BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

A. Definisi Sistem

Dalam kehidupan sehari-hari di era teknologi dan komputerisasi saat ini, kita sering mendengar istilah sistem. Misalkan sistem komputer, sistem operasi, sistem informasi, sistem geografis, sistem akademis, dan lainnya. Bahkan disadari maupun tidak, kita sering menggunakan atau memanfaatkan layanan yang diberikan oleh sistem tersebut. Kemudian muncul sebuah pertanyaan, apakah yang dimaksud dengan sistem?.

Menurut Pratama (2014:7) sistem didefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas secara bersama-sama. Secara garis besar, sebuah sistem informasi terdiri atas tiga komponen utama. Ketiga komponen tersebut mencakup *software*, *hardware*, dan *brainware*. Ketiga komponen ini saling berkaitan satu sama lain.

B. Definisi Informasi

Menurut Pratama (2014:9) informasi merupakan hasil pengolahan data dari satu atau berbagai sumber, yang kemudian diolah, sehingga memberikan nilai, arti, dan manfaat. Proses pengolahan ini memerlukan teknologi. Berbicara mengenai teknologi memang tidak harus selalu berkaitan dengan komputer,

namun komputer sendiri merupakan salah satu bentuk teknologi. Dengan kata lain, alat tulis dan mesin ketik pun dapat dimasukkan sebagai salah satu teknologi yang digunakan selain komputer dan jaringan komputer.

Pada proses pengolahan data, untuk dapat menghasilkan informasi, juga dilakukan proses verifikasi secara akurat, spesifik dan tepat waktu. Hal ini penting agar informasi dapat memberikan nilai dan pemahaman kepada pengguna. Pengguna dalam hal ini mencakup pembaca, pendengar, penonton bergantung pada bagaimana cara pengguna tersebut menikmati sajian informasi dan melalui media apa informasi tersebut disajikan.

C. Sistem Informasi

Menurut Sutanta (2011:16) dalam arti yang luas sistem informasi dapat dipahami sebagai sekumpulan subsistem yang saling berhubungan, berkumpul bersama-sama dan membentuk satu kesatuan, saling berinteraksi dan bekerja sama antara bagian satu dengan lainnya dengan cara-cara tertentu untuk melakukan fungsi pengolahan data, menerima masukan (*input*) berupa data-data, kemudian mengolahnya (*processing*), dan menghasilkan keluaran (*output*) berupa informasi sebagai dasar bagi pengambilan keputusan yang berguna dan mempunyai nilai nyata yang dapat dirasakan akibatnya baik pada saat itu juga maupun di masa mendatang.

Menurut Sutanta (2011:16) berdasarkan komponen fisik penyusunannya, sistem informasi terdiri atas komponen berikut :

a. Perangkat Keras (*Hardware*)

Perangkat keras dalam sistem informasi meliputi perangkat-perangkat yang digunakan oleh sistem komputer untuk masukan dan keluaran (*input/output device*), *memory*, *modem*, pengolah (*processor*) dan periferal lainnya.

b. Perangkat Lunak (*Software*)

Perangkat lunak dalam sistem informasi adalah berupa program-program komputer yang meliputi sistem operasi (*Operating System/OS*), bahasa pemrograman (*programming language*), dan program-program aplikasi (*application*).

c. Berkas Basis Data (*File*)

Berkas merupakan sekumpulan data dalam basis data yang disimpan dengan cara-cara tertentu sehingga dapat digunakan kembali dengan mudah dan cepat.

d. Prosedur (*Procedure*)

Prosedur meliputi prosedur pengoperasian untuk sistem informasi, manual, dan dokumen-dokumen yang memuat aturan-aturan yang berhubungan dengan sistem informasi dan lainnya.

e. Manusia (*Brainware*)

Manusia yang terlibat dalam suatu sistem informasi meliputi *operator*, *programmer*, *system analyst*, manajer sistem informasi, manajer pada tingkat operasional, manajer pada tingkat manajerial, manajer pada tingkat strategis,

teknisi, administrator basis data (*Database Administrator/DBA*), serta individu lain yang terlibat di dalamnya.

D. UML (*Unified Modelling Language*)

UML menyediakan serangkaian gambar dan diagram yang sangat baik. Beberapa diagram memfokuskan diri pada ketangguhan teori *object-oriented* dan sebagian lagi memfokuskan pada detail rancangan dan konstruksi. Semuanya dimaksudkan sebagai sarana komunikasi antar *team programmer* maupun dengan pengguna. Sistem yang kita buat tidak selalu menggambarkan aktivitas internal, hubungan dengan *supplier* dan pelanggan yang bersifat eksternal harus diperhatikan.

1. Diagram Aktivitas (*Activity Diagram*)

Menurut Widodo dan Herlawati (2011:144) diagram aktivitas lebih memfokuskan diri pada eksekusi dan alur sistem dari pada bagaimana sistem itu dirakit. Diagram ini tidak hanya memodelkan *software* melainkan memodelkan model bisnis juga. Diagram aktivitas menunjukkan aktivitas sistem dalam bentuk kumpulan aksi-aksi.

2. *Use Case*

Menurut Pilone dalam Widodo dan Herlawati (2011:21) *use case* menggambarkan fungsi tertentu dalam suatu sistem berupa komponen, kejadian atau kelas. Sedangkan menurut Whitten dalam Widodo dan Herlawati (2011:21) mengartikan *use case* sebagai urutan langkah-langkah yang secara tindakan saling

terkait (skenario), baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu bisnis tunggal.

Use case sangat menentukan karakteristik sistem yang kita buat, oleh karena itu menurut Chonoles dalam Widodo dan Herlawati (2011:22) menawarkan cara untuk menghasilkan *use case* yang baik, yakni :

- a) Pilihlah nama yang baik
- b) Ilustrasikan perilaku dengan lengkap
- c) Identifikasi perilaku dengan lengkap
- d) Menyediakan *use case* lawan (*inverse*)
- e) Batasi *use case* hingga satu perilaku saja
- f) Nyatakan *use case* dari sudut pandang aktor

3. **Relasi antar usecase / aktor**

Pada diagram *use case*, relasi digambarkan sebagai sebuah garis antara dua simbol. Pemaknaan relasi berbeda-beda tergantung bagaimana garis tersebut digambar dan tipe simbol apa yang digunakan untuk menghubungkan garis tersebut. Relasi yang digunakan UML 2.0 adalah generalisasi, inklusi dan ekstensi.

1. Generalisasi (*Generalisation*)

Menurut Widodo dan Herlawati (2011:24) generalisasi pada aktor dan *use case* dimaksudkan untuk menyederhanakan model dengan cara menarik keluar

sifat-sifat pada aktor-aktor maupun *use case* – *use case* yang sejenis. Menurut Chonoles dalam Widodo dan Herlawati (2011:24) memberikan cara untuk mengetahui kapan dibutuhkan generalisasi berdasarkan tujuan, yakni:

a. Mekanisme berbeda dengan satu tujuan yang sama (Generalisasi *Use Case*)

Jika ada lebih dari satu alternatif teknik dan cara agar aktor dapat mencapai tujuannya, tentu akan diperoleh penggunaan bersama (*sharring*) seperti: peralatan pendukung, bisnis rules, validasi data dan sebagainya. Dengan generalisasi kita dapat membuat penggunaan bersama itu secara eksplisit dan mengurangi duplikasi dengan cara membuat satu *use case* baru yang mengakomodasi penggunaan bersama itu.

b. Agen berbeda dengan satu tujuan yang sama (Generalisasi Aktor)

Jika lebih dari satu aktor mencoba membangun satu tujuan yang sama kita dapat membuat generalisasi antar aktor tersebut. Yang perlu diperhatikan adalah kemungkinan aktor-aktor yang terlibat memiliki hak akses, kemampuan dan user interface yang berbeda sehingga tidak dimungkinkan untuk dibentuknya generalisasi. Biasanya generalisasi berlaku untuk aktor yang merupakan intermediary/penengah untuk yang lainnya.

c. Ekstensi (*Extension*)

Menurut Whitten dalam Widodo dan Herlawati (2011:28) mengartikan ekstensi pada *use case* adalah *use case* yang lebih kompleks untuk menyederhanakan masalah orisinal dan karena itu memperluas fungsinya. Hubungan antara ekstensi *use case* dan *use case* yang diperluas disebut

extend relationship, diberi simbol “<<extend>>” dan hubungannya berupa garis putus-putus berpanah terbuka. Sebelum dilakukan ekstensi alangkah baiknya kita analisa terlebih dahulu *use case* dasarnya, karena *use case* dasar harus sudah lengkap.

Dalam aplikasi perbankan, misalnya kartu kredit bank Niaga, selain kita buat *use case* “Pembukaan Kartu Kredit” yang menyatakan bagaimana kartu kredit baru dibuka di bank tersebut, kita juga membuat *use case* “Tambah Member” karena bank tersebut juga menyediakan kartu kredit khusus untuk anak dan istri si-pengguna kartu baru itu.

Use case yang akan diekstensi menurut (Philone,2005:7.3) harus dengan jelas mendefinisikan titik ekstensinya (*ekstension point*). Karena UML 2.0 belum menspesifikasikan titik ini, maka kita dapat membuatnya dengan teks bebas.

d. Inklusi (*Inclusion*)

Use case dasar yang akan diinklusi tidak lengkap, berbeda dengan *use case* dasar yang akan diekstensi. Sehingga *use case* inklusi bukan merupakan *use case* optional dan tidak boleh tidak dijalankan. Simbol hubungan inklusi adalah garis putus-putus dengan anak panah terbuka dan diberi keterangan “<<include>>”.

Tabel 2.1
Perbedaan antara Inklusi Dan Ekstensi

Inklusi	Ekstensi
<i>Use case</i> terpanggil (<i>included use case</i>) selalu diperlukan oleh <i>use case</i> dasar.	<i>Use case</i> ekstensi tidak selalu dibutuhkan oleh <i>use case</i> dasar.
Yang memutuskan kapan dipanggilnya <i>use case included</i> adalah <i>use case</i> dasar.	Yang memutuskan kapan dipanggilnya <i>use case</i> ekstensi adalah <i>use case</i> ekstensi itu sendiri.
Panah hubungan dari <i>use case</i> dasar ke <i>use case included</i> .	Panah hubungan dari <i>use case</i> ekstensi ke <i>use case</i> dasar.

4. ***Black Box Testing***

Menurut Rizky (2011:264) *black box testing* adalah tipe *testing* yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Pada jenis *black box testing*, perangkat lunak tersebut akan dieksekusi kemudian berusaha dites apakah telah memenuhi kebutuhan pengguna yang didefinisikan pada saat awal tanpa harus membongkar listing programnya.

Beberapa keuntungan yang diperoleh dari jenis *testing* ini antara lain :

1. Anggota tim *tester* tidak harus dari seorang yang memiliki kemampuan teknis di bidang pemrograman.
2. Kesalahan dari perangkat lunak ataupun *bug* seringkali ditemukan oleh komponen *tester* yang berasal dari pengguna.
3. Hasil dari *black box testing* dapat memperjelas kontradiksi ataupun kerancuan yang mungkin timbul dari eksekusi sebuah perangkat lunak.

4. Proses *testing* dapat dilakukan lebih cepat dibandingkan *white box testing*.

E. Model Data *Entity Relationship*

Menurut Sutanta (2011:91) *Entity Relationship Model/ER-M* merupakan suatu model data yang dikembangkan berdasarkan objek. ER-M digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara logik. ER-M didasarkan pada suatu persepsi bahwa *real world* terdiri atas obyek-obyek dasar tersebut. ER-M digambarkan dalam bentuk diagram yang disebut ER (ER Diagram/ERD). Untuk menggambarkan ERD digunakan simbol-simbol grafis tertentu.

F. Komponen *Entity Relationship Diagram (ERD)*

Sebuah diagram ER/ERD tersusun atas tiga komponen, yaitu entitas, atribut dan kerelasian antar entitas. Secara garis besar, entitas merupakan obyek dasar yang terlibat dalam sistem.

1. Entitas (*Entity*)

Menurut (Sutanta, 2011:91) entitas menunjukkan obyek-obyek dasar yang terkait di dalam sistem. Obyek dasar dapat berupa orang, atau hal yang keterangannya perlu disimpan di dalam basis data. Untuk menggambarkan sebuah entitas digunakan aturan sebagai berikut (Sutanta, 2004):

- a. Entitas dinyatakan dengan simbol persegi panjang.
- b. Nama entitas dituliskan di dalam simbol persegi panjang.
- c. Nama entitas berupa kata benda, tunggal.
- d. Nama entitas sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

2. Atribut (*Attribute*)

Atribut sering pula disebut sebagai properti (*property*), Menurut (Sutanta, 2011:98) merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan dalam basis data. Atribut berfungsi sebagai penjelas pada sebuah entitas (Sutanta, 2004). Untuk menggambarkan atribut digunakan aturan sebagai berikut (Sutanta, 2004):

- a. Atribut dinyatakan dengan simbol elips.
- b. Nama atribut dituliskan di dalam simbol elips.
- c. Nama atribut berupa kata benda, tunggal
- d. Nama atribut sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.
- e. Atribut dihubungkan dengan entitas yang bersesuaian dengan menggunakan sebuah garis (seyogianya menggunakan garis lurus, namun dalam kondisi yang tidak memungkinkan dapat juga tidak menggunakan garis lurus).

Menurut (Sutanta, 2011:100) atribut pada sebuah entitas dapat diklasifikasikan dalam dua kelompok, yaitu :

1. Atribut sederhana (*simple attribute*), yaitu jika atribut berisi sebuah komponen nilai/elementer. Contoh atribut sederhana dan nilai atribut dalam entitas Mahasiswa adalah :

a. Kode_Angkatan : 2002 (angkatan 2002)

b. Kode_Program_Studi : 01 (Program Studi Teknik
Informatika).

c. Kode_Jenjang_Studi : 08 (Jenjang Studi Sarjana)

2. Atribut komposit (*composite attribute*), yaitu jika atribut berisi lebih dari sebuah komponen nilai. Contoh atribut komposit dan nilai atribut dalam entitas Mahasiswa adalah :

a. Nama_Mahasiswa : Mawar Menur Melati

Terdiri atas komponen nilai,

Nama depan : Mawar

Nama tengah : Menur

Nama Akhir : Melati

b. Tanggal_Lahir : 01-01-1980

Terdiri atas komponen nilai,

Tanggal : 01
Bulan : 01
Tahun : 1980

G. Kerelasiaan Antar Entitas (*Relationship*)

Menurut (Sutanta, 2011:101) kerelasiaan antar entitas mendefinisikan hubungan antara dua buah entitas. Kerelasiaan adalah kejadian atau transaksi yang terjadi di antara dua buah entitas yang keterangannya perlu disimpan dalam basis data (Martin:1975). Kejadian atau transaksi yang tidak perlu disimpan dalam basis data (sekali pun benar-benar terjadi) bukan termasuk kerelasiaan.

Aturan penggambaran kerelasiaan antar entitas adalah sebagai berikut (Sutanta, 2004:102).

1. Kerelasiaan dinyatakan dengan simbol belah ketupat.
2. Nama kerelasiaan dituliskan di dalam simbol belah ketupat.
3. Kerelasiaan menghubungkan dua entitas.
4. Nama kerelasiaan berupa kata kerja aktif (diawali dengan awalan me-), tunggal
5. Nama kerelasiaan sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

Menurut (Sutanta, 2011:102) kerelasiaan antar entitas dikelompokkan dalam tiga jenis yaitu :

1. Kereliasian jenis 1-ke-1/satu ke satu (*one to one*)
2. Kereliasian jenis n-ke-1/banyak ke satu (*many to one*) atau 1-ke-n/satu ke banyak (*one to many*)
3. Kereliasian jenis n-ke-n/banyak ke banyak (*many to many*)

H. Menggambar ERD

Menurut Indrajani (2015:17) *Entity Relational* (ER) Modelling adalah sebuah pendekatan *top-bottom* dalam perancangan basis data yang dimulai dengan mengidentifikasi data-data terpenting yang disebut dengan entitas dan hubungan antara entitas-entitas tersebut yang digambarkan dalam suatu model.

Menurut (Sutanta, 2011:113) untuk menggambarkan ERD secara lengkap dapat dilakukan dengan mengikuti serangkaian langkah berikut :

- a. Identifikasikan setiap entitas yang terlibat.
- b. Identifikasikan setiap atribut pada setiap entitas.
- c. Identifikasikan setiap kereliasian berikut jenisnya yang terjadi di antara entitas.
- d. Gambarkan simbol-simbol entitas, atribut, dan kereliasian antar entitas sedemikian sehingga simbol kereliasian dapat digambarkan dengan jelas dan tidak saling bertabrakan.
- e. Cek ERD yang terbentuk, dalam hal :
 1. Kelengkapan entitas

2. Kelengkapan atribut
3. Kelengkapan kerelasiaan antar entitas
4. Jenis kerelasiaan antar entitas

Untuk subsistem yang sederhana, seluruh rangkaian langkah tersebut akan menghasilkan sebuah diagram yang sangat lengkap, yang membuat komponen entitas, atribut dan kerelasiaan antar entitas. Permasalahan menggambar ERD mungkin akan dijumpai ketika sistem mempunyai sejumlah entitas, atribut dan kerelasiaan yang sangat banyak dan kerelasiaan-kerelasiaan antar entitas sangat kompleks sehingga sangat sulit jika digambarkan secara keseluruhan. Diagram yang terbentuk mungkin menjadi sangat kompleks dan ruwet sehingga menjadi sulit untuk dipahami. Jika demikian maka ada tiga pilihan yang dapat digunakan, yaitu :

- a. Gambarkan ERD yang hanya memuat komponen entitas dan kerelasiaan antar entitas saja. Selanjutnya rincian atribut pada setiap entitas dapat ditampilkan secara terpisah, yang disusun dalam bentuk naratif atau tabel. Cara ini paling banyak digunakan karena mengurangi kompleksitas ERD.
- b. Gambarkan ERD secara terpisah-pisah, bagian demi bagian atau sepotong demi sepotong, dimana masing-masing bagian/potongan memuat komponen entitas, atribut dan kerelasiaan antar entitas untuk suatu bagian yang lebih kecil. Misal, dalam subsistem pengolahan data akademik dapat terdiri atas bagian/potongan ERD untuk kelompok Mahasiswa, Dosen, dan Mata_Kuliah. Setiap bagian tersebut memuat entitas, atribut dan kerelasiaan yang terkait saja. Cara ini relatif

jarang digunakan, tetapi dalam beberapa kesempatan justru lebih memudahkan pemahaman.

- c. Gabungkan cara 1 dan cara 2 sekaligus, sesuai dengan kondisi ERD yang akan digambarkan. Jika cara ini yang dipilih maka kelompokkan setiap entitas ke dalam bagian-bagian atau potongan-potongan paling baik. Gambarkan ERD setiap bagian atau potongan tersebut yang hanya memuat komponen entitas dan kerelasian antar entitas saja. Selanjutnya setiap bagian/potongan tersebut hanya memuat komponen entitas dan kerelasian antar entitas saja. Selanjutnya setiap bagian/potongan dilengkapi dengan keterangan atau tabel yang memuat perincian atribut yang sesuai.

I. Penjualan

Menurut Indrajani (2015:62) kegiatan penjualan terdiri atas penjualan barang dan jasa baik secara kredit maupun secara tunai. Dalam sistem penjualan secara tunai, barang atau jasa baru diserahkan oleh perusahaan kepada pembeli jika perusahaan telah menerima kas dari pembeli.

Penjualan menurut cara bayarnya dapat dibedakan sebagai berikut :

- a. Penjualan tunai, yaitu penjualan yang dilakukan oleh perusahaan dengan cara mewajibkan pembeli dengan melakukan pembayaran harga barang terlebih dahulu sebelum barang diserahkan kepada pembeli.
- b. Penjualan kredit, yaitu penjualan yang dilakukan dengan cara memenuhi pesanan pelanggan dengan mengirimkan barang atau menyerahkan jasa, dan

untuk jangka waktu tertentu perusahaan memiliki piutang kepada pelanggannya.

Menurut Indrajani (2015:62) berikut ini dokumen yang dibutuhkan dalam siklus pendapatan perusahaan dagang :

- a. *Sales order*, formulir formal yang memiliki banyak copy yang dipersiapkan dari *customer*.
- b. *Shipping notice*, biasanya merupakan salinan dari SO atau dokumen pengiriman terpisah yang berfungsi sebagai bukti bahwa barang telah dikirimkan.
- c. *Sales invoice*, dokumen yang dikirimkan ke pelanggan untuk menyatakan beberapa jumlah penjualan.
- d. *Remittance advice*, dokumen yang menunjukkan jumlah penerimaan kas dari pelanggan.
- e. *Deposit slip*, dokumen yang menyertai penyetoran kas ke bank.
- f. *Back order*, dokumen yang dipersiapkan ketika kuantitas dari persediaan tidak mencukupi *sales order*.
- g. *Credit memo*, dokumen yang memungkinkan pengurangan kredit pelanggan untuk pengembalian penjualan/penyisihan penjualan.
- h. *Credit application*, sebuah formulir yang dipersiapkan ketika pelanggan baru menggunakan kredit.

- i. *Cash register receipt*, formulir yang digunakan oleh *retailer* untuk mencerminkan kas yang diterima.

2.2 Penelitian Terkait

Menurut Riandy (2011) dalam jurnalnya yang berjudul “Rancang Bangun Sistem Informasi Penjualan Barang Pada UD Cendana Depok Townsquare”. Saat ini sistem yang digunakan oleh UD Cendana masih dilakukan secara manual sehingga terdapat berkas-berkas yang mudah rusak dan hilang, sehingga sering terjadi kehilangan bukti-bukti penjualan tidak terstruktur dan terorganisir dengan baik. Sistem informasi penjualan barang ini memberikan informasi berupa sistem penjualan, pemesanan barang, pembelian barang, mutasi barang serta retur barang. Laporan-laporan ini nantinya dapat digunakan untuk *owner* sebagai bahan dalam pengambilan keputusan dalam pertimbangan-pertimbangan yang matang untuk peningkatan kinerja UD Cendana ini. Sistem ini dapat mempermudah proses penjualan barang karena yang pada awalnya semua dilakukan secara manual, sekarang dengan adanya sistem ini semua data yang dibutuhkan untuk penjualan barang ada dalam sistem ini.

Menurut Nurcahyono (2012) dalam jurnalnya yang berjudul “Pembangunan Aplikasi Penjualan Dan Stok Barang Pada Toko Nuansa Elektronik Pacitan”. Pada Toko Nuansa Elektronik Pacitan, pencatatan dan pengolahan data barang, jumlah dan harga barang, data para *supplier*, serta data transaksi penjualan masih dilakukan secara konvensional. Hal ini dapat dikatakan kurang efektif dan efisien, dikarenakan sulit untuk mencatat dan menghitung banyaknya jenis barang yang ada, banyaknya jumlah barang, maupun besarnya jumlah harga. Banyaknya jumlah barang yang dijual dan tingkat keramaian pembeli dapat mengakibatkan penjual mengalami kesulitan untuk mengelola dan menghitung transaksi penjualan secara cepat dan efisien. Sistem yang telah terkomputerisasi dengan program komputer akan sangat membantu sekali dalam menyelesaikan suatu masalah. Dalam hal ini pengolahan transaksi dalam menyelesaikan suatu masalah transaksi penjualan, sehingga konsumen tidak dikecewakan oleh perusahaan karena lamanya dalam pemrosesan dan pengolahan data tidak menjamin kebenaran dan ketepatannya.