

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian sebelumnya dilakukan oleh Viastri Rindengan prodi Teknik Informatika, Universitas Sam Ratulangi dengan judul “Rancang Bangun Aplikasi Audit Mutu Akademik Internal Universitas Sam Ratulangi Berbasis Web”. Tujuan dari penelitian ini adalah membuat aplikasi audit mutu akademik internal untuk memeriksa kesesuaian antara standar mutu yang diterapkan. Aplikasi ini membantu proses audit sehingga meningkatkan kualitas di Universitas Sam Ratulangi. Metode yang digunakan dalam penelitian ini adalah metodologi RAD (*Rapid Application Development*) yang merupakan metode proses perangkat lunak yang memiliki tahapan fase yang singkat [3].

Penelitian sebelumnya juga dilakukan oleh Andie dari Fakultas Teknologi Informasi, Universitas Islam Kalimantan Muhammad Arsyad Al Banjari Banjarmasin dengan judul “Sistem Informasi Audit Mutu Internal (SIAMI)”. Sebelumnya proses Audit Mutu Internal di Universitas Islam Kalimantan Muhammad Arsyad Al Banjarmasin dilaksanakan secara manual. Oleh karena itu, untuk memudahkan auditor, auditee, maupun panitia pelaksana dalam melaksanakan semua proses dalam Audit Mutu Internal (AMI) di Universitas Islam Kalimantan Muhammad Arsyad Al Banjari Banjarmasin maka perlu dibuat sistem secara otomatis berupa aplikasi guna memudahkan setiap prosesnya. Sistem Informasi Audit Mutu Internal menggunakan metode *waterfall* dalam proses pengembangan sistemnya [4].

Penelitian sebelumnya juga pernah dilakukan oleh Yoga Sendika Dharma jurusan Akuntansi, Fakultas Ekonomi, Universitas Islam Indonesia dengan judul “Audit Operasional Terhadap Pengendalian Intern Pengadaan Barang di PT. Perkebunan Nusantara II (Persero) Dalam Rangka Meningkatkan Efektivitas”. Kegiatan pengadaan barang merupakan kegiatan yang sangat vital dalam keberlangsungan operasi perusahaan sehari-hari, jika dalam pelaksanaan pengadaan barang sering terjadi penyelewengan maka akan memberikan dampak tidak efektif dan efisiennya pengadaan. Maka dalam hal ini diperlukannya suatu

pengawasan yang dapat menekan terjadinya penyelewengan tersebut salah satunya adalah dengan mengadakan audit operasional. Audit operasional sebagai sarana yang dapat menunjang keberhasilan suatu usaha sangat dibutuhkan oleh setiap perusahaan. Menurut standar profesi akuntan publik, menyatakan bahwa tidak ada jaminan penuh bahwa hasil audit akan dapat mendeteksi semua kekeliruan, ketidakberesan, dan pelanggaran hukum, tetapi diatur keharusan-keharusan bagi auditor untuk menentukan risiko mengenai suatu kekeliruan, ketidakberesan dan pelanggaran hukum yang mungkin terjadi [5].

Penelitian sebelumnya juga pernah dilakukan oleh Yadi Utama dari Fakultas Ilmu Komputer, Universitas Sriwijaya dengan judul “Sistem Informasi Berbasis Web Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Sriwijaya”. Proses penyampaian informasi Fakultas Ilmu Komputer Universitas Sriwijaya sudah menggunakan berbagai cara antara lain melalui media elektronik dan media cetak, brosur, dan website. Akan tetapi, website yang saat ini digunakan masih belum mendukung sebagai sebuah sistem informasi yang berbasis web serta belum dapat memenuhi akan informasi bagi seluruh civitas akademika maupun masyarakat umum, yang mengakibatkan sering terhambatnya kelancaran tugas tugas yang harus diselesaikan di tingkat jurusan. Maka dilakukan penelitian oleh saudara Yadi Utama untuk membuat sistem informasi berbasis web yang diharapkan dapat membantu mengatasi masalah tersebut [6].

Penelitian sebelumnya juga pernah dilakukan oleh Medi Suhartanto dengan judul “Pembuatan Website Sekolah Menengah Pertama Negeri 3 Delanggu Dengan Menggunakan Php Dan MySQL”. Selama ini prosedur penyampaian informasi dan publikasi kepada siswa, guru, maupun masyarakat umum di Sekolah Menengah Pertama Negeri 3 Delanggu masih menggunakan media seperti spanduk dan pamflet yang dirasa kurang efektif dikarenakan penyampaian informasinya masih terbatas oleh wilayah tertentu saja, dan juga anggaran biaya untuk pembuatan spanduk dan pamflet juga kurang ekonomis dan efisien. Penelitian mengenai website Sekolah Menengah Pertama Negeri 3 Delanggu diharapkan dapat membantu pihak sekolah dalam menyampaikan berbagai informasi kepada Guru, Karyawan, Siswa dan Masyarakat Umum [7].

Pada penelitian ini, sistem informasi yang dibangun adalah sistem audit pengadaan barang untuk mempermudah pihak SPI Politeknik Negeri Cilacap selaku auditor dalam melaksanakan dan mengelola data

dalam proses audit berupa Kertas Kerja Audit (KKA), berita acara, serta berkas lain yang berkaitan dengan pelaksanaan proses pengadaan barang di Politeknik Negeri Cilacap. Perbedaan dari penelitian sebelumnya adalah pada penelitian ini sistem yang dibangun menggunakan fitur notifikasi yang dapat memberikan notifikasi kepada auditor.

2.2 Landasan Teori

2.2.1 Sistem Informasi

Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan [8].

1) Karakteristik Sistem

Model umum sebuah sistem secara sederhana adalah *input*, proses, dan *output*. suatu sistem juga memiliki karakteristik atau sifat-sifat tertentu yang mencirikan bahwa hal tersebut dikatakan sebagai sebuah sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut:

a) Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja untuk sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu subsistem. Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu yang mempengaruhi proses sistem secara keseluruhan. Suatu sistem juga dapat mempunyai sistem yang lebih besar atau sering disebut “supra sistem”.

b) Batasan Sistem (Boundary)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

c) Lingkungan Luar Sistem (Environment)

Segala bentuk yang berada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem. Lingkungan luar dari sistem ini dapat menguntungkan dan juga dapat merugikan sistem. Dengan demikian, lingkungan luar harus dijaga dan dipelihara.

Lingkungan luar merugikan harus dikendalikan, Jika tidak maka akan mengganggu kelangsungan hidup sistem.

d) Penghubung Sistem (interface)

Media yang menghubungkan sistem dengan subsistem lainnya disebut penghubung sistem atau antarmuka sistem. Penghubung sistem ini memungkinkan sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Bentuk output dari satu subsistem akan menjadi input ke subsistem lain melalui penghubung sistem tersebut. Dengan demikian, dapat terjadi integrasi sistem yang membentuk satu kesatuan.

e) Masukan Sistem (Input)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem atau input sistem, yang dapat berupa pemeliharaan (maintenance input) dan sinyal (signal input). Misalnya, dalam unit sistem komputer, "program" adalah maintenance input yang digunakan untuk mengoperasikan komputer dan "data" adalah signal input untuk diproses menjadi informasi.

f) Keluaran Sistem (Output)

Keluaran sistem dapat berupa energi diproses dan diklasifikasikan menjadi keluaran yang berguna. Keluaran Ini merupakan masukan untuk subsistem lain seperti sistem informasi, maka hasil keluarannya adalah informasi. Informasi ini dapat digunakan sebagai input untuk pengambilan keputusan atau hal lain yang menjadi masukan bagi subsistem lainnya.

g) Pengolah Sistem (Proses)

Suatu sistem dapat memiliki proses yang akan mengubah input menjadi output, contohnya adalah sistem informasi audit internal. Sistem ini akan memproses data audit internal menjadi laporan yang dibutuhkan oleh pengawas atau auditor.

h) Sasaran Sistem (Objective)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan deterministik. Jika suatu sistem tidak memiliki target maka operasi sistem tidak berguna. Sebuah sistem dikatakan berhasil jika mencapai tujuan atau target yang direncanakan.

2) Kualitas Informasi

Kualitas Informasi merupakan kualitas keluaran sistem yang berbentuk informasi yang dihasilkan sistem informasi yang dipakai. Kualitas Informasi juga bisa dibilang sebagai pengukuran kualitas sebuah sistem informasi, jika sistem informasi yang menghasilkan

informasi yang akurat, relevan, dan tepat waktu maka akan berdampak positif terhadap kepuasan *user* [9].

2.2.2 Audit

Audit internal dalam hal ini inspektorat, merupakan audit yang dilakukan oleh unit pemeriksa yang merupakan bagian dari organisasi yang diawasi, dan fungsi dari auditor internal adalah melakukan fungsi pemeriksaan internal yang merupakan suatu fungsi penilaian yang independen dalam suatu organisasi untuk menguji dan mengevaluasi kegiatan organisasi yang dilakukan[10].

2.2.3 Pengadaan barang dan atau jasa

Pengadaan barang/jasa memiliki tujuan antara lain untuk memperoleh barang/jasa dengan harga yang dapat dipertanggungjawabkan dengan jumlah dan mutu yang sesuai dan tepat pada waktunya (tepat jumlah, tepat mutu dan tepat waktu). Dalam pasal 1 angka 1 Perpres 54 Tahun 2010 dinyatakan bahwa “Pengadaan Barang/Jasa Pemerintah yang selanjutnya disebut dengan Pengadaan Barang/Jasa adalah kegiatan untuk memperoleh Barang/Jasa oleh Kementerian/Lembaga/ Satuan Kerja Perangkat Daerah/Institusi lainnya yang prosesnya dimulai dari perencanaan kebutuhan sampai diselesaikannya seluruh kegiatan untuk memperoleh Barang/Jasa.”

2.2.4 Website

Website dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman [11]. Sebuah website biasanya dibangun atas banyak halaman web yang saling berhubungan atau kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak (animasi), suara, atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman.

2.2.5 Basis Data

Basis Data (*database*) terdiri dari 2 kata, yaitu basis dan data. Basis merupakan tempat berkumpul atau bersarang, dan data sendiri dapat diartikan sebuah representatif dari objek nyata yang direpresentasikan dalam bentuk angka, huruf, simbol dan sebagainya [12]. Jadi database merupakan seperangkat data yang saling berkaitan dan berinteraksi. Satu *database* menunjukkan suatu kumpulan data yang digunakan dalam satu lingkungan. *Database* ada untuk mengatasi beberapa masalah yang sering dihadapi ketika proses pengolahan data seperti :

- a) Redudansi dan Inkonsistensi
Data Penyimpanan data yang sama pada beberapa tempat atau media penyimpanan yang akhirnya mengakibatkan terjadinya pemborosan media penyimpanan. Penyimpanan data yang sama dan berulang-ulang di beberapa file dapat mengakibatkan inkonsistensi.
- b) Keamanan Data
Dengan database management sistem, sistem keamanan data bisa dicapai.
- c) Kesulitan Mengakses Data
Database dapat mengakses kesulitan dalam mengakses data karena mampu mengambil data secara langsung dengan program aplikasi yang mudah digunakan
- d) Isolasi Data untuk Standarisasi
Jika data tersebar dalam bentuk format yang tidak sama, maka ini menyulitkan dalam menulis program aplikasi untuk mengambil dan menyimpan data. Maka suatu *database* haruslah dibuat suatu format, sehingga mudah dibuat program aplikasinya.

Database dikelola dengan cara menuliskan sebuah kode perintah berupa *query* SQL yang harus dimasukan sesuai kebutuhan dan hal tersebut terbagi menjadi 2 yaitu, DDL (*Data Definition Language*) dan DML (*Data Manipulation Language*). Berikut dibawah ini merupakan pengertian DDL dan DML:

1) DDL

DDL digunakan untuk pengoperasian skema struktur pada database, perintah utama yang dapat digunakan pada DDL diantaranya, *create*, *rename*, *alter*, dan *drop*.

a) *Create*

Perintah untuk membuat database dan table. Contoh perintah, "CREATE TABLE users (id INTEGER PRIMARY KEY,

username VARCHAR(20) NULL, password VARCHAR(225) NOT NULL)”.

b) *Alter*

Perintah yang digunakan untuk merubah atau mengedit. Contoh perintah, “Alter TABLE users ADD level int(10);”.

c) *Drop*

Drop digunakan untuk menghapus tabel maupun database. Contoh perintah “Drop Table laporan;”.

2) DML

DML merupakan perintah yang dipakai untuk melakukan pengelolaan pada *database* seperti menambah, menghapus, membuat baru, dan menampilkan. Perintah utama pada DML diantaranya adalah *update, delete, insert, dan select*.

a) *Select*

Perintah yang berfungsi untuk menampilkan data yang telah dimasukan. Contoh perintah, “SELECT * FROM users;”

b) *Insert*

Perintah untuk memasukan data ke dalam database. Contoh perintahnya, “INSERT INTO laporan (id, tanggal, keterangan) values(1, '12-10-2021', 'Selesai');”

c) *Update*

Perintah yang dilakukan untuk pembaruan data. Contoh perintahnya, “UPDATE users SET nama = “Adi” WHERE id = 12;”.

d) *Delete*

Perintah untuk menghapus data. Contoh perintah, “DELETE FROM users WHERE id=3;”

2.2.6 Pemrograman Berorientasi Objek

Pemrograman berorientasi objek (PBO) merupakan salah satu metode yang memungkinkan untuk membuat sebuah *software* yang terdokumentasi dengan baik, mudah dipelihara, dapat diandalkan, dan mudah untuk digunakan. Pemecahan masalah yang dilakukan dalam PBO adalah pembuat dapat menggambarkan sebuah objek pada sebuah *software* seperti objek yang memiliki perilaku dan atribut-atribut [13]. Berikut konsep dasar dalam mendalami Pemrograman Berbasis Objek:

a) Objek

Objek merupakan pembungkus data dan fungsi secara bersama menjadi suatu unit dalam sebuah program. Objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.

b) Kelas

Kelas merupakan purwarupa yang mendefinisikan data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu.

c) *Encapsulation* (Pembungkusan)

Encapsulation adalah mekanisme pemrograman yang membungkus kode dan data yang dimanipulasi dan menjaganya supaya terhindar dari interferensi dan penggunaan yang tidak perlu. Salah satu caranya dengan membentuk objek.

d) *Polymorphism* (Perbedaan Bentuk)

Polymorphisme artinya mempunyai banyak bentuk. Dua objek atau lebih dikatakan sebagai *polymorphic*, bila objek-objek itu mempunyai antarmuka yang *identical* namun mempunyai perilaku yang berbeda.

e) Abstraksi

Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan.

f) *Inheritance* (Pewarisan)

Inheritance memungkinkan programmer meletakkan member yang sama dalam satu kelas dan kelas - kelas lain dapat mewarisi member tersebut. Kelas yang mengandung anggota yang sama dari beberapa kelas lain dinamakan *superclass* atau *parent class*. Kelas yang mewarisi dinamakan *subclass* atau *child class*. *Inheritance* menghasilkan *class hierarchy*.

2.2.7 UML

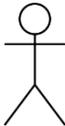
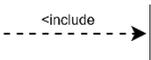
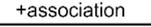
Unified Modeling Language merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah *software* yang berorientasikan pada objek. UML merupakan sebuah standar penulisan atau semacam *blueprint* dimana didalamnya

termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik [14].

1) Use Case Diagram

Use case adalah pemodelan sistem informasi yang akan dibangun dengan mencantumkan kelakuan atau interaksi antara sistem dan *actor*. *Use case* digunakan untuk memberi informasi mengenai fungsi yang berada di dalam sistem dan siapa yang menggunakan fungsi-fungsi tersebut [15].

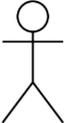
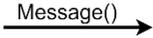
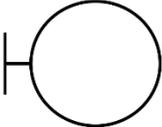
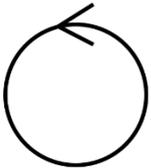
Tabel 2. 1 Simbol *Use Case Diagram*

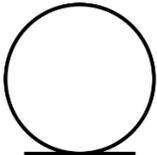
No	Simbol	Fungsi
1.		Segala sesuatu yang berinteraksi dengan sistem aplikasi komputer. Jadi aktor ini bisa berupa orang, perangkat keras atau mungkin juga objek lain dalam sistem yang sama.
2..		Menspesifikasikan bahwa perilaku <i>use case</i> merupakan bagian dari <i>use case</i> lain.
3		Menggambarkan navigasi antar class, berupa banyak objek lain yang berhubungan dengan satu objek, dan apakah suatu class menjadi bagian dari class lainnya.
4.		<i>System Boundary</i> yaitu batasan sebuah sistem.
5.		<i>Use case</i> menjelaskan urutan kegiatan yang dilakukan aktor dan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan, <i>use case</i> hanya menjelaskan apa yang dilakukan oleh aktor dan sistem, bukan bagaimana aktor dan sistem melakukan kegiatan.

2) *Sequence Diagram*

Sequence Diagram merupakan diagram yang menjelaskan *behavior* objek ke *use case*. Kelakuan objek nanti akan dijelaskan dengan mendeskripsikan jangka hidup objek dan pesan yang diterima dan dikirim [15].

Tabel 2. 2 Simbol *Sequence Diagram*

No	Simbol	Fungsi
1.		Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2..		Digunakan untuk menggambarkan <i>user</i> /Pengguna.
3.		Spesifikasi dari komunikasi antar objek yang memuat informasi – informasi tentang aktivitas yang terjadi.
4.		Digunakan untuk menggambarkan sebuah <i>form</i> .
5.		Digunakan untuk menghubungkan <i>Boundary</i> dengan tabel.

No	Simbol	Fungsi
6		Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan.

3) Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. *Class* diagram dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Berikut adalah simbol-simbol yang ada pada *class* diagram [16].

Tabel 2. 3 Simbol *Class Diagram*

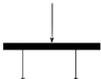
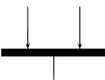
No.	Simbol	Fungsi
1.		Mendefinisikan kelas pada suatu sistem.
2.		Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.
3.		Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).

No.	Simbol	Fungsi
4.		Relasi antarkelas dengan makna kebergantungan antarkelas.
5.		Relasi antarkelas dengan makna semua-bagian (whole-part).

4) Activity Diagram

Activity diagram merupakan diagram yang berfungsi untuk menggambarkan *workflow* atau aliran kerja atau aktivitas dari sebuah sistem. Simbol-simbol yang digunakan dalam *activity* diagram yaitu[15]:

Tabel 2. 4 Simbol *Activity Diagram*

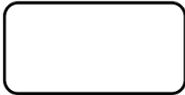
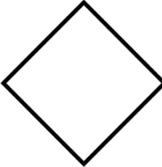
No.	Simbol	Fungsi
1.		Start Point, menggambarkan awal aktivitas.
2.		End Point, menggambarkan akhir aktivitas.
3.		Activities, menggambarkan suatu proses atau kegiatan.
4.		Fork atau percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
5.		Join atau penggabungan, digunakan untuk menghubungkan kembali activity dengan action secara paralel.
6.		Decision Points, menggambar kan pilihan untuk pengambilan keputusan true atau false.

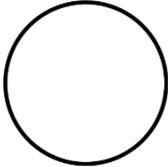
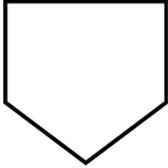
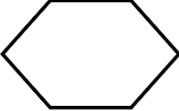
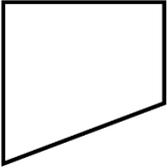
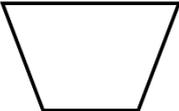
No.	Simbol	Fungsi
7.		Swimlane, pembagian activity diagram untuk menunjukkan siapa dan melakukan apa.

2.2.8 Flowchart

Flowchart atau bagan alir adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. Tujuan *flowchart* merupakan untuk membantu menggambarkan suatu alur dalam menyelesaikan masalah dengan menggunakan simbol-simbol [17]. Simbol ini dipakai sebagai alat bantu menggambarkan proses di dalam program. Berikut ini simbol – simbol di dalam *flowchart* [18]:

Tabel 2. 5 Simbol Flowchart

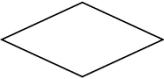
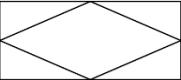
No	Gambar Simbol	Nama	Keterangan
1.		Terminal	Menyatakan permulaan atau akhir suatu program.
2.		Arus/ <i>Flow</i>	Untuk menyatakan jalanya arus suatu proses.
3.		Proses	Menunjukkan sebuah kegiatan atau proses yang dilakukan sistem.
4.		<i>Decision</i> /Logika	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan (ya/tidak).
5.		<i>Input output</i>	Menyatakan proses <i>input/output</i> tanpa tergantung dengan jenis peralatannya.

No	Gambar Simbol	Nama	Keterangan
6.		<i>Connector</i>	Menyatakan sambungan dari suatu proses ke proses lainnya dalam halaman/lembar yang sama.
7.		<i>Offline Connector</i>	Menyatakan sambungan dari suatu proses ke proses lainnya dalam halaman/lembar yang berbeda.
8.		<i>Predefined Proses</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
9.		<i>Manual input</i>	Memasukan data secara manual dengan menggunakan <i>online keyboard</i> .
10.		Manual	Menyatakan suatu Tindakan (proses) yang tidak dilakukan oleh komputer (manual).
11.		<i>Document</i>	Mencetak laporan ke printer.

2.2.9 Entity Relationship Diagram (ERD)

Pengertian ERD adalah suatu komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut yang mempresentasikan seluruh fakta dari dunia nyata yang ditinjau Entity Relationship Diagram menggambarkan data dan hubungan antar data secara global dengan menggunakan Entity Relationship Diagram [19]. ERD digunakan untuk memodelkan struktur data dan hubungan antar data. Simbol-simbol yang digunakan dalam ERD dijelaskan pada **Tabel 2.6** berikut:

Tabel 2. 6 Simbol ERD

No	Simbol	Keterangan
1.		Persegi panjang menyatakan himpunan entitas adalah orang, kejadian, atau berada.
2.		Atribut merupakan informasi yang diambil tentang sebuah entitas.
3.		Belah ketupat menyatakan himpunan relasi merupakan hubungan antar entitas.
4.		Garis sebagai penghubung antara himpunan, relasi, dan himpunan entitas dengan atributnya
5.		Entitas yang digunakan pada relasi many-to-many.

ERD memiliki derajat relasi atau biasa disebut kardinalitas. Kardinalitas menjelaskan batasan jumlah keterhubungan satu entity dengan entity lainnya. Berikut adalah macam-macam kardinalitas yang dapat dilihat pada **Tabel 2.7** berikut:

Tabel 2. 7 Macam-Macam Kardinalitas

No.	Simbol	Nama	Keterangan
1.		Relasi Satu ke	Relasi yang menunjukkan bahwa setiap himpunan

No.	Simbol	Nama	Keterangan
		Satu (<i>One to One</i>)	entitas berhubungan dengan tepat satu himpunan entitas lainnya
2.		Relasi Satu ke Banyak (<i>One to Many</i>)	Relasi yang menunjukkan bahwa hubungan antara entitas pertama dengan entitas kedua adalah satu banding banyak, begitu pula sebaliknya
3.		Relasi Banyak ke Banyak (<i>Many to Many</i>)	Relasi yang menunjukkan bahwa setiap himpunan entitas boleh berhubungan dengan banyak himpunan entitas lainnya dan sebaiknya