

# **PENDEKATAN MODEL *INFORMATION TECHNOLOGY INFRASTRUCTURE LIBRARY* (ITIL) DALAM REKAYASA SISTEM INFORMASI UNTUK PENCAPAIAN *IT-GOVERNANCE***

**Herison Surbakti**

Fakultas Sains dan Teknologi, Universitas Respati, Yogyakarta

E-mail : [herisonsurbakticc@gmail.com](mailto:herisonsurbakticc@gmail.com)

## **ABSTRAK**

Suatu organisasi hendaknya menyiapkan *blueprint* Teknologi Informasi yang akan dipergunakan untuk mendukung proses pengolahan data sebelum membangun dan mengimplementasikannya, yang pada akhirnya akan menghasilkan suatu informasi yang dibutuhkan sebagai bahan pertimbangan pengambilan keputusan dalam menentukan kebijakan organisasi pada masa mendatang.

Hampir semua organisasi membutuhkan teknologi informasi untuk mendukung kegiatan organisasi, namun tidak semua organisasi mempunyai *blueprint* Teknologi Informasi yang baik, sehingga akan mengalami kesulitan untuk mengembangkan sistem/aplikasi dibidang teknologi yang dimiliki.

Kesulitan dalam membangun *blueprint* dan *roadmap* implementasi ditimbulkan akibat dari tidak adanya standar arsitektur sistem, tools pengembangan, database management system (DBMS) dan manajemen dokumen sistem. Desain perangkat lunak sangat bergantung pada struktur organisasi perusahaan karena berhubungan dengan pemeliharaan, perbaikan sistem pengujian, integrasi sistem dan pengembangan sistem yang akan dimulai dari awal, bukan melanjutkan sistem yang sudah ada.

Hal yang tidak kalah pentingnya adalah kelancaran sistem yang sangat tergantung dari pengelolaan Teknologi Informasi, tetapi jarang sekali dilakukan dengan baik, dan oleh karena itu diperlukan suatu rekayasa sistem informasi untuk membangun infrastruktur sistem informasi dengan menggunakan sebuah *framework* Teknologi Informasi sebagai standar penggunaan sistem. Dengan *Information Technology Infrastructure Library*(ITIL) sebagai "Best Practice" *framework* di bidang IT yang dicetuskan dan banyak diimplementasikan di United Kingdom (UK), diharapkan organisasi dapat memperoleh *roadmap* implementasi dan *blueprint* untuk pemanfaatan, implementasi dan penggunaan Teknologi Informasi.

**Kata kunci :** teknologi, informasi, institusi, *blueprint*, perancangan, sistem, pemeliharaan, *framework*, best practice, infrastruktur library.

## **1. PENDAHULUAN**

*Information Technology Infrastructure Library* (ITIL) merupakan *framework* untuk mengelola infrastruktur Teknologi Informasi (TI) di suatu organisasi dan bagaimana memberikan service terbaik bagi pengguna layanan TI. ITIL pada dasarnya membahas mengenai *IT Service Management*, dan membedakan antara *incident* dan *problem*, antara *problem*, *error* dan *known error*. ITIL terdiri atas komponen-komponen proses di dalam *IT Service Management*. Komponen tersebut terdiri atas *service support* dan *service delivery* (Rudd, 2004).

### ***1.1. Service Support***

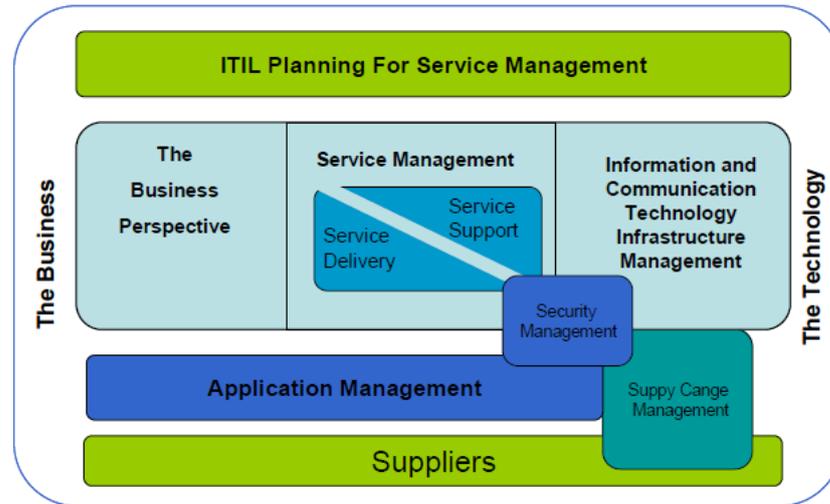
Service support membahas tentang proses dan komponen dalam operasional IT sehari-hari, seperti *Incident Management*, *Problem Management*, *Change Management*, *Configuration Management* dan *Release Management*. Pada dasarnya service support menjelaskan proses *support* dari IT dan bagaimana cara menangani suatu *incident* sebelum akhirnya masuk ke dalam *problem management*. komponen dari proses pada IT *service management* mempunyai peran masing-masing yaitu:

1. *Service desk* sebagai *single point of contact*
2. *Release Management* dalam melakukan testing suatu teknologi baru sebelum diimplementasi
3. *Build knowledge based* untuk membantu mempercepat penyelesaian suatu *incident*

### ***1.2. Service Delivery***

Komponen-komponen dari *Service Delivery* adalah *Financial Management for IT Services*, *Availability Management*, *Capacity Management*, *IT Service Continuity Management*, *Service Level Management*. *Service Delivery* ini merupakan *tactical* proses artinya untuk mendukung bisnis dari sisi IT *Service Management* dalam jangka panjang bukan untuk operasional yang dilakukan setiap hari. Contohnya dalam bagian IT *Service Continuity Management*, misalnya dilakukannya *service level agreement* (SLA), melakukan *Risk Management* berfungsi untuk mengetahui dan dapat mengidentifikasi bentuk resiko, menentukan tingkat resiko yang akan terjadi, untuk mengetahui *asset* mana saja yang mempunyai resiko, dan sebagainya. ITIL memberikan dasar dan konsep IT sebagai bagian dari *service* ke *customer* atau *user*, bagaimana proses yang baik antara satu *entity* dengan *entity* lainnya.

ITIL meliputi *service support*, *service delivery*, rencana pengembangan *service management*, *ICT infrastructure management*, *application management*, *business perspective*, *security management*, dan *software asset management*. *Service support* dan *service delivery* merupakan area utama, yang disebut juga *IT Service Management* (ITSM). Secara bersama-sama, dua area ini mengandung beberapa disiplin yang bertanggung jawab untuk penentuan dan manajemen pelayanan Teknologi Informasi (TI) yang efektif.



Gambar 1.1 Framework IT Infrastructur Library

**2. INFORMATION TECHNOLOGY INFRASTRUCTURE LIBRARY (ITIL) SERVICE**

IT Infrastructure Library yang ada saat ini merupakan suatu petunjuk yang terintegrasi dari Office of Governments Commerce (OGC) dan British Standards Institution (BSI) yang digunakan untuk penerapan teknologi informasi dan disupport oleh aplikasi-aplikasi serta modul-modul pelatihan yang terstruktur. Terdiri dari dua area pada service Management yaitu Service Support dan Service Delivery (IBM Global, 2004).

Tabel 1.1 ITIL Service Table

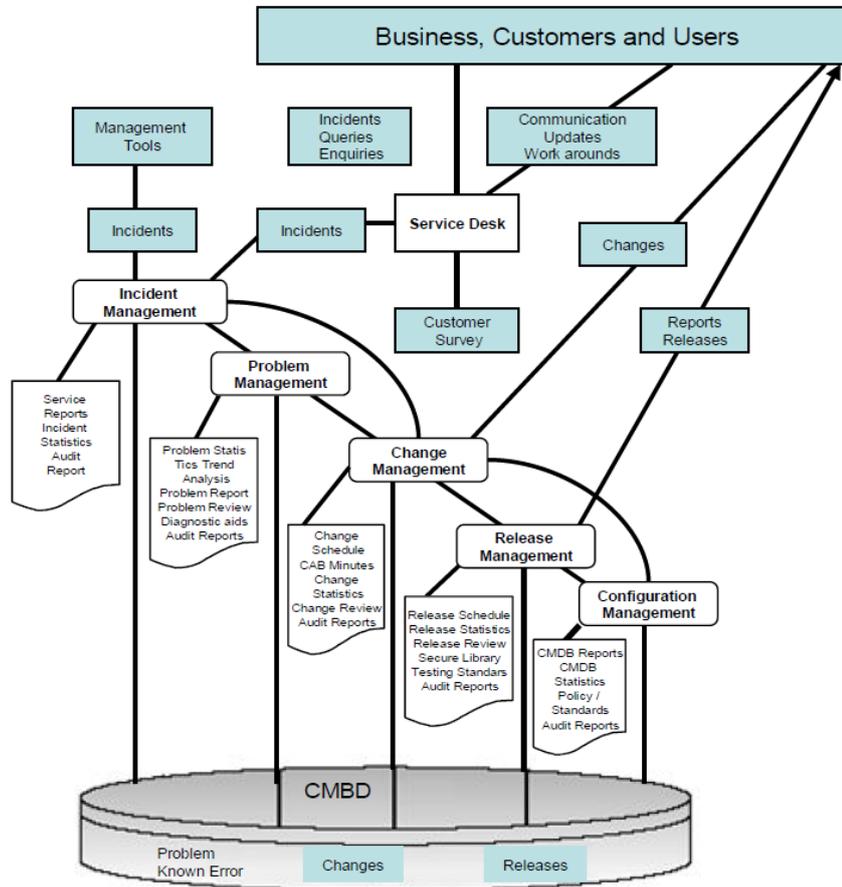
Service Support	Service Delivery
Service Desk	
Incident Management	Service Level management
Problem Management	Financial Management for IT Service
Configuration Management	Capacity Management
Change Management	IT Service Continuity Management
Release management	Availability Management

**2.1. Service Support**

Service support adalah suatu penerapan disiplin yang memungkinkan tersedianya pelayanan TI. Tanpa disiplin ini, teknologi informasi hampir tidak mungkin menyediakan pelayanan TI yang baik, dan tidak dapat dikelola dengan baik. Disiplin service support yaitu:

1. Configuration management
2. Incident Management
3. Problem management

4. Change management
5. Service/help desk
6. Release management



Gambar 2.1 Service Support Process

### 1. Configuration management

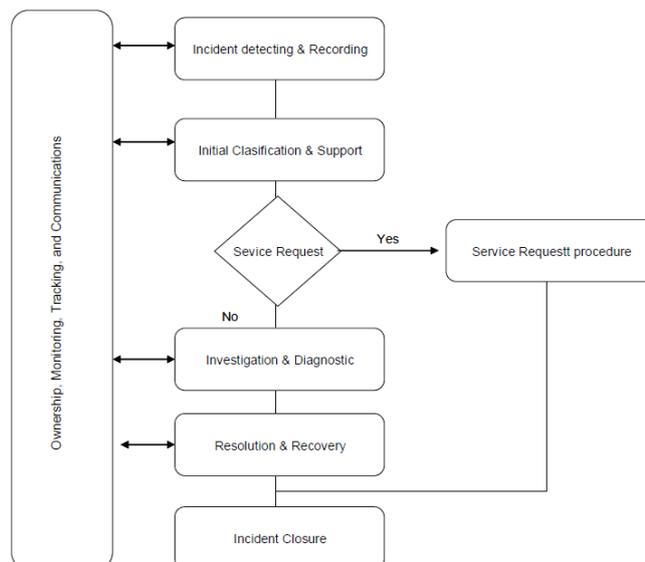
Manajemen konfigurasi terdiri dari 4 task (*identifikasi, kontrol, status, verifikasi*) adalah penerapan suatu *database* (disebut sebagai *configuration management database* - CMDB) yang berisi detail dari elemen-elemen dalam suatu perusahaan yang digunakan dalam mengatur IT *Service*. CMDB merupakan suatu pekerjaan "pendataan aset", karena CMDB berisi informasi yang terkait dengan perawatan, perpindahan, dan masalah yang terjadi dengan item- item yang ada di dalam CMDB. Selain itu CMDB juga berisi informasi tentang item - item yang sangat dibutuhkan oleh organisasi pelayanan IT, seperti hardware, software, dokumentasi, SDM.

## 2. Incident & Problem Management

*Incident & Problem Management* adalah sebuah penanganan dan pencegahan suatu kejadian/masalah yang akan mempengaruhi *IT Service* suatu organisasi (SLA Toolkit, 2002). Hal ini meliputi kepastian bahwa suatu masalah telah diperbaiki, mencegah terjadinya kembali masalah yang sama dan melakukan perawatan serta pencegahan untuk mengurangi masalah-masalah yang pernah terjadi muncul kembali. Tujuannya memperbarui atau membuat normal kembali *service* operasional secara cepat yang memungkinkan kekacauan pada bisnis menjadi sangat kecil.

*Incident & Problem Management* mempunyai peranan untuk meyakinkan pada penggunaan sumberdaya yang baik untuk mensupport bisnis, untuk mengembangkan, perbaikan dan menyimpan data yang berkaitan dengan *trouble* dengan sebaik mungkin. Contoh sebuah *trouble* adalah aplikasi tidak tersedia atau dalam keadaan *error* atau *hardware* sudah melampaui batas umur atau tidak dapat dipergunakan. *Incident & Problem Management* mempunyai tanggung jawab sebagai berikut:

1. Melakukan deteksi terhadap kejadian-kejadian dan mencatatnya
2. Membuat klasifikasi setiap kejadian dan membuat inisailasi bentuk *support* nya
3. Melakukan investigasi dan diagnostik
4. Membuat ketentuan dan memperbaikinya
5. Menyelesaikan setiap masalah / kejadian
6. Menentukan kejadian punya siapa, memantau, melacak dan melakukan komunikasi



Gambar 2.2 Incident Life Cycle

### 3. Change Management

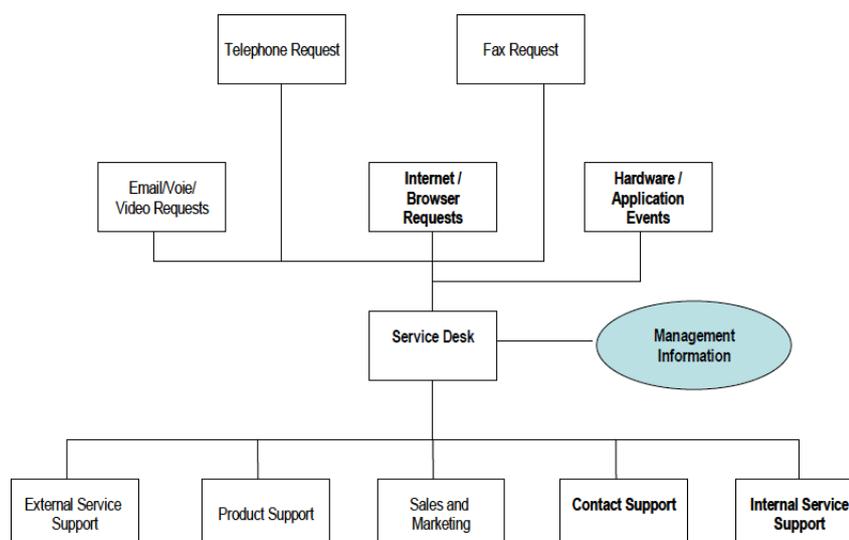
*Change Management* adalah suatu praktik yang memastikan bahwa semua perubahan terhadap configuration items tercatat, terencana dan disetujui. Task pada *change management* meliputi:

1. Memastikan bahwa ada suatu alasan yang terkait dengan bisnis pada setiap perubahan
2. Identifikasi spesifik item-item dan IT services yang akan terpengaruh oleh perubahan yang akan dilakukan.
3. Merencanakan perubahan
4. Melakukan pengetesan dari perubahan yang dilakukan
5. Mempunyai rencana cadangan jika perubahan yang dilakukan tidak berhasil atau tidak sesuai harapan.

### 4. Service Desk

*Service Desk* merupakan kontak pertama pelaku bisnis yang memanfaatkan IT services, jika terjadi sesuatu dengan IT services yang tidak mereka harapkan. Tujuan *Service Desk* adalah untuk melakukan pusat komunikasi antara *user* dan IT service Management, untuk menangani berbagai kejadian dan permintaan, untuk membagi dan menjadikan suatu *interface* untuk berbagai aktivitas lainnya misalnya, perubahan, masalah, konfigurasi, penghapusan, tingkat layanan serta kesinambungan layanan IT management (SLA Toolkit, 2002).

*Service Desk* diperlukan karena merupakan sebuah fungsi yang esensial untuk mengefektifkan *service management*. *Service Desk* dapat juga disebut sebagai *Helpdesk*, dimana *Helpdesk* merupakan suatu prinsip operasional antar *interface* dan pengguna IT.

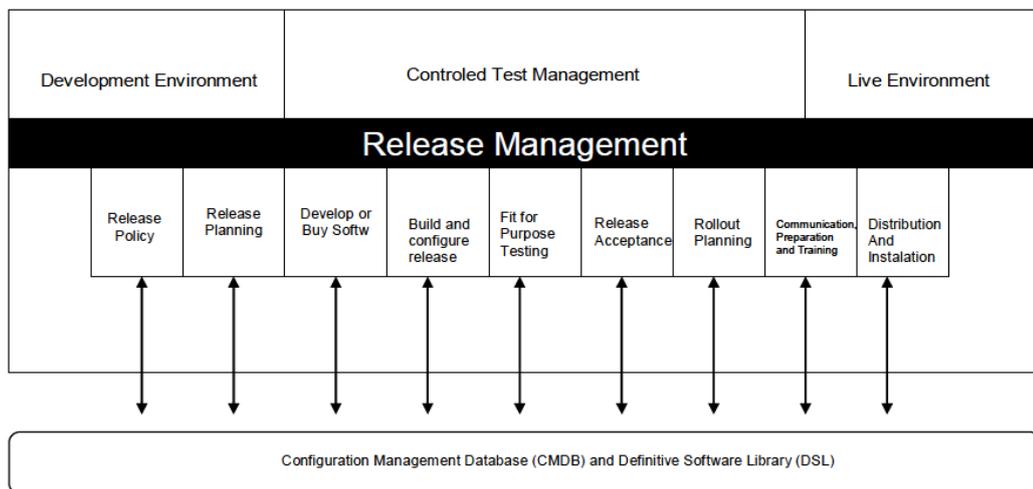


Gambar 2.3 Incident Registration Input

**5. Release management**

*Release management* adalah pengaturan semua konfigurasi *software* yang ada dalam suatu organisasi, dimana *task* ini bertanggung jawab terhadap pengaturan *software development*, instalasi, dan dukungan serta pembuatan *software*. *Release management* dibutuhkan untuk membuat daftar *hardware* yang sudah mengalami kondisi kritis, membuat daftar *software* yang tidak bisa digunakan lagi serta membuat daftar yang berhubungan dengan *software* atau *hardware* yang akan dilakukan penggantian.

*Release management* mengkoordinasikan berbagai organisasi yang memberi dukungan dan pengembang yang berhubungan *hardware*, *software*, penggabungan berbagai dokumen dan pendistribusiannya. Tanggung jawab dari *release management* adalah membuat rencana dan mengawasi dalam melakukan penggantian perangkat lunak dan penggabungan berbagai dokumen perangkat keras untuk mencapai keberhasilan.



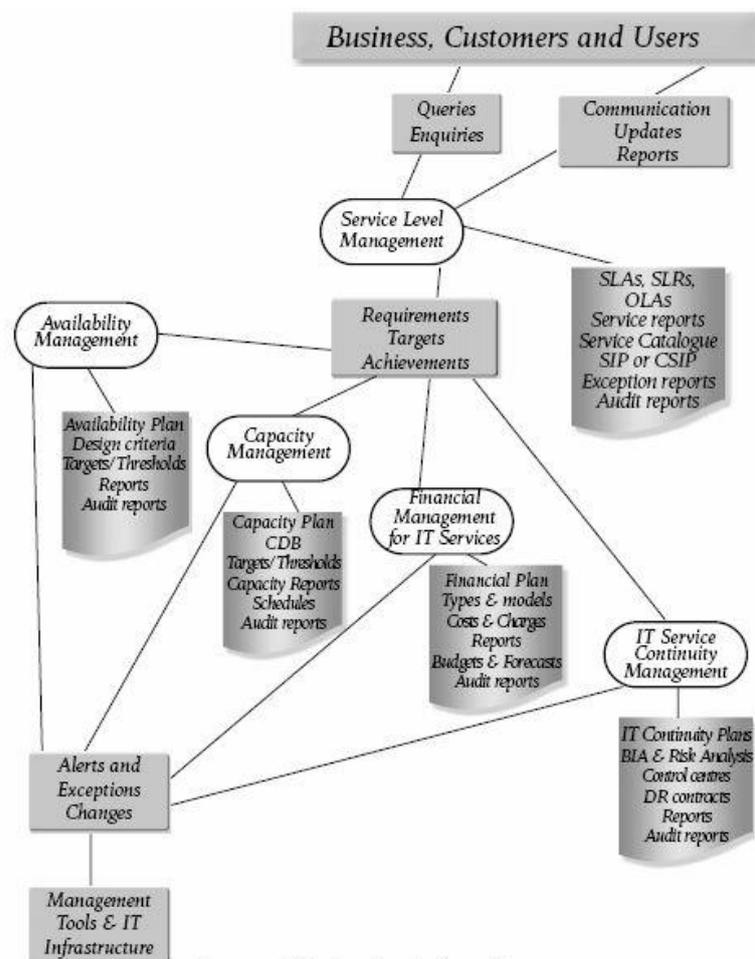
Gambar 2.4 Major Activities of Release Management

**2.2. Service Delivery**

Aspek dari *Service delivery* yang terdiri dari *Service Level Management*, *Financial management for IT Service*, *Capacity Management*, *IT Service Continuity* dan *Availability management* menjadi prinsip dalam pengembangan dan melakukan improvisasi dalam menjaga kualitas *IT Service Delivery*. Tujuan dari *Service Delivery* adalah untuk melakukan perbaikan dan secara berangsur-angsur melakukan pengembangan atau perbaikan dalam menggabungkan *IT service quality* sampai pada siklus yang tepat, pemantauan, laporan dan hasil review pada *IT service Achievements* dan sampai dapat mengganti tingkat layanan yang tidak sesuai.

### 2.3.Roadmap Service Management Consulting and Design Services

IT Service Management (ITSM) biasanya tidak berurusan langsung dengan bagian “back office” atau urusan operasional dari manajemen teknologi informasi. Terkadang pengetahuan ini merupakan arsitektur dari operasional dan bukan pembangunan teknologi. Sebagai contoh proses dari pembuatan *software* untuk penjualan, atau desain *microprocessor* tidak akan menjadi sesuatu yang fokus terhadap sebuah disiplin ilmu, tetapi lebih mengarah ke sebuah sistem komputer yang digunakan oleh bagian pemasaran dan pembangunan staf bisnis. Beberapa divisi dalam organisasi yang tidak berkecimpung dalam pengembangan teknologi, misalnya bidang *financial*, ritail dan *travel*, mempunyai sistem teknologi informasi yang signifikan dan tidak membuka rahasia dengan pelanggan. ITSM dapat diibaratkan sebagai sebuah *Enterprise Resource Planing* (ERP) untuk IT, dan meskipun dalam sejarahnya operasional di bidang teknologi informasi mungkin merupakan sebuah batasan aplikasi yang berlainan dengan kebanyakan aktivitas pada umumnya. Misalnya IT *portfolio management*, dan *Software engineering*. (Wikipedia, 2013).



Gambar 2.5 Service Delivery processes

## 2.4. Unified Modeling Language (UML)

*Unified Modeling Language* (UML) digunakan untuk melakukan pemodelan sistem/perangkat lunak dengan menggunakan tools yang ada. Dengan pemodelan menggunakan UML, rekayasa dan pengembangan perangkat dapat dilakukan dengan fokus pengembangan dan desain perangkat lunak terhadap:

1. Tinjauan umum bagaimana arsitektur sistem secara keseluruhan
2. Penelaah bagaimana objek-objek dalam sistem saling mengirimkan pesan (*message*) dan saling bekerjasama satu sama lain
3. Menguji apakah sistem/perangkat lunak sudah berfungsi seperti yang seharusnya
4. Dokumentasi sistem/perangkat lunak untuk keperluan-keperluan tertentu di masa yang akan datang

Setiap sistem yang kompleks seharusnya bisa dipandang dari sudut pandang yang berbedabeda sehingga bisa dilakukan pemahaman secara menyeluruh. Dalam upaya-nya tersebut, UML menyediakan sembilan jenis diagram yang dapat dikelompokkan berdasarkan sifatnya yang statis ataupun dinamis. Kesembilan jenis diagram untuk UML adalah:

1. **Use-Case Diagram** - bersifat statis, memperlihatkan himpunan *use-case* dan aktor-aktor. Diagram ini sangat penting terutama untuk memodelkan ataupun mengorganisasikan perilaku dari sistem yang dibutuhkan pengguna
2. **Class Diagram** - bersifat statis tetapi sering pula memuat kelas-kelas aktif dan memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi
3. **Statechart Diagram** - bersifat dinamis yang memperlihatkan *state – state* dari sistem, memuat *state*, transisi, *event*, serta aktivitas. Penting untuk memperlihatkan sifat dinamis dari antarmuka (*interface*), kelas, kolaborasi, terutama penting pada pemodelan system-sistem yang reaktif
4. **Activity Diagram** - bersifat dinamis. Merupakan tipe khusus dari *diagram state* yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem
5. **Sequence Diagram** - bersifat dinamis yang menekankan pada pengiriman pesan (*message*) dalam suatu waktu tertentu
6. **Collaboration Diagram** - bersifat dinamis yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan (*message*)
7. **Component Diagram** - bersifat statis. diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan ke dalam satu atau lebih kelas-kelas, antarmuka (*interface*) ataupun kolaborasi

8. **Diagram Objek** - bersifat statis, memperlihatkan objek-objek serta serta relasi-relasi antar-objek. Selain itu juga memperlihatkan instansiasi statis dari segala sesuatu yang dijumpai pada diagram kelas
9. **Deployment Diagram** - bersifat statis, diagram memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*). Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*)

### 3. PERANCANGAN APLIKASI, FRAMEWORK DAN PROPERTI KOLABORASI

Dari setiap analisa yang telah dibahas kemudian dalam membangun Framework Aplikasi perlu diperhatikan penanganan property untuk perancangan dan rekayasa perangkat lunak sebagai berikut:

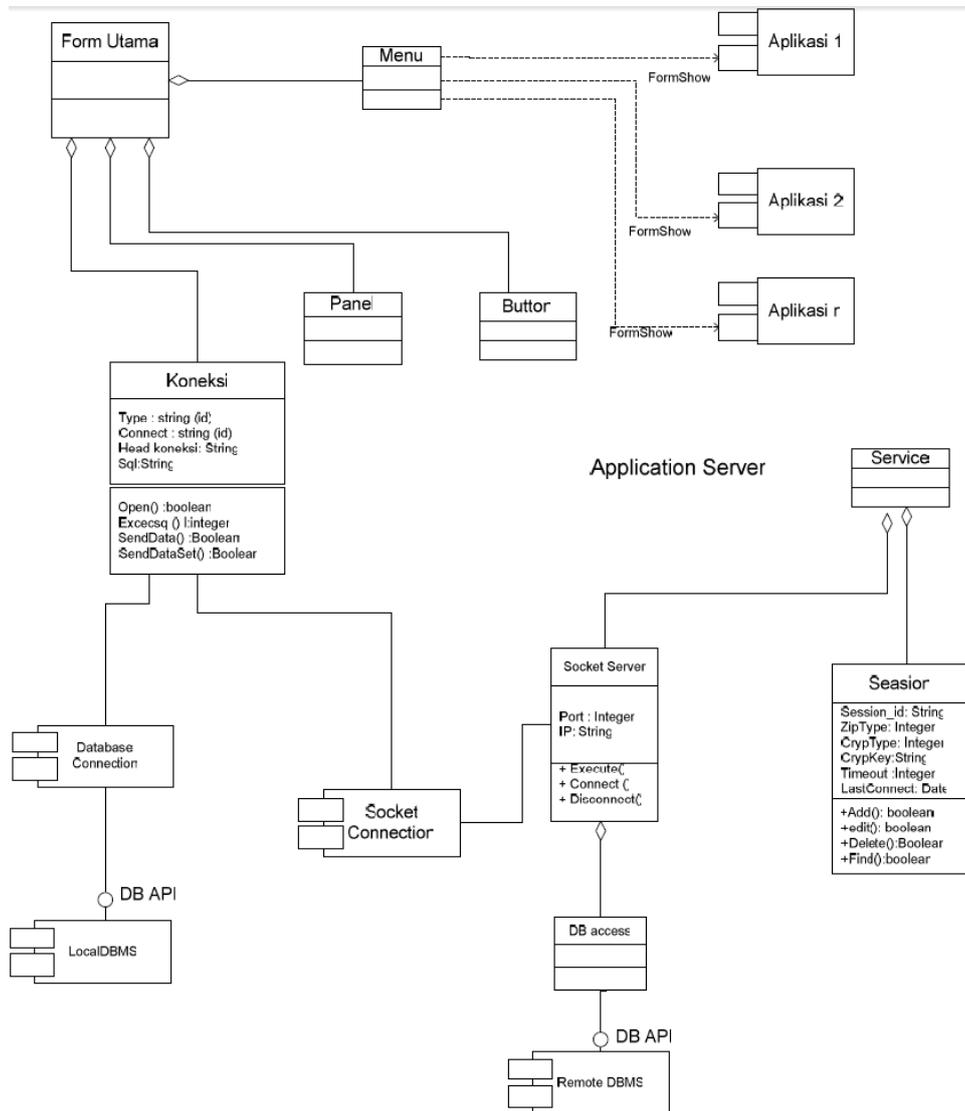
1. **User Control** bertanggung jawab mengelola daftar petugas yang berhak menggunakan sistem, dan melakukan verifikasi *login / logout*.
2. **Group Control** bertanggung jawab dalam mengelolah group dan pengelolaan keanggotaan petugas dalam group.
3. **Sesion Management** bertanggung jawab mengelola sesi yang aktif menyimpan informasi tipe kompresi, tipe enkripsi dan kunci enkripsi yang digunakan. Selain itu sesi manajemen petugas yang beroperasi adalah petugas yang memiliki sesi.
4. **Acces Control** bertanggung jawab mengelola kewenangan dari setiap grup dan anggota grup terhadap sejumlah komponen yang tersedia. *User Interface* bertanggung jawab memberikan antar muka kepada petugas untuk berinteraksi dengan sistem.
5. **Atomicity** dengan fungsi *commit / rollback* menjamin bahwa transaksi diakui semua atau dibatalkan semua. Sistem harus menjalin status rangkaian transaksi, jangan sampai sebagian sukses, sedangkan sebagian lagi dibatalkan.
6. **Concurrency Control** menjamin dua transaksi atau lebih yang meminta layanan dapat dilayani dengan tetap menjamin keberhasilan transaksi.
7. **Concurrency Contol** bertanggung jawab mencegah terjadinya *deadlock* antar proses untuk memperebutkan sumber daya yang sama. Akses konkuren kemungkinan benar terjadi di server aplikasi, seperti pada saat akses basis data dan *log file*.
8. **Coupling dan Cohesion** merupakan ukuran kualitas disain modular. Dalam membuat disain modular, derajat *coupling* rendah, ketergantungan antara komponen harus rendah, akses data *global* ditindak, serta antarmuka komponen

yang sederhana. Sedangkan derajat kohesi dapat ditingkatkan dengan spesialisasi komponen.

### 3.1. Modul Konseptual

Model *conceptual modul* Aplikasi dapat dibangun berdasarkan analisa kebutuhan perangkat lunak pada suatu organisasi yang telah dikembangkan. Antar muka terdiri form utama, *panel menu* dan *button*. Menu bertanggung jawab memberikan *link* panggilan ke komponen aplikasi. Komponen aplikasi yang ditangani menu sesuai dengan sistem yang dikembangkan (Jacobson *et al*, 1992). Banyak sedikitnya komponen aplikasi tergantung pada luas sempitnya sistem yang dibangun.

Untuk melakukan operasi basis data, antara muka memanggil komponen data komunikasi melalui antarmuka kelas Koneksi. Selanjutnya kelas ini akan memanggil komponen koneksi basis data untuk operasi basis data lokal dan koneksi socket untuk operasi basis data remote melalui *server* aplikasi. *Server* aplikasi menerima perintah dari klien melalui *server* socket dan meneruskan ke DBMS *remote* menggunakan kelas *Db\_Access*.



Gambar 3.1 Model konseptual *Framework* Aplikasi

### 3.2. Perancangan Layanan Berdasarkan ITIL Framework

IT *Service Management* dalam ITIL dibedakan menjadi *incident* dan *problem*. Kemudian dalam IT *Service Management* itu sendiri terdiri atas *service support* dan *service delivery*. *Service support* sendiri menjelaskan proses dan komponen dalam operasional IT sehari-hari seperti *Incident Management*, *Problem Management*, *Change Management*, *Configuration Management* dan *Release Management*. *Service support* menjelaskan proses *support* dari IT untuk operasional sehari-hari, dan bagaimana menangani suatu *incident* sebelum akhirnya masuk ke dalam proses *problem management* (Maurer, et al, 2000). *Service desk* sebagai *single point of contact* merupakan bagian tersendiri yang akan mengolah setiap proses yang masuk dan keluar. *Release Management* mempunyai tugas melakukan testing suatu teknologi baru sebelum diimplementasi.

*Build knowledge based* berfungsi membantu mempercepat penyelesaian suatu *incident*. Hal lain dari IT *Service Management* adalah *Service Delivery*, dimana komponen atau proses dari *Service Delivery* ini adalah merupakan sebuah *Financial Management for IT Services, Availability Management, Capacity Management* dan *IT Service Continuity Management* (SLA Toolkit, 2002). *Service Level Management* pada dasarnya adalah *Service Delivery* dengan *tactical* proses, dimana ini artinya adalah merupakan sebuah dukungan untuk *IT Service Management* dalam jangka waktu yang lebih lama bukan untuk operasional sehari hari, seperti:

1. IT Service Continuity Management
2. service level agreement (SLA)
3. Dan lain sebagainya yang merupakan kegiatan operasional organisasi harian

#### 4. KESIMPULAN

Dari pembahasan dan rekayasa sistem informasi yang telah dilakukan, akan diperoleh sebuah *roadmap* dan rencana pengembangan serta Teknologi Informasi yang handal yang berpedoman pada *Information Technology Infrastructure Library* (ITIL). Dalam pembangunan sistem dan aplikasi akan sangat membantu manajemen dalam memperoleh informasi sebagai pendukung keputusan terutama bagi sebuah organisasi dengan skala *Enterprise*.

Adapun manfaat dari pengembangan sistem/aplikasi dengan berpedoman terhadap *IT Service Management* (ITSM) dalam *framework* ITIL akan memberikan nilai tambah pada suatu organisasi, antara lain yaitu:

1. Terbangunnya standar arsitektur pengembangan sistem management yang baik
2. Desain perangkat lunak yang tidak tergantung pada struktur organisasi, sehingga tidak menjadi hambatan dalam pengembangan dan implementasi sistem yang sudah ada sebelumnya
3. Pengembangan sistem yang selalu akan mengacu kepada blueprint dan roadmap yang telah ditetapkan
4. Implementasi sistem informasi akan lebih cepat karena pengembangan sistem dan aplikasi tidak tergantung pada perubahan organisasi

#### 5. DAFTAR PUSTAKA

- [1] Colin Rudd, 2004, An Introductory Overview of ITIL® Version 1.0am , itEMS Ltd © Copyright itSMF
- [2] IBM Global, 2004, IBM and the IT Infrastructure Library, ©Copyright IBM Corporation, Services Route 100 Somers, NY 10589 U.S.A
- [3] Ivar Jacobson, Magnus Christerson, Patrik Jonson, and Gunnar Overgaard, 1992, Object-Oriented Software Engineering: A Use Case Driven Approach, Addison-Wesley
- [4] Wikipedia, 2013, ITSM, <http://id.wikipedia.org/wiki/ITSM>, diakses 20 pebruari 2013.
- [5] SLA Toolkit, 2002, Service Level Agreement, Guidance & Instructions, ©Copyright Protected Easytec Solutions

- [6] SLA Toolkit, 2002, Service Level Agreement, Template, @Copyright Protected Easytec Solutions
- [7] W. Maurer, R. Matlus, N. Frey, 2000, Strategic Analysis Report, GartnerGroup RAS Services R-11-3353