

PENGENALAN REKAYASA PERANGKAT LUNAK

Software Engineering



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

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

1. Definisi Rekayasa Perangkat Lunak

- IEEE Standard Glossary of Software Engineering Terminology, IEEE std 610.12-1990: "The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software." Dalam bahasa Indonesia "penerapan suatu pendekatan yang sistematis, disiplin dan terkuantifikasi atas pengembangan, penggunaan dan pemeliharaan peranti lunak."^[1]
- Stephen R.Schach dalam bukunya Object-Oriented and Classical Software Engineering mengartikan bahwa rekayasa perangkat lunak adalah sebuah disiplin ilmu yang mana dibuat untuk kepentingan menghasilkan perangkat lunak yang bebas dari kesalahan, pengiriman yang tepat waktu, dan memuaskan keinginan pemakainya.^[2]
- Fritz Bauer seorang Computer Scientist Jerman, mendefinisikan rekayasa perangkat lunak. Software engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and work efficiently on real machines. Yang mana ia mengartikan RPL sebagai pengembangan dan penggunaan prinsip rekayasa dalam rangka memperoleh perangkat lunak yang bisa dipercaya dan bekerja secara efisien dan dilakukan pada mesin nyata.^[3]
- Sedangkan Dari Wikipedia bahasa Indonesia: Rekayasa Perangkat Lunak adalah satu bidang profesi yang mendalami cara-cara pengembangan peranti lunak termasuk pembuatan, pemeliharaan, manajemen organisasi pengembangan Perangkat lunak dan manajemen mutu. ^[4]

Jika kita breakdown Rekayasa Perangkat Lunak atau Software Engineering memiliki dua kata Software dan Engineering. Bagi saya software adalah lebih dari sekedar program. dimana Program adalah kode yang dapat dieksekusi, yang melayani beberapa tujuan komputasi. Sedangkan Software dianggap sebagai kumpulan kode pemrograman yang dapat dieksekusi, memiliki library terkait, dan terdapat pendokumentasian. Di sisi lain, Engineering adalah tentang mengembangkan produk, menggunakan prinsip dan metode ilmiah yang terdefinisi dengan baik.

Sehingga Rekayasa perangkat lunak atau Software engineering dapat didefinisikan suatu cabang teknik yang terkait dengan pengembangan produk perangkat lunak

menggunakan prinsip, metode, dan prosedur ilmiah yang terdefinisi dengan baik, mulai dari Design , Construction , Testing , Program analysis , pembuatannya, pemeliharaan hingga manajemen organisasi dan manajemen kualitasnya.

2. Evolusi

Software Engineering atau Rekayasa Perangkat Lunak telah berkembang dari yang dianggap sebagai seni menjadi disiplin teknik yang diakui. Dimana Pada masa awal komputasi sekitar awal tahun 1937, pengembangan perangkat lunak dilakukan oleh individu atau tim kecil yang menulis kode berdasarkan pengalaman dan pengetahuan mereka sendiri. Pendekatan ini sering disebut sebagai “programming by intuition”. Seiring berkembangnya bidang komputasi, Software Engineering mengalami beberapa evolusi:

1. Tahun 1960,

Pada tahun 1960an dan 1970an, bidang Software Engineering atau Rekayasa Perangkat Lunak mulai terbentuk. Yang mana Mulai tahun 1960an, rekayasa perangkat lunak diakui sebagai bidang teknik yang diakui. Pada tahun-tahun tersebut para peneliti dan praktisi mulai mengembangkan metode formal untuk desain dan pengembangan perangkat lunak, seperti pemrograman terstruktur dan penggunaan diagram alur untuk mewakili algoritma. Akan tetapi Perkembangan rekayasa perangkat lunak pada dekade tersebut dipandang sebagai sebuah perjuangan. Masalahnya perangkat lunak yang melebihi anggaran, melebihi tenggat waktu, memerlukan debugging dan pemeliharaan ekstensif, dan tidak berhasil memenuhi kebutuhan konsumen atau bahkan tidak pernah selesai.

2. Tahun 1968,

Pada tahun ini sebuah konferensi tentang rekayasa perangkat lunak diadakan, dimana NATO mengadakan konferensi rekayasa perangkat lunak pertama yang membahas isu-isu terkait perangkat lunak. Pedoman dan praktik terbaik untuk pengembangan perangkat lunak telah ditetapkan^[5]. Dan finally istilah “rekayasa perangkat lunak” secara resmi diciptakan, lalu Dalam dekade berikutnya, rekayasa perangkat lunak terus berkembang dan matang

3. Tahun 1984,

Pada tahun ini Software Engineering Institute (SEI) didirikan sebagai pusat penelitian dan pengembangan yang didanai pemerintah federal yang berkantor pusat di kampus Universitas Carnegie Mellon di Pittsburgh, Pennsylvania, Amerika Serikat^[6]. Watts Humphrey mendirikan Program Proses Perangkat Lunak SEI, yang bertujuan untuk memahami dan mengelola proses rekayasa perangkat lunak^[6].

4. Pada tahun 1990,

Tahun 1990an muncul metodologi pengembangan perangkat lunak Agile, yang menekankan fleksibilitas dan daya tanggap terhadap perubahan.

Sehingga Setelah tahun 1990an , Software engineering sudah dianggap sebagai disiplin ilmu teknik dengan serangkaian praktik terbaik, metodologi, dan alatnya sendiri serta mengikuti prinsip yang sama seperti bidang teknik lainnya.

3. TUJUAN SOFTWARE ENGINEERING/REKAYASA 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. OUTLINE/RUANG LINGKUP SOFTWARE ENGINEERING REKAYASA PERANGKAT LUNAK

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

1. Requirements. Requirement terkait dengan berbagai spesifikasi persyaratan dan keperluan dari software.

Fakta Unik: Pemrogram komputer pertama adalah seorang wanita bernama Ada Lovelace. Dia adalah seorang ahli matematika dan intelektual yang menerjemahkan artikel Italia tentang Mesin Analitik pada tahun 1843, dengan menerjemahkan artikel tentang mesin analitik ia lalu mendeskripsikan bagaimana kode untuk sebuah perangkat lunak dapat mengolah huruf, simbol, dan sekaligus angka. Dengan kode ini, ia memprediksi kemungkinan mesin tersebut untuk mengolah musik, grafis, dan banyak fungsi lain. Ia juga menuliskan teori metode kalkulasi nomor Bernoulli pada mesin . [12]

2. Design. Design terkait dengan proses pembuatan rencana dan rancangan dalam membangun dan mengembangkan software, dimana Desain dapat terbagi menjadi beberapa tingkatan:

- a. Interface design, ini adalah tahapan desain dalam merencanakan interaksi

antara sistem dan environment(hal-hal yang mendukung sebuah software) serta cara kerja sistem.

- b. Architectural design, merencanakan komponen utama suatu sistem, termasuk tanggung jawab, properti, dan antarmuka di antara komponen-komponen tersebut.
- c. Detailed design, merencanakan elemen internal, termasuk properti, relationship, algoritma, dan struktur datanya. [7]

Fakta Unik: Menurut Gartner, lebih dari 65% pengembangan aplikasi pada tahun 2024 akan dilakukan dengan low code programming [13].

- 3. Construction. Construction mencakup pemrograman (Coding), unit testing, integration testing, dan debugging untuk mengimplementasikan desain.[8] maksud dari Pengujian selama fase ini dilakukan oleh programmer dengan tujuan untuk memverifikasi bahwa coding yang sudah dibangun seperti yang dirancang sebelumnya dan untuk mengetahui kapan kode tersebut siap untuk pengujian tingkat berikutnya.

Fakta Unik: Tahukah kamu? Ada sekitar 700 bahasa pemrograman. [9]

- 4. Testing. Testing mencakup ujicoba teknis secara empiris (dibuktikan secara pengamatan langsung dan pengalaman penggunaan), ini berguna untuk memberikan informasi kepada user yang menggunakan tentang kualitas software yang diuji cobakan. [8] .

Fakta Unik: Market software testing mencapai lebih dari \$40 miliar di tahun 2020 [10][11].

- 5. Analysis. Analysis terkait proses menganalisis software yang berhubungan dengan aspek kinerja, ketahanan, dan keamanan.

Fakta Unik: Software Developers mencapai 27,7 juta pada tahun 2023. Di masa tahun 2024, jumlah tersebut diperkirakan akan mencapai lebih dari 28,7 juta.[14]

- 6. Maintenance. Maintenance terkait pada dukungan untuk software atau perangkat lunak setelah dirilis. seperti koreksi kesalahan, pengoptimalan, penghapusan fitur yang tidak digunakan, dan peningkatan fitur. [8]

Fakta Unik: Maintenance software rata-rata bisa memakan 40% hingga 80% biaya proyek.[15]

5. DAFTAR REFERENSI

1. IEEE Standard Glossary of Software Engineering Terminology, IEEE std 610.12-1990, 1990.
2. Stephen R.Schach. Object-Oriented and Classical Software Engineering 8th

Edition. McGraw Hill. ISBN : 0073376183. 2010.

3. Fritz Bauer. Software Engineering. Information Processing. 71: 530–538.
4. Wikipedia Indonesia. Rekayasa Perangkat Lunak . Diakses pada 08 Juli 2024 dari https://id.wikipedia.org/wiki/Rekayasa_perangkat_lunak.
5. Randall, Brian. The 1968/69 NATO Software Engineering Reports. Retrieved 17 juli 2024 dari <http://homepages.cs.ncl.ac.uk/brian.randell/NATO/NATOREports/>.
6. 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.
7. GeeksforGeeks. Software Engineering | Software Design Process. Retrieved 10 Juli 2024 dari <https://www.geeksforgeeks.org/software-engineering-software-design-process/>.
8. 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.
9. Wikipedia. List of programming languages. Diakses pada 08 Juli 2024 dari https://en.wikipedia.org/wiki/List_of_programming_languages.
10. Global Market Insights. Software Testing Market Size. Retrieved 10 Juli 2024 dari <https://www.gminsights.com/industry-analysis/software-testing-market>.
11. Ivan Blagojević. Software Testing Statistics. Retrieved 10 Juli 2024 dari <https://99firms.com/blog/software-testing-statistics/>.
12. Wikipedia Commons. Ada Lovelace. Retrieved 10 Juli 2024 dari https://en.wikipedia.org/wiki/Ada_Lovelace
13. 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/>
14. Evans Data Corporation. New Developer Population Numbers Worldwide 2023-2028. Retrieved 10 Juli 2024 dari <https://www.evansdata.com/press/viewRelease.php?pressID=339>.
15. Economic Times. What is 'Software Maintenance'. Retrieved 10 Juli 2024 dari <https://economictimes.indiatimes.com/definition/software-maintenance>.

6. 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	Idem

	Perancangan Perangkat Lunak; c. Isu mendasar dalam perancangan perangkat lunak	
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 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
16	UAS	