

NO:

DATE:

Nama: Stevanus Andika Galih Setiawan

NIM: 2023110008

Prodi: Rekayasa Perangkat Lunak

Kelas: RK231

Semester: 1 (satu)

Tanggal: 18 November 2023

Matakuliah: Dasar RPL

1. B. Disiplin ilmu teknik yang berkaitan dengan semua aspek produksi perangkat lunak dari tahap awal spesifikasi sistem hingga pemeliharaan sistem setelah mulai digunakan
2. B. failure
3. B. Perangkat lunak customize
4. C. Stand alone applications
5. B. Maintainability
6. D. Efficiency
7. C. Perangkat Lunak gratis
8. B. Perangkat Lunak customize
9. C. Perangkat Lunak Malware
10. D. Software Configuration



11. D. Persona
12. A. Proses Perangkat Lunak
13. D. Model Waterfall / linier Sequential Model
14. B. Implementasi
15. D. Operation and maintenance
16. B. Masalah Pencapaian tujuan
17. c Pemecahan Masalah Rekayasa Perangkat Lunak
- 18 D. Planning
19. C. statement of work
20. B. Project Schedule
21. D. Prototipe yg meliputi perawatan file dan proses proses transaksi.
22. D. sistem Prototipe
23. C. Input / output Prototipe
24. D. Activity diagram
- 25 D. Kelas. diagram

Soal Essay

1. Suatu disiplin ilmu dalam rekayasa Perangkat Lunak yang bertujuan untuk mengelola perubahan Sistem Perangkat Lunak. Tujuannya adalah untuk Menjabarkan proses - proses yg diperlukan untuk mengontrol dan memantau perubahan yg terjadi pada Perangkat Lunak.
2. - Pengujian pada sistem yg akan dibangun: uml digunakan agar Mendukung Model Pengujian Perangkat Lunak. Developer dapat Merencanakan Pengujian yg lebih baik dengan adanya model UML.



NO:

DATE:

- Analisis dan Perancangan yg lebih baik.
dengan adanya UML, maka memungkinkan permodelan dan terstruktur dari berbagai aspek sisi sistem.
(terdiri dari use case, kelas, activity, dan lainnya).
sehingga membantu dalam analisis dan perancangan sistem dan cara sistematis dan rinci berurutan.
- Rekayasa ulang dan pemeliharaan:
Dengan adanya UML, tim developer dapat lebih mudah memahami struktur dan fungsi sistem dalam urutan.
memudahkan proses rekayasa ulang (refactoring) dan pemeliharaan perangkat lunak. dengan begitu, perubahan dapat direncanakan dan diimplementasikan dengan baik.
- Dokumentasi yang terstruktur
Dengan adanya UML, memungkinkan pembuatan dokumentasi yang terstruktur. sehingga memudahkan dengan adanya dokumentasi yang lengkap dan lebih mudah memahami desain sistem.
- untuk Pengembangan berbasis komponen.
Dengan adanya UML memudahkan developer untuk merencanakan dan mengintegrasikan komponen perangkat lunak secara efisien.
- Komunikasi yang efektif
UML menyediakan visual yang mudah dimengerti, baik pihak teknis maupun non teknis, sehingga memungkinkan terjadinya komunikasi yang efektif.



3 - Diagram class / Diagram kelas

adalah suatu penggambaran struktur sistem dari segi pendefinisian kelas-kelas yang digunakan untuk membangun sistem.

Tujuan: Diagram class dibuat agar pembuat program dapat membuat kelas-kelas sesuai rancangan di dm diagram class agar antara dokumentasi perancangan dan perangkat lunak sinkron.

- Diagram use case

bertujuan / berfungsi untuk menunjukkan interaksi antara sistem dan berbagai aktor / entitas (dalam hal ini user) melalui skenario. Digunakan untuk menggambarkan fungsionalitas / fungsi yang diharapkan pengguna dari suatu sistem.

- Diagram Deployment

bertujuan untuk menunjukkan cara sistem diimplementasikan pada lingkungan fisik, termasuk komputer dan perangkat keras jaringan, digunakan untuk memodelkan konfigurasi perangkat keras dan perangkat lunak dalam sistem.

- Diagram Component

berfungsi untuk memodelkan komponen-komponen fisik / logis dan hubungannya, digunakan memvisualisasikan bagaimana komponen saling berinteraksi.

- Diagram State Machine

berfungsi untuk objek dalam sistem berubah dari satu kondisi ke kondisi lainnya dlm respons suatu peristiwa / event tertentu.



NO:

DATE:

4. Concept diagram

- Membantu dalam memahami konsep/ide dasar suatu Sistem/Proyek. digunakan pada tahap awal perencanaan untuk penggambaran umum tentang apa yg ingin dicapai dengan Proyek tersebut.

- Digunakan untuk merinci konsep yang mungkin kompleks. Diagram ini membantu menguraikan elemen inti dan hubungan dan cara yg simple dan mudah dimengerti

- Digunakan untuk menyampaikan dgn jelas tujuan dan sasaran kepada semua pihak.

Deployment

- Digunakan untuk menadatkan dan menampilkan fisik dari komponen-komponen Perangkat lunak ke dlm lingkungan fisik (Perangkat keras, jaringan, server).

- Memberikan gambaran yang jelas spesifikasi Perangkat yang digunakan

- membantu Pemanejaan dan manajemen Sumber daya fisik.



NO :

DATE :

5 -



Actor: berfungsi menggambarkan orang / sistem / entitas yg terlibat

- -



Asosiasi: berfungsi menggambarkan bagaimana actor berkomunikasi dengan use case

- - - - ->

Generalisasi: menggambarkan proses umum yg terjadi antara use case / user / actor (dari child ke parent).

- - - - -> ^{Include}

Include: menggambarkan hubungan antara dua use case untuk menunjukkan perilaku use case yg dimasukkan dlm perilaku Base use case (Sub use case)

use case

Use case: berfungsi untuk menggambarkan aktivitas / proses / gambaran sistem menggunakan kata kerja

- - - - -> ^{Extend}

Extend: berfungsi sebagai Peristiwa dari use case lain (mengarah ke base use case)

