

LAPORAN PRAKTIKUM PENGENALAN REKAYASA PERANGKAT LUNAK

INTRODUCTION TO SOFTWARE ENGINEERING [2]



13323041 – Enrogel Jeremi Sibarani
13323033 – Agus Pranata Marpaung
13323048 – Maylani E. Sagala
DIII Teknologi Komputer

INSTITUT TEKNOLOGI DEL
FAKULTAS VOKASI

Judul Praktikum


Minggu/Sesi	:	III/2
Kode Mata Kuliah	:	1141105
Nama Mata Kuliah	:	Pengenalan Rekayasa Perangkat Lunak
Setoran	:	Buat lembar kerja tugas kelompok di word kemudian Save As .pdf dan beri nama “w03s02_SoftwareProcess_D3TK_NIM 1 (3 digit terakhir)_NIM 2 (3 digit terakhir)_NIM 3 (3 digit terakhir).pdf”
Batas Waktu Setoran	:	13 September 2023 jam 12:00
Tujuan	:	Mahasiswa dapat memahami apa yang dimaksud dengan Software Engineering.

Petunjuk

Pengenalan Rekayasa Perangkat Lunak

Soal

1. Bentuklah sebuah kelompok yang terdiri dari 3 orang mahasiswa untuk mendiskusikan jawaban dari pertanyaan di bawah ini:
 - a) Jelaskan definisi *software engineering*
 - b) Jelaskan masalah-masalah yang terjadi pada *software engineering*
 - c) Jelaskan layer dari *Software Engineering*
 - d) Jelaskan apa yang dimaksud dengan *software process*
 - e) Pahami Kembali perbedaan *activity*, *action* dan *tasks* dengan menerjemahkan, memahami, menjelaskan dengan contohnya berikut



Task Set

INFO

A task set defines the actual work to be done to accomplish the objectives of a software engineering action. For example, *elicitation* (more commonly called "requirements gathering") is an important software engineering action that occurs during the **communication** activity. The goal of requirements gathering is to understand what various stakeholders want from the software that is to be built.

For a small, relatively simple project, the task set for requirements gathering might look like this:

1. Make a list of stakeholders for the project.
2. Invite all stakeholders to an informal meeting.
3. Ask each stakeholder to make a list of features and functions required.
4. Discuss requirements and build a final list.
5. Prioritize requirements.
6. Note areas of uncertainty.

For a larger, more complex software project, a different task set would be required. It might encompass the following work tasks:

1. Make a list of stakeholders for the project.
2. Interview each stakeholder separately to determine overall wants and needs.
3. Build a preliminary list of functions and features based on stakeholder input.
4. Schedule a series of facilitated application specification meetings.
5. Conduct meetings.
6. Produce informal user scenarios as part of each meeting.
7. Refine user scenarios based on stakeholder feedback.
8. Build a revised list of stakeholder requirements.
9. Use quality function deployment techniques to prioritize requirements.
10. Package requirements so that they can be delivered incrementally.
11. Note constraints and restrictions that will be placed on the system.
12. Discuss methods for validating the system.

Both of these task sets achieve "requirements gathering," but they are quite different in their depth and formality. The software team chooses the task set that will allow it to achieve the goal of each action and still maintain quality and agility.

Jawab

1. **Software Engineering** adalah disiplin ilmiah dan praktis yang berkaitan dengan perancangan, pengembangan, pengujian, pemeliharaan, dan manajemen perangkat lunak. Ini melibatkan penggunaan pendekatan sistematis, metode, dan alat untuk menciptakan perangkat lunak berkualitas tinggi yang memenuhi kebutuhan pengguna dan mematuhi standar industri.

2. **Masalah - masalah yang terjadi pada Software Engineering**

Dalam Software Engineering, ada beberapa masalah yang sering muncul, termasuk:

- a) Penundaan Proyek: Kesulitan dalam memprediksi waktu yang diperlukan untuk menyelesaikan proyek perangkat lunak.
- b) Perubahan Persyaratan: Persyaratan pengguna yang berubah atau tidak jelas selama siklus pengembangan.
- c) Bug dan Kerentanan Keamanan: Kesalahan perangkat lunak dan masalah keamanan yang dapat merusak fungsi atau mengancam keamanan data.
- d) Kurangnya Komunikasi Tim: Kurangnya komunikasi yang efektif dalam tim pengembangan perangkat lunak dapat menyebabkan kesalahpahaman dan konflik.
- e) Manajemen Proyek yang Buruk: Kesalahan dalam perencanaan, pengelolaan sumber daya, dan alokasi anggaran proyek.

3. **Layer dari software engineering**

- a) Fokus kualitas (*a quality focus*)
Lapisan pada bagian ini menjadi hal pertama yang harus dibangun ketika membangun aplikasi. Kualitas focus dapat membantu programmer untuk mengetahui level aplikasi yang akan dibangun dan siapa sasaran dari aplikasi tersebut.
- b) Proses (*process*)
Lapisan pada bagian kedua ini mendefinisikan kerangka kerja yang harus didirikan untuk pengiriman efektif pada teknologi rekayasa perangkat lunak.
- c) Metode (*method*)
Lapisan pada bagian ketiga ini memiliki informasi tentang semua tugas yang mencakup komunikasi, analisis kebutuhan, pemodelan desain, konstruksi program, pengujian, dan dukungan.
- d) Alat (*tools*)
Lapisan pada bagian keempat ini menyediakan sistem operasi sendiri untuk proses dan metode. Alat – alat ini terintegrasi, yang berarti informasi yang dibuat oleh satu alat dapat digunakan oleh alat lain.

4. **Apa yang dimaksud dengan Software Process?**

Software Process (proses perangkat lunak) adalah serangkaian langkah, metode, dan praktik yang digunakan dalam pengembangan perangkat lunak. Ini mencakup semua tahapan dari perencanaan hingga pemeliharaan. Software Process membantu mengatur proyek perangkat lunak, meningkatkan kualitas produk, dan memastikan bahwa proyek berjalan sesuai jadwal dan anggaran.

5. Perbedaan antara Activity, Action, dan Tasks

- a) Activity (Aktivitas): Ini adalah proses atau kegiatan umum dalam siklus pengembangan perangkat lunak, seperti analisis, desain, atau pengujian.
- b) Action (Aksi): Ini adalah langkah konkret dalam suatu aktivitas. Misalnya, dalam aktivitas "pengujian," sebuah action dapat berupa menjalankan tes pengujian tertentu.
- c) Task (Tugas): Tugas adalah unit terkecil dalam pengembangan perangkat lunak. Ini adalah pekerjaan spesifik yang harus dilakukan, seperti menulis kode untuk fitur tertentu.

Contoh:

Dalam aktivitas "Pengembangan Fitur X," action mungkin termasuk merancang antarmuka pengguna dan menulis kode sumber. Sementara itu, tugas dalam action "Menulis Kode Sumber" dapat mencakup menulis fungsi A, fungsi B, dan menguji fungsi tersebut.