

METODOLOGI XPDSDM: ALTERNATIF METODOLOGI PENGEMBANGAN SISTEM INFORMASI DI INDONESIA

Arip Mulyanto

Fakultas Teknik Universitas Negeri Gorontalo

Abstrak: Seiring kemajuan teknologi informasi yang saat ini telah didukung oleh teknologi jaringan komputer, sistem informasi berbasis komputer menjadi media yang dinilai efektif dan efisien bagi penunjang operasional berjalannya suatu organisasi. Hal ini membawa dampak kepada fenomena berlomba-lombanya organisasi besar ataupun kecil untuk memiliki, mengembangkan, dan menggunakan sistem informasi yang berbasis komputer. Fenomena tersebut terjadi pula di Indonesia yang notabene hanya berlabel sebagai negara berkembang. Paper ini membahas dan menganalisa secara umum karakteristik pengembangan sistem informasi di Indonesia. Untuk selanjutnya diajukan suatu metodologi alternatif bagi pengembangan sistem informasi yang dinilai sesuai dengan karakteristik tersebut. Metodologi tersebut adalah XPDSDM yang merupakan perkawinan antara metodologi XP dan DSDM.

Kata-kata kunci: Metodologi, XPDSDM, pengembangan sistem informasi

Telah menjadi kenyataan bahwa bangsa Indonesia lebih memiliki mental sebagai pedagang dibandingkan sebagai investor (Indrajit, 2001). Hal tersebut tidak terlepas dari sejarah panjang Bangsa Indonesia, yang sebelum mencapai kemerdekaannya, telah dijajah selama kurang lebih tiga setengah abad oleh Belanda dan Portugis. Secara nyata mental tersebut membentuk kepribadian masyarakat Indonesia pada umumnya. Mental tersebut ternyata berimbas pada sebgayaan besar perilaku masyarakat Indonesia, termasuk dalam pengembangan sistem informasi. Sering kali, perancangan dan pengembangan sebuah sistem informasi didasari tujuan jangka pendek, dimana *instant feedback* senantiasa menjadi harapan dan digunakan sebagai tolok ukur evaluasi pencapaian. Sebuah sistem informasi yang dinilai tidak mampu memberikan *instant feedback* akan dinyatakan gagal dan besar kemungkinan akan dibuang atau diputuskan untuk tidak digunakan. Fenomena ini juga membawa dampak terhadap kecenderungan relatif pendek

dan singkatnya periode pengembangan sistem informasi yang ada di Indonesia.

Instant feedback itu sendiri memiliki beberapa acuan yang kerap digunakan sebagai bahan evaluasi. Sebagai contoh misalkan peningkatan *income*, sebuah organisasi yang mengelola sebuah sistem informasi akan menilai seberapa signifikan peningkatan keuntungan yang dapat diraih oleh organisasi setelah pengoperasian sistem informasi jika dibandingkan dengan kondisi sebelum adanya sistem informasi. Sayangnya durasi evaluasi keuntungan sering kali dilakukan dengan periode waktu yang relatif singkat. Umumnya hanya satu semester sampai dengan satu tahun. Sehingga penentuan keputusan bahwa sebuah sistem informasi dinyatakan dan disimpulkan gagal kerap datang dari fakta dan premis pendukung yang kurang cukup lengkap dan kurang cukup menunjang. Di sisi lain, misalkan, kepuasan pelanggan atau secara luas klien yang berhubungan dan berinteraksi dengan organisasi tidak dinilai sebagai suatu pentahapan bagi pencapaian *income generator* tadi. Kita ketahui peningkatan kepuasan pihak luar dalam berinteraksi tentu akan meningkatkan loyalitas pihak tersebut untuk senantiasa menjalin kerjasama yang baik dengan organisasi. Sedangkan *income* yang muncul dari aspek ini hanya dapat diukur hasilnya dalam suatu periode waktu yang cukup panjang, mencapai lima sampai dengan 10 tahun.

Dari aspek pendidikan, secara global Masyarakat Indonesia dapat dikategorikan pada level menengah ke bawah. Khususnya dalam bidang teknologi informasi, seringkali seorang calon *owner* tidak dapat memaparkan harapannya secara detail terhadap sistem informasi yang akan dibangun. *Owner* seringkali hanya menyatakan bahwa sistem informasi yang akan dibangun haruslah yang efektif (*profit*), efisien (*low cost*), mudah digunakan (*people oriented*) serta bagus dan indah dipandang mata (*HCI aspect*). Tentu saja ini menjadikan pengembang mengalami kesulitan dalam menspesifikasikan kebutuhan sistem. Akibatnya spesifikasi kebutuhan sistem tidak terdefinisikan dengan jelas.

Dari beberapa aspek lainnya, trend untuk saat ini, aspek politik dan ekonomi Bangsa Indonesia dinilai kurang stabil. Penilaian tersebut didukung dengan terjadinya “*riak-riak*” dalam dunia perpolitikan Indonesia seperti contohnya *reshuffle* kabinet yang penuh pro dan kontra. Dari aspek ekonomi fakta yang cukup mendukung penilaian tersebut adalah kembali langkanya beberapa kebutuhan pokok masyarakat seperti bahan bakar dan beras. Secara langsung stabilitas politik dan ekonomi akan mempengaruhi perkembangan dunia usaha yang ada di Indonesia. Sehingga sebuah sistem informasi yang akan dikembangkan sangat diharapkan bersifat fleksibel terhadap rentannya

perubahan kebijakan pemerintah (*legacy aspect*) maupun perubahan stabilitas ekonomi (*economy aspect*). Sifat fleksibilitas ini harus telah dimiliki oleh sebuah sistem informasi mulai dari tahapan pengembangan (*development*) sampai dengan tahapan pengoperasiannya (*operational*) kelak. Fleksibilitas dalam tahapan pengembangan dapat berupa sifat akomodatif terhadap perubahan spesifikasi kebutuhan sistem. Sedangkan pada tahapan operasional dapat berupa kemudahan *maintenance* maupun kemungkinan penambahan fitur baru pada sistem, pada kondisi sistem yang telah jadi.

Secara umum, proyek pengembangan SI di Indonesia memiliki karakteristik sebagai berikut: (1) Tidak jelas dan tidak tetapnya spesifikasi kebutuhan sistem informasi yang akan dibangun; (2) Durasi pengembangan yang relatif singkat; sering kali para calon *owner* dari sebuah sistem informasi ingin memperoleh *instant feed back* sesegera mungkin. (3) Multi orientasi: *people, profit, HCI*; Sistem Informasi yang dikembangkan harus efektif (*profit*), efisien (*low cost*), mudah digunakan (*people*) serta bagus dan indah dipandang mata (HCI). (4) Skalabilitas yang beragam, organisasi maupun individu yang ingin memiliki sistem informasi sangat beragam mulai dari skala kecil, menengah, sampai dengan organisasi berskala besar.

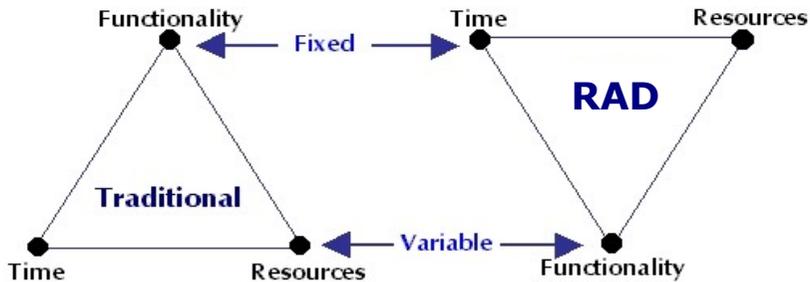
Bagian selanjutnya, akan membahas mengenai beberapa jenis metodologi dalam pengembangan sistem informasi.

Metodologi DSDM

Dynamic System Development Method (DSDM) dikembangkan oleh Konsorsium yang tersusun atas perwakilan dari beragam *vendors* dan para ahli di bidang sistem informasi (*IS Experts*) (Stapleton, 1997). DSDM lahir sebagai respon dari ketidak-puasan akan metodologi *Rapid Application Development* (RAD) yang berimej “*quick but dirty*” (Avison & Fitzgerald, 2006). Yang menjadi kritikan adalah sebuah metodologi tidak hanya dituntut mewujudkan proses pengembangan yang cepat namun juga harus tetap menjaga kualitas dari sistem informasi yang dikembangkan.

Pada awalnya, RAD dinilai sebagai alternatif solusi dari metodologi konvensional yang terkesan kaku terhadap perubahan spesifikasi dan cenderung membutuhkan waktu pengembangan yang panjang dan relatif lama. Pada RAD, komponen pengembangan sistem informasi yang bersifat tetap (*fixed*) adalah waktu dan sumber daya. Di sisi lain fungsionalitas sistem bersifat dinamis dan akomodatif terhadap perubahan (Avison & Fitzgerald, 2006). Kami menginterpretasikan fungsionalitas sistem di sini sebagai spesifikasi kebutuhan yang diakomodasikan oleh sistem. Gambar 1 mengilustrasikan perbandingan sudut pandang terhadap ketetapan komponen

pengembangan sistem pada metodologi tradisional dibandingkan dengan RAD.



Gambar 1. Waktu, sumber daya, dan fungsionalitas pada Tradisional dan RAD (www.codeproject.com, 2007)

Sebuah filosofi yang menjadi landasan berkembangnya metodologi RAD adalah, bahwa kesulitan untuk mengidentifikasi keseluruhan spesifikasi kebutuhan sistem di awal tahapan pengembangan merupakan suatu keniscayaan. Spesifikasi kebutuhan yang baru cenderung muncul ketika proses perancangan dan testing telah dilakukan. Ada kemungkinan pada suatu kasus, dimana spesifikasi kebutuhan tidak akan pernah terliputi hingga 100% meskipun tahapan pengembangan telah selesai dilaksanakan. Misalkan karena keterbatasan sumber daya ataupun ketersediaan teknologi.

Filosofi ini sangat mewarnai tahapan metodologi RAD. Pada RAD proses pengembangan kebutuhan sistem dilandaskan kepada skala prioritas menggunakan *MoSCoW rules*. Intinya adalah bahwa yang akan dikembangkan, diakomodasikan oleh sistem, terlebih dahulu adalah seluruh fungsi-fungsi yang memang dianggap paling perlu. Dilanjutkan dengan yang perlu hingga fungsi-fungsi tambahan dengan prioritas lebih rendah, jika dirasakan memungkinkan. Penentuan prioritas ini sangat tergantung kepada kehandalan sistem analis dalam merepresentasikan penentuan tiap-tiap kebutuhan sistem berdasarkan level prioritas keberadaannya terhadap bagian sistem secara keseluruhan.

Menurut Yourdon seperti tercantum pada buku *Information System Development (methodologies, technique, & Tools)* yang ditulis Avison dan Fitzgerald (2006). "Pada RAD, Penekanan yang berlebih dari pengembang terhadap aspek waktu pengembangan yang cepat menjadikan alasan bagi pengembang untuk menghilangkan fungsi-fungsi yang sebaiknya diimplementasikan. Selain itu fokus yang menjadi perhatian hanya terbatas kepada

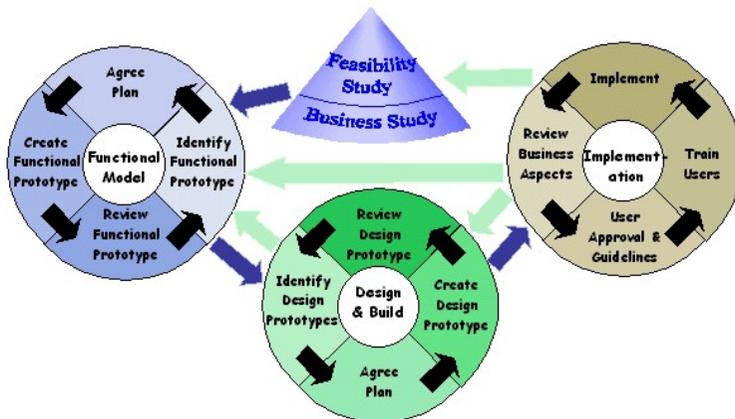
pemodelan aplikasi, data, proses, dan teknologi. Padahal aspek lainnya seperti presentasi, kontrol, keamanan dan komunikasi merupakan komponen pengembangan yang sebaiknya turut juga diperhatikan.”

Oleh karena itu, penggunaan metodologi RAD sangat tidak dianjurkan bagi aplikasi-aplikasi yang akan berdampak vital jika terjadi kesalahan. Sistem aplikasi monitoring *co-pilot* pada penerbangan misalnya, merupakan sistem aplikasi yang tidak dianjurkan menggunakan metodologi RAD pada pengembangannya.

Pada DSDM, kelemahan RAD yang hanya memberikan perhatian pada sebagian komponen pengembangan mulai diakomodasikan. DSDM menyediakan dan melengkapi RAD dengan suatu *framework* yang mengontrol proses pembangunan dan pemeliharaan sistem yang menjamin kesuksesan proses iterasi pada RAD dengan batasan waktu yang ketat dan sumber daya yang tetap. Pengembangan sistem tidak hanya menggunakan sudut pandang pengembang tetapi juga meliputi pihak terkait lainnya seperti *user*, manajer (*owner*), dan personil dari *quality assurance*. Sehingga kontrol dan pengawasan pengembangan sistem lebih bersifat objektif. Kondisi ini juga, menjadikan DSDM memiliki fleksibilitas pada aspek skalabilitas. DSDM dapat digunakan untuk pengembangan sistem dengan skala kecil sampai dengan skala besar.

Tahapan Proses DSDM

Tahapan proses DSDM diilustrasikan menggunakan diagram “*three pizzas and a chesse*”. Gambar 2 memberikan visualisasi diagram tersebut. Garis panah berwarna biru tua pada diagram mengindikasikan proses urutan transisi secara normal pada tahapan pengembangan. Garis panah berwarna hijau mengindikasikan arah yang memungkinkan untuk ditempuh sewaktu-waktu sesuai kebutuhan. Sebagai contoh misalkan ketika suatu grup tim pengembang telah menyelesaikan tahapan pada iterasi “*Design and Build*” pada suatu area, akan tetapi sistem tidak dapat dikirimkan karena membutuhkan area fungsionalitas lainnya yang belum terdefinisikan maka aliran proses pengembangan dimungkinkan kembali kepada tahapan iterasi “*Functional Model*” untuk selanjutnya melengkapi area fungsionalitas yang dibutuhkan tadi.



Gambar 2. Diagram “three pizzas and a chesse”
 (www.codeproject.com, 2007).

Feasibility Study (Studi Kelayakan)

Pada tahapan ini tim pengembang akan mengkaji apakah sistem yang akan dikembangkan cukup layak untuk dikembangkan berdasarkan keterbatasan waktu dan sumber daya yang telah ditentukan. Selain itu pada tahapan ini juga meliputi peninjauan kembali apakah metodologi DSDM cukup relevan untuk digunakan bagi pengembangan sistem. Tahapan ini dikerjakan secepat mungkin untuk mendukung pengiriman spesifikasi sistem secara cepat.

Business Study (Studi Aspek Bisnis)

Pada tahapan ini tim pengembang akan mempelajari aspek bisnis dari sistem yang akan dikembangkan. Pengkajian pada tahapan ini merupakan pengkajian pada *high level*. Pada tahapan ini akan ditentukan *business rules* yang akan diterapkan pada sistem. Tahapan ini membutuhkan suatu teknik pengumpulan data dan informasi akan spesifikasi kebutuhan sistem. Teknik *interview* dan observasi pada metodologi tradisional dianjurkan untuk tidak digunakan, karena cenderung lama. Teknik yang disarankan untuk digunakan adalah JAD (*Joint Application Development*) *workshop*. Pada tahapan ini akan ditentukan fungsional utama apa saja yang akan diprioritaskan untuk diakomodasi pada keseluruhan arsitektur sistem. Selain itu akan dirancang juga gambaran umum rencana kerja pengembangannya. Rencana kerja

meliputi juga strategi apa yang akan digunakan pada pengembangan *prototyping*. Rencana kerja ini akan mengalami perbaikan (*refinement*) selama tahapan proses pengembangan, seiring dengan semakin banyaknya informasi spesifikasi kebutuhan dan fungsional sistem yang dapat terdefiniskan. Satu komponen penting lainnya yang dihasilkan pada tahapan ini adalah, definisi dari area bisnis. Definisi area bisnis biasanya terdiri dari pemodelan proses dan data. Perlu diketahui bahwa DSDM dirancang untuk bersifat independen terhadap pendekatan teknologi pemodelan proses dan data. Baik pendekatan terstruktur maupun berorientasi objek, keduanya dapat digunakan pada DSDM. Yang menjadi penekanan pada hal ini adalah bahwa ketika, katakanlah pendekatan terstruktur yang digunakan maka pemodelan proses dan data dengan pendekatan terstruktur harus secara konsisten digunakan. Sebagai contoh untuk pemodelan proses dapat digunakan DFD, sedangkan untuk pemodelan data dapat digunakan ERD.

Functional Model Iteration (Iterasi Model Fungsional).

Pada tahapan ini informasi yang berhasil di-*capture* pada tahapan *business study* akan mengalami *refinement*. Fungsional *prototype* dari sistem akan dihasilkan untuk sekaligus di-*review*. Fungsional *prototype* adalah fungsi-fungsi apa saja yang harus diakomodasikan oleh sistem, pada tahapan *review* akan dikaji bagaimana seharusnya sistem mengakomodasikan fungsi-fungsi tersebut. *Prototype* dihasilkan untuk menelaah beragam aspek fungsional dari sistem yang terdiri dari: a) *Business*-bagaimana kesesuaian fungsional sistem terhadap fungsional bisnis yang telah ditetapkan untuk diakomodasikan?; b) *Usability*-Seberapa mudah penggunaan sistem?; c) *Performance and capacity*-Dapatkah sistem menangani suatu ukuran volume dan intensitas transaksi yang telah ditetapkan?; d) *Technique*-Terkait teknik terbaik apa yang dapat ditempuh untuk mengakomodasikan kebutuhan fungsional sistem ini?

Design And Build Iteration (Iterasi Perancangan dan Pembangunan)

Tahapan untuk merancang *prototype* yang siap dikirimkan pada *user*. Pada tahapan ini setidaknya ada beberapa kebutuhan sistem yang sudah dapat dijalankan (terakomodasikan) pada *prototype* yang dihantarkan. Secara bertahap spesifikasi kebutuhan dengan prioritas *must have* dan beberapa spesifikasi kebutuhan dengan prioritas *should have* akan mulai dikirimkan kepada *user*. Proses awal pada tahapan ini akan meliputi rancangan dan pembangunan per-*area* fungsional yang telah didefinisikan pada tahapan pemodelan fungsional. Ketika area terkait dapat dibangun maka dengan segera *prototype*-nya dihasilkan untuk selanjutnya dihantarkan kepada *user*. Akan tetapi jika terkait atau membutuhkan fungsional area lainnya yang belum didefinisikan diperlukan *back transition* kepada tahapan pemodelan fungsional untuk melengkapi kebutuhan fungsional dari area yang tidak lengkap tadi. Secara detail rancangan algoritma, rancangan *interface* sistem, dan rancangan fisik *database* akan dihasilkan pada tahapan ini. Seluruh rancangan ini secara cepat akan diwujudkan dalam bentuk *prototype* sistem. Pada tahapan ini pengujian *prototype* yang merepresentasikan sistem keseluruhan bukan merupakan fokus perhatian, meskipun pada beberapa area terkadang dibutuhkan pengujian secara parsial untuk suatu area rancangan sistem tertentu. Pada kebanyakan kasus pengembangan, pada tahapan ini merupakan tahapan dimana untuk pertama kalinya representasi spesifikasi sistem secara keseluruhan mulai dapat difungsikan untuk digunakan.

Implementation (Implementasi)

Tahapan ini merupakan tahapan terakhir dari DSDM. Pada tahapan ini meliputi penghantaran produk akhir hasil implementasi sistem secara keseluruhan berdasarkan rancangan spesifikasi kebutuhan pada tahapan *design and build*. Pada tahapan inilah sistem keseluruhan akan diujikan dan di-*review* kesesuaiannya terhadap aspek bisnis. Selain itu aktifitas melengkapi dokumentasi pengembangan serta *manual* penggunaan sistem merupakan salah satu aktifitas yang dilakukan pada tahapan ini. Idealnya yang membuat *manual* penggunaan sistem adalah *user*. Perlu diingat *user* disini adalah *user* yang terlibat aktif pada pengembangan sistem, *stakeholder* yang berperan sebagai *user*. Sebagai akhir pada tahapan ini akan dihasilkan *Project Review Document*, komponen utama dari isi pada dokumen tersebut adalah penilaian terkait dengan apakah pengembangan sistem dinyatakan telah selesai karena

dianggap telah memenuhi seluruh spesifikasi kebutuhan atau sebaliknya masih membutuhkan iterasi pengembangan.

Metodologi *Extreme Programming* (XP)

Extreme Programming (XP) sering diterapkan untuk mendukung pengembangan sebuah sistem pada skala kecil dan menengah (Mark, 2001). Berdasarkan pengalaman, penerapan XP akan berjalan optimal pada pengembangan sistem yang membutuhkan kuantitas programmer pada kisaran 3 sampai dengan 10 orang. Penerapan XP pada pengembangan sistem dengan ukuran jumlah *programmer* yang lebih besar menjadikan fenomena perubahan spesifikasi kebutuhan yang cepat pada XP menjadi sulit untuk dikomunikasikan.

Dibandingkan sebagai tahapan langkah-langkah metodologi, XP lebih menyerupai rangkaian urutan prinsip-prinsip yang dapat digunakan untuk mengembangkan sebuah sistem secara cepat (*rapid*) (Buimester, 2001). Hal inilah yang menjadikan banyak pakar IS mengkategorikan XP sebagai *lightweight methodology*. Pada XP tidak terdapat aturan-aturan yang kompleks, selain itu proses kemajuan dokumentasi memiliki karakteristik *slow down progress*.

Tahapan Proses XP

Metodologi XP tersusun atas 4 tahapan proses (fase) pengembangan sistem. Keempat tahapan tersebut adalah *Planning* (Perencanaan), *Designing* (Perancangan), *Developing code* (Pengembangan Kode Program), dan *Productionalizing* (Analisis Produksi). Berikut ini akan dipaparkan keempat tahapan proses penyusun XP.

***Planning* (Perencanaan)**

Tahapan perencanaan berkaitan dengan penentuan *scope* (batasan) dari sistem yang akan dikembangkan, penentuan prioritas pengerjaan fungsi-fungsi (terkait urutan waktu pengerjaan setiap fungsi - *timing*), penentuan anggota tim yang terlibat dalam pengembangan, pemaparan secara detail *content* dari setiap *increment progress* yang akan dikirimkan, Penentuan perkiraan besarnya biaya (*cost*) yang akan digunakan dan aktifitas yang terakhir pada tahapan ini adalah penentuan kesepakatan, terkait dengan level evaluasi yang akan digunakan (seberapa ketat level pengujian yang akan diterapkan). Pada XP tahapan akuisisi spesifikasi kebutuhan dengan mengejawantahkan *user stories*. *User stories* adalah suatu paparan dari *costumer* yang digunakan untuk menyebutkan kebutuhan apa saja yang

diinginkan *costumer* untuk dapat diakomodasikan oleh sistem. Sifat dari *user stories* sendiri masih bersifat sangat umum, dengan penentuan *scope*, *content*, *cost*, dan *timing* pada setiap *task* (fungsi) yang akan diimplementasikan diharapkan spesifikasi kebutuhan lebih jelas tergambar. Tahapan *planning* seringkali dibagi menjadi dua bagian yaitu *release plan* dan *iteration plan*. *Release plan* mencakup tahapan perencanaan keseluruhan sistem, sedangkan *iteration plan* hanya mencakup perencanaan *task* pada setiap spesifik iterasi. Durasi *iteration plan* dianjurkan pada kisaran satu sampai dengan 3 minggu.

Designing

Pada tahapan ini akan diterapkan prinsip *simplicity*, *feedback*, *courage* dan akomodatif terhadap perubahan spesifikasi kebutuhan. Telah disebutkan sebelumnya pada XP pertemuan para *stakeholder* dapat diselenggarakan secara harian. Begitupun pada tahapan ini, pertemuan harian para *stakeholder* memungkinkan untuk diselenggarakan. Pertemuan pada tahapan ini misalkan untuk menentukan kode program mana yang dinilai paling *simple* yang akan diterapkan. Sehingga jelas penentuan ukuran *simplicity* kode pemrograman melibatkan seluruh *stakeholders*. Oleh karena itu kode program sebagai hasil penentuan ini disebut dengan “*collective code ownership*”. Untuk mendukung perancangan pada tahapan ini digunakan *architecture spike*. Landasan yang menjadi dasar penggunaan *architecture spike* bagi perancangan adalah untuk mereduksi resiko kesalahan perancangan hasil pengejawantahan *user story* pada aspek perancangan teknis. *Architecture spike* adalah suatu metode yang dapat digunakan untuk menggambarkan perancangan teknis dari sistem. Biasanya berupa program sederhana yang dapat digunakan untuk mengeksplorasi solusi potensial. Pada arsitektur ini sistem dikembangkan dengan hanya memfokuskan (mengimplementasikan) permasalahan yang ingin diuji.

Developing the Code

Menerapkan konsep *pair programming*. Pada *pair programming* untuk setiap *work station* akan ditempatkan dua orang *programmer*. Peranan yang dimiliki oleh salah seorang *programmer* adalah mengkodekan sebuah *task* dengan cara terbaik (efisien dalam eksekusi). Di sisi lain *programmer* yang kedua akan memikirkan suatu strategi terbaik menerapkan *task* tersebut pada keseluruhan sistem (integrasi). Pada pelaksanaannya kedua *programmer* akan saling bertukar peranan secara dinamis. Pada tahapan ini setiap kode yang dihasilkan pada setiap iterasi akan diujikan secara mandiri maupun pengujian

penerapannya pada integrasi sistem. Dari kondisi ini sangat jelas bahwa *customer* senantiasa harus berinteraksi secara intens dengan *programmer*. *Customer* diharapkan dapat memberikan *feed back* secara cepat terkait menjawab pertanyaan tentang spesifikasi kebutuhan yang diterima yang telah diakomodasikan oleh sistem.

Productionalizing

Tahapan ini meliputi pengujian performa sistem secara keseluruhan. Tahapan ini dilakukan untuk menjamin performa terbaik dari perangkat lunak yang diproduksi. Pada beberapa metodologi tahapan ini dimasukkan pada tahapan *maintenance*.

Pembahasan

Kelemahan DSDM

DSDM dinilai lemah dari sisi tahapan pemrograman. Pada DSDM tidak dilakukan penekanan secara detail pada aspek pemrograman. Implikasinya tidak ada kontrol yang menjamin aktifitas pemrograman terlaksana dengan baik. Kenyataannya monitoring aktifitas pemrograman pada rentang waktu yang ketat merupakan suatu hal yang penting. Pada tahapan implementasi, kode program yang diterapkan hanya merupakan rangkaian kode program yang baru teruji pada *prototipe* kondisi ini memiliki potensi kegagalan hasil implementasi ketika diterapkan pada sistem sesungguhnya.

Kelemahan XP

Pada XP titik berat pengembangan diberikan pada aspek pemrograman. Setiap kode program yang diterapkan merupakan kode pemrograman yang memiliki tingkat kesederhanaan dan efisiensi eksekusi yang paling baik dari seluruh kemungkinan kode pemrograman yang teridentifikasi dapat digunakan. Di sisi lain XP memiliki karakteristik kemajuan dokumentasi yang lambat. Kurang baiknya dokumentasi mengakibatkan XP memiliki kontrol yang kurang baik terhadap daur *life cycle* sistem. Sehingga XP memiliki aspek skalabilitas yang kurang baik. XP memiliki kecenderungan hanya cocok untuk diterapkan pada pengembangan sistem berskala kecil dan menengah.

Metodologi XPDSDM

Telah dipaparkan sebelumnya bahwa beberapa kelemahan DSDM adalah pada aspek pemrograman. Di sisi lain ada fenomena yang menarik pada metodologi pengembangan sistem informasi yang lain. Metodologi

tersebut adalah *eXtreme programming*. *eXtreme programming* (XP) merupakan suatu metodologi yang menekankan pada aspek pemrograman. Akan tetapi dinilai lemah dari sisi kontrol secara keseluruhan daur *life cycle system*. XP juga dinilai kurang cukup baik pada manajemen pelaksanaan pengembangan sistem secara keseluruhan. Akibatnya penggunaan XP spesifik pada pengembangan sistem dengan skala kecil dan menengah. Hal ini menjadikan aspek skalabilitas XP dinilai kurang fleksibel. Padahal telah dipaparkan sebelumnya bahwa DSDM memiliki keunggulan pada aspek kontrol dan monitoring keseluruhan tahapan daur *life cycle system* (dinilai baik dalam aspek skalabilitas). Dari fenomena ini penyusun mendapatkan inspirasi untuk melakukan perbaikan terhadap DSDM dengan memasukkan beberapa konsep XP pada tahapan implementasi DSDM. Sebelumnya Tuffs telah menggabungkan DSDM dengan RUP (1999).

Aspek Modifikasi

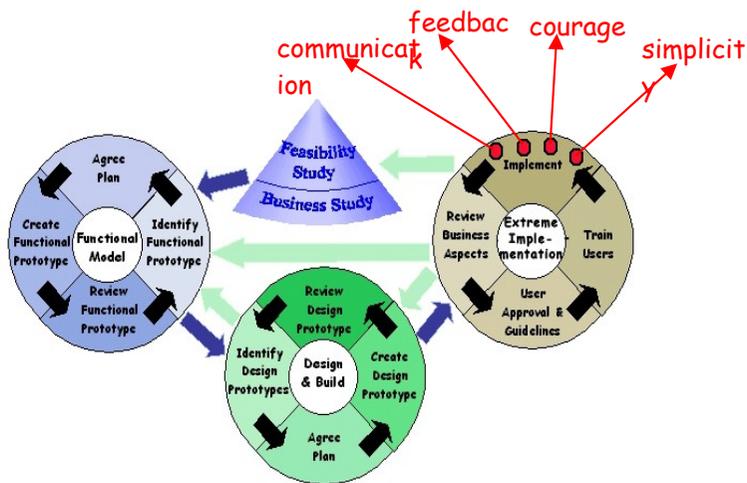
Menurut Avison dan Fitzgerald (2006) beberapa aspek yang memungkinkan untuk dilakukan perbaikan pada sebuah *Information System Development Methodology* (ISDM) diantaranya adalah terhadap: (1) Tahapannya; (2) Teknik yang digunakannya; (3) *Tools* yang digunakan; dan (4) Aspek Filosofis. Pada tulisan ini modifikasi terhadap metodologi DSDM dilakukan terhadap aspek tahapan. Tahapan yang menjadi target modifikasi adalah tahapan implementasi pada DSDM. Modifikasi yang dilakukan berupa penerapan beberapa konsep pemrograman pada XP terhadap tahapan implementasi DSDM.

XPDSDM sebagai Hasil Modifikasi

Modifikasi yang dilakukan ditempuh dengan melekatkan atau menerapkan komponen-komponen nilai pada metodologi XP kepada tahapan implementasi. Komponen-komponen nilai tersebut terdiri atas: *communication*, *feed back*, *encourage*, dan *simplicity*. Secara garis besar metodologi XPDSDM yang diajukan tersusun atas 5 tahapan utama yaitu: (1) *Feasibility Study*; (2) *Business Study*; (3) *Functional Model Iteration*; (4) *Design and Build Iteration*; (5) *Extreme Implementation*.

Terlihat sekali pada XPDSDM, jika dibandingkan dengan DSDM, memang hanya mengalami perbedaan pada tahapan yang ke-5. Hal ini sesuai dengan batasan modifikasi yang memang hanya diterapkan pada tahapan *implementation* yang dimodifikasi menjadi *Extreme Implementation*. Untuk mengilustrasikan tahapan proses pada XPDSDM tetap digunakan diagram

three pizza and a cheese, tentu saja dengan sedikit modifikasi (Gambar 3) pada bagian pizza yang ketiga (pizza berwarna coklat).



Gambar 3. Diagram *three pizzas and a chesse* XPDSM

Empat tahapan yang pertama pada XPDSM identik sama dengan DSDM. Pada Tahapan yang ke-5 yaitu *Extreme Implementation* akan dipaparkan berikut ini.

***Extreme Implementation* XPDSM**

Pada tahapan ini masih merupakan penghantaran produk akhir hasil implementasi sistem secara keseluruhan yang berdasarkan rancangan spesifikasi kebutuhan pada tahapan *design and build*. Pada tahapan ini diterapkan *pairing programming* seperti pada XP sehingga percepatan proses implementasi dimungkinkan untuk diraih. Untuk mendapatkan komponen nilai *communication* setiap task hasil implementasi penyusun keseluruhan sistem diujikan secara parsial dan ketika telah terintegrasi. Yang merancang skenario pengujian adalah user. Pada Tahapan ini, *user* dikondisikan senantiasa berinteraksi secara intens dengan *programmer* sehingga setiap adanya perubahan yang harus diakomodasikan akan segera cepat tersampaikan. Begitupun untuk *task* yang sudah cukup dapat diterima akan tercatat dengan baik. Aktifitas ini merupakan peraih komponen nilai *feedback*.

Sebagai pencapaian komponen *courage*, terkait dengan perbaikan dan perubahan kode program, kepada *programmer* diberikan hak secara bebas

untuk melakukannya, dengan catatan pencantuman proses perubahan dalam dokumentasi. Kode program yang digunakan pada implementasi merupakan kode program yang memenuhi aspek *simplicity*. Kode program diharapkan sesederhana mungkin dan seefisien mungkin dari sudut pandang waktu eksekusi. Selanjutnya seperti pada DSDM sistem keseluruhan akan diujikan dan di-*review* kesesuaiannya terhadap aspek bisnis. Selain itu aktifitas melengkapi dokumentasi pengembangan serta *manual* penggunaan sistem merupakan salah satu aktifitas yang masih dilakukan pada tahapan ini. Sebagai akhir pada tahapan ini akan dihasilkan *Project Review Document*, yang berisi penilaian terkait dengan apakah pengembangan sistem dinyatakan telah selesai dan telah memenuhi seluruh spesifikasi kebutuhan atau sebaliknya masih membutuhkan iterasi pengembangan.

Harapan Yang Ingin Dicapai

XPDSDM diharapkan dapat menggabungkan keunggulan XP dan DSDM. XP diharapkan memberikan kekuatan pada aspek pemrograman sedangkan DSDM mendukung metodologi dari aspek kontrol daur *life cycle* dari sistem secara keseluruhan. Hingga pada akhirnya XPDSDM mewujudkan sebuah metodologi pengembangan sistem informasi yang dapat mendukung permasalahan dengan karakteristik: (1) Tidak jelas dan tidak tetapnya spesifikasi kebutuhan; (2) Durasi pengembangan yang relatif singkat; (3) Multi orientasi: people, profit, HCI; (4) Skalabilitas yang beragam. Hingga pada akhirnya XPDSDM dapat diajukan sebagai metodologi pengembangan sistem informasi di Indonesia.

Simpulan

Mengkombinasikan XP dan DSDM akan menjadikan metodologi kuat dalam menghadapi tahapan pemrograman yang berat, dengan tetap menjaga kualitas hasil. Bagi DSDM aspek pemrograman menjadi kokoh, dan bagi XP skalabilitas meningkat. XPDSDM telah diujikan berhasil untuk digunakan dalam pengembangan Sistem informasi Akademik SMA Taruna Andigha (ADAPTRA). XPDSDM sangat mendukung karakteristik proyek pengembangan SI di Indonesia. Tidak jelas dan tidak tetapnya spesifikasi kebutuhan: a) Durasi pengembangan yang relatif singkat; b) Multi Orientasi; c) Skalabilitas yang beragam; d) Dapat diharapkan untuk keseluruhannya terakomodasi

DAFTAR PUSTAKA

- Avison D. and Fitzgerald G. 2006. *Information systems development: methodologies, techniques and tools*. Mc Graw-Hill.
- Baumeister, Hubert. 2001. "Using XP to develop a CRM framework". Institut für Informatik Ludwig-Maximilians-Universität Oettingenstr. 67 D-80538 München, Germany +49 89 2180 9375
- DSDM Consortium.1998. *Guidelines for Introducing DSDM Into an Organization, evolving DSDM Culture*. UK.
- Eko, Richardus Indarajit.2001. Pengantar Konsep Dasar Manajemen Sistem Informasi dan Teknologi Informasi. Jakarta: Elexmedia Komputindo.
<http://codeproject.com> [Akses 22 Mei 2009]
- Mark, Paulk. C. 2001. "Extreme Programming from a CMM Perspective". Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213 USA.
- Norfolk, David.2000."Understanding DSDM "; Jurnal: PC Network Advisor. Design/Implementing Programming. Issue 67.
- Stapleton, Jenifer.1997."Dynamic System Development Method: The Method in Practice". Addison- Wesley Longman. ISBN 0-201-17889-3.