KARAKTERISTIK PERANGKAT LUNAK

Software Engineering

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

Pertemuan ke-2

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	: **Yudhi Fajar Saputra,**
Semester	: **3/ 2024**
Hari Pertemuan / Jam	: -
Tempat Pertemuan	: **Ruang Kelas A.06**

Software Engineering/Rekayasa perangkat lunak adalah proses merancang, mengembangkan, dan memelihara sistem software. Pastinya software yang baik adalah software yang bisa memenuhi kebutuhan penggunanya, menjalankan fungsi yang diharapkan dengan bagus, dan bisa dimaintanance. Berdasarkan ISO/IEC 25002:2024, software quality diklasifikasikan ke beberapa karakteristik dan sub-karakteristik [1], diantaranya adalah sebagai berikut:

1. **FUNCTIONALITY**

   Functionality terkait terhadap fitur-fitur dan kemampuan yang disediakan oleh program atau software kepada penggunanya. Idealnya semakin banyak fungsi atau fitur yang dimiliki suatu program atau software, maka akan menjadi semakin super aplikasi tersebut dan menjadi aplikasi atau software serbaguna, namun juga semakin kompleks aplikasi atau software tersebut. Akan tetapi hal yang penting dilakukan adalah menyeimbangkan kebutuhan akan fungsionalitas dengan kebutuhan pengguna dan juga kemudahan dalam penggunaan, pemeliharaan, dan skalabilitas.

   Fungsionality yang dimaksud adalah:

   1. Suitability/ Kesesuaian
   2. Accuracy/ Ketepatan
   3. Interoperability/ Kemampuan Interaksi Antar Aplikasi
   4. Security/ Keamanan
   5. Compliance/Sesuai Prosedur

   Fakta menarik: *Riset yang dilakukan oleh GoodFirms Research menyatakan bahwa sebanyak 53,80% perusahaan sebagai peserta responden mengungkapkan bahwa memenuhi revisi klien yang selalu berubah merupakan tantangan terbesar dalam mengembangkan aplikasi/software bagi tim pengembang.* [2]

2. **RELIABILITY**

   Reliability adalah karakteristik aplikasi/software yang mengacu pada kemampuannya untuk menjalankan fungsi yang diinginkan dengan benar dan konsisten dalam beberapa waktu. Alasan Reliabity merupakan aspek penting dari kualitas aplikasi/software, karena membantu memastikan bahwa aplikasi atau

software akan bekerja secara benar dan tidak mengalami error diluar dugaan.

Berdasarkan ISO/IEC 25002:2024, Beberapa aspek yang mempengaruhi kemampuan aplikasi atau software untuk mempertahankan tingkat kinerja dari aplikasi atau software tersebut adalah sebagai berikut:

1. Maturity/ Kematangan: Maturity adalah aspek yang terkait kesiap sediaan sistem atau aplikasi untuk digunakan.
2. Fault tolerance/ Toleransi kesalahan: Fault tolerance adalah aspek yang terkait tentang kemampuan suatu sistem untuk menahan kegagalan dalam menjalankan program.
3. Recoverability/ Pemulihan: Recoverability terkait kemampuan untuk memulihkan sistem dengan cepat dan efisien.
4. Compliance/Sesuai Prosedur: Aspek Compliance akan memberikan perkiraan tingkat risiko dan kemungkinan potensi kegagalan dan error aplikasi yang akan terjadi ketika dioperasikan.

Fakta menarik: *GoodFirms Research dalam penelitian nya mempublikasikan bahwa pengembangan aplikasi/software mencakup identifikasi kebutuhan, perencanaan, desain, pengembangan, pengujian, dan penerapan, memerlukan rentang waktu dari satu hingga sembilan bulan. Menariknya sebagian besar perusahaan yang disurvei membutuhkan jangka waktu rata-rata 4,5 bulan dalam pengembangan aplikasi/software, yakni sebesar 38,50% menyatakan butuh waktu 4-5 bulan dalam membangun aplikasi/software.* [2]

3. USABILITY

Usability mengacu pada sejauh mana aplikasi atau software tersebut dapat digunakan dengan mudah. jumlah usaha atau waktu yang diperlukan untuk mempelajari cara menggunakan perangkat lunak.

Berdasar dari Capaian Pembelajaran Mata Kuliah, maka mempelajari SE/RPL ini sangat perlu. Berikut adalah tujuan dari rekayasa perangkat lunak:

1. Membangun dan mengembangkan Software yang Lebih Berkualitas
2. Membangun dan mengembangkan Software yang Lebih Mudah Digunakan
3. Membangun dan mengembangkan Software Lebih Hemat dan Efisien
4. Membangun dan mengembangkan Software yang Tepat Waktu
5. Membangun dan mengembangkan Software Sesuai Kebutuhan
6. Update Software Agar Lebih Fungsional dari waktu ke waktu
7. Membantu Memelihara dan Merawat Software

8. Membantu keamanan software agar lebih optimal

4. EFFICIENCY

Mengutip dari wikipedia Outline atau Ruang lingkup dari SE/RPL adalah sebagai berikut:

5. MAINTAINABILITY

Mengutip dari wikipedia Outline atau Ruang lingkup dari SE/RPL adalah sebagai berikut:

6. PORTABILITY

Mengutip dari wikipedia Outline atau Ruang lingkup dari SE/RPL adalah sebagai berikut:

7. DAFTAR REFERENSI

1. ISO/IEC 25002:2024. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Quality model overview and usage. Diakses pada 08 Juli 2024 dari https://www.iso.org/standard/78175.html.
2. Mark Raymond. Remarkably Useful Stats and Trends on Software Development | GoodFirms Research. Diakses pada 08 Juli 2024 dari https://www.goodfirms.co/resources/software-development-research.
3.
4. Stephen R.Schach. Object-Oriented and Classical Software Engineering 8th Edition. McGraw Hill. ISBN : 0073376183. 2010.
5. Fritz Bauer. Software Engineering. Information Processing. 71: 530–538.
6. Wikipedia Indonesia. Rekayasa Perangkat Lunak . Diakses pada 08 Juli 2024 dari https://id.wikipedia.org/wiki/Rekayasa_perangkat_lunak.
7. Randall, Brian. The 1968/69 NATO Software Engineering Reports. Retrieved 17 juli 2024 dari http://homepages.cs.ncl.ac.uk/brian.randell/NATO/NATOReports/.
8. Linda Hutz Pesante (January 1, 2003). Anthony Ralston; Edwin D. Reilly (eds.). Software engineering institute (SEI). Encyclopedia of Computer Science. Chichester, West Sussex, UK: John Wiley and Sons Ltd.: 1611–1613. ISBN 978-0-470-86412-8.
9. GeeksforGeeks. Software Engineering | Software Design Process. Retrieved 10 Juli 2024 dari https://www.geeksforgeeks.org/software-engineering-software-design-process/.
10. Pierre Bourque; Richard E. (Dick) Fairley, eds. (2014). Guide to the Software Engineering Body of Knowledge Version 3.0 (SWEBOK). IEEE Computer Society. Retrieved 10 Juli 2024 dari https://www.computer.org/education/bodies-of-knowledge/software-engineering. 2014.
11. Wikipedia. List of programming languages. Diakses pada 08 Juli 2024 dari https://en.wikipedia.org/wiki/List_of_programming_languages.
12. Global Market Insights. Software Testing Market Size. Retrieved 10 Juli 2024 dari https://www.gminsights.com/industry-analysis/software-testing-market.
13. Ivan Blagojević. Software Testing Statistics. Retrieved 10 Juli 2024 dari https://99firms.com/blog/software-testing-statistics/ .

14. Wikipedia Commons. Ada Lovelace. Retrieved 10 Juli 2024 dari https://en.wikipedia.org/wiki/Ada_Lovelace

15. Gartner, Magic Quadrant for Enterprise Low-Code Application Platforms, . Retrieved 10 Juli 2024 dari https://www.mendix.com/resources/gartner-magic-quadrant-for-low-code-application-platforms/

16. Evans Data Corporation. New Developer Population Numbers Worldwide 2023-2028. Retrieved 10 Juli 2024 dari https://www.evansdata.com/press/viewRelease.php?pressID=339.

17. Economic Times. What is 'Software Maintenance'. Retrieved 10 Juli 2024 dari https://economictimes.indiatimes.com/definition/software-maintenance.

8. Daftar Bacaan

1. Etet
2. Tes
3. Tes
4.

9. 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
8	**UTS**	
9	a. Konsep dan Isu dalam b. Desain User Interface c. Prinsip Desain antarmuka (user experience, user guidance, user diversity) d. Software configuration management: definisi dan skenario kerja	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	Idem

	testing	
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
16	**UAS**	