Rangkuman MKPL

Bab I – Introduction

* Keunikan software quality assurance (assurance = jaminan)
  + Terdapat beberapa perbedaan antara industri non-IT dan industri IT dalam hal produknya, perbedaannya antar a lain :

|  |  |  |
| --- | --- | --- |
| Karakteristik | Software Product | Non-IT Industrial Product |
| Complexity | Produknya sangat kompleks, sehingga memungkinkan sangat banyak langkah operasional yang dibutuhkan | Tingkat kerumitannya lebih rendah, karena langkah operasionalnya biasanya hanya memiliki ribuan langkah |
| Visibility of Product | Produk yang dihasilkan bersifat abstrak, sehingga mustahil untuk melihat adanya cacat produksi secara langsung (dengan mata) (hasil produk disimpan di media penyimpanan eksternal) | Produknya nyata, dapat disentuh dan dilihat, sehingga jika ada cacat dapat dideteksi secara langsung (secara kasat mata) |
| Proses Produksi | Kesempatan untuk mendeteksi adanya cacat hanya ada satu kali, yaitu pada saat fase *product development* | Kesempatan untuk mendeteksi cacat produk dapat dilakukan pada semua fase produksi :   * Product development * Product production planning * Manufaktur |

* + Kesimpulan, keunikan proses pengembangan software adalah :
    - Kerumitannya tinggi jika dibandingkan dengan produk industri lainnya
    - Hasil produk yang abstrak
    - Kesempatan untuk mendeteksi adanya cacat sangat terbatas, yaitu pada saat fase product development
* Lingkungan dimana metode SQA dikembangkan (?)
  + Software dikembangkan oleh berbagai kalangan, yaitu :
    - Siswa dan pelajar mengembangkan software sebagai abgian dari pendidikan
    - Pengembang software amatir yang mengembangkan software karena hobi
    - Engineer di berbagai bidang, yang mengembangkan software untuk kepentingan pekerjaan mereka
    - Software developer professional (sistem analis dan programmer) mengembangkan software sebagai mata pencaharian.
  + Sedangkan untuk karakteristik lingkungan pada SQA antara lain
    - **Contractual condition (Terikat kontrak)**, software yang ada merupakan hasil dari komitmen dan kondisi yang telah tertera pada kontrak antara software developer dan customer, aktivitasnya antara lain :
      * Functional requirement
      * Project budget
      * Timeline project
    - **Bergantung dengan hubungan antara customer dan supplier,** dalam proses pengembangan software terdapat banyak aktivitas yang membutuhkan arahan langsung atau keterlibatan langsung dari customer, sehingga tim harus selalu menjaga komunikasinya dengan customer, beberapa aktivitas tersebut adalah
      * Mempertimbangkan adanya perubahan pada software
      * Mendiskusikan pendapat kustomer tentang berbagai aspek project
      * Mendapatkan persetujuan jika ada perubahan yang akan dilakukan oleh tim developer
    - **Membutuhkan kerja tim,** dalam beberapa kasus software akan dirasa sangat berat jika hanya dikerjakan seorang professional sehingga tim harus membagi pekerjaan yang dilakukan, tidak hanya itu, yang harus diperhatikan antara lain
      * Timetable, waktu kerja yang harus ditempuh untuk menyelesaikan project tersebut harus diperhatikan
      * Kebutuhan resource dengan kemampuan khusus untuk menyelesaikan project
    - **Membutuhkan kerja sama dan koordinasi dengan tim software yang lain,** tak jarang dibutuhkna lebih dari satu tim untuk menyelesaikan project pada pengembangan software skala besar. Dengan kata lain perlunya kerjasama dengan pihak berikut sangat dibutuhkan dan sangat mungkin terjadi :
      * Tim development lain pada organisasi yang sama
      * Tim pengembang hardware pada organisasi yang sama
      * Tim pengembang software dan hardware dari supplier yang berbeda
      * Tim pengembang software dan hardware dari pihak customer yang ikut serta dalam pengembangan project
    - **Adanya interaksi dengan sistem software yang lain**, dengan pesatnya perkembangan teknologi yang ada tentunya sangat banyak sistem sistem baru yang diciptakan, hal ini mengakibatkan perlunya ada integrasi software yang dibuat dengan sistem lainnya. Hal ini disebut interfacing, yang memungkinkannya integrasi antara dua sistem yang berbeda, diantaranya
      * Input interfaces, dimana sistem software lain mengirimkan data ke sistem software yang dikembangkan
      * Output intefaces, dimana sistem software yang dikembangkan mengirimkan data yang sudah diproses ke sistem software lainnya
      * Input/Ouput interface dari/ke control board mesin, pada mesin medis ataupun mesin laboratorium, mesin pemrosesan bahan mentah, dll
    - **Keharusan untuk menyelesaikan proyek meskipun anggota tim berganti,** tak jarang ada anggota tim yang meninggalkan tim atau resign di tengah-tengah project yang masih sedang berlangsung, hal ini tentunya membutuhkan penanganan khusus dari tim leader untuk mencari pengganti kekosongan tim tersebut. Dan dengan hilangnya anggota tim, apapun keadaannya, berapapun biaya yang harus dikeluarkan untuk anggota tim yang baru, project harus tetap berjalan.
    - **Keharusan untuk melakukan software maintenance dalam beberapa waktu,** customer mengharapkan product yang sudah jadi bisa bertahan lama yang biasanya antara waktu 5 – 10 tahun. Selama periode tersebut, kebutuhan untuk adanya maintenance akan meningkat dari waktu ke waktu, sehingga maintenance sangat diperlukan untuk memenuhi ekspektasi kustomer dalam produk yang ada.

**Bab II – Introduction**

* Software
  + Software merupakan program komputer, prosedur, dan dokumentasi serta data yang terkait tentang operasional sebuah software, dengan kata lain, software memiliki komponen
    - **Kode program,** dibutuhkan karena badan dari software tersebut adalah kode kode yang berkaitan satu sama lain
    - **Prosedur (procedure),**  dibutuhkan untuk mendefinisikan jalannya program dan bagaimana program berjalan, method yang dieksekusi dan lainnya
    - **Dokumentasi**, dibutuhkan sebagai sarana untuk kerjasama antara elemen yang terkait dalam pengembangan software
    - **Data yang berkaitan dengan jalannya program,** data yang termasuk antara lain adalah parameter, data pengguna, serta testing data.
* Software error, fault dan failures
  + Software error, mencakup kesalahan gramatikal (typo) sebuah baris code atau lebih, atau penerapan logika yang salah dalam sebuah kode program.
  + Software faults, merupakan software error yang menyebabkan kesalahan fungsional dari sebuah software
  + Software failures, akan terjadi ketika software faults di”aktifkan” yaitu pada saat pengguna mencoba untuk menjalankan sebuah fitur program yang memiliki error
* Klasifikasi penyebab software error
  + Adanya software error menyebabkan jeleknya kualitas software, penyebab dari software error sendiri sangat beragam, software error ini diklasifikasikan menjadi,
    - Kesalahan pada saat definisi requirement
    - Gagalnya komunikasi antara client-developer
    - Kesalahan yang disengaja oleh tim akibat pengubahan software requirement yang ada
    - Kesalahan desain logika
    - Kesalahan pada koding
    - Ketidakpatuhan seorang anggota tim terhadap koding standar dan dokumentasi yang telah ditentukan
    - Kesalahan pada saat fase testing
    - Kesalahan prosedur
    - Kesalahan pada dokumentasi
* Definisi software quality
  + Software quality memiliki beberapa definisi diantaranya adalah,
    - Tingkat kesesuaian/kesamaan antara sistem, komponen atau proses yang dibuat dengan requirement yang telah ditentukan
    - Tingkat dimana sistem, komponen atau proses yang dibuat memenuhi ekspektasi atau kebutuhan dari customer/user
  + Software quality menurut pressman, adalah kesesuaian dengan requirement functional & performance, dokumentasi, dan karakteristik yang diharapkan dalamseluruh pengembangan software yang pernah ada
* Software quality assurance
  + Memiliki beberapa definisi, diantaranya adalah :
    - Pola terstruktur dan sistematis dari semua aksi yang dibutuhkan untuk meyakinkan suatu produk sesuai dengan technical requirement yang sudah ada
    - Serangkaian kegiatan yang didesain untuk mengevaluasi proses dimana produk dikembangkan atau diproduksi
  + Quality control didefinisikan sebagai serangkaian aktifitas yang didesain untuk mengevaluasi kualitas sebuah produk yang tengah dikembangkan.
  + Quality control dan quality assurance memiliki tujuan yang berbeda
  + Quality control hanyalah bagian dari quality assurance
  + Ada beberapa klasifikasi tujuan dari SQA, yaitu
    - Software development (orientasi proses)
      * Memastikan tingkat keyakinan bahwa software yang dibuat akan sesuai atau memenuhi requirement teknis (technical requirement)
      * Memastikan tingkat keyakinan bahwa software yang dibuat akan sesuai dengan jadwal dan budget yang ada pada awal kontrak
      * Sebagai pembuka untuk melakukan aktifitas peningkatan efisiensi dari pengembangan software dan aktifitas SQA
    - Software maintenance
      * Memastikan tingkat keyakinan bahwa software maintenance akan memenuhi/sesuai dengan requirement teknis yang ada
      * Memastikan bahwa kegiatan maintenance akan memenuhi/sesuai dengan jadwal dan budget yang telah dibuat
      * Sebagai aktifitas permulaan untuk meningkatkan efisiensi dari software maintenance dan aktifitas SQA kedepannya

Bab 3 – Software Quality Factors

* Pentingnya software quality requirement yang lengkap
  + Tidak jarang user mendapati software yang mereka dapatkan memenuhi semua requirement dasar yang mereka definisikan **namun** mengalami beberapa permasalahan pada saat software tersebut dijalankan. Permasalahan ini bukan terkait error program, melainkan permasalahan yang disebabkan oleh keadaan yang tidak terduga oleh user, namun memang tidak semua permaslahan yang dihadapi sama diantara user yang ada.
  + Beberapa permasalahan tersebut memiliki karakteristik yang sama, diantaranya adalah
    - Seluruh requirement dasar telah terpenuhi
    - Permasalahan utama adalah adanya performa buruk pada saat software dijalankan, saat maintenance, dan isu kehandalan ataupun waktu yang dibutuhkan untuk melatih user
    - Penyebab dari buruknya performa dari software tersebut adalah kurangnya definisi requirement untuk mengatasi masalahh masalah yang sepele.
  + Dari karakteristik tersebut berarti sangat diperlukannya definisi requirement yang lengkap dan mendetail untuk mengatasi seluruh maslah yang dapat muncul pada saat penggunaan software
* Klasifikasi software requirement sebagai faktor penentu software quality
  + Untuk membuat requirement yang lengkap dan mendetail, terdapat model yang dinamakan McCall factor model yang memecah software requirement menjadi 11 software quality factor, dari 11 faktor tersebut dipecah menjadi 3 kategori, yaitu,
    - **Product Operation Factors**
      * Correctness
      * Reliability
      * Efficiency
      * Integrity
      * Usability
    - **Product Revision Factors**
      * Maintainability
      * Flexibility
      * Testability
    - **Product Transition Factors**
      * Portability
      * Reusability
      * Interopability
* Product operation
  + **Correctness requirement,** didefinisikan dalam sebuah list, list ini berisi output yang dibutuhkan dari sebuah sistem software, (seperti query untuk menampilkan saldo kustomer pada sebuah sistem informasi akuntansi), **singkatnya**, untuk requirement correctness yang perlu dicantumkan adalah hasil output dari sebuah aksi yang dapat dilakukan, sebenarnya untuk correctness dibagi menjadi beberapa dimensi, antara lain
    - Ouput yang diinginkan
    - Tingkat keakuratan output yang diinginkan (dapat berbeda-beda tergantung dengan data dan kalkulasinya)
    - Tingkat kelengkapan informasi pada output
    - Ke-aktualan informasi (tingkat up-to-dateness)
    - Waktu tersedianya informasi ouput

Contoh : *“software harus dapat menghasilkan 11 macam laporan dan 8 macam query untuk menampilkan informasi keuangan perusahaan, dengan tingkat error tidak lebih dari 1%, data yang dihasilkan juga mencantumkan user yang bertanggung jawab atas data tersebut, data laporan tersebut harus diperbaharui setidaknya 10 menit setelah data baru masuk.”*

* + **Reliability requirement**merupakan requirement yang berkaitan dengan penyediaan layanan. Requirement ini menetapkan seberapa sering software dibolehkan untuk “gagal”

Contoh : “*Frekuensi kegagalan software pendeteksi serangan ini tidak boleh lebih dari 1 kali dalam 2 tahun, dengan tingkat kegagalan deteksi harus dengan rasio 1:100000 kasus serangan”.*

* + **Efficiency requirement** berkaitan dengan resource yang dibutuhkan untuk menjalankan seluruh fungsi yang ada pada software. Yang dituliskan pada requirement ini biasanya adalah spesifikasi dari komputer yang akan digunakan untuk mengakses aplikasi tersebut.

Contoh : *“unit yang digunakan berupa unit pendeteksi gempa & tsunami yang ditempatkan dalam 12 node tersebar diseluruh penjuru samudra dalam radius 12Km dari pantai dengan kekuatan baterai sebesar 1000 mili-ampere yang dapat menyediakan waktu aktif mesin untuk kurang lebih 1 bulan, sistem harus mencatat dan mengirimkan data sekali sehari ke pusat oseanografi menggunakan teknologi wifi”*

* + **Integrity requirement** berkaitan dengan keamanan sistem software, yang didefinisikan pada requirement ini adalah hak akses dari user, siapa saja yang dapat mengakses data dan apa saja operasi yang dapat dilakukan oleh masing-masing user tersebut.

Contoh : *“Sistem e-budgeting DKI jakarta memungkinkan data laporan budgeting tahunan dapat diakses oleh masyarakat luas,pengaksesan berupa melihat, menyalin data namun tidak dapat melakukan insert data baru ke database. Setiap pergantian data harus disetujui gubernur dengan menyediakan password yang hanya diketahui gubernur DKI Jakarta”*

* + **Usability requirement** berkaitan dengan resource yang dibutuhkan untuk melatih pegawai baru untuk mengoperasikan software tersebut

Contoh : “*Software harus dapat dipelajari oleh staff baru dalam waktu dua minggu (16 jam pelatihan), dan diharuskan segera setelah pelatihan, staff baru dapat meng-handle 45 service perharinya”*

* Product revision
  + **Maintainability requirement,** menentukan seberapa besar effort yang harus dilakukan dan dibutuhkan oleh user dan tim maintenance untuk mengidentifikasi penyebab software falure, mengataisnya, dan memverifikasi solusi permasalahannya

Contoh : “*Jumlah software module tidak boleh lebih dari 30 statement, dan style programming harus mengikuti standar coding persuhaan.”*

* + **Flexibility requirement,** merupakan penentuan effort yang dibutuhkan untuk sebuah software dapat memenuhi berbagai kebutuhan user yang beragam

Contoh : “*Software pembuatan CV ini harus memiliki fitur untuk membuat field baru yang dapat di-custom oleh user, seperti menambah field dengan kontent list, text-box, multimedia, atau link”*

* + **Testability rquirement**, berisi tentang definisi kemampuan sebuah sistem dalam sebuah melakukan sebuah test-run terlebih dahulu sebelum menjalankan software tersebut. Biasanya digunakan untuk menentukan apakah software tersebut bekerja dengan benar sebelum software tersebut benar benar digunakan.

Contoh : “*Control unit canerized machinery diprogram untuk melakukan packing produk sarden yang baik dan memfilter sarden dengan kualitas buruk, sistem harus dapat melakukan testing terlebih dahulu setiap paginya sebelum mesin digunakan untuk memfilter sarden yang akan di packing untuk mengatasi kesalahan yang dapat terjadi pada saat packing”*

* Product transition
  + **Portability requirement**, cenderung berkutat dengan kesanggupan sebuah sistem untuk dioperasikan menggunakan sistem operasi atau environment yang berbeda-beda.

Contoh : *“Software harus dapat berjalan minimal pada platform windows 7, sistem operasi linux dan data dapat diakses melalui mobile android OS”*

* + **Reusability requirement,** menentukan bahwa sebuah software harus didesain untuk dapat digunakan kembali modulnya atau keseluruhan softwarenya untuk project kedepannya.

Contoh : *“Software informasi manajemen dokumen mahasiswa ini harus didesain dan diprogram agar nantinya dapat digunakan secara ulang untuk sistem informasi beasiswa kedepannya, yang direncanakan untuk dikembangkan tahun depan, modul tersebut harus dapat menyediakan informasi, dan akses dokumen yang telah dibuat dan tersimpan.”*

* + **Interoperability requirement,** fokus pada kemampuan software yang dibuat untuk dapat melakukan komunikasi dengan sistem lainnya melalui interfaces atau firmware peralatan keras lainnya.

Contoh : ”*Software pendeteksi kanker ini diharuskan untuk memproses data yang didapatkan dari mesin EKG dan mesin X-ray yang ada dengan mengikuti standar laboratorium yang ada”.*

* Alternative model
  + Alternative model menghasilkan lima faktor baru yaitu
    - **Verifiability requirement,** mendefinisikan metode verifikasi yang harus dipenuhi oleh software, biasanya mencakup modularitas, kemudahan dan kesesuaian metode verifikasi yang dilakuakn
    - **Expandibility requirement,** mencakup tentang kemampuan software untuk dikembangkan kedepannya, ditambahkan fiitur aplikasi atau yang lainnya
    - **Safety requiremetn,** didefinisikan untuk memastikan keselamatan pengguna yang bertindak sebagai operator jika ada error terjadi pada suatu sistem.

Contoh : “*Sebuah sistem diharuskan untuk mengontrol PH sebuah tanah dengan cara menyemprotkan cairan asam dengan pH tertentu pada jangka waktu tertentu, jika terjadi sebuah error pada sistem, diharuskan terdapat alarm dan penanganan pertama pada sistem komputer untuk memastikan keselamatan pekerja disekitar lokasi”*

* + - **Manageability requirement,** didefinisikan untuk mendefinisikan software harus memiliki sebuah tools administratif khusus untuk memodifkasi jalannya sistem software yang ada

Contoh : “*Sistem aplikasi TickFarm harus dapat mencatat log kelembaban udara dan tingkat pH tanah untuk dianalisa kemudian waktu,issue yang terjadi pada saat pengembangan harus dapat dikontrol pada sebuah development control board yang dapat memungkinkan modifikasi oleh anggota tim pengembang kemudian hari”.*

* + - **Survivability requirement**, mencakup kesanggupan atau waktu hidup sebuah sistem agar dapat digunakan secara terus menerus atau dalam jangka waktu yang lama. Requirement ini juga mendefinisikan waktu minimum sistem failure yang diperbolehkan.

Contoh : “*Software perbankan ini direncanakan untuk diakses selama 24 jam sehari dengan ekspektasi transaksi terjadi lebih dari 13000 kali per hari, dengan ini ditentukan survavibility requirement sebagai berikut : waktu down sistem tidak boleh lebih dari 30 menit terhitung pada saat anomali sistem dirasakan terjadi”*

Bab 4 – Komponen software quality assurance

* Pre-project componentes
  + Untuk memastikan bahwa komitmen untuk menyelesaikan project telah disepakati dengan memperhitungkan resource yang dibutuhkan dan budget yang dimiliki, dan untuk memastikan bahwa rencana pengembangan telah benar benar ditentukan secara matang
  + Komponennya adalah Contract Review, dan development and quality plans
  + Contract review, memiliki beberapa aktifitas diantaranya adalah,
    - Klarifikasi requirement dari customer
    - Melakukan review jadwal project dan resource yang dibutuhkan
    - Mengevaluasi kapasitas staff untuk menyelesaikan project
    - Mengevaluasi kapasitas customer untuk memenuhi kewajibannya
    - Mengevaluasi resiko developmentnya
  + Development and quality plans, lebih fokus kepada,
    - Jadwal
    - Manpower dan resource hardware yang dibutuhkan
    - Evaluasi resiko
    - Isu organisasi : anggota tim, subkontraktor, partnerships
    - Metodologi project, tools pengembangan
    - Rencana reuse software
* Software project lifecycle component
  + Tujuannyauntuk mendeteksi adanya desain dan programming error
  + Dibagi menjadi duas stage, yaitu
    - Development life cycle
    - Operation-maintenance
  + Memiliki komponen sebagai berikut,
    - Reviews, (1)
    - Expr
    - Expert opinion (1)
    - Software testing (1)
    - Software maintenance(2)
    - Support dari pihak ketiga (1)
* Infrastructure component for error prevention and improvements
  + Tujuan utamanya untuk menghilangkan atau setidaknya mengurangi tingkat error dengan dasar pengalaman organisasi tentang SQA
  + Memiliki beberapa komponen, yaitu
    - Prosedur dan instruksi kerja
    - Template dan checklist
    - Pelatihan staff, retraining dan sertifikasi
    - Tindakan preventif dan korektif
    - Manajemen konfigurasi
    - Kontrol dokumentasi
* Management SQA components
  + Tujuannya untuk mengkontrol aktivitas pengembangan dan maintenance dan awal dari managerial support yang dapat mencegah atau meminimalisir adanya jadwal/budget faillure. (mengontrol resource)
  + Komponennya adalah,
    - Project progress control
    - Software quality metrics
* SQA standards, system certification and assesment components
  + Tujuan utamanya adalah pemanfaatan pengetahuan tentang organisasi internasional (?), peningkatan kerjasama dengan organisasi lain, dan memastikan pencapaian kualitas sistem mencapai taraf yang dianggap layak.
  + (dari buku binder) tujuannya adalah, meningkatkan kemampuan staff, meningkatkan kapabilitas koordinasi antar staff, dan memastikan kesesuaian hasil testing dan standar mutu dari tim
  + Komponennya antara lain,
    - Quality management standards
    - Project process standards
* Human Component
  + Tujuan utamanya adalah untuk mendukung implementasi dari SQA components, mendeteksi penyimpangan dari SQA procedure, dan untuk menyarankan adanya improvement
  + Dibagi menjadi
    - Manager
    - Testing personel
    - SQA unit dan praktisi SQ
* Tingkat fundamental dalam SQ component, Human > Quality Infrastructure, management, standar > Lifecycle > Pre Project (?)

Bab 5 – Contract Review

* Contract review merupakan elemen software quality yang bertujuan untuk mengurangi kemungkinan terjadinya situasi yang tidak diinginkan.
* Ada beberapa situasi yang dapat membuat sebuah software company menandatangani kontrak dengan customernya, yaitu
  + Partisipasi melalui tender
  + Sumbit proposal sesuai dengan RFP milik customer
  + Mendapatkan pesanan langsung dari customer
  + Menerima permintaan internal dari departemen lain di suatu organisasi.
* Proses review sendiri terbagi menjadi dua tahap, yaitu
  + Proposal draft review – review proposal draft sebelum disubmit ke customer
    - Tujuan dari propsal draft review antara lain untuk memastikan hal hal dibawah ini sudah terlaksana
      * Requirement customer sudah diklarifikasi dan didokumentasikan
        + Meliputi functional requirement
        + Sistem operasi milik customer
        + Interfaces yang dibutuhkan
        + Kehandalan sistem
        + Kegunaan istem
        + Durasi garansi
        + Proposal untuk maintenance, dll
      * Langkah alternatif dalam menyelesaikan project sudah didefinisikan
        + Integrasi software yang telah dibeli dan akan digunakan kembali
        + Partnership
        + Subcontractor
        + Perbandingan alternatif yang layak
      * Aspek formal antara customer dan firma software telah dispesifikan (sudah ada kondisi kontrak dispesifikan)
        + Daftar dokumentasi yang harus dibuat
        + Partisipasi customer pada progress review, design review dan testing
        + Kirteria penyempurnaan project,
        + Prosedur untuk menagani komplain dari customer terhadap masalah yang terjadi
      * Identifikasi resiko pengembangan sudah didefinisikan
      * Estimasi jadwal dan budget project telah disiapkan
        + Man-days yang dibutuhkan untuk setiap fase project dan berapa besar biaya yang harus dikeluarkan
        + Manpower resource yang dibutuhkan untuk memenuhi kewajiban garansi yang diberikan
      * Pemeriksaan kesanggupan perusahaan software untuk menyelesaikan project telah dilakukan
        + Ketersediaan staff khusus
        + Ketersediaan resource komputer dan device pengembangan lainnya
        + Kemampuan untuk menaungi tuntutan kustomer requirement yang membutuhkan penggunaan tools khusus pada saat pengembangan program
      * Pemeriksaan kesanggupan customer untuk berkomitmen menyelesaikan project telah dilakukan
        + Kemampuan finansial
        + Suplai seluruh fasilitas, data dan kewajiban staff-staff yang terkait
        + Rekrutmen dan training staff yang sudah ada
      * Pendefinisian kondisi partisipasi dari partner dan subcontractor telah dibuat
        + Alokasi pertanggungjawaban untuk menyelesaikan bagian tugas untuk masing masing pihak, termasuk jadwal dan metode untuk koordinasi
        + Pengalokasian pembayaran termasuk bonus, pinalti diantara para partner
        + Jadwal pembayaran termasuk bonus dan pinalti kepada subcontractor
      * Definisi mengenai hak kekayaan intelektual yang digunakan dan yang telah dibuat
        + Mengamankan hak kepemilikan atas perangkat lunak yang dibeli dari orang lain.
        + Mengamankan hak kepemilikan data file yang dibeli dari pihak lain
        + Mengamankan hak kepemilikan software yang dibuat oleh supplier dan subcontractor selama periode pengembangan dan ketika digunakan oleh client
  + Contract draft review – review contract draft sebelum ditandatangani
    - Tujuan dari contract draft review adalah memastikan bahwa aktivitas dibawah ini telah dilakukan secara baik,
      * Sudah tidak ada lagi masalah yang kurang jelas solusinya pada contract draft
      * Semua nota kesepahaman antara keduabelah pihak pada proposal yang diajukan telah benar benar didokumentasikan dengan baik
      * Tidak boleh ada pembaharuan, perubahan atau penghapusan pada contract draft
* Implementasi contract review
  + Ada beberapa faktor yang mempengaruhi diperpanjangnya sebuah contract review, yaitu
    - Skala project
    - Kerumitan teknis dari project
    - Tingkat kemampuan staff dalam memahami dan mengerjakan project
    - Kerumitan susunan organisasi sebuah project
  + Contract review dilakukan oleh beberapa elemen, antara lain
    - Ketua atau anggota perwakilan dari tim proposal
    - Anggota dari tim proposal
    - Professional yang khusus didatangkan yang berasal dari luar perusahaan atau anggota perusahaan yang bukan bagian dari proposal tim
    - Tim expert yang sengaja didatangkan dari luar
  + Contract review juga dapat diimplementasikan pada major proposal, dengan karakteristik/pengertian dari major proposal adalah sebagai berikut
    - Project dengan skala yang sangat besar
    - Tingkat kerumitannya sangat tinggi
    - Membutuhkan staff professional dengan keahlian yang bermacam-macam bahkan terbilang baru bagi perusahaan
    - Kerumitan struktur organisasinya tinggi
  + Dalam implementasinya, tentu ada kesulitan pada major proposal, yaitu
    - Time pressure, deadline yang singkat dan dibutuhkan manajemen waktu yang ketat karena untuk mengerjakan project dibutuhkan keahlian baru, pendekatan baru dan kerumitan arsitektur yang tentunya membutuhkan waktu untuk menyelesaikannya
    - Untuk membuat contract review yang layak, dibutuhkan professional yang handal dibidangnya, yang berarti resource yang dikeluarkan akan semakin besar
    - Anggota tim contract review terbilang sangat sibuk, karena kerjanya tidak hanya terpetak dalam satu project, melainkan bisa mencakup banyak project
  + Dengan tingkat kesulitan yang tinggi, tentunya diperlukan strategi untuk menerapkan contract review pada major proposal, yaitu
    - Contract review harus dijadwalkan
    - Yang membawa jalannya contract review adalah tim itu sendiri
    - Harus ada ketua untuk contract team