

EVALUASI MANAJEMEN PROYEK PEMBUATAN PIRANTI LUNAK BERBASIS WEB PADA PT. MEDIA PLUS JAKARTA

Budi Tjahjono

Fasilkom – Universitas INDONUSA Esa Unggul, Jakarta

Jl. Arjuna Utara Tol Tomang Kebun Jeruk, Jakarta 11510

budi.tjahyono@indonusa.ac.id

Abstrak

Perkembangan perusahaan *software house* di Indonesia semakin meningkat dari segi jumlah dan kemampuan. Hal ini menunjukkan persaingan antar perusahaan tersebut juga semakin ketat. Kemampuan yang lebih baik dari sebuah *software house* dalam mengelola proyek piranti lunak untuk memenuhi kebutuhan *user* menjadi nilai tambah bagi perusahaan. Kebutuhan akan aplikasi *web* bagi perusahaan klien yang ingin memperkenalkan dan/atau mengembangkan usahanya semakin meningkat. Oleh karena itu, diperlukan penerapan metodologi yang tepat untuk mendukung pengelolaan proyek piranti lunak, khususnya yang berbasis web. Metodologi penelitian yang dipergunakan adalah studi pustaka dengan studi kasus di PT Media plus. Teknis analisis data yang dipakai adalah *Critical Path Method* (CPM), *Staged Delivery* dan *Joint Application Development* (JAD) yang saling mendukung satu sama lain. Ketiga teknik tersebut kemudian dipergunakan sebagai metodologi bagi pengelolaan proyek piranti lunak. Adapun hasil penelitian menunjukkan bahwa PT Media plus telah mampu merencanakan aktivitas-aktivitas proyek dengan *work breakdown structure* dan menggunakan *Microsoft Project* sebagai *tool* untuk estimasi jadwal aktivitas proyek. Akan tetapi masih ada pelaksanaan aktivitas-aktivitas yang mengalami kemunduran. Untuk itu, perusahaan dapat mengembangkan metodologi yang lebih baik untuk meningkatkan ketepatan penyelesaian proyek agar perusahaan dapat menunjukkan nilai tambah kepada *user* atau klien dari segi waktu. Perusahaan disarankan untuk menerapkan *Staged Delivery* dan JAD serta memanfaatkan CPM untuk mendukung pengelolaan proyek piranti lunak.

Kata Kunci: Pengelolaan, Proyek piranti lunak, *Web*, *Staged Delivery*, *Critical Path Method*, *Joint Application Development*

Pendahuluan

Di Indonesia, dunia industri terus berkembang dan persaingan di antara perusahaan-perusahaan komersial baik di bidang produk maupun jasa semakin ketat. Ini dapat dilihat dari hampir setiap perusahaan baik perusahaan di bidang produk maupun jasa terlibat dalam banyak proyek. Bahkan beberapa proyek bergengsi ditenderkan. Perusahaan-perusahaan maju saling berlomba untuk memenangkan tender tersebut. Perusahaan-perusahaan yang ingin ikut serta berkompetisi harus berusaha menunjukkan nilai tambah mereka untuk meraih lebih banyak keuntungan dan berpotensi menjadi pemimpin pasar.

Perkembangan bisnis ini jelas sangat berpengaruh pada investasi dari para investor untuk melakukan penelitian dan pengembangan

dalam berbagai hal mengenai bisnis di Indonesia. Semakin banyak melakukan penelitian dan pengembangan, maka pengertian akan bisnis, analisis terhadap bisnis tersebut, dan prediksi perkembangannya akan semakin tajam dan semakin dimengerti. Bila pengertian sudah terjadi dengan baik, biasanya perusahaan akan mulai mengaitkannya dengan *tools* (alat kerja yang efektif dan efisien) dan sumber daya manusia yang berkualitas untuk bersaing dengan pasar domestik ataupun dengan pasar internasional. Dengan demikian, piranti lunak untuk membantu mereka menjadi lebih efektif dan efisien akan sangat dibutuhkan, sehingga dengan sendirinya *software house* akan semakin berkembang.

Dari segi sumber daya manusia (SDM), Indonesia memang bukanlah negara yang

dikenal sebagai penghasil SDM yang unggul di bidang teknologi piranti lunak di dunia. Tetapi cukup banyak para *developer* (dulu disebut *programmer*) yang berasal dari Indonesia (bekerja di Indonesia maupun di luar negeri) yang menjadi *developer* penting dalam banyak produk piranti lunak yang bagus dan mempunyai posisi baik di dunia internasional.

Dengan demikian, dapat dikatakan bahwa potensi SDM Indonesia tidak jelek sama sekali, tetapi juga tidak sangat baik. Keberadaan *software house* bukanlah hanya berkaitan dengan potensi sumber daya manusia, melainkan dengan banyak hal, seperti pengertian akan bisnis yang akan dikomputerisasikan, teknik-teknik unggulan dalam menata dan mengatur rencana serta pelaksanaan proyek-proyek sistem informasi.

Tuntutan perusahaan yang membutuhkan sistem informasi yang menunjang proses bisnis juga terus bertambah. Kebutuhan-kebutuhan bisnis juga bisa berubah kapan saja, terutama saat ini kebutuhan bisnis yang sedang populer adalah penggunaan *web site* sebagai alat bantu memperkenalkan perusahaan ke masyarakat luas. Hal ini menjadi suatu tantangan bagi *software house* untuk menghasilkan sistem atau aplikasi yang baik dan sesuai bagi kebutuhan perusahaan. Oleh karena itu, ilmu manajemen proyek piranti lunak yang baik menjadi kunci keberhasilan pembuatan piranti lunak bagi perusahaan-perusahaan yang membutuhkan. Pengelolaan yang dikenal sebagai "*project management*" adalah salah satu cara yang ditawarkan untuk keberhasilan suatu proyek. Manajemen proyek tidak hanya melihat dari sisi kebutuhan bisnis perusahaan saja. Manajemen proyek teknologi informasi atau sistem informasi melihat dari berbagai segi seperti integrasi, jadwal, biaya, sumber daya manusia, kualitas, komunikasi, resiko proyek dan lain-lain.

Melalui penelitian ini, penulis akan mengadakan evaluasi dan usulan pengelolaan proyek pembuatan piranti lunak berbasis *web* melalui studi kasus di PT. Media Plus dengan harapan dapat memberikan masukan pengelolaan proyek pembuatan piranti lunak yang tepat waktu sehingga pada akhirnya dapat

memberikan dampak positif bagi perkembangan *software house* itu sendiri

Permasalahan

Adapun beberapa masalah yang dapat diidentifikasi oleh perusahaan dalam hal manajemen proyek piranti lunak antara lain:

1. Bagaimana mengelola beberapa proyek pada saat yang hampir bersamaan dengan sumber daya yang tersedia dan terbatas?
2. Apa model yang tepat yang digunakan untuk mengelola proyek-proyek tersebut supaya dapat selesai tepat waktu dan memenuhi kebutuhan *user*?
3. Bagaimana mengatasi masalah perubahan kebutuhan *user* yang sering terjadi dalam pengelolaan proyek?

Tujuan dan Manfaat Penulisan

Tujuan penelitian yang dilakukan oleh penulis adalah sebagai berikut:

1. Penulis mampu mengelola beberapa proyek, mulai dari tahap *definition* sampai tahap *operation* (proyek siap untuk serah terima) dengan baik.
2. Penulis juga mengestimasi jadwal, sumber daya, dan masalah yang mungkin timbul.
3. Penulis juga membuat penjadwalan untuk membantu pengelolaan proyek-proyek tersebut dan mengurangi masalah yang mungkin terjadi
4. Penulis juga menerapkan metode JAD yang nantinya akan dipakai dalam tahapan requirement.
5. Membantu perusahaan dalam memasarkan produk-produknya dan menarik pelanggan baru .

Tinjauan Teori Sejarah Internet

Menurut Hahn (1996), *internet* adalah nama sebuah sistem yang sangat luas, mencakup seluruh dunia yang terdiri dari orang-orang, informasi, dan komputer. *Internet* sangatlah luas dan kompleks untuk dapat dipahami secara detil oleh seorang manusia. Orang yang ahli di bidang *internet* belum tentu memahami secara keseluruhan apalagi orang yang belum mengenal sama sekali tentang *internet*.

Internet adalah sebuah istilah untuk sistem *world-wide* yang sangat luas. Sistem tersebut terdiri dari berjuta-juta jaringan komputer yang dapat dihubungkan dengan kabel, saluran telepon, sambungan satelit, dan sebagainya yang keseluruhan alat itu bertujuan untuk menghubungkan antara *client* dan *server* supaya dapat berkomunikasi satu dengan yang lainnya (Hahn,1996).

Menurut Ellsworth (1997), *Internet* merupakan jaringan besar yang dibentuk oleh interkoneksi jaringan komputer-komputer tunggal di seluruh dunia lewat saluran telepon, satelit, dan sistem telekomunikasi lainnya.

Jadi *internet* dapat dianggap sebagai suatu jaringan di mana para peserta dapat saling berkomunikasi dan berbagi informasi di manapun mereka berada. Komputer yang terhubung dalam *internet*, disebut *host* atau *node*. Komputer tersebut dapat berupa sebuah sistem komputer *mainframe*, mini komputer di sebuah laboratorium, *workstation*, komputer pribadi, bahkan juga sebuah *laptop*. *Host* ini saling berhubungan dalam suatu jaringan raksasa yang mencakup hampir seluruh dunia.

Dengan menggunakan *internet* berarti kita duduk di depan layar komputer dan menggunakan sebuah program untuk menampilkan beberapa tugas. Ini dapat dilakukan di kantor, sekolah, rumah dengan menggunakan berbagai tipe komputer, seperti sebuah PC atau sebuah Macintosh (Hahn, 1996).

Proyek

Menurut Rakos (1990) proyek adalah sekumpulan aktivitas yang menghasilkan suatu *deliverable* atau serahan. Proyek selalu dimulai dengan sebuah masalah, kemudian *user* meminta tim proyek untuk memberikan solusi atas masalah yang ada, solusi tersebut adalah hasil dari proyek.

Proyek (Anonim, 1996) adalah suatu usaha dalam waktu tertentu (sementara) yang diambil untuk menciptakan produk atau jasa yang unik. Proyek memiliki awal dan akhir yang jelas. Proyek dilaksanakan pada semua level organisasi. Proyek bisa melibatkan satu atau ribuan orang, satu unit tunggal dari suatu organisasi atau pun antar unit dari organisasi yang berbeda.

Menurut Olson (2004), suatu proyek melibatkan aktivitas yang baru dan kompleks, tujuan yang dapat didefinisikan melintasi berbagai level organisasi dan merupakan aktivitas yang unik. Karena proyek melibatkan banyak aktivitas, maka proyek biasanya mengandung tingkat ketidakpastian dan resiko yang tinggi. Oleh karena itu pula, tingkat sumber daya yang diperlukan untuk penyelesaian proyek, biasanya sulit untuk diestimasi.

Definisi Manajemen Proyek

Manajemen proyek (Anonim, 1996) adalah penerapan dari pengetahuan, keahlian, alat-alat (*tool*) dan teknik pada aktivitas proyek guna memenuhi atau mencapai kebutuhan dan harapan *stakeholder* dari suatu proyek.

Stakeholder utama dalam setiap proyek meliputi:

- *Project Manager*: individu yang bertanggung jawab untuk mengatur proyek.
- *Customer /client*: individu atau organisasi yang akan menggunakan produk proyek tersebut.
- Organisasi : perusahaan di mana karyawannya secara langsung terlibat dalam pelaksanaan proyek.
- Sponsor: individu atau *group* dalam organisasi yang menyediakan sumber daya keuangan untuk proyek tersebut.

Pemenuhan atau pencapaian kebutuhan *stakeholder* meliputi keseimbangan permintaan yang bersaing antara:

- Cakupan, waktu, biaya dan kualitas
- *Stakeholder* dengan kebutuhan dan harapan yang berbeda
- Permintaan yang teridentifikasi (kebutuhan) dan permintaan yang tidak teridentifikasi (harapan).

Manajemen proyek yang dibahas dalam tulisan ini adalah manajemen proyek yang berhubungan dengan pembuatan piranti lunak. Jadi manajemen proyek piranti lunak adalah penerapan dari pengetahuan, keahlian, alat-alat (*tool*) dan teknik pada aktivitas pembuatan aplikasi komputer guna memenuhi atau mencapai kebutuhan dan harapan *stakeholder* dari suatu proyek.

Tahapan Pengelolaan Proyek Piranti Lunak

Menurut Rakos (1990), terdapat tujuh tahapan dalam manajemen proyek piranti lunak. Tujuh tahapan itu adalah:

1. Definition

Tujuan dari tahapan ini adalah untuk memperoleh pemahaman terhadap masalah *user* guna mengestimasi biaya dan waktu. Tiga aktivitas utama dalam tahapan ini adalah:

- a. Memperoleh pemahaman yang baik mengenai masalah *user* dan apa yang dibutuhkan untuk memecahkan masalah tersebut.
- b. Memutuskan apakah proyek tersebut diumumkan atau tidak. Seorang *Project Manager* (PM) harus memastikan bahwa suatu proyek adalah layak secara teknis dan memiliki kesempatan untuk berhasil sebelum PM memutuskan untuk mengambil proyek itu.
- c. Jika proyek dinyatakan berjalan, PM harus menganalisa semua resiko yang mungkin timbul dari proyek tersebut.

PM harus menulis beberapa dokumen dan mencapai beberapa *milestone* pada akhir tahapan ini. Pertama, PM harus menulis *Requirement Document* yang jelas dan lengkap supaya tim proyek dapat memahami masalah *user* secara penuh dan mengestimasi biaya dari solusi tersebut. Bahasa yang digunakan dalam *Requirement Document* berorientasi pada bisnis dan dimengerti oleh *user*. *Requirement Document* dapat diperoleh dengan cara mewawancarai langsung *user* yang terlibat dalam proyek. *Milestone* pertama adalah persetujuan (*approval* atau *sign off*) *Requirement Document* oleh *user* dan tim proyek. Kedua, PM harus menyiapkan *Preliminary Project Plan* (PPP) yang menjadi langkah awal dari perencanaan untuk mengembangkan produk dan sumber daya yang dibutuhkan untuk masing-masing langkah. Rencana ini menggarisbesarkan seberapa lama sumber daya dan berapa besar biaya yang dibutuhkan. Ketiga, PM harus memberikan estimasi tersebut dalam bentuk proposal kepada *user*.

2. Analysis

Tujuan dari tahapan ini adalah mendefinisikan secara tepat apa yang akan dilakukan oleh sistem bagi *user* dan bagaimana sistem tersebut akan cocok atau sesuai dengan lingkungan *user*. Aktivitas utama dan sekaligus menjadi *milestone* dari tahapan ini adalah membuat *Functional Specification* yang mendefinisikan *behavior* dari sistem dalam bentuk narasi dan gambar. Contohnya, *user interface* yang terdapat menu-menu, perintah (*command*), tanggapan (*responses*), laporan (*report*), dan pesan (*message*) dirincikan serinci mungkin. Semua kebutuhan tampilan harus ada termasuk perubahan dalam lingkungan *user* dikarenakan kebutuhan baru juga dijelaskan semua *deliverables* (hal-hal yang bisa diberikan kepada *user*) meliputi perangkat keras, piranti lunak, pelatihan (*training*), dokumentasi dan garansi (*warranty*) dirincikan. *Functional Specification* adalah segala sesuatu yang dapat sistem lakukan bagi *user*. *Functional Specification* mungkin akan dinegosiasikan dan direvisi. Pada saat disetujui, *Functional Specification* harus ditandatangani oleh kedua belah pihak yaitu *user* dan PM.

3. Design

Aktivitas utama dari tahapan ini adalah menciptakan level atas dan menengah dari desain sistem dan mendokumentasikannya dalam *Design Specification*. Aktivitas kedua adalah memulai *Acceptance Test Plan* (ATP) yang merupakan suatu dokumen yang merincikan pengujian-pengujian yang akan digunakan untuk mendemonstrasikan semua fungsi sistem kepada *user* pada tahap *Acceptance*. *Milestone* utama yang dicapai dari tahapan desain adalah pada saat *Design Specification* ditinjau ulang (*review*) dan dinyatakan bebas dari kesalahan (*error-free*).

Perancangan suatu sistem piranti lunak terdapat dua langkah yaitu membagi sistem ke dalam *Functional Component* dan kemudian menghubungkan komponen-komponen itu. Pada akhir tahapan desain, tim proyek telah mencapai beberapa *milestone* yaitu *Design Specification Document* dan kewajiban ATP. ATP *checklist* meliputi *Functions vs Test Table*, pemenuhan semua *Functional Specification*,

pendefinisian *setting* pengujian, kesadaran *user* dan tim proyek akan tinjauan ulang (*review*) ATP, perbaikan (revisi) ATP jika diperlukan, dan ATP harus ditandatangani oleh *user*. *Milestone* lain yang dicapai adalah rencana proyek terutama estimasi waktu dan biaya yang ditinjau ulang. Tahapan desain memberikan perkiraan perhitungan jumlah modul dan kompleksitas untuk masing-masing modul.

4. Programming

Aktivitas pada tahapan ini adalah menulis program. *Milestone* yang dicapai adalah program yang diuji, *Rencana Pengujian Sistem (System Test Plan)*, dan awal dokumentasi bagi *user*. Beberapa hal yang perlu diperhatikan (*checklist*) sebelum menulis program adalah:

- Apakah tinjauan ulang desain (*design reviews*) memerlukan kerja ulang (*rework*)? Kalau ya, waktu harus dijadwalkan lagi dan *programming* harus ditunda.
- Apakah sumber daya yang sudah direncanakan termasuk *programmer* masih tersedia?
- Apakah *programmer* yang terlibat sudah diberikan pelatihan seperti penggunaan sistem operasi, bahasa pemrograman, program paket dan alat-alat pemrograman yang akan digunakan?
- Apakah lingkungan pemrograman baik? *Programmer* membutuhkan piranti lunak pengembangan dan alat pemrograman yang mudah digunakan, komputer yang menyediakan respon cepat, tersedia saat dibutuhkan, dan harus dapat diandalkan.

Menurut Rakos (1990), langkah-langkah *programming* adalah sebagai berikut:

1. Merencanakan integrasi atau kesatuan dari program yang ditulis
2. Merancang modul
3. Menelusuri desain modul untuk memastikan bahwa desain yang dibuat adalah yang terbaik dimana semua fungsi telah dimasukkan
4. Merencanakan bagaimana cara untuk menguji modul
5. Meng-*coding* setiap modul
6. Menguji modul
7. Menguji level terendah dari integrasi

8. Menyimpan hasil dari semua pengujian; mengumpulkan modul yang telah selesai ke dalam integrasi
9. Memulai penyusunan dokumentasi untuk *user*, antara lain: *User's Guide* (Panduan *User*), *Maintenance Guide* (Panduan Pemeliharaan), *Operator's Guide/System Manager's Guide*, serta Dokumentasi Pelatihan (jika diperlukan kursus khusus mengajarkan cara penggunaan sistem). Akan tetapi, *User's Guide* yang baik seharusnya sudah cukup untuk sebagian besar kasus.

5. System Test

Tahapan ini memiliki dua aktivitas yaitu modul-modul diintegrasikan ke dalam suatu sistem dan sistem itu diuji. Pengujian dilakukan untuk memastikan bahwa modul-modul dapat bekerja bersama dan fungsi-fungsi sistem berperan seperti seharusnya. Menurut Rakos (1990), sulit mengestimasi berapa lama *system test* harus dilakukan. Tim proyek tidak dapat menyatakan bahwa sejumlah *N bugs* akan memerlukan waktu masing-masing empat jam untuk perbaikan. Suatu petunjuk praktis menyatakan bahwa *system test* akan memakan seperdelapan waktu dari total waktu yang diperlukan dalam pemrograman. Jika aktivitas-aktivitas yang ada telah memakan waktu lebih lama dari rencana, maka *system test* akan memakan waktu lama pula. PM yang baik harus memikirkan banyak kemungkinan pada saat menjadwalkan *system test*.

Aktivitas pada tahapan ini adalah merencanakan *system test* dalam dokumen *System Test Plan (STP)*. STP tidak hanya mendokumentasikan urutan modul-modul yang diintegrasikan tetapi juga pengujian yang akan dijalankan pada setiap tahapan dari integrasi.

Isi dari STP adalah:

- jadwal pengujian, staf dan kebutuhan sumber daya pengujian.
- manajemen konfigurasi, integrasi, dan alat-alat pengujian yang digunakan
- urutan integrasi.
- *checklist* pengujian untuk menjalankan masing-masing langkah dari integrasi dan sumber dari pengujian.

- *checklist* dari data yang buruk dan prosedur-prosedur untuk dicoba.
- proses regresi (kemunduran).
- pengujian *load* dan kinerja.
- daftar *deliverables* (*source code*, dokumentasi).

6. Acceptance

Tujuan dari tahapan ini adalah pengakuan atau pernyataan tertulis dari *user* bahwa tim proyek telah memberikan hasil sesuai perjanjian. Pada tahapan ini, tim proyek hanya menjalankan pengujian yang sudah didefinisikan pada ATP pada sesi yang formal dan teratur untuk mendemonstrasikan fungsi-fungsi sistem yang sudah dijanjikan. *Milestone* utama dan nyata dari tahapan *Acceptance* adalah tanda tangan penerimaan *user*. Beberapa hal yang harus diperhatikan sebelum tahapan ini dimulai adalah:

- ATP harus sudah ditulis, ditinjau ulang oleh *user* dan direvisi jika diperlukan.
- waktu spesifik telah ditentukan dan dikonfirmasi dengan *user* yang berwenang untuk penandatanganan penerimaan ini.
- personil tim proyek yang dibutuhkan untuk menjalankan tahapan ini harus diinformasikan terlebih dahulu berkaitan dengan negosiasi, hal-hal teknis, perbaikan dan tanda tangan persetujuan.
- sumber daya yang dibutuhkan untuk tahapan ini harus dipersiapkan, antara lain piranti lunak, perangkat keras, formulir untuk tanda tangan persetujuan, dan pengandaan *acceptance script*.
- dokumentasi bagi *user* harus tersedia karena merupakan bagian dari *deliverables*.
- tim proyek telah menjalankan ATP dengan baik dan melakukan koreksi yang dibutuhkan.
- prosedur penerimaan telah disetujui sebelumnya.

7. Operation

Pada tahapan ini, tim proyek siap untuk menjalankan sistem dan membiarkan *user* yang sebenarnya untuk mulai menggunakannya. Pada tahapan ini, seluruh tim proyek belum lepas tangan dalam proyek pembuatan sistem.

Jika *user* menemukan masalah atau pertanyaan, maka tim proyek yang terlibat harus menjawabnya secepat mungkin. Aktivitas utama dari tahapan ini adalah *warranty* (garansi). *Warranty* adalah suatu periode di mana tim proyek memperbaiki masalah-masalah yang muncul dari sistem setelah tahapan *Acceptance*. Aktivitas lainnya adalah pertemuan peninjauan ulang pasca proyek untuk memastikan kesalahan-kesalahan yang dibuat dalam proyek tidak diulang kembali. *Milestone* tahapan ini adalah sistem operasional secara penuh dan penjualan proyek berikutnya.

Menurut Rakos (1990), pada umumnya masa garansi (*warranty period*) dalam industri piranti lunak adalah enam bulan sampai satu tahun. Penyediaan garansi berupa salah satu atau kombinasi dari tiga cara berikut:

1. Menyediakan seseorang pada sisi *user* untuk menjawab permasalahan. Orang ini seharusnya merupakan *project leader* atau anggota tim proyek senior yang mengetahui setiap aspek dari sistem.
2. Menyediakan seseorang yang dapat menjawab permasalahan dan dapat diakses via telepon. Semua anggota tim proyek harus bisa diakses.
3. Menyediakan seseorang yang dapat menjawab permasalahan dalam satu periode waktu yang singkat setelah panggilan telepon diterima.

Tim proyek sudah selesai dengan keseluruhan proyek, jika:

- Sistem baru sudah selesai dan berjalan dengan lancar.
- Konversi atau *cut over* dari sistem lama sudah lengkap.
- *End user* diberikan pelatihan dan merasa sesuai dengan sistem baru.
- *Warranty* disediakan.
- Tinjauan ulang pasca proyek diadakan dan semua hal yang dapat bermanfaat bagi proyek mendatang didokumentasi.
- Tanggung jawab dan metode pemeliharaan yang berjalan harus didefinisikan.

Perencanaan Proyek Piranti Lunak

Setiap usaha pengembangan piranti lunak akan melalui suatu siklus hidup yang

merupakan model perspektif mengenai semua aktivitas yang terjadi antara awal dan akhir proyek. Model yang paling kuno dipergunakan adalah *Waterfall model* yang sekaligus menjadi dasar bagi perkembangan model lainnya. Suatu model yang sebagian dari siklus hidupnya menggunakan *Waterfall Model* adalah *Staged Delivery*.

Waterfall Model

Menurut Olson (2004), *Waterfall Model* menggunakan *loop* umpan balik antara tahapan-tahapan dari pengembangan piranti lunak untuk meminimalisasi kerja ulang (*rework*). Model ini dinamakan *waterfall* karena setiap langkahnya mengikuti *predecessor* (aktivitas yang harus dilaksanakan sebelum aktivitas yang mengikutinya dimulai) secara berurutan. Model ini memiliki beberapa keuntungan, antara lain:

- Mendorong adanya perencanaan sebelum perancangan
- Mendekomposisi (menguraikan) pengembangan sistem ke dalam *subgoal* dengan *milestone* yang menunjuk pada penyelesaian *intermediate products*.
- Memungkinkan PM untuk memantau kemajuan proyek dengan lebih tepat dan menyediakan struktur proyek.
-

Pada *Waterfall Model* yang asli, masalah-masalah tidak diperhatikan sampai dengan penyelesaian proyek, sehingga memakan biaya yang mahal untuk memperbaikinya. Oleh karena itu, *loop* umpan balik ditambahkan, sejalan dengan *prototyping* untuk mengetahui masalah di awal. *Waterfall Model* tidak memungkinkan respon yang cepat pada penyelesaian perubahan dalam proyek. Urutan aktifitas yang teratur pada *Waterfall Model* tidak mengakomodasi pengembangan baru.

Staged Delivery

Staged Delivery adalah suatu model siklus hidup di mana piranti lunak dikembangkan dalam tahapan-tahapan. *Staged Delivery* tidak mengurangi waktu yang dibutuhkan untuk membangun produk piranti lunak, tetapi secara substansial mengurangi resiko yang terlibat dalam membangun produk dan juga menyediakan tanda-tanda kemampuan

yang dapat diukur dan kelihatan bagi klien dan berguna bagi manajemen dalam menentukan status proyek. *Staged Delivery* dapat meningkatkan keseluruhan kualitas kode, mengurangi resiko pembatalan proyek dan mendukung usaha membangun anggaran.

Dengan *Staged Delivery*, tahapan pengembangan melewati langkah-langkah *Waterfall model*, yaitu mendefinisikan konsep piranti lunak, menganalisa kebutuhan, dan menciptakan suatu perancangan arsitektural. Setelah itu, tahapan dilanjutkan dengan perancangan rinci, pengkodean, *debugging* dan pengujian dalam masing-masing langkah, dan menciptakan produk yang dapat diluncurkan pada akhir masing-masing langkah.

Staged Delivery memberikan dukungan dalam beberapa hal, yaitu :

- Kemampuan pengembangan piranti lunak yang lebih kelihatan
- Siklus peluncuran produk yang lebih pendek untuk pengembangan piranti lunak *shrink wrap*
- Peringatan awal akan masalah-masalah
- Biaya *overhead* bisa dikurangi karena *Staged Delivery* juga mengeliminasi waktu administratif yang digunakan *developer* untuk menciptakan laporan kemajuan (*progress report*) dan laporan pemantauan kemajuan tradisional lainnya.
- Pengurangan kesalahan estimasi
- Minimalisasi masalah integrasi

Model ini adalah model daur hidup di mana tim proyek menunjukkan piranti lunak yang dikembangkan kepada klien dalam langkah-langkah yang berurutan. Dengan menggunakan model ini, tim proyek mengetahui dengan pasti apa yang sedang dibangun ketika tim proyek mulai membangunnya. Hal yang membuat model ini khusus adalah tim proyek tidak menyerahkan piranti lunak pada akhir proyek dalam satu kali kejadian. Tim proyek menyerahkannya dalam langkah-langkah yang berurutan sepanjang proyek. Model ini dikenal sebagai *Incremental Implementation*.

Keuntungan utama dari model ini adalah memungkinkan tim proyek untuk menyerahkan fungsionalitas yang berguna ke klien lebih awal dari pada jika tim proyek menyerahkan 100%

proyek pada akhir proyek. Jika tim proyek merencanakan langkah-langkah dengan hati-hati, maka dapat menyerahkan fungsionalitas yang paling penting paling awal dan klien dapat mulai menggunakan piranti lunak pada waktu itu.

Model ini juga menyediakan tanda-tanda kemajuan proyek yang dapat diukur lebih awal dari pada pendekatan yang kurang *incremental*. Tanda kemajuan seperti itu dapat menjadi sesuatu yang berharga dalam menjaga tekanan jadwal pada tingkat manajerial. Kekurangan utama dari model ini adalah tidak berfungsi tanpa perencanaan yang seksama baik di tingkat manajemen maupun teknis. Pada tingkat manajemen, *Project Manager* harus merencanakan langkah-langkah yang berarti kepada klien dan mendistribusikan aktivitas-aktivitas kerja di antara anggota tim proyek sehingga mereka dapat menyelesaikan pekerjaan mereka tepat waktu dengan *deadline* dari langkah-langkah yang ada. Pada tingkat teknis, tim proyek harus membuat ketergantungan teknis yang jelas antara komponen-komponen yang berbeda dalam produk. Suatu kesalahan yang umum adalah menunda pengembangan suatu komponen sampai langkah keempat hanya untuk menemukan bahwa suatu komponen yang direncanakan untuk langkah kedua tidak dapat bekerja tanpa langkah keempat itu.

Joint Application Development (JAD)

Menurut McConnell (1996) *Joint Application Development* (JAD) adalah metodologi *requirement definition* dan desain *user interface* di mana *end-user*, pihak manajemen, dan *developer* menghadiri pertemuan secara intensif untuk mengumpulkan rincian sistem. JAD lebih fokus pada masalah bisnis daripada rincian teknis. JAD paling dapat diaplikasikan pada pengembangan sistem bisnis tetapi dapat dipergunakan dengan sukses pada piranti lunak sistem dan *shrink wrap*. Metodologi ini menyediakan penghematan dengan memperpendek waktu yang diperlukan untuk mengumpulkan kebutuhan suatu sistem dan dengan mengumpulkan kebutuhan yang lebih baik sehingga mengurangi jumlah biaya dan mengurangi perubahan kebutuhan. Keberhasilan JAD

tergantung pada kepemimpinan yang efektif dari sesi JAD, partisipasi *end-user* utama, pihak manajemen, *developer*, dan pencapaian sinergi kelompok selama sesi JAD.

JAD singkatan dari *Joint Application Development*. "Joint" menunjuk pada fakta bahwa *developer*, *end-user*, dan pihak yang terlibat lainnya bersama-sama merancang konsep produk. Ini merupakan suatu proses terstruktur untuk mengumpulkan kebutuhan dan negosiasi. JAD memberikan penghematan dalam beberapa hal, yaitu:

1. JAD berkomitmen dengan eksekutif puncak mengenai proses perencanaan piranti lunak. JAD melibatkan eksekutif puncak dari awal pengembangan produk. Keterlibatan eksekutif di awal ini memperpendek siklus persetujuan produk.
2. JAD memperpendek tahapan *requirement specification*. JAD mengurangi sejumlah waktu yang diperlukan untuk mengumpulkan kebutuhan yang secara otomatis juga memperpendek keseluruhan siklus pengembangan. Hal ini sangatlah berguna karena *requirement specification* adalah suatu aktivitas penting yang dapat menambah waktu mingguan atau bulanan pada awal dan akhir proyek.
3. JAD mengurangi fitur dari nilai-nilai yang dapat ditanyakan. Dengan mengurangi fitur ini dan membuat produk lebih kecil, JAD mengurangi waktu pengembangan.
4. JAD membantu memperoleh kebutuhan tepat di awal analisa kebutuhan dan *end-user* biasanya berbicara dengan bahasa yang berbeda yang berarti mereka kurang efektif berkomunikasi tentang kebutuhan piranti lunak. JAD meningkatkan komunikasi dan mengurangi kerja ulang yang dihasilkan dari kebutuhan yang salah.
5. JAD membantu memperoleh *user interface* (UI) tepat di awal. Beberapa produk membutuhkan kerja ulang yang ekstensif karena *end-user* menolak UI dari produk. Sesi desain JAD berfokus pada desain UI. Karena *end-user* dilibatkan dalam sesi ini, UI yang dikembangkan biasanya diterima oleh *end-user* pada akhirnya.
6. JAD mengurangi perselisihan organisasional. Banyak proyek berjalan pincang

karena konflik akan objektif atau agenda tersembunyi. Dengan mengumpulkan semua pembuat keputusan bersama untuk merancang sistem, JAD membawa isu-isu tersebut untuk diperjelas lebih awal dari proyek.

JAD memiliki 2 fase utama, yaitu fase perencanaan JAD dan fase perancangan JAD. Dalam kedua fase, JAD lebih fokus pada masalah perancangan bisnis daripada rincian teknis murni. Selama fase perencanaan JAD, penekanan ada pada pemetaan kemampuan sistem piranti lunak secara luas.

Hasil utama dari fase ini adalah tujuan sistem, *preliminary effort* dan estimasi jadwal, dan keputusan apakah tim proyek melanjutkan ke pengembangan produk lebih lanjut. Perencanaan JAD juga mempersiapkan fase perancangan JAD. Jika keputusan yang diambil adalah tetap dengan melanjutkan produk, maka dilanjutkan dengan fase perancangan JAD. Fase ini mendatangkan kebutuhan yang lebih detil dan tujuannya adalah untuk menciptakan desain tingkatan *user*. Meskipun disebut perancangan, fase ini tidak hanya fokus pada perancangan fungsional sistem saja. Perancangan JAD menggunakan *prototyping* secara ekstensif dan hasil utama dari fase ini adalah desain *user interface* (UI) yang rinci, suatu skema *database* (jika sesuai), dan anggaran yang disesuaikan serta estimasi jadwal.

Metode Penjadwalan

Menurut Olson (2004), *Critical Path Method* (CPM) adalah metodologi penjadwalan proyek untuk menentukan jadwal aktivitas yang optimal dengan memperhatikan waktu. CPM menyediakan dasar untuk mengidentifikasi aktivitas-aktivitas khusus yang kritis dan membantu menentukan aktivitas-aktivitas mana yang dapat ditunda dalam kondisi *resource* yang terbatas. Aktivitas kritis adalah aktivitas yang harus diselesaikan sesuai jadwal. Menurut Shelford dan Remillard (2003), *project manager* harus fokus pada jalur kritis, aktivitas-aktivitas yang harus diselesaikan tepat waktu, jika tidak, keseluruhan proyek akan terlambat. Suatu jalur kritis adalah rantai aktivitas kritis mulai dari awal proyek sampai dengan penyelesaian proyek. Jalur kritis dapat dilihat dengan

mengikuti jalur durasi terpanjang melalui Gantt Chart atau diagram jadwal lainnya. Gantt Chart menunjukkan informasi tugas (*task*) dalam proyek sebagai suatu serial batang (*bars*) sepanjang skala waktu (*timescale*). *Bars* secara grafik menunjukkan durasi *task* dengan tanggal mulai dan selesai seiring dengan kemajuan terhadap waktu. Informasi yang diperlukan dalam penentuan jalur kritis, meliputi:

Early Start (ES)

Untuk setiap aktivitas yang tidak memiliki *predecessor*, aktivitas tersebut dapat dijadwalkan untuk dilaksanakan secepat mungkin. Jadi ES merupakan waktu mulai paling awal yang dapat dijadwalkan untuk suatu aktivitas (*earliest start time*).

Early Finish (EF)

Waktu selesai paling awal suatu kegiatan (*earliest finish time*). Bila hanya ada satu kegiatan terdahulu, maka EF suatu kegiatan terdahulu merupakan ES kegiatan berikutnya. EF dapat diperoleh dengan menjumlahkan (aktivitas awal) *early start time* dengan *duration* dimulai dari arah atas.

Late Start (LS)

Waktu paling akhir suatu kegiatan dapat dijadwalkan untuk dimulai (*latest allowable start time*), yaitu waktu paling akhir kegiatan boleh dimulai tanpa memperlambat penyelesaian proyek secara keseluruhan. *Late Start* dihitung mulai dari arah bawah (hitungan mundur), dengan mengurangi *late finish time* dengan *duration*.

Late Finish (LF)

Waktu paling akhir aktivitas boleh selesai (*latest allowable finish time*) tanpa memperlambat penyelesaian proyek. *Late finish* juga dihitung mulai dari arah bawah (hitungan mundur), dengan mengambil *early finish* yang paling akhir atau dengan mengambil nilai minimum dari *late start* aktivitas-aktivitas selanjutnya.

Slack

Merupakan jarak antara *early start* dengan *late start* atau *early finish* dengan *late*

finish. Slack dapat diperoleh dengan mengurangi *late start* dengan *early start* atau *late finish* dengan *early finish*. Aktivitas-aktivitas yang memiliki nilai *slack* sama dengan nol (0) disebut aktivitas kritis akan membentuk sebuah jalur kritis. Jika ada aktivitas kritis yang ditunda maka akan mempengaruhi penyelesaian proyek secara keseluruhan.

Penjadwalan AON

Menurut Jack (2003), Di dalam merancang CPM, penulis juga menambahkan pendekatan yang menggambarkan jaringan kerja yang nantinya memudahkan project manager didalam menerapkan jalur kritis yang dibuat. Metode ini digambarkan dalam bentuk simpul (node), metode AON ini juga berfungsi memperjelas jalur kritis yang dibuat oleh project manager.

Sistem yang Sedang Berjalan

PT. Media plus, didirikan pada tahun 1998, dengan visi awalnya adalah membuat piranti lunak yang memberikan manfaat dan perbedaan yang signifikan kepada klien-kliennya. Seiring dengan perkembangan *internet* di Indonesia, pada akhir tahun '90an, media plus mulai berfokus pada aplikasi *internet* yang sedang *trend* di pangsa pasar saat itu.

Media plus juga membentuk sebuah unit bisnis baru bernama Equalitas. Tahun 2001, media plus memperluas layanannya dengan menyediakan CTI (*Computer Telephony Integration*), ERP, dan *training services*.

Tahun 2002, unit bisnis Equalitas bergabung dengan Media plus. Setelah penggabungan ini, Media plus memperluas lagi layanannya pada aplikasi berbasis *web* untuk melengkapi solusi piranti lunak.

Dalam perkembangan pada layanan aplikasi berbasis *web*, media plus telah mengembangkan suatu aplikasi yang memudahkan pembuatan *web site* perusahaan dengan fitur-fitur handal dan mudah dikelola dengan fasilitas *Content Management System* (CMS) yang dikenal dengan *Frontline WebCentral* (FLWC). FLWC ini dilengkapi dengan berbagai modul yang dapat dipilih dan diintegrasikan dengan aplikasi lain. Skala dan fungsi *web site* dapat disesuaikan dengan kebutuhan perusahaan di masa mendatang.

Pada awal berdirinya, Media plus memiliki divisi *Sales and Consultant* yang tergabung menjadi satu divisi dengan jumlah karyawan berkisar lima sampai tujuh orang. Selain itu, Media plus memiliki divisi *Production* yang terbagi atas beberapa sub divisi yaitu *Graphic Design* berkisar enam sampai tujuh orang, *Multimedia* sebanyak empat orang, *Information Design* (yang bertanggung jawab pada tahap awal proyek di mana karyawan ini membuat *function* dan struktur sistem), dan *Programmer* yang terbagi dua yaitu *Front End/Web Developer* dan *Enterprise Developer* (bertanggung jawab dalam hal *data base*). Saat itu Media plus juga memiliki *project coordinator* yang bertanggung jawab untuk melakukan estimasi jadwal dan biaya.

Pada tahun 2002, divisi *Sales and Consultant* dipisah untuk mengatasi masalah *bottleneck* tersebut. Media plus membentuk divisi *Sales* dan divisi *Consultant* yang memiliki tugas dan tanggung jawab masing-masing. Tugas dan tanggung jawab divisi *Sales* adalah melakukan *follow up* terhadap klien dan kemudian mempertemukan *consultant* dengan klien tersebut. Tugas dan tanggung jawab *consultant* adalah *sales support* (membuat pertemuan bersama dengan klien dan sales untuk membahas kebutuhan *user*), sebagai konsultan klien saat proyek sudah selesai, sebagai *project manager* pada proyek tersebut, dan juga berperan sebagai *role* atau posisi sebelumnya (sebelum divisi *Sales and Consultant* dipisah). Akan tetapi, pemisahan divisi *Sales and Consultant* tersebut belum dapat mengoptimalkan kinerja karyawan. Saat itu, masih terjadi *overload* pada posisi *consultant* yang juga berperan sebagai *project manager* pada proyek yang ditangani.

Pada bulan Februari 2004, divisi *Sales* berganti nama menjadi *Business Group* dan terbagi menjadi empat jenis sales atau industri yang berbeda yaitu *Finance, Banking and Investment (FBI), Government, Manufacturing,* dan *Telecommunication*; divisi *Consultant* berganti nama menjadi *Solution Group* yang terbagi menjadi tujuh jenis produk yaitu *Collaboration (Document Management and Business Intelligence), Enterprise Resource Planning & Customer Relationship Management & Accoun-*

ting, Banking Application, Web & Multimedia, Hardware, Professional Services, dan Training. Masing-masing *Solution Group* dikepalai oleh seorang *solution group manager*.

Proyek Web Sekolah Pelita Harapan

Proyek *web* Sekolah Pelita Harapan (SPH) ini adalah proyek yang dikembangkan untuk menggantikan *web* SPH yang lama khususnya SPH Karawaci. *Web* SPH ini dapat diakses lewat situs www.sph.edu. *Web* ini akan digunakan oleh siswa, orang tua siswa, alumni, orang tua calon siswa, dan calon siswa dari SPH. *Web* yang lama mengandung informasi yang kurang detil mengenai profil dan produk dari SPH. Sedangkan *web* baru menampilkan informasi mengenai setiap level pada *Junior School* yaitu mulai dari Taman Kanak-kanak sampai dengan *Grade 9* (atau setingkat SMP kelas 3) dan level *High School* (setingkat SMU). Akan tetapi, informasi yang ditampilkan tidak termasuk rincian biaya karena pihak SPH mengharapakan orang tua dan calon siswa dapat menghubungi atau mendatangi pihak SPH secara langsung sehingga informasi tersebut dapat sampai kepada orang yang tepat.

Proyek Web Hortikultura

Proyek ini tergolong proyek kecil karena tingkat kompleksitas yang rendah dan resiko yang kecil. *Web* Hortikultura ini bersifat informatif dengan berorientasi pada para peneliti sebagai *end user*. Pada *web* ini ditampilkan informasi mengenai buah-buahan, sayur-sayuran, dan tanaman hias. Badan Hortikultura yang menjadi klien dari Media plus ini merupakan suatu badan yang bernaung di bawah Departemen Pertanian. Sebelumnya Badan Hortikultura sudah memiliki *web* informatif yang hanya dapat diakses jika *user* sudah mengakses *web* yang dimiliki oleh Departemen Pertanian. Karena *web* ini belum memberikan kontribusi yang maksimal bagi para peneliti, maka Badan ini bermaksud untuk memaksimalkan informasi dan tampilan *web* serta membuat *domain* sendiri melalui proyek *Web* Hortikultura (www.litbanghortikultura.go.id).

Pembahasan Evaluasi Pengelolaan Manajemen Proyek

Proyek yang akan dievaluasi akan menghasilkan beberapa metode tambahan didalam siklus pengelolaan waterfall pada tahapan requirement, dimana dalam tahapan tersebut akan ditambahkan metode JAD [Joint Application Development] yang nantinya akan memberikan kinerja yang lebih efektif didalam mengumpulkan user requirement. selain metode JAD, penulis juga menerapkan Staged Delivery dimana metode ini nantinya akan memberikan dukungan perencanaan proyek yang lebih jelas dari segi penyerahan fungsionalitas yang lebih awal kepada klien sehingga klien dapat melihat kemajuan proyek secara nyata.

Metode Penelitian

Penulis menggunakan beberapa metode penelitian manajemen proyek di media plus sebagai berikut:

- Studi pustaka guna memperoleh landasan teori yang mendukung penelitian ini, metode studi pustaka memiliki keunggulan dalam masalah waktu yang diperlukan untuk mengumpulkan data dan informasi pendukung, namun memiliki kekurangan dalam hal validitas dan keakuratan data yang diperoleh.
- Studi lapangan ke objek penelitian yakni media plus mengkaji secara lebih permasalahan yang akan diteliti. Studi lapangan memiliki keunggulan dalam mengekstraksi data secara lebih akurat dan mendalam dari sumber utama, namun kelemahan metode ini adalah waktu yang dibutuhkan lebih lama untuk pelaksanaannya.

Pembahasan Evaluasi Umum Manajemen Proyek

Dari hasil pemantauan dan analisis terhadap jalannya proyek dari awal sampai dengan akhir proyek. Hal-hal meliputi *Work Breakdown Structure* (WBS), *milestones*, aktivitas-aktivitas yang mengalami kemunduran serta hal yang menyebabkan kemunduran aktivitas dibahas dalam masing-masing proyek.

Evaluasi Perencanaan Proyek

Proyek yang sudah berjalan di Media plus, yakni SPH, dan Hortikultura melakukan perencanaan piranti lunak dengan cara *Work Breakdown Structure* (WBS) sehingga anggota tim proyek mengetahui semua aktivitas yang harus diselesaikan untuk mencapai serahan proyek. Berikut adalah WBS dari masing-masing proyek. WBS ini kemudian digunakan untuk melakukan estimasi waktu atau penjadwalan proyek-proyek di Media plus.

Dari analisis yang ditulis mengenai penggunaan *Waterfall Model* sebagai siklus hidup proyek *web* yang dikelola Media plus. Dengan model ini, dapat dilihat bahwa klien cenderung dapat meminta perubahan atau revisi berkali-kali. Pada akhirnya semua proyek cenderung tidak selesai tepat pada waktu yang sudah direncanakan sebelumnya melalui *Microsoft Project tool*.

Evaluasi Penjadwalan Proyek

Penulis juga memanfaatkan suatu *tool* untuk manajemen proyek piranti lunak, yaitu *Microsoft Project 2003*. *Tool* ini banyak disarankan oleh pakar-pakar yang menulis tentang manajemen proyek piranti lunak umumnya dan proyek *web* khususnya. Dengan menggunakan *tool* ini, tim proyek khususnya PM pada proyek SPH, dan Hortikultura dapat mengatur penjadwalan dan memantau perkembangan aktivitas proyek melalui menu *Gantt Chart*.

Evaluasi Pengendalian Proyek

Dari hasil analisis pengendalian proyek, Media plus telah melakukan pengendalian terhadap proyek piranti lunak berbasis *web*. Pengendalian ini dilakukan dengan mengadakan pertemuan (*meeting*) dan pelaporan (*reporting*).

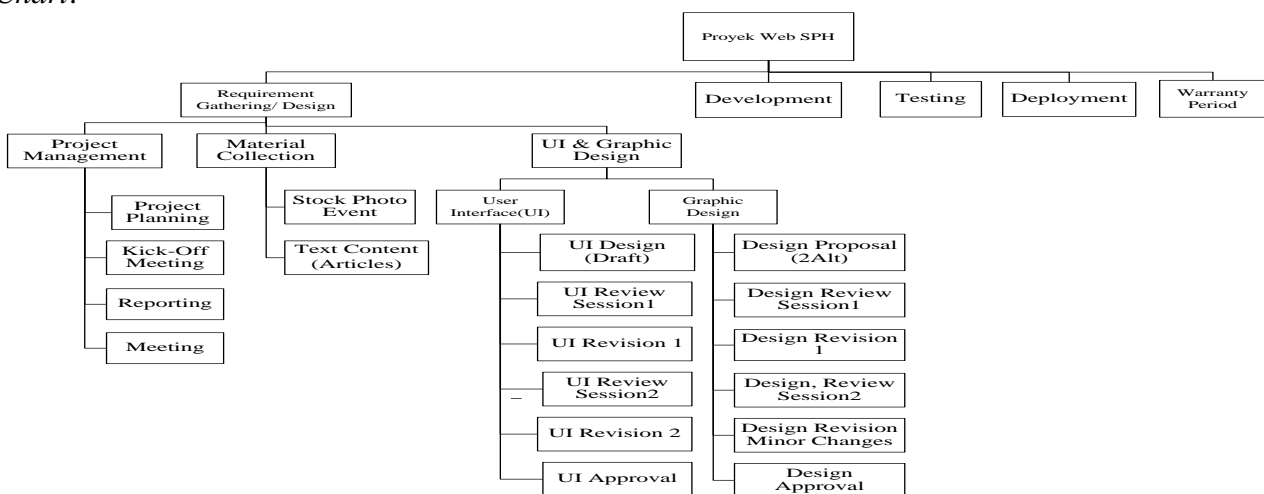
Dalam setiap pertemuan yang diadakan, *project manager* memberikan pelaporan kemajuan proyek yang ditangani. Pelaporan ini dibuat baik secara lisan maupun tulisan oleh *project manager* dengan format masing-masing (belum ada standarisasi), dan bahkan melalui *email*. Saat ini pembuatan laporan di Media plus belum menggunakan fasilitas yang tersedia dalam *Microsoft Project*.

Evaluasi Khusus Manajemen Proyek

Hal-hal meliputi *Work Breakdown Structure* (WBS), *milestones*, aktivitas-aktivitas yang mengalami kemunduran serta hal yang menyebabkan kemunduran aktivitas dibahas dalam masing-masing proyek.

Evaluasi Manajemen Proyek Web Sekolah Pelita Harapan

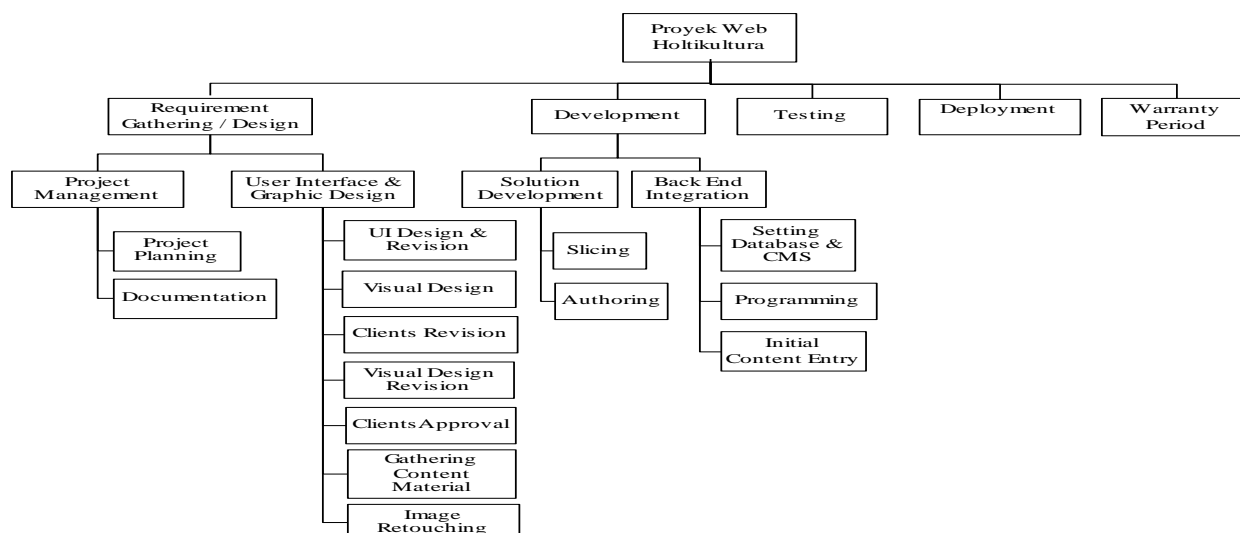
Proyek *web* Sekolah Pelita Harapan (SPH) terdiri atas sekumpulan aktivitas atau *task* yang terstruktur dalam gambar *Work Breakdown Structure*. (gambar 1 dan gambar 2).



Sumber: Self Co, piled

Gambar 1

Work Breakdown Structure dari Tahapan Requirement Gathering / Design Proyek Web SPH



Sumber: Self Co,piled

Gambar 2

Work Breakdown Structure dari Tahapan Requirement Gathering /Design dan Development Proyek Web Hortikultura

Kesimpulan

Media plus adalah sebuah perusahaan *software house* yang sedang berkembang dan memiliki beberapa *solution group* untuk menjawab berbagai jenis kebutuhan *user* seperti *Collaboration (Document Management & Business Intelligence), Professional Services, Hardware, Web & Multimedia, Banking Application* sampai dengan *Training*. Pada tulisan ini penulis melakukan penelitian pada *Web & Multimedia Solution Group* khususnya pengelolaan pada proyek web Sekolah Pelita Harapan (SPH), dan Hortikultura.

Daftar Pustaka

Anonim, "A Guide to The Project Management Body of Knowledge", Project Management Institute, Upper Darby, 1996.

Cleland, David I, Ireland, Lewis R, "Project Management: Strategic Design and Implementation", 4th edition, McGraw-Hill, Inc., Singapore, 2002.

Ellsworth, Jill H, Matthew V, "Marketing on the Internet: Pemasaran di Internet", Grasindo, Jakarta, 1997.

Hahn, Harley, "The Internet Complete Reference", 2nd edition, McGraw-Hil Inc., New York, 1996.

Hughes, Bob; Cotterell, Mike, "Software Project Management", 2nd edition, McGraw-Hil Inc, New York, 1999.

Jack R. Meredith. Samuel J.Mantel, Jr, "Project Management, A Managerial Approach" 5th, John Wiley & Sons, Inc., USA, 2003.

McConnell, Steve, "Rapid Development: Taming Wild Software Schedules", Microsoft Press, Washington, 1996.

Olson, David L, "Introduction to Information Systems Project Management", McGraw-Hill Inc., New York, 2004.

Pressman, Roger S, "Software Engineering: A Practitioner's Approach", 5th edition, McGraw-Hil Inc., New York, 2001.

Rakos, John J, "Software Project Management: For Small to Medium Sized Projects", Prentice Hall Inc., New Jersey, 1990.

Shelford, Thomas J, Remillard, Gregory A, "Real Web Project Management: Case Studies and Best Practices from the Trenches", Addison-Wesley, Boston, 2003.