutama dalam industri yang memerlukan kepatuhan terhadap standar tertentu (misalnya, otomotif, aerospace) <sup>[2]</sup>. DOORS mendukung dokumentasi kebutuhan, pelacakan perubahan, dan manajemen versi

### **c. Manfaat SRS**

## **2. PERANGKAT UNTUK PEMODELAN KEBUTUHAN**

Kuesioner adalah metode yang digunakan untuk mengumpulkan informasi dari sejumlah besar pemangku kepentingan dari formulir yang berisikan pertanyaan.

Kuesioner dapat berupa pertanyaan tertutup (dengan jawaban pilihan ganda) atau pertanyaan terbuka. Kendall dan Kendall menekankan bahwa kuesioner sangat efektif ketika perlu menjangkau banyak orang dalam waktu singkat, namun harus dirancang dengan hati-hati untuk menghindari bias dan memastikan bahwa semua aspek penting dicakup [2].

**Kelebihan:** Efisien dalam menjangkau banyak responden, mudah dianalisis secara statistik.

**Kekurangan:** Jawaban yang diberikan mungkin tidak mendalam, responden bisa saja tidak memahami pertanyaan dengan baik.

### 3. PERANGKAT UNTUK WIREFRAMING DAN PROTOTYPING

Observasi adalah metode dengan melakukan pengamatan langsung terhadap bagaimana pengguna berinteraksi dengan sistem saat ini atau proses bisnis yang ada. Menurut Pressman dan Maxim, observasi membantu dalam memahami kebutuhan yang mungkin tidak diungkapkan oleh pengguna karena mereka sudah terbiasa dengan cara kerja tertentu [3]. Metode ini sangat berguna untuk mengidentifikasi kebutuhan yang tidak disadari oleh pengguna.

**Kelebihan:** Memberikan wawasan nyata tentang bagaimana pengguna bekerja dan apa yang mereka butuhkan.

**Kekurangan:** Bisa mengganggu pekerjaan pengguna, dan analisisnya bisa saja subyektif.

### 4. PERANGKAT UNTUK DOKUMENTASI KEBUTUHAN

Metode analisis dokumen penting untuk mengidentifikasi kebutuhan yang telah terdokumentasi sebelumnya dan untuk memastikan bahwa tidak ada kebutuhan yang diabaikan. Ini membantu analis untuk memahami konteks dan sejarah dari sistem yang akan dikembangkan atau diubah. Pohl menjelaskan bahwa analisis dokumen melibatkan peninjauan terhadap dokumen yang sudah ada, seperti spesifikasi sistem sebelumnya, manual pengguna, dan laporan [4].

**Kelebihan:** Menghemat waktu, memberikan pandangan tentang kebutuhan yang telah terdokumentasi sebelumnya.

**Kekurangan:** Dokumen mungkin tidak lengkap atau tidak mutakhir, sehingga informasi yang diperoleh bisa kurang relevan.

### 5. PERANGKAT UNTUK PENGUJIAN KEBUTUHAN

Focus group discussion adalah diskusi terstruktur dengan sekelompok pemangku kepentingan yang difasilitasi oleh analis. Robertson dan Robertson menekankan bahwa metode ini memungkinkan berbagai perspektif untuk diungkapkan dan dibahas secara bersama-sama [5]. Focus group sangat berguna untuk memahami

kebutuhan yang lebih umum dan untuk mengidentifikasi perbedaan pandangan di antara pemangku kepentingan sehingga mendapatkan berbagai perspektif secara bersamaan.

**Kelebihan:** Menghasilkan berbagai perspektif, efektif untuk menggali kebutuhan yang bersifat umum.

**Kekurangan:** Bisa didominasi oleh individu tertentu, sulit untuk menangani perbedaan pendapat

## 6. PERANGKAT UNTUK KOLABORASI DAN VALIDASI KEBUTUHAN

Prototyping melibatkan pembuatan model awal dari sistem yang diusulkan untuk menguji dan memvalidasi kebutuhan sebagai sample awal sistem yang dibangun. McConnell menjelaskan bahwa prototyping sangat berguna ketika kebutuhan tidak sepenuhnya dipahami di awal proyek [6]. Pengguna dapat berinteraksi dengan prototipe untuk memberikan tanggapan, yang kemudian digunakan untuk memperbaiki dan mengklarifikasi kebutuhan sebelum pengembangan penuh dimulai.

**Kelebihan:** Membantu dalam mengidentifikasi kebutuhan yang lebih konkret, pengguna dapat melihat dan merasakan sistem sebelum finalisasi.

**Kekurangan:** Memerlukan waktu dan biaya tambahan, bisa mengakibatkan kebutuhan yang berlebihan jika pengguna menginginkan fitur yang tidak sesuai dengan ruang lingkup proyek.

## 7. PERANGKAT UNTUK PENGAMBILAN KEPUTUSAN

Workshop adalah sesi kolaboratif di mana pemangku kepentingan berkumpul untuk mengidentifikasi, mendiskusikan, dan memprioritaskan kebutuhan. Windle menekankan bahwa workshop memungkinkan pemangku kepentingan untuk berkolaborasi secara langsung, menghasilkan konsensus, dan memberikan pemahaman yang lebih mendalam tentang kebutuhan dari berbagai perspektif [7]. Ini juga merupakan cara yang efisien untuk mengumpulkan kebutuhan dari banyak orang dalam waktu yang singkat.

**Kelebihan:** Efektif dalam menyatukan berbagai pandangan, cepat dalam mengumpulkan kebutuhan dari banyak pihak.

**Kekurangan:** Memerlukan perencanaan yang matang, mungkin tidak cocok untuk proyek dengan banyak pemangku kepentingan.

## 8. PERANGKAT UNTUK PENGELOLAAN VERSI DAN PELACAKAN PERUBAHAN

Use case modeling adalah teknik untuk mendokumentasikan interaksi antara pengguna (aktor) dan sistem dalam bentuk skenario. Overgaard dan Palmkvist menjelaskan bahwa use case membantu dalam mengidentifikasi kebutuhan

fungsi sistem dengan cara yang terstruktur dan mudah dipahami oleh pemangku kepentingan [8]. Setiap use case menggambarkan bagaimana sistem akan merespons tindakan yang dilakukan oleh aktor dalam situasi tertentu, maka dari itu Teknik ini sangat berguna dalam memvisualisasikan bagaimana sistem akan digunakan dalam situasi dunia nyata.

**Kelebihan:** Membantu dalam mengidentifikasi kebutuhan fungsional, mudah dipahami oleh semua pihak yang terlibat.

**Kekurangan:** Fokus pada kebutuhan fungsional saja, sehingga mungkin kurang dalam menangkap kebutuhan non-fungsional.

**KESIMPULAN:** Menggunakan perangkat bantu yang tepat dalam proses analisis kebutuhan membantu memastikan bahwa kebutuhan dikumpulkan, dianalisis, dan didokumentasikan dengan benar, sehingga meminimalkan risiko kesalahan dan meningkatkan kualitas perangkat lunak yang dikembangkan.

## 9. DAFTAR REFERENSI

1. Karl Wiegers dan Joy Beatty. (2013). Software Requirements (Developer Best Practices) 3rd Edition. Microsoft Press. ISBN-10: 0735679665
2. Hooks, I. F., & Farry, K. A. (2022). Customer-Centered Products: Creating Successful Products through Smart Requirements Management. Amacom Books. ISBN-10: 0814405681
3. IEEE. IEEE Standard 830-1998 - Recommended Practice for Software Requirements Specifications. ISBN:978-0-7381-0448-5. Reterived at 08 August 2024 from <https://ieeexplore.ieee.org/document/720574>
4. Sommerville, Ian. (2015). Software Engineering 10th Edition. Pearson Education, Inc.. ISBN-13: 978-0-13-703515-1.
5. Roger S. Pressman S. R, Maxim. B. (2019). Software Engineering: A Practitioner's Approach 9th Edition. McGraw Hill. ISBN 9780078022128.
- 6.
7. Kenneth E. Kendall dan Julie E. Kendall (2014). Systems Analysis and Design. Pearson. ISBN-10: 013478555X. 2014
8. Roger S. Pressman S. R, Maxim. B. (2019). Software Engineering: A Practitioner's Approach 9th Edition. McGraw Hill. ISBN 9780078022128.
9. Klaus Pohl (2010). Requirements Engineering: Fundamentals, Principles, and Techniques. Springer. ISBN-10: 3642125778.
10. Suzanne Robertson & James Robertson (2013). Mastering the Requirements

Process: Getting Requirements Right. Addison-Wesley Professional. ISSN-10: 0321815742

11. Steve McConnell. (1996). Rapid Development: Taming Wild Software Schedules. Microsoft Press. ISBN-10: 9781556159008

12. Daniel R. Windle. (2002). Software Requirements Using the Unified Process: A Practical Approach. Prentice Hall. ISBN-10: 0130969729.

13. Gunnar Overgaard & Karin Palmkvist. (2005). Applying Use Cases: A Practical Guide. Addison-Wesley Professional. ISBN-10: 0201708531.

14. Dean Leffingwell & Don Widrig. (2003). Managing Software Requirements: A Unified Approach. Addison-Wesley. ISBN-10: 0201615932

15. Benjamin M. Brothers (2011). The Art of Software Modeling. Auerbach Publications. ISBN-10: 1420044621

## 10. Daftar Bacaan

1. Sama seperti pada daftar referensi

## 11. JADWAL PERKULIAHAN DAN TOPIK BAHASAN

<b>Pertemuan Ke-</b>	<b>TOPIK BAHASAN</b>	<b>BACAAN</b>
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	Idem

	<ul style="list-style-type: none"> <li>b. Konsep dasar, Konteks, Proses, dan Prinsip Perancangan Perangkat Lunak;</li> <li>c. Isu mendasar dalam perancangan perangkat lunak</li> </ul>	
7	<ul style="list-style-type: none"> <li>a. Alat bantu perancangan (DFD dan UML)</li> <li>b. Macam-macam diagram yang terdapat pada UML (Class Diagram, Use Case Diagram, Activity Diagram, Sequence Diagram)</li> </ul>	Idem
<b>8</b>	<b>UTS</b>	
9	<ul style="list-style-type: none"> <li>a. Konsep dan Isu dalam</li> <li>b. Desain User Interface</li> <li>c. Prinsip Desain antarmuka (user experience, user guidance, user diversity)</li> <li>d. Software configuration management: definisi dan skenario kerja</li> </ul>	Idem
10	<ul style="list-style-type: none"> <li>a. Perencanaan dalam pengujian</li> <li>b. Proses testing: (black box testing, white box testing)</li> <li>c. Integration testing dan user testing</li> <li>d. Faults, Error dan Failures</li> </ul>	Idem
11	Review Teknik Pengujian Perangkat Lunak dari proses testing	Idem
12	<ul style="list-style-type: none"> <li>a. Pengujian unit</li> <li>b. Pengujian integrasi</li> <li>c. Pengujian sistem</li> <li>d. Debugging dan quality assurance</li> </ul>	Idem
13	<ul style="list-style-type: none"> <li>a. Quality assurance pada perangkat lunak</li> <li>b. Keamanan data akses</li> </ul>	Idem
14	<ul style="list-style-type: none"> <li>a. Definisi pemeliharaan perangkat lunak.</li> <li>b. Konsep Pemeliharaan Perangkat lunak</li> </ul>	Idem
15	Teknik pemeliharaan perangkat lunak (Pemeliharaan korektif, pemeliharaan adaptif, pemeliharaan perfektif, pemeliharaan preventif)	Idem
<b>16</b>	<b>UAS</b>	