

PERBANDINGAN TEKNOLOGI SYSTEM SOFTWARE, APPLICATION SOFTWARE, EMBEDDED SOFTWARE DAN WEB APPLICATIONS

Danandjaya Saputra (danandj@gmail.com),
Nanny Raras Setyoningrum (ra2s_ukhti@yahoo.co.id)
Zuhry (zuhryculli@gmail.com)
Mahasiswa Program Pasca Sarjana, Magister Teknik Informatika
STMIK Amikom, Yogyakarta



ABSTRAK

Perangkat lunak (Software) merupakan program-program komputer yang berguna untuk menjalankan suatu pekerjaan sesuai dengan yang dikehendaki. Program tersebut ditulis dengan bahasa khusus yang dimengerti oleh komputer. Perangkat lunak terdiri atas perangkat lunak system (system software), perangkat lunak aplikasi (application software), aplikasi web (web applications) dan perangkat lunak tertanam (Embedded Software). Perbedaan diantara perangkat lunak tersebut adalah perangkat lunak sistem melaksanakan tugas-tugas dasar tertentu yang diperlukan semua pengguna untuk komputer. Aplikasi software adalah program yang dirancang untuk membuat pengguna lebih produktif. Web Application atau aplikasi web adalah suatu perangkat lunak yang diakses menggunakan web browser melalui suatu jaringan seperti Internet atau intranet. Embedded software adalah perangkat lunak yang ditulis untuk mengontrol mesin atau perangkat elektronik. Embedded software dibuat dan ditanamkan ke chip peralatan elektronik

Kata Kunci: software engineering, system software, application software, embedded software, web applications

1. PENDAHULUAN

1.1 Software Engineering

Software Engineering benar-benar tidak dapat lepas dari kehidupan modern. *Software engineering* atau lebih kita kenal dengan rekayasa perangkat lunak menjadi mesin yang mengendalikan pengambilan keputusan di dalam dunia bisnis. Seorang *software engineer* pasti melakukan rekayasa perangkat lunak dalam pembuatan *software* sesuai dengan tatacara yang ditentukan.

Banyak definisi mengenai *software engineering*, Menurut Fritz Bauer (dikutip dalam Yasin, V, 2012, halaman 15), *software engineering* merupakan pembuatan *software* dengan menggunakan prinsip engineering untuk mendapatkan *software* yang ekonomis, handal dan efisien. Menurut IEEE, *software engineering* merupakan pendekatan aplikasi yang sistematis, disiplin, terukur untuk mengembangkan, melakukan operasi dan perbaikan *software*. Sedangkan menurut Roger S. Pressman (2002), perangkat lunak adalah (1) perintah (program computer) yang bila dieksekusi memberikan fungsi dan unjuk kerja seperti yang diinginkan, (2) struktur data yang memungkinkan program memanipulasi informasi secara proporsional, (3) dokumen yang menggambarkan operasi dan kegunaan program.

1.2 Software Engineering Layers

Untuk mendapatkan hasil *software* yang berkualitas maka harus memperhatikan manajemen kualitas dalam pengembangan *software*. Lapisan-lapisan dalam *software engineering* (rekayasa perangkat lunak) seperti terlihat pada gambar 1.



Gambar 1. *Software engineering layers*

Proses merupakan salah satu landasan dalam *software engineering*. Menjadi elemen penting yang menyatukan lapisan lainnya dan memungkinkan pengembangan yang rasional dan tepat waktu.

Metode, merupakan cara yang dilakukan untuk membangun suatu *software*. Metode mencakup analisis kebutuhan, desain, konstruksi program, pengujian, dan support. Prinsip yang dilakukan oleh software engineer bergantung pada jenis masalah yang akan dibuat kedalam bentuk model.

Tools akan memberikan dukungan untuk lapisan proses dan metode. Saat alat-alat saling terintegrasi, memungkinkan informasi yang dibuat oleh tool dapat digunakan oleh tool yang lain.

1.3 Fase *Software Engineering*

Proses dalam rekayasa perangkat lunak dapat dikelompokkan kedalam 3 fase generic dimana setiap fase memiliki tujuan yang berbeda. Fase-fase tersebut yaitu :

a. Definition phase

Pada fase ini seorang software engineer mengidentifikasi informasi untuk diproses, apa saja fungsi yang akan dibuat, sistem apa yang akan diterapkan, desain apa yang harus dibuat sehingga menghasilkan software yang bagus.

b. Development phase

Dalam arsitektur perangkat lunak, development phase berisi bagaimana detail prosedural diimplementasikan, bagaimana *interface* dibangun, bagaimana desain *engineer* ditranslate kedalam sebuah bahasa program, dan bagaimana testing dilakukan. Pada umumnya ada 3 kegiatan yang selalu dilakukan *software engineer*, meliputi desain *software*, *code generation*, *testing software*.

c. Support phase

Fase ini isinya adalah pengulangan fase definisi dan development. Terjadi karena adanya perubahan yang memungkinkan baik dari user ataupun faktor lain. Lingkup dari aktifitas ini meliputi (umbrella activities):

- tracking dan kontrol software
- review formal teknis
- jaminan kualitas software
- manajemen pengaturan software
- dokumentasi persiapan dan pembuatan
- manajemen penggunaan kembali
- Pengukuran
- manajemen resiko

2. *SYSTEM SOFTWARE*

System software melaksanakan tugas-tugas dasar tertentu yang diperlukan semua pengguna untuk komputer. *System software* disiapkan oleh pembuat *software* (vendor) atau perusahaan yang mengkhususkan diri dalam membuat *software* (penjual atau pemasok *software*).

Dasar dari *system software*, meliputi : sistem operasi(*operating system*), program utilitas dan penerjemah bahasa komputer(*language translator*).

a. Sistem operasi

Sistem operasi (*operating system*) adalah software yang berfungsi untuk mengaktifkan seluruh perangkat yang terpasang pada komputer sehingga masing-masingnya dapat saling berkomunikasi, tanpa ada sistem operasi maka komputer tak dapat difungsikan sama sekali. Contoh sistem operasi adalah: DOS, Unix, Linux, OS/2, Windows, Mac OS dan lain-lain.

Secara umum *operating system* adalah mengelola seluruh sumber-daya yang terdapat pada sistem komputer dan menyediakan sekumpulan layanan (*system calls*) ke pengguna sehingga memudahkan dan kenyamanan penggunaan serta pemanfaatan sumber daya sistem komputer.

Untuk menghindari konflik yang terjadi pada saat pengguna menggunakan sumber daya yang sama, sistem operasi mengatur pengguna mana yang dapat mengakses suatu sumber-daya. Sistem operasi juga sering disebut *resource allocator*.

b. Program utilitas

Program utilitas berfungsi untuk membantu atau mengisi kekurangan/kelemahan dari sistem operasi, misalnya PC Tools dapat melakukan perintah format sebagaimana DOS, tapi PC Tools mampu memberikan keterangan dan animasi yang bagus dalam proses pemformatan. File yang telah dihapus oleh DOS tidak dapat dikembalikan lagi tapi dengan program bantu, hal ini dapat dilakukan. Seperti Norton Utility, Scandisk, PC Tools, dan lain-lain. Program utilitas misalnya :

- Program Antivirus
Mengidentifikasi dan menghapus virus-virus dalam memory, media storage, dan file yang akan dimasukkan.
- Spyware
Program yang ditempatkan pada komputer tanpa pengetahuan user yang mengumpulkan informasi-informasi rahasia tentang pengguna.
- Utilitas Pengkompres File

Memperkecil ukuran untuk membebaskan ruang dan meningkatkan kinerja. Compressed files kadang-kadang disebut file zipped. Ada beberapa program utility, yaitu: PKZIP, WinZip, WinRAR dan lain-lain.

- Software CD/DVD Burning

Membaca teks, grafik, audio, dan file video untuk dapat direkam ulang atau dapat dibaca ulang oleh CD atau DVD.

- Utilitas Perawatan Komputer Pribadi

Mengidentifikasi dan memperbaiki masalah sistem operasi dan masalah disk.

Selain itu ada juga utilitas yang lain, dimana memungkinkan untuk memulihkan file yang hilang atau rusak, memantau kinerja sistem, bahkan mengendalikan arus data antara pengguna dan komputer.

3. APPLICATION SOFTWARE

Application software adalah program yang dirancang untuk membuat pengguna lebih produktif. Software aplikasi umumnya melakukan pekerjaan pemrosesan informasi bagi pengguna akhir (*end user*). Contohnya : Word Processing, spreadsheet dan lain-lain. Program aplikasi khusus menyelesaikan tugas pemrosesan informasi yang mendukung fungsi atau proses bisnis tertentu, misalnya software aplikasi bisnis.

Software aplikasi bisnis merupakan perangkat lunak yang membantu orang-orang menjadi lebih efisien dan efektif. Contohnya, software aplikasi bisnis mendukung perekayasaan ulang dan otomatisasi proses bisnis dengan aplikasi *e-bussines* strategis seperti manajemen hubungan pelanggan, *enterprise resources planning*, dan manajemen rantai pasokan. Contoh lainnya adalah software yang dapat dioperasikan dalam web seperti *electronic commerce*, atau dalam berbagai area fungsional seperti manajemen sumber daya manusia, akuntansi dan keuangan. Masih ada software lainnya yang memberdayakan para manajer dan praktisi bisnis dengan alat pendukung keputusan seperti penambang data, portal informasi perusahaan, atau sistem manajemen pengetahuan.

4. EMBEDDED SOFTWARE

Embedded software adalah perangkat lunak yang ditulis untuk mengontrol mesin atau perangkat elektronik. *Embedded software* dibuat dan ditanamkan ke chip peralatan elektronik, mulai dari peralatan elektronik di telepon, pesawat, komputer, mobil, peralatan audio, robot, mainan, sistem keamanan, playstation, televisi dan jam tangan digital.

Embedded software (Perangkat lunak tertanam) saat ini telah menjadi tugas yang cukup menyulitkan untuk pengembang software karena diperlukan fungsi seperti algoritma control dengan bahasa pemrograman yang tepat. Embedded software ditulis dalam bahasa yang lebih dekat atau dikenali oleh hardware, misalnya Assembly, C, C++, Java Dengan meningkatnya kompleksitas dan kualitas tuntutan Embedded Software, bagaimanapun, peran software engineering menjadi lebih penting agar menghasilkan Embedded software yang berkualitas.

a. Desain sistem embedded software, terdiri dari :

- *Periodic Stimuli*, Ini terjadi pada interval waktu prediksi. Sebagai contoh, sistem dapat memeriksa sensor setiap 50 milidetik dan mengambil tindakan (respon) tergantung pada nilai sensor (stimulus).
- *Aperiodic Stimuli*, Ini terjadi tidak teratur dan tak terduga dan biasanya mengisyaratkan menggunakan mekanisme interrupt komputer. Contoh stimulus tersebut akan interupsi yang menunjukkan bahwa suatu I / O Transfer adalah lengkap dan data yang tersedia dalam buffer.

b. Architectural patterns dalam embedded software, terdiri dari :

- *Observe and React*, Pola ini digunakan ketika satu set sensor secara rutin dipantau dan ditampilkan. Ketika sensor menunjukkan bahwa beberapa peristiwa telah terjadi (misalnya, panggilan masuk di ponsel), sistem bereaksi dengan memulai proses.
- *Environmental Control*, Pola ini digunakan ketika sistem termasuk sensor, yang memberikan informasi tentang lingkungan dan aktuator yang dapat mengubah lingkungan. Dalam menanggapi perubahan lingkungan terdeteksi oleh sensor, sinyal kontrol dikirim ke aktuator sistem.
- *Process Pipeline*, Pola ini digunakan ketika data harus diubah dari satu representasi yang lain sebelum dapat diproses. transformasi diimplementasikan sebagai urutan langkah-langkah pengolahan, yang dapat dilakukan secara bersamaan. Hal ini memungkinkan untuk pengolahan data yang sangat cepat, karena inti atau prosesor yang terpisah dapat mengeksekusi setiap transformasi.

c. Real-time operating system dalam embedded software, terdiri dari :

- *A real time clock*, yang menyediakan informasi yang diperlukan untuk jadwal proses berkala.
- Interrupt handler, yang mengelola permintaan aperiodik untuk layanan.
- *Scheduller*, yang bertanggung jawab untuk memeriksa proses yang dapat dieksekusi dan memilih salah satu dari ini untuk eksekusi.
- *Resource Manager*, yang mengalokasikan memori yang sesuai dan prosesor sumber daya untuk proses yang telah dijadwalkan untuk eksekusi.

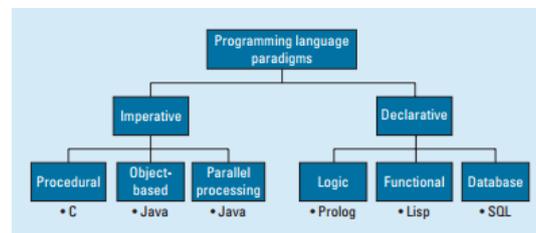
- *Dispatcher*, yang bertanggung jawab untuk memulai pelaksanaan proses

5. WEB APPLICATIONS

Dalam proses pengembangan aplikasi web berbeda dengan proses pengembangan perangkat lunak konvensional karena aplikasi web kontennya sangat kompleks dan fungsionalitas sangat luas dari proses rekayasa pada *end-users*. Web menggunakan proses pembangunan inkremental. Hal ini karena persyaratan berkembang dari waktu ke waktu dan perubahan akan sering terjadi. Hal ini para pengguna harus mengikuti jaman yang dilakukan para pengembang lalu pengguna tetap mengikutinya dan begitu seterusnya.

Perubahan adalah sebuah proses berkelanjutan dari proses pengembangan aplikasi web meliputi dari tahap analisis, desain, implementasi dan pemeliharaan. Urutan ini bisa dilakukan dalam pembuatan aplikasi web skala kecil maupun skala besar, namun perbedaan dalam skala kecil pembuatan aplikasi web biasanya harganya murah sedangkan pada skala besar harganya mahal.

Setiap tahunnya sudah banyak bahasa pemrograman yang berkembang namun salah satu cara untuk memahami persamaan dan perbedaan setiap pengembang adalah mengklasifikasi kedalam hirarki paradigma berdasarkan model perhitungan mereka. Biasanya, tingkat klasifikasi terdiri atas dua paradigma penting seperti bahasa C, Java dan JavaScript dan deklaratif seperti CSS, HTML, dan SQL yang dimana dibagi menjadi sub paradigma seperti pada gambar berikut ini :



Gambar 2. Paradigma Bahasa Pemrograman

Perbedaan utama antara kedua tingkat diatas yaitu bahwa deklaratif tentukan program apa yang seharusnya dilakukan tapi bukan bagaimana melakukan. Kemudian bahasa-bahasa yang imperatif memiliki kekuatan yang lebih eksperesif tetapi umumnya sulit untuk digunakan dan dipahami.

Karakteristik Aplikasi Web, dapat dilihat pada tabel dibawah ini :

<p>Aplikasi Web Sederhana</p> <ul style="list-style-type: none"> • kurang penekanan pada aestherics / antar muka pengguna • konten informasi statis • navigasi sederhana • Sistem berdiri sendiri • Kinerja tinggi bukan syarat utama • dikembangkan oleh individu / tim kecil • digunakan untuk penyebaran informasi adalah aplikasi non inti
<p>Aplikasi Web yang kompleks</p> <ul style="list-style-type: none"> • lebih menekankan pada estetika, user interface • informasinya dinamis • navigasi yang kompleks • terintegrasi dengan database dan sistem lainnya • diperlukan kinerja tinggi • membutuhkan tim pengembangan yang besar

6. KESIMPULAN

Perkembangan *software engineering* dapat menghasilkan *system software*, *application software*, *embedded software* dan *web applications*, dari masing masing jenis tersebut memiliki perbedaan, pada pembahasan diatas dapat disimpulkan bahwa :

System software melaksanakan tugas-tugas dasar tertentu yang diperlukan semua pengguna untuk komputer. *System software* disiapkan oleh pembuat *software* (vendor) atau perusahaan yang mengkhususkan diri dalam membuat *software* (penjual atau pemasok *software*). Dasar dari *system software*, meliputi : sistem operasi(*operating system*), program utilitas dan penerjemah bahasa komputer(*language translator*).

Application software adalah program yang dirancang untuk membuat pengguna lebih produktif. Software aplikasi umumnya melakukan pekerjaan pemrosesan informasi bagi pengguna akhir (*end user*). Contohnya : Word Processing, spreadsheet dan lain-lain.

Embedded software adalah perangkat lunak yang ditulis untuk mengontrol mesin atau perangkat elektronik. *Embedded software* dibuat dan ditanamkan ke chip peralatan elektronik, mulai dari peralatan elektronik di telepon, pesawat, komputer, mobil, peralatan audio, robot, mainan, sistem keamanan, playstation, televisi dan jam tangan digital.

Web application atau aplikasi web adalah suatu perangkat lunak yang diakses menggunakan *web browser* melalui suatu jaringan seperti internet atau intranet.

DAFTAR PUSTAKA

Budhi, Romy. 2009. *Embedded System*. Yogyakarta : Penerbit ANDI.

Chuanqi Kan. 2016. DoCloud:An Elastic Cloud Platform for Web Applications Based on Docker. School of Electronic Information and Electrical Engineering Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai, China

Mouna Hammoudi, Gregg Rothermel, Andea Stocco : WATERFALL: An Incremental Approach for Repairing Record-Replay Tests of Web Applications. University of Nebraska - Lincoln, USA, Università di Genova, Italy

Peter Liggesmeyer & Mario Trapp, Trends in Embedded Software Engineering, *IEEE Software Magazine*, pp 19-24, 2009.

Pressman, Roger S. 2002. *Rekayasa Perangkat Lunak*. Yogyakarta : Penerbit ANDI.

Tihomir Orehovački, Snježana Babić. Qualitative Approach to Determining the Relevant Facets of Mobile Quality of Educational Social Web Applications. Juraj Dobrila University of Pula, Department of Information and Communication Technologies, Pula, Croatia

Yasin, Verdi. 2012. *Rekayasa Perangkat Lunak Berorientasi Objek*. Jakarta : Mitra Wacana Media.