

DAFTAR ISI

| | Halaman |
|---|---------|
| I. PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 1 |
| 1.3 Tujuan | 1 |
| II. PEMBAHASAN | 2 |
| 2.1 Pengertian Software | 2 |
| 2.2 Pengertian Software Engineering | 3 |
| a. Metode Software Engineering | 4 |
| b. Peralatan Software Engineering | 10 |
| c. Prosedur Software Engineering | 11 |
| III. KESIMPULAN | 12 |
| DAFTAR PUSTAKA | 13 |

SOFTWARE ENGINEERING

I PENDAHULUAN

1.1 Latar Belakang

Munculnya software engineering terjadi ketika adanya krisis software di era tahun 1960-an. Krisis tersebut akibat dari lahirnya komputer generasi ke III yang ditandai dengan penggunaan IC (*Integrated Circuit*). Kemampuan hardware yang meningkat, membuat adanya kebutuhan untuk memproduksi software yang lebih baik. Akibatnya software yang dihasilkan menjadi beberapa kali lebih besar dan kompleks. Pendekatan informal yang digunakan dalam pengembangan perangkat lunak pada saat itu, menjadi tidak cukup efektif (secara biaya, waktu dan kualitas). Biaya hardware mulai jatuh dan biaya perangkat lunak menjadi naik cepat. Oleh karena itu muncul pemikiran untuk menggunakan pendekatan yang lebih efektif, standard dan terukur dalam mengembangkan perangkat lunak. Krisis software adalah sekumpulan masalah yang ditemukan dalam pengembangan software komputer. Masalahnya tidak hanya terbatas pada software yang tidak berfungsi sebagaimana mestinya, tetapi krisis software ini terdiri dari masalah yang berhubungan dengan :

- a. Bagaimana mengembangkan software.
- b. Bagaimana memelihara software yang ada, yang berkembang dalam jumlah besar.
- c. Bagaimana mengimbangi permintaan software yang makin besar.

1.2 Rumusan Masalah

- Pengertian Software
- Pengertian Software Engineering
- Fungsi dan pengembangan

1.3 Tujuan

- Memahami pengertian dari software engineering
- Memahami elemen-elemen dalam software engineering
- Memahami fase-fase atau langkah-langkah yang ada dalam software engineering

II. PEMBAHASAN

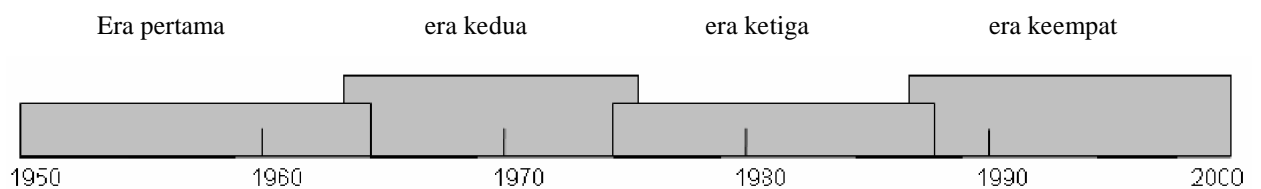
2.1 Pengertian Software

Perangkat lunak atau software adalah definisi dan organisasi dari sekumpulan perintah dan fungsi yang dienkapsulasi dalam bentuk yang dapat dieksekusi oleh komputer.

Secara umum karakteristik software adalah sebagai berikut:

1. Software merupakan elemen sistem logika dan bukan elemen sistem fisik seperti hardware
2. Elemen itu tidak aus, tetapi bisa rusak.
3. Elemen software itu direkayasa atau dikembangkan dan bukan dibuat di pabrik seperti hardware.

Evolusi perangkat lunak:



- Era pertama :

- Batch Orientation

Suatu orientasi dimana proses dilakukan setelah data dikumpulkan dalam satuan waktu tertentu, atau proses dilakukan setelah data terkumpul. Kebalikan dari batch adalah online atau Interactive Process. Keuntungan dari Interactive adalah mendapatkan data yang selalu up to date.

- Limited distribution

Suatu penyebaran software yang terbatas pada perusahaan-perusahaan tertentu.

- Custom software

Software yang dikembangkan berdasarkan perusahaan-perusahaan tertentu.

- Era kedua :

- Multi user

Suatu sistem dimana satu komputer digunakan oleh beberapa user pada saat yang sama.

- Real Time

Suatu sistem yang dapat mengumpulkan, menganalisa dan mentransformasikan data dari berbagai sumber, mengontrol proses dan menghasilkan output dalam mili detik.

- Database

Perkembangan yang pesat dari alat penyimpanan data yang online menyebabkan muncul generasi pertama DBMS.

- Product Software

Software dikembangkan untuk dijual kepada masyarakat luas.

- Era ketiga :

- Distributed system

Suatu sistem yang tidak hanya dipusatkan pada komputer induk, daerah atau bidang lainnya yang juga memiliki komputer yang ukurannya lebih kecil dari komputer induk.

- Embedded Intelligence

Suatu product yang diberi tambahan “Intelligence” dan biasanya ditambahkan mikroprocessor yang mutakhir. Contohnya adalah automobil, robot, peralatan diagnostic serum darah.

- Low Cost Hardware

Harga hardware yang semakin rendah, ini dimungkinkan karena munculnya Personal Computer (PC).

- Consumer Impact

Adanya perkembangan komputer yang murah menyebabkan banyaknya software yang dikembangkan, software ini memberi dampak yang besar terhadap masyarakat.

- Era keempat :

- Expert system

Suatu penerapan A.I. (Artificial Intelligence) pada bidang-bidang tertentu, misalnya bidang kedokteran, komunikasi, dll.

- AI Machine

Suatu mesin yang dapat meniru kerja dari sebagian otak manusia. Misalnya mesin robot, komputer catur.

- Parallel Architecture

Arsitektur komputer yang memungkinkan proses kerja LAN paralel, yang dimungkinkan adanya prosesor berbeda dalam satu computer.

2.2 Pengertian Software Engineering

Software engineering adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal requirement capturing (analisa kebutuhan pengguna), specification (menentukan spesifikasi dari kebutuhan pengguna), design, coding, testing sampai pemeliharaan sistem setelah digunakan.

Definisi Software Engineering menurut IEEE (Institute of Electrical and Electronics Engineers) pada proyek SWEBOK (Software Engineering Body of Knowledge) adalah aplikasi sistematis, disiplin, pendekatan kuantitatif untuk pengembangan, operasi dan pemeliharaan dari software; atau dapat disimpulkan sebagai teknik aplikasi untuk perangkat lunak. Intinya Software engineering berkaitan dengan pembangunan produk program.

Software engineering terdiri dari 3 elemen kunci, yaitu :

1. Metode,
2. Peralatan (tools),
3. Prosedur,

yang memungkinkan manajer mengontrol proses pengembangan software dan memberikan praktisi dasar yang baik untuk pembentukan software berkualitas tinggi.

a. Metode Software Engineering

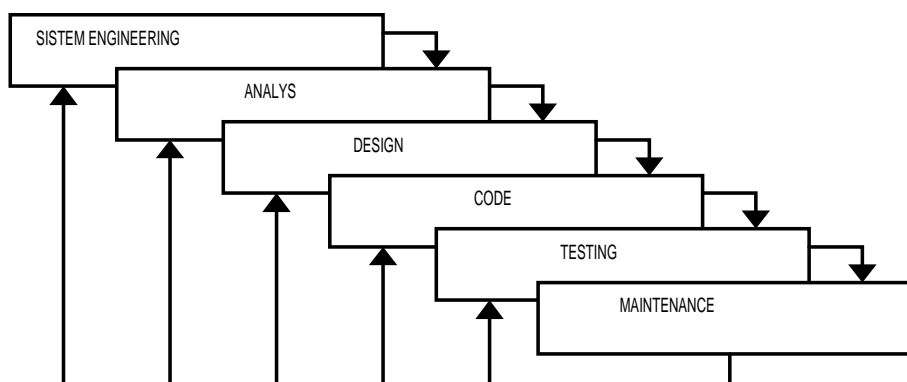
Metode software engineering memberikan teknik-teknik bagaimana membentuk software.

Metode ini terdiri dari serangkaian tugas :

- Perencanaan dan estimasi proyek
- Analisis kebutuhan sistem dan software
- Desain struktur data
- Arsitektur program dan prosedur algoritma
- Coding
- Testing dan pemeliharaan

Dalam model atau paradigma software engineering, dikenal ada 3 metode yang secara luas dipergunakan, yaitu :

1. Classic Life Cycle Pradigm - Model Water Fall - Model Siklus Hidup Klasik



A. System Engineering and Analysis

Karena software merupakan bagian terbesar dari sistem, maka pekerjaan dimulai dengan cara menerapkan kebutuhan semua elemen sistem dan mengalokasikan sebagian kebutuhan tersebut ke software. Pandangan terhadap sistem adalah penting, terutama pada saat software harus berhubungan dengan elemen lain, seperti :

- Hardware
- Software
- Database

B. Analisis kebutuhan software

Suatu proses pengumpulan kebutuhan software untuk mengerti sifat-sifat program yang dibentuk software engineering, atau analisis harus mengerti fungsi software yang diinginkan, performance dan interface terhadap elemen lainnya. Hasil dari analisis ini didokumentasikan dan direview / dibahas / ditinjau bersama-sama customer.

C. Design

Desain software sesungguhnya adalah *proses multi step* (proses yang terdiri dari banyak langkah) yang memfokuskan pada 3 atribut program yang berbeda, yaitu :

- Struktur data
- Arsitektur software
- Rincian prosedur

Proses desain menterjemahkan kebutuhan ke dalam representasi software yang dapat diukur kualitasnya sebelum mulai coding. Hasil dari desain ini didokumentasikan dan menjadi bagian dari konfigurasi software.

D. Coding

Desain harus diterjemahkan ke dalam bentuk yang dapat dibaca oleh mesin

E. Testing

Segera sesudah objek program dihasilkan, pengetesan program dimulai. Proses testing difokuskan pada logika internal software. Jaminan bahwa semua pernyataan atau statements sudah dites dan lingkungan external menjamin bahwa definisi input akan menghasilkan output yang diinginkan.

F. Maintenance

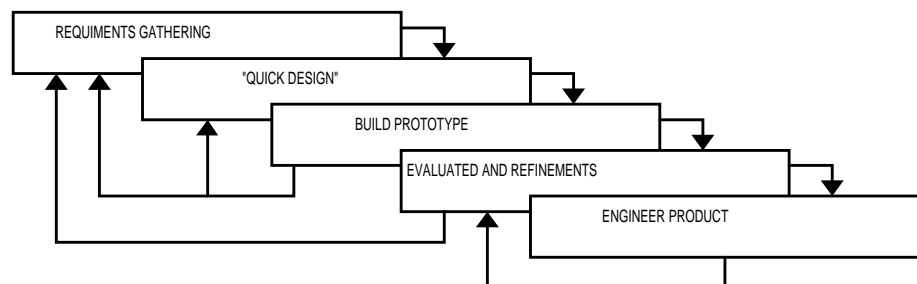
Software yang sudah dikirim ke customer data berubah karena

- Software mengalami error
- Software harus diadaptasi untuk menyesuaikan dengan lingkungan external, misalnya adanya sistem operasi baru atau peripheral baru.
- Software yang lebih disempurnakan karena adanya permintaan dari customer.

Masalah yang dihadapi dari model siklus hidup klasik adalah :

- Proyek yang sebenarnya jarang mengikuti aliran sequential yang ditawarkan model ini. Iterasi (Pengulangan) selalu terjadi dan menimbulkan masalah pada aplikasi yang dibentuk oleh model ini.
- Seringkali pada awalnya customer sulit menentukan semua kebutuhan secara explicit (jelas).
- Customer harus sabar karena versi program yang jalan tidak akan tersedia sampai proyek software selesai dalam waktu yang lama.

2. Prototype Paradigm



Seringkali seorang customer sulit menentukan input yang lebih terinci, proses yang diinginkan dan output yang diharapkan. Tentu saja ini menyebabkan developer tidak yakin dengan efisiensi algoritma yang dibuatnya, sulit menyesuaikan sistem operasi, serta interaksi manusia dan mesin yang harus diambil. Dalam hal seperti ini, pendekatan prototype untuk software engineering merupakan langkah yang terbaik. Prototype sebenarnya adalah suatu proses yang memungkinkan developer membuat sebuah model software.

Ada 2 bentuk dari model ini, yaitu :

A. Paper Prototype

Menggambarkan interaksi manusia dan mesin dalam sebuah bentuk yang memungkinkan user mengerti bagaimana interaksi itu terjadi.

B. Working Prototype

Adalah prototype yang mengimplementasikan beberapa bagian dari fungsi software yang diinginkan seperti pada pendekatan pengembangan software. Model ini dimulai dengan :

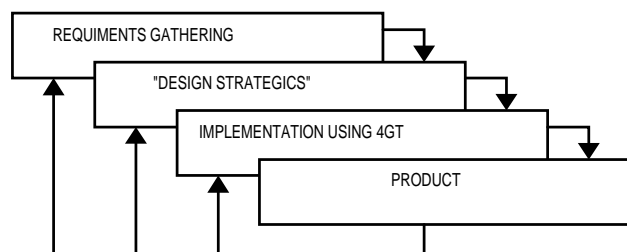
- Pengumpulan kebutuhan developer dan customer
- Menentukan semua tujuan software
- Mengidentifikasi kebutuhan-kebutuhan yang diketahui

Hasil dari pengumpulan kebutuhan diteruskan pada Quick Design. Quick Design ini memfokuskan pada representasi aspek-aspek software yang dapat dilihat oleh user, misalnya format input dan output, selanjutnya dari desain cepat diteruskan pada pembentukan prototype (langkah ke 3). Prototype ini dievaluasi oleh customer / user dan digunakan untuk memperbaiki kebutuhan-kebutuhan software. Proses iterasi terjadi agar prototype yang dihasilkan memenuhi kebutuhan customer, juga pada saat yang sama developer mengerti lebih baik tentang apa yang harus dikerjakan.

Masalah yang dihadapi oleh prototyping paradigm ini adalah :

- Customer hanya melihat pada apa yang dihasilkan oleh software, tidak peduli pada hal-hal yang berhubungan dengan kualitas software dan pemeliharaan jangka panjang.
- Developer seringkali menyetujui apa yang diterangkan oleh customer agar prototype dapat dihasilkan dengan cepat. Akibatnya timbul pemilihan sistem operasi / bahasa pemrograman yang tidak tepat.

3. Fourth Generation Tehnique Paradigm - Model tehnik generasi ke 4 / 4GT



Istilah Fourth Generation Technique (4GT) meliputi seperangkat peralatan software yang memungkinkan seorang developer software menerapkan beberapa karakteristik software pada tingkat yang tinggi, yang ke mudian menghasilkan source code dan object code secara otomatis sesuai dengan spesifikasi yang ditentukan developer. Saat ini peralatan / tools 4GT adalah bahasa non prosedur untuk :

- DataBase Query
- Pembentukan laporan (Report Generation)
- Manipulasi data
- Definisi dan interaksi layar (screen)
- Pembentukan object dan source (Object and source generation)
- Kemampuan grafik yang tinggi, dan
- Kemampuan spreadsheet

Keterangan gambar :

- Model 4GT untuk software engineering dimulai dengan rangkaian pengumpulan kebutuhan. Idealnya, seorang customer menjelaskan kebutuhan-kebutuhan yang selanjutnya diterjemahkan ke dalam prototype. Tetapi ini tidak dapat dilakukan karena customer tidak yakin dengan apa yang diperlukan, tidak jelas dalam menetapkan fakta-fakta yang diketahui dan tidak dapat menentukan informasi yang diinginkan oleh peralatan 4GT.
- Untuk aplikasi kecil adalah mungkin bergerak langsung dari langkah pengumpulan kebutuhan ke implementasi yang menggunakan bahasa non prosedur fourth generation (generasi ke 4). Tetapi untuk proyek besar, pengembangan strategi desain sistem tetap diperlukan, sekalipun kita menggunakan 4GL. Penggunaan 4GT tanpa desain untuk proyek besar akan menyebabkan masalah yang sama yang ditemui dalam pengembangan software yang menggunakan pendekatan konvensional.
- Implementasi yang menggunakan 4GL memungkinkan developer software menjelaskan hasil yang diinginkan yang kemudian diterjemahkan ke dalam bentuk source code dan object code secara otomatis.
- Langkah yang terakhir adalah mengubah implementasi 4GT ke dalam sebuah product. Selanjutnya developer harus melakukan penyetoran, pengembangan dokumentasi dan pelaksanaan semua aktifitas lainnya yang diwujudkan dalam model software engineering.

Masalah yang dihadapi dalam model 4GT adalah adanya sebagian orang yang beranggapan bahwa :

1. peralatan 4GT tidak semudah penggunaan bahasa pemrograman,
2. source code yang dihasilkan oleh peralatan ini tidak efisien,

3. pemeliharaan sistem software besar yang dikembangkan dengan 4GT masih merupakan tanda tanya.

Dari model yang disebut di atas dapat diambil suatu kesimpulan, bahwa proses pengembangan software terdiri dari 3 fase, yaitu :

1. Fase Definisi
2. Fase Pengembangan (Development)
3. Fase Pemeliharaan (Maintenance)

1. Fase Definisi

- Mengidentifikasi informasi apa yang dikerjakan proses
- Fungsi dan performance apa yang diinginkan
- Interface apa yang dibutuhkan
- Hambatan desain apa yang ada, dan
- Kriteria validasi apa yang dibutuhkan untuk menetapkan keberhasilan sistem.

A. Sistem Analis

Sistem analis menetapkan peranan dari setiap elemen dalam sistem berbasis komputer, terutama mengalokasikan peranan software.

B. Sistem Software Planning

Dalam sistem ini, setelah lingkungan software dialokasikan, maka langkah dari sistem software planning ini adalah :

- Pengalokasian sumber / resource
- Estimasi biaya
- Penetapan tugas pekerjaan dan jadwal.

C. Requirement Analysis

Penetapan lingkup untuk software memberikan petunjuk / arah. Namun definisi yang lebih rinci dari informasi dan fungsi software diperlukan sebelum pekerjaan dimulai.

2. Fase Pengembangan

Fase pengembangan berfokus pada “*How*”. Selama pengembangan, developer software berusaha menjelaskan :

- Bagaimana struktur data dan arsitektur software yang didesain

- Bagaimana rincian prosedur diimplementasikan (diterapkan)
- Bagaimana desain diterjemahkan ke dalam bahasa pemrograman atau bahasa non prosedur, dan
- Bagaimana pengetesan akan dilaksanakan.

A. Desain software (Software Design)

Desain menterjemahkan kebutuhan-kebutuhan software ke dalam sekumpulan representasi (grafik, tabel, diagram, atau bahasa yang menjelaskan struktur data, arsitektur software dan prosedur algoritma).

B. Coding

Representasi desain harus diterjemahkan ke dalam bahasa tiruan / artificial language yang menghasilkan perintah-perintah yang dapat dieksekusi oleh komputer.

C. Software Testing

Sejara sesudah software diimplementasikan dalam bentuk yang dapat dieksekusi oleh mesin, software perlu ditest untuk menemukan kesalahan (merupakan fungsi logika dan implementasi).

3. Fase Pemeliharaan

Fase pemeliharaan berfokus pada “*Change*” atau perubahan. Ini dapat disebabkan :

- Perubahan karena software error (Corective Maintenance)
- Perubahan karena software disesuaikan / diadaptasi dengan lingkungan external, misalnya
munculnya CPU baru, sistem operasi baru (Adaptive Maintenance)
- Perubahan software yang disebabkan customer / user meminta fungsi tambahan, misalnya fungsi grafik, fungsi matematik, dll (Perfective Maintenance)

b. Peralatan Software Engineering

Peralatan software engineering memberikan dukungan atau semiautomasi untuk metode.

Contohnya :

- CASE (Case Aided Software Engineering), yaitu suatu software yang menggabungkan software, hardware, dan database software engineering untuk menghasilkan suatu lingkungan software engineering.
- Database Software Engineering, adalah sebuah struktur data yang berisi informasi penting tentang analisis, desain, kode dan testing.

- Analogi dengan CASE pada hardware adalah : CAD, CAM, CAE

c. Prosedur Software Engineering

Terdiri dari :

- urutan-urutan di mana metode tersebut diterapkan
- dokumen
- laporan-laporan
- formulir-formulir yang diperlukan
- mengontrol kualitas software
- mengkoordinasi perubahan yang terjadi pada software

III. KESIMPULAN

Dari makalah ini dapat disimpulkan bahwa :

1. Software engineering adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal requirement capturing (analisa kebutuhan pengguna), specification (menentukan spesifikasi dari kebutuhan pengguna), design, coding, testing sampai pemeliharaan sistem setelah digunakan.
2. Software engineering terdiri dari 3 elemen kunci, yaitu :
 - a. Metode,
 - b. Peralatan (tools),
 - c. Prosedur,
3. Proses pengembangan software terdiri dari 3 fase, yaitu :
 - a. Fase Definisi
 - b. Fase Pengembangan (Development)
 - c. Fase Pemeliharaan (Maintenance)

DAFTAR PUSTAKA

[www.umb.ac.id/rekayasa perangkat lunak](http://www.umb.ac.id/rekayasa_perangkat_lunak) diakses Selasa, 19 mei 2009 pukul 19.00 WIB

[www.wikipedia.com/software engineering](http://www.wikipedia.com/software_engineering) diakses Selasa, 19 Mei 2009 pukul 19.30 WIB

[www.ilmukomputer.com/software engineering](http://www.ilmukomputer.com/software_engineering) diakses Selasa, 19 Mei 2009 pukul 20.00 WIB