

MANAJEMEN PROYEK PERANGKAT LUNAK

Furry Arifin

Jurusan Sistem Informasi, Fakultas Ilmu Komputer, Binus University
Jl. KH. Syahdan No. 9, Palmerah, Jakarta Barat 11480.
furry_arifin@gmail.com

ABSTRACT

This article contains several things to be controlled by a Project Manager in order to be able to develop a software engineering project plan. The development methodology begins with survey, analysis and engineering in software planning. Establishing a good project team structure is as important as selecting team members who are also experts in their field. Perceiving that the construction and implementation of information technology-based systems are executed using a project management approach, the project manager is then the most responsible officer for the team performance. The project Manager will lead his team to planning and developing information technology projects in accordance with their intended target. There are three fields that must be considered in an information system project, namely: information systems, information technology, and information management.

Keywords: *project, project management, software project*

ABSTRAK

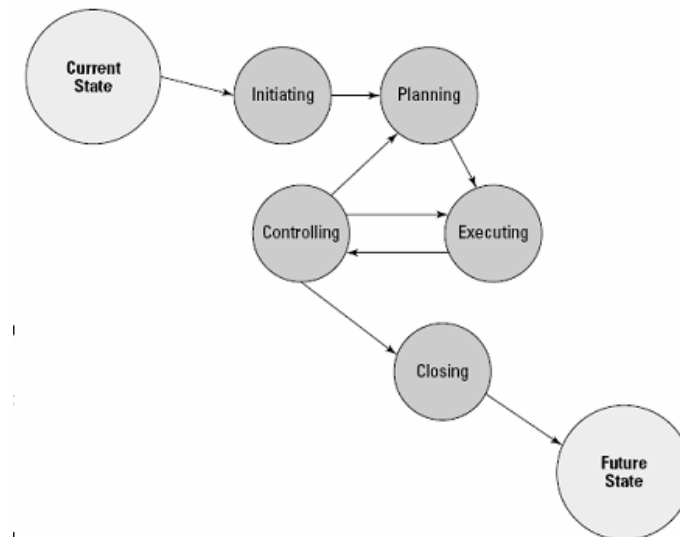
Tulisan ini berisi beberapa hal yang perlu dikuasai oleh seorang Manajer Proyek (Project Manager) agar dapat menyusun suatu rencana proyek rekayasa perangkat lunak. Metodologi pengembangan dimulai dengan survei, analisis dan rekayasa dalam perencanaan perangkat lunak. Membentuk sebuah struktur tim proyek yang baik sama pentingnya dengan memilih anggota tim yang berpengalaman di bidangnya. Melihat bahwa konstruksi dan implementasi sistem berbasis teknologi informasi dilakukan menggunakan pendekatan manajemen proyek, maka project manager adalah orang yang paling bertanggung jawab terhadap kinerja tim. Project Manager inilah yang akan memimpin timnya dalam merencanakan dan mengembangkan proyek teknologi informasi sesuai dengan target yang diharapkan. Terdapat tiga bidang yang harus diperhatikan dalam setiap proyek sistem informasi, yaitu: system informasi, teknologi informasi, dan manajemen informasi.

Kata kunci: *proyek, manajemen proyek, proyek perangkat lunak*

PENDAHULUAN

Perencanaan proyek perangkat lunak adalah manajemen proyek yang berfokus kegiatan mengembangkan *software*. Sifat manajemen proyek haruslah memenuhi hal-hal ini: (1) Menyelesaikan masalah; (2) Mengerjakan sesuatu hingga selesai; (3) Memiliki batas waktu mulai dan selesainya; (4) Membutuhkan resource/sumber daya dan waktu; (5) Bagi beberapa orang merupakan kesempatan (*opportunity*) dan menarik.

Manajemen itu berupa persiapan pekerjaan, pelaksanaan rencana, pengendalian proyek tersebut dan terakhir penutupan proyek dengan sebuah kesimpulan, yaitu sukses. Tahapan dalam mengembangkan proyek perangkat lunak (Gambar 1) terdiri dari: (1) *Initiating* – proyek sedang dalam proses untuk dipilih/disetujui, disponsori, didanai, dan diluncurkan; (2) *Planning* – perencanaan adalah proses yang berulang (perhatikan gambar). Perencanaan pada dasarnya menggambarkan proses bagaimana proyek akan dilaksanakan hingga selesai; (3) *Executing* – setelah proyek direncanakan, tim proyek memulai pekerjaannya; (4) *Controlling* – selama tim proyek mengerjakan tugasnya, project manager mengontrolnya; (5) *Closing* – setelah proyek diselesaikan, project manager akan menutup proyek *software*.



Gambar 1. Tahapan-tahapan proyek.

Banyak proyek gagal di awal, bukan di akhir. Artinya, persiapan adalah bagian yang sangat penting bagi proyek *software*. Persiapan diwujudkan dalam bentuk perencanaan proyek.

Tujuan Perencanaan Proyek

Perencanaan proyek Rekayasa Perangkat Lunak dari berbagai sudut pandang kurang lebih memiliki tujuan sebagai berikut: (1) Bagi Project Manager: menggambarkan status proyek kepada manajer senior dan *Stakeholder* dan merencanakan aktivitas tim proyek; (2) Bagi anggota tim proyek: memahami konteks pekerjaan; (3) Bagi manajer senior: memastikan apakah biaya dan waktu yang dialokasikan masuk akal dan terkendali dan melihat apakah proyek dilaksanakan secara efisien dan *cost effective*; (4) Bagi *Stakeholder*: memastikan apakah proyek masih berada pada jalurnya dan memastikan kebutuhan mereka sedang diakomodir oleh proyek.

Perencanaan proyek perangkat lunak membahas berbagai tindakan atau pekerjaan yang perlu dilakukan oleh semua yang terlibat di dalam proyek, termasuk dokumen-dokumen yang sebaiknya dibuat. Dokumen Perencanaan Proyek Rekayasa Perangkat Lunak terdiri atas sub-sub dokumen seperti *Vision and Scope*, *Statement of Work*, *Resource List*, *Work Breakdown Structure*, *Project Schedule*, dan *Risk Plan*. Berikutnya tiap-tiap dokumen tersebut akan dibahas secara lebih terperinci.

Vision and Scope

Dokumen ini adalah hasil kerja pertama dari seorang project manager. Berikutnya dokumen ini akan menjadi *tool* utama bagi project manager untuk acuan bagi dokumen-dokumen dan proses-proses berikutnya. Dokumen *Vision and Scope* yang baik dapat mencegah terjadinya masalah-masalah yang dapat memakan biaya yang besar. Dengan menunjukkan dokumen ini, baik kepada *Stakeholder* maupun anggota tim proyek, diharapkan pemahaman yang sama tentang proyek yang sedang berjalan dapat diraih. Dokumen ini dapat dibagi menjadi dua bagian.

Pertama, *problem statement* yang terdiri atas empat sub bab, antara lain: (1) Latar belakang proyek. Sub bab ini menceritakan dengan cukup mendalam baik latar belakang masalah maupun penjelasan mengenai mengapa organisasi memutuskan untuk membangun *software* untuk mengatasi masalah tersebut. Pada sub bab ini diceritakan sebab munculnya masalah, sejarah organisasi dengan permasalahan tersebut dan mengapa akhirnya diputuskan untuk membangun perangkat lunak yang diproyekkan; (2) *Stakeholder*. Pada sub bab ini akan diberikan daftar *stakeholder* yang dilibatkan dalam proyek. Mulai dari *customer* hingga manajer-manajer senior. *Stakeholder* ini bisa berupa nama atau jabatan. Tim proyek harus paham dengan siapa mereka bekerja dan apa bidang kerja mereka. Daftar juga dilengkapi dengan alasan dicantulkannya *stakeholder* tersebut. Untuk proyek-proyek besar tentu saja pencantuman nama tidak relevan karena akan menjadi terlalu panjang daftarnya; (3) Pengguna. Sub bab ini berisi daftar calon pengguna *software*. Sama dengan *Stakeholder*, bisa berupa nama atau jabatan. Daftar juga dilengkapi dengan alasan dicantulkannya pengguna tersebut; (4) Resiko. Sub bab ini akan diisi dengan faktor-faktor yang mungkin menjadi pemicu munculnya masalah, seperti keterlambatan dan permasalahan lain. Resiko yang dimaksud pada sub bab ini bisa faktor internal maupun eksternal.

Bagian kedua adalah *Vision of the Solution* yang terdiri atas empat sub bab, yaitu: (1) *Vision statement*. Tujuan *vision statement* adalah menggambarkan apa yang ingin dicapai setelah proyek berjalan. Di dalam sub bab ini disebutkan faktor-faktor apa yang harus terpenuhi untuk menandakan kapan proyek dinyatakan selesai. Selain itu tujuan dari proyek juga harus jelas disebutkan di dalam sub bab ini. Waktu terbaik untuk membuat *vision statement* adalah setelah tim melakukan proses *Requirement Engineering*. Gambaran produk yang ingin dicapai tersebut akan menjadi batasan ruang lingkup (*scope*) yang harus diperhatikan oleh semua pihak yang terlibat di dalam project. Ruang lingkup ini membutuhkan biaya dan waktu tertentu. Project manager yang baik akan mempersembahkan *software* tepat seperti yang telah dijanjikan kepada *Stakeholder* dan *user*, tepat pada waktunya dengan menghabiskan biaya (menerima bayaran) tepat sama dengan perjanjiannya dengan customer. Mungkin ada pendapat bahwa memberikan sedikit bonus fungsi terhadap *software*, dengan asumsi bahwa *Stakeholder* atau *user* akan menyukainya, tapi pendapat itu adalah kesalahan. Antara ruang lingkup, waktu dan biaya/harga harus ada keseimbangan. Jika ada penambahan pada ruang lingkup, maka seharusnya ada penambahan waktu atau biaya, jika tidak maka akan menyebabkan ketidakadilan bagi tim proyek/pengembang. Begitu juga sebaliknya. Perubahan ruang lingkup mestinya diatur dengan *Change Control System*; (2) Daftar fitur. Sebuah paket *software* umumnya dapat dibagi-bagi menjadi beberapa fitur. Jumlah yang umumnya dapat diterima adalah sekitar sepuluh fitur. Jumlah ini sudah cukup menggambarkan kompleksitas *software* namun tetap nyaman dibaca oleh tim pengembang. Tiap fitur sebaiknya ditulis dalam paragraph yang terpisah atau dalam bentuk pointer-pointer. Deskripsi fitur-fitur ini tidak perlu sangat detil, cukup beberapa kalimat yang menggambarkan penjelasan umum tentang fitur tersebut. Fitur-fitur ini mungkin mengalami

penambahan atau pengurangan, sesuai dengan permintaan *Stakeholder*. Jika perlu, sebuah fitur dapat dilengkapi dengan use case. Namun tentu saja use case dibuat agar cukup simpel untuk dipahami oleh semua *Stakeholder*; (3) Ruang lingkup tiap fase (jika perlu). Terkadang *deadline* yang diberikan untuk mengerjakan sebuah proyek *software* membuat pengerjaan seluruh fitur yang diajukan tidak mungkin selesai. Oleh karenanya dibuat solusi untuk membagi *software* menjadi beberapa fase rilis. *Software* akan dirilis pada saat *deadline* tercapai, namun dengan fitur yang dikurangi. Sedangkan rilis berikutnya lah yang memuat semua fitur; (4) Fitur-fitur yang ada pada rilis awal seharusnya adalah fitur yang sifatnya lebih penting daripada fitur lainnya, menurut *Stakeholder*. Hal semacam ini perlu dikonsultasikan kepada tim pengembang. Fitur yang tidak akan dibuat terkadang *Stakeholder* meminta fitur yang memang harus dibuang/ditinggalkan karena tidak masuk akal untuk diselesaikan dalam waktu yang tersedia, atau karena sebab-sebab lain. Fitur-fitur semacam ini perlu dicantumkan pada sub bab ini. Ini dicantumkan untuk diketahui semua pihak agar ada kesepahaman dan agar semua setuju dengan penghapusan fitur ini.

Statement of Work

Statement of Work (SOW) adalah dokumen yang menggambarkan semua produk yang akan dihasilkan selama proyek berjalan dan siala yang akan mengerjakannya. Secara lebih detil, di dalam SOW akan dirinci: (1) Daftar fitur yang akan dibuat – jika *software* akan dirilis dalam fase-fase, fiturnya juga harus dibagi ke dalam fase-fase tersebut; (2) Deskripsi hasil kerja (work product: spesifikasi kebutuhan, *source code*, *test plan*, laporan *defect*, dll) yang akan dibuat; (3) Estimasi usaha setiap *work product* tersebut.

Estimasi dibutuhkan agar proyek dapat berjalan dan selesai tepat waktu. Project manager perlu membantu timnya untuk membuat estimasi yang tepat. Sebuah pendekatan perlu diambil untuk menyeragamkan teknik estimasi ini. Salah satu teknik estimasi yang dapat dipilih adalah Wideband Delphi. Berikut ini langkah-langkah di dalam Wideband Delphi.

Pertama, memilih tim estimasi. Project manager memilih seorang moderator dan tim estimasi yang terdiri atas 3 hingga 7 orang. Jika tim yang telah dipilih merasa bahwa dokumen Vision and Scope kurang memberikan informasi, maka project manager harus memperbaiki dokumen tersebut.

Kedua, *kickoff meeting*. Pada rapat ini, tim akan membuat sebuah Work Breakdown Structure dan mendiskusikan berbagai asumsi yang muncul. Langkah-langkah yang dapat dijadikan acuan ketika rapat berlangsung kurang lebih sebagai berikut: moderator menjelaskan metode Wideband Delphi - moderator mereview dokumen Vision and Scope dan dokumen-dokumen pendukungnya jika ada anggota tim yang belum membacanya – tim mendiskusikan produk yang akan dibuat dengan berbagai asumsinya – tim membuat 10 hingga 20 pekerjaan utama sebagai representasi pekerjaan level tertinggi pada WBS – tim mendiskusikan estimasi terhadap WBS (jam, minggu, halaman, dll.) tersebut hingga mendapatkan kata sepakat.

Ketiga, *individual preparation*. Setelah *kickoff meeting* tiap anggota berusaha mengestimasi tiap-tiap pekerjaan di dalam WBS secara mandiri. Tahapan ini disebut sebagai *Individual Preparation*. Sebelumnya, moderator mencatat semua asumsi dan WBS kemudian membagikannya kepada semua anggota tim. Format berikut ini bisa dijadikan acuan untuk mendokumentasikan Individual Preparation.

Keempat, *estimation session*. Pada rapat ini, anggota tim bersama-sama merevisi estimasi-estimasi yang telah dibuat hingga menemukan kata sepakat. Dokumen berikut dapat dijadikan acuan sebagai contoh untuk membuat dokumentasi selama Estimation Session. Kepada setiap anggota tim akan dibagikan dokumen semacam ini (yang kosong) untuk kemudian direvisi selama jalannya *estimation session*.

Resource List

Resource list adalah daftar *resource* ‘sumber daya’ yang digunakan selama proyek berlangsung. Daftar ini berisi apa saja yang dibutuhkan berdasarkan jadwal proyek dengan mencantumkan deskripsi *resource* tersebut serta limit ketersediaan *resource* tersebut. Daftar semacam ini umumnya dapat dibuat menggunakan *software* manajemen proyek. Tetapi bisa juga dibuat dengan *worksheet* atau *word processor*. Setelah SOW dan *resource list* dibuat, seorang project manager harus membuat jadwal proyek (*project schedule*). Ini bisa dilakukan dengan urutan sebagai berikut: membuat Work Breakdown Structure – membuat estimasi usaha yang dibutuhkan oleh setiap pekerjaan pada *work breakdown structure* (WBS) – jadwal proyek dibuat dengan mengalokasikan *resource* dan waktu berdasarkan kalender untuk tiap pekerjaan pada WBS.

Jika WBS mengalami revisi (setelah melakukan estimasi, misalnya), seperti penambahan, perubahan atau penghapusan pekerjaan, revisi ini harus tercatat di dalam dokumen Project Plan dengan disertai dengan keterangan waktu kapan dibuatnya perubahan tersebut.

Work Breakdown Structure

Work breakdown structure (WBS) berisi daftar pekerjaan yang jika diselesaikan akan menghasilkan *work product*. WBS menyebutkan: apa saja pekerjaan yang akan dilakukan, tipe-tipe *resource* yang dibutuhkan untuk bekerja, estimasi tiap elemen pekerjaan, dan identifikasi lokasi penyimpanan. Tetapi WBS tidak mencantumkan siapa yang mengerjakan pekerjaan-pekerjaan itu dan kapan pekerjaan itu akan diselesaikan.

Project Schedule

Project Schedule ‘jadwal proyek’ dibuat oleh project manager untuk mengatur manusia di dalam proyek dan menunjukkan kepada organisasi bagaimana pekerjaan (proyek) akan dilaksanakan. Ini adalah alat untuk memantau (bagi project manager) apakah proyek dan tim masih terkendali atau tidak. Project schedule berbentuk kalender yang dihubungkan dengan pekerjaan yang harus dikerjakan dan daftar *resource* yang dibutuhkan. Sebelum jadwal dibuat, WBS harus terlebih dahulu ada, jika tidak maka jadwal tersebut akan terkesan mengada-ada.

Untuk membuat project schedule, ada beberapa *software* yang bisa dijadikan pilihan. Pilihan *software* yang gratis dan open source antara lain: Open Workbench, dotProject, netOffice dan Tutos. Beberapa hal perlu diperhatikan ketika membuat *project schedule*, seperti: (1) Alokasi *resource* pada tiap pekerjaan – *resource* bisa berupa berbagai hal seperti manusia, barang, peralatan (komputer, proyektor, dll), tempat (ruang rapat, misalnya) atau layanan (seperti training atau tim pendukung out source) yang dibutuhkan dan mungkin ketersediaannya terbatas.

Bagaimanapun juga *resource* yang utama adalah manusia; (2) Identifikasi setiap ketergantungan – sebuah pekerjaan disebut memiliki ketergantungan jika melibatkan aktivitas, *resource* atau *work product* yang dihasilkan pekerjaan/aktivitas lain. Contoh: *test plan* tidak mungkin dilaksanakan selama *software* belum diimplementasikan/ditulis. Program baru dapat ditulis setelah *class* atau modul dibuat dan dideskripsikan pada tahapan desain. Tiap pekerjaan pada WBS perlu diberi nomor, dengan angka tersebut bergantung pada nomor pekerjaan syaratnya; (3) Pembuatan jadwal – tiap pekerjaan juga memiliki jangka waktu pekerjaan. Dengan demikian jadwal bisa dibuat.

Risk Plan

Risk plan adalah daftar resiko/masalah yang mungkin terjadi selama proyek berlangsung dan bagaimana menangani terjadinya resiko tersebut. Bagaimanapun juga ketidakpastian adalah musuh

semua rencana, termasuk rencana proyek. Terkadang ada saja waktu-waktu yang tidak menyenangkan bagi proyek, banyak kesulitan terjadi misalnya suatu *resource* tiba-tiba tidak tersedia. Oleh karenanya *risk plan* adalah persiapan terbaik menghadapi ketidakpastian.

Langkah-langkah berikut dapat menjadi acuan untuk mendapatkan *Risk Plan*: (1) Pembahasan resiko potensial. Project manager akan memimpin sebuah sesi/rapat untuk mengidentifikasi masalah-masalah yang mungkin akan muncul. Anggota tim akan dipancing untuk mengemukakan resiko-resiko yang terpikirkan. Project manager akan menuliskannya di papan tulis setiap ada yang mengemukakan pendapat yang relevan. Sedikit pendapat mungkin akan muncul pada awalnya, kemudian berlanjut dengan tanggapan yang susul-menyusul hingga akhirnya suasana mendingin sampai akhirnya pendapat terakhir diutarakan. Resiko yang dimaksud di sini adalah resiko spesifik. Jika suatu resiko dirasa belum spesifik maka project manager akan memancing agar permasalahan disampaikan secara lebih spesifik. Sumber masalah yang baik lainnya adalah asumsi-asumsi yang muncul ketika membuat *vision and scope* dan melakukan estimasi dengan metode Wideband Dephi; (2) Estimasi dampak tiap resiko/masalah. Tim akan memberikan *rating* untuk setiap resiko. Nilainya berkisar dari 1 (masalah dengan resiko kecil) hingga 5 (masalah dengan resiko besar, kemungkinan munculnya besar, mungkin menghabiskan biaya besar dan sulit untuk membereskannya); (3) Buat sebuah risk plan. Tim akan mengidentifikasi langkah-langkah yang akan di ambil untuk mengatasi masalah-masalah yang akan muncul tersebut, dimulai dari resiko bernilai 5.

Organisasi proyek sistem informasi

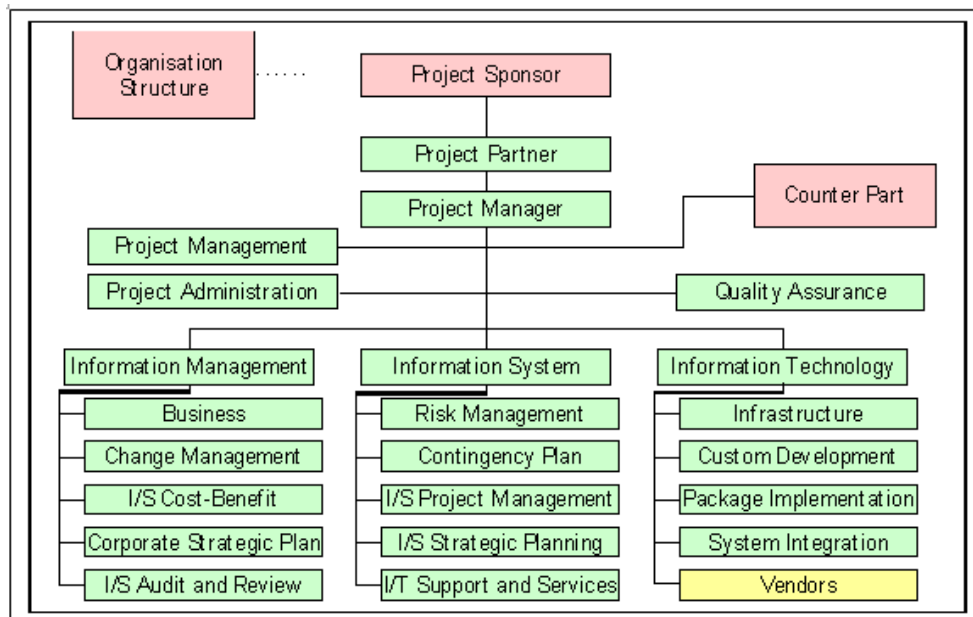
Pada kenyataannya, terdapat banyak jenis proyek sistem informasi. Dari yang bersifat strategis - seperti penyusunan buku biru pengembangan sistem informasi perusahaan, analisis kebutuhan sistem informasi, perencanaan strategis sistem informasi, manajemen resiko sistem informasi – sampai dengan yang sangat teknis – seperti instalasi infrastruktur jaringan, implementasi paket aplikasi yang dibeli di pasaran, pembuatan *software* untuk keperluan tertentu, audit teknologi informasi, pembangunan jaringan pengamanan sistem, dan sebagainya.

Sebelum sistem yang bersangkutan menjadi bagian integral dari perusahaan, biasanya sebuah tim dibentuk untuk melaksanakan sebuah proyek pengembangan sistem informasi. Dilihat dari sisi SDM-nya, struktur proyek sistem informasi melibatkan paling tidak tiga unsur: para karyawan dari divisi-divisi yang terkait dengan sistem informasi (Divisi Teknologi Informasi, Departemen EDP, Dinas Sistem Informasi, dan lain-lain), konsultan eksternal yang disewa khusus sebagai tenaga ahli dan spesialis, serta para vendor sebagai penyedia perangkat keras dan perangkat lunak.

Tim inti proyek sistem informasi dapat dikategorikan menjadi tiga bagian utama: Information Management, Information System, dan Information Technology.

Tim Information Management merupakan kumpulan para ahli manajemen yang sangat menguasai ilmu informasi untuk kebutuhan bisnis. Menganalisis dan menangani aspek-aspek bisnis dan manajemen (*relationship oriented*) dari suatu sistem informasi (relasi antara sistem informasi dengan fungsi-fungsi organisasi lainnya, hubungan antara sistem informasi dengan manajemen kunci perusahaan, dan lain sebagainya) merupakan tanggung jawab dari tim ini, dan tanggung jawab lainnya seperti: (1) Para ahli manajemen perubahan akan mengusulkan cara terbaik implementasi sebuah sistem baru sebagai pengganti sistem lama; (2) Pakar analisis *cost-benefit* akan melihat aspek-aspek informasi dari kaca mata pembiayaan investasi dan manfaat yang diberikan kepada perusahaan; (3) Ahli bisnis pada industri dimana perusahaan berada akan sangat mempercepat proses analisis dan meningkatkan kualitas output karena pengalaman yang telah dimiliki; (3) Pakar pembuatan perencanaan bisnis perusahaan akan menjaga agar proyek sistem informasi yang dijalankan sejalan dan tidak berada di luar jalur-jalur kerangka strategi perusahaan yang telah disepakati; (4) Ahli audit teknologi informasi akan menilai seberapa jauh efektivitas sistem yang ada sekarang dengan

kebutuhan bisnis, dan bagaimana keamanan sistem sehubungan dengan kontrol internal perusahaan; dan lain sebagainya.



Gambar 1. Organisasi tim proyek.

Tim Sistem Informasi akan memfokuskan diri pada jenis-jenis informasi apa yang dibutuhkan oleh perusahaan dalam menjalankan bisnis sehari-hari (*demand oriented*) dan hal-hal lain yang berkaitan dengannya, seperti: (1) Ahli strategi sistem informasi merupakan pakar utama yang dibutuhkan tim untuk mendefinisikan jenis-jenis informasi yang harus dihasilkan dengan segala karakteristiknya bagi perusahaan; (2) Pakar manajemen resiko akan melihat resiko bisnis yang mungkin dihadapi seandainya terjadi kesalahan atau hal-hal lain dalam sistem informasi yang dibangun; (3) Ahli *contingency planning* akan mempersiapkan prosedur yang harus dilakukan perusahaan jika ada gangguan teknis sistem yang dapat mengganggu aktivitas perusahaan; (4) Pakar manajemen proyek akan mempersiapkan strategi dalam menjalankan portfolio proyek sistem informasi yang beragam berdasarkan skala prioritas dan keterbatasan sumber daya; (5) Spesialisasi supports dan services yang akan mengajukan usulan bagaimana memelihara sistem informasi setelah proyek selesai dilaksanakan (pasca implementasi), termasuk di dalamnya kemungkinan outsourcing, usulan pembangunan infrastruktur khusus (call center, help desk, dan lain-lain), sampai dengan prosedur-prosedur yang harus dilakukan.

Fungsi terakhir adalah Information Strategy yang merupakan jawaban atas kebutuhan sistem informasi yang telah didefinisikan (*supply oriented*). Semua rencana strategis yang telah disusun oleh kedua tim sebelumnya pada akhirnya harus diimplementasikan secara teknis. Dilihat dari karakteristik pekerjaannya, biasanya ada empat jenis pengembangan teknologi yang dilakukan: (1) Membangun infrastruktur jaringan teknologi informasi, biasanya berupa proyek-proyek perencanaan dan pembangunan jaringan komunikasi seperti LAN (Local Area Network), WAN (Wide Area Network), Intranet, Internet, dan Extranet; (2) Membeli paket *software* atau *hardware* yang siap pakai di pasaran dan mengimplementasikannya dalam perusahaan (package implementation); (3) Membuat *software* sendiri berdasarkan kebutuhan dan desain yang dibuat oleh tim yang dibentuk secara khusus oleh perusahaan (custom development atau in-house development); atau (4) Melakukan dua atau ketiga hal diatas dan mengintegrasikannya satu sama lain (system integration).

Terlihat bahwa vendor-vendor teknologi informasi akan menjadi bagian dari tim ini karena merekalah yang akan menawarkan produk-produk *software* dan *hardware* (maupun komponen-komponen terkait lainnya) yang dibutuhkan dalam pengembangan sistem informasi.

Sebagai catatan, struktur yang dijelaskan ini merupakan suatu struktur yang ideal dan biasanya selalu dipergunakan untuk proyek-proyek sistem informasi berskala menengah ke atas. Untuk proyek-proyek kecil dan spesifik, beberapa fungsi dapat dihilangkan karena tidak begitu signifikan. Terlalu banyak SDM yang terlibat dalam suatu proyek sistem informasi belum tentu mempercepat pelaksanaan proyek, terlalu sedikit juga akan memiliki potensi mengurangi kualitas output.

METODE

Manajemen proyek perangkat lunak merupakan bagian yang penting dalam pembangunan perangkat lunak. Sekalipun tidak bersifat teknis seperti pengkodean, hal-hal dalam manajemen proyek ini mampu menentukan apakah proyek akan berjalan dengan baik sehingga menghasilkan produk yang baik. Hal-hal yang berkaitan dengan manajemen adalah pengelolaan personel dan koordinasi tim, proses, pengukuran proyek-termasuk menentukan harga dari perangkat lunak, penjadwalan dan sebagainya.

Metode dalam manajemen proyek meliputi survei untuk mengumpulkan data, melakukan perhitungan analisis terhadap hasil survei, dan merancang perencanaan proyek menggunakan *tools* yang sesuai.

HASIL DAN PEMBAHASAN

Manajemen Personel, Produk dan Proses

Manajemen proyek perangkat lunak mengatur empat hal penting: personel, produk, proses dan proyek. Empat hal ini berurutan mulai dari yang paling penting. Personel merupakan mendapat tempat paling penting karena tanpa personel yang baik dan tepat maka tiga hal lain tidak bisa berjalan dengan baik.

Kategori Personel

Proses pembangunan perangkat lunak melibatkan banyak personel. Personel-personel ini digambarkan seperti pemain, dan dikategorikan dalam lima kategori pemain, yaitu (1) manajer senior: yang menentukan usaha yang dikerjakan, dan pemegang keputusan dalam proyek; (2) manajer proyek (teknis) – pemimpin tim: yang membuat rencana, memotivasi, mengatur dan mengendalikan praktisi yang mengerjakan perangkat lunak; (3) praktisi: yang mengerjakan perangkat lunak; (4) klien: yang menentukan kebutuhan perangkat lunak dan pihak lain yang berkaitan dengan hasil produk; (5) pengguna perangkat lunak: yang berinteraksi langsung dengan perangkat lunak yang dibangun. Efektifitas kerja masing-masing personel di atas harus diusahakan oleh pemimpin tim. Pemimpin tim ini yang mengatur tim proyek agar dapat memberikan yang terbaik dari masing-masing personel.

Pemimpin Tim

Pemimpin tim perangkat lunak disini adalah Project Manager. Seorang pemimpin tim diharuskan mempunyai ketrampilan memimpin yang cukup. Seseorang tidak menjadi pemimpin tim

secara kebetulan tapi sungguh-sungguh karena punya kemampuan. Kemampuan yang dibutuhkan dalam kepemimpinan seperti: mampu memotivasi, mampu berorganisasi: mengatur proses yang ada atau membuat yang baru dalam rangka mewujudkan ide/konsep menjadi produk, mampu mendorong keluarnya ide-ide baru – memberi dorongan, menciptakan situasi yang kondusif untuk lahirnya ide baru, mencari penyelesaian masalah (problem solving) – menganalisis masalah-masalah teknis ataupun manajemen/organisasi kemudian mendapatkan jalan keluar atau memotivasi anggota untuk mampu menyelesaikan masalah. Akomodatif terhadap perubahan yang mungkin terjadi., mampu menjadi manajer – menggunakan wewenangnya pada saat yang tepat, atau memberikan kebebasan pada anggota timnya jika diperlukan, mampu menghargai kerja – menghargai hasil yang dicapai, ide yang dilontarkan dan pendapat yang diajukan oleh anggota timnya.

Tim Perangkat Lunak (Software Team)

Struktur organisasi dalam tim ini bisa mengadaptasi dari banyak struktur organisasi yang sudah ada. Berikut beberapa pilihan pembagian tugas/penugasan yang bisa diterapkan untuk tim perangkat lunak yang terdiri dari n personel yang bekerja selama k tahun:

n personel ditugaskan untuk sejumlah m tugas yang berbeda dengan sedikit tugas gabungan → koordinasi adalah tugas dari manajer yang mungkin saja punya enam proyek lainnya;
 n personel di tugaskan untuk sejumlah m tugas yang berbeda dengan $m < n$ sehingga terbentuk tim informal. Pemimpin tim khusus perlu ada → koordinasi antar tim adalah tanggung jawab manajer;
 n personel dibagi menjadi sejumlah t tim. Tiap tim ditugaskan mengerjakan satu atau lebih tugas. Tiap tugas mempunyai struktur yang ditentukan sebelumnya bagi semua tim → koordinasi dikendalikan oleh tim dan manager;

Sekalipun masing-masing pilihan punya argumentasi sendiri-sendiri, dari pengamatan yang dilakukan, pilihan no 3 dianggap lebih produktif.

Cara atau gaya manajemen, jumlah personel, tingkat kemampuan para personel dan masalah-masalah yang dihadapi tim menentukan bentuk struktur organisasi yang bisa diterapkan. Contoh-contoh struktur organisasi tim akan dijelaskan berikut ini. Contoh pertama adalah *democratic decentralized* (DD), dimana tidak ada pemimpin yang permanen, koordinator ditunjuk untuk jangka waktu yang pendek, keputusan diambil berdasarkan konsensus bersama, komunikasi horizontal antar anggota tim (posisi sejajar semua) → cocok untuk masalah yang sulit/rumit, cocok untuk proyek besar, tim cenderung awet dan bertahan lama, pekerjaan memuaskan, cocok untuk masalah yang modularitasnya rendah, perlu banyak waktu untuk menyelesaikan proyek. Kedua adalah *controlled decentralized* (CD), dimana pemimpin tim ditentukan, ada wakil pemimpin dan mereka berbagi tugas, penyelesaian masalah adalah tugas tim dan implementasinya dibagi di antara beberapa sub-tim oleh pemimpin, komunikasi horizontal di antara sub-tim dan di antara personel, komunikasi vertikal berdasarkan struktur hirarki → sentralisasi untuk penyelesaian masalah, cocok untuk masalah yang sederhana, cukup cocok untuk proyek besar, masalah dengan modularitas tinggi, menghasilkan sedikit kesalahan. Contoh ketiga adalah *controlled centralized* (CC), dimana penyelesaian masalah dikerjakan oleh pemimpin, pemimpin melakukan koordinasi internal tim, komunikasi lebih banyak vertikal antara pemimpin dan anggota tim → cocok untuk masalah yang sederhana, melakukan penyelesaian, masalah lebih cepat, masalah dengan modularitas tinggi, menghasilkan sedikit kesalahan.

Pengukuran Perangkat Lunak

Metrik dalam perancangan perangkat lunak didefinisikan oleh IEEE Glossary of SE sebagai “A quantitative measure of the degree to which a system, component, or process possesses a given attribute” yang artinya ‘pengukuran secara kuantitatif pada tingkat sistem, komponen atau proses berdasarkan katagori yang ditetapkan’.

Pengukuran Berdasarkan Ukuran

Pengukuran berdasarkan perangkat lunak-perangkat lunak yang sudah diproduksi/dibuat sebelumnya, lengkap dengan karakteristik lain seperti line of code (LOC), harga, waktu yang diperlukan pada tiap fungsi atau proyek yang dibangun, kesalahan (error) yang ditemukan. Dari total LOC, harga dan lama waktu dapat diperoleh misalnya harga per KLOC (seribu baris kode) dan kesalahan per KLOC.

| Functions | estimated LOC | LOC/pm | \$/LOC | Cost | Effort (months) |
|-----------|---------------|--------|--------|---------|-----------------|
| UICF | 2340 | 315 | 14 | 32,000 | 7.4 |
| 2DGA | 5380 | 220 | 20 | 107,000 | 24.4 |
| 3DGA | 6800 | 220 | 20 | 136,000 | 30.9 |
| DSM | 3350 | 240 | 18 | 60,000 | 13.9 |
| CGDF | 4950 | 200 | 22 | 109,000 | 24.7 |
| PCF | 2140 | 140 | 28 | 60,000 | 15.2 |
| DAM | 8400 | 300 | 18 | 151,000 | 28.0 |
| Totals | 33,360 | | | 655,000 | 145.0 |

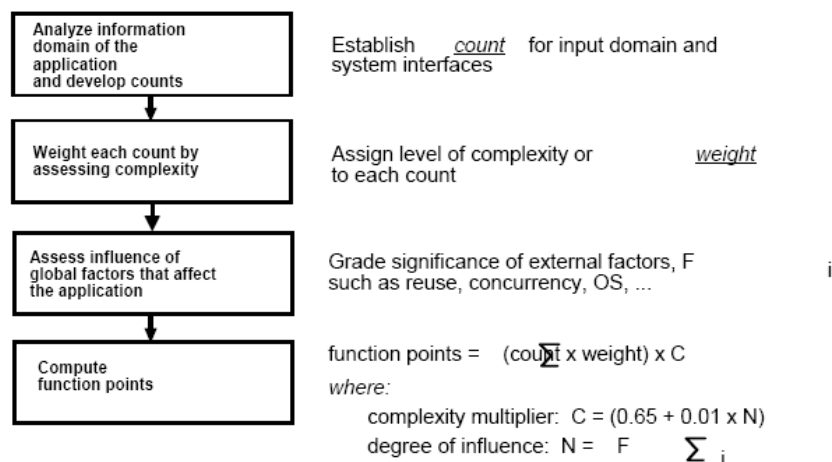
Gambar 2. Contoh tabel pengukuran berdasarkan ukuran.

Cara ini kurang diterima secara universal karena penggunaan LOC untuk kunci ukuran bergantung pada bahasa pemrograman yang digunakan.

Pengukuran Berdasarkan Fungsi (*Function Point – FP*)

Function point (Gambar 3) ditentukan berdasarkan bagian-bagian perangkat lunak yang bisa dihitung (Gambar 4), seperti: (1) Jumlah input dan output dari/untuk pengguna; (2) Jumlah *user inquiry*: *inquiry* didefinisikan sebagai online input yang menghasilkan respon langsung dari *software* dalam bentuk online output; (3) Jumlah *file*: *file* yang terpisah database, bagian *file*; (4) Jumlah external interface: misalnya data *file* pada *storage* media yang digunakan untuk mengirimkan informasi ke sistem lain.

Yang kurang jelas dalam proses ini dan kurang detail adalah bagaimana menentukan berat (*weight*).



Gambar 3. Alur pengukuran dengan *Function Point*.

| measurement parameter | count | weighting factor | | | = | □ |
|--------------------------|-------|------------------|------|---------|---|---|
| | | simple | avg. | complex | | |
| number of user inputs | □ | X 3 | 4 | 6 | = | □ |
| number of user outputs | □ | X 4 | 5 | 7 | = | □ |
| number of user inquiries | □ | X 3 | 4 | 6 | = | □ |
| number of files | □ | X 7 | 10 | 15 | = | □ |
| number of ext.interfaces | □ | X 5 | 7 | 10 | = | □ |
| count-total | → | | | | | □ |
| complexity multiplier | → | | | | | □ |
| function points | → | | | | | □ |

Gambar 4. Penghitungan Function Point.

Berikut adalah faktor-faktor yang dianggap penting berdasarkan skala prioritas 0 (tidak penting) sampai 5 (sangat penting): *data communications, distributed functions, heavily used configuration, transaction rate, online data entry, end user efficiency, online update, complex processing, installation case, operational case, multiple sites dan facilitate change.*

Ukuran untuk Organisasi Kecil *Defect Removal Efficiency (DRE)*

Untuk organisasi yang kecil mungkin bisa menggunakan ukuran seperti: (1) Waktu (hari atau jam) mulai dari permintaan/request sampai evaluasi lengkap → *tqueue*; (2) Usaha (personel-waktu) untuk melakukan evaluasi → *Weval*; (3) Waktu (jam atau hari) dari selesainya evaluasi sampai penugasan lain ke personel → *teval*; (4) Usaha (personel – jam) yang dibutuhkan untuk membuat perubahan → *Wchange*; (5) Waktu (jam atau hari) untuk melakukan perubahan, → *tchange*; (6) Kesalahan yang terjadi selama pengerjaan untuk melakukan perubahan → *Echange*; (7) Cacat yang terjadi setelah perubahan diserahkan ke klien → *Dchange*.

Setelah ukuran-ukuran tersebut dikumpulkan beberapa hal bisa dihitung seperti total waktu dari permintaan perubahan sampai implementasi dari perubahan. Persentase usaha yang dibutuhkan untuk evaluasi dan implementasi bisa ditetapkan. *Defect Removal Efficiency (DRE)* bisa dihitung dengan: $DRE = Echange / (Echange + Dchange)$.

PENUTUP

Sebuah proyek perangkat lunak perlu di kelola dengan manajerial baik. Langkah-langkah dalam manajemen dimulai dari persiapan pekerjaan, pelaksanaan rencana, mengendalikan proyek tersebut dan terakhir menutup proyek dengan sebuah kesimpulan, yaitu sukses. Secara lebih sistematis, tahapan-tahapan proyek terdiri dari *initiating, planning, executing, controlling dan closing.* Kegagalan proyek di awal, bukan di akhir disebabkan oleh persiapan yang kurang baik. Oleh karenanya, bagian perencanaan awal ini adalah bagian yang sangat penting bagi proyek *software.* Persiapan diwujudkan dalam bentuk perencanaan proyek.

DAFTAR PUSTAKA

Cash, J. I., Eccles, R. G., Nohria, N., Nolan, R.L. (1994). *Building the Information-Age Organization: Structure, Control, and Information Technologies.* Boston: Richard D. Irwin.

Leicester University. (1997). *Handbook of Management Information System Development*. Leicester, UK: Learning Resources.

Pressman, Roger. S. (2001). *Software Engineering: A Practitioner's Approach* (5th ed.). New York: McGrawHill.