

2. LANDASAN TEORI

2.1. Pengertian Sistem Informasi Akuntansi

Menurut Weygandt (1996: 236), sistem informasi akuntansi adalah sistem dari pengumpulan dan pemrosesan data-data transaksi dan menyebarkan informasi keuangan kepada pihak-pihak yang berkepentingan.

Menurut Romney, definisi sistem adalah: *“A system is a set of two or more interrelated components that interact to achieve a goal. Systems are almost always composed of smaller subsystems, each performing a specific function important to and supportive of the larger system of which it is a part.”* (Romney 2000:2)

Sistem adalah sekumpulan dari dua atau lebih komponen yang saling berhubungan dan saling berinteraksi untuk mencapai suatu tujuan. Sistem hampir selalu terdiri dari subsistem yang lebih kecil, dimana masing-masing melakukan suatu fungsi tertentu yang penting untuk mendukung sistem yang lebih besar.

Suatu sistem informasi akuntansi terdiri dari sekumpulan orang, prosedur, dan teknologi informasi. Sistem informasi akuntansi melakukan tiga fungsi penting dalam organisasi:

1. Mengumpulkan dan menyimpan data tentang aktivitas-aktivitas dan transaksi-transaksi sehingga organisasi dapat meninjau apa yang telah terjadi.
2. Memproses data menjadi informasi yang berguna untuk pengambilan keputusan yang memungkinkan manajemen untuk merencanakan, melaksanakan, dan mengontrol aktivitas-aktivitas.
3. Menyediakan kontrol yang memadai untuk melindungi aset organisasi termasuk data. Kontrol ini memastikan bahwa data tersedia saat dibutuhkan dan data tersebut akurat dan dapat diandalkan.

Menurut Romney (2003: 3), sebagian besar perusahaan melakukan banyak transaksi yang serupa dan berulang-ulang. Transaksi-transaksi ini dapat dikelompokkan dalam lima siklus dasar, dimana tiap siklus merupakan sebuah subsistem dasar dalam sistem informasi akuntansi:

1. Siklus pengeluaran, terdiri dari kegiatan-kegiatan yang meliputi pembelian dan pembayaran barang atau jasa yang digunakan oleh organisasi.

2. Siklus produksi, terdiri dari kegiatan-kegiatan yang meliputi merubah bahan mentah dan tenaga kerja menjadi barang jadi. (Catatan: Hanya perusahaan manufaktur yang mempunyai siklus produksi; organisasi ritel, seperti S&S, hanya melakukan pembelian barang jadi untuk kemudian dijual lagi.)
3. Siklus sumber daya manusia/gaji, terdiri dari kegiatan-kegiatan yang meliputi penerimaan dan pembayaran karyawan.
4. Siklus pendapatan, terdiri dari kegiatan-kegiatan yang meliputi penjualan barang atau jasa dan menagih pembayaran atas penjualan tersebut.
5. Siklus keuangan, terdiri dari kegiatan-kegiatan yang meliputi memperoleh dana yang dibutuhkan untuk menjalankan organisasi dan membayar kreditor serta mendistribusikan keuntungan pada para investor.

Dari kelima siklus diatas, yang menjadi fokus pada penelitian ini adalah siklus pendapatan.

2.2. Sistem Informasi Akuntansi Terkomputerisasi

Menurut Gelinas (1996: 17), tahap-tahap pada siklus akuntansi manual adalah mencatat transaksi penjualan pada jurnal penjualan, *posting* jurnal penjualan ke *subsidiary ledger*, *posting* jumlah total jurnal penjualan ke *general ledger*, dan mempersiapkan *trial balance*. Sedangkan tahap-tahap pada siklus akuntansi terotomatisasi (terkomputerisasi) adalah menginput data transaksi penjualan, memproses data transaksi penjualan dan memasukkan dalam *file* piutang, memproses data transaksi penjualan dan memasukkan dalam *file general ledger*, dan mencetak *trial balance* berdasarkan data yang terdapat dalam *file general ledger*.

Menurut Bodnar (2004: 10), istilah sistem informasi akuntansi meliputi penggunaan teknologi informasi untuk menyediakan informasi bagi pengguna. Teknologi informasi ini antara lain komputer, mesin pembaca *bar code*, *scanner*, dan lain sebagainya.

Dari pernyataan diatas dapat disimpulkan bahwa sistem informasi akuntansi terkomputerisasi merupakan sistem informasi akuntansi yang menggunakan teknologi komputer didalamnya untuk membantu kegiatan atau

proses yang terdapat dalam sistem tersebut. Penggunaan komputer yang dimaksud adalah penggunaan komputer mulai dari tahap input data hingga output.

2.3. Pengertian Data dan Informasi

Menurut Romney, definisi dari data dan informasi adalah: *“The term of data refers to any and all of the facts that are collected, stored, and processed by an information system. Information is data that has been organized and processed so that it is meaningful.”* (Romney 2000:13)

Data adalah semua fakta yang didapat, disimpan, dan diproses oleh sebuah sistem informasi. Dan informasi adalah data yang telah diorganisasi dan diproses sehingga dapat berarti/digunakan. Sedangkan definisi data dan informasi menurut Whitten adalah:

“Data are raw facts about the organization and its business transactions. Most data items have little meaning and use by themselves. Information is data that has been refined and organized by processing and purposeful intelligence. The latter, purposeful intelligence, is crucial to the definition—People provide the purpose and the intelligence that produces true information.” (Whitten 2000: 45)

Data adalah fakta-fakta mentah tentang suatu organisasi dan transaksi-transaksi bisnisnya, sementara informasi adalah data yang telah diolah dan diorganisir melalui pemrosesan oleh pihak yang memiliki tujuan tertentu. Dengan kata lain, data merupakan hasil dari kegiatan bisnis. Dan informasi adalah sumber daya yang dibuat dari data untuk memenuhi kebutuhan manajemen dan pengambilan keputusan dari suatu bisnis.

Karakteristik dari informasi yang berguna, menurut Romney (Romney 2000:15) adalah:

- **Relevant.** Informasi dikatakan relevan bila dapat mengurangi ketidakpastian, meningkatkan kemampuan pembuat keputusan untuk membuat prediksi, atau konfirmasi, atau memperbaiki ekspektasi mereka sebelumnya.
- **Reliable.** Informasi dikatakan *reliable* (dapat diandalkan) bila bebas dari kesalahan atau bias dan secara akurat menyajikan kejadian atau kegiatan dari organisasi.

- **Complete.** Informasi dikatakan lengkap bila tidak mengabaikan aspek penting dari kejadian atau kegiatan dasar dari hal yang dinilai.
- **Timely.** Informasi dikatakan tepat waktu bila informasi tersebut disajikan pada waktu pembuat keputusan tersebut membutuhkan.
- **Understandable.** Informasi dikatakan *understandable* (dapat dimengerti) bila informasi tersebut disajikan dalam format (susunan) yang berguna dan dapat dimengerti/jelas.
- **Verifiable.** Informasi dikatakan *verifiable* (dapat diverifikasi) bila dua orang yang berpengetahuan bertindak secara independen dan masing-masing menghasilkan informasi yang sama.

2.4. Kegiatan Pemrosesan Data

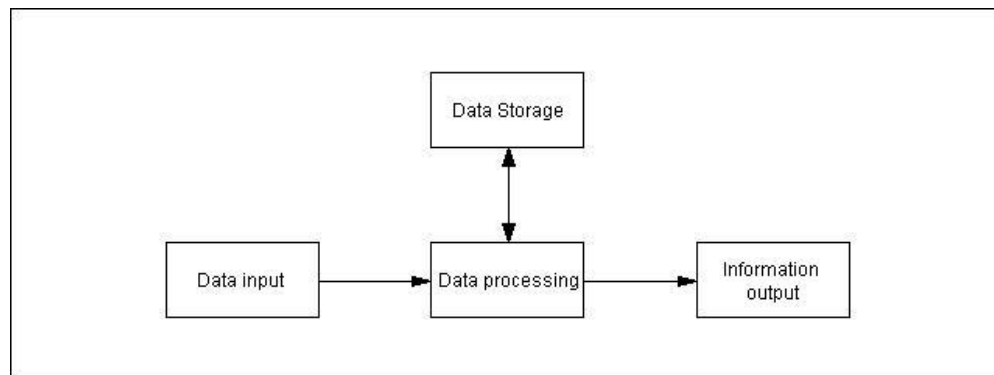
Menurut Romney, siklus *data processing* ada empat tahap, yaitu *data input*, *data storage*, *data processing*, dan *information output*, seperti yang terlihat pada gambar 2.1. (Romney 2000:93)

Penjelasan dari keempat tahap tersebut adalah sebagai berikut:

a. *Data Input*

Tahap memasukkan data merupakan saat dimana data transaksi dicatat pada suatu lembar dokumen yang disebut *source document*. Contoh dari *source document* internal antara lain *sales orders*, *purchase requisitions*, dan *receiving reports*.

Gambar 2.1. Empat Tahap dari Data Processing Cycle



(Sumber: Romney, 2000, hal. 93)

Sedangkan contoh dari *source document* eksternal antara lain *invoice* dari *suppliers* dan *check and remittance advices* dari pelanggan. Untuk memudahkan proses data selanjutnya, *data input* membutuhkan persiapan atau pengolahan sebagai berikut:

- Klasifikasi dengan memberikan kode identitas (*account number*, *department number*, dsb.) pada *data records* berdasarkan sistem yang telah ditetapkan, seperti *chart of accounts*.
- Verifikasi untuk memastikan akurasi data. Mencegah kesalahan pada saat memasukkan data lebih murah dan efisien daripada mendeteksi dan memperbaiki kesalahan data yang sudah dimasukkan dalam sistem.
- Pengiriman dari satu lokasi ke lokasi lainnya. Sebagai contoh, kebanyakan ATM bank mencatat dan mengirimkan data transaksi ke kantor utama untuk diproses.

b. *Data Storage*

Data perusahaan merupakan sumber daya yang penting. Namun, keberadaannya saja tidak menjamin kegunaan dari data tersebut. Sebuah organisasi harus memiliki data yang siap dan mudah diakses agar dapat berfungsi sebagaimana mestinya.

Komputer menyimpan data dengan mengorganisasi data dari unit yang lebih kecil menjadi lebih besar dan berarti. Hirarki penyimpanan data ini dimulai dari *field*, yaitu elemen yang paling kecil dari data. Beberapa *field* dikelompokkan menjadi sebuah *record* yang kemudian nantinya dikelompokkan hingga menjadi sebuah *file*. Beberapa *file* dikelompokkan menjadi sebuah *database*, yang merupakan elemen data yang paling besar.

Ada dua jenis *file* yang digunakan untuk menyimpan data, yaitu *master file* dan *transaction file*. *Master file* berisi semua data yang dibutuhkan oleh perusahaan tentang suatu hal, dimana secara konseptual serupa dengan *ledger* pada sistem manual. Sebagai contoh, catatan dalam sebuah *master file* piutang meliputi informasi seperti nama pelanggan, alamat, dan batas kredit. *Transaction file* berisi data tentang jenis transaksi tertentu, dimana secara konseptual serupa dengan jurnal pada sistem manual. Sebagai contoh, *file* transaksi penjualan yang berisi data tentang semua penjualan pada suatu periode.

c. *Data Processing*

Kegiatan proses data pada umumnya adalah pemeliharaan data, dimana data transaksi secara berkala diproses untuk melakukan *update* pada data yang telah ada. Ada empat jenis pemeliharaan data yang umum dilakukan, yaitu:

- *Additions*, memasukkan data baru ke dalam *master file*.
- *Deletions*, menghapus data yang ada dalam *master file*.
- *Updates*, merevisi neraca dalam *master file*, umumnya dengan menambah atau mengurangi suatu jumlah yang tersimpan dalam sebuah dokumen transaksi.
- *Changes*, memodifikasi nilai data yang terdapat pada *fields* dalam *master file*, seperti mengubah *credit ratings* dan alamat pelanggan.

d. *Information Output*

Information output merupakan tahap akhir dari siklus pemrosesan data. Hal-hal yang dilakukan pada tahap ini antara lain: menentukan *format* laporan (berbentuk dokumen, laporan, atau berupa respon atas pertanyaan), membuat laporan sesuai dengan tujuan dan kegunaannya (untuk pihak eksternal atau internal perusahaan).

2.5. Pengertian Siklus Pendapatan

Menurut Romney, definisi siklus pendapatan adalah: “*The revenue cycle is a recurring set of business activities and related information processing operations associated with providing goods and services to customers and collecting cash in payment for those sales.*” (Romney 2000:415)

Siklus pendapatan adalah suatu kegiatan yang berulang-ulang dari aktivitas bisnis dan operasi proses informasi terkait yang berhubungan dengan penyediaan barang dan jasa untuk pelanggan dan menagih uang pembayaran untuk penjualan tersebut. Salah satu tujuan dari sistem informasi akuntansi dalam siklus pendapatan adalah untuk mendukung performa kegiatan bisnis organisasi dengan memproses data transaksi secara efisien.

2.5.1. Aktivitas Dalam Siklus Pendapatan

Menurut Romney (Romney 2000: 415), ada empat kegiatan dasar bisnis dalam siklus pendapatan: *sales order entry*, *shipping*, *billing*, dan *cash collection*.

Dibawah ini merupakan penjelasan dari keempat kegiatan dasar tersebut.

2.5.1.1. Sales Order Entry

Menurut Romney (Romney 2000: 416), fungsi dari *sales order entry* meliputi mengumpulkan dan memproses *customer orders*. Aktivitas ini dilakukan oleh departemen *sales order* yang memberikan laporan ke *vice president of marketing*.

Fungsi dari *sales order entry* adalah untuk mendapatkan informasi yang diperlukan tentang tersedianya persediaan dan status kredit pelanggan dari kontrol persediaan dan fungsi akuntansi masing-masing. Keputusan yang berkaitan dengan kebijakan kredit, termasuk persetujuan kredit untuk pelanggan baru dan peningkatan *credit limit* pelanggan dibuat oleh manajer kredit, yang nantinya dilaporkan pada bendahara dan puncaknya pada *vice president* keuangan. *Sales Order Entry* mempunyai 3 aktivitas utama, yaitu:

a. Responding to Customer Inquiries

Pertanyaan pelanggan dapat langsung ditangani oleh *sales order department* atau oleh *customer service department*, yang nantinya memberikan laporan pada *vice president of marketing*.

b. Credit Approval.

Penjualan kredit seharusnya terlebih dahulu disetujui sebelum diproses. Penentuan *credit limit* bagi tiap pelanggan ditetapkan berdasarkan sejarah/catatan kredit dimasa lalu dan kemampuan untuk melakukan pembayaran.

c. Checking Inventory Availability.

Jumlah persediaan yang ada harus selalu diperiksa, sehingga pelanggan dapat diinformasikan tentang ketersediaan barang dan perkiraan tanggal pengiriman. Fungsi ini penting karena bila catatan persediaan tidak akurat dan *up to date*, pelanggan mungkin akan merasa kecewa ketika terjadi keterlambatan dalam memenuhi permintaan mereka.

Kegiatan *sales order* diawali adanya pemesanan barang oleh Pelanggan. Setelah *sales order* diterima, pegawai bagian *sales order* akan memasukkan beberapa data transaksi, antara lain: *customer account number*, *salesperson number*, *product numbers and quantities*, dan tanggal pengiriman yang

diinginkan. *Customer number* digunakan untuk mengakses data sebelumnya yang terdapat dalam *customer file*. Begitu pula dengan *product number*, digunakan untuk mengakses data yang telah terdapat dalam *inventory file*. Pada titik ini, beberapa *edit checks* akan dilakukan untuk memastikan akurasi input yang meliputi:

- *Validity checks* terhadap catatan pelanggan dan *inventory item numbers* yang dilakukan dengan mencocokkan mereka dengan informasi dalam *customer and inventory master files* masing-masing.
- *Redundant data checks* terhadap nomor akun pelanggan dan nama dan *item numbers* dan deskripsi yang telah dibuat untuk mendeteksi kasus yang valid tetapi salah dalam memasukkan nomor akun atau item.
- *Field checks* dilakukan untuk memastikan bahwa hanya data numerik yang dimasukkan dalam *numeric fields* dan untuk menghindari kesalahan proses yang dilakukan kemudian.
- *A reasonableness test* dilakukan untuk menguji akurasi dari kuantitas yang dipesan. Tes ini membandingkan kuantitas yang dipesan dengan jumlah standar yang biasanya dipesan seperti yang diindikasikan dalam *master inventory item record*.
- *Range checks* atas permintaan dan tanggal pengiriman dilakukan untuk memverifikasi *feasibility*.
- *A completeness test* dijalankan untuk memverifikasi bahwa tiap catatan transaksi mengandung semua *data items* yang pantas/seharusnya.

Setelah melalui semua pengujian diatas, jumlah permintaan ditotal, kemudian dicocokkan dengan *credit limit* pelanggan. Apabila jumlah tersebut melebihi *credit limit* yang ada, maka akan dibuat *credit rejection report*, yang nantinya akan diberikan pada *credit manager* untuk dipertimbangkan apakah permintaan tersebut ditolak atau diberi kenaikan batas kredit. Selanjutnya sistem akan memeriksa apakah persediaan mencukupi untuk menerima permintaan.

Dokumen utama yang dihasilkan oleh *sales order entry* adalah *sales order*, yang berisi nomor barang, jumlah barang, harga, dan syarat penjualan. Satu *copy* dari *sales order* diarsipkan pada departemen *sales order*, dimana nantinya dapat digunakan sebagai referensi dalam merespon pertanyaan pelanggan. *Copy* lainnya

diberikan pada pelanggan sebagai pemberitahuan penerimaan permintaan. Selain itu *copy* dari *sales order* juga diberikan pada departemen *billing* dan *shipping*. *Copy* yang diberikan pada departemen *shipping* seringkali dilampirkan pada paket yang berfungsi sebagai *packing slip* yang mengidentifikasi muatan dalam pengiriman. Dokumen lain yang dihasilkan oleh *sales order entry* adalah *picking ticket*, yang memberi perintah bagian persediaan untuk mengirim barang ke bagian pengiriman.

2.5.1.2. Shipping

Menurut Romney (Romney 2000: 425), langkah kedua dalam siklus pendapatan ini meliputi memenuhi permintaan pelanggan dan mengirim barang yang diinginkan. Untuk itu departemen pengiriman bertanggungjawab dalam pengiriman barang pada pelanggan.

Picking ticket yang dicetak oleh *sales order entry* memicu proses pengiriman. Pekerja gudang menggunakan *picking ticket* sebagai dasar untuk memindahkan persediaan ke bagian pengiriman. Setelah petugas pengiriman menghitung barang yang dikirim dari gudang, akan dilakukan beberapa *edit checks* yang sama dengan yang ada pada tahap *sales order entry*. Proses ini akan melakukan *update* kuantitas persediaan dalam *inventory master file* dan menghasilkan beberapa *copy* dari *bill of lading*. *Bill of lading* adalah sebuah *legal contract* yang menjelaskan pertanggungjawaban untuk barang dalam *transit*. Dokumen ini berisi: pengirim, sumber, tujuan, dan beberapa perintah khusus pengiriman, selain itu dokumen ini juga menjelaskan siapa yang harus membayar biaya pengiriman. Bila pelanggan yang membayar biaya pengiriman, maka *copy* dari *bill of lading* ini juga berfungsi sebagai *freight bill*, yang mengindikasikan jumlah yang harus dibayar oleh pelanggan pada pengirim. Dalam beberapa kasus, *freight bill* merupakan dokumen yang terpisah. Satu *copy* dari *bill of lading* disimpan oleh departemen pengiriman untuk melacak dan mengkonfirmasi transfer barang pada pengirim. *Copy* lain dari *bill of lading* diberikan pada departemen *billing* untuk mengindikasikan bahwa barang telah dikirimkan dan bahwa sebuah *invoice* harus disiapkan dan dikirimkan. Pengirim juga menyimpan satu *copy* dari *bill of lading*.

2.5.1.3. *Billing and Account Receivable*

Menurut Romney (Romney 2000:429), pada langkah ketiga dalam siklus pendapatan terdapat dua aktivitas yang dilakukan, yaitu pembuatan faktur pelanggan dan mengurus laporan/catatan pelanggan. Proses ini dilakukan oleh departemen piutang/*billing*, yang nantinya memberikan laporan pada direktur akuntansi, dan *controller*.

a. *Types of Billing Systems.*

Ada dua jenis sistem penagihan, yaitu *postbilling system* dan *prebilling system*. *Postbilling system* adalah sistem penagihan yang mempersiapkan *invoice* setelah ada konfirmasi bahwa barang telah dikirimkan. *Postbilling system* umumnya digunakan pada perusahaan manufaktur, dimana sering terjadi jeda waktu antara penerimaan permintaan dan pengiriman barang.

Prebilling system adalah sistem penagihan yang mempersiapkan *invoice* (tetapi tidak langsung dikirim) segera setelah permintaan diterima. *File* persediaan, piutang, dan *general ledger* juga diupdate pada saat itu. *Prebilling system* juga mengeliminasi kebutuhan untuk beberapa dokumen. Contohnya, *invoice* juga berperan sebagai *sales order*.

Ketika sebuah *copy* dari *bill of lading* diterima dari departemen pengiriman, petugas *billing* mencocokkannya dengan *copy* dari *sales order* yang berkaitan yang berasal dari *sales order entry*. Petugas *billing* kemudian membuat *invoice* dengan memasukkan informasi tentang nomor dan kuantitas barang yang dikirimkan. Setelah itu sistem melakukan *edit check* atas data yang dimasukkan, seperti menguji validitas nomor barang yang dimasukkan dan membandingkan kuantitas dengan yang terdaftar pada *open sales order files*.

Setelah *edit check*, beberapa operasi seperti berikut dilakukan:

1. Catatan baru dibuat dalam *sales invoice file*, dan beberapa *copy* dari *sales invoice* dicetak.
2. *Customer master file* diakses, dan akun pelanggan didebet untuk jumlah dari penjualan.
3. *Sales order* yang terbuka ditutup pada *sales order history file*.

4. Terakhir, setelah semua *invoice* telah diproses, sistem membuat sebuah ringkasan *journal entry* yang merefleksikan jumlah total yang dimasukkan pada penjualan dan akun piutang pada *general ledger*.

b. *Method for Maintaining Accounts Receivable.*

Ada dua metode dalam mengurus piutang:

- *Open-invoice method.*

Pada *open-invoice method*, pelanggan membayar menurut tiap *invoice*. Biasanya dua *copy* dari *invoice* diberikan pada pelanggan, dan nantinya diminta mengembalikan satu *copy* bersama dengan pembayaran. *Copy* ini merupakan *turnaround document* yang disebut *remittance advice*. Pembayaran pelanggan kemudian digunakan pada *invoice* tertentu. Pernyataan bulanan yang dihasilkan dalam *open-invoice method* mencatat semua *outstanding invoices* dan disimpan secara individual.

- *Balance-forward method.*

Pada *balance-forward method*, pelanggan membayar menurut jumlah yang ada pada *statement* bulanan, dan bukan menurut *invoice* secara individual. *Remittance* digunakan pada *total amount balance*, bukan pada *invoice* tertentu. *Statement* bulanan biasanya memperlihatkan *beginning balance*, *all current charges*, dan *current balance due*, tetapi *statement* tidak menyimpan *invoice* secara individual.

2.5.1.4. *Cash Collection*

Menurut Romney (Romney 2000:436), langkah keempat dalam siklus pendapatan adalah *cash collection*. Ada dua departemen yang terlibat dalam aktivitas ini. Pertama adalah kasir, yang melapor pada bendahara, mengurus pengiriman/pembayaran uang dan mendepositokannya ke bank; kedua adalah fungsi piutang, yang melapor pada *controller* tentang laporan kredit pelanggan untuk pembayaran yang telah diterima.

Proses *cash collection* dimulai ketika petugas menerima cek dari para pelanggan. Cek tersebut kemudian disahkan untuk dimasukkan ke salah satu akun bank milik perusahaan. Sementara itu petugas juga mempersiapkan *remittance list*, sebuah dokumen yang mendaftar semua cek yang diterima. Semua cek dan

satu *copy* dari *remittance list* diberikan pada kasir, yang kemudian mempersiapkannya untuk didepositokan. *Copy* kedua dari *remittance list* diberikan pada departemen audit internal, yang nantinya digunakan untuk melakukan rekonsiliasi pada *bank statements*. *Copy* ketiga dari *remittance list* dan *remittance advices* diberikan pada bagian piutang yang digunakan untuk *update* akun pelanggan.

2.5.2. Pengendalian pada Siklus Pendapatan

Menurut Romney (Romney 2000:439), fungsi kedua dari sistem informasi akuntansi adalah memberikan kontrol yang memadai untuk memastikan bahwa tujuan berikut ini tercapai:

1. Semua transaksi diotorisasikan dengan baik
2. Semua transaksi yang tercatat adalah valid.
3. Semua transaksi valid yang diotorisasikan dicatat.
4. Semua transaksi dicatat secara akurat.
5. Pengamanan aset (kas, persediaan, dan data) dari kehilangan dan pencurian.
6. Kegiatan bisnis dijalankan dengan efektif dan efisien.

2.6. Sistem Penjualan Kredit

Menurut Mulyadi (1997: 204), kegiatan penjualan terdiri dari transaksi penjualan barang atau jasa, baik secara kredit maupun secara tunai. Dalam transaksi penjualan kredit, jika order dari pelanggan telah dipenuhi dengan pengiriman barang atau penyerahan jasa, untuk jangka waktu tertentu perusahaan memiliki piutang kepada pelanggannya. Kegiatan penjualan secara kredit ini ditangani oleh perusahaan melalui sistem penjualan kredit.

Penjualan kredit dilaksanakan oleh perusahaan dengan cara mengirimkan barang sesuai dengan order yang diterima dari pembeli dan untuk jangka waktu tertentu perusahaan mempunyai tagihan kepada pembeli tersebut. Untuk menghindari tidak tertagihnya piutang, setiap penjualan kredit yang pertama kepada sorang pembeli selalu didahului dengan analisis terhadap dapat atau tidaknya pembeli tersebut diberi kredit.

Fungsi yang terkait dalam sistem penjualan kredit adalah fungsi penjualan, fungsi kredit, fungsi gudang, fungsi pengiriman, fungsi penagihan, dan fungsi akuntansi. Informasi yang umumnya diperlukan manajemen dari kegiatan penjualan kredit adalah:

- a. Jumlah pendapatan penjualan menurut jenis produk atau kelompok produk selama jangka waktu tertentu.
- b. Jumlah piutang kepada setiap debitur dari transaksi penjualan kredit.
- c. Jumlah harga pokok produk yang dijual selama jangka waktu tertentu.
- d. Nama dan alamat pembeli.
- e. Kuantitas produk yang dijual.
- f. Nama wiraniaga yang melakukan penjualan.
- g. Otorisasi pejabat yang berwenang.

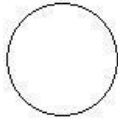
2.7. Data Flow Diagram (DFD)

Definisi *data flow diagram* (DFD) adalah: “A data flow diagram (DFD) graphically describes the flow of data within an organization. It is used to document existing systems and to plan and design new ones.” (Romney 2000:58) Menurut Romney, DFD menjelaskan melalui gambar tentang arus data dalam sebuah organisasi. DFD digunakan untuk mendokumentasi sistem yang ada dan untuk merencanakan dan mendesain sistem yang baru.

DFD dapat disubdivisikan menjadi level yang lebih rendah secara berurutan untuk menggambarkan detail yang lebih banyak. Level tertinggi pada DFD disebut sebagai *context diagram*. Sebuah context diagram menyajikan secara ringkas gambaran level dari sebuah sistem. (Romney 2000:63)

DFD terdiri dari empat elemen dasar, dimana masing-masing direpresentasikan dalam suatu simbol yang dijelaskan dalam gambar 2.2.

Gambar 2.2. Simbol-simbol pada Data Flow Diagram (DFD)

Symbol	Name	Explanation
	Data sources and destinations	The people and organizations that send data to and receive data from the system are represented by square boxes. Data destinations are also referred to as data sinks.
	Data flows	The flow of data into or out of a process is represented by curved or straight lines with arrows.
	Transformation processes	The processes that transform data from inputs to outputs are represented by circles. They are often referred to as bubbles.
	Data stores	The storage of data is represented by two horizontal lines.

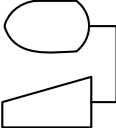
(Sumber: Romney, 2000, hal. 59)

2.8. Flowchart

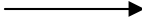
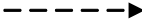
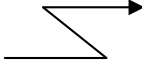
Menurut Romney, “A *flowchart* is an analytical technique used to describe some aspect of an information system in a clear, concise, and logical manner”. (Romney 2000:65)

Jadi, *flowchart* adalah sebuah teknik analitis yang digunakan untuk menggambarkan aspek-aspek dari sebuah sistem informasi secara jelas, singkat, dan logis. Dalam *flowchart* digunakan simbol-simbol standar untuk menggambarkan prosedur-prosedur proses transaksi pada sebuah perusahaan. Simbol-simbol tersebut dijelaskan pada gambar 2.3.

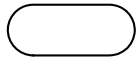
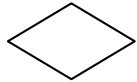
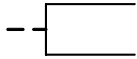
Gambar 2.3. Simbol Umum dalam *Flowchart*

Symbol	Name	Explanation
<i>Input/Output Symbols</i>		
	Document	A document or report; the document may be prepared by hand or printed by a computer
	Multiple copies of one document	Illustrated by overlapping the document symbol and printing the document number on the face of the document in the upper-right corner
	Input/output; Journal/ledger	Any function of input or output on a program flowchart Represents accounting journals and ledgers on document flowchart
	Display	Information displayed by an online output device such as a CRT terminal or personal computer monitor
	On-line keying	Data entry by on-line device such as a CRT terminal or personal computer
	CRT terminal, personal computer	The display and on-line keying symbols are used together to represent CRT terminals and personal computers
	Transmittal tape	Manually prepared control totals; used for control purposes to compare to computer-generated totals
<i>Processing Symbols</i>		
	Computer processing	A computer-performed processing function; usually results in a change in data or information
	Manual operation	A processing operation performed manually
	Auxiliary operation	A processing function done by a device that is not a computer
	Off-line keying operation	An operation utilizing an off-line keying device (e.g., key to disk, cash register)

Gambar 2.3. Simbol Umum dalam *Flowchart* (sambungan)

Symbol	Name	Explanation
<i>Storage Symbols</i>		
	Magnetic disk	Data stored permanently on a magnetic disk; used for master files
	Magnetic tape	Data stored on a magnetic tape
	Magnetic diskette	Data stored on a diskette
	On-line storage	Data stored in a temporary on-line file in a direct-access medium such as a disk
	File	File of documents manually stored and retrieved; inscribed letter indicates file-ordering sequence: N = numerically A = alphabetically D = by date
<i>Flow and Miscellaneous Symbols</i>		
	Document or processing flow	Direction of processing or document flow; normal flow is down and to the right
	Data/information flow	Direction of data/information flow; often used to show data copied from one document to another
	Communications link	Transmission of data from one location to another via communication lines
	On-page connector	Connects the processing flow on the same page; its usage avoids connecting lines crisscrossing a page
	Off-page connector	An entry from, or an exit to, another page
	Flow of goods	Physical movement of goods; used primarily with document flowcharts

Gambar 2.3. Simbol Umum dalam *Flowchart* (sambungan)

Symbol	Name	Explanation
<i>Flow and Miscellaneous Symbols (continued)</i>		
	Terminal	A beginning, end, or point of interruption in a process or program; also used to indicate an external party
	Decision	A decision-making step; used in a computer program flowchart to show branching to alternative paths
	Annotation	Addition of descriptive comments or explanatory notes as clarification

(Sumber: Romney 2000:69)

2.9. *Resources, Events, and Agents (REA) Diagram*

Resources, Events, and Agents (REA) data model adalah sebuah perangkat *modeling* konseptual yang secara khusus didesain untuk membuat struktur untuk mendesain AIS *data base*. REA data model membuat struktur dengan dua cara: (1) dengan mengidentifikasi entitas apa yang seharusnya dimasukkan dalam AIS data base; dan (2) dengan menjelaskan terlebih dahulu bagaimana membuat struktur atas entitas-entitas dalam AIS data base. (Romney 2000:182)

Tipe-tipe entitas dalam REA *data model* adalah:

- *Resources*, didefinisikan sebagai segala sesuatu yang memiliki nilai ekonomi terhadap organisasi. Misalnya kas, persediaan, peralatan, suplai, gudang, pabrik, dan tanah.
- *Events*, merupakan berbagai macam kegiatan bisnis perusahaan dimana manajemen ingin mendapatkan informasi untuk merencanakan atau mengendalikan tujuan. Beberapa *events* mempengaruhi secara langsung kuantitas dari *resources*.
- *Agents*, merupakan sekumpulan orang dan organisasi yang berpartisipasi dalam *events* dan tentang informasi apa yang diinginkan untuk perencanaan, pengendalian, dan evaluasi tujuan.

Berikut ini merupakan elemen-elemen yang terdapat pada REA *data model*:

a. *Entities.*

Sebuah entitas adalah sesuatu yang dapat diidentifikasi dalam “lingkungan kerja” user. Contoh dari entitas adalah EMPLOYEE, CUSTOMER 12345, dan SALES-ORDER 1000. Kumpulan entitas yang dibuat menurut struktur atau susunan dari entitas tersebut disebut sebagai *entity class*. Sedangkan penyajian dari entitas tertentu, seperti CUSTOMER 12345 disebut sebagai *entity instance*, dimana entitas tersebut digambarkan oleh nilai atribut entitas. (Kroenke 2002:52)

b. *Attributes.*

Entitas memiliki atribut, yaitu sesuatu yang menggambarkan karakteristik dari entitas. Contoh dari atribut adalah EmployeeName, DateOfHire, dan JobSkillCode. (Kroenke 2002:52)

c. *Identifiers.*

Entity instance memiliki *identifiers*, yaitu atribut yang menyebutkan atau mengidentifikasikan *entity instance*. Sebagai contoh, *EMPLOYEE instances* dapat diidentifikasi dengan *SocialSecurityNumber*, *EmployeeNumber*, atau *EmployeeName*. *EMPLOYEE instances* tidak mungkin diidentifikasi dengan atribut seperti *Salary* atau *DateOfHire* karena umumnya atribut ini tidak digunakan untuk penyebutan nama. *Identifier* dari sebuah entitas terdiri dari satu atau lebih atribut entitas. Sebuah *identifier* dapat bersifat unik maupun non-unik. Jika bersifat unik, nilainya akan mengidentifikasikan satu, dan hanya satu, *entity instance*. *Identifier* yang bersifat unik disebut juga dengan *primary key*. Jika bersifat non-unik, nilainya akan mengidentifikasikan sekumpulan *entity instance*. *Identifier* yang bersifat non-unik disebut juga dengan *secondary key*. Apabila *identifier* pada *record* berasal dari *file* lain dalam *database*, *identifier* ini disebut sebagai *foreign key*. *EmployeeNumber* merupakan *unique identifier*, sedangkan *EmployeeName* merupakan *nonunique identifier* (karena ada kemungkinan terdapat nama yang sama pada dua atau lebih karyawan). *Identifiers* yang terdiri dari dua atau lebih atribut disebut *composite identifiers*. Contohnya adalah {AreaCode, LocalNumber}, {ProjectName, TaskName}, dan {FirstName, LastName, PhoneExtension}. (Kroenke 2002:53)

d. *Relationships*.

Entitas-entitas dapat dihubungkan antara satu dengan lainnya dalam *relationships*. Seperti pada entitas, *E-R model* berisikan keduanya, *relationship classes* dan *relationship instances*. *Relationship classes* merupakan hubungan antar *entity classes*, dan *relationship instances* merupakan hubungan antar *entity instances*. *Relationship* dapat memiliki atribut.

Sebuah *relationship class* dapat melibatkan banyak *entity classes*. Jumlah *entity classes* dalam *relationship* merupakan *degree* (tingkat) dari *relationship*. *Relationship degree 2* merupakan yang paling banyak dijumpai dan sering disebut dengan istilah *binary relationship*. (Kroenke 2002:53)

Ada tiga jenis *relationship* antar entitas yang mungkin terjadi, tergantung pada *maximum cardinality* dari tiap entitas yang bersangkutan:

1. *One to one relationship*, terjadi ketika *maximum cardinality* dari tiap entitas adalah 1.
2. *One to many relationship*, terjadi ketika *maximum cardinality* dari suatu entitas adalah 1 dan *maximum cardinality* dari entitas lain adalah N.
3. *Many to many relationship*, terjadi ketika *maximum cardinality* dari kedua entitas adalah N.

(Romney 2000:190)

e. *Cardinality*.

Cardinality dari suatu *relationship* mengindikasikan seberapa banyak keberadaan dari suatu *entity* yang dapat dihubungkan pada suatu *entity* lain dalam sebuah *relationship*. *Cardinality* seringkali diekspresikan dalam sepasang angka. Angka pertama disebut *minimum cardinality*, sedangkan angka kedua disebut *maximum cardinality*. *Minimum cardinality* mengindikasikan angka terkecil dari baris yang dapat dihubungkan dalam *relationship*. *Maximum cardinality* mengindikasikan angka terbesar dari baris yang dapat dihubungkan dalam *relationship*. (Romney 2000:188)

2.10. Sistem Manajemen Database

Menurut James A. Hall (2001: 454), *database mangement system* (DBMS) adalah sistem *software* khusus yang diprogram untuk mengetahui

elemen-elemen data mana saja yang dapat diakses oleh *user*. Program dari *user* mengirimkan permintaan atas data kepada DBMS, dimana validitas dan hak untuk masuk ke dalam *database* tergantung pada tingkatan wewenang dari pengguna. Jika permintaan data berasal dari *user* yang tidak memiliki wewenang untuk mengakses data tersebut, maka permintaan tersebut akan ditolak. Tujuan dari DBMS adalah untuk menyediakan pengendalian akses ke *database*.

Secara lebih spesifik, DBMS ditujukan untuk menciptakan kondisi pengelolaan data yang berciri sebagai berikut:

- a. Memudahkan pengaksesan data.
- b. Mengeliminasi redundansi dan inkonsistensi data.
- c. Mengisolasi data untuk distandarisasikan.
- d. Mudah digunakan untuk banyak pemakai.
- e. Mengamankan data.
- f. Menjaga integritas data.
- g. Menciptakan independensi data.

2.11. Pengendalian Aplikasi

Menurut Romney: “The primary objective of application controls is to ensure the accuracy of a specific application’s inputs, files, programs, and outputs, rather than to control the computer system in general.” (Romney 2000:308)

Jadi, tujuan utama dari pengendalian aplikasi adalah untuk memastikan akurasi dari kegiatan aplikasi tertentu, seperti input, *file*, program, dan output, dan bukan untuk mengontrol sistem komputer secara umum. Pengendalian aplikasi ini dibagi menjadi 5 kategori yang penjelasannya terdapat dibawah ini.

2.11.1. Source Data Controls

Berikut merupakan beberapa pengendalian terhadap sumber data yang mengatur akurasi, validitas, dan kelengkapan input:

- *Key verification*. *Key verification* merupakan pengendalian sumber data yang mahal dan hanya digunakan untuk input yang krusial seperti nomor pelanggan, jumlah dan kuantitas permintaan.

- *Check digit verification*. Contoh dari pengendalian ini adalah dengan penerapan penghitungan *digit* dari nomor ID.
- *Prenumbered forms sequence test*. Ketika *prenumbered forms* digunakan, sistem akan mengidentifikasi dan melaporkan kehilangan atau menduplikasi nomor *form*.
- *Turnaround documents*. Karena turnaround documents merupakan output sistem yang dapat dibaca oleh mesin, mereka lebih akurat daripada catatan yang dimasukkan secara manual.
- *Authorization*. Data seharusnya tidak dimasukkan ke dalam sistem kecuali data tersebut sudah diotorisasikan secara baik.
- *Cancellation of documents*. Dokumen yang telah dimasukkan ke dalam sistem seharusnya tidak dapat dikeluarkan sehingga mereka tidak dapat dimasukkan kembali ke dalam sistem baik secara sengaja maupun tidak sengaja.
- *Visual scanning*. *Source documents* seharusnya diperiksa ulang untuk kelayakan dan kepatutan sebelum dimasukkan ke dalam sistem.
- *Data control function*. Ketika data diterima untuk diproses, petugas pengendalian data mencatat data masuk, memeriksa *user authorizations*, memonitor proses data, merekonsiliasi semua pengendalian setelah tiap langkah proses, mengingatkan pengguna dari input yang salah, dan memasukkan kembali koreksi dari semua kesalahan.

(Romney 2000:308)

2.11.2. *Input Validation Routines*

Input validation routines merupakan program yang memeriksa validitas dan akurasi dari pemasukkan data ketika dimasukkan ke dalam sistem. Program ini disebut *edit programs*, dan pemeriksaan akurasi yang dilakukan disebut *edit checks*. Pada proses *on-line*, *edit check* dilakukan selama proses pemasukkan data sumber. Beberapa *edit checks* digunakan dalam *input validation routines*:

- *Sequence check* menguji apakah *batch* dari input data berada pada urutan yang tepat secara numerik atau alfabetik.

- *Field check* menentukan apakah karakter dalam sebuah *field* diketik dengan tepat.
- *Sign check* menentukan apakah data dalam sebuah *field* sudah berisi tanda aritmatika yang benar.
- *Validity check* membandingkan nomor ID atau kode transaksi dengan yang telah disahkan.
- *Limit check* menguji jumlah numerik untuk memastikan tidak melebihi atau kurang dari batas yang telah ditentukan.
- *Range check* hampir sama dengan *limit check* kecuali bahwa *range check* mempunyai batas atas dan bawah.
- *Reasonableness test* menentukan kebenaran logis dari input dan data yang disimpan.
- *Redundant data check* menggunakan dua pengidentifikasi dalam tiap transaksi yang dicatat untuk mengkonfirmasi bahwa catatan *data base* yang benar telah diupdate.
- *Capacity check* dilakukan untuk memastikan bahwa data dapat dimuat dalam *field* yang bersangkutan.

(Romney 2000:309)

2.11.3. *Data Processing and File Maintenance Controls*

Berikut merupakan pengendalian umum yang membantu menjaga akurasi dan kelengkapan proses data dan penyimpanan data:

- *Data currency check*. Pemeriksaan ini dilakukan untuk menentukan data mana yang tidak diperlukan karena sudah tidak berlaku.
- *Default values*. Merupakan nilai standar yang ada pada *field* (sebelum dilakukan input).
- *Data matching*. Merupakan kegiatan pencocokkan dua atau lebih jenis data.
- *Exception reporting*. Merupakan kegiatan pelaporan atas segala kondisi yang tidak benar.

- *External data reconsilliation*. Merupakan kegiatan rekonsiliasi terhadap jumlah total yang ada dalam *data base* dengan jumlah yang terdapat pada pihak eksternal.
- *Control account reconcilliation*. Merupakan kegiatan rekonsiliasi atas *general ledger* terhadap semua *subsidiary account*.
- *File security*. Merupakan kegiatan pencatatan *file* yang masuk dan keluar, pelaksanaan mekanisme proteksi penulisan, dan *backup copy* dari *file* yang disimpan.
- *File conversion control*. Merupakan kegiatan pengawasan terhadap pemindahan data dari *file* yang lama ke *file* yang baru. Kegiatan ini juga memastikan tidak adanya kesalahan selama pemindahan data.

(Romney 2000:312)

Dari kelima kategori pengendalian aplikasi diatas, yang akan dipakai pada penelitian ini adalah *input validation routines*.

2.11.4. Output Controls

Menurut Romney, fungsi pengendalian data seharusnya menghasilkan output yang layak dalam susunan yang tepat dan seharusnya merekonsiliasi semua pengendalian input dan output yang berkaitan. Pengendalian data juga bertanggungjawab atas pendistribusian output komputer ke departemen pengguna yang tepat. Tindakan khusus seharusnya dilakukan dalam menangani cek serta dokumen atau laporan sensitif lainnya. *User* bertanggungjawab terhadap kelengkapan dan akurasi dari semua output komputer yang mereka terima. Alat penghancur dokumen dapat digunakan untuk menghancurkan data penting yang sudah tidak terpakai seperti daftar pelanggan, data penelitian, dan daftar gaji.

(Romney 2000:313)

2.12. System Development Life Cycle (SDLC)

System Development Life Cycle (SDLC) adalah suatu proses untuk merancang dan mengimplementasikan sebuah sistem baru pada suatu organisasi.

(Romney 2000:617)

Berikut merupakan tiga dari lima langkah proses dalam SDLC yang nantinya digunakan pada penelitian ini:

1. *System analysis*. Dalam tahap ini semua informasi yang diperlukan untuk membeli atau mengembangkan sebuah sistem baru dikumpulkan. Permintaan untuk pengembangan sistem diprioritaskan pada penggunaan secara maksimal sumber daya pengembangan yang terbatas. Bila sebuah proyek lulus uji awal, sistem yang telah ada diteliti untuk menentukan sifat dan bidang dari proyek dan memahami kekuatan serta kelemahannya. Setelah itu sebuah penyelidikan dilakukan untuk menentukan kelayakannya.
Jika sistem yang diajukan dianggap layak, informasi yang dibutuhkan oleh pengguna dan manajer sistem diidentifikasi dan didokumentasikan. Hal ini merupakan hal yang paling penting dari analisa sistem, dimana kebutuhan ini digunakan untuk mengembangkan dan mendokumentasikan kebutuhan sistem. Kebutuhan sistem digunakan untuk memilih atau mengembangkan sebuah sistem baru. Sebuah laporan analisa sistem dibuat dan diajukan pada komite sistem informasi.
2. *Conceptual design*. Pada tahap ini perusahaan memutuskan bagaimana memenuhi kebutuhan pengguna. Tugas pertama adalah mengidentifikasi dan mengevaluasi alternatif rancangan yang tepat. Ada banyak cara untuk mendapatkan sebuah sistem yang baru, termasuk membeli *software*, mengembangkan sendiri sistem dengan bantuan staf sistem informasi atau pengguna, atau menyerahkan pekerjaan pengembangan ini pada orang lain. Spesifikasi yang mendetail menentukan sistem seperti apa yang harus dikembangkan, meliputi apa yang dikerjakan sistem dan bagaimana mengendalikannya.
3. *Physical design*. Pada tahap ini perusahaan mentranslasikan *user-oriented requirements* dari *conceptual design* ke dalam detail yang lebih spesifik yang digunakan untuk pengkodean dan menguji program komputer. Dokumen input dan output didesain, program komputer ditulis, files dibuat, prosedur dikembangkan, dan pengendalian dibuat dalam sistem yang baru.

(Romney 2000:617)