

BAB II

LANDASAN TEORI

2.1 Sistem Informasi

Menurut Satzinger, Jackson, dan Burd (Pragita, C., Firdaus, Y., & Perdana, E.G., 2013) sistem informasi adalah sekumpulan dari komponen yang saling berhubungan dan berfungsi untuk mengumpulkan, memproses, menyimpan dan menyediakan *output* berupa informasi yang dibutuhkan untuk menyelesaikan tugas-tugas bisnis.

Menurut pendapat Jogiyanto (Syaroh, S., Utama, D.N., & Kurniawan, E., 2011) sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategis dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan.

Dari beberapa pendapat di atas, disimpulkan bahwa sistem informasi adalah sekumpulan komponen yang menyediakan *output* berupa informasi untuk mendukung kegiatan operasi dan bersifat manajerial untuk mencapai tujuan bisnis suatu organisasi.

2.2 Audit Sistem Informasi

Menurut Weber (Pragita, C., Firdaus, Y., & Perdana, E.G., 2013) mengemukakan bahwa audit sistem informasi adalah proses pengumpulan dan penilaian bukti-bukti untuk menentukan apakah sistem komputer dapat

mengamankan aset, memelihara integritas data, dapat mendorong pencapaian tujuan organisasi secara efektif dan menggunakan sumber daya secara efisien.

Sedangkan menurut Gondodiyoto & Hendarti (Syaroh, S., Utama, D.N., & Kurniawan, E., 2011) audit sistem informasi memberikan evaluasi yang bersifat independen atas kebijakan, prosedur, standar, pengukuran dan praktik untuk menjaga/mencegah informasi yang bersifat elektronik dari kehilangan, kerusakan, penelusuran yang tidak disengaja dan sebagainya.

Berdasarkan pendapat di atas, audit sistem informasi adalah suatu kegiatan yang mengumpulkan bukti-bukti, lalu dari bukti-bukti tersebut dijadikan bahan untuk evaluasi informasi yang bersifat elektronik demi pencapaian tujuan organisasi secara efektif.

2.2.1 Tujuan Audit Sistem Informasi

Menurut Gondodiyoto & Hendarti (Syaroh, S., Utama, D.N., & Kurniawan, E., 2011) tujuan audit sistem informasi dijelaskan sebagai berikut:

1. Untuk mengidentifikasi sistem yang ada pada tiap divisi/unit/departemen maupun yang digunakan secara menyeluruh.
2. Untuk dapat memahami seberapa besar sistem informasi dapat mendukung kebutuhan strategis, kegiatan operasional departemen/unit/divisi, kelompok kerja maupun para petugas dalam melaksanakan kegiatannya.
3. Untuk mengetahui bidang atau area, fungsi, kegiatan atau proses bisnis yang didukung dengan sistem serta teknologi informasi yang ada.
4. Untuk menganalisa tingkat kepentingan data/informasi yang dihasilkan oleh sistem dalam rangka mendukung kebutuhan para pemakainya.

5. Untuk mengetahui keterkaitan antara sistem pengolahan dan transfer informasi.
6. Untuk mengidentifikasi kesenjangan antara sistem dan kebutuhan.
7. Untuk membuat peta dari alur informasi yang ada.

2.3 Tata Kelola Teknologi Informasi

Menurut Grembergen dan Haes (2009:1), tata kelola teknologi informasi adalah konsep yang relative baru dalam literature dan semakin mendapatkan ketertarikan lebih banyak dalam dunia akademik dan praktisi. Tata kelola teknologi informasi merupakan penentuan dan pelaksanaan atau implementasi dari proses, struktur dan mekanisme relasional yang memudahkan pihak bisnis dan teknologi informasi dalam melaksanakan tanggung jawab mereka dalam mendukung keselarasan bisnis dan teknologi informasi serta memberikan suatu nilai tersendiri dari investasi teknologi informasi bagi bisnis.

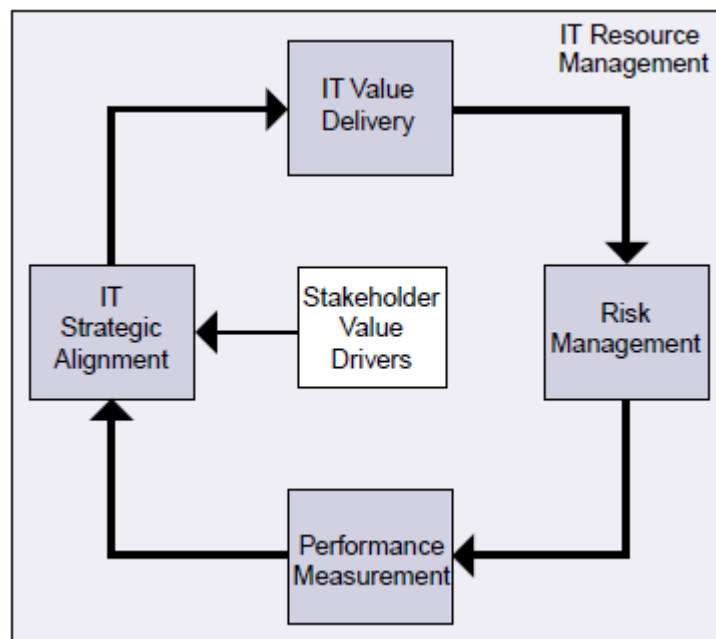
Menurut Weill and Ross (2004:8), tata kelola teknologi informasi fokus kepada pada penentuan siapa yang memiliki wewenang atas pengambilan keputusan dan menentukan kerangka kerja yang dapat dipertanggungjawabkan dalam pemanfaatan TI di perusahaan.

Dari beberapa definisi di atas, dapat disimpulkan bahwa tata kelola teknologi informasi menekankan pada siapa yang berwenang dalam pengambilan keputusan, bagaimana prosedur untuk pengambilan dan pelaksanaan keputusan tersebut serta langkah kerja yang digunakan agar implementasi strategi TI selaras dengan pengembangan strategi dan tujuan bisnis perusahaan.

2.4 Fokus Area Tata Kelola TI

Menurut ITGI (2003) pada dasarnya, tata kelola TI fokus kepada 2 (dua) hal: nilai layanan TI terhadap bisnis dan mitigasi risiko TI. Nilai layanan TI terhadap bisnis didorong oleh keselarasan strategi TI dengan strategi bisnis. Mitigasi risiko TI didorong oleh menanankan akuntabilitas ke organisasi.

Terdapat 5 (lima) bidang fokus utama tata kelola TI yang semuanya didorong oleh *stakeholder value*. Dua diantaranya merupakan hasil yaitu: *value delivery* dan *risk management* dan yang lainnya adalah penggerak yaitu: *strategic alignment*, *resource management* dan *performance measurement* seperti terlihat pada Gambar 2.1.



Gambar 2.1 Fokus Area Tata Kelola TI

Sumber: ITGI (2003)

Tata kelola TI juga merupakan siklus hidup berkesinambungan (*continuous lifecycle*), sehingga untuk penerapannya dapat dimulai dari tahap mana pun. Pada umumnya dimulai dari strategi dan keselarasan strategi TI dengan

strategi bisnis perusahaan (*IT Strategic Alignment*). Berikutnya pada saat dilakukan implementasi akan diberikan nilai strategi (*IT Value Delivery*). Selanjutnya diperlukan mitigasi terhadap resiko yang mungkin muncul (*Risk Management*). Secara berkala dan berkesinambungan strategi dimonitor dan hasilnya diukur, dilaporkan dan ditindaklanjuti untuk mengetahui kinerja bisnis yang sedang berjalan (*Performance Measurement*). Umumnya strategi dievaluasi kembali dan diselaraskan (jika diperlukan) setahun sekali. Semua bidang tata kelola TI ini berada dalam ruang lingkup *IT Resource Management*. Berikut adalah penjelasan masing-masing bidang fokus utama tata kelola TI, seperti pada Gambar 2.1:

1. *IT Strategic Alignment*, fokus pada keselarasan strategi TI dengan strategi bisnis dan dapat memberikan solusi bersama (*collaborative solutions*). TI dipandang strategis oleh perusahaan karena dapat memberikan nilai tambah (*value added*) pada produk dan layanan yang dihasilkan, membantu meningkatkan posisi daya saing perusahaan, meningkatkan efisiensi serta meningkatkan efektivitas manajerial. Keselarasan ini juga mencakup keselarasan operasional TI dengan operasional bisnis.
2. *IT Value Delivery*, berkonsentrasi pada optimasi biaya dan memastikan TI dapat memberikan nilai kepada bisnis dalam bentuk keunggulan kompetitif, pemenuhan layanan secara tepat waktu, kepuasan pelanggan, meningkatkan produktivitas karyawan dan keuntungan.
3. *Risk Management*, menangani pengamatan aset TI dan pemulihan bencana. Manajemen resiko dapat berupa mitigasi yaitu mengurangi risiko yang terjadi

setelah bencana, *transfer* yaitu membagi atau memindahkan risiko dan *accept* yaitu menerima risiko yang ada dan memonitor dampaknya.

4. *Resource Management*, melakukan optimasi pengetahuan dan infrastruktur TI serta mengoptimalkan investasi, penggunaan dan alokasi sumber daya TI (SDM, aplikasi, teknologi, fasilitas dan data) dalam melayani kebutuhan bisnis. Salah satu tantangan terbesar dalam beberapa tahun terakhir adalah melakukan *outsourcing* dan mengelola layanan *outsourcing*.
5. *Performance Measurement*, melakukan penelusuran proyek yang telah dilakukan dan memonitor layanan TI. *Balance Scorecard* menerjemahkan strategi ke dalam tindakan untuk mencapai tujuan dengan sistem pengukuran kinerja yang melampaui sistem akuntansi konvensional, mengukur hubungan-hubungan dan aset berbasis pengetahuan yang diperlukan untuk bersaing di era informasi: fokus pelanggan, efisiensi proses dan kemampuan untuk belajar dan tumbuh.

2.5 IT Service Management

Menurut itSMF (2011:4) *service* atau layanan adalah cara memberikan manfaat kepada pelanggan dengan memfasilitasi hasil yang ingin dicapai oleh pelanggan tanpa harus menanggung biaya atau risiko tertentu. Contoh sederhana dari hasil yang ingin dicapai oleh pelanggan (*customer outcomes*) dengan memanfaatkan fasilitas layanan TI: “Staf penjualan menghabiskan lebih banyak waktu berinteraksi dengan pelanggan” difasilitasi oleh “layanan *remote access* yang memungkinkan akses yang handal ke sistem penjualan perusahaan dari laptop staf penjualan”. Hasil yang ingin dicapai oleh pelanggan adalah alasan

kenapa pelanggan membeli atau menggunakan layanan tersebut. Manfaat layanan kepada pelanggan secara langsung tergantung pada seberapa baik memfasilitasi hasil (*outcomes*) ini.

Menurut itSMF (2011:5) *service management* atau manajemen pelayanan adalah sekumpulan kapabilitas khusus organisasi untuk memberikan nilai manfaat kepada pelanggan dalam bentuk layanan. Kapabilitas ini mencakup seluruh proses, metode, fungsi, peran dan aktivitas-aktivitas yang dilakukan oleh penyedia layanan dalam memberikan layanan kepada pelanggan. Manajemen pelayanan tidak hanya terbatas pada pemberian layanan kepada pelanggan namun juga mencakup siklus hidup (*lifecycle*) seluruh komponen infrastruktur dan proses mulai dari strategi (*strategy*), desain (*design*), transisi (*transition*), operasional (*operation*) dan perbaikan terus-menerus (*continual improvement*). Input dari *service management* adalah sumber daya (*resources*) dan kapabilitas (*capabilities*) yang merupakan aset dari penyedia layanan. Sedangkan outputnya adalah layanan (*services*) yang memberikan manfaat kepada pelanggan. Manajemen pelayanan yang efektif merupakan aset strategis bagi penyedia layanan sehingga dapat menjalankan *core business*-nya dalam menyediakan layanan yang dapat memberikan manfaat dengan memfasilitasi hasil yang ingin dicapai oleh pelanggan.

2.6 ITIL

ITIL (*Information Technology Infrastructure Library*) adalah kerangka kerja umum yang menggambarkan *best practice* untuk *IT Service Management* (ITSM). ITIL menyediakan panduan bagi penyedia layanan (*service provider*)

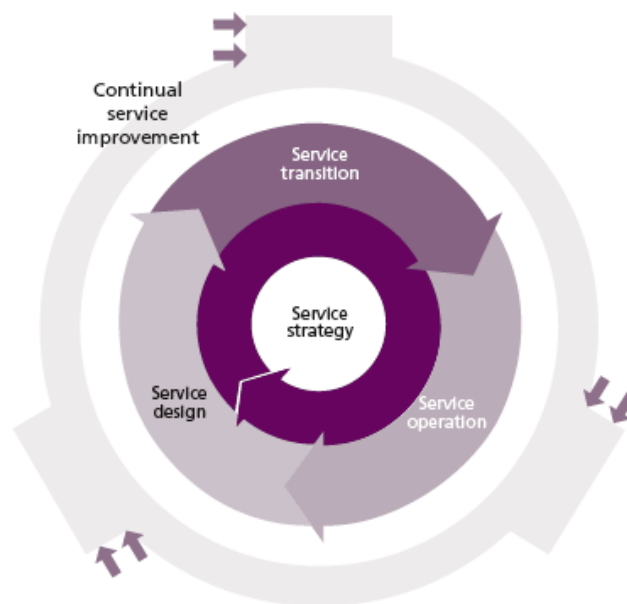
dalam mendukung penyediaan kualitas layanan TI dan proses, fungsi serta kapabilitas lainnya yang diperlukan. ITIL telah digunakan oleh ratusan organisasi di seluruh dunia dan menawarkan panduan *best practice* yang berlaku umum untuk semua organisasi yang menyediakan layanan. ITIL bukan merupakan suatu standar yang harus diikuti, melainkan panduan yang harus dibaca dan dipahami serta digunakan untuk menciptakan nilai bagi penyedia layanan dan juga pelanggannya (Cabinet Office, 2011a:3).

ITIL berfokus pada pengukuran dan perbaikan secara terus-menerus terhadap kualitas layanan TI yang diberikan, baik dari perspektif bisnis maupun perspektif pelanggan. Fokus ini merupakan faktor utama keberhasilan penerapan ITIL di seluruh dunia dengan menerapkan teknik dan proses di dalam organisasi mereka telah memberikan kontribusi positif terhadap hasil yang ingin dicapai (itSMF, 2011:6). Manfaat penerapan ITIL untuk organisasi antara lain:

1. Meningkatkan kepuasan pengguna dan pelanggan terhadap layanan TI
2. Meningkatkan ketersediaan layanan (*service availability*) yang secara langsung berdampak pada meningkatnya keuntungan dan pendapatan bisnis
3. Penghematan keuangan dari berkurangnya pekerjaan berulang (*rework*), kehilangan waktu, serta peningkatan manajemen dan penggunaan sumber daya
4. Mempercepat waktu peluncuran produk dan layanan baru ke pasar
5. Pengambilan keputusan yang lebih baik dan mengurangi risiko

2.6.1 ITIL V3 2011 Service Lifecycle

ITIL V3 2011 membagi layanan dalam 5 (lima) service lifecycle yakni Service Strategy, Service Design, Service Transition, Service Operation dan Continual Service Improvement. Kelima tahapan service lifecycle tersebut, seperti terlihat pada Gambar 2.2, menggunakan desain “hub-and-spoke” dimana Service Strategy sebagai "hub" dan Service Design, Service Transition dan Service Operation sebagai tahapan service lifecycle yang terus bergulir atau “spokes”. Sementara Continual Service Improvement mengelilingi dan mendukung semua tahapan service lifecycle (Cabinet Office, 2011a).

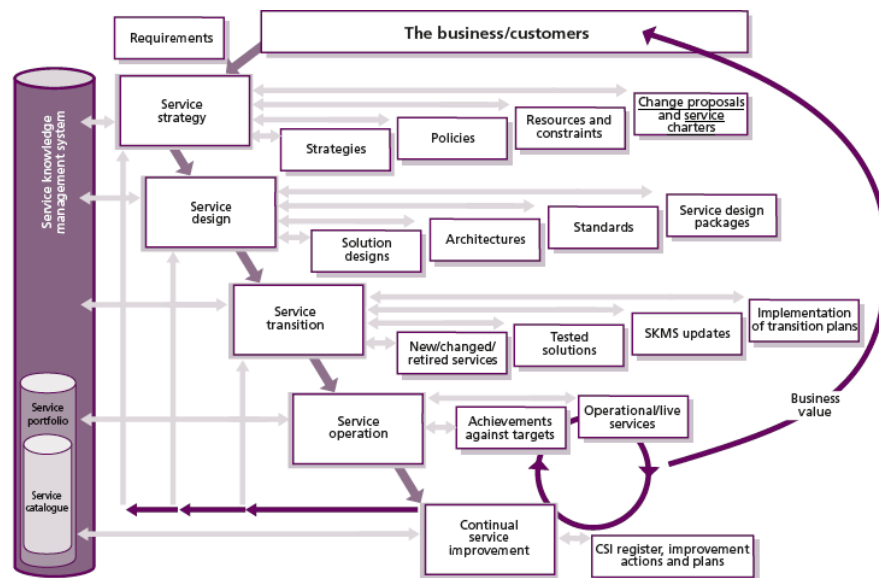


Gambar 2.2 ITIL V3 *Service Lifecycle*

Sumber: Cabinet Office. (2011a)

Integrasi tahapan *service lifecycle* dengan beberapa kunci penghubung, *input* dan *output* yang ada pada masing-masing tahapan terlihat seperti Gambar 2.3. Portofolio layanan (*service portofolio*) menjadi tulang punggung (*‘the spine’*) dari seluruh *service lifecycle*. *Service lifecycle* dimulai dari adanya

kebutuhan bisnis (*business requirements*). Kebutuhan bisnis diidentifikasi dan disepakati pada tahap *Service Strategy*. Tahap berikutnya adalah *Service Design* dimana solusi layanan dibuat bersama dalam *Service Design Package* yang berisikan hal-hal yang akan digunakan pada tahap berikutnya.



Gambar 2.3 Integrasi Tahapan *Service Lifecycle*

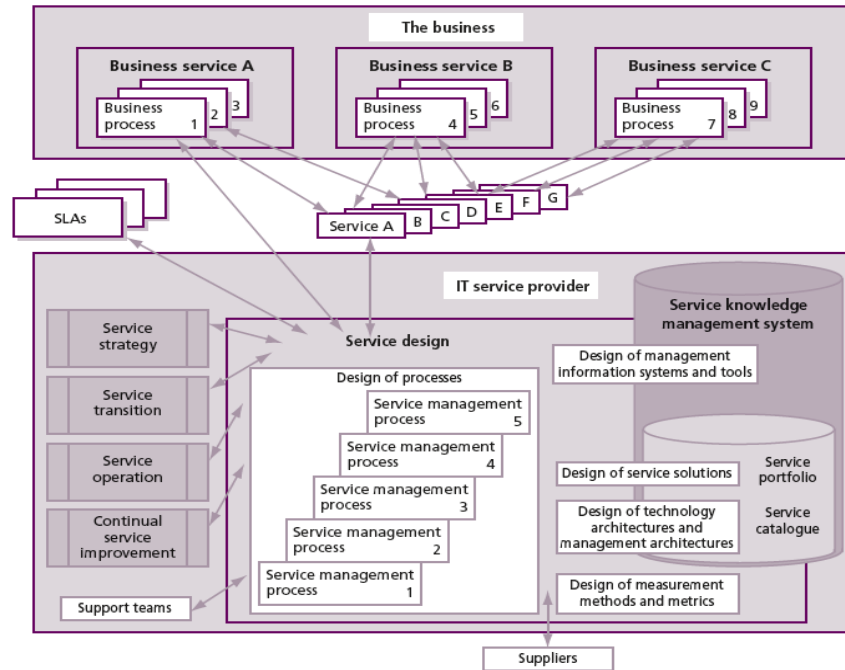
Sumber: Cabinet Office. (2011a)

Tahap selanjutnya adalah *Service Transition* dimana solusi layanan dievaluasi, diuji dan divalidasi. Demikian halnya dengan layanan SKMS (*Service Knowledge Management System*) yang harus diperbarui. Pada tahap ini dilakukan implementasi rencana transisi ke lingkungan operasional (*live environment*). Pada tahap *Service Operation*, pencapaian hasil dibandingkan terhadap target yang telah ditetapkan sebelumnya. Sedangkan pada tahap *Continual Service Improvement* akan dilakukan perbaikan terhadap kekurangan atau kegagalan dimana pun dalam setiap tahap siklus hidup layanan. Integrasi tahapan-tahapan ini akan dapat memberikan nilai bisnis (*business value*) kepada organisasi atau pelanggan. Karena merupakan siklus hidup maka prosesnya akan berulang

kembali ke tahapan *Service Strategy* yang akan mengakomodasi perubahan-perubahan proposal dan layanan terhadap pelanggan.

2.6.2 ITIL Service Design

Tujuan dari tahapan *Service Design* dalam *service lifecycle* adalah untuk membantu organisasi dalam merancang suatu layanan TI agar seefektif mungkin sehingga diharapkan hanya ada sedikit perbaikan yang harus dilakukan selama siklus hidup berlangsung. Namun demikian perbaikan berkesinambungan harus tetap menjadi bagian tak terpisahkan dalam semua aktivitas *Service Design* untuk memastikan bahwa solusi dan desain menjadi lebih efektif dari waktu ke waktu serta mampu mengidentifikasi tren perubahan bisnis yang menawarkan peluang perbaikan. *Service Design* menyediakan panduan untuk desain layanan TI yang tepat dan inovatif yang mampu memenuhi kebutuhan bisnis saat ini dan masa depan. Tahapan *Service Design* dimulai dari persiapan kebutuhan bisnis baru atau perubahan kebutuhan bisnis dan diakhiri oleh pengembangan solusi layanan (Cabinet Office, 2011a). Cakupan dari *Service Design* terlihat pada Gambar 2.4.



Gambar 2.4 Ruang Lingkup *Service Design*

Sumber: Cabinet Office. (2011a)

Gambar 2.4 menunjukkan bagaimana TI dan bisnis berinteraksi melalui penyediaan layanan dan bagaimana pekerjaan *Service Design* merupakan bagian dari keseluruhan *service lifecycle*. Gambar 2.4 juga menunjukkan lima aspek individu (*individual aspects*) dari *Service Design*, yakni:

1. *Service solutions for new or changed services*
2. *The management information systems and tools*
3. *The technology architectures and management architectures*
4. *The processes required*
5. *The measurement methods and metrics*

Sedangkan proses-proses yang tercakup dalam *Service Design* menurut Cabinet Office (2011a) adalah

1. *Design coordination*
2. *Service catalogue management*

3. *Service level management*
4. *Availability management*
5. *Capacity management*
6. *IT service continuity management (ITSCM)*
7. *Information security management*
8. *Supplier management*

Proses-proses di atas bertanggung jawab dalam menyediakan informasi kunci (*key information*) untuk desain solusi layanan baru atau perubahan layanan.

2.6.3 ITIL Service Transition

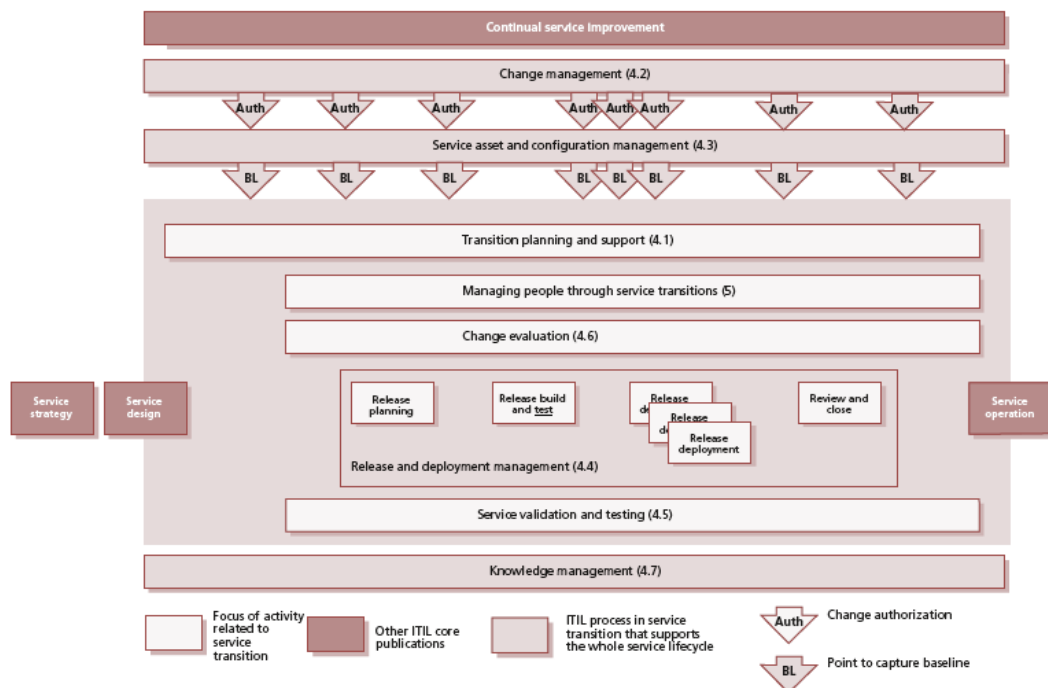
Tujuan dari tahapan *Service Transition* dalam *service lifecycle* adalah untuk membantu organisasi dalam membuat perencanaan dan mengelola perubahan layanan secara efisien dan efektif, mengelola risiko yang terjadi karena perubahan layanan, serta menempatkan layanan ke lingkungan operasional dengan sukses. *Service Transition* menyediakan panduan dalam pengembangan dan peningkatan kapabilitas untuk transisi layanan baru atau perubahan layanan ke lingkungan operasional termasuk rilis perencanaan, pembuatan, pengujian, evaluasi dan *deployment* layanan (Cabinet Office, 2011b).

Cabinet Office (2011b) membagi proses-proses dalam *Service Transition* ke dalam dua kelompok, yaitu:

1. Proses yang mendukung seluruh *service lifecycle*, meliputi:
 - 1) *Change management*
 - 2) *Service asset and configuration management*
 - 3) *Knowledge management*
2. Proses yang aktivitasnya fokus pada *Service Transition*, meliputi:

- 1) *Transition planning and support*
- 2) *Release and deployment management*
- 3) *Service testing and validation*
- 4) *Change evaluation*

Proses-proses yang tercakup pada *Service Transition* dan tahapan *service lifecycle* lainnya terlihat seperti pada Gambar 2.5 di bawah ini.



Gambar 2.5 Ruang Lingkup *Service Transition*

Sumber: Cabinet Office. (2011b)

2.6.4 ITIL Service Operation

Tujuan dari tahapan *Service Operation* dalam *service lifecycle* adalah membantu organisasi dalam menjaga kepuasan dan kepercayaan bisnis melalui penyampaian (*delivery*) layanan TI yang efektif dan efisien, meminimasi dampak gangguan layanan terhadap aktivitas bisnis sehari-hari serta memastikan bahwa akses ke layanan TI hanya diberikan kepada pihak yang berwenang menerima

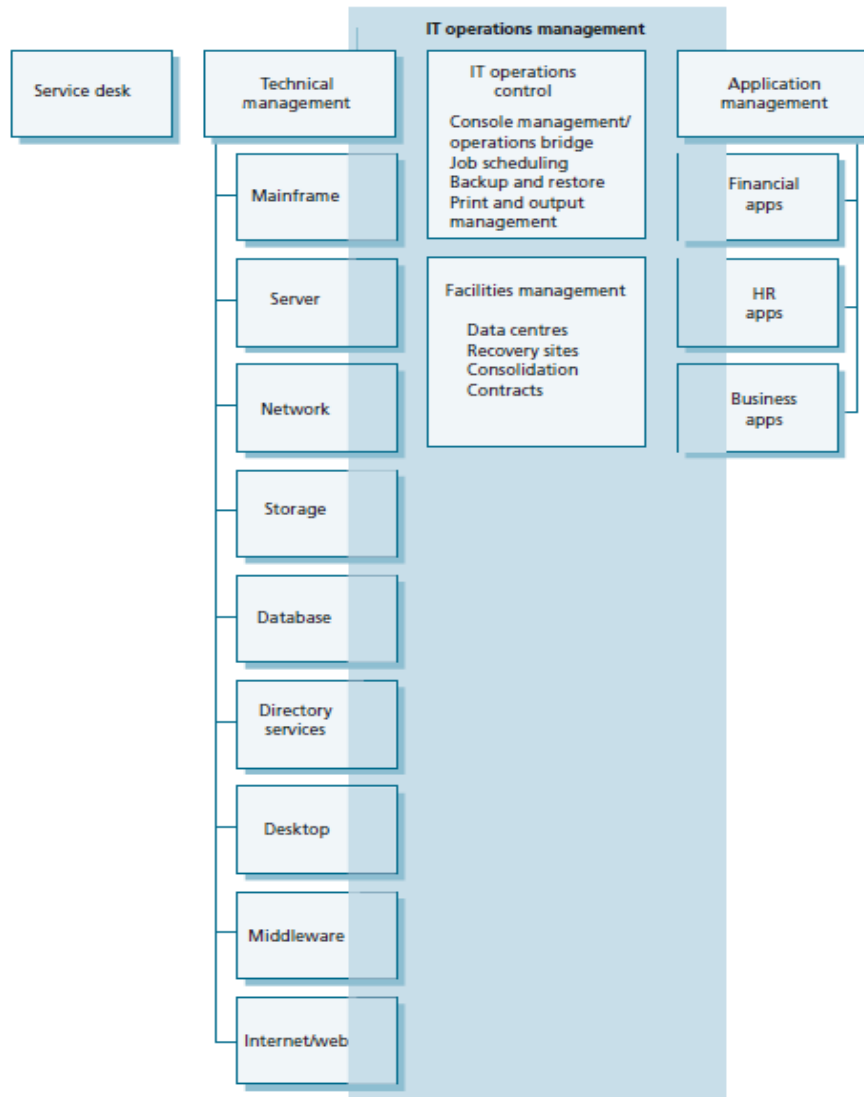
layanan TI tersebut. *Service Operation* menggambarkan proses, fungsi, organisasi, *tools* yang digunakan untuk mendukung aktivitas yang sedang berlangsung dalam memberikan dan mendukung layanan (Cabinet Office, 2011c).

Proses-proses yang tercakup dalam *Service Operation* menurut Cabinet Office (2011c) adalah:

1. *Event management*
2. *Incident management*
3. *Request fulfilment*
4. *Problem management*
5. *Access management*

Selain memiliki beberapa proses, *Service Operation* juga memiliki beberapa fungsi (*function*). Fungsi adalah sebuah team atau sekelompok orang dan *tools* atau *resources* lain yang digunakan untuk melaksanakan satu atau lebih proses atau aktivitas. Empat fungsi *Service Operation* seperti terlihat pada Gambar 2.6 adalah:

1. *Service desk*
2. *Technical management*
3. *IT operations management*
4. *Application management*



Gambar 2.6 Fungsi Pada *Service Operation*

Sumber: Cabinet Office. (2011c)

2.7 COBIT

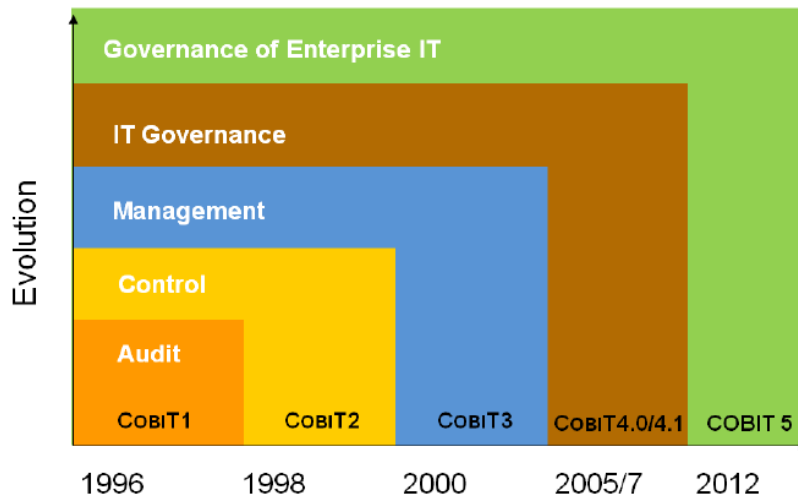
Sarbanes-Oxley dan *Committee of Sponsoring Organizations of the Treadway Commission* (COSO) di Amerika Serikat, *the Combined Code on Governance* di Inggris dan *the Organization for Economic Cooperation and Development* (OECD) *Principles of Corporate Governance* di Eropa telah menentukan panduan untuk tata kelola perusahaan. Untuk teknologi informasi,

COBIT telah menjadi standar global untuk tata kelola TI. COBIT menyediakan kerangka kerja yang baik untuk mengontrol TI sesuai dengan panduan Sarbanes-Oxley dan standar tata kelola global lainnya. Organisasi di seluruh dunia menggunakan prinsip-prinsip yang ditetapkan dalam COBIT untuk meningkatkan kinerja TI (Gallegos, F., & Senft, S., 2009).

COBIT (*Control Objective for Information and Related Technology*) adalah kerangka kerja (*framework*) tata kelola TI yang merupakan sekumpulan pengukuran yang diakui secara umum untuk proses pengelolaan TI yang dikeluarkan oleh ISACA (*Information Systems Audit and Control Association*) sebuah organisasi profesi internasional di bidang tata kelola TI.

2.7.1 COBIT 5

Menurut ISACA (2012a), COBIT 5 merupakan *update* dari COBIT versi sebelumnya yaitu COBIT 4.1 ditambahkan dengan Val IT 2.0 dan Risk IT. Val IT adalah framework tata kelola untuk menciptakan nilai bisnis dari investasi TI, sedangkan Risk IT adalah framework yang ditujukan untuk mengisi gap antara kerangka kerja manajemen risiko yang umum dan yang detail. COBIT pertama kali dirilis tahun 1996 awalnya untuk bidang audit. COBIT 2 dalam bidang kontrol dirilis tahun 1999 sampai akhirnya COBIT terbaru COBIT 5 bidang tata kelola TI organisasi dirilis tahun 2012. Gambar 2.7 berikut ini menunjukkan evolusi dari COBIT.

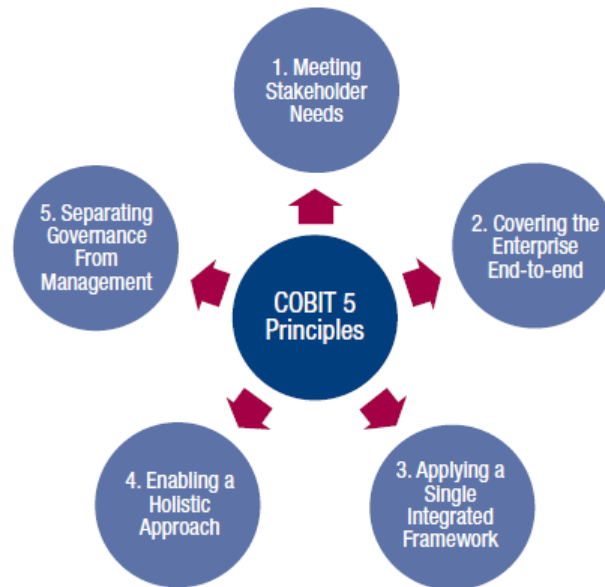


Gambar 2.7 Evolusi COBIT

Sumber: ISACA (2012a)

COBIT 5 menyediakan kerangka kerja komprehensif dalam membantu organisasi untuk pencapaian sasaran atau tujuan dalam tata kelola dan manajemen TI organisasi. COBIT 5 dapat membantu organisasi dalam mendapatkan nilai TI yang optimal dengan menjaga keseimbangan antara manfaat yang didapat, tingkat risiko yang minimal dan penggunaan sumber daya. COBIT 5 memberikan peluang TI dikelola dan diatur secara menyeluruh (*holistic*) untuk kepentingan seluruh organisasi, berperan dalam bisnis organisasi secara *end-to-end*, dan bertanggung jawab pada bidang fungsional TI karena TI berhubungan dengan kepentingan *stakeholder* internal maupun eksternal. COBIT 5 adalah kerangka kerja umum dan dapat digunakan untuk semua organisasi baik organisasi kecil maupun besar, organisasi komersial maupun sosial atau organisasi publik (ISACA, 2012b:13).

Terdapat 5 (lima) prinsip kunci dalam COBIT 5 seperti terlihat pada Gambar 2.8 dibawah ini:



Gambar 2.8 COBIT 5 Principles

Sumber: ISACA (2012b)

1. Prinsip 1: *Meeting Stakeholder Needs*

Memenuhi kebutuhan *stakeholder* - keberadaan organisasi untuk menciptakan nilai/manfaat untuk para *stakeholder* dengan menjaga keseimbangan antara manfaat yang didapat, tingkat risiko yang minimal, dan penggunaan sumber daya.

2. Prinsip 2: *Covering the Enterprise End-to-end*

Mencakup keseluruhan organisasi - COBIT 5 mengintegrasikan tata kelola TI organisasi ke dalam tata kelola organisasi:

- 1) Mencakup semua fungsi dan proses dalam perusahaan, dengan tidak hanya fokus pada fungsi TI tetapi memperlakukan dan menangani informasi dan teknologi sebagai aset yang sama dengan aset perusahaan lainnya.

- 2) Mempertimbangkan pelaku tata kelola dan manajemen TI secara menyeluruh baik teknologi ataupun orangnya, internal atau external perusahaan.
3. Prinsip 3: *Applying a Single, Integrated Framework*

Kerangka kerja terintegrasi - terdapat banyak standar dan *best practice* yang berkaitan dengan TI dimana masing-masing memberikan panduan aktivitas TI. COBIT 5 selaras dengan standar dan kerangka kerja lainnya seperti COSO, COSO ERM, ISO/IEC 9000, ISO/IEC 31000 untuk *enterprise* dan ISO/IEC 38500, ITIL, ISO/IEC 27000 series, TOGAF, PMBOK/PRINCE2, CMMI untuk *IT-related*. Oleh karenanya COBIT 5 menjadi kerangka kerja menyeluruh untuk tata kelola dan manajemen TI organisasi.
4. Prinsip 4: *Enabling a Holistic Approach*

Menggunakan pendekatan menyeluruh - tata kelola dan manajemen TI organisasi yang efisien dan efektif memerlukan suatu pendekatan menyeluruh dengan mempertimbangkan interaksi antar komponen. COBIT 5 mendefinisikan 7 (tujuh) kategori *enabler* dalam mendukung implementasi suatu sistem tata kelola dan manajemen yang komprehensif untuk TI organisasi. Tujuh kategori *enabler* tersebut adalah *Principles, Policies and Frameworks, Processes, Organisational Structures, Culture, Ethics and Behaviour, Information, Services, Infrastructure and Applications* serta *People, Skills and Competencies*.
5. Prinsip 5: *Separating Governance From Management*

Memisahkan tata kelola dari manajemen - kerangka kerja COBIT 5 membedakan dengan jelas antara tata kelola dengan manajemen. Keduanya memiliki jenis aktivitas yang berbeda, memerlukan struktur organisasi yang berbeda dan memiliki tujuan yang berbeda.

Perbedaan mendasar antara tata kelola (*governance*) dengan manajemen (*management*) adalah:

1) Governance

“Governance ensures that stakeholder needs, conditions and options are evaluated to determine balanced, agreed-on enterprise objectives to be achieved; setting direction through prioritisation and decision making; and monitoring performance and compliance against agreed-on direction and objectives (EDM)” (ISACA, 2012b:14).

Dari definisi di atas tata kelola memastikan bahwa kebutuhan *stakeholder*, kondisi dan pilihan-pilihan dievaluasi untuk menentukan keseimbangan, memastikan tujuan organisasi yang telah disepakati dapat tercapai, memberikan arahan melalui prioritas dan pengambilan keputusan; dan memonitor kinerja dan kepatuhan terhadap arahan dan tujuan yang telah disepakati. Pada umumnya tata kelola menjadi tanggung jawab dewan direksi di bawah kepemimpinan seorang *chairperson*.

2) Management

“Management plans, builds, runs and monitors activities in alignment with the direction set by the governance body to achieve the enterprise objectives (PBRM)” (ISACA, 2012b:14).

Dari penjelasan di atas, manajemen merencanakan, membangun, menjalankan dan memonitor aktivitas agar selaras dengan arah yang telah ditetapkan oleh badan tata kelola dalam mencapai tujuan organisasi. Pada umumnya manajemen merupakan tanggung jawab dari manajemen eksekutif (*executive management*) di bawah kepemimpinan CEO (*Chief Executive Officer*).

Dari penjelasan di atas dapat disimpulkan bahwa COBIT 5 menyatukan 5 (lima) prinsip yang digunakan oleh organisasi untuk membangun kerangka kerja tata kelola dan manajemen yang efektif berdasarkan tujuh kategori *enabler* yang dapat mengoptimalkan investasi TI dan untuk kepentingan *stakeholder*.

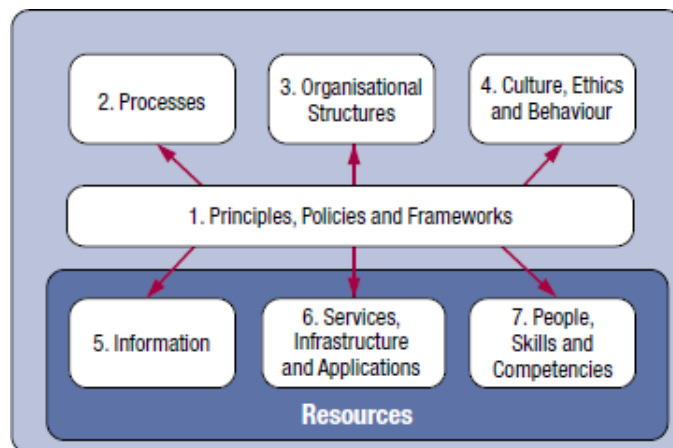
2.7.1.1 COBIT 5 Enablers

Menurut ISACA (2012b:27), *Enabler* adalah faktor yang mempengaruhi tata kelola dan manajemen TI. Tujuan TI mendefinisikan *enabler* mana yang musti dicapai.⁷ (tujuh) kategori *enabler* COBIT 5 adalah:

1. *Principles, policies and frameworks* panduan untuk menerjemahkan perilaku yang diinginkan ke dalam panduan praktis untuk manajemen sehari-hari.
2. *Processes* menggambarkan beberapa praktek dan kegiatan terorganisir untuk mencapai tujuan tertentu dan menghasilkan beberapa output dalam mendukung pencapaian tujuan keseluruhan yang berkaitan dengan IT.
3. *Organisational structures* kunci entitas pengambilan keputusan dalam suatu perusahaan.

4. *Culture, ethics and behavior* individu dan perusahaan yang sangat sering diremehkan sebagai faktor keberhasilan dalam kegiatan tata kelola dan manajemen.
5. *Informationa* dalam menyebar di seluruh organisasi dan mencakup semua informasi yang dihasilkan dan digunakan oleh perusahaan. Informasi diperlukan untuk menjaga jalannya organisasi dan tata kelola yang baik, tetapi pada tingkat operasional, informasi ini berupa produk utama dari perusahaan.
6. *Services, infrastructure and applications* termasuk infrastruktur, teknologi dan aplikasi yang menyediakan perusahaan dengan pengolahan dan jasa teknologi informasi.
7. *People, skills and competencies* terkait dengan orang-orang dan keahlian yang dibutuhkan untuk berhasil menyelesaikan semua kegiatan dan untuk membuat keputusan yang benar dan mengambil tindakan korektif.

Gambar 2.9 merupakan 7 (tujuh) kategori *enabler* dalam implementasi COBIT 5.



Gambar 2.9 COBIT 5 *Enablers*

Sumber: ISACA (2012b)

2.7.1.2 Area, Domain dan Proses COBIT 5

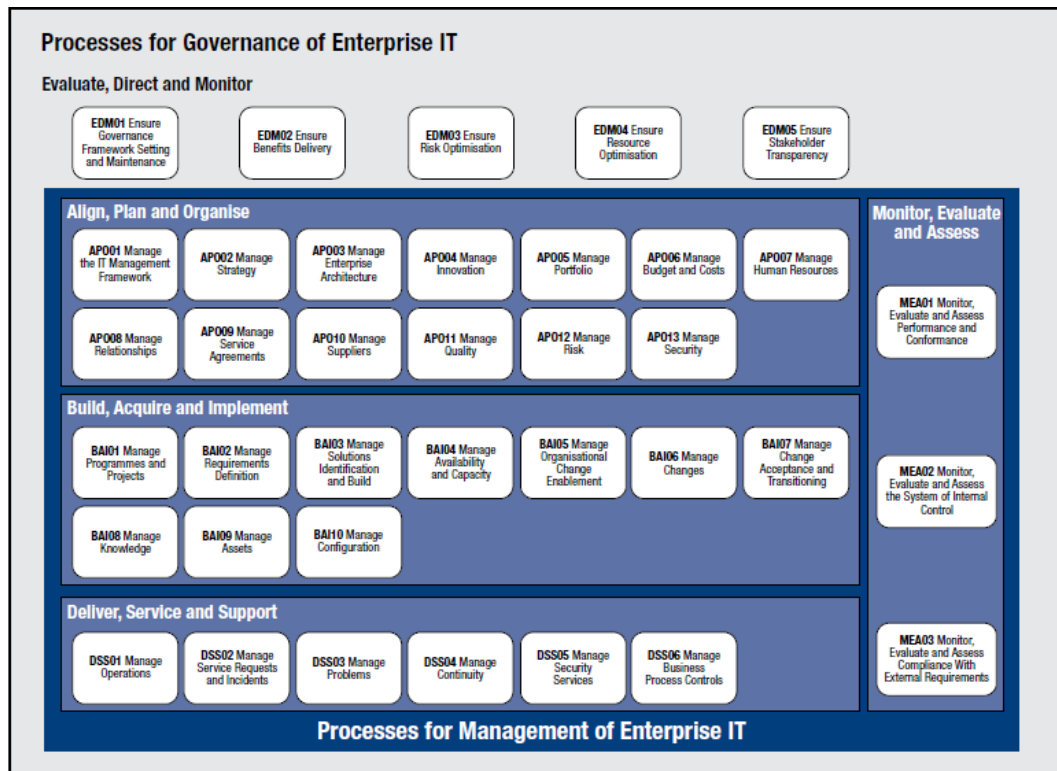
COBIT 5 memiliki 2 area aktivitas utama, 6 *domain*, 37 proses, 210 *process practice* dan 1112 aktivitas. Dua area aktivitas utama yakni area *Governance* dan *Management*. Area *governance* memiliki satu *domain* yakni EDM (*Evaluate, Direct, Monitor*) dengan 5 (lima) proses. Setiap proses memiliki beberapa *process practice* atau *governance practice*. Sedangkan area *management* (PBRM) terdiri dari 4 (empat) *domain* yakni APO (*Align, Plan and Organise*), BAI (*Build, Acquire, Implement*), DSS (*Deliver, Service and Support*) dan MEA (*Monitor, Evaluate and Assess*) dengan total 32 proses. Setiap proses memiliki beberapa *process practice* atau *management process* (ISACA, 2012b:33). Berikut 37 proses yang ada pada COBIT 5.

1. Domain EDM (*Evaluate, Direct, Monitor*), terdiri dari 5 (lima) proses:
 - 1) EDM01 *Ensure governance foramework setting and maintenance*
 - 2) EDM02 *Ensure benefits delivery*
 - 3) EDM03 *Ensure risk optimization*
 - 4) EDM04 *Ensure resource optimization*
 - 5) EDM05 *Ensure stakeholder transparency*
2. Domain APO (*Align, Plan and Organise*) terdiri dari 13 (tiga belas) proses:
 - 1) APO01 *Manage the IT management framework*
 - 2) APO02 *Manage strategy*
 - 3) APO03 *Manage enterprise architecture*
 - 4) APO04 *Manage innovation*
 - 5) APO05 *Manage portfolio*

- 6) APO06 *Manage budget and costs*
 - 7) APO07 *Manage human resources*
 - 8) APO08 *Manage relationships*
 - 9) APO09 *Manage service agreements*
 - 10) APO10 *Manage suppliers*
 - 11) APO11 *Manage quality*
 - 12) APO12 *Manage risk*
 - 13) APO13 *Manage security*
3. Domain BAI (*Build, Acquire, Implement*) terdiri dari 10 (sepuluh) proses:
- 1) BAI01 *Manage programmes and projects*
 - 2) BAI02 *Manage requirements definition*
 - 3) BAI03 *Manage solutions identification and build*
 - 4) BAI04 *Manage availability and capacity*
 - 5) BAI05 *Manage organisational change enablement*
 - 6) BAI06 *Manage changes*
 - 7) BAI07 *Manage change acceptance and transitioning*
 - 8) BAI08 *Manage knowledge*
 - 9) BAI09 *Manage assets*
 - 10) BAI10 *Manage configuration*
4. Domain DSS (*Deliver, Service and Support*) terdiri dari 6 (enam) proses:
- 1) DSS01 *Manage operations*
 - 2) DSS02 *Manage service requests and incidents*
 - 3) DSS03 *Manage problems*
 - 4) DSS04 *Manage continuity*

- 5) DSS05 *Manage security services*
 - 6) DSS06 *Manage business process controls*
5. Domain MEA (*Monitor, Evaluate and Assess*) terdiri dari 3 (tiga) proses:
- 1) MEA01 *Monitor, evaluate and assess performance and conformance*
 - 2) MEA02 *Monitor, evaluate and assess the system of internal control*
 - 3) MEA03 *Monitor, evaluate and assess compliance with external requirements*

Gambar area, domain, dan proses-proses tata kelola dan manajemen TI organisasi pada COBIT 5 terlihat pada Gambar 2.10 berikut ini.

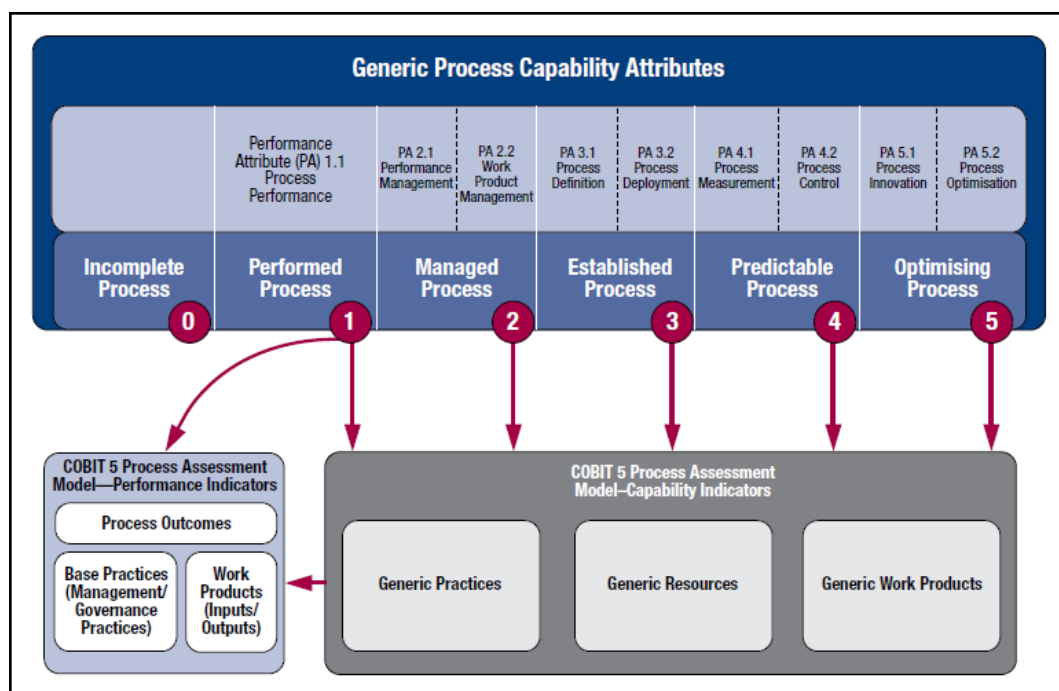


Gambar 2.10 Area, Domain Dan Proses COBIT 5

Sumber: ISACA (2012b)

2.7.1.3 Model Kapabilitas Proses COBIT 5

Pengguna COBIT 4.1 menggunakan *process maturity model* dimana model ini mengukur tingkat kematangan proses TI disuatu organisasi pada saat ini atau biasa disebut dengan ‘*as-is*’, menentukan proses TI yang dituju atau ‘*to-be*’, menentukan *gap* dan cara meningkatkan proses sesuai target tingkat kematangan TI. Pada COBIT 5 menggunakan *process capability model* yang mengacu pada standar ISO/IEC 15504 *Software Engineering - Process Assessment*. Pada COBIT 5 setiap level menuntut pemenuhan level sebelumnya dahulu baru bisa naik ke level selanjutnya. Jadi perlu dinilai terlebih dahulu untuk level 1 berdasarkan hasil proses (*process outcome*), dasar praktik (*base practices*) dan produk kerja (*work products*) setiap proses. Gambar 2.11 berikut menunjukkan pendekatan proses kapabilitas COBIT 5 (ISACA, 2012b:41).



Gambar 2.11 Model Kapabilitas Proses COBIT 5

Sumber: ISACA (2012b)

Setiap tingkat kapabilitas dapat dicapai hanya ketika semua level sebelumnya harus mencapai 100% (*Fully achieved*).

2.7.2 Kerangka Pengukuran Kapabilitas COBIT 5

Pengukuran atau penilaian tingkat kapabilitas proses meliputi (ISACA, 2013a:10):

1. Pendefinisian tingkat kapabilitas
2. Atribut proses yang digunakan untuk mengukur tingkat kapabilitas
3. Indikator yang menjadi dasar pengukuran pencapaian setiap atribut proses (berdasarkan dan sesuai dengan ISO/IEC 15504)
4. Skala penilaian standar

2.7.2.1 Definisi Tingkat Kapabilitas Proses

Kapabilitas tiap proses dinyatakan dalam tingkat proses 0 (*Incomplete*) sampai dengan 5 (*Optimizing*) seperti terlihat pada Tabel 2.1. Masing-masing tingkat kapabilitas proses diselaraskan dengan situasi proses tersebut di organisasi. Proses dengan tingkat kapabilitas 0 tidak memiliki atribut. Tingkat kapabilitas 0 mencerminkan proses tidak dilaksanakan atau proses telah gagal setidaknya untuk mencapai sebagian hasil yang diharapkan.

Tabel 2.1 Tingkat Kapabilitas Proses

Process Level	Capability
0 (Incomplete)	The process is not implemented or fails to achieve its process purpose. At this level, there is little or no evidence of any systematic achievement of the process purpose.
1 (Performed)	The implemented process achieves its process purpose.
2 (Managed)	The performed process is now implemented in a managed fashion (planned, monitored and adjusted) and its work products are appropriately established, controlled and maintained.
3 (Established)	The managed process is now implemented using a defined process that is capable of achieving its process outcomes.
4 (Predictable)	The established process now operates within defined limits to achieve its process outcomes.
5 (Optimizing)	The predictable process is continuously improved to meet relevant current and projected business goals.

Sumber: ISACA (2013a)

Skala penilaian melibatkan 6 (enam) tingkat kapabilitas sebagai berikut.

1. Level 0 Incomplete process

Proses ini tidak diimplementasikan atau gagal untuk mencapai tujuan prosesnya. Pada tingkat ini, ada sedikit atau tidak ada bukti dari setiap pencapaian yang sistematis dari tujuan proses.

2. Level 1 Performed process (one attribute)

Proses ini diimplementasikan mencapai tujuan prosesnya.

3. Level 2 Managed process (two attributes)

Berdasarkan proses level 1 pada proses ini diimplementasikan secara terkelola (direncanakan, dimonitor dan disesuaikan) dan produk kerja yang tepat dibangun, dikendalikan dan dijaga.

4. Level 3 Established process (two attributes)

Berdasarkan proses level 2 pada proses ini diimplementasikan menggunakan proses yang telah didefinisikan yang mampu dicapai hasil prosesnya.

5. Level 4 Predictable process (two attributes)

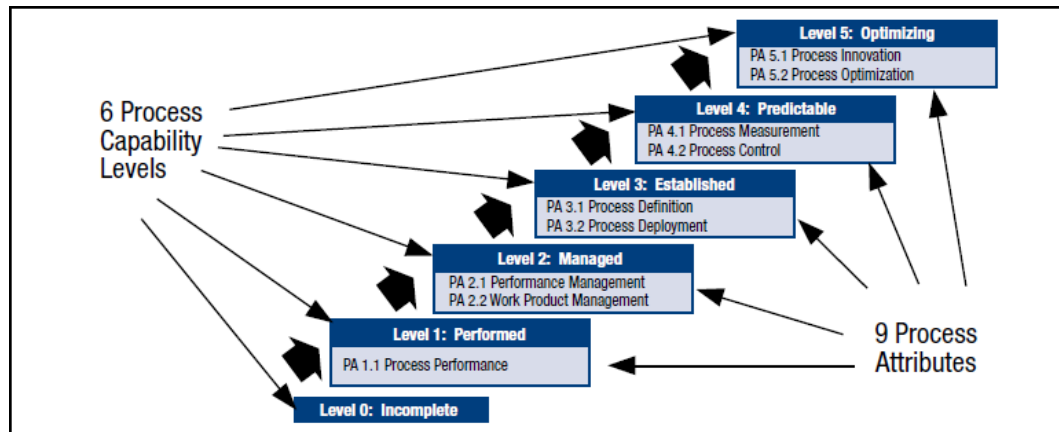
Berdasarkan proses level 3 pada proses ini dioperasikan dalam batasan yang telah ditetapkan untuk mencapai hasil prosesnya.

6. Level 5 Optimizing process (two attributes)

Berdasarkan proses level 4 terus ditingkatkan untuk memenuhi tujuan bisnis saat ini dan proyeksi yang relevan.

2.7.2.2 Atribut Proses

Pengukuran tingkat kapabilitas proses menggunakan COBIT PAM (*Process Assessment Model*) dilakukan berdasarkan 9 proses atribut (PA) yang ditentukan oleh ISO/IEC 15504 seperti terlihat pada Gambar 2.12.



Gambar 2.12 Atribut Proses

Sumber: ISACA (2013a)

Berikut adalah enam tingkat kapabilitas proses beserta PA yang ada pada tiap level (kecuali level 0 tidak memiliki PA):

1. Level 0 - Incomplete Process
2. Level 1 - Performed Process
 - PA 1.1 Process Performance Attribute
3. Level 2 - Managed Process
 - PA 2.1 Performance Management Attribute
 - PA 2.2 Work Product Management Attribute
4. Level 3 - Established Process
 - PA 3.1 Process Definition Attribute
 - PA 3.2 Process Deployment Attribute
5. Level 4 - Predictable Process

PA 4.1 Process Measurement Attribute

PA 4.2 Process Control Attribute

6. Level 5 - Optimizing Process

PA 5.1 Process Innovation Attribute

PA 5.2 Process Optimization Attribute

2.7.2.3 Indikator Penilaian

Indikator-indikator penilaian dalam COBIT PAM menjadi dasar untuk menentukan apakah atribut proses telah tercapai. Indikator penilaian dibagi menjadi 2 (dua) yaitu:

1. Indikator tingkat kapabilitas 1

Indikator bersifat spesifik untuk setiap proses dan menilai apakah atribut “proses yang diimplementasikan telah mencapai tujuan proses” telah tercapai.

2. Indikator tingkat kapabilitas 2 - 5

Penilaian kapabilitas berdasarkan pada indikator kinerja bersifat generik. Indikator ini berlaku umum untuk semua proses namun berbeda-beda untuk setiap tingkat kapabilitas.

2.7.2.4 Skala Penilaian

Masing-masing atribut proses dinilai berdasarkan skala penilaian standar dari ISO/IEC 15504-2, terdiri dari:

1. N - *Not achieved*. Tidak ada atau hanya sedikit bukti dari pencapaian atribut yang telah ditentukan pada proses yang dinilai.

2. P - *Partially achieved*. Ada beberapa bukti dari pencapaian atribut pada proses yang dinilai namun beberapa aspek dari pencapaian atribut tersebut masih belum dapat diprediksi.
3. L - *Largely achieved*. Ada bukti dari pendekatan sistematis dan pencapaian yang signifikan dari atribut pada proses yang dinilai namun beberapa kelemahan yang terkait dengan atribut masih ada dalam proses yang dinilai.
4. F - *Fully achieved*. Ada bukti dari pendekatan yang lengkap dan sistematis serta pencapaian penuh dari atribut pada proses yang dinilai dan tidak ada kelemahan yang signifikan terkait dengan atribut pada proses yang dinilai.

Dibutuhkan derajat atau tingkat penafsiran yang konsisten untuk menentukan penilaian. Tabel 2.2 menjelaskan penilaian yang diterjemahkan ke dalam skala presentase yang menunjukkan besarnya pencapaian.

Tabel 2.2 Tingkat Penilaian

N	Not achieved	0 to 15% achievement
P	Partially achieved	>15% to 50% achievement
L	Largely achieved	>50% to 85% achievement
F	Fully achieved	>85% to 100% achievement

Sumber: ISACA (2013a)

2.7.2.5 Indikator Kapabilitas Proses

Menurut ISACA (2013b), indikator kapabilitas proses yang menghubungkan atribut proses (PA) dengan level kapabilitas proses 1 sampai dengan 5. Berikut penjelasan indikator kapabilitas proses dari 9 (sembilan) atribut proses (PA) termasuk level kapabilitas proses 1 sampai dengan 5 dimana level 0

tidak termasuk jenis indikator karena level 0 mencerminkan proses yang gagal atau tidak terimplementasi untuk mencapai hasil (*outcome*) yang diinginkan.

1. Level 1 - Performed Process

Pada level ini menentukan apakah suatu proses mencapai tujuannya. Ketentuan atribut proses (PA) pada level 1 adalah sebagai berikut:

PA 1.1 Process Performance Attribute

Pengukuran mengenai seberapa jauh tujuan dari suatu proses berhasil diraih. Pencapaian penuh atas atribut ini mengakibatkan proses tersebut meraih tujuan yang sudah ditentukan.

2. Level 2 - Managed Process

Pada level ini proses diimplementasikan dalam suatu cara pengelolaan (direncanakan, dimonitor dan disesuaikan) dan hasil pekerjaan tersebut ditetapkan dengan tepat, diawasi dan dipelihara.

PA 2.1 Performance Management Attribute

Mengukur sampai mana performa proses dikelola.

PA 2.2 Work Product Management Attribute

Mengukur sejauh mana hasil kerja (*work product*) yang dihasilkan oleh proses yang dikelola dengan tepat. Hasil kerja (*work product*) yang dimaksud dalam hal ini adalah pencapaian hasil proses.

3. Level 3 - Established Process

Pada level ini proses diimplementasikan melalui suatu proses yang telah didefinisikan dan mampu untuk mencapai hasil dari proses tersebut yang telah disepakati bersama.

PA 3.1 Process Definition Attribute

Mengukur sejauh mana standar proses dikelola untuk mendukung pengerjaan dari proses yang telah didefinisikan.

PA 3.2 Process Deployment Attribute

Mengukur sejauh mana standar proses secara efektif telah dijalankan seperti proses yang telah didefinisikan untuk mencapai hasil dari proses.

4. Level 4 - Predictable Process

Pada level ini proses beroperasi di dalam batasan agar mampu meraih hasil dari proses tersebut.

PA 4.1 Process Measurement Attribute

Pengukuran mengenai seberapa jauh hasil pengukuran digunakan untuk memastikan bahwa performa proses mendukung pencapaian tujuan proses sesuai dengan tujuan bisnis perusahaan. Pengukuran dapat berupa proses atau produk atau kedua - duanya.

PA 4.2 Process Control Attribute

Pengukuran seberapa jauh suatu proses secara kuantitatif dapat menghasilkan proses yang stabil, mampu dan dapat diprediksi dalam batasan yang telah ditentukan.

5. Level 5 - Optimizing Process

Pada level ini proses ditingkatkan secara berkelanjutan untuk mewujudkan arah yang relevan dan target bisnis yang terproyeksi.

PA 5.1 Process Innovation Attribute

Mengukur sebuah perubahan proses yang telah diidentifikasi dari analisis penyebab umum dari variasi kinerja dan dari investigasi pendekatan inovatif untuk mendefinisikan dan melaksanakan proses.

PA 5.2 Process Optimization Attribute

Mengukur perubahan definisi, manajemen dan kinerja proses agar memiliki hasil yang berdampak secara efektif untuk mencapai tujuan perbaikan proses yang relevan.

2.7.2.6 Menentukan Tingkat Kapabilitas Proses

Suatu proses dinyatakan telah mencapai tingkat kapabilitas tertentu jika atribut proses pada tingkat tersebut telah mencapai nilai L (*Largely achieved*) atau F (*Fully achieved*) untuk seluruh atribut proses di tingkat yang lebih rendah. Penentuan tingkat kapabilitas suatu proses dapat dilihat pada Tabel 2.3.

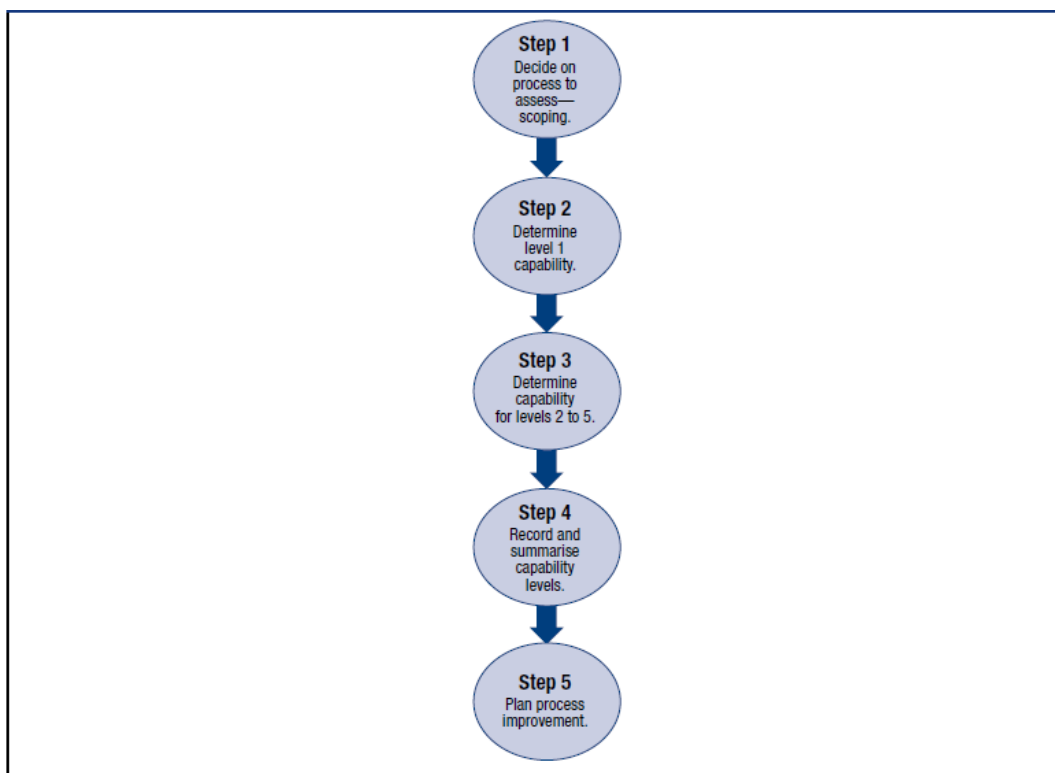
Tabel 2.3 Penentuan Tingkat Kapabilitas Proses

Scale	Process Attributes	Rating
Level 1	Process Performance	Largely or fully
Level 2	Process Performance Performance Management Work Product Management	Fully Largely or fully Largely or fully
Level 3	Process Performance Performance Management Work Product Management Process Definition Process Deployment	Fully Fully Fully Largely or fully Largely or fully
Level 4	Process Performance Performance Management Work Product Management Process Definition Process Deployment Process Measurement Process Control	Fully Fully Fully Fully Fully Largely or fully Largely or fully
Level 5	Process Performance Performance Management Work Product Management Process Definition Process Deployment Process Measurement Process Control Process Innovation Process Optimization	Fully Fully Fully Fully Fully Fully Fully Largely or fully Largely or fully

Sumber: ISACA (2013a)

2.7.2.7 Langkah Penilaian Tingkat Kapabilitas

Penilaian tingkat kapabilitas untuk setiap proses menggunakan *self-assessment process* yang merupakan pendekatan sederhana untuk memberikan penilaian terhadap proses yang dapat dilakukan oleh manajemen TI sebagai pelopor untuk penilaian yang lebih formal. *Self-assessment process* dapat mengidentifikasi kesenjangan proses yang dapat disempurnakan dengan penilaian formal. Gambar 2.13 di bawah ini menunjukkan tahapan *self-assessment process*.



Gambar 2.13 Self-assessment Process

Sumber: ISACA (2013a)

1. Langkah 1 - Menentukan proses yang akan dinilai - Ruang Lingkup.

Menentukan proses-proses COBIT 5 yang akan dinilai. Proses tersebut menjadi fokus perhatian manajemen perusahaan atau yang berkaitan dengan tujuan bisnis terkait TI. Pada tahap ini, target tingkat kapabilitas proses

ditentukan dengan mempertimbangkan dampak kepada tujuan bisnis dari perusahaan jika tingkat kapabilitas yang ingin dicapai tidak dapat terpenuhi.

2. Langkah 2 - Menentukan apakah proses yang dipilih telah mencapai tingkat kapabilitas level 1.

Menentukan apakah proses COBIT 5 yang dinilai telah mencapai tingkat kapabilitas level 1. Indikator untuk tingkat kapabilitas level 1 bersifat spesifik dan berbeda untuk setiap proses. Penilaian dilakukan terhadap pencapaian hasil (*outcome*) dari PA (*process attribute*) tingkat kapabilitas level 1.

3. Langkah 3 - Menentukan apakah proses yang dipilih telah mencapai tingkat kapabilitas level 2 sampai dengan 5.

Menentukan apakah proses COBIT 5 yang dinilai telah mencapai tingkat kapabilitas level 2 sampai dengan 5. Kriteria penilaian tingkat kapabilitas level 2 sampai dengan 5 bersifat generik untuk semua proses namun berbeda untuk tiap tingkat kapabilitas.

4. Langkah 4 - Mencatat dan meringkas tingkat kapabilitas untuk semua proses yang telah dinilai.

Tiap tingkat kapabilitas ditentukan dimana indikator penilaian tingkat kapabilitas harus '*Largely achieved*' atau '*Fully achieved*'.

5. Langkah 5 - Mengembangkan rencana perbaikan proses

Rencana perbaikan proses bisa dilakukan dengan 2 (dua) cara. Pertama, membuat rencana perbaikan awal yang didasarkan pada *self-assessment* dengan menentukan area yang paling penting sesuai tujuan bisnis perusahaan dan fokus pada *gaps* antara tingkat kapabilitas proses sekarang (*current*) dan

tujuan (*target*). Kedua, melakukan penilaian independen yang lebih formal, berdasarkan COBIT PAM dan panduan penilai (*assessor guide*).

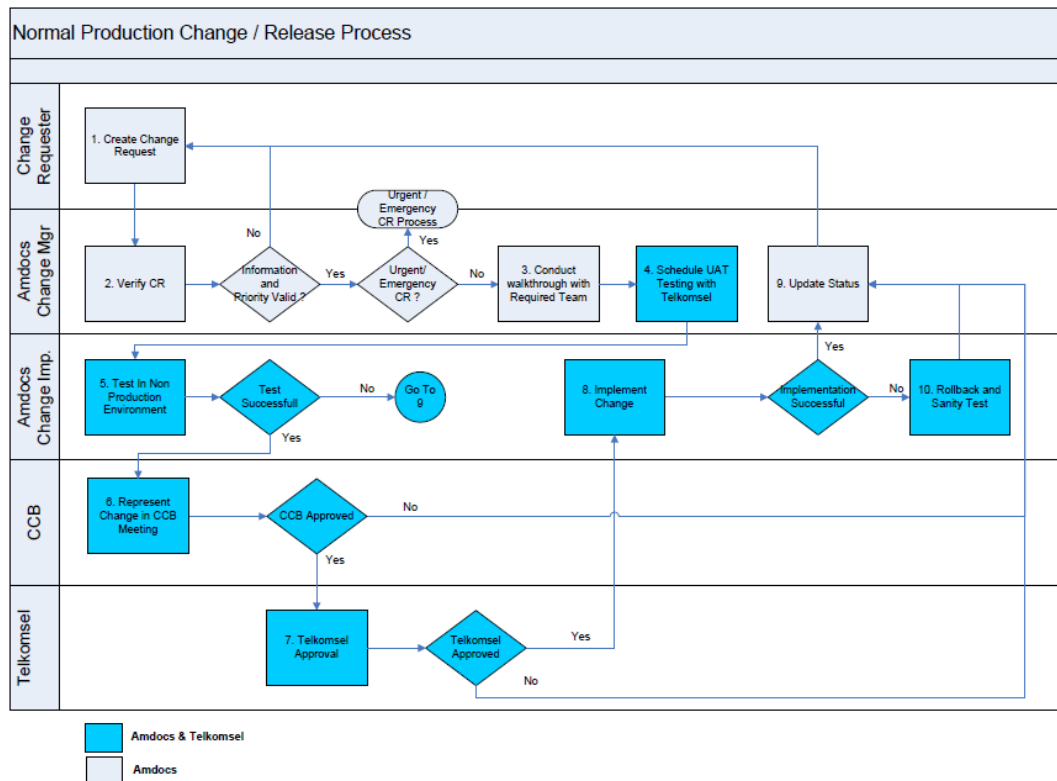
2.8 Service Management Process

Untuk menjalankan bisnis *manage services* dari sistem OCS maka PT. Application Solutions memiliki beberapa proses manajemen layanan. *Service Management Process* di PT. Application Solutions secara umum di bagi menjadi 3 kategori:

1. *Production Change / Release Process*
2. *Emergency Change Request / Hotfix Process*
3. *Incident Process*

2.8.1 Production Change / Release Process

Proses perubahan pada sistem OCS yang terjadwal dengan rutin dimana perubahan diminta oleh pihak XYZ dan dibuat untuk kebutuhan bisnis XYZ. Pada prosesnya perubahan ini dinamakan dengan *Business As Usual* (BAU). Proses perubahan pada sistem OCS terlihat pada Gambar 2.14.



Gambar 2.14 Service Mgmt Production Change / Release Process

Sumber: PT. Application Solutions (2011b)

Tahapan perubahan pada sistem OCS adalah sebagai berikut:

1. Membuat permintaan perubahan (*Change Request*) - Pemohon perubahan membuat permintaan perubahan baru menggunakan aplikasi UTS.
2. Memverifikasi permintaan perubahan (*Request For Change*) - *Change Manager* akan memverifikasi informasi dari permintaan perubahan seperti prioritas perubahan, deskripsi perubahan, rencana implementasi, rencana pengujian, kontak yang terlibat terhadap perubahan, dampak terhadap sistem, rincian *outage*, dampak terhadap aplikasi, dll. Lalu melakukan pengecekan berikut ini.
 - 1) Apakah informasi dan prioritas valid?

1. Jika ya, *Change Manager* memproses RFC tersebut

2. Jika tidak, *Change Manager* mengirim kembali RFC tersebut ke pemohon perubahan. *Change Manager* akan memastikan bahwa alasan yang tepat diberikan didalam formulir RFC pemohon
- 2) Apakah RFC penting / *emergency*?
1. Jika ya, *Change Manager* akan merujuk pada *Emergency Change Request / Hotfix Process*
 2. Jika tidak, RFC akan diproses dengan mengikuti aliran proses permintaan perubahan dan diteruskan dengan aliran proses berikutnya.
3. Mengumpulkan petunjuk dengan tim terkait perubahan - Jika RFC melibatkan beberapa tim maka *Change Manager* mengumpulkan petunjuk dengan tim terkait agar masing-masing *stakeholder* dari tim tersebut memahami perubahan dan siap untuk implementasi.
 4. Menjadwalkan UAT (*User Acceptance Test*) dengan XYZ - *Change Manager* membuat jadwal UAT dengan XYZ.
 5. Melakukan tes di sistem pengujian (*testbed*) - Pelaksana perubahan mengimplementasikan dan melakukan pengujian di sistem pengujian.
- Apakah pengetesan berhasil?
- 1) Jika ya, *Change Manager* akan menjelaskan perubahan ke CCB (*Change Control Board*)
 - 2) Jika tidak, pelaksana perubahan melakukan *rollback* dan *sanity test* dan memperbarui status RFC menjadi gagal (*failed*)

6. Menjelaskan perubahan di rapat CCB - Pemohon perubahan akan mempresentasikan perubahan dalam rapat CCB dan menyajikan kasus untuk persetujuan.

Apakah CCB menyetujui?

- 1) Jika ya, CCB memberikan persetujuan dan mengajukan permintaan perubahan ke XYZ untuk mendapatkan persetujuan XYZ
- 2) Jika tidak, *Change Manager* akan memperbarui status permintaan perubahan

7. Persetujuan XYZ - XYZ CCB/Divisi Operasional akan memeriksa permintaan perubahan.

Apakah XYZ menyetujui?

- 1) Jika ya, implementasi akan dilanjutkan oleh pelaksana perubahan
- 2) Jika tidak, *Change Manager* akan memperbarui status permintaan perubahan

8. Implementasi Perubahan - Pelaksana perubahan akan menerapkan perubahan sesuai jadwal yang telah ditentukan oleh CCB XYZ

Apakah implementasi sukses?

- 1) Jika ya, *Change Manager* akan memperbarui status permintaan perubahan
- 2) Jika tidak, Pelaksana perubahan akan melakukan *rollback* dan *sanity test*. Pelaksana perubahan juga akan melakukan investigasi kegagalan implementasi

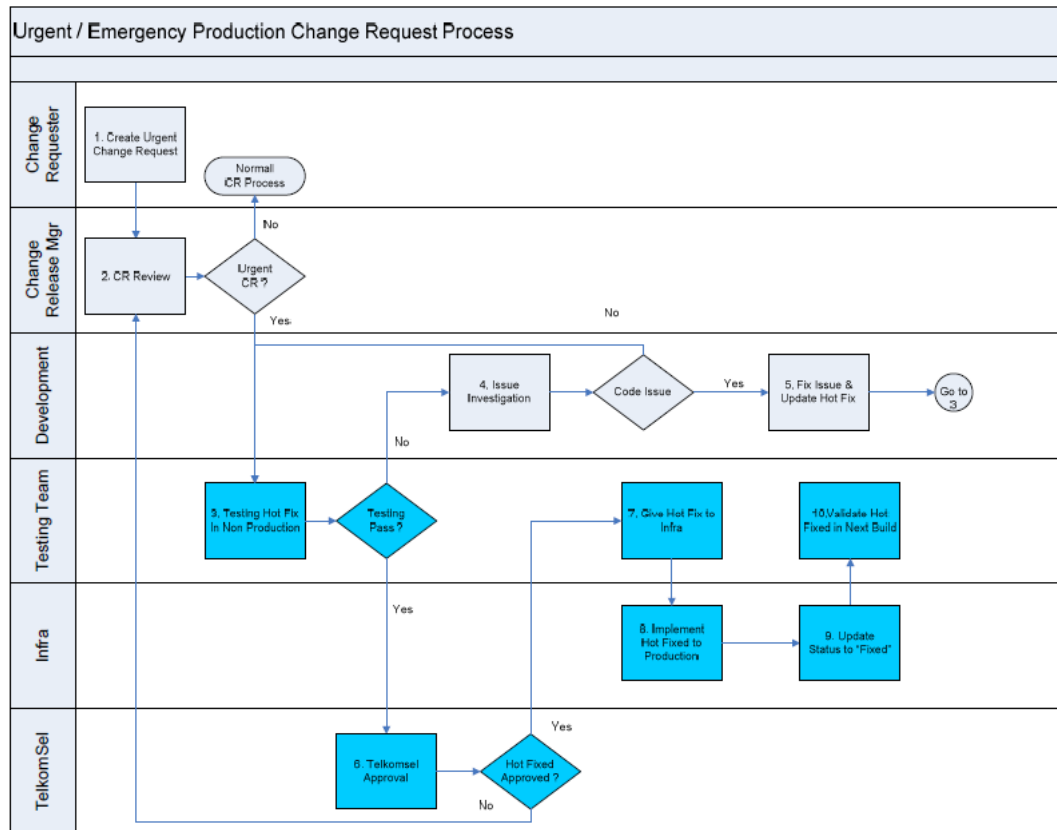
9. *Update* status - Jika perubahan telah berhasil dilaksanakan, maka *Change Manager* memperbarui aplikasi UTS dengan status yang benar dan

mengirimkan informasi terbaru kepada pemohon perubahan (XYZ) atau tim manajemen perubahan.

10. *Rollback* dan *Sanity Test* - Jika pelaksana perubahan menemukan akar masalah gagalnya perubahan maka pelaksana perubahan harus memperbaiki masalah tersebut dan melakukan tes *sanity* untuk menyelesaikan proses perubahan dengan sukses. Pelaksana perubahan akan melakukan aktivitas *rollback* jika diperlukan untuk memperbaiki masalah dalam implementasi perubahan dengan persetujuan *Change Manager*.

2.8.2 Emergency Change Request / Hotfix Process

Emergency Change Request adalah perubahan yang diminta oleh XYZ untuk kebutuhan bisnis namun perlu adanya *enhancement* (peningkatan) di *core* sistem OCS atau perubahan pada kode program dan perubahan yang sifatnya perbaikan *bug* pada sistem yang terdeteksi oleh internal PT. Application Solutions atau biasa disebut dengan “*emergency hotfix*”. Ketika skenario ini terjadi maka PT. Application Solutions akan memberikan prosedur yang sudah terdokumentasi untuk melaksanakan perubahan *emergency* ini. Proses perubahan *emergency* pada sistem OCS terlihat pada Gambar 2.15.



Gambar 2.15 Service Mgmt Emergency Change Request / Hotfix Process

Sumber: PT. Application Solutions. (2011b)

Tahapan perubahan *emergency* pada sistem OCS adalah sebagai berikut:

1. Membuat permintaan perubahan penting - Pemohon perubahan membuat permintaan penting. Pemohon perubahan membuat permintaan perubahan baru menggunakan aplikasi UTS.
2. Pemeriksaan RFC - *Change Manager* mengesahkan RFC dan memverifikasi apakah “*hotfix*” ini penting atau tidak.

Apakah RFC ini penting / *emergency*?

- 1) Jika ya, *Change Manager* akan berdiskusi dengan tim testing untuk pengetesan “*hotfix*” di sistem pengujian.
- 2) Jika tidak, RFC tersebut akan merujuk pada *Production Change / Release Process*

3. Pengetesan *hotfix* di sistem pengujian - tim testing mulai melakukan *testing* di sistem pengujian.

Apakah hasil tes sukses?

- 1) Jika ya, *Change Manager* akan berdiskusi dengan XYZ dan menginformasikan hasil pengetesan untuk mendapatkan persetujuan untuk dijalankan di *production / live system*.
- 2) Jika tidak, *hotfix* tersebut akan dikembalikan ke bagian *development* untuk diinvestigasi lebih lanjut. Notifikasi melalui *email* akan diberikan ke semua *stakeholder* terkait kegagalan implementasi *hotfix*.

4. Investigasi masalah - tim *development* akan melakukan investigasi untuk *hotfix* yang gagal dalam pengetesan.

Apakah masalah pada kode program?

- 1) Jika ya, tim *development* akan melakukan perbaikan pada *hotfix* dengan menggunakan simulasi rekayasa sistem di Amdocs pusat.
- 2) Jika tidak, apabila tim *development* memastikan apabila kegagalan tersebut bukan karena kode program seperti masalah *server*. Maka tim *development* akan menginformasikan tim *testing* untuk melakukan *testing* ulang terhadap *hotfix* tersebut (langkah 3).

5. Menyelesaikan masalah dan memperbarui *hotfix* - tim *development* akan melakukan perbaikan pada kode program *hotfix*. Setelah selesai maka tim *development* akan memberitahu tim *testing* untuk melakukan pengetesan ulang (langkah 3).

6. Persetujuan XYZ - *Change Manager* akan meminta persetujuan XYZ untuk implementasi *hotfix*. *Change Manager* menjamin *hotfix* telah diverifikasi di *livesystem* oleh tim *development*, CC telah diupdate, dan XYZ diberitahu dan RFC terkait diperbarui dengan status kemajuan.

Apakah *hotfix* disetujui oleh XYZ?

- 1) Jika ya, tim *testing* akan memberikan *hotfix* kepada tim infrastruktur
 - 2) Jika tidak, kembali ke pemeriksaan RFC (Langkah 2)
7. Memberikan *hotfix* ke tim infrastruktur - tim *testing* memberikan kode *hotfix* kepada tim infrastruktur.
 8. Implementasi *hotfix* - tim infrastruktur akan memindahkan / mengimplementasikan *hotfix* ini pada *livesystem* dan memberitahu pihak-pihak terkait terhadap *hotfix* apabila sudah selesai.
 9. Perbarui status menjadi “*Fixed*” - tim infrastruktur akan memperbarui status *hotfix* menjadi *fixed*.
 10. Validasi *hotfix* pada jadwal rilis produk berikutnya.

2.8.3 Incident Process

Alur proses insiden dibagi menjadi 2 (dua) yaitu:

1. Alur proses insiden P1/P2

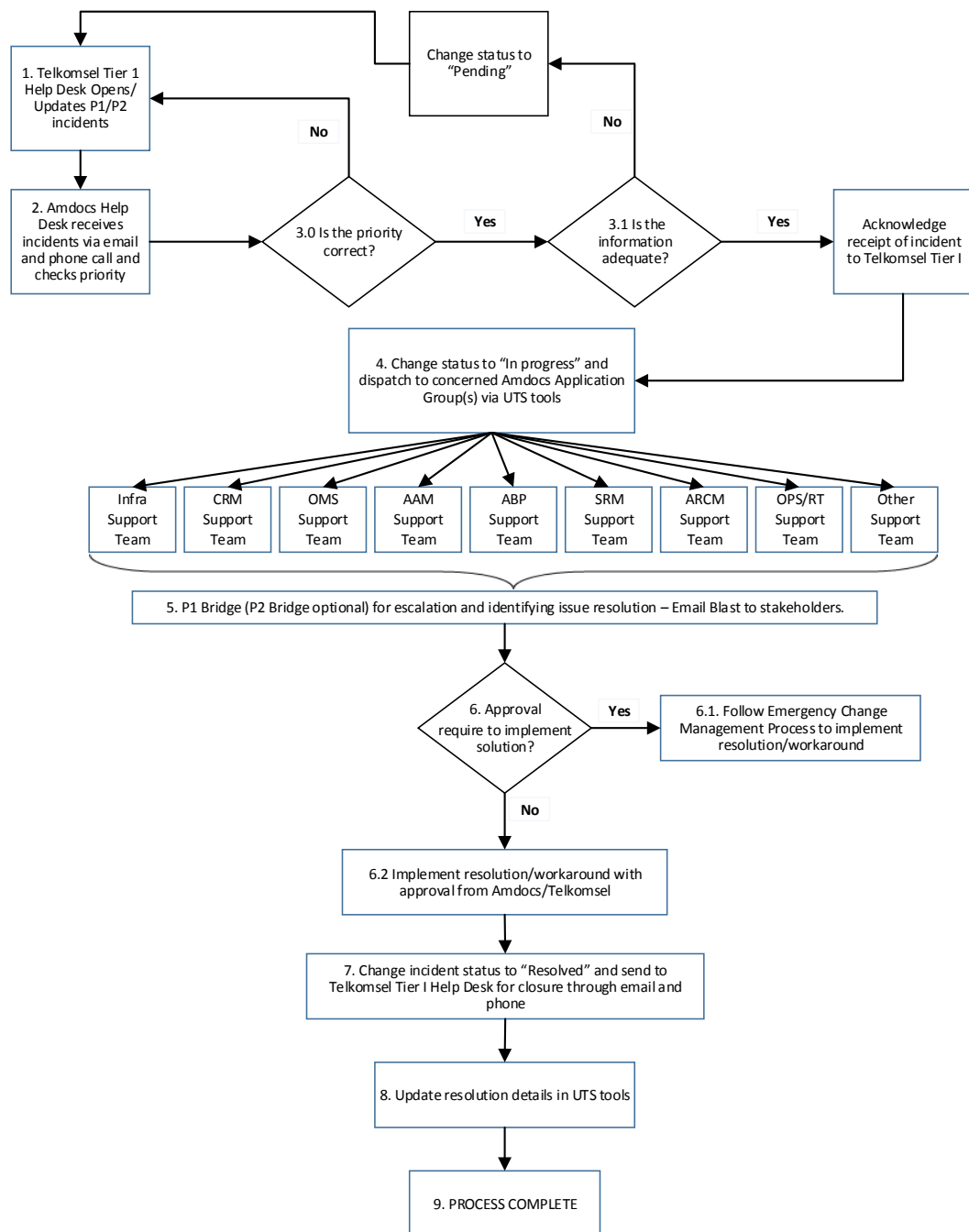
Insiden P1 yaitu insiden yang bersifat *emergency* dimana *response time* terhadap insiden ini < 15menit dan perbaikan layanan harus <1 jam dan penyelesaian masalah < 48jam. Sedangkan insiden P2 yaitu insiden yang bersifat *high* dimana dimana *response time* terhadap

insiden ini < 15menit dan perbaikan layanan harus < 24jam dan penyelesaian masalah < 4minggu.

2. Alur proses insiden P3/P4

Insiden P3 yaitu insiden yang bersifat *medium* dimana *response time* terhadap insiden ini < 15menit dan perbaikan layanan harus < 1bulan dan penyelesaian masalah < 3bulan. Sedangkan insiden P4 yaitu insiden yang bersifat *low* dimana *response time* terhadap insiden ini < 15menit dan perbaikan layanan < 3bulan dan penyelesaian masalah apabila ada *software maintenance release* baru.

Alur proses insiden P1/P2 terlihat pada Gambar 2.16.



Gambar 2.16 Alur Proses Insiden P1/P2

Sumber: PT. Application Solutions (2011a)

Deskripsi alur proses insiden P1/P2 adalah sebagai berikut:

1. XYZ Tier 1 *Helpdesk* membuat tiket insiden

1) Tiket insiden P1/P2 dibuat oleh XYZ Tier 1 *Helpdesk* melalui aplikasi UTS;

2) Notifikasi *email* akan muncul pada Amdocs *Helpdesk* sebagai penerima (*assignee*) tiket;

3) XYZ Tier 1 *Helpdesk* melakukan panggilan telepon ke Amdocs *Helpdesk* untuk menyampaikan permasalahan insiden PI/P2

2. Amdocs *Helpdesk* menerima tiket insiden

1) Amdocs *Helpdesk* menerima notifikasi tiket insiden melalui *email*;

2) Amdocs *Helpdesk updates* diaplikasi UTS

3. Apakah prioritas insiden sudah benar?

Jika prioritas insiden sudah benar, lalu Amdocs *Helpdesk* akan memeriksa informasi yang diberikan dalam tiket insiden tersebut:

1) Jika informasi yang diterima sudah jelas maka lanjut ke langkah 4 (empat)

2) Jika informasi yang diperlukan untuk menyelesaikan insiden itu tidak ada atau kurang lengkap maka Amdocs *Helpdesk* akan merubah status tiket menjadi “*Pending*” pada aplikasi UTS dan menugaskan (*assign*) tiket kembali ke XYZ Tier 1 *Helpdesk* untuk informasi lebih lanjut melalui email dan menindaklanjuti melalui telepon.

3) Jika Amdocs *Helpdesk* menentukan prioritas insiden tidak tepat, prioritas insiden itu akan berubah, setelah berdiskusi dengan XYZ Tier 1 *Helpdesk* atau Tim Manajemen Permasalahan (*Problem Management Team*).

4. Pengakuan Insiden Tiket

Jika insiden itu benar P1 atau P2 dan semua informasi insiden yang diperlukan tersedia, maka Amdocs *Helpdesk* akan mengubah status insiden menjadi "*In Progress*" di aplikasi UTS dan mengirimkan insiden tersebut ke group aplikasi Amdocs yang berkaitan dengan permasalahan. Amdocs *Helpdesk* juga akan menghubungi pemimpin / manajer group Amdocs melalui telepon.

5. Bridge P1 untuk eskalasi dan mengidentifikasi resolusi permasalahan
 - 1) Internal P1 *Bridge* akan dimulai segera oleh Amdocs *Helpdesk* sehingga semua tim Amdocs bersangkutan dapat bekerja bersama-sama dan mengidentifikasi resolusi / solusi atas masalah tersebut.
 - 2) Selain internal P1 *Bridge*, XYZ Tier 1 *Helpdesk* dapat juga memulai *bridge* dengan mengundang *stakeholder* untuk bertukar informasi pada insiden P1 yang sedang terjadi. Amdocs *Helpdesk* yang bertanggungjawab untuk menghubungkan XYZ *bridge* dengan tim Amdocs.
6. Persetujuan Resolusi / Solusi
 - 1) Jika resolusi / solusi yang diusulkan memerlukan persetujuan khusus untuk pelaksanaan (contoh: restart aplikasi atau implementasi *emergency hotfix*) maka *Emergency Change Request / Hotfix Process* akan digunakan untuk melaksanakan resolusi.
 - 2) Jika tidak memerlukan persetujuan khusus untuk resolusi / solusi yang diusulkan maka group aplikasi Amdocs yang berhubungan dengan insiden dan Amdocs *Helpdesk* akan melanjutkan pelaksanaan perbaikan.

7. Masalah terselesaikan

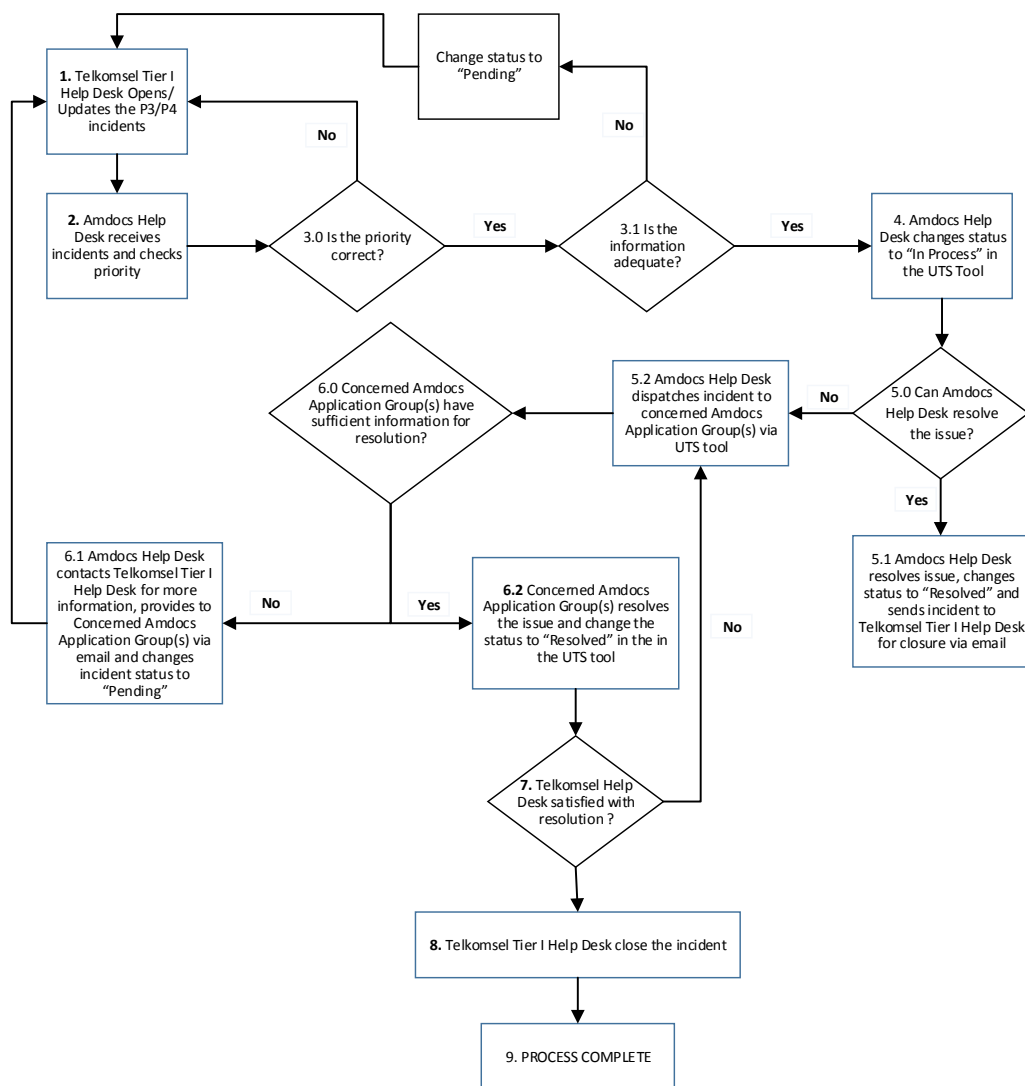
Group aplikasi Amdocs yang berhubungan dengan insiden dan Amdocs *Helpdesk* mengubah status tiket insiden menjadi “*Resolved*” di aplikasi UTS dan menginformasikan XYZ Tier I *Helpdesk* melalui email dan telepon.

8. Memperbarui resolusi

Group aplikasi Amdocs yang berhubungan dengan insiden akan *update* detail resolusi di aplikasi UTS.

9. Proses Selesai

Alur proses insiden P3/P4 terlihat pada Gambar 2.17.



Gambar 2.17 Alur Proses Insiden P3/P4
 Sumber: PT. Application Solutions (2011a)

Deskripsi alur proses insiden P3/P4 adalah sebagai berikut:

1. XYZ Tier 1 *Helpdesk* membuat tiket insiden
 - 1) Tiket insiden P3/P4 dibuat oleh XYZ Tier 1 *Helpdesk* melalui aplikasi UTS;
 - 2) Notifikasi email akan muncul pada Amdocs *Helpdesk* sebagai penerima (*assignee*) tiket;

2. XYZ Tier 1 *Helpdesk* melakukan panggilan telepon ke Amdocs *Helpdesk* untuk menyampaikan permasalahan insiden P3/P4

3. Amdocs *Helpdesk* menerima tiket insiden

1) Amdocs *Helpdesk* menerima notifikasi tiket insiden melalui email;

2) Amdocs *Helpdesk updates* di aplikasi UTS

4. Apakah prioritas insiden sudah benar?

Jika prioritas insiden dengan benar ditentukan, lalu Amdocs *Helpdesk* akan memeriksa informasi yang diberikan dengan tiket insiden tersebut:

1) Jika informasi yang diterima sudah jelas maka lanjut ke langkah 4 (empat)

2) Jika informasi yang diperlukan untuk menyelesaikan insiden itu tidak ada atau kurang lengkap maka Amdocs *Helpdesk* akan merubah status tiket menjadi “*Pending*” pada aplikasi UTS dan menugaskan (*assign*) tiket kembali ke XYZ Tier 1 *Helpdesk* untuk informasi lebih lanjut melalui email dan menindaklanjuti melalui telepon.

5. Jika Amdocs *Helpdesk* melihat prioritas insiden tidak tepat, prioritas insiden itu akan berubah, setelah berdiskusi dengan XYZ Tier 1 *Helpdesk* atau Tim Manajemen Permasalahan (*Problem Management Team*).

6. Pengakuan Insiden Tiket

Jika insiden itu benar P3 atau P4 dan semua informasi insiden yang diperlukan tersedia, maka Amdocs *Helpdesk* akan mengubah status insiden menjadi “*In Progress*” di aplikasi UTS dan mengirimkan insiden tersebut ke grup aplikasi Amdocs yang berkaitan dengan permasalahan.

Amdocs *Helpdesk* juga akan menghubungi pemimpin / manajer group aplikasi Amdocs melalui telepon.

7. Apakah Amdocs *Helpdesk* dapat memecahkan masalah?

- 1) Jika ya - Amdocs *Helpdesk* akan memberikan resolusi, mengubah status tiket insiden menjadi "*Resolved*", menambahkan rincian lengkap resolusi di aplikasi UTS dan kemudian mengirim tiket insiden tersebut kepada XYZ Tier I *Helpdesk* untuk penutupan tiket.
- 2) Jika tidak – Amdocs *Helpdesk* akan mengirim tiker insiden ke grup aplikasi Amdocs yang berhubungan dengan insiden melalui aplikasi UTS.

8. Apakah informasi dari insiden tiket lengkap untuk group aplikasi Amdocs?

- 1) Jika informasi dari insiden tiket lengkap maka grup aplikasi Amdocs akan memberikan resolusi, mengubah status tiket insiden menjadi "*Resolved*", menambahkan rincian lengkap resolusi di aplikasi UTS dan kemudian mengirim tiket insiden tersebut kepada XYZ Tier I *Helpdesk* untuk penutupan tiket.
- 2) Jika resolusi teridentifikasi *defect* yang membutuhkan CR (*Change Request*), pemimpin grup aplikasi Amdocs akan meminta XYZ Tier I *Helpdesk* membuat CR (merujuk pada *Emergency Change Request / Hotfix Process*) dan mengubah status insiden tersebut menjadi "*Resolved*". Pimpinan grup aplikasi Amdocs juga akan

membuka *internal defect*, sesuai dengan CR yang dibuka oleh XYZ dengan menggunakan aplikasi UTS.

- 3) Jika informasi yang diperlukan untuk menyelesaikan insiden itu tidak ada atau kurang lengkap maka grup aplikasi Amdocs akan merubah status tiket menjadi “*Pending*” pada aplikasi UTS dan menugaskan (*assign*) tiket kembali ke XYZ Tier 1 *Helpdesk* untuk informasi lebih lanjut melalui email dan menindaklanjuti melalui telepon.

9. Apakah XYZ Tier I *Helpdesk* puas dengan resolusi?

- 1) Jika XYZ Tier I *Helpdesk* tidak setuju dengan resolusi yang diusulkan oleh Amdocs, ia akan mengirim insiden tersebut kembali ke Amdocs *Helpdesk* untuk rincian lebih lanjut dari resolusi. Amdocs *Helpdesk* akan mengirimkan kejadian tersebut ke grup aplikasi Amdocs untuk memperbarui komentar.
- 2) Jika resolusi ok maka proses dijelaskan pada Langkah 8
- 3) Penutupan tiket secara otomatis terjadi apabila XYZ tidak merespon kembali kasus yang telah diselesaikan dalam 1 (satu) minggu waktu kerja.

10. XYZ Tier I *Helpdesk* menerima resolusi

XYZ Tier I *Helpdesk* akan menerima insiden dengan resolusi yang dijelaskan oleh grup aplikasi Amdocs dan akan menutup tiket insiden tersebut. Secara otomatis UTS sistem akan menginformasikan status tiket ke Amdocs *Helpdesk* melalui email.

11. Proses Selesai