

ANALISA METODE *CLASSIC LIFE CYCLE (WATERFALL)* UNTUK PENGEMBANGAN PERANGKAT LUNAK MULTIMEDIA

IWAN BINANTO

Teknik Informatika

Universitas Sanata Dharma, Yogyakarta, Indonesia

iwana@usd.ac.id

Abstrak

Metode Waterfall merupakan metode pengembangan perangkat lunak yang selalu diperkenalkan kepada mahasiswa jurusan ilmu komputer/teknik informatika melalui suatu matakuliah. Hal ini mengakibatkan metode Waterfall hampir selalu digunakan sebagai metode untuk pengembangan perangkat lunak. Padahal tidak semua perangkat lunak dapat dikembangkan menggunakan metode tersebut, salah satunya adalah perangkat lunak multimedia. Penelitian ini menggunakan metode kualitatif untuk mendiskusikan metode Waterfall sebagai metode pengembangan perangkat lunak multimedia. Hasilnya metode Waterfall tidak sesuai diterapkan sebagai metode pengembangan perangkat lunak multimedia karena karakteristik perangkat lunak multimedia yang berbeda dengan perangkat lunak lainnya.

Katakunci: pengembangan perangkat lunak multimedia, *waterfall*, *software engineering*, *classic life cycle*

1. PENDAHULUAN

Metode Waterfall sering disebut dengan *classic life cycle* (Pressman, 2010). Metode ini merupakan metode pengembangan perangkat lunak terstruktur yang paling dikenal dan banyak digunakan secara luas, tidak hanya di lingkup akademisi tetapi juga di industri. Hal ini disebabkan Waterfall merupakan metode yang tua dan matang (Huo et. al., 2004), (Khalaf & Al-Jedaiah, 2008), (Petersen et. al., 2009), (Pressman, 2010). Oleh sebab itu, metode ini hampir selalu digunakan untuk pengembangan perangkat lunak terutama di lingkup akademisi khususnya mahasiswa yang sedang menyelesaikan Tugas Akhirnya.

Saat ini banyak perangkat lunak yang mengintegrasikan berbagai macam media ke dalamnya sebagai sarana peningkatan kualitas dan daya tarik perangkat lunak tersebut. Perangkat lunak yang menggabungkan berbagai macam media pada makalah ini disebut dengan perangkat lunak multimedia. Perangkat lunak yang tidak mengintegrasikan berbagai macam media ke dalamnya pada makalah ini disebut dengan perangkat lunak klasik

Perangkat lunak multimedia mempunyai karakteristik yang berbeda dengan perangkat lunak klasik. Hal ini disebabkan multimedia merupakan kombinasi dari elemen teks, gambar diam/foto/seni grafis, suara, animasi, dan video yang dimanipulasi secara digital (Molina & Vilamil, 1997), (French & Haynes, 2003), (Vaughan, 2004).

Penulis mengamati bahwa metode Waterfall banyak digunakan untuk pengembangan perangkat lunak multimedia terutama di kalangan mahasiswa. Penggunaan ini sepertinya sesuai dan tidak ada kendala dalam implementasinya. Tetapi jika melihat karakteristik perangkat lunak multimedia yang berbeda dengan perangkat lunak klasik, mestinya tidak sesuai dan akan mempunyai kendala yang

berarti. Oleh karena itu penulis akan menganalisa apakah metode Waterfall sesuai sebagai metode pengembangan perangkat lunak multimedia.

2. METODE PENELITIAN

Makalah ini menggunakan metode kualitatif dan merupakan penelitian pustaka. Makalah ini memusatkan perhatian pada metode Waterfall selaku metode pengembangan perangkat lunak yang tua dan matang. Terutama metode Waterfall versi Sommerville (2011). Hal ini berangkat dari keprihatinan peneliti tentang penggunaan metode pengembangan perangkat lunak yang tidak sesuai dengan peruntukannya.

3. ANALISA DAN INTERPRETASI DATA

3.1. Waterfall

Metode Waterfall pertama kali diperkenalkan oleh Royce pada tahun 1970 (Petersen et.al., 2009) dengan 7 (tujuh) tahapan yang berurut walaupun juga mempunyai *feedback loop* antar tahapan jika diperlukan, seperti terlihat pada gambar 1 (Royce, 1970). Metode ini mengalami banyak perbaikan dan perubahan diantaranya adalah perubahan langkah dari 7 (tujuh) menjadi 5 (lima) tahapan (Pressman, 2010), (Sommerville, 2011) seperti terlihat pada gambar 2a dan 2b.

Metode Waterfall yang dibahas pada subbab ini merupakan metode Waterfall versi Sommerville (2011). Pertimbangan menggunakan versi Sommerville karena metode ini merupakan versi terbaru dari metode Waterfall pada makalah ini. Metode ini mempunyai tahapan-tahapan sebagai berikut:

a. *Requirements analysis and definition*

Layanan sistem, kendala, dan tujuan ditetapkan oleh hasil konsultasi dengan pengguna yang kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

b. *System and software design*

Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.

c. *Implementation and unit testing*

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

d. *Integration and system testing*

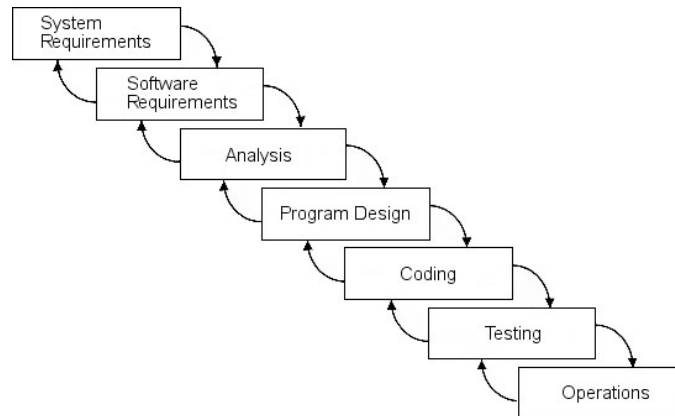
Unit-unit individu program atau program digabung dan diuji sebagai sebuah sistem lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak. Setelah pengujian, perangkat lunak dapat dikirimkan ke *customer*.

e. *Operation and maintenance*

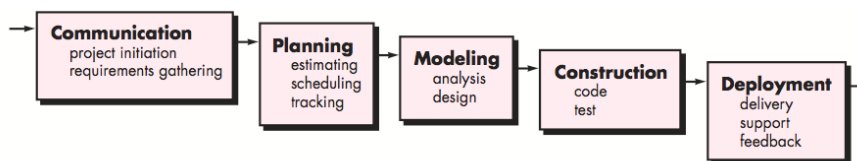
Biasanya (walaupun tidak selalu), tahapan ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata. *Maintenance* melibatkan pembetulan kesalahan yang tidak ditemukan pada tahapan-tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai kebutuhan baru.

Pada prinsipnya, setiap tahapan di metode Waterfall menghasilkan satu atau lebih dokumen yang sudah disetujui ("ditandatangani"). Tahap berikutnya tidak dapat dimulai sebelum tahapan sebelumnya selesai. Dalam tataran praktis, tahapan-tahapan tersebut saling tumpang tindih (*overlap*) dan memberikan informasi satu sama lain.

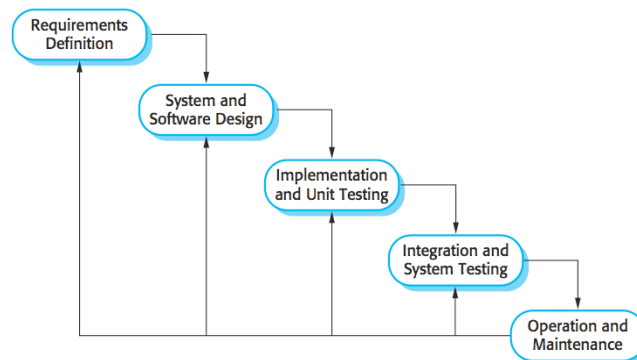
Pada waktu perancangan (*design*), masalah-masalah dengan persyaratan diidentifikasi. Pada waktu pengkodean (*coding*), dapat ditemukan masalah perancangan, walaupun juga masalah lainnya. Proses pengembangan perangkat lunak bukan merupakan model linier yang sederhana karena juga melibatkan umpan balik (*feedback*) dari satu tahapan ke tahapan lainnya. Dokumen yang dihasilkan pada setiap tahapan ada kemungkinan harus diubah supaya sesuai dengan perubahan yang sudah dibuat (Sommerville, 2011).



Gambar 1. Metode Waterfall versi Royce (1970)



Gambar 2a. Metode Waterfall versi Pressman (2010)



Gambar 2b. Metode Waterfall versi Sommerville (2011)

Iterasi menjadi mahal dan melibatkan pengulangan pekerjaan yang signifikan dilihat dari biaya produksi dan penyetujuan dokumen. Oleh karena itu, setelah sejumlah kecil iterasi, akan dilakukan “pembekuan” sebagian dari pengembangan, seperti spesifikasi, dan melanjutkan ke tahapan selanjutnya. Beberapa masalah disisakan untuk diselesaikan kemudian atau diabaikan (Sommerville, 2011).

Pada tahapan akhir (*operation and maintenance*) perangkat lunak mulai digunakan. Kesalahan dan kelalaian pada *software requirements* awal dapat ditemukan. Beberapa kesalahan perancangan atau yang lainnya terjadi dan kebutuhan akan fungsi baru teridentifikasi. Oleh karena itu, sistem harus berevolusi (membuat

perubahan) untuk tetap dapat digunakan. Membuat perubahan ini (*maintenance software*) dapat melibatkan pengulangan tahapan sebelumnya (Sommerville, 2011).

Metode Waterfall merupakan contoh dari tahapan yang *plan-driven*, yaitu secara prinsip pengembang harus merencanakan dan menjadwalkan semua kegiatan tahapan sebelum mulai mengembangkan perangkat lunak. Komitmen dibuat di awal tahapan sehingga membuat sulit untuk merespon ketika ada perubahan pada kebutuhan *customer* (Sommerville, 2011).

Metode Waterfall tetap digunakan hingga saat ini karena keuntungan-keuntungannya. Salah satu keuntungan metode Waterfall adalah linearitasnya sehingga memudahkan dan memperjelas penyajian *Quality Assurance* (Khalaf & Al-Jedaiah, 2008). Mungkin dengan alasan ini Godfrey (1995) memberikan turunan metode waterfall untuk menjadi metode pengembangan perangkat lunak multimedia.

3.2. Karakteristik Perangkat Lunak Multimedia

Perangkat lunak multimedia mempunyai karakteristik yang berbeda dengan perangkat lunak lainnya. Hal ini disebabkan multimedia merupakan kombinasi dari elemen teks, gambar diam/foto/seni grafis, suara, animasi, dan video yang dimanipulasi secara digital (Villamil & Molina, 1997), (French & Haynes, 2003), (Vaughan, 2004). Untuk memanipulasi elemen-elemen multimedia ini dibutuhkan keahlian di masing-masing elemen tersebut. Elemen-elemen ini akan dikumpulkan lebih dulu sebelum digabungkan ke dalam perangkat lunak multimedia yang sedang dibangun untuk menjadi sebuah perangkat lunak yang utuh. Membuat dan/atau mengubah elemen-elemen multimedia membutuhkan keahlian khusus sesuai dengan elemen yang diperlukan dan tidak membutuhkan keahlian membuat program.

Ada hal lain disamping membuat dan/atau mengubah elemen-elemen yang digunakan, yaitu mengumpulkan elemen-elemen yang sudah dibuat agar mempermudah pengembangan perangkat lunak multimedia. Oleh sebab itu diperlukan metode pengembangan perangkat lunak multimedia yang dapat mengakomodasi hal ini.

Metode pengembangan perangkat lunak multimedia hanya sedikit sehingga aktifitas pengembangan dan metode yang digunakan sangat beragam tergantung pada latar belakang produsernya. Tetapi walaupun beragam, aktifitas yang umum dilakukan adalah tahap desain dan produksi yang berbasis tim dan kompleks. Hal ini akan menghasilkan produk akhir yang spesifik dan mempunyai ciri khas yang unik (Sherwood & Root, 1998). Beberapa metode pengembangan perangkat lunak multimedia yang ada, menyinggung tentang tim pengembang bahkan memasukkan manajemen tim ke dalam metode (Villamil & Molina, 1997), (Sherwood & Root, 1998), (Vaughan, 2004).

3.3. Pembahasan

Seperti dipaparkan di atas, multimedia mempunyai karakteristik khusus yang tidak dipunyai perangkat lunak lainnya, yaitu elemen-elemen pembangunnya. Karakteristik khusus ini membutuhkan penanganan khusus ketika mengembangkan perangkat lunak multimedia. Elemen-elemen ini tidak dapat disamakan dengan kode-kode program atau unit program karena masing-masing elemen membutuhkan keahlian khusus yang berbeda-beda.

Sebagai contoh, pada kasus membuat dan/atau mengubah gambar diam yang merupakan salah satu elemen multimedia. Pada kasus ini, dimisalkan gambar diam membutuhkan banyak perubahan yang disesuaikan dengan rancangan yang dibuat. Untuk mengubah gambar diam ini tidak terakomodasi pada metode Waterfall karena

tidak ada tahapan untuk mengubah elemen multimedia. Pada kasus seperti ini akhirnya perubahan gambar diam dipaksakan masuk ke salah satu tahapan pada Waterfall. Tahapan yang paling mungkin adalah di tahapan *Implementation and Unit Testing* (Sommerville, 2011). Jika menggunakan versi lama metode Waterfall akan masuk ke tahapan *Coding* (Royce, 1970) atau *Construction* (Pressman, 2010).

Hal ini tidak sesuai karena membuat dan/atau mengubah gambar diam bukan merupakan *Implementation and Unit Testing* yang notabene membuat program atau unit program. Membuat dan/atau mengubah gambar diam tidak sama dengan membuat program atau unit program karena memang tidak melakukan hal tersebut. Membuat dan/atau mengubah gambar diam merupakan bagian dari manipulasi elemen multimedia dan/atau pengumpulan elemen multimedia.

Pada kasus di atas, baru dibahas tentang membuat dan/atau mengubah gambar diam yang dikerjakan secara teknis. Masih ada pemaksaan lain di tahapan sebelumnya, yaitu *System and Software Design*. Tahapan ini merupakan tahapan untuk merancang sebuah perangkat lunak yang tidak mengakomodasi perancangan elemen-elemen multimedia di dalamnya.

Pemaksaan ini mengakibatkan tahapan pada metode Waterfall digunakan dengan tidak semestinya dan tidak sesuai. Peneliti mengkhawatirkan akan terjadi kesulitan dalam pengembangan perangkat lunak multimedia terutama ketika perangkat lunak yang dikembangkan mengalami pembaruan. Terutama ketika pengembang secara ketat mengikuti tahapan-tahapan metode pengembangan perangkat lunaknya.

4. KESIMPULAN

Berdasarkan paparan di atas, dapat disimpulkan bahwa metode Waterfall tidak sesuai untuk digunakan sebagai metode pengembangan perangkat lunak multimedia. Pemaksaan penggunaan metode Waterfall menjadi metode pengembangan perangkat lunak multimedia dikhawatirkan akan mempersulit pengembang itu sendiri ketika perangkat lunak membutuhkan pembaruan.

DAFTAR PUSTAKA

- French, F., Haynes, R., 2003, *Notes for Course Developers: Multimedia Development Life Cycle*, [pdf], ([www2.plymouth.ac.uk/ed/ELT documents/materials/mmguide.pdf](http://www2.plymouth.ac.uk/ed/ELT/documents/materials/mmguide.pdf), diakses tanggal 13 Maret 2014)
- Godfrey, R., 1995, "New Wine in Old Bottle: Multimedia Design Methodology", *ASCILITE '95*, Melbourne, Australia.
- Huo, M., Verner, J., Zhu, L., Babar, M.A. (2004) "Software Quality and Agile Methods", *Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC '04)*.
- Khalaf, S.J., Al-Jedaiah, M.N. (2008) "Software Quality and Assurance in Waterfall model and XP – A Comparative Study", *WSEAS Transactions on Computers Vol. 7, Issue 12 pp. 1968-1976*.
- Petersen, K., Wohlin, C., Baca, D. (2009) "The Waterfall Model in Large-Scale Development", *Proceedings of 10th International Conference, PROFES 2009, pp. 386-400*.
- Pressman, R.S. (2010) *Software Engineering: a Practitioner's Approach 7th Edition*, McGraw-Hill Higher Education.
- Sherwood, C. and Rout, T., 1998, *A Structured Methodology for Multimedia Product*

- and Systems Development*, ASCILITE '98
- Sommerville, I. (2011) *Software Engineering 9th Edition*. Addison-Wesley.
- Royce, W. W. (1970) "Managing The Development of Large Software Systems", *Proceedings of IEEE WESCON*, pp. 1-9.
- Vaughan, T. (2004) *Multimedia: Making It Work 6th Edition*, McGraw-Hill Companies, Inc.
- Villamil, J., Molina, L. (1997) *Multimedia: Production, Planning, and Delivery*, Que Education & Training