

2. LANDASAN TEORI

2.1. Landasan Teori

2.1.1. Audit Sistem Informasi

Sistem informasi adalah sekumpulan prosedur formal dimana data dikumpulkan dan diproses menjadi informasi, dan didistribusikan ke *user* (A. Hall 2001:7).

Sistem informasi sendiri dibedakan menjadi 2, yaitu sistem informasi manajemen dan sistem informasi akuntansi (A. Hall 2001:7).

Sistem informasi manajemen memproses transaksi *nonfinansial*, yang secara normal tidak diproses oleh sistem informasi akuntansi (A. Hall 2001:10). Sedangkan sistem informasi akuntansi memproses transaksi *fmanaal* dan transaksi *nonfinansial* yang secara langsung mempengaruhi pemrosesan transaksi *fmanaal* (A. Hall 2001:9).

Sistem informasi akuntansi adalah komponen organisasi yang dirancang untuk mengolah data keuangan menjadi informasi atau laporan keuangan yang ditujukan kepada pihak eksternal maupun internal perusahaan. Data dapat diolah menjadi informasi dengan cara manual maupun dengan bantuan komputer (Wing 1994:9, Lindrawati 2001:28).

Dari definisi di atas dapat diambil kesimpulan bahwa sistem informasi akuntansi dapat dilakukan dengan 2 cara, yaitu: secara manual dan dengan bantuan komputer.

Dalam hal ini, penulis hanya akan membahas sistem informasi akuntansi yang berbasis komputer, dan untuk selanjutnya, yang dimaksud oleh penulis dengan sistem informasi merujuk pada sistem informasi akuntansi.

Sistem informasi yang baik harus memiliki pengendalian yang bertujuan untuk menghindari kesalahan, kecurangan maupun penyelewengan yang mungkin terjadi.

Dua jenis pengendalian yang digunakan oleh organisasi yang menggunakan sistem informasi berbasis komputer (Wing 1994:102, Lindrawati 2001:33):

1. Pengendalian aplikasi

Pengendalian aplikasi adalah pengendalian yang dirancang untuk menjamin bahwa semua transaksi yang terjadi sudah mendapat otorisasi dan sudah dicatat, diklasifikasi, diproses dan dilaporkan dengan benar. Pengendalian aplikasi juga didesain untuk mencegah, mendeteksi, dan memperbaiki *error* dan *irregularities* dalam transaksi sebagaimana mereka diproses (Romney dan Steinbart 2000:254).

2. Pengendalian umum

Pengendalian umum adalah pengawasan terhadap berbagai prosedur, aktivitas dan aktiva yang tidak secara langsung tercakup dalam pengawasan aplikasi. Pengendalian umum didesain untuk meyakinkan bahwa lingkungan pengendalian organisasi stabil dan ditangani dengan baik untuk mencapai keefektifan pengendalian aplikasi (Romney dan Steinbart 2000:254).

Organisasi harus mengendalikan dan melakukan audit terhadap sistem informasi yang berbasis komputer, karena biaya *error* dan *irregulancies* yang mungkin timbul dalam suatu sistem cukup besar.

Audit sistem informasi adalah suatu fungsi yang telah dikembangkan untuk menilai apakah sistem komputer telah menjaga aset, integritas data, dan melaksanakan tujuan dari organisasi yang hendak dicapai, yaitu efektifitas dan efisiensi (Weber 1999:1). Tujuan ini penting bagi individu yang bekerja baik di dalam atau di luar organisasi.

Untuk menghadapi kompleksitas dari audit sistem informasi, sistem dibagi ke dalam *subsystem*, untuk dinilai dan kemudian menentukan dampak dari tingkat reliabilitas dari masing-masing *subsystem* untuk keseluruhan reliabilitas dari sistem.

Dua sistem utama yang perlu untuk dibagi, sebagai dasar untuk memimpin audit sistem informasi (Weber 1999:38):

1. Sistem manajemen, menyediakan infrastruktur yang stabil dimana sistem informasi dapat dibangun dan dioperasikan dengan dasar hari ke hari. Sistem

manajemen dapat dibagi ke dalam beberapa *subsystem* yang menyajikan *input level information systems management, systems development management, programming management, data administration, quality assurance, security administration, dan operations management* (Weber 1999:39).

2. Sistem aplikasi, mengambil dasar pemrosesan transaksi, pelaporan manajemen, dan dukungan keputusan. Sistem aplikasi dapat dibagi ke dalam *subsystem* yang menyajikan fungsi *boundary, input, communication, processing, database, output* (Weber 1999:39).

Audit sistem informasi melibatkan penilaian reliabilitas pengendalian dalam setiap *subsystem management* dan *subapplication system*.

Selama melaksanakan audit sistem informasi, ada beberapa risiko dimana prosedur audit akan gagal untuk mendeteksi kerugian yang material atau rekening yang salah saji, jika ada. Risiko ini adalah fungsi dari 3 faktor (Weber 1999:44, Mulyadi dan Puradiredja 1998:159):

1. *Inherent risk*.
2. *Control risk*.
3. *Detection risk*.

Karena auditor tidak dapat mempengaruhi *inherent risk* atau *control risk*, maka mereka mengatur sifat dasar dan tingkat prosedur audit yang mereka gunakan untuk mempengaruhi *detection risk* dan keseluruhan tingkat risiko audit.

Ada 5 langkah utama dalam melakukan audit (Weber 1999:47):

1. Perencanaan audit, dimana auditor mencoba untuk memperoleh pemahaman dari pengendalian internal yang digunakan dalam organisasi.
2. Pengujian pengendalian, dimana auditor melakukan pengujian terhadap pengendalian yang penting apakah mereka telah beroperasi secara efektif
3. Pengujian transaksi, dimana auditor melakukan pengujian substantif untuk menilai apakah kerugian yang material atau rekening salah saji telah terjadi atau mungkin terjadi.
4. Pengujian keseimbangan atau hasil keseluruhan, dimana auditor mencari untuk memperoleh bukti yang cukup untuk membuat penilaian terakhir atas tingkat kerugian atau rekening salah saji yang telah terjadi atau mungkin terjadi.

5. Melengkapi audit, dimana auditor memberikan opini apakah kerugian yang material atau rekening salah saji telah terjadi atau mungkin terjadi.

Selama fase pengujian pengendalian, satu keputusan penting yang dibuat auditor adalah apakah pengujian pengendalian dilakukan dengan audit di sekitar komputer atau audit melalui komputer. Mereka mungkin melakukan audit di sekitar komputer jika sistem aplikasi yang digunakan sederhana, *inherent risk* rendah, dan reliabilitas dari pemrosesan sistem internal dapat dengan mudah diduga. Namun mereka harus melakukan audit melalui komputer jika sistem aplikasi dan *inherent risk* tinggi dan sulit untuk menduga pemrosesan internal yang dilakukan oleh sistem.

2.1.2. Pengendalian Aplikasi

Pengendalian sistem aplikasi meyakinkan sistem aplikasi individu untuk menjaga aset, menjaga integritas data, dan mencapai tujuan efektivitas dan efisiensi.

Beberapa tipe dari pengendalian aplikasi adalah sebagai berikut (Weber 1999:365):

1. *Boundary control.*
2. *Input control.*
3. *Communication control.*
4. *Processing control.*
5. *Database control.*
6. *Output control.*

Namun penulis memutuskan hanya akan membahas beberapa bagian tertentu saja dari pengendalian aplikasi, yaitu : pengendalian input dan pengendalian output.

2.1.2.1. Pengendalian Input

Komponen dalam input *subsystem* bertanggung jawab untuk membawa data dan instruksi ke dalam sistem aplikasi (Weber 1999 : 420). Pengendalian input pada tahap ini mencoba untuk meyakinkan bahwa transaksi yang terjadi valid, akurat, dan lengkap (A. Hall 2001:822, Wing 1994:102, Lindrawati

2001:33). Pengendalian input didesain untuk mencegah atau mendeteksi *error* pada tahap input dalam pemrosesan data (Bodnar dan Hopwood 1998:187). Selain itu juga menjamin bahwa data yang ada sesegera mungkin diproses (Wing 1994:102, Lindrawati 2001:33). Pada saat komputer digunakan untuk pemrosesan data, tahap input meliputi konversi dari data transaksi menjadi *machine-readable format* (Bodnar dan Hopwood 1998:187). Secara khusus pengendalian input meliputi item di bawah ini (Weber 1999:421);

1. *Data input methods*

Terdapat 2 jenis prosedur input data (A. Hall 2001:822):

- a. *Direct input (real time)*, menggunakan teknik *real time editing* untuk mengidentifikasi dan memperbaiki *error* dengan segera dan dengan demikian mengurangi jumlah *error* yang masuk ke dalam sistem.
- b. *Source document input (batch)*, lebih membutuhkan keterlibatan manusia dan lebih cenderung pada kesalahan tata usaha, bila dibandingkan dengan prosedur *direct input*.

Beberapa metode input data yang sering digunakan (Weber 1999 : 421):

1. Dalam banyak kasus, data mengenai keadaan atau kejadian akan dimasukkan secara langsung ke dalam sistem dengan menggunakan input *unit*. Input *unit* ini digunakan untuk menerima masukan atau input baik berupa program maupun data (Wing 1994:30, Lindrawati 2001:30). Contoh: *keyboard, mouse, joystick, scanner, barcode reader, voice recognition, trackball, digitizer, light pen*.
2. Data mengenai keadaan atau kejadian juga dapat disimpan ke dalam media sebelum dimasukkan ke dalam sistem. Pada saat data disimpan dalam beberapa media sebelum dimasukkan ke dalam sistem aplikasi, toaik *prerecordmg* atau *postrecordmg* dari suatu keadaan atau kejadian dapat terjadi (Weber 1999:422). *Prerecordmg* terjadi apabila keadaan atau kejadian dicatat dalam media sebelum keberadaan keadaan tersebut dapat dideteksi atau kejadian tersebut teijadi. Sedangkan *postrecordmg* terjadi apabila keadaan atau kejadian dicatat dalam media setelah keberadaan keadaan tersebut telah dideteksi atau suatu kejadian yang telah terjadi.

c. Kombinasi dari metode input data juga dapat digunakan.

2. *Source document design*

Beberapa metode input data menggunakan dokumen sumber untuk mencatat data yang akan dimasukkan ke dalam sistem komputer. Dokumen sumber juga digunakan pada saat terjadi penundaan antara waktu penangkapan data mengenai keadaan atau kejadian dan waktu memasukkan data tersebut ke dalam sistem komputer.

Dari sudut pandang pengendalian, desain yang baik dari dokumen sumber memiliki beberapa tujuan (Weber 1999:423):

- a. Untuk mengurangi kecenderungan kesalahan pencatatan data.
- b. Untuk meningkatkan kecepatan pencatatan data.
- c. Untuk memudahkan pemasukkan data ke dalam komputer.

Beberapa panduan desain yang penting (Weber 1999:424):

- a. *Preprint* apabila memungkinkan.
- b. Menyediakan *titles, headings, notes, dan instructions*.
- c. Menggunakan teknik untuk menekankan dan menyoroti perbedaan.
- d. Menyusunnya agar mudah digunakan.
- e. Menggunakan pendekatan "*caption abovefill-in area*" untuk *captions* dan
- f. Apabila memungkinkan, menyediakan pilihan jawaban atas pertanyaan untuk menghindari kelalaian.
- g. Menggunakan *tick marks* atau *indicator values* untuk mengidentifikasi *field-size errors*.
- h. Mengkombinasikan instruksi dengan pertanyaan.
- i. Jarak *item* sesuai *dalamform*.
- j. Desain untuk mempermudah penguncian.
- k. Pemberian nomor urut pada dokumen sumber.
1. Penyesuaian dengan aturan organisasi.

Selain beberapa panduan desain di atas, organisasi juga harus mengimplementasikan prosedur pengendalian atas dokumen sumber seperti (A.Hall 2001:823):

- a. *Use prenumbered source documents.* Dokumen sumber harus diberi nomor unit dari *printer* dengan urutan nomor yang unik pada setiap dokumen.
 - b. *Use source documents in sequence.* Dokumen sumber harus dibagikan ke *user* dan digunakan secara berurutan.
 - c. *Periodically audit source documents.* Kehilangan dokumen sumber dapat diidentifikasi dengan cara merekonsiliasi urutan nomor dokumen.
3. *Data-entry screen design*

Desain *screen* yang berkualitas tinggi sangat penting untuk meminimalkan input *error* dan untuk mencapai efektivitas dan efisiensi pemasukkan data.

- a. *Screen organization*

Screen data-entry harus didesain, sehingga mereka teratur dan secara simetris seimbang. Jika *screen* digunakan untuk *direct-entry* input dari data, *layout* dari elemen atas *screen* harus mencerminkan cara bagaimana data ini diperoleh selama penangkapan data. Jika *screen* digunakan untuk *source-document data entry*, *screen* harus menjadi gambaran dari dokumen sumber dimana data pertama kali ditangkap dan direkam.

- b. *Caption design*

Captions mengindikasikan sifat dasar dari data yang dimasukkan ke dalam *field* pada *screen*. Desainnya meliputi struktur, ukuran, tipe huruf, intensitas *display*, *format*, *alignment*, *justification*, dan *spacing*. *Captions* harus diganti secara penuh jika *screen* digunakan untuk penangkapan *direct-entry* data. Jika pemasukkan data berdasarkan atas dokumen sumber, bagaimanapun, *captions* dapat disingkat karena *user* dapat menunjuk pada dokumen sumber untuk memperoleh arti penuh dari suatu *caption*.

- c. *Data-entry field design*

Data-entry fields harus segera mengikuti *caption* yang berhubungan dengan mereka, baik pada baris yang sama dalam rangka *field* yang

mengikuti, atau pada beberapa baris segera dibawah *caption*. Pada waktu *screen* penangkapan *direct-entry* data digunakan, bantuan yang lengkap dapat digunakan untuk mengurangi *error* penguncian. Sebaliknya, bila *screen* dokumen sumber digunakan, bantuan yang lengkap tidak dibutuhkan karena operator *keyboard* dapat menunjuk dokumen sumber untuk instruksi yang lengkap.

d. *Tabbing and skipping*

Skipping yang otomatis ke *field* yang baru harus dihindari dalam desain *screen data-entry* karena 2 alasan (Weber 1999:429) :

1. Dengan bentuk *skip* otomatis, operator *keyboard* mungkin membuat suatu *field-size error* yang tetap tidak terdeteksi karena kursor secara sederhana melakukan *skip ke field* yang baru.
2. Dalam banyak aplikasi, *data-entry fields* sering tidak diisi.

e. *Color*

Dalam desain *screen data-entry*, warna dapat digunakan untuk membantu dalam menemukan lokasi dari *caption* tertentu atau data *item*, untuk memisahkan area atas suatu *display*, atau untuk mengindikasikan status yang berubah.

f. *Response time*

Selama pemasukkan data, waktu respons adalah interval yang berlalu diantara pemasukkan data *item* dan indikasi dari sistem bahwa siap untuk menerima data *item* yang baru (Weber 1999:430).

g. *Display rate*

Tingkat *display* adalah tingkat dimana karakter atau gambaran atas *screen* ditayangkan (Weber 1999:430).

h. *Prompting and help facilities*

Fasilitas *prompting* menyediakan saran atau informasi yang segera, mengenai tindakan yang harus diambil oleh *user* pada saat mereka bekerja dengan *screen data-entry*. Fasilitas *help* menyediakan saran atau informasi untuk dilihat mengenai tindakan yang harus diambil oleh *user* pada saat mereka bekerja dengan *screen data-entry*.

4. *Data code controls*

Coding controls adalah pengecekan terhadap integritas kode data yang digunakan dalam pemrosesan (A. Hall 2001:823).

Terdapat 5 tipe dari *data coding errors* (Weber 1999:433 , A. Hall 2001:823):

- a. *Addihon*. Karakter ekstra ditambahkan ke dalam kode.
- b. *Truncation*. Suatu karakter dihilangkan dari kode.
- c. *Transcnptwn*. Karakter yang salah dicatat
- d. *Transposition*. Karakter yang berdekatan dari kode ditukar.
- e. *Double transposition*. Karakter-karakter yang dipisahkan oleh satu atau lebih karakter ditukar.

Liraa faktor yang mempengaruhi frekuensi terjadinya *coding errors* ini adalah (Weber 1999:433):

- a. Panjang kode
- b. Campuran alfabet / *numeric*
- c. Pilihan karakter
- d. Campuran huruf *uppercase / lowercase*
- e. Perkiraan rangkaian karakter

Coding digunakan untuk tujuan mengklasifikasikan data, memasukkan data ke dalam komputer dan mengambil bennacam-macam informasi yang berhubungan dengannya (Wing 1994:30, Lindrawati 2001:32).

Sistem *coding* idealnya raencapai 5 tujuan (Weber 1999:434), yaitu:

- a. *Flexibility*, kode harus mengijinkan kemudahan penambahan item baru atau kategori.
- b. *Meaningfulness*, bila memungkinkan, kode harus mengindikasikan nilai atribut dari entitas tersebut.
- e. *Compactness*, kode harus dapat menyampaikan informasi maksimum, yang disampaikan dengan jumlah karakter minimum.
- d. *Convenience*, kode harus mudah untuk disandikan, dibaca, dan dikunci.
- e. *Evolvability*, bila memungkinkan, kode dapat disesuaikan untuk mengubah permintaan *user*.

Empat tipe sistem *coding* (Weber 1999:434), yaitu:

- a. *Serial code, serial codes* menetapkan nomor (alfabet) yang berurutan pada sebuah entitas, *irrespective* dari atribut entitas.
- b. *Block sequence code, block sequence codes* menetapkan blok angka untuk kategori khusus dari entitas
- c. *Hierarchical code, hierarchical codes* membutuhkan seleksi dari set atribut entitas untuk dikodekan dan diurutkan berdasarkan kepentingan.
- d. *Association code, association codes*, atribut entitas yang akan dikodekan, dipilih dan kode yang unik ditetapkan untuk setiap nilai atribut.

5. *Validation of data input*

Input validation controls diharapkan untuk mendeteksi *error* dalam data transaksi sebelum data diproses (A. Hall 2001:827). Untuk beberapa tingkat tipe dari pengecekan validasi input data yang dilakukan tergantung atas sifat dasar metode input data yang digunakan. Empat tahap pengendalian validasi input, yang dapat dilakukan pada saat input data dikunci di terminal (Weber 1999:443, A.Hall 2001:828):

- a. *Field check*, pengujian validasi yang dilakukan pada *field* tidak tergantung pada *field* yang lain dalam input *record*, atau dalam input *record* yang lain. *Field check* melibatkan prosedur terprogram yang menguji karakteristik data dalam *field* (A. Hall 2001:828). Tipe dari *field check* (A. Hall 2001:829):
 1. *Missing data/blanks*, untuk menilai isi dari *field* atas kehadiran ruang kosong.
 2. *Numeric-alphabetic data checks*, untuk menentukan apakah bentuk yang tepat dari data ada dalam *field*.
 3. *Zero-value checks*, digunakan untuk memverifikasi *field* tertentu yang diisi dengan nol.
 4. *Limit checks*, menentukan apakah nilai dari *field* tidak melebihi batas yang diotorisasi.
 5. *Range checks*, menetapkan batas atas dan bawah untuk nilai data yang dapat diterima.

6. *Validity checks*, membandingkan nilai aktual dalam *field* dengan nilai yang dapat diterima.
 7. *Check digit*, mengidentifikasi *error* dalam *field* dengan cara menguji validitas internal dari kode.
- b. *Record check*, pengujian validasi yang dilakukan *p&d&field* tergantung pada hubungan timbal balik logika *field* dengan *field* yang lain dalam input *record*. Prosedur *record check* mengesahkan seluruh *record* dengan menguji hubungan timbal balik dari nilai *field* yang dimiliki (A. Hall 2001:830). Tipe dari *record check* (A. Hall 2001:830):
1. *Reasonableness*, menentukan apakah nilai dalam *field*, yang telah melewati *limit check* dan *range check*, masuk akal bila dibandingkan dengan/ze/<idatayang lain dalam *record*.
 2. *Sign checks*, menguji untuk melihat apakah tanda dari *field* benar untuk tipe *record* yang diproses.
 3. *Sequence checks*, digunakan untuk menentukan apakah *record* ada yang terlewat.
- c. *File check*, pengujian validasi menilai apakah karakteristik dari *file* yang digunakan selama pemasukkan sama dan sebangun dengan keadaan karakteristik dari *fde*. Tujuan dari *file check* adalah untuk meyakinkan bahwa/;/e yang benar yang akan diproses oleh sistem (A. Hall 2001:830). Tipe dari *file check* (Weber 1999:445, A. Hall 2001:830):
1. *Internal label checks*, memverifikasi apakah/;/e yang diproses benar yang dipanggil oleh program.
 2. *Version checks*. digunakan untuk memverifikasi apakah versi dari *file* yang diproses benar.
 3. *Expiration date check*, mencegah/;/e dari penghapusan sebelum batas waktu kadaluwarsa lewat.
6. *Instruction input*
- Selama memasukkan instruksi, *user* sering mencoba untuk mengkomunikasikan tindakan yang kompleks, yang mereka inginkan agar dilakukan oleh sistem.

Terdapat 6 cara utama dimana instruksi dapat dimasukkan ke dalam sistem informasi (Weber 1999:446):

a. *Menu-driven languages*

Cara temudah bagi *user* untuk menyediakan instruksi bagi sistem aplikasi adalah melalui suatu *menu*. Sistem menyajikan *nser* dengan daftar pilihan. *User* kemudian memilih pilihannya dengan beberapa cara.

b. *Question—answer dialog*

Question-answer dialog digunakan terutama untuk memperoleh input data. *Question-answer dialog* juga digunakan untuk memperoleh input instruksi dalam hubungannya dengan input data.

c. *Command languages*

Command languages membutuhkan *user* untuk menspesifikasi perintah untuk meminta beberapa proses dan suatu set pendapat yang menspesifikasi dengan tepat bagaimana proses tersebut harus dijalankan.

d. *Form-based languages*

Form-based languages membutuhkan *user* untuk menspesifikasi perintah dan data dalam hubungan input atau output.

e. *Natural languages*

Tujuan akhir dari penelitian atas *natural languages interfaces* adalah untuk memungkinkan interaksi *free-form natural languages* relatif agar terjadi diantara *user* dan sistem aplikasi, mungkin melalui alat penghasil / pengenalan pidato.

f. *Direct manipulation languages*

Tiga atribut dari *direct manipulation interfaces* (Weber 1999:452):

1. Jarak penglihatan dari obyek kepentingan.
2. Kecepatan, *reversible*, dan tindakan tambahan.
3. Penggunaan *direct manipulation devices* daripada *command language syntax* untuk memanipulasi obyek kepentingan.

7. *Validation of instruction input*

Tiga tipe pengecekan validasi yang dapat dilakukan melalui pemasukkan transaksi (Weber 1999:453)

a. *Lexical validation*, sistem menilai setiap kata yang dimasukkan oleh *user*.

Tiga tipe kata yang dapat ditemui (Weber 1999:453):

1. *Identifiers (labels, variables)*

2. *Terminals (operators, reserve words)*

3. *Literals (numerical constant, strings)*

b. *Syntactic validation*, sistem membaca *string* dari kata yang diidentifikasi dan disahkan oleh *lexical analyzer* dan mencoba untuk menentukan urutan operasi dari *string* kata yang diharapkan untuk diminta

c. *Semantic validation*, sistem melengkapi analisisnya mengenai arti dari instruksi yang dimasukkan.

2.1.2.2. Pengendalian Output

Output *subsystem* menyediakan fungsi yang menentukan isi data yang akan disediakan untuk *user*, dan cara data tersebut akan diformat dan disajikan untuk *user* dan cara data tersebut dipersiapkan dan dikirim bagi *user* (Weber 1999:615). Pengendalian output meyakinkan bahwa output *subsystem* tidak hilang, tidak mengarah ke tujuan yang salah, tidak dikorupsi, dan tidak melanggar privasi (A. Hall 2001:838, Wing 1994-102, Lindrawati 2001:33).

Komponen utama dari output *subsystem* adalah *software* dan personel yang menentukan isi, format dan batas waktu dari data untuk disediakan bagi *user*, berbagai macam *hardware devices* yang digunakan untuk menghasilkan data output yang telah diformat untuk *user*, dan *hardware, software*, serta personel yang mengirimkan output ke *user* (Weber 1999:615).

Pilihan pengendalian yang digunakan untuk melindungi output *subsystem* dipengaruhi oleh tipe metode pemrosesan yang digunakan (A. Hall 2001:839). Pengendalian yang terdapat dalam output *subsystem* (Weber 1999:616):

1. *Inference controls*

Dalam beberapa kasus, bagaimanapun, *inference control* diharapkan untuk memberikan akses ke data *item* tetapi untuk membatasi tipe informasi yang

dapat diperoleh dari mengakses data *Item*. Dua tipe utama dan *reference controls* (Weber 1999 : 617):

a. *Restriction controls*

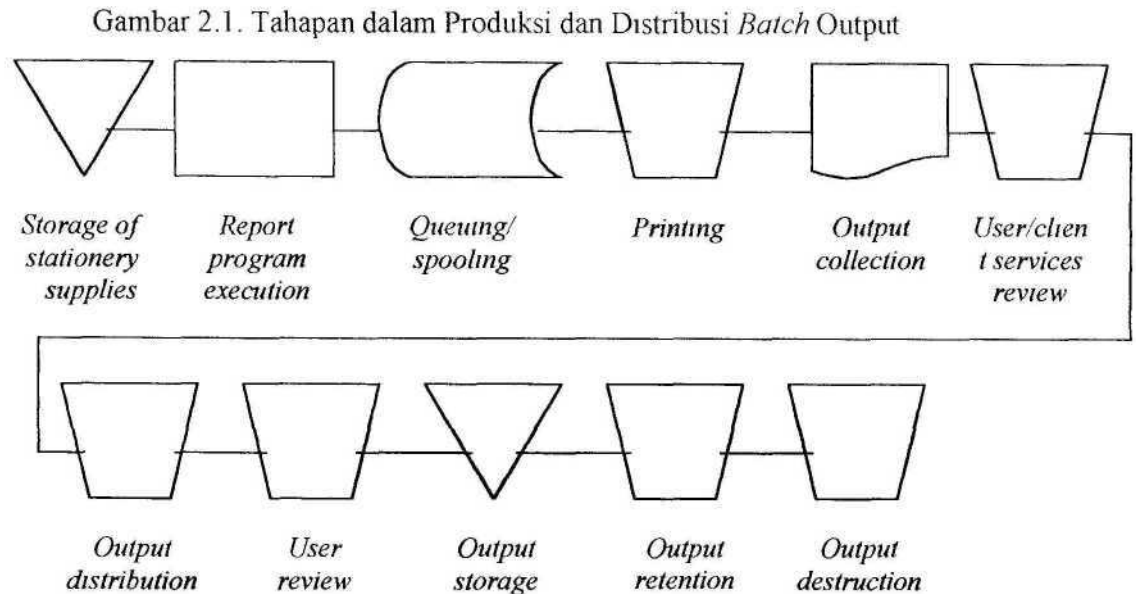
Restriction control membatasi sekumpulan respon yang disediakan bagi *user*, untuk mencoba melindungi kerahasiaan data mengenai seseorang dalam *database*.

b. *Perturbation controls*

Perturbation control dapat dipakai dalam penggunaan *record* sebagai input menjadi fungsi statistik. Alternatif yang lain, *perturbation control* juga dapat dipakai dalam hasil yang diperoleh setelah fungsi statistik dihitung.

2. *Batch output production and distribution controls*

Batch output adalah output yang diproduksi pada beberapa fasilitas operasi dan kemudian didistribusikan ke karyawan atau *user* dari output (Weber 1999:625). *Batch subsystem* biasanya memproduksi output dalam bentuk *hard-copy*, yang membutuhkan keterlibatan *intermediaries* dalam produksi dan distribusinya (A. Hall 2001:839). Pengendalian dalam produksi dan distribusi atas *batch output* dilakukan untuk meyakinkan bahwa output yang akurat, lengkap, dan tepat waktu telah disediakan hanya untuk karyawan dan *user* yang memiliki otorisasi.



(Sumber : Weber, 1999, hal. 626)

Berikut adalah penjelasan mengenai tahapan dalam produksi dan distribusi dari *batch* output (Weber 1999:626):

a. *Stationery supplies storage control*

Untuk beberapa alasan, bagaimanapun, tipe *preprinted stationery* masih digunakan (Weber 1999:626):

1. Untuk beberapa aplikasi, fitur *preprinted* atau kertas khusus untuk pelaksanaan *preprinting*, mengurangi terjadinya pemalsuan.
2. Organisasi mungkin menganggap bahwa *preprinted stationery* masih lebih menarik secara visual daripada *printing constant output* yang dihasilkan melalui laser *printer* di atas *plaw-paper stationery*.
3. Kecepatan *printing* menurun dan biaya *printing* meningkat sebesar jumlah informasi yang harus dicetak oleh laser *printer*.

b. *Report program execution controls*

Tiga hal yang harus diperhatikan dalam hubungannya dengan pelaksanaan program laporan (Weber 1999:627):

1. Hanya orang yang memiliki otorisasi yang mampu melaksanakannya.
2. Batasan tindakan ditentukan untuk *user* yang memiliki otorisasi dari program laporan, harus sesuai dengan kebutuhan mereka.

3. Program laporan yang memproduksi sejumlah besar output harus meliputi fasilitas *checkpoint/restart*.

c. *Output spooling/printer file controls*

Dalam operasi pemrosesan data berskala besar, *output devices*, seperti *line printer*, dapat menjadi timbunan dengan banyak program yang secara bersamaan meminta *resource* yang terbatas ini. Hal ini dapat menyebabkan *bottleneck*, yang dampaknya mengikis *throughput* dari sistem. Aplikasi menunggu untuk mencetak output yang dikerjakan oleh memori komputer dan mencegah aplikasi yang lain untuk memasuki arus pemrosesan. Untuk menghilangkan masalah ini, aplikasi sering didesain untuk mengarahkan output mereka ke *magnetic diskfile* daripada langsung ke *printer*. Ini disebut dengan *spooling* (A. Hall 2001:840). Kemudian pada saat telah siap, maka output akan dicetak.

d. *Printing controls*

Pada saat *printer* tersedia, *print run program* memproduksi *hard copy output* dari *file output*. *Print programs* sering merupakan sistem yang kompleks yang membutuhkan campur tangan operator.

Pengendalian dalam *printing* memiliki 3 tujuan (Weber 1999:628):

1. Meyakinkan bahwa laporan dicetak pada *printer* yang sesuai.
2. Mencegah pihak yang tidak memiliki otorisasi dari *scanming* data yang penting yang dicetak dalam laporan.
3. Meyakinkan bahwa pengendalian yang sesuai dilakukan dalam *printing negotiableform* atau instrumennya.

Bahkan jika laporan dicetak pada lokasi yang aman, masih perlu untuk membatasi *disclosures* dari isi laporan. Jika *preprinted stationery* digunakan, nomor dari *form* yang dihasilkan harus direkonsiliasi dengan nomor dari *form* yang diterima dari *stationery supplies*. Dalam hal ini *prenumbering* dari *form* membantu proses rekonsiliasi. Bahkan jika *preprinted stationery* tidak digunakan, bagaimanapun, jumlah *form* yang dihasilkan masih harus direkonsiliasi dengan beberapa tipe *control totals*.

e. *Report collection controls*

Pada saat output telah diproduksi, maka harus diamankan untuk mencegah kerugian atau pemindahan yang tidak diotorisasi, terutama jika output terdiri dari instrumen yang *negotiable*. Jika perwakilan grup jasa dari *user/client* memiliki tanggung jawab untuk mengumpulkan output, maka mereka harus menjaga *record* dari output yang mereka tangani.

f. *User/client services review controls*

Sebelum output dibagikan ke *user*, perwakilan jasa *user/client* mungkin akan mengeceknya untuk beberapa *error* tertentu (Weber 1999:631):

1. Apakah halaman pada laporan yang dicetak tidak terbaca karena laser printer rendah pada *toner*.
2. Apakah kualitas *daiifilm output* memuaskan.
3. Apakah *tape cartridge* atau *CD-ROM* telah ditandai dengan sesuai.
4. Apakah halaman pada laporan yang dicetak ada yang hilang.
5. Apakah halaman pada laporan telah dicetak miring.

Tipe pengecekan ini memiliki 2 tujuan (Weber 1999 : 631):

1. Menyediakan tingkat jasa yang lebih tinggi dengan fungsi sistem informasi pada *user*.
2. Jika *error* atau *irregularities* dalam output dapat dideteksi setelahnya, akan lebih mudah untuk mendeteksi kemungkinan sumbernya.

g. *Report distribution controls*

Risiko utama yang berhubungan dengan distribusi laporan meliputi laporan yang hilang, dicuri, atau kesalahan tujuan dalam persinggahan ke *user* (A. Hall 2001 : 841) Jika perwakilan grup jasa dari *user client* memiliki tanggung jawab untuk mengumpulkan output, maka mereka harus yakin bahwa output tersebut didistribusikan dengan aman dan tepat pada *user* yang benar. Distribusi dapat terjadi dalam berbagai cara (Weber 1999:631):

1. Output mungkin ditempatkan dalam tempat penyimpanan yang terkunci yang dibersihkan oleh *user* secara periodik.
2. Output mungkin akan dikirimkan secara langsung pada *user*.

3. Output mungkin akan diposkan pada *user*, melalui internet mail atau jasa pos biasa.
4. Output mungkin akan ditangani oleh *user* atau perwakilan *user*, yang mewakili diri mereka sendiri dalam mengumpulkan output.
5. Output mungkin akan dikirimkan melalui jasa kurir.
6. Output mungkin akan diberikan pada organisasi distribusi surat.

Untuk menilai pengendalian pada aktivitas distribusi output, *record* harus dijaga tanggal dan waktunya bilamana output tersebut didistribusikan dan identitas dari orang yang menerima output tersebut. Pengendalian harus dilakukan untuk meyakinkan bahwa output telah dikirimkan dengan tepat waktu.

h. *User output controls*

User harus menyajikan *review* yang sama dengan yang dibawa oleh perwakilan jasa *user/chent* dalam fungsi sistem informasi untuk mendeteksi masalah dengan output. *Error* yang dideteksi oleh *user* harus dilaporkan ke *computer services management* yang sesuai. Beberapa *error* mungkin merupakan gejala dari desain sistem yang tidak sesuai, prosedur yang tidak benar, *error* yang terjadi karena ketidaksengajaan selama *maintenance* sistem, atau akses yang tidak terotorisasi ke *d&tafile* atau program (A. Hall 2001:843). Pengendalian harus dilakukan untuk meyakinkan bahwa *user* melakukan *review* dengan tepat waktu.

i. *Storage controls*

Tiga pengendalian utama yang harus ada dalam hubungannya dengan penyimpanan output (Weber 1999:633):

1. Output harus disimpan dalam lingkungan yang mengijinkannya untuk dilindungi dalam jangka waktu yang dibutuhkan.
2. Output harus disimpan dengan aman.
3. Pengendalian inventori yang sesuai harus dilakukan terhadap penyimpanan output.

j. *Retention control*

Setelah suatu laporan mencapai tujuannya, maka harus disimpan dalam lokasi yang aman sampai periode *retentwn* yang dimilikinya habis. Faktor

yang mempengaruhi jangka waktu penyimpanan laporan dalam bentuk *hard-copy* (A. Hall 2001:843):

1. Permintaan menunit undang-undang yang dispesifikasikan oleh agen pemerintah.
2. Jumlah kopi dari laporan yang ada.
3. Keberadaan laporan yang berupa *magnetic* atau *optical images*, yang berperan sebagai *backup* yang permanen.

Keputusan harus dibuat mengenai berapa lama setiap tipe output akan dijaga. Keputusan ini dapat mempengaruhi pilihan media output dan cara penyimpanan output tersebut. Keputusan atas periode penjagaan juga mempengaruhi bagaimana penyimpanan output harus ditangani.

k. *Destruction controls*

Setelah *retention date* telah lewat, maka laporan harus dimusnahkan dalam cara yang konsisten dengan isinya yang sensitif. Penghancuran laporan dapat dilakukan dengan mudah menggunakan penghancur kertas. Penyimpanan output pada media magnetik juga perlu dihancurkan atau dihapus pada saat *retention date* telah habis.

3. *Batch report design controls*

Elemen yang penting pada pelaksanaan pengendalian produksi dan distribusi yang efektif dalam laporan *batch* output adalah kualitas desainnya. Desain laporan yang baik menyediakan fasilitas aliran laporan yang rapi melalui berbagai fase output yang dilakukan pada tahap sebelumnya.

Informasi pengendalian untuk dicantumkan dalam laporan yang baik:

1. Nama laporan
Tujuan: untuk idenrifikasi laporan secara cepat.
2. Waktu dan tanggal produksi
Tujuan: untuk mencegah kebingungan pada saat laporan diproduksi beberapa kali dalam 1 hari atau untuk beberapa alasan laporan harus diproduksi lagi.

3. Daftar distribusi
Tujuan: menyediakan fasilitas distribusi jumlah salinan laporan yang benar untuk *nser* yang terotorisasi serta membantu meyakinkan bahwa salinan laporan yang tidak terotorisasi tidak diproduksi.
4. *Processing period covered*
Tujuan: agar *user* dapat melihat periode tanggal atau waktu yang telah ditutup oleh laporan.
5. Program yang memproduksi laporan
Tujuan: untuk identifikasi asal sistem atau program.
6. Orang yang dapat dihubungi
Tujuan: mengindikasikan siapa yang harus dikontak pada saat laporan telah diproduksi kurang baik dan saat *errors* dan *irreguhties* ditemukan pada laporan.
7. Klasifikasi keamanan
Tujuan: menyiapkan *operator/user/client services representatives* atas kesensitifitasan data yang terdapat dalam laporan.
8. *Retention date*
Tujuan: mengindikasikan tanggal waktu laporan harus sudah dihancurkan.
9. Metode penghancuran
Tujuan: mengindikasikan prosedur khusus yang harus diikuti untuk menghancurkan laporan.
10. *Page heading*
Tujuan: untuk menunjukkan isi dari halaman laporan dan mungkin klasifikasi keamanan dari data yang terdapat dalam laporan.
11. Nomor halaman
Tujuan: untuk mencegah perpindahan yang tidak terdeteksi dari halaman laporan.
12. *End-of-job marker*
Tujuan: untuk mencegah perpindahan halaman akhir laporan yang tidak terdeteksi.

2.1.3. Siklus Pendapatan

2.1.3.1. Hotel

Definisi hotel dalam Surat Keputusan Menparpostel yaitu: SK KM 34/HK 103/MPPT.87, sebagai berikut:

"Hotel adalah suatu jenis akomodasi yang mempergunakan sebagian/seluruh bangunan untuk menyediakan jasa pelayanan penginapan, makan dan minum, serta jasa lainnya bagi umum, yang dikelola secara komersial serta memenuhi ketentuan persyaratan yang ditetapkan dalam surat keputusan" (Endar Sugiarto 1997:1).

Di hotel pada umumnya terdapat bermacam-macam departemen atau bagian, antara lain (Endar Sugiarto 1997:2):

1. *personnel department*
2. *engineering department*
3. *accounting department*
4. *security department*
5. *F&Bdepartment*
6. *marketing & sales department*
7. *room division department*
8. *front office department*

2.1.3.2. Siklus Pendapatan di Hotel

Siklus pendapatan adalah sekumpulan aktivitas bisnis yang dilakukan dan berhubungan dengan aktivitas pemrosesan yang diasosiasikan sebagai penyediaan barang dan jasa pada *customer* serta pengumpulan pembayaran dari *customer* sehubungan dengan penjualan tersebut (Romney dan Steinbart 2000 : 415).

Hampir semua kegiatan yang ada di kantor depan berhubungan dengan tamu, baik secara langsung maupun tidak langsung. Karena itu, kantor depan hotel memiliki peranan penting dalam operasional sebuah hotel.

Front Office adalah jantung, dan pusat dari segala macam kegiatan para tamu. Semua tamu yang datang dan menginap di hotel pasti mempunyai maksud dan tujuan tertentu, misalnya untuk berbisnis atau sekedar bersantai. Semua

proses tersebut dapat disederhanakan dalam 4 dasar kegiatan yang merupakan siklus dan perputaran kegiatan tamu selama menginap (Endar Sugiarto 1997:2).

1. Sebelum *kedatangan (pre-arrival)* (Endar Sugiarto 1997:6).
 - a. menerima reservasi dengan menggunakan berbagai media
 - b. memasukkan data ke dalam agenda reservasi
 - c. dari agenda reservasi dimasukkan ke dalam form reservasi
 - d. data dari *form* reservasi dimasukkan ke dalam komputer
 - e. diarsipkan
 - f. memberikan *confirmation reservation letter* pada pemesan.

2. Pada saat kedatangan (*arrival*) (Endar Sugiarto 1997:7)

Bila tamu belum membuat reservasi, resepsionis akan memberikan blanko registrasi untuk diisi. Kepada tamu yang pernah menginap sebelumnya tapi pada saat tersebut *walk-in* (tamu *check-in* tanpa membuat pemesanan kamar sebelumnya) cukup dimintai tanda tangan di kartu registrasinya (*registration card*). Data lainnya sudah ada di *guest history* sehingga bisa diisi sendiri oleh petugas. Setelah selesai mengisi blanko registrasi, petugas memeriksanya. Kemudian resepsionis mencarikan kamar sesuai pesanan, menyiapkan kunci kamar, mengisi *guest card*, dan menyiapkan *welcome drink card*.

3. Saat tamu tinggal di hotel (*occupancy*) (Endar Sugiarto 1997:8)

Sebagai koordinator kegiatan, *front office* diharapkan dapat memberikan kepuasan maksimal. Hotel senantiasa berupaya agar tamu melakukan banyak transaksi selama tinggal, seperti makan di *restaurant* yang ada (tidak di luar hotel), menggunakan fasilitas *laundry* dan *dry cleaning*, *business centre*, *sport facilities*, *room service*, telepon, bar, diskotik, dan lain-lain. Transaksi tamu yang paling tinggi selama menginap adalah biaya sewa kamar (*room charges*). Bagi tamu yang *check-in* dengan menggunakan sistem pembayaran *credit card* harus selalu dimonitor *credit limit*-nya, agar tidak terjadi *over limit*. Semua transaksi tamu selama menginap dicatat oleh bagian *accounting (front office cashier)*. Pada malam hari diperiksa oleh *night audit*. Sehingga pada saat *check out* di kemudian harinya, tamu tidak mempunyai masalah dalam pembayaran.

4. Tamu meninggalkan hotel *{departure}* (Endar Sugiarto 1997:8).

Proses *check out* dimulai ketika tamu menghubungi *bell captain* agar mengirim *bellboy* untuk mengangkut barang bawaannya. Lalu pergi ke *FOC* (*Front Office Cashier*) untuk membayar semua rekeningnya, kunci kamar harus diserahkan.

FOC akan menanyakan apakah tamu memakai fasilitas *minibar* sebelum meninggalkan kamar. *FOC* akan mengeceknya dengan menggunakan semacam radio pager yang berhubungan langsung dengan *minibar checker*, Petugas yang dipager harus langsung bergegas ke kamar. Bila terlihat pemakaian *minibar* segera masuk posting ke *FOC*.

Bila hotel sudah *full computerized* termasuk *connected telephon*, maka *minibar checker* dapat melakukan *billing* dari pesawat telepon tersebut, beberapa saat kemudian *bill* sudah bisa dicetak di *FOC*. Tujuan menanyakan adanya pemakaian *minibar* adalah agar tidak terjadi *late charges* (keterlambatan pembebanan biaya) kepada tamu sebelum tamu meninggalkan hotel. kepadanya diserahkan *luggage release* yang berguna untuk mengambil barang bawaan tamu di *Bell Captam Counter*. *Registration card* yang sudah distempel *checked-out* akan diambil oleh karyawan bagian statistik untuk pembuatan *guest history record*,

Form-form yang biasanya ada di *Front Office Department*:

1. *reservation form, form* yang digunakan untuk melakukan reservasi tamu.
2. *reservation confirmation form, form* untuk konfirmasi bagi tamu yang sudah melakukan reservasi.
3. *welcome letter*, kartu selamat datang yang diberikan kepada tamu, yang ditandatangani oleh GM.
4. *miscellaneous charge, form* yang digunakan untuk pengganti nota.
5. *paid out, form* yang digunakan untuk mengeluarkan uang, untuk keperluan tamu.
6. *guest history arrival report*, laporan mengenai sejarah tamu yang akan datang, dari *Front Office* untuk outlet-outlet
7. *group reservation form, form* reservasi untuk tamu rombongan.

8. *group rooming list*, daftar tamu rombongan yang mencantumkan nama tamu dan kamar yang akan dihuninya.
9. *hotel tariff/price list*, daftar harga kamar hotel.
10. *complimentary order form*, form untuk tamu yang dijamin oleh GM Hotel, sehingga tidak perlu bayar.
11. *advance payment receipt*, form untuk deposit *advance petty cash*.
12. *walk-in guest list*, daftar tamu yang *in-house*.
13. *arrival list*, daftar tamu yang akan *check-in*.
14. *departure list*, daftar tamu yang akan *check-out*.
15. *exchange currency voucher*, form yang digunakan apabila tamu membayar menggunakan mata uang asing.
16. *cash receipt*, kuitansi yang legalitasnya berdasarkan kebijakan hotel yang bersangkutan.
17. *credit card slip*, form untuk melakukan *refund* pada kartu kredit.
18. *registration card*, form registrasi untuk diisi oleh tamu pada saat *check-in*.
19. *express check-out form*, form untuk melakukan *check-out* tanpa melalui *Front Office*.
20. *no show list*, laporan mengenai tamu yang sudah reservasi namun tidak jadi *check-in*.
21. *managers daily report*, laporan mengenai transaksi yang terjadi secara garis besar dan *forecast* untuk transaksi.
22. *maintain work order*, form yang menjelaskan semua kerusakan yang terjadi.
23. *guest ledger report*, laporan mengenai semua transaksi yang terjadi dan *posting* yang dilakukan.
24. *room discrepancy report*, laporan mengenai status kamar.
25. *remittance report*, form yang berisi laporan mengenai uang kas kecil (*petty cash*).

2.1.3.3. Hubungan antara *Front Office Department* dengan *Accounting Department*

Bagian *accounting* adalah bagian yang berhubungan dengan masalah keuangan hotel.

Hubungan kerja dengan *accounting* meliputi hal-hal sebagai berikut (Endar Sugiarto 1997 : 11):

1. penagihan kredit tamu yang belum terbayar
2. *petty cash*, kas kecil yang dipegang oleh masing-masing kasir.
3. *safe deposit box*, kotak deposit untuk menyimpan barang berharga milik tamu.
4. *remittance offunds*, laporan mengenai uang kas kecil (*petty cash*).
5. penggajian bulanan yang berhubungan dengan *overtime*.
6. *night audit*, audit malam yang dilakukan untuk memeriksa transaksi setiap hari.
7. *hotel room revenue*, pendapatan hotel yang berasal dari pendapatan kamar.

2.1.3.4. Macam-macam Sistem Pembayaran Tamu

Hal ini ditentukan pada saat melakukan pemesanan kamar atau pada saat *check-in*. Jenis-jenis pembayaran tersebut dapat digolongkan menjadi (Endar Sugiarto 1997:91):

1. Pembayaran dengan *cash money/bank note*
 Untuk melakukan pembayaran ini seorang tamu dapat menggunakan *local currency* atau mata uang lokal yang berlaku di negara tersebut maupun *foreign currency* yaitu mata uang asing yang bisa diterima oleh hotel bersangkutan.
2. Pembayaran dengan *credit card*
Credit card adalah sebuah kartu yang dapat berflingsi sebagai kredit yang dapat digunakan untuk melakukan berbagai transaksi bisnis.
3. Pembayaran dengan *traveller cheque*
Traveller cheque adalah kertas berharga yang dapat difimsikan sebagai alat pembayaran pada sejumlah transaksi bisnis tertentu.
4. Pembayaran dengan *voucher*
Voucher adalah surat berharga yang dikeluarkan oleh pihak *travel agent* (*travel agent voucher*) atau pihak *airline* (*airlme voucher*) atau pihak lain yang dapat dipakai sebagai alat pembayaran pada saat *check-m* di hotel.

5. Sistem pembayaran dengan *guarantee letter*

Guarantee letter adalah sistem pembayaran di hotel dengan menggunakan surat jaminan. Dalam hal ini harus ada perjanjian kerjasama terlebih dahulu diantara perusahaan bersangkutan dengan pihak hotel.

6. Sistem pembayaran *complimentary*

Sistem pembayaran *complimentary* adalah sistem pembayaran gratis/bebas.

2.2. Kajian Peneliti Terdahulu

Penelitian dengan topik sejenis pernah dibahas oleh Tirtohadi (2000) dengan judul Penerapan *Electronic Data Processing Auditing* untuk menilai pengendalian internal atas siklus pembelian pada PT. S. S. Utama di Surabaya. Penelitian ini menekankan pada penerapan *EDP Auditing* dalam menilai pengendalian internal pada siklus pembelian.