BAB II

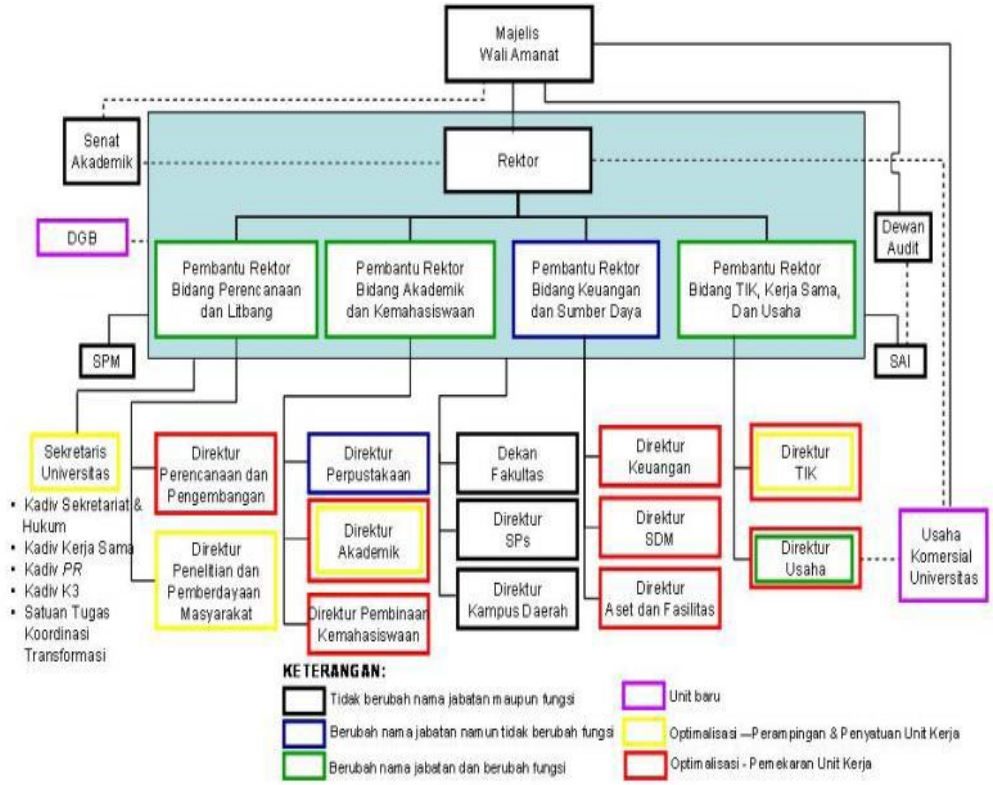
TINJAUAN PUSTAKA

2.1 Tinjauan Instansi

2.1.1 Sejarah Umum Perusahaan

Sejak ditetapkan sebagai PT. BHMN, secara bertahap UPI melakukan reorganisasi. Dengan merujuk pada Ketentuan Peralihan pada PP No.6/2004. Pada tahun 2004 Rektor UPI telah membentuk organ -organ utama UPI, yakni Senat Akademik (SA) dan mengusulkan pengangkatan anggota Majelis Wali Amanat (MWA) kepada Mendiknas. Selanjutnya pada bulan Juni 2005 MWA UPI telah berhasil menetapkan Rektor UPI BHMN periode kepemimpinan 2005 - 2010 dan kemudian mekanisme yang demokratis Rektor telah menetapkan empat Pembantu Rektor, masing-masing untuk bidang akademik, bidang umum dan sumber daya, kemahasiswaan dan alumni, penelitian dan pengembangan. Lebih lanjut pada bulan Agustus 2005 MWA UPI membentuk Dewan Audit (DA). Namun perlu dikemukakan bahwa struktur organisasi ini masih sementara, sebab reorganisasi masih terus dilakukan menuju struktur organisasi yang bersesuaian dengan PP No.6/2004 dan Anggaran Rumah Tangga (ART) UPI yang tengah dalam proses. Pada awalnya, organisasi Teknologi Informasi (TI) di UPI berada di BAPSI (Biro Perencanaan dan sistem informasi) dan 4 unit pelaksana teknis (UPT UPI net, UPT Puskom, UPT P3MP dan UPT Pendilkom) dalam memberikan pelayanan dan pengelolaan TIK. Berdasar pada tuntutan perubahan UPI menjadi lembaga

Pendidikan Tinggi BHMN dan efisiensi penyelenggaraan pendidikan, saat ini maka UPT-UPT tersebut kemudian dilebur menjadi satu direktorat yang menangani TIK, yaitu Direktorat Teknologi Informasi dan Komunikasi (Direktorat TIK). Direktorat TIK UPI berada di bawah Rektor. Dengan kewenangannya Rektor dapat menetapkan garis koordinasi, baik tugas teknis maupun finansial. Garis 30 kerja Direktorat TIK tidak hanya meliputi di kampus pusat, namun juga meliputi kampus UPI Cibiru, Sumedang, Tasikmalaya, Purwakarta dan Serang. Peran dan fungsi Direktorat TIK di UPI menjadi vital bagi UPI, hal ini disebabkan karena teknologi informasi dijadikan tulang punggung sistem tata kelola universitas menuju tata kelola universitas yang baik yang transparan dan akun tabel. Selain menjadi tulang punggung tata kelola organisasi, Direktorat TIK harus memberikan pelayanan yang optimal bagi civitas akademika UPI baik dalam bidang palayanan sistem informasi, perawatan jaringan yang dimiliki dan pelayanan akses internet. Direktorat TIK secara struktural organisasi bertanggung jawab kepada Pembantu Rektor Bidang Teknologi Informasi dan komunikasi, Kerjasama dan Usaha. seperti yang digambarkan dibawah ini.



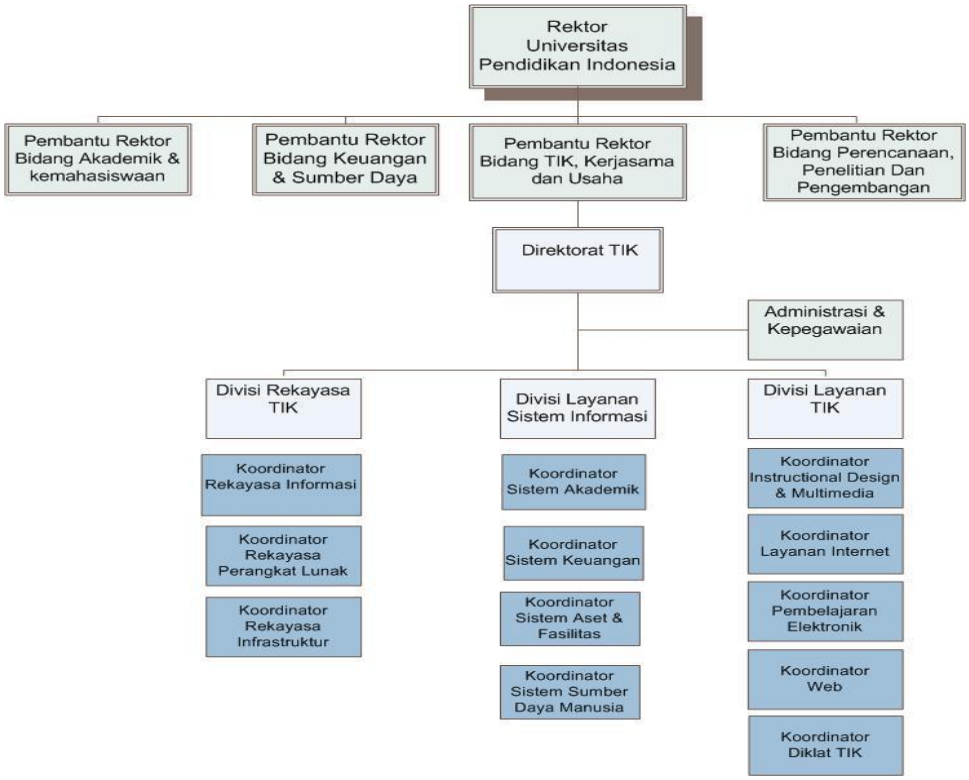
Gambar 2. 1 Struktur Organisasi UPI

Direktorat TIK didirikan tanggal 6 Juli 2007 dengan SK Rektor No.3792/H40/KL/2007. Pembentukkan Direktorat TIK sesuai dengan rencana strategis UPI tahun 2006-2010 bidang teknologi informasi dan komunikasi yang mencantumkan antara lain:

1. Memperluas pembangunan infrastruktur TIK;
2. Menyempurnakan sistem informasi manajemen universitas yang lengkap, akurat dan mutahir;
3. Meningkatkan kapasitas dan fungsi UPI Net;
4. Mengembangkan *e-learning*, dan *e-library*;

5. Mengembangkan *e-management* baik dalam bidang akademik maupun non-akademik;
6. Mengembangkan perpustakaan berbasis TIK (*e-library*);
7. Ketercapaian realisasi program-program tersebut dapat dilihat melalui indicator berikut:
8. Terpasangnya infrastruktur TIK yang menghubungkan seluruh unit di kampus induk dan kampus daerah dengan tingkat keamanan yang tinggi;
9. Meningkatnya *bandwidth* menjadi 15 Mbps. Sebanyak 25% sivitas akademika dan tenaga administrasi universitas dapat memanfaatkan layanan internet;
10. Sebanyak 10 program studi memanfaatkan jaringan TIK untuk menyelenggarakan perkuliahan dengan sistem *e-learning*.
11. Sistem informasi manajemen universitas yang dapat memberikan 75% layanan data dan informasi secara lengkap, akurat, dan mutahir;
12. Layanan *e-library* untuk 25% mahasiswa dan dosen.

Direktorat TIK dipimpin oleh seorang direktur yang membawahi 3 (tiga) kepala divisi, yaitu Divisi Rekayasa TIK, Divisi Layanan Teknologi Informasi dan Divisi Layanan Sistem Informasi. Struktur Organisasi Direktorat TIK dapat dilihat pada gambar berikut:



Gambar 2. 2 Struktur Organisasi Direktorat TIK

2.1.2 Deskripsi Jabatan

2.1.2.1 Visi

UPI menempatkan posisi, peran dan fungsinya dalam pembangunan pendidikan nasional dan pembangunan Sumber Daya Manusia (SDM) indonesia secara menyeluruh dalam jangka panjang, untuk memberikan kesempatan kepada setiap orang untuk tumbuh dan berkembang menjadi manusia berbudaya, beradab, efisien dan efektivitas proses pendidikan dalam upaya mewujudkan makna proses pendidikan, memiliki peran mendasar dan strategis. *Globalisasi* sebagai suatu fenomena khusus dalam peradaban manusia telah mendudukan teknologi informasi

dan komunikasi menjadi alat dalam percepatan prosesnya. Sebagai bentuk komitmen terhadap proses globalisasi, Direktorat TIK akan berada di belakang proses perkembangan UPI menjadi universitas terkemuka dalam percaturan pendidikan nasional dan internasional serta pengembangan manusia dalam upaya penelitian, pengembangan, dan implementasi teknologi informasi dan komunikasi dalam dunia pendidikan.

2.1.2.2 Misi

Tujuan didirikannya Direktorat TIK adalah untuk menciptakan Universitas Pendidikan Indonesia menjadi kampus yang ilmiah, efektif, dan religious dengan berbasis teknologi komunikasi dan informasi. Tugas dan fungsi Direktorat TIK meliputi:

1. Membangun dan menciptakan infrastruktur teknologi informasi dan komunikasi yang diperlukan UPI.
2. Mengembangkan, menyediakan, dan mengaplikasikan sistem informasi berbasis teknologi informasi dan komunikasi yang diperlukan UPI.
3. Mengembangkan, menyediakan, dan mengaplikasikan sistem informasi berbasis teknologi informasi dan komunikasi yang diperlukan UPI.
4. Memberikan pelayanan secara teknis maupun non teknis yang berhubungan dengan teknologi informasi dan komunikasi.
5. Mengadakan penelitian, pendidikan, pelatihan, dan pengabdian yang berhubungan dengan teknologi informasi dan komunikasi
6. Meningkatkan dan menyediakan tenaga profesional dalam system informasi dan TIK.

7. Meningkatkan dan memanfaatkan berbagai fasilitas dalam meningkatkan kemandirian perguruan tinggi.
8. Menjalin kerjasama yang saling menguntungkan dengan berbagai pihak untuk kemajuan UPI.

2.1.3 Tugas Pokok dan Fungsi

2.1.3. 1 Direktur

1. Menyelenggarakan dan mengkoordinasikan pelaksanaan kegiatan layanan sistem informasi dan layanan teknologi informasi dan komunikasi
2. Menyelenggarakan penyusunan rencana kegiatan bidang Teknologi Informasi dan komunikasi
3. Mengkoordinasikan kegiatan pengembangan bidang Teknologi Informasi dan Komunikasi
4. Mengevaluasi dan mengendalikan pelaksanaan kegiatan Teknologi Informasi dan komunikasi
5. Melaporkan Tugas pokok dan fungsi dalam Bidang Teknologi Informasi dan komunikasi kepada rektor melalui Pembantu Rektor Bidang TIK, Kerjasama dan Usaha

2.1.3. 2 Divisi - Divisi Direktorat TIK

Didalam menjalankan operasional hariannya Direktorat TIK dipimpin oleh seorang direktur yang membawahi tiga divisi dan bagian administrasi & kepegawaian yaitu:

1. Divisi Rekayasa Teknologi Informasi dan Komunikasi
2. Divisi Layanan Sistem Informasi
3. Divisi Layanan Teknologi Informasi dan komunikasi

Pada setiap divisi terdapat koordinator yang bertugas membantu kepala divisi dalam menjalankan tugas pokok dan fungsinya. Deskripsi tugas pokok dan fungsi divisi dan koordinator adalah sebagai berikut:

A. Divisi Rekayasa Teknologi Informasi dan Komunikasi

1. Koordinator Rekayasa Informasi
   a. Pengembangan dan Pembaharuan rencana strategi TIK
   b. Pengembangan dan pembaharuan cetak biru SIUPI
   c. Mengembangkan menyusun dan mempublikasikan laporan baik yang bersifat rutin maupun bersifat khusus
   d. Pengembangan, sosialisai, dan pemanfaatan sistem pendukung pengambilan keputusan (*Decision support system*)
   e. Pengembangan, sosialisasi, dan pemanfaatan sistem informasi Eksekutif (*Executive Information System*)
   f. Menyelenggarakan kegiatan bisnis intelejen dan Rekayasa TIK dan Pengembangan Sistem informasi
   g. Melakukan kegiatan penelitian dan pengembangan rekayasa informasi
   h. Melakukan pelayanan kepada masyarakat dalam pengembangan rekayasa informasi

2. Koordinator Rekayasa Perangkat Lunak
    a. Merencanakan, menyediakan, mensosialisasikan, memanfaatkan dan memelihara perangkat lunak yang dibutuhkan UPI BHMN
    b. Pengembangan standarisasi perangkat lunak dalam pengembangan dan pemanfaatan perangkat lunak di lingkungan UPI BHMN
    c. Mengembangkan metoda pengembangan perangkat lunak
    d. Melakukan kegiatan rekayasa perangkat lunak untuk memenuhi kebutuhan sistem manajemen informasi dan penyelenggaraan aktifitas pendidikan dan penelitian
    e. Pengelolaan proyek pengembangan perangkat lunak
    f. Melakukan dokumentasi perangkat lunak
    g. Mengembangkan dan mensosialisasikan manual operation perangkat lunak
    h. Melakukan pelayanan kepada masyarakat dalam pengembangan perangkat lunak.
3. Koordinator Rekayasa Infrastruktur
    a. Mengembangkan desain infrastruktur TIK
    b. Melakukan Instalasi dan konfigurasi infrastruktur TIK
    c. Melakukan monitoring dan pemeliharaan infrastruktur TIK
    d. Mengembangkan Standard *Operating Procedure* pemanfaatan infrastruktur TIK
    e. Melakukan kegiatan penelitian dan pengembangan rekayasa infrastruktur TIK

f. Memberikan pelayanan kepada masyarakat dalam pengembangan dan pemanfaatan infrastruktur TIK

4. Koordinator Diklat TIK
   a. Menyusun proyeksi kebutuhan program diklat TIK SDM UPI (4 Tahun)
   b. Mengembangkan program dan kurikulum Diklat TIK moda reguler dan terbuka
   c. Menyiapkan SDM bidang teknologi informasi yang memenuhi standard kualifikasi yang ditentukan
   d. Menyelenggarakan program pelatihan baik yang bersifat eksternal maupun publik yang disesuaikan dengan kebutuhan dan proyeksi kebutuhan masa yang akan datang
   e. Membuat standard kualifikasi SDM dibidang TIK
   f. Menyusun kurikulum untuk pendidikan dan pelatihan TIK
   g. Menyusun dan membuat bahan ajar pelatihan TIK
   h. Membuat dan mengembangkan alat tes
   i. Menyelenggarakan sertifikasi bagi civitas akademika UPI dan umum di bidang TIK
   j. Menyelenggarakan program pelatihan baik pre-*service* program maupun in-*service* program
   k. Menyelenggarakan program pelatihan yang bersifat pelayanan kepada publik

B. Divisi Layanan Sistem Informasi

1. Koordinator Sistem Informasi Akademik
    a. Melakukan koordinasi dan analisis kebutuhan sistem informasi akademik dengan unit yang menangani akademik
    b. Melakukan perencanaan pengembangan sistem informasi akademik
    c. Mengembangkan, mengsosialisasikan, mengimplementasikan, dan komputerisasi sistem akademik
    d. Melakukan pengaturan, pengawasan, pemeliharaan, perbaikan dan pengamanan aplikasi komputer sistem akademik
2. Koordinator Sistem Informasi Keuangan
    a. Melakukan koordinasi dan analisis kebutuhan sistem informasi keuangan dengan unit yang menangani keuangan
    b. Melakukan perencanaan sistem informasi keuangan
    c. Mengembangkan, mensosialisasikan, mengimplementasikan, dan komputerisasi sistem keuangan
    d. Melakukan pengaturan, pengawasan, pemeliharaan, perbaikan, dan pengamanan aplikasi komputer sistem keuangan
3. Koordinator Sistem Informasi Aset dan Fasilitas
    a. Melakukan koordinasi dan analisis kebutuhan sistem informasi asset dan fasilitas dengan unit yang menangani aset dan fasilitas
    b. Melakukan perencanaan pengembangan sistem informasi aset dan fasilitas

c. Mengembangkan, mensosialisasikan, mengimplementasikan, dan komputerisasi sistem aset dan fasilitas
d. Melakukan pengaturan, pengawasan, pemeliharaan, perbaikan, dan pengamanan aplikasi komputer sistem keuangan

4. Koordinator Sistem Informasi Sumber Daya manusia
   a. Melakukan koordinasi dan analisis kebutuhan sistem informasi Sumber daya manusia dengan unit yang menangani Sumber daya manusia
   b. Melakukan perencanaan pengembangan sistem informasi Sumber daya manusia
   c. Mengembangkan, mensosialisasikan, mengimplementasikan, dan komputerisasi sistem informasi Sumber daya manusia
   d. Melakukan pengaturan, pengawasan, pemeliharaan, perbaikan, dan pengamanan aplikasi komputer sistem informasi Sumber daya manusia

C. Divisi Layanan Teknologi Informasi

1. Koordinator Desain Instruksional dan Multimedia
   a. Membuat dan merancang instruksional desain dan multimedia untuk meningkatkan kualitas sistem pendidikan di UPI
   b. Mengadakan riset dan pengembangan desain instruksional dan multimedia
   c. Memberikan layanan pengemabngan desain instruksional dan multimedia kepada masyarakat

2. Koordintor Layanan Internet
    a. Mengelola dan mengatur layanan internet UPI secara efektif dan efisien
    b. Mengembangkan sistem keamanan, sistem layanan dan administrasi dan sistem kendali dan monitoring
    c. Mengembangkan, menggunakan dan memelihara *Access Control System* (ACS)
    d. Melayani dan membantu *user* ketika ada masalah
3. Koordinator Pembelajaran Elektronik
    a. Mengembangkan sistem pembelajaran elektronik melalui *learning management system* (LMS)
    b. Mengembangkan sistem evaluasi perkuliahan secara *online*
    c. Melaksanakan digitalisasi bahan ajar
4. Koordinator Web Upi
    a. Mengembangkan dan mengelola website resmi UPI (www.upi.edu) dan aplikasi web upi
    b. Pengaturan domain upi.edu dan upi.ac.id
    c. Pengaturan dan pengawasan terhadap semua website unit - unit di UPI (dibawah domain upi.edu dan upi.ac.id)
    d. Riset dan pengembangan aplikasi web untuk dunia pendidikan

2.2 ITSM (*Information Technology Service Management)*

Manajemen Layanan Teknologi Informasi adalah suatu metode pengelolaan sistem teknologi informasi (TI) yang secara filosofis terpusat pada perspektif konsumen layanan TI terhadap bisnis perusahaan. ITSM merupakan kebalikan dari pendekatan manajemen TI dan interaksi bisnis yang terpusat pada teknologi. Istilah ITSM tidak berasal dari suatu organisasi, pengarang, atau pemasok tertentu dan awal penggunaan frase inipun tidak jelas kapan dimulainya.

ITSM berfokus pada proses dan karenanya terkait dan memiliki minat yang sama dengan kerangka kerja dan metodologi gerakan perbaikan proses (seperti TQM, *Six Sigma*, *Business Process Management*, dan CMMI). Disiplin ini tidak mempermasalahkan detail penggunaan produk suatu pemasok tertentu atau detail teknis suatu sistem yang dikelola, melainkan berfokus pada upaya penyediaan kerangka kerja untuk menstrukturkan aktivitas yang terkait dengan TI dan interaksi antara personil teknis TI dengan pengguna teknologi informasi.

ITSM umumnya menangani masalah operasional manajemen teknologi informasi (kadang disebut *operations architecture*, arsitektur operasi) dan bukan pada pengembangan teknologinya sendiri. Contohnya, proses pembuatan perangkat lunak komputer untuk dijual bukanlah fokus dari disiplin ini, melainkan sistem komputer yang digunakan oleh bagian pemasaran dan pengembangan bisnis di perusahaan perangkat lunak-lah yang merupakan fokus perhatiannya. Banyak pula perusahaan non-teknologi, seperti pada industri keuangan, ritel, dan

pariwisata, yang memiliki sistem TI yang berperan penting, walaupun tidak terpapar langsung kepada konsumennya.

Sesuai dengan fungsi ini, ITSM sering dianggap sebagai analogi disiplin ERP pada TI, walaupun sejarahnya yang berakar pada operasi TI dapat membatasi penerapannya pada aktivitas utama TI lainnya seperti manajemen portfolio TI dan rekayasa perangkat lunak.

Kerangka kerja (*framework*) yang dianggap dapat memberikan contoh penerapan ITSM di antaranya:

1. *Information Technology Infrastructure Library* (ITIL)
2. *Control Objectives for Information and Related Technology* (COBIT)
3. *Software Maintenance Maturity Model*
4. *PRM-IT IBM's Process Reference Model for IT*
5. *Application Services Library* (ASL)
6. *Business Information Services Library* (BISL)
7. *Microsoft Operations Framework* (MOF)
8. *eSourcing Capability Model for Service Providers* (eSCM-SP) dan *eSourcing Capability Model for Client Organizations* (eSCM-CL) dari *ITSqc for Sourcing Management*.

2.2.1 ITIL (*Information Technology Infrastructure Library*)

ITIL adalah suatu rangkaian konsep dan teknik pengelolaan infrastruktur, pengembangan, serta operasi teknologi informasi (TI). ITIL diterbitkan dalam suatu rangkaian buku yang masing-masing membahas suatu topik pengelolaan TI. Nama ITIL dan IT *Infrastructure Library* merupakan merek dagang terdaftar dari *Office of Government Commerce* (OGC) Britania Raya. ITIL memberikan deskripsi detil tentang beberapa praktik TI penting dengan daftar cek, tugas, serta prosedur yang menyeluruh yang dapat disesuaikan dengan segala jenis organisasi TI.

Walaupun dikembangkan sejak dasawarsa 1980-an, penggunaan ITIL baru meluas pada pertengahan 1990-an dengan spesifikasi versi keduanya (ITIL v2) yang paling dikenal dengan dua set bukunya yang berhubungan dengan ITSM (*IT Service Management*), yaitu *Service Delivery* (*Antar Layanan*) dan *Service Support* (*Dukungan Layanan*).

Pada 30 Juni 2007, OGC menerbitkan versi ketiga ITIL (ITIL v3) yang intinya terdiri dari lima bagian dan lebih menekankan pada pengelolaan siklus hidup layanan yang disediakan oleh teknologi informasi. Kelima bagian tersebut adalah:

1. *Service Strategy*
2. *Service Design*
3. *Service Transition*

4. *Service Operation*
5. *Continual Service Improvement*

Kelima bagian tersebut dikemas dalam bentuk buku, atau biasa disebut sebagai *core guidance publications*. Setiap buku dalam kelompok utama ini berisi:

1. *Practice fundamentals* – menjelaskan latar belakang tahapan *lifecycle* serta kontribusinya terhadap pengelolaan layanan TI secara keseluruhan.
2. *Practice principles* – menjelaskan konsep - konsep kebijakan serta tata kelola tahanan *lifecycle* yang menjadi acuan setiap proses terkait dalam tahapan ini.
3. *Lifecycle processes* and *activities* – menjelaskan berbagai proses maupun aktivitas yang menjadi kegiatan utama tahapan *lifecycle*. Misalnya proses *financial management* dan *demand management* dalam tahapan *Service Strategy*.
4. *Supporting organization structures and roles* – proses - proses ITIL tidak akan dapat berjalan dengan baik tanpa *defini roles* dan *responsibilities*. Bagian ini menjelaskan semua aspek yang terkait dengan kesiapan model dan struktur organisasi.
5. *Technology considerations* – menjelaskan solusi-solusi otomatisasi atau *software* ITIL yang dapat digunakan pada tahapan *lifecycle*, serta persyaratannya.

6. *Practice Implementation* – berisi acuan/panduan bagi organisasi TI yang ingin mengimplementasikan atau yang ingin meningkatkan proses-proses ITIL.
7. *Complementary guideline* – berisi acuan model-model best practice lain selain ITIL yang dapat digunakan sebagai referensi bagian tahapan *lifecycle.*
8. *Examples and templates* – berisi template maupun contoh-contoh pengaplikasian proses.

Di samping buku-buku dalam *core guidance publications*, ada juga *complementary guidance*. Dimana buku-buku dalam kategori nantinya dimaksudkan untuk memberikan model, acuan dan panduan bagi penerapan ITIL pada sektor-sektor tertentu seperti jenis industri tertentu, tipe organisasi serta arsitektur teknologi. Dengan demikian, ITIL akan dapat lebih diterima serta diadaptasi sesuai dengan lingkungan serta *behaviour* dari setiap organisasi TI.

2.2.1.1 Siklus Layanan ITIL

Kelima bagian ITIL yang seperti tersebut di atas biasanya disebut juga sebagai bagian dari sebuah siklus. Dikenal pula dengan sebutan Sikuls Layanan ITIL. Secara singkat, masing-masing bagian dijelaskan sebagai berikut.

1. *Service Strategy*

Inti dari ITIL *Service Lifecycle* adalah *service strategy*. *Service Strategy* memberikan panduan kepada pengimplementasi ITSM pada bagaimana memandang konsep ITSM bukan hanya sebagai sebuah kemampuan organisasi (dalam memberikan, mengelola, serta mengoperasikan layanan TI), tapi juga sebagai sebuah aset strategis perusahaan. Panduan ini disajikan dalam bentuk prinsip-prinsip dasar dari konsep ITSM, acuan-acuan serta proses - proses inti yang beroperasi di keseluruhan tahapan ITIL *Service Lifecycle.*

Topik-topik yang dibahas dalam tahapan *lifecycle* ini mencakup pembentukan pasar untuk menjual layanan, tipe-tipe, dan karakteristik penyedia layanan internal maupun eksternal, aset-aset layanan, konsep portofolio layanan serta strategi implementasi keseluruhan ITIL *Service Lifecycle*. Proses-proses yang dicakup dalam *Service Strategy*, di samping topik-topik di atas adalah:

1. *Service Portfolio Management*
2. *Financial Management*
3. *Demand Management*

Bagi organisasi TI yang baru akan mengimplementasikan ITIL, *Service Strategy* digunakan sebagai panduan untuk menentukan tujuan/sasaran serta ekspektasi nilai kinerja dalam mengelola layanan TI

serta untuk mengidentifikasi, memilih serta memprioritaskan berbagai rencana perbaikan operasional maupun organisasional di dalam organisasi TI.

Bagi organisasi TI yang saat ini telah mengimplementasikan ITIL, *Service Strategy* digunakan sebagai panduan untuk melakukan *review strategis* bagi semua proses dan perangkat (*roles*, *responsibilities*, teknologi pendukung, dll) ITSM di organisasinya, serta untuk meningkatkan kapabilitas dari semua proses serta perangkat ITSM tersebut.

2. *Service Design*

Agar layanan TI dapat memberikan manfaat kepada pihak bisnis, layanan-layanan TI tersebut harus terlebih dahulu di desain dengan acuan tujuan bisnis dari pelanggan. *Service Design* memberikan panduan kepada organisasi TI untuk dapat secara sistematis dan *best practice* mendesain dan membangun layanan TI maupun implementasi ITSM itu sendiri. *Service Design* berisi prinsip-prinsip dan metode-metode desain untuk mengkonversi tujuan-tujuan strategis organisasi TI dan bisnis menjadi portofolio/koleksi layanan TI serta aset-aset layanan, seperti *server*, *storage*, dan sebagainya.

Ruang lingkup *Service Design* tidak melulu hanya untuk mendesain layanan TI baru, namun juga proses-proses perubahan

maupun peningkatan kualitas layanan, kontinyuitas layanan maupun kinerja dari layanan.

Proses-proses yang dicakup dalam *Service Design* yaitu:

1. *Service Catalog Management*
2. *Service Level Management*
3. *Supplier Management*
4. *Capacity Management*
5. *Availability Management*
6. *IT Service Continuity Management*
7. *Information Security Management*

3. *Service Transition*

Service Transition menyediakan panduan kepada organisasi TI untuk dapat mengembangkan serta kemampuan untuk mengubah hasil desain layanan TI baik yang baru maupun layanan TI yang diubah spesifikasinya ke dalam lingkungan operasional. Tahapan *lifecycle* ini memberikan gambaran bagaimana sebuah kebutuhan yang didefinisikan dalam *Service Strategy* kemudian dibentuk dalam *Service Design* untuk selanjutnya secara efektif direalisasikan dalam *Service Operation*.

Proses-proses yang dicakup dalam *Service Transition* yaitu:

1. *Transition Planning and Support*
2. *Change Management*
3. *Service Asset & Configuration Management*
4. *Release & Deployment Management*
5. *Service Validation*
6. *Evaluation*
7. *Knowledge Management*

4. *Service Operation*

Service Operation merupakan tahapan *lifecycle* yang mencakup semua kegiatan operasional harian pengelolaan layanan-layanan TI. Di dalamnya terdapat berbagai panduan pada bagaimana mengelola layanan TI secara efisien dan efektif serta menjamin tingkat kinerja yang telah diperjanjikan dengan pelanggan sebelumnya. Panduan-panduan ini mencakup bagaiman menjaga kestabilan operasional layanan TI serta pengelolaan perubahan desain, skala, ruang lingkup, serta target kinerja layanan TI.

Proses-proses yang dicakup dalam *Service Operation* yaitu:

1. *Event Management*
2. *Incident Management*
3. *Problem Management*
4. *Request Fulfillment*

5. Access Management

5. *Continual Service Improvement*

Continual Service Improvement (CSI) memberikan panduan penting dalam menyusun serta memelihara kualitas layanan dari proses *design, transisi* dan pengoperasiannya. CSI mengkombinasikan berbagai prinsip dan metode dari manajemen kualitas, salah satunya adalah *Plan-Do-Check-Act* (PDCA) atau yang dikenal sebagi *Deming Quality Cycle*

2.3 Definisi UML (*Unified Modeling Language*)

Unified Modelling Language (UML) adalah sebuah bahasa yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari tiga notasi yang telah ada sebelumnya: *Grady Booch* OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

2.3.1 Teknik-Teknik UML

Dalam tahap perancangan satu system diperlukan adanya teknik-teknik penyusunan *system* untuk menganalisa dan mempelajari objek yang ada unuk mengetahui apakah mereka dapat digunakan kembali atau diadaptasi untuk pemakaina baru atau menetukan satu objek baru atau yang dimodifikasi yang digabung dengan objek yang sudah ada ke dalam suatu aplikasi komputasi. Teknik-teknik tersebut adalah sebagai berikut:

1. *Use Case Diagram*

   Diagram ini menunjukan sekumpulan kasus fungsional dan actor (jenis kelas khusus) dan keterhubungannya diagram ini dapat sangat membantu bila kita sedang menyusun *requitment* sebuah *system*, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada *system*.

2. *Sequence Diagram*

Diagram ini menunjukan interaksi yang terjadi antar objek. Diagram ini merupakan pandangan dinamis terhadap *system*. Diagram ini menekankan pada basis keberurutan waktu dari pesan-pesan yang terjadi.

3. *Collaboration Diagram*

   Diagram ini merupakan diagram interaksi. Diagram ini menekankan pada organisasi stuktur dari objek-objek yang mengirim dan menerima pesan.

4. *Class Diagram*

   Diagram ini adalah seebuah spesifikasi yang jika di instansiasi akan menghasilkan sebuah objek yang merupakan inti pengembangan dan desain yang berorientasi objek. Kelas mengambarkan keadaan suatu *system,* sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut.

5. *Statechart Diagram*

   Statechart diagram menunjukkan transisi dan perubahan keadaan suatu objek pada *system* sebagai akibat dari stimulasi yang diterima. Dalam UML, state digambarkan berbentuk segi empat dengan sudut tumpul dan memiliki nama sesuai dengan kondisi saat itu.

6. *Activity Diagram*

   Actifity diagram menggambarkan berbagai alir aktifitas dalam *system* yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses parallel yang mungkin terjadi pada beberapa eksekusi.

7. *Component Diagram*

   Component Diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan di antaranya.

8. *Deployment Diagram*

   Deployment diagram menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur *system*, dimana komponen akan diletakkan (pada mesin, *server*, atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server* dan hal-hal lain yang bersifat fisikal.

2.4 Software Pendukung

2.4.1 *Personal Home Page (PHP)*

PHP atau *Personal Home Page* merupakan sebuah bahasa *scripting* yang dibundel dengan HTML, yang dijalankan di sisi *server*. Sebagain besar intinya berasal dari C, Java dan Perl dengan beberapa tambahan fungsi khusus PHP. Bahasa ini memungkinkan para pembuat aplikasi web menyajikan halaman HTML dinamis dan interaktif dengan cepat dan mudah, yang dihasilkan *server*. PHP juga dimaksudkan untuk mengganti teknologi lama seperti *Common Gateway Interface* (CGI).

PHP membuat proses pengembangan aplikasi menjadi mudah karena kelebihan-kelebihannya, yaitu :

1. *Script* (kode program) terintegrasi dengan file HTML, sehingga *developer* bisa berkonsentrasi langsung pada penampilan dokumen webnya.
2. Tidak ada proses *compiling* dan *linking*.
3. Berorientasi obyek.
4. Sintaksis pemogramannya mudah dipelajari, dan menyerupai C dan Perl.

Integrasi yang sangat luas ke berbagai *server database*. Menulis web yang terhubung ke database menjadi sangat sederhana. Pada umumnya PHP menggunakan MySQL sebagai database, namun PHP juga mendukung database yang lain seperti *Oracle*, *Sybase*, MySQL, *Solid*, ODBC, *Postgre*SQL, Adabas D, *FilePro*, *Velocis*, *Informix*, *dBase*, UNIX dbm.

2.4.2 *Framework CodeIgniter*

Framework adalah kumpulan perintah atau fungsi dasar yang membentuk aturan-aturan tertentu dan saling berinteraksi satu sama lain sehingga dalam pembuatan aplikasi website kita harus mengikuti aturan dari *framework* tersebut. Dengan *framework* (dalam hal ini framework php), kita tidak perlu memikirkan kode perintah/fungsi dasar dari aplikasi website kita. Seperti bagaimana mengambil data dari database untuk ditampilkan. Kita hanya memikirkan apa kode sql-nya dan ditampilkan kemana? Hal-hal penunjang lainnya seperti koneksi database validasi form, GUI, dan keamanan; telah disediakan oleh *framework* sehingga jumlah baris kode yang kita buat

jauh lebih sedikit dibandingkan jika semua kode dari kita. Jadi, keuntungan yang dapat diperoleh dari pengguna *framework* adalah:

1. Waktu pembuatan aplikasi website kita jauh lebih singkat.
2. Kode aplikasi website menjadi lebih mudah dibaca, karena sedikit dan sifatnya pokok. Detailnya adalah kode dari *framework* dan ini mungkin tidak perlu dipikirkan,terjamin.
3. Website kita menjadi lebih mudah diperbaiki, karena kita tidak perlu fokus ke semua komponen kode website, terutama kode *system framework*.
4. Kita tidak perlu lagi membuat kode penunjang aplikasi website seperti koneksi database, validasi form, GUI, dan keamanan.
5. Pikiran kita menjadi lebih terfokus ke kode alur permasalahan website, apa yang ditampilkan dan layanan apa saja yang diberikan dari aplikasi website tersebut.
6. Jika dikerjakan *team work*, maka akan lebih terarah karena *system framework*, mengharuskan adanya keteraturan peletakan kode. seperti bagian pengambilan database terpisah dengan bagian pengaturan tampilan untuk pengunjung.

Jadi intinya banyak keuntungan yang dapat diperoleh, terutama bagi developer web yang selalu dihantui waktu deadline proyek websitenya. Tidak selesai waktunya sesuai target, klien akan kecewa dan mungkin akan lari ke developer lain. Jika anda bersikeras membuat website dari nol kode maka akan butuh banyak waktu dan pasti akan ketinggalan dengan teman anda yang menggunakan *framework*.

Framework CI paling mudah dikuasai untuk seorang pemula, karena help-nya cukup tersedia diwebsitenya. bahkan saat kita mendownload *framework* ini dari internet, sudah disertai help-nya yang mudah dimengerti, mulai dari langkah instalasi sampai fungsi-fungsinya tersedia. Adanya help/dokumentasi sangat mudah bagi pemula dalam mempelajari lingkungan pengembangan website dengan CI.

Framework Codeigniter dapat dioperasikan dalam PHP 4.3.2 + maupun 5 sehingga kita membuat aplikasi website pada sebuah *server* yang masih belum support PHP5, ini tidak akan menjadi masalah,karena masih tetap dapat berkerja. Mvc (*model view controller*) adalah sebuah pola program yang bertujuan memisahkan logika bisnis, logika data, dan logika tampilan (*interface*), atau secara sederhana memisahkan antara proses, data dan tampilan mvc mengatur arsitektur sebuah aplikasi. umumnya apliksai yang dibangun dengan konsep MVC adalah apliksai yang dibangun dengan konsep MVC adalah aplikasi yang cukup besar, karena salah satu keuntungan dari MVC itu adalah kemudahan maintenance, dan pengembangan apliksai tersebut. CI menggunakan konsep ini anda harus memisahkan kode database ke folder model, kode proses ke folder controller dan kode tampilan ke folder view.

2.4.3 *Hyper Text Markup Language* (HTML)

HTML (*Hyper Text Markup Language*) adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web dan menampilkan

berbagai informasi di dalam sebuah *browser* Internet. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML *(Standard Generalized Markup Language)*, HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium* (W3C).

HTML berupa kode-kode tag yang menginstruksikan browser untuk menghasilkan tampilan sesuai dengan yang diinginkan. Sebuah file yang merupakan file HTML dapat dibuka dengan menggunakan *browser* web seperti *Mozilla Firefox* atau *Microsoft Internet Explorer*. HTML juga dapat dikenali oleh aplikasi pembuka email ataupun dari PDA dan program lain yang memiliki kemampuan browser.

HTML dokumen tersebut mirip dengan dokumen teks biasa, hanya dalam dokumen ini sebuah teks bisa memuat instruksi yang ditandai dengan kode atau lebih dikenal dengan TAG tertentu. Sebagai contoh jika ingin membuat teks ditampilkan menjadi tebal seperti: Tambil Tebal, maka penulisannya dilakukan dengan cara: Tampil Tebal . Tanda digunakan untuk mengaktifkan instruksi cetak tebal, diikuti oleh teks yang ingin ditebalkan, dan diakhiri dengan tanda untuk menonaktifkan cetak tebal tersebut.

Secara garis besar, terdapat beberapa jenis elemen dari HTML:

Structural. tanda yang menentukan level atau tingkatan dari sebuah teks (contoh,<h1>Golf</h1> akan memerintahkan *browser* untuk menampilkan “Golf” sebagai teks tebal besar yang menunjukkan sebagai Heading 1

Presentational. tanda yang menentukan tampilan dari sebuah teks tidak peduli dengan level dari teks tersebut (contoh, boldface akan menampilkan bold. Tanda presentational saat ini sudah mulai digantikan oleh CSS dan tidak direkomendasikan untuk mengatur tampilan teks,

Hypertext. tanda yang menunjukkan pranala ke bagian dari dokumen tersebut atau pranala ke dokumen lain (contoh, Web Saya akan menampilkan Web Saya sebagai sebuah hyperlink ke URL tertentu), Elemen widget yang membuat objek-objek lain seperti tombol (<button>), list (<li>), dan garis horizontal (<hr>). Selain markup presentational, markup yang lin tidak menentukan bagaimana tampilan dari sebuah teks. Namun untuk saat ini, penggunaan tag HTML untuk menentukan tampilan telah dianjurkan untuk mulai ditinggalkan dan sebagai gantinya digunakan *Cascading Style Sheets*.

Contoh HTML Sederhana :

```
<!DOCTYPE html>
<html>
<head>
<title>Website Saya</title>
```

```
</head>
<body>
<p>Selamat Datang di Website Saya</p>
</body>
</html>
```

2.4.4 *Structured Query Language* (SQL)

SQL (*Structured Query Language*) adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam basis data relasional. Bahasa ini secara *de facto* merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua server basis data yang ada mendukung bahasa ini untuk melakukan manajemen datanya.

2.4.4.1 Sejarah *Structured Query Language* (SQL)

Sejarah SQL dimulai dari artikel seorang peneliti dari IBM bernama EF Codd yang membahas tentang ide pembuatan basis data relasional pada bulan Juni 1970. Artikel ini juga membahas kemungkinan pembuatan bahasa standar untuk mengakses data dalam basis data tersebut. Bahasa tersebut kemudian diberi nama SEQUEL (*Structured English Query Language*).

Setelah terbitnya artikel tersebut, IBM mengadakan proyek pembuatan basis data relasional berbasis bahasa SEQUEL. Akan tetapi, karena permasalahan hukum mengenai penamaan SEQUEL, IBM pun mengubahnya menjadi SQL. Implementasi basis data relasional dikenal dengan *System/R*.

Di akhir tahun 1970-an, muncul perusahaan bernama *Oracle* yang membuat *server* basis data populer yang bernama sama dengan nama perusahaannya. Dengan naiknya kepopuleran *Oracle*, maka SQL juga ikut populer sehingga saat ini menjadi standar *de facto* bahasa dalam manajemen basis data.

2.4.4.2 Standarisasi *Structured Query Language* (SQL)

Secara umum, SQL terdiri dari dua bahasa, yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML). Implementasi DDL dan DML berbeda untuk tiap sistem manajemen basis data (SMBD), namun secara umum implementasi tiap bahasa ini memiliki bentuk standar yang ditetapkan ANSI.

1. *Data Definition Language* (DDL)

   DDL digunakan untuk mendefinisikan, mengubah, serta menghapus basis data dan objek-objek yang diperlukan dalam basis data, misalnya tabel, *view*, *user*, dan sebagainya. Secara umum, DDL yang digunakan adalah `CREATE` untuk membuat objek baru, `USE` untuk menggunakan objek, `ALTER` untuk mengubah objek yang sudah ada, dan `DROP` untuk menghapus objek. DDL biasanya digunakan oleh administrator basis data dalam pembuatan sebuah aplikasi basis data.

2. *Data Manipulation Language* (DML)

DML digunakan untuk memanipulasi data yang ada dalam suatu tabel. Perintah yang umum dilakukan adalah:

SELECT untuk menampilkan data

INSERT untuk menambahkan data baru

UPDATE untuk mengubah data yang sudah ada

DELETE untuk menghapus data

2.4.4.3 MySql

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris : *database management system*) atau DBMS yang *multithread, multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti *Apache*, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah : David Axmark, Allan Larsson dan Michael "Monty" Widenius.

MySQL memiliki beberapa keistimewaan, antara lain :

1. *Portabilitas*. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga*, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. *Multi-user*. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance tuning*, MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed / unsigned integer, float, double, char, text, date, times tamp*, dan lain-lain.
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60

ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protocol TCP/IP, Unis Socket (UNIX), atau Named Pipes (NT).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan basis data lainnya semacam *Posgre* SQL ataupun *Oracle*.

2.4.5 Cascading Style Sheet (CSS)

Cascading Style Sheet (CSS) merupakan salah satu bahasa pemrograman web untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam.

Sama halnya *styles* dalam aplikasi pengolahan kata seperti *Microsoft Word* yang dapat mengatur beberapa *style*, misalnya *heading*, *subbab*, *bodytext*, *footer*, *image*, dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (*file*). Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML.

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna hyperlink, warna *mouse over*, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. CSS adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda.

Fakta Menggunakan CSS diantaranya :

Telah didukung oleh kebanyakan *browser* versi terbaru, tetapi tidak didukung oleh *browser-browser* lama.

Lebih fleksibel dalam penempatan posisi layout. Dalam *layouting* CSS, kita mengenal Z-*Index* untuk menempatkan objek dalam posisi yang sama.

Menjaga HTML dalam penggunaan tag yang minimal, hal ini berpengaruh terhadap ukuran berkas dan kecepatan pengunduhan.

Dapat menampilkan konten utama terlebih dahulu, sementara gambar dapat ditampilkan sesudahnya.

Penerjemahan CSS setiap *browser* berbeda, tata letak akan berubah jika dilihat di berbagai *browser*

CSS adalah layouting "Masa Depan" dengan penggabungan bersama XHTML.

2.4.6 *Java Script*

2.4.6.1 Sejarah Java Script

JavaScript pertama kali diperkenalkan oleh *Netscape* pada tahun 1995. Pada awalnya bahasa yang sekarang disebut *JavaScript* ini dulunya dinamai *"LiveScript""* yang berfungsi sebagai bahasa sederhana untuk *browser Netscape Navigator* 2 yang sangat populer pada saat itu. Kemudian sejalan dengan sedang giatnya kerjasama antara *Netscape* dan Sun (pengembang bahasa pemrograman "*Java*") pada masa itu, maka *Netscape* memberikan nama "*JavaScript*" kepada bahasa tersebut pada tanggal 4 desember 1995.

Pada saat yang bersamaan Microsoft sendiri mencoba untuk mengadaptasikan teknologi ini yang mereka sebut sebagai "*Jscript*" di *browser* milik mereka yaitu Internet Explorer 3. *JavaScript* sendiri merupakan modifikasi dari bahasa pemrograman C++ dengan pola penulisan yang lebih sederhana dari bahasa pemrograman C++.

2.4.6.2 Pengertian *Java Script*

JavaScript adalah bahasa pemrograman berbasis *prototipe* yang berjalan disisi klien. Jika kita berbicara dalam konteks web, sederhananya, kita dapat

memahami *JavaScript* sebagai bahasa pemrograman yang berjalan khusus untuk di *browser* atau halaman web agar halaman web menjadi lebih hidup. Kalau dilihat dari suku katanya terdiri dari dua suku kata, yaitu *Java* dan *Script*. *Java* adalah Bahasa pemrograman berorientasi objek, sedangkan *Script* adalah serangkaian instruksi program.

Secara fungsional, *JavaScript* digunakan untuk menyediakan akses *script* pada objek yang dibenamkan (embedded). Contoh sederhana dari penggunaan *JavaScript* adalah membuka halaman pop up, fungsi validasi pada form sebelum data dikirimkan ke *server*, merubah *image cursor* ketika melewati objek tertentu, dan lain lain.

2.4.6.3 Kelebihan *Java Script*

JavaScript bekerja pada sisi *browser*. maksudnya begini : untuk menampilkan halaman web, *user* menuliskan alamat web di *address bar url*. setelah itu, *browser* "mengambil" *file* html (dengan *file JavaScript* yang melekat padanya jika memang ada) ke *server* yang beralamat di URL yang diketikan oleh *user*. Selesai file diambil, file ditampilkan pada *browser*. Nah, setelah file *JavaScript* berada pada *browser*, barulah *script JavaScript* tersebut bekerja.

Efek dari *Javascript* yang bekerja pada sisi *browser* ini, *Javascript* dapat merespon perintah *user* dengan cepat, dan membuat halaman web menjadi lebih responsif. *JavaScript* melakukan apa yang tidak bisa dilakukan oleh

HTML, PHP, dan CSS : menangani hal – hal yang membutuhkan respon cepat terhadap aksi dari user.