

**PEMELIHARAAN KOREKTIF, PEMELIHARAAN
ADAPTIF, PEMELIHARAAN PERFEKTIF,
PEMELIHARAAN PREVENTIF DALAM
PEMELIHARAAN PERANGKAT LUNAK
Mata Kuliah: Software Engineering**



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

Pertemuan ke-6

Topik Bahasan ke-15

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**

Pemeliharaan perangkat lunak merupakan bagian integral dari siklus hidup perangkat lunak yang bertujuan untuk memastikan bahwa perangkat lunak terus berfungsi dengan baik setelah diterapkan. Pemeliharaan ini dapat dibagi menjadi beberapa jenis, masing-masing dengan fokus dan tujuan yang berbeda, tergantung pada masalah atau kebutuhan yang dihadapi. **Pemeliharaan korektif** berfokus pada perbaikan bug atau kesalahan yang ditemukan setelah perangkat lunak digunakan, sehingga memastikan sistem tetap berjalan dengan lancar. **Pemeliharaan adaptif** dilakukan untuk menyesuaikan perangkat lunak dengan perubahan yang terjadi di lingkungan operasional, seperti pembaruan perangkat keras, perangkat lunak, atau perubahan kebijakan. Sementara itu, **pemeliharaan perfektif** berfokus pada peningkatan fungsionalitas dan kinerja perangkat lunak berdasarkan umpan balik dari pengguna atau perubahan dalam kebutuhan bisnis. Terakhir, **pemeliharaan preventif** berusaha untuk mencegah masalah di masa depan dengan memperbaiki kode atau sistem yang berpotensi menyebabkan kesalahan atau kerusakan. Keempat jenis pemeliharaan ini bekerja secara bersamaan untuk menjaga perangkat lunak tetap efisien, relevan, dan aman sepanjang siklus hidupnya[1,2].

1. PEMELIHARAAN KOREKTIF

Pemeliharaan korektif berfokus pada perbaikan kesalahan (bug) atau masalah yang ditemukan setelah perangkat lunak diterapkan dan digunakan. Kesalahan ini bisa terdeteksi selama penggunaan normal perangkat lunak atau melalui laporan dari pengguna. Tujuan dari pemeliharaan ini adalah untuk memperbaiki kesalahan yang menghalangi fungsionalitas perangkat lunak atau menyebabkan kerusakan.

Karakteristik Pemeliharaan Korektif:

- 1) Perbaikan kesalahan atau bug setelah perangkat lunak digunakan: Pemeliharaan korektif berfokus pada identifikasi dan perbaikan kesalahan fungsional yang ada dalam perangkat lunak setelah penggunaan. Ini bisa mencakup masalah teknis seperti crash, bug dalam algoritma, atau kesalahan yang menghalangi pengguna mencapai hasil yang diinginkan
- 2) Penyelesaian Masalah yang Ditemukan Setelah Implementasi: kesalahan ini ditemukan setelah perangkat lunak diterapkan dalam lingkungan nyata. Pengguna atau penguji perangkat lunak mungkin menemukan masalah yang

tidak terdeteksi selama fase pengujian awal, yang kemudian perlu diperbaiki

- 3) Respon terhadap Laporan Pengguna: Salah satu sumber utama dari pemeliharaan korektif adalah laporan masalah atau keluhan dari pengguna akhir yang berinteraksi dengan perangkat lunak. Pengguna dapat melaporkan kesalahan yang mereka temui selama penggunaan sehari-hari, yang kemudian direspons oleh tim pengembang.

Contoh: Jika aplikasi spada mengalami crash saat melakukan penyelesaian ujian, pemeliharaan korektif akan dilakukan untuk memperbaiki bug tersebut.

2. PEMELIHARAAN ADAPTIF

Pemeliharaan adaptif adalah jenis pemeliharaan perangkat lunak yang berfokus pada penyesuaian perangkat lunak terhadap perubahan yang terjadi di lingkungan operasionalnya. Perubahan tersebut bisa melibatkan pembaruan pada perangkat keras, perangkat lunak lain, sistem operasi, atau bahkan kebijakan organisasi yang mempengaruhi cara perangkat lunak dijalankan. Tujuan utama dari pemeliharaan adaptif adalah untuk memastikan bahwa perangkat lunak tetap kompatibel dan berfungsi dengan baik meskipun ada perubahan luar yang mempengaruhi sistem.

Perangkat lunak tidak berdiri sendiri; dimana selalu berada dalam lingkungan yang dinamis, yang bisa berubah seiring waktu. Oleh karena itu, pemeliharaan adaptif sangat penting untuk memastikan perangkat lunak tidak hanya berfungsi dengan baik pada waktu tertentu, tetapi juga tetap relevan dan dapat beroperasi di masa depan ketika lingkungan eksternal berubah

Karakteristik Pemeliharaan Adaptif

- 1) **Penyesuaian dengan Perubahan Eksternal** Pemeliharaan adaptif dilakukan ketika ada perubahan di lingkungan operasional perangkat lunak yang memerlukan penyesuaian pada perangkat lunak tersebut. Misalnya, jika perangkat lunak berjalan di atas sistem operasi tertentu, tetapi sistem operasi tersebut diperbarui atau diganti dengan versi baru, perangkat lunak perlu disesuaikan untuk tetap berjalan dengan lancar
- 2) **Menjamin Kompatibilitas** Tujuan utama dari pemeliharaan adaptif adalah untuk memastikan perangkat lunak tetap kompatibel dengan perubahan pada infrastruktur teknologi. Ini mencakup pembaruan pada perangkat keras, perangkat lunak lain yang digunakan bersama perangkat lunak tersebut, atau perubahan standar industri yang berpengaruh pada operasi perangkat lunak
- 3) **Menangani Perubahan yang Tidak Dapat Diharapkan** Pemeliharaan adaptif juga dapat mencakup penyesuaian terhadap perubahan yang tidak dapat diprediksi atau diantisipasi sejak awal pengembangan perangkat lunak. Ini

berarti bahwa perangkat lunak harus dapat beradaptasi dengan perubahan kebutuhan atau teknologi yang baru muncul selama masa pakainya

Contoh: Jika versi baru dari versi php, perangkat lunak yang berjalan di atasnya mungkin perlu diadaptasi untuk memastikan kompatibilitasnya.

4) PEMELIHARAAN PERFEKTIF

Pemeliharaan perfektif adalah jenis pemeliharaan perangkat lunak yang berfokus pada peningkatan kinerja, fungsionalitas, dan kualitas perangkat lunak setelah perangkat lunak tersebut digunakan. Tujuan utama dari pemeliharaan perfektif adalah untuk memperbaiki dan meningkatkan perangkat lunak agar lebih sesuai dengan kebutuhan pengguna atau lebih efisien dalam operasionalnya. Pemeliharaan ini biasanya dilakukan dengan menambahkan fitur baru, memperbaiki antarmuka pengguna, mengoptimalkan kinerja, atau memperbaiki aspek-aspek teknis lain dari perangkat lunak yang sudah ada.

Pemeliharaan perfektif berbeda dengan pemeliharaan korektif, yang lebih berfokus pada perbaikan kesalahan atau bug dalam perangkat lunak. Pemeliharaan perfektif bertujuan untuk menjadikan perangkat lunak lebih baik dari segi kualitas dan fungsionalitas, berdasarkan umpan balik yang diterima setelah perangkat lunak digunakan di dunia nyata.

Karakteristik Pemeliharaan Perfektif

- 1) **Peningkatan Fungsionalitas:** Pemeliharaan perfektif dapat melibatkan penambahan fitur baru atau peningkatan fitur yang sudah ada agar perangkat lunak lebih sesuai dengan kebutuhan pengguna atau perkembangan pasar. Peningkatan ini dapat berupa fungsionalitas baru, integrasi dengan sistem lain, atau pembaruan untuk memenuhi tren teknologi terbaru
- 2) **Pengoptimalan Kinerja:** Salah satu aspek penting dari pemeliharaan perfektif adalah pengoptimalan perangkat lunak untuk meningkatkan kinerjanya. Hal ini bisa mencakup peningkatan waktu respons, pengurangan penggunaan sumber daya (misalnya, memori dan CPU), dan peningkatan skalabilitas perangkat lunak untuk menangani lebih banyak pengguna atau lebih banyak data
- 3) **Peningkatan Pengalaman Pengguna (User Experience/UX):** Pemeliharaan perfektif seringkali melibatkan perbaikan pada antarmuka pengguna (UI) untuk memastikan bahwa perangkat lunak lebih mudah digunakan dan lebih intuitif. Ini bisa meliputi desain ulang layar, penataan ulang elemen antarmuka, atau penambahan fitur untuk meningkatkan kenyamanan pengguna
- 4) **Perbaikan Kualitas Perangkat Lunak:** Pemeliharaan perfektif juga melibatkan usaha untuk meningkatkan kualitas perangkat lunak secara keseluruhan, baik dari segi keandalan, keamanan, maupun kestabilan. Ini termasuk perbaikan pada kode

sumber, penghapusan kode yang tidak terpakai, serta pengurangan kerentanannya terhadap ancaman keamanan.

Contoh: Menambahkan fitur Grade pada penilaian SPADA dalam fitur penilaian.

5) PEMELIHARAAN PREVENTIF

Pemeliharaan preventif adalah jenis pemeliharaan perangkat lunak yang bertujuan untuk mencegah masalah atau kerusakan di masa depan dengan melakukan perawatan secara proaktif. Pemeliharaan ini berfokus pada pengidentifikasian potensi masalah sebelum mereka muncul, memperbaiki bagian-bagian perangkat lunak yang dapat menimbulkan kesalahan atau kerusakan, dan memastikan perangkat lunak tetap dapat berjalan dengan baik dalam jangka panjang. Tujuan utama dari pemeliharaan preventif adalah untuk meningkatkan keandalan dan kestabilan perangkat lunak dengan mengurangi kemungkinan terjadinya kegagalan yang dapat mengganggu operasional perangkat lunak.

Pemeliharaan preventif dilakukan dengan cara mengidentifikasi dan mengatasi potensi kerentanannya, memperbarui komponen perangkat lunak yang sudah usang, serta memperbaiki bagian-bagian yang mungkin tidak segera menunjukkan gejala masalah namun dapat menyebabkan masalah serius di masa depan.

Karakteristik Pemeliharaan Preventif

1. **Perawatan Proaktif** Pemeliharaan preventif dilakukan dengan tujuan mencegah masalah yang mungkin muncul. Hal ini berbeda dengan pemeliharaan korektif yang bertindak setelah masalah ditemukan. Pemeliharaan preventif menargetkan potensi masalah yang dapat terjadi dan melakukan tindakan untuk menghindari masalah tersebut
2. **Meningkatkan Keandalan dan Kestabilan** Salah satu tujuan utama pemeliharaan preventif adalah meningkatkan keandalan perangkat lunak agar sistem dapat berjalan stabil tanpa gangguan, bahkan dalam jangka waktu yang lama. Perangkat lunak yang memiliki keandalan tinggi akan lebih jarang mengalami kerusakan yang dapat mengganggu fungsionalitasnya
3. **Mengurangi Risiko Kegagalan di Masa Depan** Pemeliharaan preventif bertujuan untuk mengurangi kemungkinan terjadinya kesalahan atau kegagalan di masa depan dengan memperbaiki kerentanannya sejak dini. Ini bisa mencakup memperbaiki kode yang berisiko tinggi atau menghilangkan bagian perangkat lunak yang sudah usang dan rentan.
4. **Pembaruan dan Penggantian Komponen Usang** Dalam pemeliharaan preventif, salah satu hal yang sering dilakukan adalah pembaruan atau penggantian komponen perangkat lunak yang sudah usang atau tidak lagi mendukung perkembangan teknologi terkini. Ini mencakup memperbarui pustaka, framework,

atau sistem operasi yang digunakan oleh perangkat lunak

Peningkatan Pengalaman Pengguna (User Experience/UX): Pemeliharaan perfektif seringkali melibatkan perbaikan pada antarmuka pengguna (UI) untuk memastikan bahwa perangkat lunak lebih mudah digunakan dan lebih intuitif. Ini bisa meliputi desain ulang layar, penataan ulang elemen antarmuka, atau penambahan fitur untuk meningkatkan kenyamanan pengguna

Contoh: Mengupdate Aplikasi SPADA untuk melindungi dari potensi ancaman keamanan judi slot.

3. DAFTAR REFERENSI

1. Sommerville, I. (2011). *Software Engineering* (9th ed.). Addison-Wesley.
2. Pressman, R. S. (2014). *Software Engineering: A Practitioner's Approach* (8th ed.). McGraw-Hill.
3. Fowler, M. (2018). *Refactoring: Improving the Design of Existing Code*. Addison-Wesley
4. Lacity, M. C., & Willcocks, L. P. (2017). *Robotic Process Automation: The Next Transformation in Business Services*. Palgrave Macmillan.

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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> a. Konsep dalam User Interface b. Prinsip Desain Antarmuka (user experience, user guidance, user diversity) 	Idem
10	<ul style="list-style-type: none"> a. Proses testing: (black box testing, white box testing) b. Integration testing dan user testing c. Faults, Error dan Failures 	Idem
11	Review Teknik Pengujian Perangkat Lunak dari proses testing	Idem
12	Pengujian unit, Pengujian integrasi, Pengujian sistem, Pengujian Penerimaan	Idem
13	<ul style="list-style-type: none"> a. Quality assurance pada perangkat lunak b. Keamanan data akses 	Idem
14	Definisi pemeliharaan perangkat lunak dan Konsep Pemeliharaan Perangkat lunak	Idem
15	Teknik pemeliharaan perangkat lunak (Pemeliharaan korektif, pemeliharaan adaptif, pemeliharaan perfektif, pemeliharaan preventif)	Idem
16	UAS	