

**QUALITY ASSURANCE 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**

Quality Assurance (QA) dalam pengembangan perangkat lunak adalah proses sistematis yang memastikan bahwa perangkat lunak yang dikembangkan sesuai dengan standar kualitas yang ditetapkan, bebas dari failure, dan mampu memenuhi kebutuhan pengguna akhir. QA berfokus pada pencegahan masalah selama seluruh siklus hidup pengembangan perangkat lunak (Software Development Life Cycle, SDLC) dan tidak hanya sekedar menemukan kesalahan di akhir proses.

1. Tujuan QA dalam Pengembangan Perangkat Lunak

Beberapa tujuan utama dalam QA adalah sebagai berikut:

1) Mencegah Cacat (Defect Prevention)

QA bertujuan untuk mengidentifikasi dan mencegah cacat sejak awal proses pengembangan melalui perencanaan dan kontrol kualitas. Dengan demikian, QA meminimalkan kesalahan yang mungkin terjadi pada tahap-tahap berikutnya, mengurangi kebutuhan untuk perbaikan di akhir siklus pengembangan.

2) Memastikan Kualitas Produk (Product Quality Assurance)

QA memastikan bahwa perangkat lunak memenuhi standar kualitas yang telah ditetapkan, mencakup aspek fungsional, keandalan, keamanan, kinerja, dan kegunaan. Dengan ini, QA membantu menghasilkan perangkat lunak yang andal dan sesuai kebutuhan pengguna

3) Memenuhi Persyaratan Pengguna (User Requirement Fulfillment)

QA memastikan bahwa produk akhir sesuai dengan spesifikasi dan memenuhi kebutuhan serta ekspektasi pengguna akhir. Hal ini meningkatkan kepuasan pengguna dan memastikan perangkat lunak memiliki nilai guna yang tinggi.

4) Meningkatkan Efisiensi Pengembangan (Development Efficiency)

QA berfokus pada peningkatan proses pengembangan dengan menerapkan praktik terbaik, alat, dan teknik yang mendukung efektivitas kerja tim. Ini memungkinkan tim pengembang bekerja lebih cepat dan efisien, dengan meminimalkan waktu yang terbuang akibat kesalahan.

5) Meningkatkan Pemeliharaan dan Skalabilitas (Maintainability and Scalability)

QA bertujuan memastikan perangkat lunak mudah dipelihara dan ditingkatkan di masa depan. Dengan menerapkan standar pengkodean yang baik dan dokumentasi yang lengkap, QA memastikan bahwa perangkat lunak dapat dikembangkan lebih lanjut tanpa kesulitan besar.

6) Mengurangi Risiko dan Biaya (Risk and Cost Reduction)

Dengan menemukan dan memperbaiki cacat lebih awal, QA mengurangi biaya yang terkait dengan perbaikan masalah di tahap akhir atau setelah rilis produk. QA juga mengurangi risiko yang dapat mempengaruhi bisnis, terutama yang terkait dengan kepatuhan, keamanan, dan kegagalan sistem.

7) Meningkatkan Kepuasan Pengguna dan Reputasi Produk (User Satisfaction and Product Reputation)

QA memastikan produk berkualitas tinggi yang memberikan pengalaman positif bagi pengguna, sehingga meningkatkan kepuasan mereka. Produk berkualitas juga membantu menjaga reputasi perusahaan dan meningkatkan daya saing di pasar.

2. Aspek Quality Assurance (QA) dalam pengembangan perangkat lunak

Berikut adalah aspek-aspek **Quality Assurance (QA)** dalam pengembangan perangkat lunak:

1) Fungsi dan Kesesuaian (Functionality and Compliance)

QA memastikan perangkat lunak berfungsi sesuai dengan spesifikasi dan memenuhi standar atau regulasi industri yang berlaku. Fungsi perangkat lunak diuji untuk memverifikasi bahwa setiap fitur bekerja sebagaimana mestinya dan sesuai dengan keinginan user [1].

2) Keandalan (Reliability)

Aspek keandalan dalam QA mengevaluasi kemampuan perangkat lunak untuk berjalan tanpa kesalahan dalam waktu tertentu dan pada kondisi tertentu. Pengujian keandalan memastikan stabilitas perangkat lunak dalam situasi penggunaan yang berbeda[6].

3) Kinerja (Performance)

Pengujian kinerja (performance testing) mengukur kecepatan, respons, dan efisiensi perangkat lunak dalam menangani beban kerja. Teknik ini termasuk load testing dan stress testing untuk menilai bagaimana perangkat lunak berfungsi di bawah beban tinggi [3].

4) Kemudahan Penggunaan (Usability)

Kemudahan penggunaan adalah aspek yang memastikan perangkat lunak mudah dipelajari dan digunakan oleh pengguna. QA pada aspek ini mengevaluasi

antarmuka pengguna dan interaksi untuk memastikan pengalaman yang optimal bagi pengguna[5].

5) Keamanan (Security)

Pengujian keamanan bertujuan untuk mengidentifikasi dan mengeliminasi kerentanan yang dapat disalahgunakan oleh pihak tidak berwenang. Keamanan perangkat lunak mencakup perlindungan data dan autentikasi untuk menjaga kerahasiaan dan integritas informasi[4].

6) Portabilitas dan Kompatibilitas (Portability and Compatibility)

QA mengevaluasi kemampuan perangkat lunak untuk beroperasi pada berbagai platform dan sistem. Pengujian kompatibilitas memastikan perangkat lunak berfungsi dengan baik di berbagai lingkungan seperti sistem operasi dan perangkat keras yang berbeda[6].

7) Pemeliharaan (Maintainability)

Pemeliharaan perangkat lunak meliputi kualitas kode, dokumentasi, dan struktur yang memungkinkan perangkat lunak mudah diperbaiki, diperbarui, atau dikembangkan lebih tinggi lagi [6].

8) Pengujian Fungsional dan Non-Fungsional (Functional and Non-Functional Testing)

Pengujian fungsional mengevaluasi aspek dasar perangkat lunak, sedangkan pengujian non-fungsional mencakup aspek-aspek di luar fungsi dasar, seperti kinerja, keamanan, dan kegunaan [1].

3. DAFTAR REFERENSI

1. ISO/IEC 9126: Quality characteristics and guidelines for quality assurance.
2. IEEE Standard 1633 - Reliability Program Standard.
3. Performance Testing Guidance for Web Applications by Microsoft.
4. OWASP (Open Web Application Security Project) standards.
5. ISO 9241-11: Ergonomics of human-system interaction – Usability.
6. IEEE Standard for Software Compatibility Testing.
7. IEEE Standard 1061 - Software Quality Metrics Methodology.
8. SDLC Quality Assurance Model – Process Optimization.
9. Functional vs. Non-Functional Requirements in Quality Assurance.
10. ISO/IEC/IEEE 29119: *Software Testing* - Panduan Standar Internasional untuk praktik terbaik dalam pengujian perangkat lunak

11. IEEE Std. 829-2008: *Standard for Software and System Test Documentation* -

Panduan standar untuk dokumentasi dalam pengujian perangkat lunak dan sistem

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
8	UTS	
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	Pengujian unit, Pengujian integrasi, Pengujian sistem, Pengujian Penerimaan	Idem
13	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	