

BAB II

LANDASAN TEORI

Dalam merancang dan membangun suatu sistem informasi, sangatlah penting untuk mengetahui terlebih dahulu dasar-dasar teori yang digunakan. Dasar-dasar teori tersebut digunakan sebagai landasan berpikir dalam melakukan penelitian ini serta untuk pembahasan lebih lanjut sehingga terbentuk suatu aplikasi sesuai dengan tujuan penelitian yang diharapkan.

Adapun landasan teori yang digunakan untuk membuat Rancang Bangun Aplikasi Penilaian Kinerja Karyawan ini antara lain sebagai berikut.

2.1 Penilaian Kinerja

Menurut Mathis dan Jackson (2006:382), penilaian kinerja (*performance appraisal*) adalah proses mengevaluasi karyawan seberapa baik menyelesaikan pekerjaan mereka jika dibandingkan dengan seperangkat standar yang dimiliki perusahaan dan kemudian mengkomunikasikan informasi tersebut kepada karyawan. Dalam penilaian kinerja, dinilai kontribusi karyawan kepada perusahaan selama periode waktu tertentu. Penilaian kinerja harus memberikan umpan balik kinerja (*feedback*) kepada karyawan agar mengetahui seberapa baik mereka bekerja jika dibandingkan dengan standar-standar perusahaan. Apabila penilaian kinerja dilakukan secara benar, para manajer, staf, dan akhirnya perusahaan akan diuntungkan dengan pemastian bahwa upaya-upaya individu memberikan kontribusi kepada fokus strategi perusahaan.

2.1.1 Kinerja

Penilaian kinerja karyawan adalah masalah penting bagi seluruh perusahaan. Untuk mendapatkan kinerja yang memuaskan tidak terjadi secara otomatis. Kualitas kinerja akan diketahui dengan menggunakan sistem penilaian dari manajemen yang baik.

Menurut Mathis dan Jackson (2006:378), kinerja pada dasarnya adalah apa yang dilakukan atau tidak dilakukan karyawan. Kinerja karyawan mempengaruhi seberapa banyak mereka memberikan kontribusi kepada perusahaan. Kontribusi tersebut antara lain:

1. Kuantitas *output*.
2. Kualitas *output*.
3. Jangka waktu *output*.
4. Kehadiran di tempat kerja.
5. Sikap kooperatif.

Kontribusi yang sudah diberikan karyawan kepada perusahaan membantu manajer dalam melakukan analisis terhadap kualitas kinerja karyawan. Kontribusi-kontribusi tersebut bisa menjadi bahan pertimbangan dalam melakukan penilaian kinerja karyawan. Berdasarkan kontribusi yang dihasilkan, pihak manajer bisa menentukan kriteria-kriteria apa saja yang akan digunakan dalam melakukan penilaian kinerja karyawan.

2.1.2 Kriteria Pekerjaan

Menurut Mathis dan Jackson (2006:378), kriteria pekerjaan adalah elemen-elemen penting dalam pekerjaan (dalam artian, kriteria pekerjaan menjelaskan apa saja yang sudah dibayar oleh organisasi untuk dikerjakan

karyawannya). Oleh karena itu, kriteria-kriteria ini penting. Kinerja individual dalam kriteria pekerjaan haruslah diukur, dibandingkan, dengan standar yang ada, dan hasilnya harus dikomunikasikan kepada setiap karyawan.

Pada suatu pekerjaan selalu memiliki lebih dari satu kriteria pekerjaan. Kriteria pekerjaan yang banyak, menjadi suatu aturan yang berperan penting bagi kesuksesan pekerjaan. Kriteria juga menjadi sesuatu yang bernilai bagi seorang karyawan sesuai dengan deskripsi pekerjaan karyawan tersebut. Berdasarkan kriteria, maka pekerjaan bisa memiliki beberapa tugas dan tanggung jawab.

2.1.3 Pihak Penilai

Pihak penilai merupakan pemegang peran penting dari proses penilaian kinerja. Dari penilaian yang dilakukan oleh pihak penilai inilah yang memiliki kemungkinan untuk membuat proses penilaian ini bersifat murni sebenarnya sebagai hasil proses penilaian secara objektif.

Menurut Rivai (2004:314), ada beberapa pihak yang menjadi pihak penilai dalam suatu proses penilaian, diantaranya:

1. Atasan.
2. Bawahan langsung (jika karyawan yang dinilai mempunyai bawahan langsung).
3. Penilaian oleh rekan kerja.
4. Penilaian oleh beberapa pihak terkait.

2.2 Standar Kinerja

Menurut Mathis dan Jackson (2006:380), standar kinerja (*performance standards*) mendefinisikan suatu tingkat hasil yang diharapkan dari kinerja dan

merupakan “pembanding kinerja” (*benchmarks*), atau “tujuan”, atau “target” tergantung pada pendekatan yang diambil. Standar kinerja yang realistis, dapat diukur, dipahami dengan jelas, akan bermanfaat baik bagi perusahaan maupun karyawan. Hal tersebut harus ditetapkan sebelum pekerjaan dilakukan. Standar-standar yang didefinisikan dengan baik memastikan setiap orang yang terlibat mengetahui tingkat pencapaian yang diharapkan.

2.3 Metode 360 Derajat

Menurut Ilyas (2002:16), bahwa penilaian kinerja yang baik adalah dengan mengevaluasi 360 derajat. Teknik ini merupakan pengembangan terakhir dari teknik penilaian sendiri. Teknik ini akan memberikan data yang lebih baik dan dapat dipercaya karena dilakukan penilaian silang bawahan, mitra, dan atasan personel.

Proses penilaian itu dilaksanakan dengan mengevaluasi diri sendiri dan menggabungkan seluruh informasi atau *feedback* baik dari manajer, rekan sejawat, atau pelanggan. Semua pihak yang menjadi anggota dalam organisasi dilibatkan dalam memberi informasi yang sangat diperlukan dalam penilaian. Pandangan klasik yang menganggap bahwa yang berhak menilai hanya pemimpin saja harus segera ditinggalkan dan berusaha menerapkan penilaian kinerja menurut paradigma baru.

Menurut Randall (1999), terdapat beberapa cara penilaian kinerja yang secara strategis dapat mengungkap kinerja bawahan secara lebih komprehensif. Berbagai penilaian kinerja tersebut adalah:

1. Penilaian atasan

Istilah atasan yang mengacu pada pimpinan langsung bawahan yang sedang dievaluasi. Banyak perusahaan yang menganggap atasan lebih mengetahui pekerjaan dan kinerja bawahan dari pada siapapun dan karena itu perusahaan memberikan seluruh tanggung jawab penilai kepada atasan.

2. Penilaian diri sendiri

Penggunaan penilaian diri sendiri, khususnya melalui partisipasi bawahan dalam menetapkan tujuan, dipopulerkan sebagai komponen *Management By Objectives* (MBO). Bawahan yang berpartisipasi dalam proses evaluasi mungkin akan lebih terlibat dan punya komitmen pada tujuan. Partisipasi bawahan mungkin juga akan membantu menjelaskan peran karyawan dan mengurangi konflik peran.

3. Penilaian rekan sejawat atau anggota tim.

Penggunaan penilaian anggota tim meningkat saat memasuki abad ke-21 ditinjau dari fokus korporasi Amerika yaitu partisipasi karyawan, kerjasama tim, dan pemberi wewenang. Salah satu alasannya adalah bawahan penilaian rekan sejawat terlihat sebagai alat prediksi kinerja masa mendatang yang bermanfaat.

4. Penilaian ke atas atau terbalik

Yaitu penilaian yang dilakukan oleh karyawan untuk menilai manajemen perusahaan, bagaimana opini karyawan tentang manajemen organisasi. Meskipun karyawan tidak mempunyai akses ke informasi mengenai seluruh dimensi kinerja manajer, mereka sering mempunyai akses ke informasi mengenai interaksi manajer-bawahan.

Manfaat yang akan diperoleh perusahaan menerapkan penilaian 360 derajat adalah semua penilaian yang diberikan oleh manajer, bawahan, rekan sejawat, dan diri sendiri dapat memberikan hasil penilaian yang objektif mengenai kinerja pihak yang dinilai.

2.3.1 Kelebihan Metode 360 Derajat

Menurut Randall (1999), 360 Derajat lebih efektif digunakan karena:

1. Metode 360 Derajat memberikan pemahaman terhadap individu mengenai bagaimana efektivitasnya sebagai karyawan, kolega maupun staf berdasarkan pandangan orang lain.
2. Metode 360 Derajat juga memberikan suatu *insight skill* dan perilaku yang diinginkan oleh organisasi sesuai dengan visi, misi dan tujuan yang diembannya.
3. 360 Derajat juga menghemat waktu manajer karena banyak orang yang terlibat dalam proses tersebut.
4. 360 Derajat juga membantu kerjasama tim supaya lebih efektif. Tim lebih memahami mengenai kinerja anggota tim dibandingkan *supervisor*.

2.3.2 Kekurangan Metode 360 Derajat

Menurut Randall (1999), adapun kekurangan dari metode 360 Derajat ini adalah:

1. Efektifitas 360 Derajat sangat ditentukan oleh seberapa terbuka budaya organisasi.

2. 360 Derajat tidak efektif dilaksanakan di lingkungan organisasi yang budayanya masih sangat paternalistik, sungkan (tertutup) dan tingginya politik kantor.
3. 360 Derajat tidak bisa digunakan untuk mengevaluasi karyawan baru atau calon karyawan.

2.3.3 Proses Penilaian Menggunakan Metode 360 Derajat

Prosedur penilaian meliputi:

1. Karyawan dinilai oleh atasan, rekan kerja dan bawahan di unit kerja masing-masing dan unit kerja terkait yang berhubungan dengan unit kerja karyawan.
2. Sistem penilaian dengan proporsional dan professional dengan mengesampingkan subjektifitas, dimana penilai ditunjuk oleh bagian HRD.

2.4 Skala *Likert*

Angket atau disebut juga kuisioner adalah daftar pertanyaan yang diberikan kepada orang lain yang bersedia memberikan respon sesuai dengan permintaan pengguna. Tujuan penyebaran angket adalah mencari informasi dari responden tanpa khawatir bila responden memberikan jawaban yang tidak sesuai dengan kenyataan (Riduwan, 2005). Dalam penelitian ini, angket dibutuhkan untuk mengukur tingkat kelayakan pengguna aplikasi.

Menurut Husein (2003), Skala *Likert* berhubungan dengan pernyataan seseorang terhadap sesuatu. Skor pada skala *Likert* berarah *positif* dan *negative*. Skala *Likert* digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau kelompok tentang kejadian atau gejala social.

Perhitungan skor penilaian untuk setiap pertanyaan (QS) didapatkan dari jumlah pengguna (PM) dikalikan dengan skala nilai (N). jumlah skor tertinggi (STot) didapatkan dari skala tertinggi (NT) dikalikan jumlah pertanyaan (Qtot) dikalikan total pengguna (Ptot). Sedangkan nilai persentase akhir (Pre) diperoleh dari jumlah skor hasil pengumpulan data (JSA) dibagi jumlah skor tertinggi (STot) dikalikan 100%.

Rumus skala *Likert*:

$$QS(n) = PM \times N \dots\dots\dots(2.1)$$

$$STot = NT \times Qtot \times Ptot \dots\dots\dots(2.2)$$

$$Pre = \frac{JSA}{STot} \times 100\% \dots\dots\dots(2.3)$$

Keterangan:

QS(n) = Skor pertanyaan ke-n

PM = Jumlah pengguna yang menjawab

N = Skala nilai

STot = Total skor tertinggi

NT = Skala nilai tertinggi

Qtot = Total pertanyaan

Ptot = Total pengguna

Pre = Persentase akhir (%)

JSA = Jumlah skor akhir

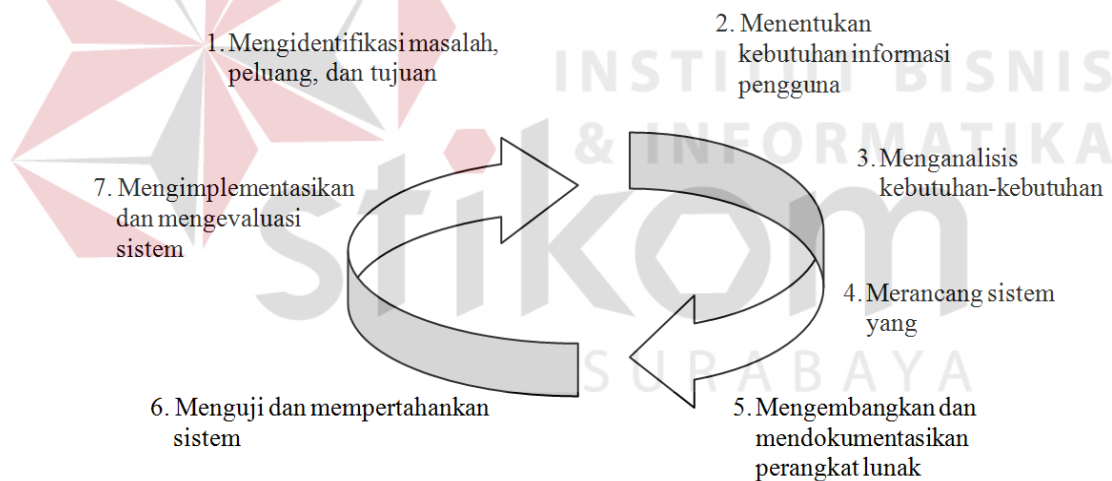
Analisis dilakukan dengan melihat persentase akhir dari proses perhitungan skor. Nilai persentase kemudian dicocokkan dengan kriteria interpretasi skor yang dapat dilihat pada Tabel 2.1.

Tabel 2.1 Keterangan Nilai

Nilai	Keterangan
0 – 20%	Sangat Kurang
21 – 40%	Kurang
41 – 60%	Cukup
61 – 80%	Baik
81 – 100%	Sangat Baik

2.5 Siklus Hidup Pengembangan Sistem

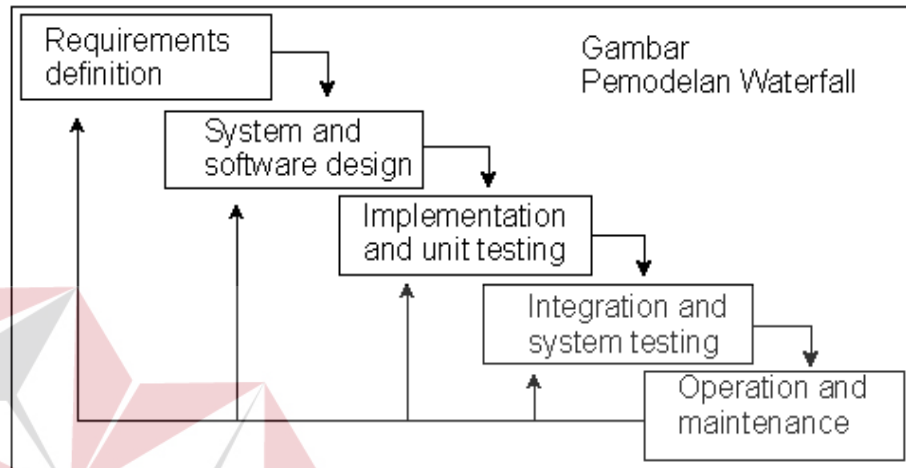
Menurut Kendall dan Kendall (2003:11), Siklus Hidup Pengembangan Sistem adalah pendekatan melalui beberapa tahap untuk menganalisis dan merancang sistem dimana sistem tersebut telah dikembangkan dengan sangat baik melalui penggunaan siklus kegiatan penganalisis dan pemakai secara spesifik.



Gambar 2.1 Tujuh Tahap Siklus Hidup Pengembangan Sistem (Kendall dan Kendall, 2003)

Menurut Pressman (2007), menjelaskan bahwa nama lain dari Model *Waterfall* adalah *Linear Sequential Model*. Model ini merupakan model yang paling banyak dipakai dalam *Software Engineering*. Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu

menuju ke tahap analisis, *design*, *coding*, *testing* dan *maintenance*. Disebut dengan *Waterfall* karena tahap demi tahap yang harus dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Berikut ini adalah gambar dari Model *Waterfall* secara umum.



Gambar 2.2 Model *Waterfall* (Pressman, 2007)

Pada Gambar 2.2 di atas adalah gambar tahapan umum dari model proses *Waterfall*. Akan tetapi menurut Pressman (2007), memecah model ini meskipun secara garis besar sama dengan tahapan-tahapan Model *Waterfall* pada umumnya. Berikut ini adalah penjelasan dari tahap-tahap yang dilakukan di dalam Model *Waterfall* menurut Pressman:

1. *Software Requirements Analysis*

Proses pencarian kebutuhan diintensifkan dan difokuskan pada perangkat lunak. Untuk mengetahui sifat dari program yang akan dibuat, maka para *software engineer* harus mengerti tentang *user interface*. Dari kedua aktivitas tersebut (pencarian kebutuhan sistem dan perangkat lunak) harus didokumentasikan dan ditunjukkan kepada pelanggan.

2. *Design*

Proses ini digunakan untuk mengubah kebutuhan-kebutuhan di atas menjadi representasi ke dalam bentuk “*blueprint*” perangkat lunak sebelum pengkodean dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Seperti dua aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari perangkat lunak.

3. *Coding*

Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses pengkodean. Tahap ini merupakan implementasi dari tahap desain yang secara teknis nantinya dikerjakan oleh *programmer*.

4. *Testing*

Sesuatu yang dibuat haruslah diujicobakan. Demikian juga dengan perangkat lunak. Semua fungsi-fungsi perangkat lunak harus diujicobakan, agar perangkat lunak bebas dari *error* dan hasilnya benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

5. *Maintenance*

Pemeliharaan suatu perangkat lunak diperlukan, termasuk di dalamnya adalah pengembangan, karena perangkat lunak yang dibuat tidak selamanya hanya seperti itu. Pengembangan diperlukan ketika adanya perubahan dari eksternal perusahaan seperti ketika ada pergantian sistem operasi atau perangkat lain.

2.6 Analisis dan Perancangan Sistem

Menurut Kendall dan Kendall (2003:7), Analisis dan Perancangan Sistem berupaya menganalisis input data atau aliran data secara sistematis, memproses atau mentransformasikan data, menyimpan data, dan menghasilkan output informasi dalam konteks bisnis khusus. Selanjutnya, analisis dan perancangan sistem digunakan untuk menganalisis, merancang, dan mengimplementasikan peningkatan-peningkatan fungsi bisnis yang bisa dicapai melalui penggunaan sistem informasi terkomputerisasi.

2.7 Web

Menurut Shelly dan Vermaat (2010), *Web* adalah koleksi dokumen elektronik milik semua orang di dunia yang mengaksesnya melalui internet menggunakan *web browser*. Menurut Simamarta (2010), Aplikasi *web* adalah sebuah sistem informasi yang mendukung interaksi pengguna melalui antarmuka berbasis *web*. Fitur-fitur aplikasi *web* biasanya berupa data *persistence*, mendukung transaksi dan komposisi halaman *web* dinamis yang dapat dipertimbangkan sebagai *hibridasi*, antara *hipermedia* dan sistem informasi. Aplikasi web adalah bagian dari *client-side* yang dapat dijalankan oleh *browser web*. *Client-side* mempunyai tanggung jawab untuk pengeksekusian proses bisnis.

Interaksi *web* menurut Simamarta (2010), dibagi dalam tiga langkah utama, yaitu:

1. Permintaan

Pengguna mengirimkan permintaan ke *server web*, melalui halaman *web* yang ditampilkan pada *browser web*.

2. Pemrosesan

Server web menerima permintaan yang dikirimkan oleh pengguna, kemudian memproses permintaan tersebut.

3. Jawaban

Browser menampilkan hasil dari permintaan pada jendela *browser*.

2.8 Testing

Menurut Romeo (2003), *testing* adalah proses pemantauan kepercayaan akan kinerja program atau sistem sebagaimana yang diharapkan. *Testing software* adalah proses mengoperasikan *software* dalam suatu kondisi yang dikendalikan untuk verifikasi, mendeteksi *error* dan validasi. Verifikasi adalah pengecekan atau pengetesan entitas-entitas, termasuk *software*, untuk pemenuhan dan konsistensi dengan melakukan evaluasi hasil terhadap kebutuhan yang telah ditetapkan. Validasi adalah melihat kebenaran sistem apakah proses yang telah dituliskan sudah sesuai dengan apa yang dibutuhkan oleh pengguna. Deteksi *error* adalah testing yang berorientasi untuk membuat kesalahan secara intensif, untuk menentukan apakah suatu hal tersebut terjadi bilamana tidak seharusnya terjadi atau suatu hal tersebut tidak terjadi. *Test case* merupakan suatu tes yang dilakukan berdasarkan pada suatu inisialisasi, masukan, kondisi ataupun hasil yang telah ditentukan sebelumnya. Adapun kegunaan dari *test case* ini, adalah sebagai berikut.

1. Untuk melakukan *testing* kesesuaian suatu komponen terhadap disain *White Box Testing*.
2. Untuk melakukan *testing* kesesuaian suatu komponen terhadap spesifikasi *Black Box Testing*.

2.8.1 *White Box Testing*

Menurut Romeo (2003), *white box testing* adalah suatu metode desain *test case* yang menggunakan struktur kendali dari desain prosedural. Seringkali *white box testing* diasosiasikan dengan pengukuran cakupan tes, yang mengukur persentase jalur-jalur dari tipe yang dipilih untuk dieksekusi oleh *test cases*. *White box testing* dapat menjamin semua struktur *internal* data dapat dites untuk memastikan validitasnya.

Cakupan pernyataan, cabang dan jalur adalah suatu teknik *white box testing* yang menggunakan alur logika dari program untuk membuat *test cases*. Alur logika adalah cara dimana suatu bagian dari program tertentu dieksekusi saat menjalankan program. Alur logika suatu program dapat direpresentasikan dengan *flow graph*.

2.8.2 *Black Box Testing*

Menurut Romeo (2003), *black box testing* dilakukan tanpa adanya suatu pengetahuan tentang detail struktur internal dari sistem atau komponen yang dites, juga disebut sebagai *functional testing*. *Black box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*.

Dengan adanya *black box testing*, perancang *software* dapat menggunakan kebutuhan fungsional pada suatu program. *Black box testing* dilakukan untuk melakukan pengecekan apakah sebuah *software* telah bebas dari *error* dan fungsi-fungsi yang diperlukan telah berjalan sesuai dengan yang diharapkan.