

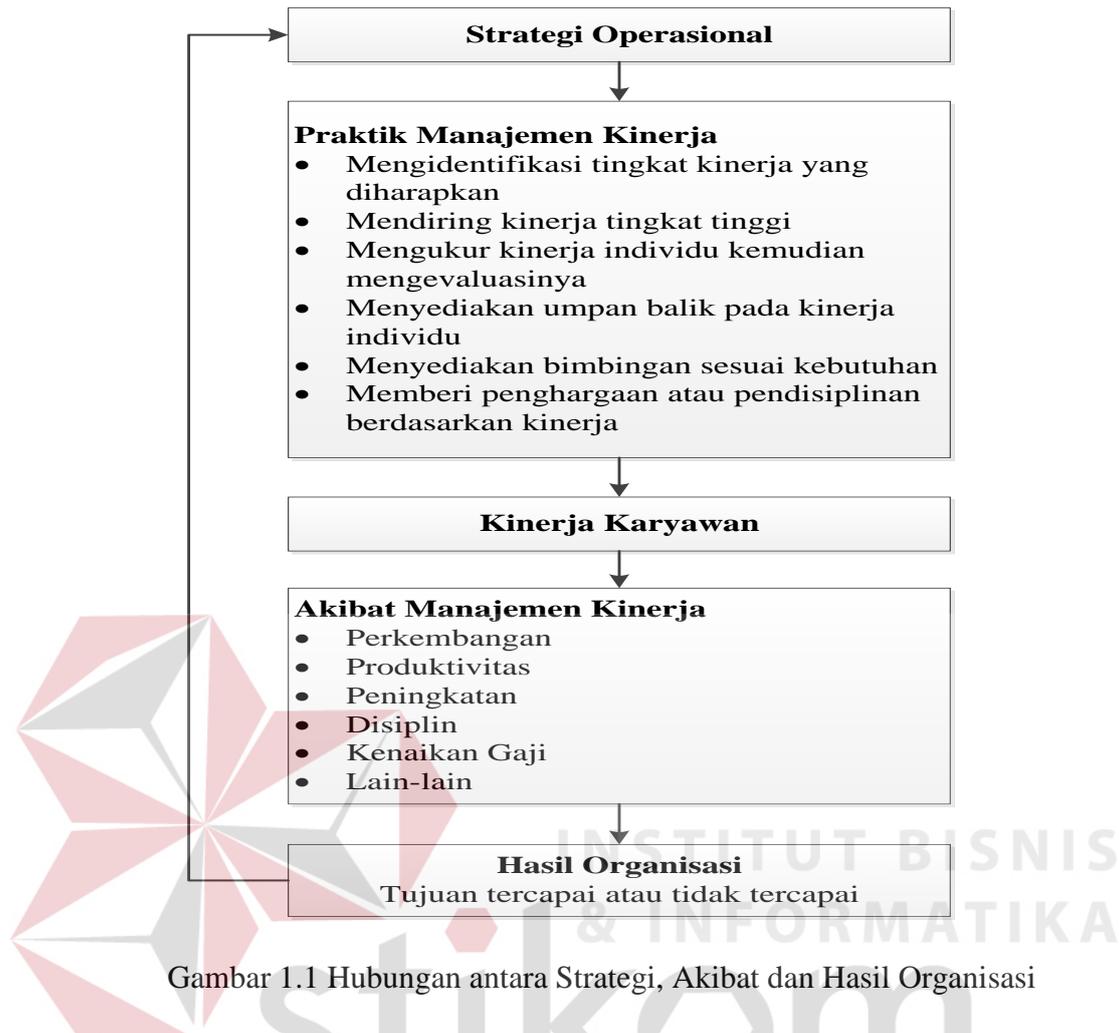
## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Manajemen Sumber Daya Manusia**

Manajemen sumber daya manusia merupakan suatu perencanaan, pengorganisasian, pelaksanaan dan pengawasan terhadap pengadaan, pengembangan, pemberian balas jasa, pengintegrasian, pemeliharaan dan pemisahan tenaga kerja dalam rangka mencapai tujuan perusahaan. (Mangkunegara, 2013).

Strategi organisasional sangatlah penting, yaitu sebuah penetapan tujuan-tujuan dan sasaran-sasaran jangka panjang yang bersifat mendasar bagi sebuah organisasi yang dilanjutkan dengan penetapan rencana aktivitas dan pengalokasian sumber daya yang diperlukan guna mencapai sasaran-sasaran tersebut. Dalam suatu strategi operasional tersebut akan terdapat praktik manajemen kinerja yang dilakukan pada kinerja karyawan. Pada praktik manajemen kinerja akan menimbulkan suatu akibat yang sangatlah berpengaruh penting pada hasil organisasi akan tercapai atau tidaknya. Penjelasan Gambar 2.1 Hubungan antara Strategi, Akibat dan Hasil Organisasi sebagai berikut: (Mathis and Jackson 2009)

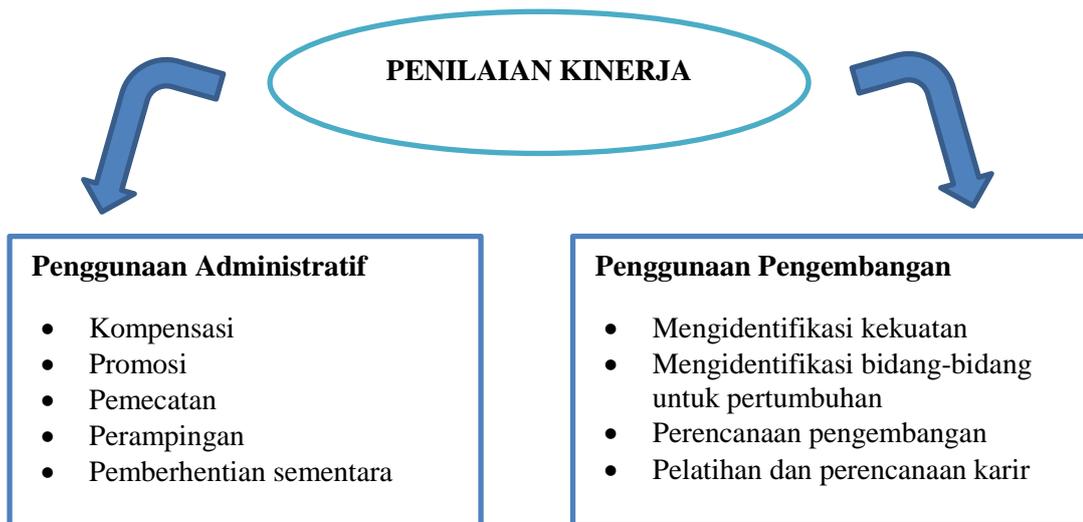


Gambar 1.1 Hubungan antara Strategi, Akibat dan Hasil Organisasi

## 2.2 Penilaian Kinerja Karyawan

Penilaian kinerja (*performance appraisal*) adalah proses mengevaluasi seberapa baik karyawan melakukan pekerjaan mereka jika dibandingkan dengan seperangkat standar, dan kemudian mengkomunikasikan informasi tersebut kepada karyawan. Penilaian kinerja juga disebut pemeringkatan karyawan, evaluasi karyawan, tinjauan kinerja, evaluasi kinerja, dan penilaian hasil (Mathis and Jackson 2009).

Terdapat 2 penggunaan dalam penilaian kinerja, yaitu penggunaan administratif dan penggunaan pengembangan. Dapat dilihat pada Gambar 2.2 Penggunaan-Penggunaan dalam Penilaian Kinerja.



Gambar 1.2 Penggunaan-Penggunaan dalam Penilaian Kinerja

Penjelasan dari penggunaan administratif adalah sistem penilaian kinerja sering kali menjadi penghubung antara penghargaan yang diinginkan karyawan dan produktivitas mereka. Hubungan tersebut dapat diperkirakan sebagai berikut:

Produktivitas → Penilaian Kinerja → Penghargaan

Kompensasi yang berbasis kinerja menegaskan ide bahwa kenaikan gaji seharusnya diberikan untuk pencapaian kinerja daripada untuk senioritas. Dalam sistem ini, manajer historis telah menjadi pengevaluasi dari kinerja bawahan dan juga yang membuat rekomendasi kompensasi untuk karyawan. Jika ada bagian dari proses penilaian yang gagal, para karyawan yang berkinerja baik tidak menerima kenaikan gaji yang lebih besar, yang akan menyebabkan adanya ketidakadilan dalam kompensasi yang dirasakan karyawan.

Dalam penggunaan administratif pada penilaian kinerja tidak hanya mendapatkan sebuah kompensasi saja, tetapi penggunaan administratif lainnya seperti promosi, pemecatan, perampingan serta pemberhentian sementara.

Penggunaan pengembangan penilaian kinerja dapat menjadi sumber utama informasi dan umpan balik untuk karyawan, yang sering kali merupakan kunci perkembangan mereka di masa depan. Dalam proses mengidentifikasi kekuatan, kelemahan, potensi, dan kebutuhan pelatihan karyawan melalui umpan balik penilaian kinerja, para supervisor dapat menginformasikan kepada karyawan mengenai kemajuan mereka, mendiskusikan area-area yang membutuhkan pengembangan, dan mengidentifikasi rencana pengembangan.

Tujuan dari umpan balik pengembangan adalah lebih kepada mengubah atau menguatkan perilaku individu, daripada untuk membandingkan antar individu seperti kasus penggunaan administratif dalam penilaian kinerja. Penguatan positif untuk perilaku yang diharap memberi kontribusi pada pengembangan individu dan organisasional

### **2.3 Standar Kompetensi**

Standar kompetensi dapat didefinisikan sebagai suatu kemampuan baku (patokan kemampuan) yang dilandasi oleh pengetahuan keterampilan dan didukung sikap kerja yang diperlukan dalam melaksanakan suatu tugas/pekerjaan di tempat kerja. (Daryanto & Bintoro 2014).

Selain daripada itu standar kompetensi dapat juga dikatakan sebagai persyaratan atau patokan yang meliputi tuntutan bagi seseorang untuk melaksanakan tugasnya di tempat kerja.

### **2.4 Kategori Standar Kompetensi**

Hampir semua sistem yang berdasarkan kompetensi melakukan pengembangan standar kompetensi dengan mengacu pada tiga kategori yaitu

standar industri, standar perusahaan, standar umum atau lingkup industri (Daryanto & Bintoro 2014).

Standar industri merupakan standar yang memiliki relevansi lebih luas dengan industri tertentu. Standar-standar ini akan memiliki kompetensi sesuai dengan industri tersebut.

Standar perusahaan adalah standar yang dikembangkan hanya untuk satu perusahaan tertentu. Kebutuhan standar perusahaan dibuat melalui pengakuan organisasi-organisasi, bahwa standar yang dikembangkan pada tingkat industri tidak relevan dengan mereka, sebagaimana mereka juga pada industri lain. Namun demikian, pada standar perusahaan banyak kompetensi-kompetensi yang mungkin diambil dari standar-standar lain dan standar industri.

Standar umum atau lingkup industri merupakan standar yang dapat diaplikasikan pada bidang kejuruan yang memiliki cakupan lebih luas, namun masih dalam konteks standar kompetensi manajemen atau kompetensi pelatihan yang dapat diaplikasikan pada setiap pekerjaan dan bermakna sama. Dalam kompetensi ini boleh jadi ada standar antar industri yang lain.

## **2.5 Penilaian Berdasarkan Kompetensi**

Struktur standar kompetensi suatu bidang keahlian terdiri dari sejumlah unit kompetensi. Setiap unit kompetensi memuat komponen-komponen sebagai berikut: (Daryanto & Bintoro 2014).

### **1. Judul unit/ kompetensi**

Judul unit/kompetensi pernyataan yang akan dikerjakan oleh seseorang di tempat kerja atau pernyataan yang memberikan penjelasan umum tentang pekerjaan yang harus dilakukan di tempat kerja.

## 2. Uraian unit/ kompetensi

Uraian kompetensi dimaksudkan untuk memberikan informasi tentang kompetensi yang telah dituliskan dalam judul kompetensi tersebut.

## 3. Elemen/ Sub kompetensi

Setiap kompetensi terbagi atas beberapa sub kompetensi dan dalam standar kompetensi dikenal dengan istilah elemen. Elemen atau sub kompetensi adalah dasar pembentukan standar kompetensi yang merupakan pernyataan sederhana dari kegiatan yang harus dilaksanakan untuk memperlihatkan bahwa seseorang telah mencapai standar yang dibutuhkan.

## 4. Kriteria unjuk kerja

Kriteria unjuk kerja untuk elemen tertentu dijelaskan atau dinyatakan dalam bentuk kriteria dengan jelas dan detail.

Penilaian kompetensi diawali dengan penentuan kompetensi yang akan dinilai, setelah itu melakukan pembobotan terhadap setiap kompetensi yaitu *knowledge, skill, attitude*. Pembobotan dilakukan bertujuan untuk mendapatkan hasil akhir yang berkorelasi antara setiap kompetensi tersebut. Dari hasil akhir yang didapat maka dilakukan penyesuaian nilai terhadap setiap penilain kriteria kompetensi. (Fuad & Gofur 2009)

### **2.6 Software Engineering Body of Knowledge (SWEBOK)**

SWEBOK adalah sebuah panduan (*guideline*) yang dihasilkan dari sebuah *project* gagasan IEEE Computer Society. Panduan (*guideline*) ini disusun sejak tahun 1998 dimana tim tersebut mulai menyusun pemahaman standar (*body*

*of knowledge*) tentang bidang ilmu *software engineering*. Terdapat 5 tujuan utama pada SWEBOK, yaitu: (IEEE Computer Society 2014)

1. Untuk memperlihatkan kesamaan pandangan tentang rekayasa sistem di seluruh dunia.
2. Untuk memperjelas tempat dan menetapkan batas dari rekayasa sistem dan hubungannya dengan disiplin ilmu lain seperti ilmu komputer, manajemen proyek, teknik komputer dan matematika.
3. Untuk membuat karakter isi dari disiplin ilmu rekayasa sistem.
4. Untuk memberikan akses topik ke SWEBOK.
5. Untuk memberikan pengetahuan dasar bagi pengembangan kurikulum dan sertifikasi serta perijinan.

Terdapat 4 (empat) *Knowledge Areas* (KAs) utama dari pengembangan *software* pada SWEBOK. Berikut adalah penjelasan tentang *Knowledge Areas* (KAs) tersebut:

### 2.6.1 *Software Requirements*

Kebutuhan perangkat lunak dapat diartikan sebagai properti yang harus ditampilkan dalam rangka memecahkan beberapa masalah di dunia nyata. Area pengetahuan dari *software requirements* adalah elisitasi, analisis, spesifikasi, dan validasi persyaratan perangkat lunak. (IEEE Computer Society 2014)

*Software requirements* menghasilkan informasi tentang desain yang akan menjadi dasar, sehingga dapat mengetahui dimana sebuah sistem akan digunakan, oleh siapa, dan layanan apa yang harus disediakan. Berikut ini adalah tahapan dalam *software requirements*:

## 1. *Requirements Elicitation*

Tahap elisitasi kebutuhan adalah tahap pertama dalam membangun sebuah perangkat lunak. salah satu prinsip dasar yang baik pada proses elisitasi adalah melakukan komunikasi secara efektif antara berbagai pemangku kepentingan. Selanjutnya, dari komunikasi ini dilanjutkan ke proses *Software Development Life Cycle* (SDLC). Proses ini adalah proses yang sangat penting sebelum pembangunan perangkat lunak dimulai. Elemen penting lain dari persyaratan elisitasi adalah bagaimana ruang lingkup dari proyek yang akan dikerjakan.

## 2. *Requirements Analysis*

*Requirements analysis* adalah tahapan yang digunakan untuk mempelajari kebutuhan pengguna, sehingga didapatkan definisi kebutuhan sistem atau perangkat lunak yang bertujuan untuk mendefinisikan apa yang harus dikerjakan oleh perangkat lunak dalam memenuhi keinginan pengguna dan memahami masalah secara menyeluruh. Pada tahap ini secara detil terdapat proses sebagai berikut:

1. Mendeteksi dan menyelesaikan permasalahan sesuai dengan kebutuhan organisasi.
2. Menemukan batasan dari perangkat lunak dan bagaimana perangkat lunak berinteraksi dengan organisasi dan lingkungan operasional.
3. Menguraikan spesifikasi kebutuhan.

## 3. *Requirements Specification*

*Requirements specification* adalah sebuah kegiatan yang mengacu pada pembuatan dokumen yang dapat ditinjau secara sistematis, dievaluasi, dan

disetujui. Pada tahap ini, sama sekali tidak dibahas bagaimana metode pengembangan yang akan dilakukan. Dokumen spesifikasi kebutuhan ini membahas beberapa hal yang berkaitan dengan *system definition document*, *system requirements specification*, dan *software requirements specification*.

#### **4. Requirements Specification**

Diperlukan validasi dan verifikasi terhadap dokumen-dokumen persyaratan yang telah dibuat. Persyaratan-persyaratan divalidasi untuk menjamin bahwa *engineer* perangkat lunak telah memahami persyaratan, serta perlu juga untuk memverifikasi bahwa dokumen persyaratan telah sesuai dengan standar perusahaan dan dapat dimengerti, konsisten, serta lengkap. Proses validasi dan verifikasi ini melibatkan pengguna sebagai pihak yang menilai dan memberi *feedback* (umpan balik).

##### **2.6.2 Software Design**

*Software design* adalah tahap yang memainkan peran penting dalam mengembangkan perangkat lunak. *Software design* adalah proses yang mendefinisikan arsitektur, komponen, *interface*, dan karakteristik sebuah sistem atau komponen lainnya serta hasil dari proses tersebut, sehingga dapat dikatakan bahwa *software design* adalah cetak biru dari solusi yang akan diimplementasikan (IEEE Computer Society 2014).

*Software design* secara spesifik memiliki hubungan dengan *software requirements*, *software construction*, *software engineering management*, *software engineering models and methods*, *software quality*, dan *computing foundations*. Disamping itu, *software design* berupaya menganalisis *input* data secara

sistematis, memproses atau mentransformasikan data, menyimpan data, dan menghasilkan *output* informasi. Berikut ini adalah tahapan dalam *software design*:

### 1. *Software Structure and Architecture*

*Software structure and architecture* adalah proses mendeskripsikan dan mendefinisikan bagaimana *software* dibentuk dan diorganisasikan ke dalam komponen-komponen yang akan membentuk *software* tersebut. Hasil dari proses ini adalah sekumpulan model yang mendeskripsikan tujuan serta gambaran dari *software* yang akan dibangun, adapun model-model ini bisa dibangun dengan menggunakan *modeling language* (bahasa yang digunakan untuk membuat dan menyajikan informasi atau *knowledge*). *Modeling language* yang digunakan pada pengembangan sistem ini, yaitu *system flow* dan *data flow diagram* sebagai berikut:

#### A.1. *System Flow*

*System flow* atau bagan alir sistem merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. *System flow* menunjukkan urutan-urutan dari prosedur yang ada di dalam sistem dan menunjukkan apa yang dikerjakan sistem. Simbol-simbol yang digunakan dalam *system flow* ditunjukkan pada Gambar 2.3 Simbol-Simbol pada *System Flow*.

Mengenai penjelasan dari simbol-simbol yang digunakan dalam *system flow* adalah sebagai berikut:

##### 1. Simbol Dokumen

Menunjukkan dokumen *input* dan *output* baik untuk proses manual atau komputer.

##### 2. Simbol Kegiatan Manual

Menunjukkan pekerjaan manual.

3. Simbol Simpanan *Offline*

Menunjukkan file non-komputer yang diarsip.

4. Simbol Proses

Menunjukkan kegiatan proses dari operasi program komputer.

5. Simbol *Database*

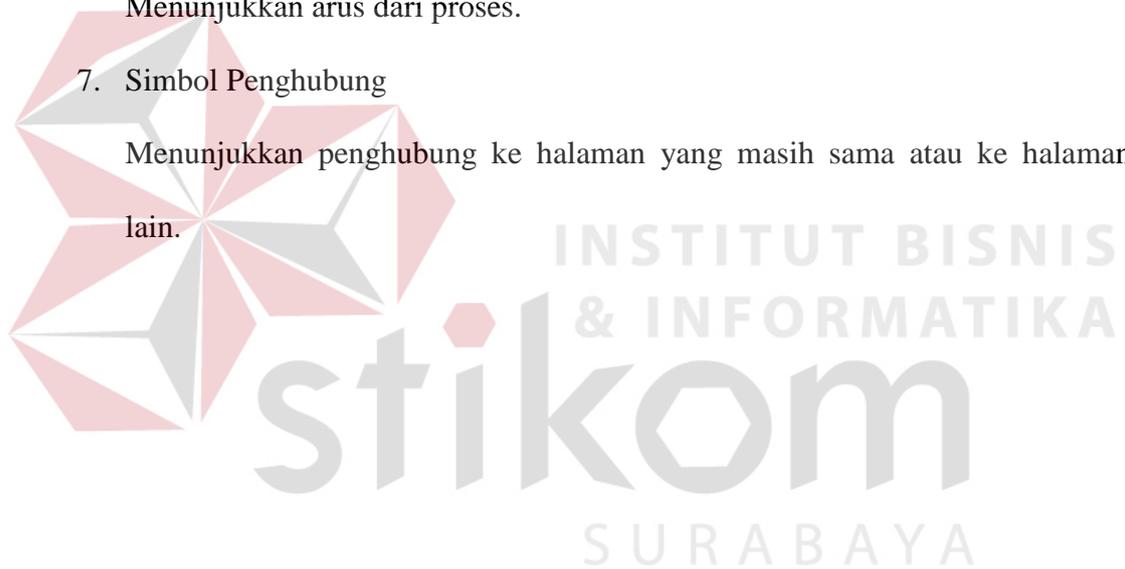
Menunjukkan tempat untuk menyimpan data hasil operasi komputer.

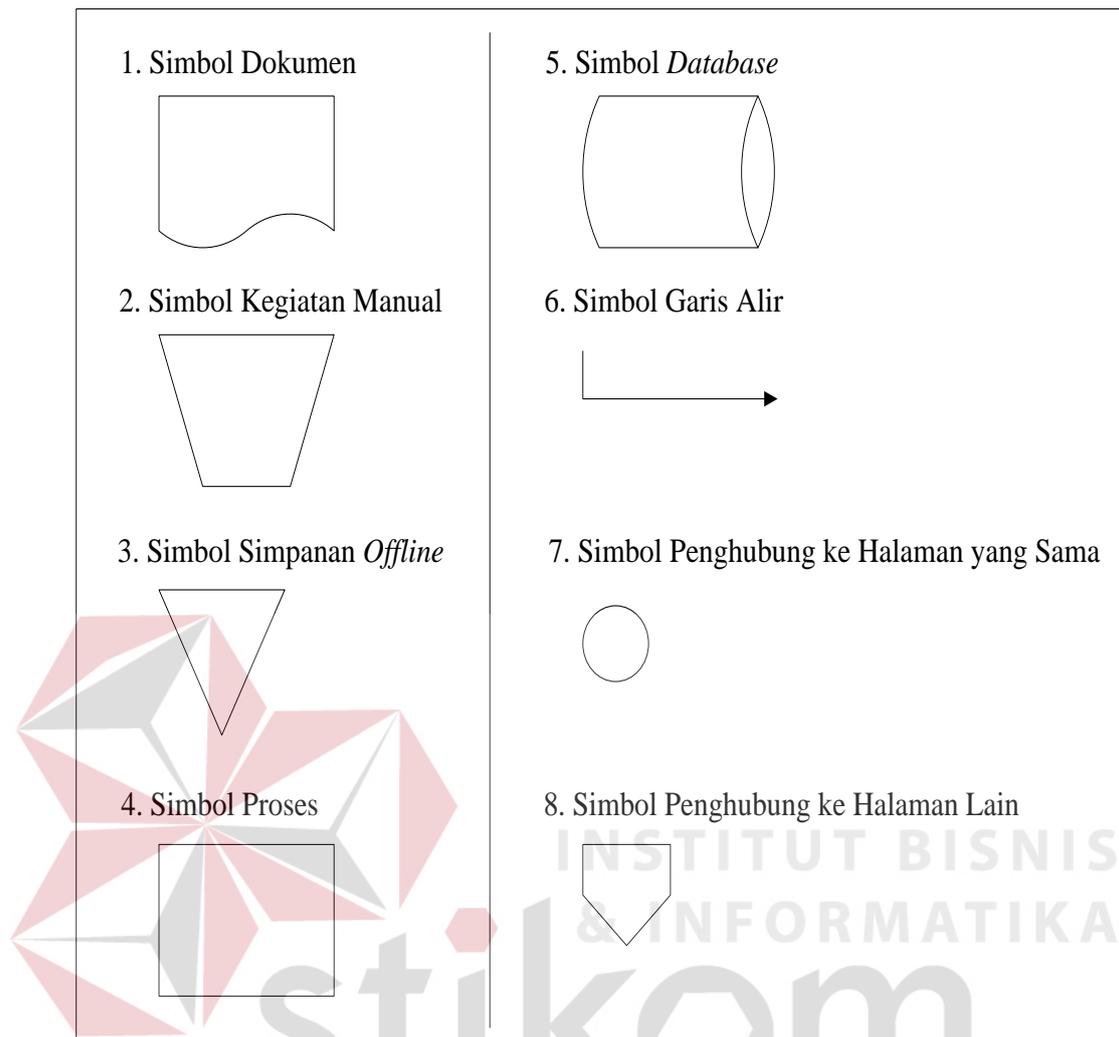
6. Simbol Garis Alir

Menunjukkan arus dari proses.

7. Simbol Penghubung

Menunjukkan penghubung ke halaman yang masih sama atau ke halaman lain.





Gambar 1.3 Simbol-Simbol pada *System Flow*

## A.2. *Data Flow Diagram*

*Data flow diagram* sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik di tempat data tersebut mengalir. Menurut Yourdon (1989, p139) diagram aliran data adalah model atau alat yang digunakan untuk menggambarkan sistem sebagai jaringan dari sekumpulan proses fungsional, yang dihubungkan satu dengan lainnya oleh suatu aliran data dan meneruskannya menjadi data. Ada tiga tingkatan dalam diagram aliran data, yaitu:

### 1. *Context Diagram*

Merupakan tingkatan yang paling pertama, yang menggambarkan ruang lingkup sistem dari sistem yang digunakan. Diagram ini hanya memiliki satu proses yang menggambarkan sistem secara keseluruhan dan hubungan antara sistem dengan unit-unit di luar sistem tersebut.

### 2. Diagram Nol

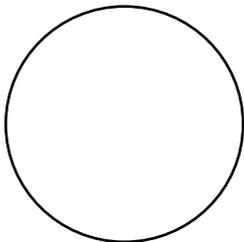
Merupakan diagram yang menggambarkan proses-proses dan aliran data yang terjadi di dalam suatu sistem. Proses-proses ini dapat dipecah menjadi proses-proses dan aliran data yang lebih detail.

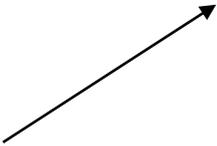
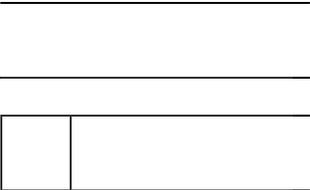
### 3. Diagram Rinci

Merupakan diagram yang menggambarkan rincian proses-proses yang ada pada diagram nol dan rincian proses-proses ini dapat dipecah lagi menjadi proses-proses yang lebih detail.

Menurut Yourdon (1989, p141-152), diagram aliran data terdiri dari dari simbol-simbol pada Tabel 2.1 Simbol yang Digunakan pada Diagram Aliran Data.

Tabel 1.1 Simbol yang Digunakan pada Diagram Aliran Data

Nama	Penjelasan	Notasi
Proses <i>(bubble</i> atau <i>function</i> atau <i>transformation)</i>	Proses menggambarkan bagian dari sistem yang mengolah masukan menjadi keluaran. Proses digambarkan dengan sebuah lingkaran.	

Nama	Penjelasan	Notasi
Aliran ( <i>flow</i> )	Aliran menggambarkan perpindahan informasi dari satu bagian ke bagian lain dari sistem. Awal panah menggambarkan asal data sedangkan arah panah menggambarkan tujuan.	
<i>Store</i>	Simbol ini digunakan untuk menggambarkan penyimpanan data.	
Terminator	Merupakan simbol yang menggambarkan entitas yang dapat berupa orang, kelompok, atau organisasi yang berhubungan dengan sistem.	

## 2. *User Interface Design*

*User interface* merupakan bagian yang menjembatani interaksi antara manusia dan komputer, sehingga dapat mengoperasikan dan mengontrol komputer yang efektif. *User interface* dirancang untuk menyesuaikan keterampilan, pengalaman, dan harapan pengguna. Tujuan dari *user interface design* adalah membuat interaksi pengguna yang sederhana dan efisien mungkin. Proses desain harus menyeimbangkan fungsi teknis dan elemen visual untuk menciptakan

sebuah sistem yang tidak hanya operasional saja tetapi juga disesuaikan dengan kebutuhan pengguna.

### 2.6.3 *Software Construction*

Pada tahap ini ialah melakukan konversi hasil desain ke sistem informasi yang lengkap melalui tahapan *coding* atau pengkodean termasuk bagaimana, membuat basis data dan menyiapkan prosedur kasus pengujian, mempersiapkan berkas atau *file* pengujian, pengkodean pengompilasian, memperbaiki dan membersihkan program serta melakukan peminjaman pengujian. *Construction* ini memiliki beberapa tahapan secara umum (IEEE *Computer Society* 2014).

#### 1. *Software Construction Fundamentals*

Pada tahap pertama dilakukan pendefinisian dasar tentang prinsip-prinsip yang digunakan dalam proses implementasi seperti minimalisasi kompleksitas, mengantisipasi perubahan, dan standar yang digunakan.

#### 2. *Managing Construction*

Bagian ini mendefinisikan tentang model implementasi yang digunakan, rencana implementasi, dan ukuran pencapaian dari implementasi tersebut.

#### 3. *Practical Considerations*

Bagian ini membahas tentang desain implementasi yang digunakan, bahasa pemrograman yang digunakan, kualitas dari implementasi yang dilakukan, proses pengetesan dan integritas.

Dalam proses konstruksi atau pembuatan sistem, digunakan beberapa *tools* pendukung, yaitu:

## 1. Adobe Dreamweaver

Adobe dreamweaver adalah aplikasi desain dan pengembangan *web* yang menyediakan *editor* WYSIWYG visual dan kode *editor* dengan fitur standar seperti *syntax high lighting*, *code completion*, dan *code collapsing* serta fitur lebih canggih seperti *real-time syntax checking* dan *code introspection* untuk menghasilkan petunjuk kode untuk membantu pengguna dalam menulis kode. Tata letak tampilan *design* memfasilitasi desain cepat dan pembuatan kode seperti memungkinkan pengguna dengan cepat membuat tata letak dan manipulasi elemen HTML. Dreamweaver memiliki fitur *browser* yang terintegrasi untuk melihat halaman *web* yang dikembangkan di jendela pratinjau program sendiri agar konten memungkinkan untuk terbuka di *web browser* yang telah ter-*install*.

## 2. Oracle 10g

Oracle database merupakan sebuah *object-relational* database. Oracle database didesain untuk menyimpan dan mengelola informasi yang berskala enterprise. Oracle database dapat menghemat biaya manajemen dan menyediakan layanan dengan kualitas yang tinggi (Greenberg 2004).

Oracle database berfungsi dalam mengelola informasi secara terbuka, komprehensif dan terintegrasi. Oracle database menyediakan solusi yang efisien dan efektif karena kemampuannya, yaitu dapat bekerja di lingkungan *client/server* (pemrosesan tersebar), menangani manajemen *space* dan basis data yang besar, mendukung akses data secara simultan, *performance* pemrosesan transaksi yang tinggi, menjamin ketersediaan yang terkontrol.

### 3. PHP

Menurut Saputra (2011, p.1) PHP atau yang memiliki kepanjangan PHP *Hypertext Preprocessor* merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu *website* dinamis. PHP menyatu dengan kode HTML, maksudnya adalah beda kondisi. HTML digunakan sebagai pembangun atau pondasi dari kerangka *layout web*, sedangkan PHP difungsikan sebagai prosesnya sehingga dengan adanya PHP tersebut, *web* akan sangat mudah dilakukan perawatan.

PHP berjalan pada sisi *server* sehingga PHP disebut juga sebagai bahasa *server side scripting*. Artinya bahwa dalam setiap menjalankan PHP, wajib adanya *web server*.

#### 2.6.4 *Software Testing*

*Software testing* meliputi verifikasi yang dinamis dari tingkah laku sebuah sistem yang diwakili oleh beberapa contoh kasus uji coba (IEEE Computer Society 2014).

Kasus uji coba tersebut dilakukan dengan memberikan masukan kepada sistem agar muncul tingkah laku/reaksi yang diharapkan, begitu pula sebaliknya.

Dalam uji coba sistem, terdapat hal-hal yang harus diperhatikan, yaitu:

1. Fundamental dari uji coba sistem tersebut. Didalamnya dijelaskan mengenai terminologi dari uji coba terkait, kunci masalah dari uji coba, dan hubungan uji coba tersebut dengan aktifitas lainnya di dalam sistem tersebut.
2. Tingkatan dari uji coba. Didalamnya dijelaskan tentang target dari uji coba dan tujuan dari uji coba tersebut.

3. Teknik dari uji coba. Didalamnya meliputi uji coba berdasarkan intuisi dan pengalaman dari seorang *tester*, diikuti oleh teknik berdasarkan spesifikasi, teknik berdasarkan kode, teknik berdasarkan kesalahan, teknik berdasarkan *stakeholder*, dan teknik dasar yang relatif tergantung dari sistem tersebut.
4. Pengukuran dari uji coba terkait. Didalamnya dijelaskan bahwa pengukuran tersebut dikelompokkan menjadi dua, yakni yang berhubungan dengan evaluasi ketika uji coba dilakukan serta ketika uji coba selesai dilakukan.
5. Proses uji coba itu sendiri, yang berisi tentang pertimbangan praktis dan aktifitas uji coba.

*Black box testing* adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi, dianalogikan seperti kita melihat suatu kotak hitam dan kita hanya bisa melihat penampilan luarnya saja tanpa mengerti tentang apa dibalik bungkus hitamnya. Sama seperti pengujian *black box*, mengevaluasi hanya dari tampilan luarnya (*interface*), yaitu fungsionalitasnya tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detilnya (hanya mengetahui *input* dan *output*).

*Black box testing* adalah metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi yang bertentangan dengan struktur internal atau kerja. Uji kasus dibangun di sekitar spesifikasi dan persyaratan, yakni, aplikasi apa yang seharusnya dilakukan. Menggunakan deskripsi eksternal perangkat lunak, termasuk spesifikasi, persyaratan, dan desain untuk menurunkan uji kasus. Tes ini dapat menjadi fungsional atau non-fungsional, meskipun biasanya fungsional. Perancang uji memilih *input* yang valid dan tidak valid dan menentukan *output* yang benar. Tidak ada pengetahuan tentang struktur internal benda uji itu.

Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak: unit, integrasi, fungsional, sistem dan penerimaan. Ini biasanya terdiri dari kebanyakan jika tidak semua pengujian pada tingkat yang lebih tinggi, tetapi juga bisa mendominasi unit *testing* juga. (Hill 2009)

Pengujian pada *black box* berusaha menemukan kesalahan seperti:

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan kinerja
6. Inisialisasi dan kesalahan terminasi

