



**BAB II**  
**TINJAUAN PUSTAKA**  
**DAN LANDASAN TEORI**

## **BAB II**

### **DASAR TEORI**

#### **2.1. Tinjauan Pustaka**

Sebelumnya telah ada penelitian dengan judul yang berkaitan yang dilakukan oleh Muthohari [1] dengan judul “Pengembangan Aplikasi Kasir Pada Sistem Informasi Rumah Makan Padang Ariung”. Aplikasi kasir adalah suatu aplikasi yang ditujukan untuk rumah makan padang ariung yang dalam proses penulisan transaksi masih dilakukan dengan cara manual, hal ini dapat dikatakan kurang efektif dan efisien, dikarenakan sering terjadi kesalahan dalam menghitung jumlah yang harus dibayar dan banyaknya waktu yang dibutuhkan untuk penulisan laporan yang tepat dan akurat, selain itu dikarenakan prosesnya masih dilakukan secara manual, arsip yang disimpan merupakan bukti salah satunya terjadi transaksi dan tingkat keramaian pembeli dapat mengakibatkan penjual mengalami kesulitan mengelola dan menghitung transaksi penjualan secara cepat, tepat, dan efisien. Oleh karena itu, maka dibuat suatu sistem yang dapat menampilkan paket makanan dan dapat membuat struk atau nota pembayaran. Hasil yang telah dicapai adalah sistem baru ini dirancang sesuai kebutuhan lapangan sehingga pada saat proses penginputan menu makanan dan transaksi pembayaran, dapat menghasilkan pengkajian secara cepat, baik, dan benar sehingga proses transaksi lebih efektif dan efisien.

Penelitian tentang sistem yang berkaitan dengan judul dari peneliti sebelumnya juga dilakukan oleh [2] yang berjudul “Perancangan Sistem Informasi Kasir Pada Resto Seafood Bahari Tegal”. Resto Seafood Bahari merupakan suatu usaha yang bergerak dibidang kuliner di Kota Tegal; yang menyajikan berbagai hidangan dari hasil laut. Dalam kegiatan transaksi penjualan di resto Seafood Bahari masih dilakukan secara manual atau konvensional sehingga sering terjadi kesalahan dalam menghitung jumlah yang harus dibayar dan memerlukan banyak waktu untuk melakukan penulisan laporan yang tepat dan akurat, selain itu karena kegiatan transaksi penjualan masih dilakukan secara manual maka arsip yang disimpan merupakan bukti satu-satunya dalam kegiatan transaksi. Oleh karena itu, maka penulis membuat sistem informasi kasir yang diharapkan dapat membantu kegiatan administrasi dan kinerja kasir menjadi lebih optimal. Manfaat yang dihasilkan dari sistem ini adalah

dapat melakukan penginputan data laporan transaksi penjualan dalam kurun waktu tertentu, menghasilkan laporan data barang dari pengolahan data yang masuk, membuat laporan pembelian serta pengolahan data barang.

Pada penelitian yang dilakukan oleh Kristanto [3] yang berjudul “Perancangan Sistem Informasi Kasir Pada Toko Rejomaret di Wonogiri”. Dalam melakukan segala kegiatan usaha di Toko Rejo Maret masih dilakukan secara konvensional atau manual sehingga menimbulkan permasalahan dalam proses transaksi, penyimpanan arsip yang tidak teratur yang menyebabkan sering terjadi kesulitan dalam melakukan pengecekan sisa barang, penginputan data barang, pencarian data barang dan segala proses pengolahan data barang menjadi cukup lama karena harus memeriksa tumpukan buku satu persatu. Dengan alasan tersebut, penulis mencoba merancang sistem informasi kasir di toko ‘Rejomaret’ wonogiri yang dapat membantu pemilik toko dalam melakukan administrasi semua kegiatan transaksi. Hasil pengolahan data dari pada sistem ini berupa laporan data barang, data supplier, data kategori, data penjualan per Tanggal, data penjualan per periode, data penjualan bulanan, data penjualan mingguan, data penjualan harian, data pembelian per periode, data pembelian bulanan dan seluruh pembelian.

Penelitian yang berjudul “Perancangan Sistem Informasi Aplikasi Kasir pada Kafe Restorasi Kopi Berbasis Web“ dilakukan oleh Latifah [4]. Pertumbuhan usaha kafe yang begitu pesat di Indonesia ini dengan banyaknya pengunjung, sehingga kafe mendapatkan kesulitan untuk memerhatikan orderan pesanan. Pengunjung menunggu terlalu lama untuk mendapatkan pesanan yang dipesan sehingga kurang adanya kenyamanan bagi pengunjung. Sistem informasi aplikasi kasir ini dirancang dalam proses pemesanan menu makanan dan minuman dengan berbasis website menggunakan kode respons cepat (QR) yang tujuannya untuk mwmbuat operasional kafe berjalan menjadi lebih cepat, aman, efektif, dan akurat. Metodologi yang digunakan yaitu Rational Unified Process, sebagai salah satu tahap pengembangan aplikasi dengan empat tahapan yang digunakan yaitu inception, elaboration, construction, dan transition dengan memakai pemodelan UML. Pada aplikasi sistem informasi kasir ini menghasilkan pengelolaan laporan penjualan kafe menjadi lebih baik karena adanya proses yang terkomputerisasi, maka data dapat tersimpan dengan aman. Sistem informasi ini memakai database MySQL dan dikembangkan menggunakan bahasa pemrograman PHP.

Penelitian dilakukan oleh Gunawan [5] dengan judul “Sistem Informasi Penjualan Berbasis Web pada Restoran Caki Cake Karawang”. Saat ini pengolahan informasi pada restoran CAKI CAKE masih secara manual sehingga menghasilkan output yang tidak efisien dan efektif, pada pengolahan menu pesanan pelanggan masih dicatat dalam nota pesanan kemudian diberikan kepada bagian dapur dan kasir untuk dilakukan proses pengolahan hidangan dan rekapitulasi tagihan. Oleh karena itu kerap terjadi kesulitan dalam mengelola pesanan seperti menentukan pesanan yang terlebih dahulu diolah atau diantarkan serta diperlukannya waktu untuk melakukan perhitungan secara manual yang dapat menimbulkan antrian yang tentunya dapat memperburuk kualitas pelayanan dengan semakin banyaknya pengunjung. Dengan adanya masalah yang dihadapi oleh restoran CAKI CAKE tersebut maka membutuhkan kehadiran website. Hal ini sangat penting karena keberadaan suatu website dapat membantu penyampaian informasi produk dan harga secara detail kepada konsumen. Diharapkan akan mampu menjawab permasalahan yang dihadapi oleh restoran CAKI CAKE sehingga dapat menunjang kinerja restoran menjadi lebih efisien dan meningkatkan kepercayaan serta memberikan kepuasan tersendiri bagi para konsumen.

Pada penelitian ini, penulis bermaksud membangun Sistem Informasi Penjualan Berbasis *Website* yang akan diterapkan di Rumah Makan Ayam Penyet Stasiun Maos dengan metode waterfall dan ahasa pemrograman PHP serta database MySQL. Tahap pengujian menggunakan metode *Black Box* agar dapat memberikan hasil yang lebih baik dan dapat mengetahui dari setiap langkah-langkah jalannya sistem serta berdampak baik terhadap pengembangan dan evaluasi. Pada penelitian sebelumnya sebagian berisi tentang proses pemesanan, transaksi, hingga rekapitulasi atau laporan keuangan. Pada penelitian ini sistem dapat melayani pemesanan secara online dan melayani pengiriman makanan atau *delivery order*. Sistem informasi ini diharapkan dapat digunakan oleh Warung Makan Ayam Penyet Stasiun Maos untuk mempermudah petugas kasir yang mempercepat semua proses transaksi penjualan dan pelayanan serta memberi kemudahan dalam sistem pembukuan atau pendataan agar lebih teratur. Dengan adanya sistem informasi berbasis website ini, maka proses transaksi yang dilakukan lebih optimal dan pelayanan pelanggan menjadi lebih efektif dan efisien.

## 2.2. Landasan Teori

Dalam penelitian ini diperlukan adanya teori yang mendasar dalam menunjang proses penelitian, berikut antara lain:

### 2.2.1. Rekayasa Perangkat Lunak

Perangkat lunak adalah objek tertentu yang dapat dijalankan seperti sumber, kode objek, atau sebuah program yang lengkap [5]. Perangkat lunak tidak hanya mencakup program, tetapi juga semua dokumentasi dan konfigurasi data yang saling terhubung sehingga program beroperasi dengan benar [10].

Rekayasa perangkat lunak adalah disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal spesifikasi sampai pemeliharaan sistem. Rekayasa perangkat lunak juga mencakup masalah pemilihan metode yang paling sesuai dengan keadaan.

Proses perangkat lunak sebagai kerangka kerja untuk tugas-tugas yang dibutuhkan dalam membangun perangkat lunak dengan kualitas produk yang baik [5]. Proses perangkat lunak juga meliputi teknologi yang mempopulasikan metode-metode (*method*) teknis, alat-alat bantu (*tools*) otomatis dan prosedur-prosedur (*procedure*) atau sering juga disebut dengan proses. Berikut penjelasannya:

#### a. Metode

Metode dapat diartikan sebagai cara bagaimana secara teknis membangun perangkat lunak yang harus berada pada sebuah komitmen dasar menuju kualitas. Metode ini menyangkut serangkaian tugas luas yaitu :

1. Perencanaan proyek dan estimasi.
2. Analisis kebutuhan sistem dan *software*.
3. Rancangan struktur data, yang terdiri dari 3:
  - Variabel
  - Elementary Data
  - Struktur Data
4. Arsitektur program
5. Algoritma prosedur
6. Pengkodean (*coding*)
7. Testing
8. Pemeliharaan

#### b. Alat bantu

Alat bantu atau *tools-tools* perangkat lunak memberikan topangan yang otomaatis ataupun semi-otomatis pada proses-proses dalam metode-

metode yang ada seperti *Computer Aided Software Engineering* (CASE) yang terdiri dari :

1. *Easy Case Data Flow Diagram* (DFD)
2. *Entity Relationship for Windows* (ERWIN)
3. *Entity Relationship Diagram* (ERD)

Fungsi dari CASE yaitu mengkombinasikan atau menggabungkan perangkat lunak, perangkat keras dan *database* rekayasa perangkat lunak / *software engineering database* (*repository* yang berisi informasi penting tentang analisis desain, konstruksi program serta pengujian) dengan CAD/CAE (*Computer Aided Design / Computer Aided Engineering*) untuk perangkat keras.

### c. **Prosedur**

Prosedur merupakan penggabungan antara metode dan alat bantu. Prosedur mendefinisikan urutan (*sequence*) metode yang akan digunakan untuk seorang engineer. Prosedur juga selalu mendefinisikan kontrol membantu keyakinan kualitas dan perubahan koordinasi, dan mendefinisikan keluaran (berupa dokumen, laporan dan formulir yang dibutuhkan). Proses rekayasa perangkat lunak secara linier model mengusulkan sebuah pendekatan yang sistematis dan sekuensial biasanya disebut dengan model air terjun (*waterfall model*). *Waterfall model* mengusulkan sebuah pendekatan yang sistematis dan sekuensial. Tahapan tersebut proses tersebut adalah sebagai berikut (Pressman, Roger S<sup>[7]</sup>) :

1. **Rekayasa dan penyusunan sistem**  
Tahap ini dimulai dengan menyusun kebutuhan untuk seluruh elemen sistem dan kemudian mengalokasikan beberapa subset dari kebutuhan tersebut pada perangkat lunak harus berinteraksi dengan elemen lainnya, seperti perangkat keras, manusia dan *database*.
2. **Analisis kebutuhan perangkat lunak**  
Pada tahap ini lebih diintensifkan dan difokuskan pada perangkat lunak. Pengembang perangkat lunak harus paham mengenai fungsi yang dibutuhkan, perilaku, daya guna dan tampilan layar dari perangkat lunak yang akan dikembangkan.
3. **Desain**  
Desain perangkat lunak sebenarnya merupakan proses bertahap yang berfokus pada empat atribut pada sebuah program : struktur data, arsitektur, perangkat lunak, representasi tampilan layar, dan detail prosedural (algoritmik). Proses desain menterjemahkan

kebutuhan menjadi suatu representasi perangkat lunak yang dapat diakses sebelum pengkodean dimulai.

4. Pengkodean

Proses menterjemahkan bentuk desain menjadi bentuk yang dapat dibaca oleh mesin.

5. Pengujian

Setelah kode dihasilkan, dilakukan pengujian program. Proses pengujian berfokus pada bagian internal perangkat lunak secara logis untuk memastikan bahwa setiap pernyataan (*statement*) telah diuji, dan pada bagian eksternal fungsi, dimana diberlakukan pengujian untuk menemukan kesalahan (*error*) dan memastikan bahwa masukan (*input*) yang ditentukan akan memberikan hasil yang diharapkan.

6. Pemeliharaan

Ketika perangkat lunak telah selesai dikembangkan dan dikirim kepada pelanggan, perangkat lunak tersebut mungkin akan mengalami masalah atau kesalahan yang tidak diharapkan sebelumnya. Untuk itu, tahapan pemeliharaan dilakukan dengan tujuan melakukan penyesuaian dan perbaikan pada perangkat lunak tersebut

**d. *Flowchart***

*Flowchart* adalah suatu bagan yang mempunyai arus menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma. Ada dua *flowchart* yang menggambarkan proses dengan komputer, yaitu:

1) Sistem *Flowchart*

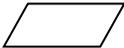
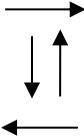
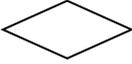
Bagan yang memperlihatkan urutan proses dalam sistem dengan menunjukkan alat media input, output serta jenis media penyimpanan dalam proses pengolahan data

2) Program *Flowchart*

Bagan yang memperlihatkan urutan instruksi yang digambarkan dengan simbol tertentu untuk memecahkan masalah dalam suatu program.

Berikut ini adalah simbol-simbol yang terdapat pada *flowchart* dapat dilihat di Tabel 2.1

**Tabel 2. 1** Simbol *Flowchart*

No.	Simbol	Nama	Keterangan
1.		<i>Terminal</i>	Memulai dan mengakhiri suatu program
2.		<i>Input/Output</i>	Memasukan data maupun menunjukkan hasil dari suatu process tanpa tergantung dengan jenis peralatannya
3.		<i>Flow</i>	Menghubungkan antara simbol satu dengan simbol yang lain atau menyatakan jalannya arus dalam suatu proses. Simbol arus ini sering disebut juga dengan connecting line.
4.		<i>Decision</i>	Memilih proses berdasarkan kondisi yang ada.
5.		<i>Document</i>	Merupakan simbol untuk data yang terbentuk informasi.
6.		<i>Manual Operation</i>	Menunjukkan pengolahan yang tidak dilakukan oleh komputer/pc.
7.		<i>Punched Card</i>	Menyatakan input berasal dari kartu atau output ditulis ke kartu.
8.		<i>Connector</i>	Menyatakan sambungan dari suatu proses ke proses

No.	Simbol	Nama	Keterangan
			lainnya dalam halaman / lembar yang sama.
9.		<i>Offline Connector</i>	Menyatakan sambungan dari satu proses ke proses lainnya dalam halaman atau lembar yang berbeda.
10.		<i>Predefined Process</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.

### 2.2.2. Basis Data

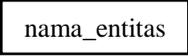
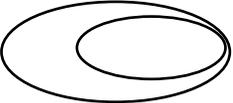
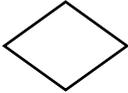
Basis data merupakan himpunan kelompok data yang saling berhubungan dan disimpan dalam media penyimpanan elektronik setelah melalui proses pengelompokan dan pemilahan data sehingga data disimpan tanpa adanya pengulangan (*redundancy*) yang tidak perlu agar dapat dimanfaatkan kembali dengan cepat<sup>(3)</sup>). Pengolahan data bertujuan untuk mempermudah dan mempercepat proses pengambilan data yang telah disimpan.

Tabel yang terbentuk merupakan implementasi dari rancangan yang telah dibuat dalam bentuk *Entity Relationship Diagram* (ERD). ERD menunjukkan hubungan antar entitas yang nantinya akan membentuk tabel didalam DBMS. DBMS bertujuan untuk menyediakan antarmuka (*interface*) kepada pengguna dalam melakukan pengelolaan data. Sistem tersebut akan menyembunyikan detail data disimpan dan dikelola.

ERD (*Entity Relationship Diagram*) adalah diagram yang digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS (*Object-Oriented Database Management System*) maka perancangan basis data tidak perlu menggunakan ERD.

Berikut ini adalah simbol-simbol yang terdapat pada ERD dapat dilihat pada Tabel 2.2

Tabel 2. 2 Simbol ERD

No.	Simbol	Nama	Keterangan
1.		Entitas / <i>Entity</i>	Persegi panjang yang mewakili sekumpulan atau himpunan objek yang berada pada sistem.
2.		Atribut	<i>Field</i> atau kolom data yang perlu disimpan dalam entitas.
3.		Atribut kunci primer	<i>Field</i> atau kolom data yang disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan.
4.		Atribut multinilai	<i>Field</i> yang butuh disimpan dalam entitas yang dapat memiliki nilai lebih dari satu.
5.		Relasi	Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja.
6.		<i>Link</i>	Garis yang mewakili hubungan antara atribut dengan entitas dan himpunan entitas dengan entitas dan sebaliknya.

### 2.2.3. Pemrograman Berorientasi Objek

Pemrograman berorientasi objek merupakan suatu metode pemrograman yang berbasiskan pada objek. Pada pemrograman berorientasi objek, antara objek yang satu berinteraksi dengan objek lainnya dengan tujuan mendapatkan informasi antar objek. Pemrograman berorientasi objek merupakan bagian dari pengembangan sistem berorientasi objek.

Pemrograman berorientasi objek (Sutopo, Hadi Ariesto<sup>[11]</sup>) merujuk pada cara berpikir baru tentang perangkat lunak berdasarkan abstraksi yang terdapat dalam dunia nyata. Hal yang harus diperhatikan dalam pengembangan berorientasi objek ada 2, diantaranya: konsep mengidentifikasi dan mengorganisasi domain aplikasi dengan fokus penggunaan bahasa pemrograman, berorientasi atau tidak. Dalam mengorganisasikan domain diperlukan metodologi berorientasi objek. Metodologi berorientasi objek memiliki 3 karakteristik utama, yaitu:

1. Pengkapsulan (*Encapsulation*)

Pengkapsulan merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses. Data, prosedur atau fungsi dikemas bersama-sama dalam suatu objek, sehingga prosedur atau fungsi lain dari luar tidak dapat mengaksesnya. Data terlindung dari prosedur objek atau objek lain kecuali prosedur yang berada pada objek itu sendiri. Sarana untuk melakukan pengkapsulan adalah kelas. Dalam sebuah kelas, dimungkinkan terdapat beberapa objek sekaligus.

2. Pengkapsulan (*Encapsulation*)

Pewarisan adalah teknik yang menyatakan bahwa anak dari suatu kelas (*subclass*) akan mewarisi seluruh data, atribut maupun metoda yang dimiliki induknya (*superclass*) secara langsung. Dengan demikian, pembuatan atribut maupun metoda yang sama pada kelas yang berbeda, dapat dikurangi.

3. Polimorfisme (*Polymorphism*)

Polimorfisme yaitu konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku berbeda. Polimorfisme mempunyai arti bahwa operasi yang sama mungkin mempunyai perbedaan kelas yang berbeda. Implementasi spesifik dari suatu operasi kelas tertentu disebut metoda. Karena operator berorientasi objek bersifat polimorfisme, maka dimungkinkan ada metoda yang sama dengan perilaku yang berbeda. Polimorfisme dapat dilakukan dengan mekanisme *overloading* atau *overriding*. *Overloading* (*shadowing*) merupakan konsep yang mengizinkan adanya beberapa *method* atau *constructor* dengan nama sama dalam sebuah kelas, sedangkan *overriding* (penumpukan) merupakan konsep yang mengizinkan sebuah *subclass* melakukan perubahan implementasi *method* yang berasal dari *superclass*-nya.

UML (Unified Modeling Language) adalah bahasa untuk memodelkan perangkat lunak yang akan dibuat. Unified Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-

model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek.

a) *Usecase Diagram*

*Usecase diagram* digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*). Sehingga pembuatan *usecase diagram* lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah *usecase diagram* mempresentasikan sebuah interaksi antara aktor dengan sistem. Berikut ini adalah simbol-simbol yang terdapat pada *usecase diagram* dapat dilihat di Tabel 2.3

**Tabel 2. 3** Simbol *Usecase Diagram*

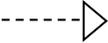
No.	Simbol	Nama	Keterangan
1.		<i>Use case</i>	Deskripsi dari urutan aksi – aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
2.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
3.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
4.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5.		<i>Extend</i>	Menspesifikasikan bahwa <i>use case target</i> memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.

No.	Simbol	Nama	Keterangan
6.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas

b) *Class Diagram*

*Class diagram* adalah inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut atau properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode atau fungsi). Kelas memiliki tiga area pokok yaitu nama (*class name*), atribut, dan metode (*operation*). Berikut ini adalah simbol-simbol yang terdapat pada *class diagram* dapat dilihat di Tabel 2.4

**Tabel 2. 4** Simbol *Class Diagram*

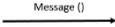
No.	Simbol	Nama	Keterangan
1.		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
2.		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
3.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
4.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung

No.	Simbol	Nama	Keterangan
			padanya elemen yang tidak mandiri.
5.		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6.		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.

c) *Sequence Diagram*

*Sequence diagram* merupakan penggambaran interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem. Berikut ini adalah simbol-simbol yang terdapat pada *sequence diagram* dapat dilihat di tabel 2.5

**Tabel 2. 5** Simbol *Sequence Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2.		<i>Actor</i>	Menggambarkan <i>user</i> atau pengguna.
3.		<i>Message</i>	Spesifikasi dari komunikasi antar <i>objek</i> yang memuat informasi – informasi tentang aktivitas yang terjadi.

No.	Simbol	Nama	Keterangan
4.		<i>Boundary</i>	Menggambarkan sebuah <i>form</i> .
5.		<i>Control Class</i>	Menghubungkan <i>boundary</i> dengan Tabel.
6.		<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan.

d) *Activity Diagram*

*Activity diagram* merupakan penggambaran rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya. Pembuatan *activity diagram* pada awal pemodelan proses dapat membantu memahami keseluruhan proses. *Activity diagram* juga digunakan untuk menggambarkan interaksi antara beberapa use case. Berikut ini adalah simbol-simbol yang terdapat pada *activity diagram* dapat dilihat di Tabel 2.6

**Tabel 2. 6** Simbol *Activity Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>ActivityAQ</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2.		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.

No.	Simbol	Nama	Keterangan
3.		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4.		<i>Actifity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan.
5.		<i>Extends</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

#### 2.2.4. Sistem Penjualan

Menurut [7] definisi dari sistem penjualan, suatu proses melalui prosedur-prosedur yang meliputi urutan kegiatan sejak diterimanya pesanan dari pembeli, pengecekan barang ada atau tidak ada dan diteruskan dengan pengiriman barang disertai dengan pembuatan faktur dan mengadakan pencatatan atas penjualan yang berlaku (Mcleod, Raymond, 2001). Analisis Sistem Penjualan Barang meliputi:

1. Analisis Informasi (Information Analysis) Dalam melakukan penyajian informasi, sistem mampu memberikan informasi secara real time. Dan dalam menyajikan informasi jangkauannya tak terbatas.
2. Analisis Ekonomi (Economy Analysis) Secara ekonomi sistem penjualan yang lama membutuhkan lebih banyak biaya karena karyawan yang dibutuhkan lebih banyak. Selain itu biaya untuk keperluan administrasi lebih banyak, sehingga biaya operasional juga lebih banyak.
3. Analisis Pengendalian (Control Analysis) Dalam sistem lama lebih sulit melakukan control, karena pemrosesan data dilakukan oleh manusia sehingga kemungkinan terjadi kesalahan lebih besar.
4. Analisis Efisiensi (Efficiency Analysis) Penjualan secara manual kurang efisien, karena semua transaksi dan atau administrasi dilakukan secara manual. Selain itu efisiensi karyawan juga kurang, karena jumlah karyawan yang dipekerjakan lebih banyak.

5. Analisis Pelayanan (Service Analysis) Dengan sistem lama maka dalam hal pelayanan terhadap pelanggan memiliki keterbatasan waktu dan jangkauan pelayanan. Selain itu proses transaksi lebih lama. Hal ini berpengaruh terhadap kualitas pelayanan.