

BAB II

TINJAUAN PUSTAKA

2.1 Konsep Dasar Sistem Informasi

2.1.1 Pengertian Sistem

Sistem adalah entitas atau satuan yang terdiri dari dua atau lebih komponen atau subsistem (sistem yang lebih kecil) yang saling terhubung dan terkait untuk mencapai satu tujuan (Rudyanto, 2012:1).

2.1.2 Pengertian Informasi

Informasi merupakan data yang telah diolah menurut konteks tertentu dan informasi juga merupakan hasil pemahaman individu terhadap relas-relasi antara fakta/data (Wasistiono, 2010:86). Secara etimologis, informasi adalah bahasa serapan dari bahasa asing, inggris. Jika mengacu pada *Oxford English Dictionary*, pemaknaan informasi dalam bahasa inggris adalah “*the act of informing, or giving form or shape to the mind, as in education, instruction, or trining.*” Kata “*inform*” itu sendiri sebenarnya dari kata dalam bahasa Latin, “*informare*” yang artinya memberikan bentuk atau membentuk suatu ide.

Informasi dapat dipahami sebagai pemrosesan input yang terorganisir, memiliki arti, dan berguna bagi orang yang menerimanya (Rudyanto, 2012:1).

2.1.3 Pengertian Sistem Informasi

Sistem informasi merupakan jaringan yang terdapat dalam suatu organisasi yang membuat para pimpinan bisa mengakses informasi secara terus menerus (Wasistiono 2013:3). Dengan informasi tersebut, para pimpinan bisa membuat putusan yang ditujukan untuk meningkatkan kinerja organisasi. Karena pada umumnya sistem informasi ditandai dengan penggunaan komputer, para pimpinan bisa menggunakan sistem informasi untuk mengakses catatan-catatan Lembaga secara *on line* dan membuat ringkasan informasi yang diperoleh serta membuat laporan-laporan.

2.2 Konsep dasar Evaluasi Kinerja

2.2.1 Pengertian Kinerja

Kinerja adalah penampilan yang melakukan, menggambarkan dan menghasilkan sesuatu hal sesuatu hal, baik yang bersifat fisik dan *non-fisik* yang sesuai dengan petunjuk, fungsi dan tugasnya yang didasari oleh pengetahuan, sikan dan keterampilan (Rahadi, 2011:4).

2.2.2 Pengertian Sistem Penilaian Prestasi Kerja

Proses untuk mengukur prestasi kerja karyawan berdasarkan peraturan yang telah ditetapkan, dengan cara membandingkan sasaran (hasil kerjanya) dengan persyaratan deskripsi pekerjaan yaitu standard pekerjaan yang telah ditetapkan selama periode tertentu. *Standard* kerja tersebut dapat dibuat baik secara kualitatif maupun kuantitatif (Rahadi, 2011:3).

2.2.3 Evaluasi Kinerja

Evaluasi kinerja merupakan sarana untuk memperbaiki mereka yang tidak melakukan tugasnya dengan baik di dalam organisasi. Banyak organisasi berusaha mencapai sasaran untuk kedudukan yang terbaik dan terpercaya dalam bidangnya. Untuk itu sangat tergantung dari para pelaksanaanya, yaitu para karyawannya agar mereka mencapai sasaran yang telah ditetapkan oleh organisasi dalam *corporate planning*-nya (Rahadi, 2010:94).

Secara singkat dapat disimpulkan bahwa prinsip dasar evaluasi kinerja sebagai berikut:

- a. Fokusnya adalah membina kekuatan untuk menyelesaikan setiap persoalan yang timbul dalam pelaksanaan evaluasi kinerja. Jadi bukan semata-mata menyelesaikan persoalan itu sendiri, namun pimpinan dan karyawan mampu menyelesaikan persoalannya dengan baik setiap saat, setiap ada persoalan baru. Jadi yang penting adalah kemampuannya.
- b. Selalu didasarkan atas suatu pertemuan pendapat, misalnya dari hasil diskusi antara karyawan dnegan penyedia langsung, suatu diskusi yang konstruktif untuk mencari jalan yang terbaik dalam meningkatkan mutu dan baku yang tinggi.

- c. Suatu proses manajemen yang alami, jangan merasa dan menimbulkan kesan terpaksa, namun dimasukkan secara sadar ke dalam *corporate planning*, dilakukan secara *periodic*, terarah dan terprogram, bukan kegiatan yang hanya setahun sekali atau kegiatan yang dilakukan jika manajer ingat saja.

2.2.4 Standard Kinerja

Standard kinerja adalah tolak ukur minimal kinerja yang harus dicapai karyawan secara individual atau kelompok pada semua indikator kinerjanya (Rahadi, 2010:20). Dalam definisi ini, standard kinerja adalah tolak ukur minimal, kinerja karyawan yang umum untuk kebanyakan pekerjaan meliputi elemen sebagai berikut:

- a. Kuantitas dari hasil,
- b. Kualitas dari hasil,
- c. Ketepatan waktu dari hasil,
- d. Kehadiran, dan
- e. Kemampuan bekerja sama.

2.2.5 Kriteria Capaian Kinerja Kemenag

Capaian kinerja Kementerian Agama (Kemenag) memiliki tiga kriteria, diantaranya:

- a. Absensi pegawai
Kemenag mewajibkan pegawainya untuk mengerjakan tugas kantornya selama 7,5 jam setiap harinya khususnya pada hari Senin – Jumat dan tidak terhitung untuk hari libur nasional.
- b. Catatan kegiatan pegawai
Terget capaian tiap pegawai tidak bisa ditentukan karena banyaknya tugas yang dijadikan tugas tambahan. Seperti menjadi MC, pembicara, dokumenter, dinas luar kota dll. Semakin tinggi jabatan seseorang semakin tinggi pula target kegiatan yang harus dicapai ditiap bulannya. Ditiap bulannya akan diambil evaluasi oleh atasan kepada bawahannya

untuk melihat pegawai yang sudah mencapai persentase 100 dan pegawai yang belum mencapai persentase 100 lalu di ambil keputusan.

c. Keuangan pegawai

Potongan-potongan yang diberikan pada pegawai karena kelalaiannya dijadikan penilaian oleh atasan.

2.3 Pengertian Monitoring

Monitoring (pengawasan) adalah suatu proses untuk menerapkan pekerjaan apa yang sudah dilaksanakan, menilainya, dan bila perlu mengoreksi dengan maksud agar pelaksanaan pekerjaan sesuai dengan rencana semula (Manullang, 2012:173).

2.4 Metode Analisis Masalah

Untuk mengidentifikasi masalah, maka harus dilakukan analisis terhadap kinerja, informasi, ekonomi, pengendalian, efisiensi, dan pelayanan. Panduan ini dikenal dengan analisis PIECES (*Performance, Information, Economic, Control, Efficiency, Service*) (Sutabri, 2012). Analisis dilakukan pada sistem informasi lama. Dari analisis ini biasanya didapatkan beberapa masalah dan akhirnya dapat ditemukan masalah utamanya.

Untuk lebih jelasnya lagi mengenai PIECES, di bawah ini akan dijelaskan mengenai pengertian dari masing-masing komponen PIECES.

a. Analisis Kinerja Sistem (*Performance*)

Kinerja adalah suatu kemampuan sistem dalam menyelesaikan tugas dengan cepat sehingga sasaran dapat segera tercapai. Kinerja diukur dengan jumlah produksi (*throughput*) dan waktu yang digunakan untuk menyesuaikan perpindahan pekerjaan (*response time*).

b. Analisis Informasi (*Information*)

Informasi merupakan hal penting karena dengan informasi tersebut pihak manajemen (*marketing*) dan *user* dapat melakukan langkah selanjutnya. Apabila kemampuan sistem informasi baik, maka user akan mendapatkan informasi yang akurat, tepat waktu dan relevan sesuai dengan yang diharapkan.

c. Analisis Ekonomi (*Economy*)

Pemanfaatan biaya yang digunakan dari pemanfaatan informasi. Peningkatan terhadap kebutuhan ekonomis mempengaruhi pengendalian biaya dan peningkatan manfaat. Saat ini banyak perusahaan dan manajemen mulai menerapkan paperless sistem (meminimalkan penggunaan kertas) dalam rangka penghematan. Oleh karena itu dilihat dari penggunaan bahan kertas yang berlebihan dan biaya iklan di media cetak untuk media publikasi, sistem ini dinilai kurang ekonomis.

d. Analisis Pengendalian (*Control*)

Analisis ini digunakan untuk membandingkan sistem yang dianalisis berdasarkan pada segi ketepatan waktu, kemudahan akses, dan ketelitian data yang diproses.

e. Analisis Efisiensi (*Efficiency*)

Efisiensi berhubungan dengan bagaimana sumber tersebut dapat digunakan secara optimal. Operasi pada suatu perusahaan dikatakan efisien atau tidak biasanya didasarkan pada tugas dan tanggung jawab dalam melaksanakan kegiatan.

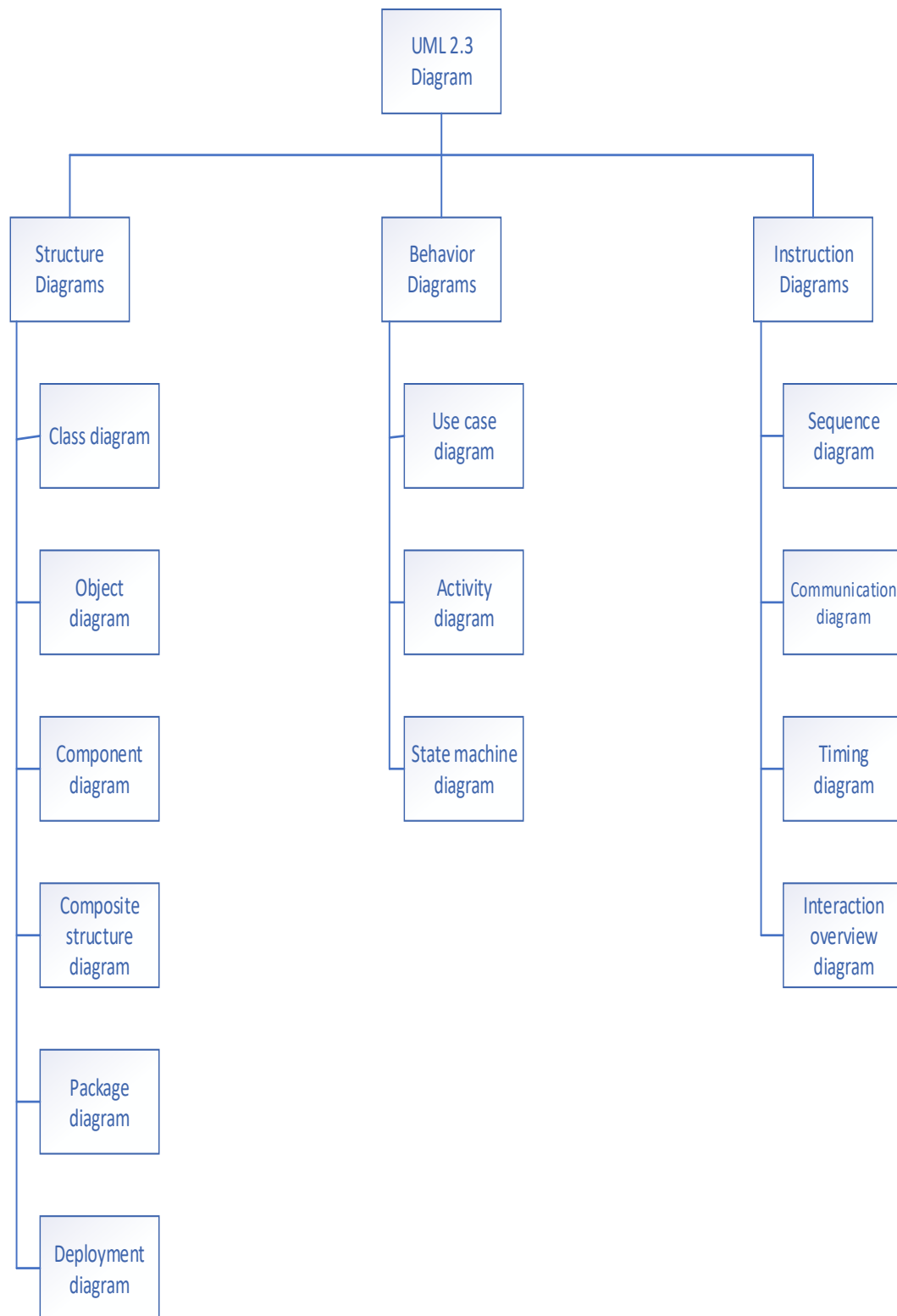
f. Analisis Pelayanan (*Service*)

Peningkatan pelayanan memperlihatkan kategori yang beragam. Proyek yang dipilih merupakan peningkatan pelayanan yang lebih baik bagi manajemen (*marketing*), user dan bagian lain yang merupakan simbol kualitas dari suatu sistem informasi.

2.5 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa & Shalahuddin, 2014:133).

Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar di bawah ini.



Gambar 2.1 Diagram UML (Rosa & Shalahuddin, 2014:134)

2.5.1 *Use Case Diagram*

Use Case atau diagram *use case* merupakan permodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang berhak menggunakan fungsi-fungsi itu (Rosa & Shalahuddin 2014:155).

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

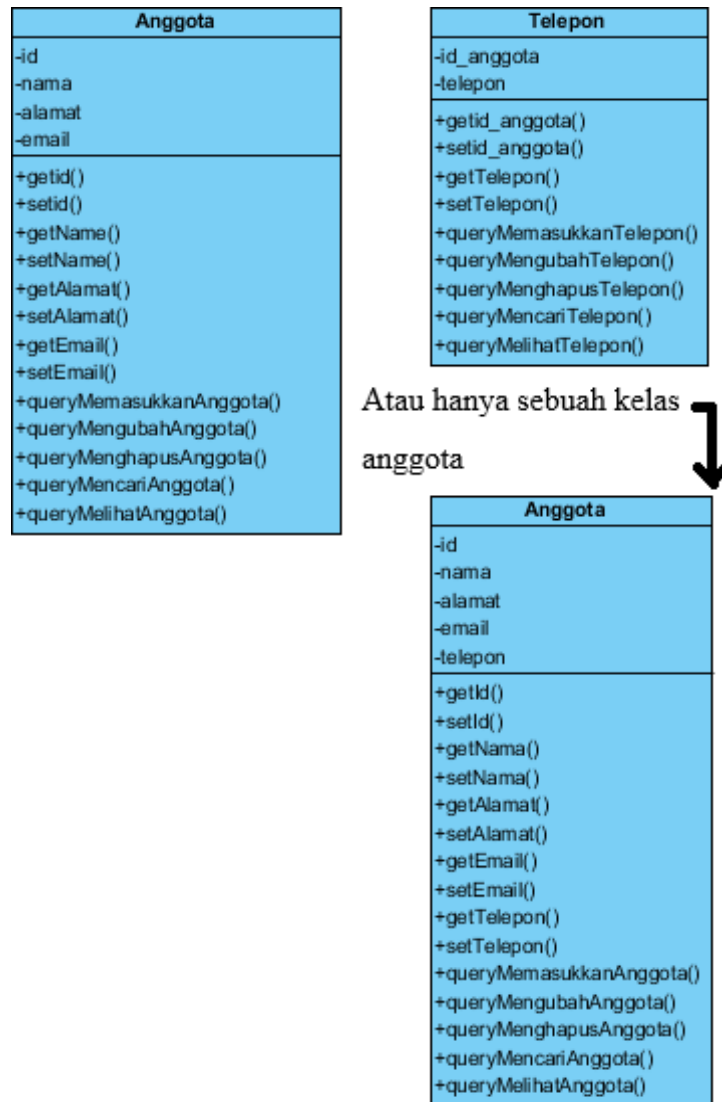
- a. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

2.5.2 *Class Diagram*

Class Diagram atau diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi (Rosa & Shalahuddin, 2014:141).

- a. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
- b. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.
- c. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:
 - 1) Kelas main.

- 2) Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
- 3) Kelas yang menangani tampilan sistem (*view*)
- 4) Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.
- 5) Kelas yang diambil dari pendefinisian *use case* (*controller*)
- 6) Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.
- 7) Kelas yang diambil dari pendefinisian data (*model*)
- 8) Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data. Semua tabel yang dibuat di basis data dapat dijadikan kelas, namun untuk tabel dari hasil relasi atau atribut multivalued pada ERD dapat dijadikan kelas tersendiri dapat juga tidak asalkan pengaksesannya dapat dipertanggungjawabkan atau tetap ada di dalam perancangan kelas. Misalkan dalam tabel TTelepon dan TAnggota pada studi kasus maka perancangan kelas dapat mengandung kelas Telepon dan Anggota di mana didalamnya ada sebuah atribut berupa larik (*array*) bertipe *string* dengan nama telepon. Ilustrasinya dapat dilihat pada gambar berikut:



Gambar 2.2 Perancangan kelas data untuk tabel dari atribut multivalue

2.5.3 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa & Shalahuddin 2014:161).

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

- a. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- b. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
- d. Rancangan *menu* yang ditampilkan pada perangkat lunak.

2.5.4 Sequence Diagram

Diagram sekuen menggambarkan kelakuan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirim dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat scenario yang ada pada *use case* (Rosa & Shalahuddin, 2014:165).

2.5.5 Package Diagram

Bersifat statis. Diagram ini memperlibatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.

2.5.6 Communication Diagram

Bersifat dinamis. Diagram sebagai pengganti diagram kalaborasi UML yang melakukan organisasi structural dari objek-objek yang menerima serta mengirim pesan.

2.5.7 *Statechart Diagram*

Bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*state*), *transisi*, kejadian serta aktivitas.

2.5.8 *Component Diagram*

Bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan *sistem* perangkat lunak pada komponen-komponen yang telah ada sebelumnya.

2.5.9 *Deployment Diagram*

Bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*). Memuat simpul-simpul beserta komponen-komponen yang di dalamnya. Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai kebutuhan. Pada UML dimungkinkan kita menggunakan diagram-diagram lainnya misalnya data *flow diagram*, *entity relationship diagram* dan sebagainya

2.6 Metodologi Pengembangan Sistem

Metodologi memberikan kerangka dasar dan seperangkat prosedur acuan yang bisa dilakukan dalam rangka mengembangkan sistem informasi (Wasistiono, 2013:17). Dalam pengembangan sistem informasi pemerintahan, metodologi biasanya:

- a. Menggunakan pengalaman para ahli dan pengembangan sistem sebagai referensi dan memberikan para manajer yang baru terlibat dalam proses pengembangan dengan beberapa tahapan / langkah yang harus dilakukan.
- b. Menyediakan catatan historis proses pengembangan yang akan berguna untuk perencanaan pada masa yang akan datang dan untuk evaluasi sistem informasi.
- c. Membuat para manajer lebih baik dalam mengawasi *progress* upaya pengembangan dan meningkatkan penggunaan hasil.

- d. Mengizinkan transfer desain dari suatu aplikasi ke aplikasi yang lain, dan mentransfer satu personel dari satu proyek ke proyek lain.

2.7 Standardisasi Sistem Informasi Perintahan

Standardisasi penting karena mampu menghemat sumber daya, mempercepat pengembangan sistem informasi pemerintahan, dan membantu pemerintah, dan membantu pemerintah dalam mengikuti trend kemajuan teknologi informasi. Standar sistem informasi biasanya terdiri dari standar data, standar teknis, standar metodologi, dan standar keamanan (Wasistiono 2013: 19).

- a. Standar data sangat membantu dalam mengurangi duplikasi dan ketidaksesuaian ketika mengumpulkan, memproses, menyebarkan data, dan menaikkan pertukaran informasi. Klasifikasi data merupakan salah satu isu penting lagi dalam standar data.
- b. Standar teknis meliputi hardware, software, dan telekomunikasi. Sebagai dasar utama dari pengembangan sistem informasi, seperangkat standar yang konsisten untuk deksripsi arsitektur teknis sistem informasi (jaringan, *mainframe*, workstation, dll).
- c. Standar metodologi meliputi pemilihan seperangkat metodologi yang meliputi semua langkah perencanaan informasi, desain, implementasi, dan evaluasi. Misalnya:
 - 1) Metode perencanaan sistem informasi;
 - 2) Metode desain sistem informasi;
 - 3) Metode rekayasa software;
 - 4) Jaminan mutu dan mutu pengujian;
 - 5) Metode keamanan dan pemeliharaan; dan
 - 6) Metode penilaian kinerja;
- d. Standar keamanan. Kebijakan pemerintah bisa dikeluarkan dalam rangka menetapkan dan memperbaiki tanggung jawab kemanan dalam pemerintahan.

2.8 *Rapid Application Development (RAD)*

Metode *Rapid Application Development* (RAD) adalah suatu pendekatan berorientasi terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat – perangkat lunak (Kendall & Kendall, 2010:163).

Ada tiga fase dalam RAD yang melibatkan penganalisis dan pengguna dalam tahap penilaian, perancangan, dan penerapan. RAD melibatkan pengguna dalam setiap bagian upaya pengembangan dengan partisipasi mendalam dalam bagian perancangan bisnis. Berikut ini merupakan tiga tahap dari fase RAD (Kendall & Kendall, 2010:164) :

a. *Requirements Planning Phase*

Dalam fase ini, pengguna dan penganalisis bertemu untuk mengidentifikasi tujuan – tujuan aplikasi atau sistem serta untuk mengidentifikasi syarat – syarat informasi yang ditimbulkan dari tujuan – tujuan tersebut. Fase ini memerlukan peran aktif mendalam dari kedua kelompok tersebut, tidak hanya menunjukkan proposal atau dokumen. Selain itu, juga melibatkan pengguna dari beberapa tingkat yang berbeda dalam organisasi. Orientasi dalam fase ini ialah menyelesaikan masalah – masalah perusahaan (Kendall & Kendall, 2010:164). Pembahasan dalam perencanaan syarat meliputi :

- 1) Gambaran umum perusahaan, menjelaskan tentang profil perusahaan
- 2) Visi dan misi, menjelaskan mengenai tujuan dan cita – cita yang ingin dicapai.
- 3) Struktur organisasi, menjelaskan mengenai susunan dan hubungan antar posisi yang ada pada perusahaan dan menjelaskan pemisahan kegiatan pekerjaan antara yang satu dengan yang lain.
- 4) Analisis sistem berjalan (*As-Is*), menjelaskan mengenai proses bisnis pada sistem yang saat ini berlangsung/sistem berjalan.
- 5) Identifikasi masalah, menjelaskan permasalahan yang ada dalam proses bisnis perusahaan.
- 6) Analisis sistem yang diusulkan (*To-Be*), menjelaskan mengenai sistem yang diusulkan oleh peneliti terhadap perusahaan.

b. *RAD Design Workshop*

Fase ini adalah fase untuk merancang dan memperbaiki yang bias digambarkan sebagai *workshop*. Selama *RAD Design workshop*, pengguna merespons working prototype yang ada dan menganalisis memperbaiki modul – modul yang dirancang berdasarkan respons pengguna (Kendall & Kendall, 2010:164). Pada fase ini menggunakan beberapa tahap perancangan, yaitu :

1) Perancangan Proses

Tahap ini menggambarkan usulan proses dengan menggunakan tools *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*.

2) Perancangan Basis Data

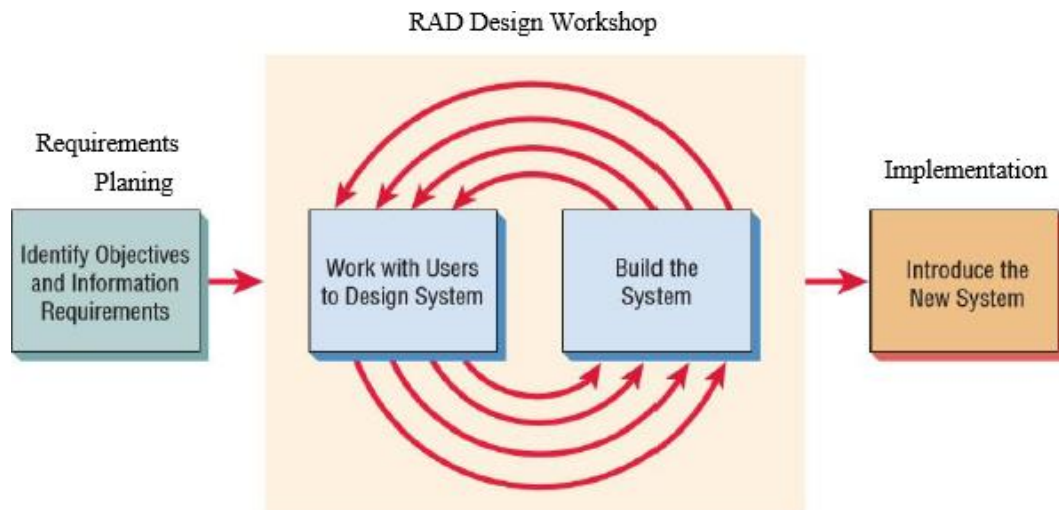
Pada tahap ini akan dirancang kebutuhan basis data dengan menggunakan tools *Class Diagram*.

3) Perancangan Antarmuka

Tahap ini akan dirancang sketsa tampilan untuk *user*.

c. *Implementation Phase*

Analisis sistem bekerja dengan para pengguna secara intens selama *RAD Design Workshop* untuk merancang aspek – aspek bisnis dan nonteknis dari perusahaan. Segera sesudah aspek – aspek ini disetujui dan sistem – sistem dibangun dan disarung, sistem – sistem baru atau bagian dari sistem di ujicoba dan kemudian diperkenalkan kepada organisasi (Kendall & Kendall, 2010:164). Pada tahap ini akan membahas mengenai pemrograman dan pengujian menggunakan *black-box testing*.



Gambar 2.3 *Rapid Application Development Phases* (Kendall & Kendall, 2010:164)

2.9 Konsep Database

2.9.1 Pengertian *Database*

Database adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat (Rosa & Shalahuddin, 2014:133).

2.9.2 *Database Management System (DBMS)*

DBMS (*Database Management System*) atau dalam bahasa Indonesia sering disebut sebagai Sistem Manajemen Basis Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data (Rosa & Shalahuddin, 2014:133). Suatu sistem aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut:

- a. Menyediakan fasilitas untuk mengelola akses data.
- b. Mampu menangani integritas data.
- c. Mampu menangani akses data yang dilakukan secara terdistribusi.
- d. Mampu menangani *backup* data.

2.9.3 *Relational Database Management System (RDMS)*

Relational Database Management System (RDMS) adalah basis data yang berdasarkan koleksi beberapa *table* yang mana setiap *table* memiliki sebuah *primary-key*, yaitu kolom yang memiliki nilai untuk setiap *record*-nya. *Table* tersebut saling terkait satu dengan lainnya dengan menempatkan *primary-key* dari suatu *table* ke *table* lainnya sebagai *foreign-key* (Dennis, Wixom & Tegarden. 2009:330).

2.9.4 *Mapping Problem Domain Objects ke RDMS*

Cara memetakan *problem domain object* dalam *class diagram* ke format RDBMS ada delapan tahapan, yaitu (Dennis, Wixom, & Tegarden, 2009:343) :

- a. Petakan semua *concrete problem* dalam *domain class* ke tabel RDBMS. Jika, sebuah *abstract problem* dalam *domain class* memiliki beberapa *subclass* langsung, petakan juga *abstract class* ke dalam tabel RDBMS.
- b. Petakan atribut bernilai tunggal ke dalam kolom tabel.
- c. Petakan metode ke *stored procedure* atau *program modules*.
- d. Petakan agregasi yang bernilai tunggal dan hubungan asosiasi dengan sebuah kolom yang dapat menyimpan *primary key* dari tabel terkait, yaitu menambahkan *foreign-key* ke dalam tabel. Lakukan hal tersebut untuk kedua sisi yang saling berhubungan.
- e. Petakan atribut-atribut *multi-valued* dan *group* yang berulang ke tabel yang baru dan buat hubungan *one-to-many* dari tabel yang asli ke tabel yang baru.
- f. Petakan *multi-valued* agregasi dan hubungan asosiasi ke tabel asosiatif baru yang menghubungkan dua tabel asli bersama-sama. Masukkan *primary-key* dari kedua tabel tersebut ke dalam tabel asosiatif yang baru, yaitu menambahkan *foreign-key* ke tabel asosiatif yang baru.
- g. Untuk hubungan agregasi dan asosiasi dengan tipe campuran, salin *primary-key* dari sisi *single-valued* (1..1 atau 0..1) ke kolom baru pada tabel yang memiliki hubungan *multi-valued* (1..* atau 0..*) yang dapat menyimpan *primary-key* dari tabel yang terkait. Yaitu menambahkan *foreign-key* ke tabel *multi-valued*.

- h. 1) Untuk hubungan generalisasi, pastikan bahwa *primary-key* dari *subclass* sama dengan *primary-key* dari *superclass*. *Multiplicity* dari hubungan asosiasi yang baru dari *subclass* ke *superclass* harus 1..1. Jika *superclass* konkret, yaitu mereka dapat menginisiasi sendiri, maka *multiplicity* dari *superclass* ke *subclass* adalah 0..*. Jika tidak konkret hubungannya adalah 1..1. Selanjutnya, *Exclusive-or* (XOR) harus ditambahkan ke dalam asosiasi. Lakukan langkah ini untuk setiap *superclass*.
- 2) Atau sesuaikan inheritance dengan menyalin atribut *superclass* ke semua *subclass* dan hapus *superclass* dari desain.

2.10 Perangkat Lunak Perancangan

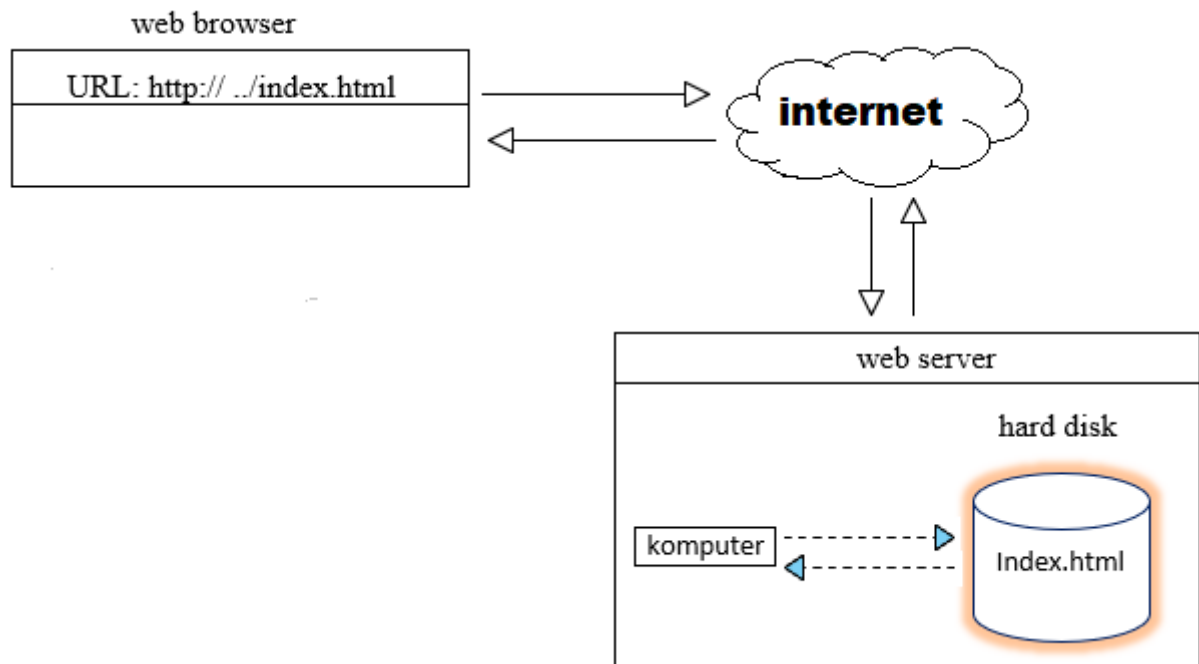
2.10.1 Web dan Konfigurasi

World Wide Web (WWW), sering disingkat dengan *web*, adalah suatu layanan di dalam jaringan internet yang berupa ruang informasi (Raharjo, 2011:2). Dengan adanya *web*, *user* dapat memperoleh atau menemukan informasi yang diinginkan dengan cara mengikuti *link* (*hyperlink*) yang disediakan didalam dokumen yang ditampilkan oleh aplikasi *web browser*.

Banyak keuntungan yang dapat diperoleh dari layanan *web*, diantaranya:

- a. Informasi mudah didistribusikan dan dapat diakses oleh semua pengguna internet di seluruh dunia
- b. Konfigurasi server dapat dilakukan secara lebih mudah
- c. Instalasi (*upload*) aplikasi hanya dilakukan sekali, tanpa harus melakukan instalasi aplikasi di setiap komputer *user* yang ingin mengakses aplikasi tersebut
- d. Tidak tergantung pada *platform*, artinya informasi maupun aplikasi dapat diakses dari komputer yang memiliki sistem operasi berbeda.

Gambar dibawah ini akan menunjukkan cara kerja dari WWW secara *global*:

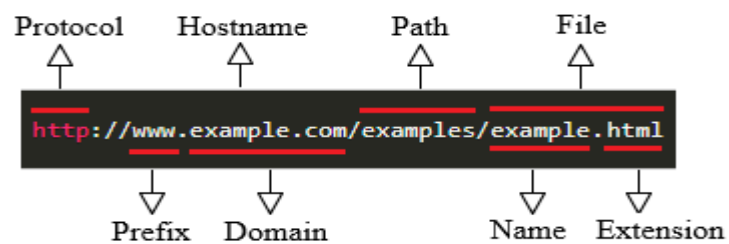


Gambar 2.4 alur web bekerja

Berikut ini penjelasan dari gambar di atas:

- User (web client)* melakukan permintaan (*request*) informasi ke *web server* dengan cara menuliskan URL, atau alamat di dalam aplikasi *web browser*. Contoh Url, adalah:

<http://www.example.com/examples/example.html>



Gambar 2.5 URL

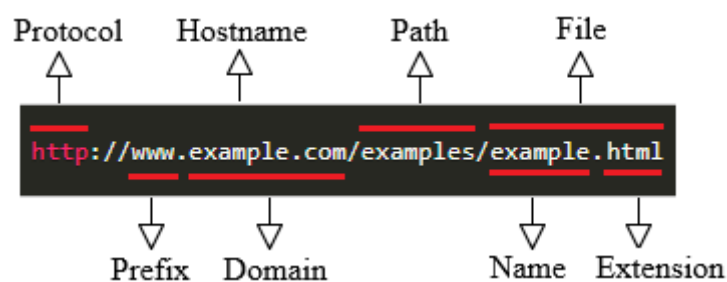
- Melalui jaringan internet, permintaan tersebut akan diarahkan ke *web server* yang sesuai dengan URL atau alamat yang dimaksud.
- Web server* akan mencari direktori dan *file* yang sesuai di dalam *hard disk*.

- d. Melalui jaringan internet, *web server* akan mengirimkan *file* tersebut ke *web client*.
- e. Dalam *web client*, *file* tersebut (yang masih berupa kode HTML) akan diterjemahkan oleh aplikasi *web browser* menjadi suatu dokumen yang dapat disajikan ke hadapan *user*.

2.10.2 Aplikasi Web Browser dan URL

Aplikasi *web browser* adalah *software* yang berfungsi untuk menampilkan dan melakukan interaksi dengan dokumen-dokumen yang tersimpan dalam suatu *web server* (Raharjo 2011: 7). Untuk mengakses informasi dari suatu situs *web*, kita perlu melakukan instalasi aplikasi *web browser* pada komputer-komputer klien. Contoh aplikasi *web browser* yang beredar saat ini adalah Internet Explorer, Mozilla Firefox, dan lain-lain.

URL adalah singkatan dari *Uniform Resource Locator*, yaitu rangkaian karakter yang disusun berdasarkan aturan/standar tertentu, yang digunakan untuk menunjukkan alamat suatu sumber (misal: dokumen atau gambar) di internet. Untuk mengakses informasi yang terdapat pada suatu situs *web*, kita perlu menuliskan URL dari situs web bersangkutan melalui aplikasi *web browser*. Aturan penulisan URL tersusun atas *protocol*, *hostname*, *path*, dan *file*, seperti yang ditunjukkan oleh contoh berikut:



Gambar 2.6 URL

2.10.3 Bootstrap

Bootstrap merupakan sebuah *frontend framework* yang lebih cepat dan mudah dalam pembangunan *web* dan sebagai standar baru untuk filosofi *mobile-first*. Ia menggunakan HTML, CSS dan Javascript (Bhaumik, 2015).

Bootstrap merupakan produk *open-source* yang dibangun oleh Mark Otto dan Jacob Thornton dimana ketika itu mereka adalah pegawai Twitter. Disana terdapat kebutuhan untuk membentuk standarisasi alat *frontend* untuk keseluruhan perusahaan (Spurlock, 2013:1).

2.10.4 Internet

Internet atau *interconnection network* merupakan sebuah sistem komunikasi *global* yang menghubungkan jaringan antarkomputer di seluruh dunia yang berkomunikasi satu sama lain (Wasistiono, 2013: 113). Internet merupakan jaringan di jaringan komputer. Internet merupakan koleksi untuk dari jaringan di seluruh dunia. Komputer pada internet menggunakan standar-standar komunikasi yang kompatibel dan berbagi kemampuan untuk berhubungan antara satu dengan lainnya serta kemampuan untuk berhubungan antara bertukar data.

2.10.5 XAMPP

Sekarang ini banyak paket *software* instalasi *web* server yang disediakan secara gratis diantaranya menggunakan XAMPP (Madcoms, 2011: 341). Dengan menggunakan paket *software* instalasi ini, maka sudah dapat melakukan beberapa instalasi *software* pendukung webserver, yaitu Apache, PHP, phpMyAdmin, dan database MySQL.

2.10.6 Framework Laravel 5.4

Laravel adalah salah satu dari beberapa kerangka bahasa pemrograman PHP yang menawarkan *code modular*. Hal ini dicapai melalui kombinasi *driver* dan sistem *bundle*-nya. *Driver* memungkinkan kita untuk dengan mudah mengubah dan memperluas *caching*, *session*, *database*, dan fungsi otentikasi. Penggunaan *bundle* mampu mengemas hingga segala jenis kode untuk digunakan kembali atau untuk memberikan kepada seluruh pengguna Laravel. Laravel sangat menarik, karena

apapun yang ditulis dalam laravel dapat dikemas dalam sebuah kemasan (McCool, 2012:3).

2.10.7 Visual Studio Code

Visual Studio Code (VS Code) adalah sebuah editor kode *open-source* untuk pengembangan dan *modern cloud debugging* dan pengembangan aplikasi. VS Code didukung dengan lebih dari 30 bahasa pemrograman, *markup* dan basis data. Beberapa adalah, Javascript, c#, c++, PHP, Java, HTML, R, CSS, SQL, *Markdown*, TypeScript, less, Sass, JSON, XML, dan Phyton (Kahlert & Giza, 2016:6).

2.11 Pengujian Perangkat Lunak

2.11.1 Black-Box Testing

Black-box testing (pengujian kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan (Rosa & Shalahuddin 2014:276).

2.11.2 White-Box Testing

White-box testing (pengujian kotak putih) yaitu menguji perangkat lunak dari segi desain dan *kode* program apakah mampu *menghasilkan* fungsi-fungsi, masukkan, dan keluaran yang sesuai dengan spesifikasi kebutuhan. Pengujian kotak putih dilakukan dengan memeriksa logik dari kode program (Rosa & Shalahuddin, 2014:276).

2.12 Penelitian Terkait

Penelitian terkait merupakan hal yang sangat berperan besar dalam perancangan yang akan dibuat, karena dengan adanya penelitian terkait dapat dijadikan tolak ukur bagi penulis untuk melakukan perancangan sistem. Maka dari itu, penelitian terkait penting untuk dilakukan agar dapat menjadi acuan yang dapat membangun penulis untuk menyelesaikan sistem ini. Berikut ini merupakan jenis penelitian yang terkait dengan sistem informasi registrasi dan uji fungsi.

Tabel 2.1 Penelitian Terkait

No	Judul Penelitian	Nama Peneliti dan Tahun	Masalah	Metode	Hasil
1.	Sistem Informasi Monitoring dan Evaluasi Konseling untuk Siswa SAM/Sederajat dalam menentukan Jurusan di Perguruan Tinggi	Dimas Prakoso (2017)	Dibutuhkannya sistem pemilihan jurusan di perguruan tinggi berdasarkan nilai rapat dan formulir angket minat jurusan	Waterfall	Sistem pemilihan jurusan yang sesuai dengan potensi siswanya untuk memilih jurusan di perguruan tinggi
2.	Perancangan Sistem Informasi Monitoring Pelaksanaan Proyek <i>Mechanical & Electrical</i> Berbasis <i>Cloud</i> (Studi Kasus di PT. MALMASS MITRA TEKNIK	Bernand Dayamuntari Hermawan (2017)	Monitoring proyek dan pengecekan hasil gambar dilakukan secara manual. Atasan kesulitan mendapatkan perkembangan proyek dan dalam pengambilan laporan secara bulanan.	Waterfall	Sistem monitoring berbasis cloud untuk manajemen kemajuan proyek.
3.	Perancangan Sistem Monitoring Penyelenggaraan Jasa Training	Lulu Rachmawati (2017)	Belum maksimalnya dalam pengelolaan data pelatihan, data peserta, data invoice absensi kehadiran, sertifikat hingga laporan.	RAD	Sistem yang memudahkan penyelenggaraan <i>training</i> agar lebih efektif dan efisien.

	pada PT. SJH SEJAHTERA				
--	---------------------------	--	--	--	--

Untuk penelitian ini penulis menggunakan metode pengembangan sistem RAD (*Rapid Application Development*). RAD dipilih karena modul yang digunakan terlalu banyak serta pengembangan aplikasi yang berfokus pada waktu penyelesaian sehingga untuk fleksibilitas dalam pengembangan sistem dapat dikendalikan serta jika ada perubahan pada setiap modul, maka pengembangan secara fleksibel dapat merubah modul tersebut dan modul yang berkaitan tanpa harus mengulang dari tahapan awal alur penelitian.

Adapun evaluasi monitoring yang dilakukan meliputi jumlah kegiatan yang sudah di kerjakan pegawai baik atasan maupun bawahan apakah sudah melewati target atau belum serta jumlah kegiatan baik dari proses pengeditan maupun penghapusan yang sudah dilakukan evaluasi akan dicatat sebagai kelalaian pegawai. Ketika pegawai sudah melakukan kelalaian akan di berikan notifikasi peringatan.