

Tugas 2

(latihan soal2 midtest)

1. Secara kategoris, ada tiga buah jenis kebutuhan perangkat lunak. Sebutkan dan Jelaskan!
2. Secara teknis pelaksanaan pekerjaan analisis kebutuhan perangkat lunak pada dasarnya terdiri dari urutan aktivitas. Sebutkan dan Jelaskan!
3. Sebutkan dan jelaskan beberapa metode analisis yang populer saat ini!
4. Keberhasilan pengembangan perangkat lunak bisa dilihat dari 10 aspek atau titik pandang. Sebutkan!
5. Sebutkan dan jelaskan perangkat pemodelan pada analisis terstruktur !
6. Sebutkan dan jelaskan empat elemen yang membentuk suatu Data Flow Diagram !
7. Jelaskan dua versi analisis terstruktur, Analisis Terstruktur Klasik (Classical Structured Analysis) dan Analisis Terstruktur Modern (Modern Structured Analysis).
8. Sebutkan dan jelaskan karakteristik atau sifat-sifat yang dimiliki sebuah sistem berorientasi objek !
9. Sebutkan dan jelaskan keuntungan menggunakan metodologi berorientasi objek!
10. Sebutkan beberapa diagram yang digunakan pada [UML](#) (min.5) !

Jawaban

1. 3 jenis kebutuhan perangkat lunak
 - kebutuhan fungsional : menggambarkan apa yang harus dilakukan oleh perangkat lunak, mencakup fungsi, fitur, dan layanan yang harus ada
 - kebutuhan Non-Fungsional : mengacu pada kualitas perangkat lunak seperti performa, keandalan, skalabilitas, dan keamanan.
 - kebutuhan Domain : Merupakan kebutuhan spesifik untuk domain tertentu, seperti aturan bisnis atau regulasi yang harus diikuti oleh perangkat lunak
2.
 - Pengumpulan : mengumpulkan kebutuhan dari pengguna, stakeholder, atau dokumen terkait.
 - Analisis : mengorganisasi dan menganalisis kebutuhan untuk memastikan kelengkapan dan konsistensinya
 - Spesifikasi : mendokumentasikan kebutuhan secara jelas dalam dokumen spesifikasi
 - validasi : memastikan bahwa kebutuhan mencerminkan apa yang diinginkan oleh stakeholder
3. Analisis SWOT: Untuk mengidentifikasi Strengths, Weaknesses, Opportunities, dan Threats.
Use Case Analysis: Mengidentifikasi interaksi antara pengguna dan sistem melalui diagram use case.
Requirement Prioritization: Menentukan prioritas kebutuhan berdasarkan nilai bisnis dan kompleksitas.
Prototyping: Membuat prototipe untuk mengklarifikasi kebutuhan pengguna.

4. - Kesesuaian dengan kebutuhan pengguna.
 - Kemudahan penggunaan.
 - Performa yang memadai.
 - Skalabilitas untuk masa depan.
 - Keandalan (reliability).
 - Keamanan (security).
 - Pemeliharaan yang mudah.
 - Dokumentasi yang lengkap.
 - Efisiensi dalam penggunaan sumber daya.
 - Dukungan teknis yang memadai.
5. - Data Flow Diagram (DFD): Untuk memodelkan aliran data dalam sistem.
 - Entity-Relationship Diagram (ERD): Untuk memodelkan hubungan antara entitas dalam basis data.
 - State Diagram: Untuk memodelkan perubahan keadaan suatu objek.
 - Context Diagram: Untuk menunjukkan sistem secara keseluruhan dan hubungannya dengan entitas eksternal.
6. - External Entity: Representasi entitas eksternal yang berinteraksi dengan sistem (contoh: pengguna).
 - Process: Menggambarkan transformasi data dari input menjadi output.
 - Data Store: Tempat penyimpanan data dalam sistem (contoh: database).
 - Data Flow: Menggambarkan pergerakan data antar elemen.
7. - Classical Structured Analysis: Fokus pada penggunaan DFD, ERD, dan diagram hierarki untuk merancang sistem berdasarkan alur data.
 - Modern Structured Analysis: Memperluas metode klasik dengan menambahkan pendekatan berbasis objek, prototyping, dan teknik berbasis user-centric.
8. - Enkapsulasi: Menyembunyikan detail implementasi dalam objek.
 - Abstraksi: Fokus pada aspek penting tanpa memperhatikan detail.
 - Pewarisan (Inheritance): Pewarisan atribut dan metode dari kelas induk.
 - Polimorfisme: Kemampuan satu interface untuk digunakan oleh beberapa implementasi.
 - Reusability: Kode dapat digunakan kembali.
9. - Pemeliharaan yang lebih mudah: Struktur modular memudahkan perbaikan.
 - Skalabilitas: Mudah untuk menambah fitur baru.
 - Reusability: Komponen dapat digunakan ulang.
 - Efisiensi dalam pengembangan: Pendekatan yang terorganisir mempercepat proses.
 - Kolaborasi tim: Representasi visual memudahkan komunikasi antar anggota tim.
10. - Use Case Diagram: Menggambarkan interaksi antara pengguna dan sistem.
 - Class Diagram: Memodelkan struktur kelas dan hubungan antar kelas.
 - Sequence Diagram: Menggambarkan urutan interaksi antara objek.
 - Activity Diagram: Menunjukkan alur kerja dalam sistem.
 - State Diagram: Menggambarkan perubahan status objek dalam sistem.