# BAB 2
# TINJAUAN PUSTAKA

Analisis biaya pengurangan biaya pengeluaran perusahaan selalu bisa jadi hal yang sangat penting untuk kemudian di perhitungkan. Semua pengusaha pasti ingin selalu menekan biaya pengeluaran dengan tidak mengurangi kualitas barang produksi . dari mulai menghitung biaya pengiriman , biaya simpan, sumber daya manusia, pemilihan bahan baku sampai biaya produksi. Namun dalam penelitian ini yang akan di bahas adalah mengurangi biaya kirim dan menghitung waktu pemesanan kembali. Dari permasalahan yang ada dalam pemyusunan laporan dibutuhkan referensi untuk bahan acuan. Jurnal yang digunakan adalah jurnal dalam kurung waktu 5 tahun terakhir.

## 2.1 Tinjauan Pustaka

Dibawah ini merupakan jurnal penelitian terkait dengan pengurangan biaya kirim dan penentuan waktu pemesanan kembali dengan kurung waktu 5 tahun dengan berbagai penyesuaian cara perhitungan mengikuti data perusahan.

**Table 2.1** Penelitian Terkait

| No | Nama Peneliti dan Tahun | Masalah | Metode | Hasil |
|---|---|---|---|---|
| 1. | Tomi Lukmana, 2015 | Penerapan Metode EOQ dan ROP ( Studi Kasus : PD. Baru) | EOQ dan ROP | Aplikasi *desktop* PD.Baru dapat di kembangkan menjadi lebih efisien dengan menambahkan fitur-fitur seperti akuntansi, penggunaan *barcode*, menggunakan metode *Customer* |

| No | Nama Peneliti dan Tahun | Masalah | Metode | Hasil |
|---|---|---|---|---|
| | | | | *Relationship Management* yang lebih lanjut, sistem penggajian pegawai dan penggunaan metode FIFO LIFO. |
| 2. | Ahmad Ardiansyah, Baibul Tujni, 2014 | Sistem Informasi Perhitungan Stok Barang Pada Toko Sepatu Medley Menggunakan Metode Economic Order Quantity (EOQ) | Metode *EOQ* | Penelitian ini menghasilkan sistem informasi penghitungan stok barang pada toko sepatu *medley* menggunakan metode *economic order quantity* (EOQ). |
| 3. | Maulan Irwadi, 2015 | Penerapan Reorder Point Untuk Persediaan Bahan Baku Produksi Alat Pabrik Kelapa Sawit Pada PT. Swakarya Adhi Usaha Kabupaten Banyumas | Metode Reorder Point | Mampu mengurangi frekuensi biaya pemesanan menggunakan metode ROP dan EOQ. |

## 2.2 PT. Ritz Gallery

PT. Ritz Gallery merupakan perusahaan yang bergerak di bidang *furniture* yang beralamat di Jalan Sunan Mantingan, Kelurahan Karang kebagusan, Kecamatan Jepara. PT. Ritz Gallery merupakan perusahaan yang mengekspor produk *furniture* keberbagai Negara diantaranya adalah Korea Selatan, Jepang, Thailand dan Russia. Perusahaan dituntut agar dapat menghasilkan produksi yang sesuai permintaan pelanggan dan mengirimkannya sesuai dengan permintaan pelanggan sesuai dengan waktu yang telah ditentukan.

Dari data yang di dapat barang masuk dari *supplier* ke PT. Ritz Gallery dari bulan Agustus 2015 sampai Juli 2016 sejumlah 18609 buah. Perusahaan biasa memesan barang ke supplyer setiap seminggu sekali, hal ini menyebabkan pengeluaran biaya pengiriman terbilang banyak karena dalam sekali kirim perusahaan harus membayar Rp. 900.000. Jika dalam satu tahun ada 52 minggu perusahaan harus mengeluarkan uang sebesar Rp. 46.800.000 juta dalam setahun, banyaknya nilai uang perusahan dalam mengeluarkan biaya pengiriman barang dan waktu penentuan pemesanan barang yang masih manual membutuhkan suatu cara perhitungan yang bisa mengelola data dengan cepat dan menghasilkan suatu informasi yang lebih relevan, akurat dan tepat waktu. Saat ini sistem informasi persediaan yang ada di PT. Ritz Gallery bersifat sederhana dan dilakukan dengan cara manual. Teknologi komputerisasi akan mempermudah pegawai perusahaan dalam melakukan kegiatan perhitungan biaya barang masuk pergudangan di perusahaan tersebutFormat Caption

## 2.3 Optimalisasi Biaya

Optimal mempunyai arti yang terbaik berasal dari kata optimalisasi Atau bisa diartikan usaha yang dilakukan untuk mendapatkan hasil atau sebuah proses pencapaian kerja dengan maksut mendapatkan untung tanpa mengurangi kualitas dan mutu kerja.

Menurut kamus besar Bahasa Indonesia yaitu optimalisasi merupkan kata dari yang paling baik ,jadi arti dari kata optimalisasi ialah proses meningkatkan atau meningkatkan atau memperbaiki. [1]

Kesimpulanya optimalisasi adalah keadaan paling baik yang bisa dilakukan perusahaan untuk memanajemen persediaan stok. Dari keseluruhan biaya produksi yang harus dibayar perusahaan dalam permintaan barang setiap periode bisa dilihat dari penilitian ini

## 2.4 Proses Produksi

Adalah kegiatan yang dilakukan dalam menciptakan, membuat atau mengolah dari bahan jadi atau mentah atau bahan setengah jadi hingga akhirnya tercipta suatu karya atau barang yang memiliki nilai jual dan mendapatkan keuntungan dengan bantuan mesin, tenaga kerja, materi dan sebagainya. Hasilnya bisa berupa produk atau jasa tergantung dengan perusahaan atau organisasi sebagai pengolah. [2]

## 2.5 Metode *Economyc Order Quantity (EOQ)*

Semua badan usaha pasti ingin yang terbaik dalam kebijakan penyediaan bahan baku yang cepat dan tepat , namun tidak mengeluarkan biaya yang besar dan mengganggu dalam proses produksi. Demi hal tersebut ada metode yang dapat digunakan yaitu metode EOQ (*Economic Order Quantity)*. Ada beberapa pendapat mengenai pengertian dari EOQ yaitu sebagai berikut [2] :

1. *EOQ* adalah jumlah atau volume ekonomis yang dilakukan setiap sekali pesan barang. Maka dari itu untuk menghindari kehabisan stok perlu di perhitungkan dengan cara yang paling ekonomis , tepat dengan biaya seminim mungkin
2. (*Economic Order Quantity*) *EOQ* adalah metode mencari jumlah terkecil biaya persediaan dengan memaksimalkan jumlah pesanan. Digunakan dalam menentukan pembelian total bahan baku yang dibeli dalam sekali pesan untuk digunakan per tahun nya.

Maksut model *EOQ* yaitu meminimalkan biaya persediaan dengan menentukan jumlah ekonomis dalam sekali pesan. Ada beberapa faktor untuk bisa mewujudkan cara meminimalkan biaya persediaan yaitu:

### 2.5.1 Keunggulan dan Kelemahan Metode *EOQ*

1. Kelebihan metode *EOQ* adalah sebagai berikut [1]:
   a. Dapat digunakan untuk mengetahui berapa banyak persediaan yang harus dipesan dalam hal ini bahan baku, dan kapan seharusnya pemesanan dilakukan
   b. Dapat mengatasi ketidakpastian permintaan dengan adanya persediaan pengaman *(safety stok)*
   c. Mudah di aplikasikan pada proses produksi secara masal

2. Kekurangan metode *EOQ* adalah sebagai berikut:
   a. Mendapatkan pemasok sebagai mitra bisnis sementara Karena paradigma untung rugi ditetapkan oleh mereka sehingga menyebabkan berganti-ganti pemasok
   b. Karena berganti ganti pemasok akibatnya dapat menggangu proses produksi akibat relasi perusahaan dengan pemasok yang tidak berdasar pada hubungan kerjasama yang erat.
   c. Tidak tepat di gunakan pada pemesanan bahan atau barang yang tidak tahan lama atau mudah busuk.

### 2.5.2 Perkiraan Pemakaian

Dalam menentukan pembelian Bahan baku yang akan digunakan selama satu periode harus di perkirakan terlebih dahulu oleh manajemen sebelum kegiatan pembelian bahan baku dilakukan. Ini adalah perhitungan mengenai berapa jumlah barang yang nantinya digunakan untuk kebutuhan selama satu tahun kedepan. Dimana, penentuan pembelian bahan baku bisa di tentukan dari peramalan produksi dimasa yang akan datang

### 2.5.3 Harga Dari Bahan

Dalam kebijakan persediaan bahan baku, harga bahan baku merupakan satu faktor yang penting.berapa besar dana yang akan di keluarkan berdasar dari penyusunan perhitungan harga bahan baku akan di sediakan

sebagai investasi stok bahan baku dan biaaya modal yang digunakan juga di perhitungkan.

### 2.5.4 Biaya Persediaan

Dalam penentuan besarnya persediaan bahan baku sudah seharusnya menghitungkan biaya penyelenggaraan persediaan bahan baku. karena mempunyai hubungan dengan biaya persediaan.

### 2.5.5 Biaya Penyimpanan (*Holding Cost* atau *Carrying Cost*)

Biaya ini dikeluarkan perusahaan karena adanya penyimpanan bahan di dalam gudang atau tempat penyimpanan. Biaya yang di keluarkan berbeda-beda mengikuti jumlaj rata-rata persediaan yang sudah ada. Biasanya biaya ini ditentukan sebagai suatu porsentase nilai uang dari persediaan per unit dalamsatu tahun. Semakin besar biaya penyimpanan maka semakin besar jumlah bahan yang disimpan

### 2.5.6 Biaya Pemesanan (*Ordering Cost*)

Ini mencakup biaya semua biaya mulai dari mengadakan persiapan pesanan sampai dengan barang yang dipesan datang. Sifat dari biaya ini adalah relatif konstan, tidak tergantung besarnya jumlah barang yang dipesan. Jadi semakin tinggi frekuensi pemesanan, maka biaya ini akan semakin besar

### 2.5.7 Waktu Tunggu (*Lead Time*)

*Lead time* atau waktu tunggu merupakan jumlah waktu yang dibutuhkan perusahaan dari mulai awal proses pemesanan barang sampai dengan barang sampai di perusahaan. Ini penting karena menyangkut penentuan waktu atau titik pemesanan yang akan dilakukan kembali . apabila perusahaan dapat memperhitungkan waktu pemesanan kembali dengan baik, perusahaan akan dapat dengan mudah menentukan waktu yang tepat untuk pesan kembali ke supplier. Hal ini akan mengurangi kekurangan atau kelebihan stok di dalam perusahaan

### 2.5.8 Persediaan Pengaman (*Safety Stock*)

Persediaan pengaman adalah jumlah stok yang digunakan perusahaan dengan maksut untuk mengurangi resiko apabila terjadi kekurangan barang di gudang, bisa dikarenakan karena keterlambatan pengiriman atau jika terjadi kerusakan barang pada saat di simpan. Hal ini diharapkan bisa meminimalisir kekurangan stok agar perusaahaan tidak mengecewakan pelanggan. Tidak semua jenis barang bisa di jadikan *safety stock.* Perusahaan akan tetap mempertahankan jumlah persediaan meskipun nantinya dapat berganti dengan jenis yang baru [2]

Didalam *Economic Order Quantity* (EOQ) Ada beberapa penyebab yang menjadi pengaruh penting yaitu:

1. Biaya penyimpanan perunit
2. Biaya pemesanan tiap kali pesan
3. Kebutuhan bahan baku untuk suatu periode tertentu
4. Harga bahan baku

Berdasarkan unsur tersebut maka EOQ bisa digunakan rumus perhitungan

$$\text{EOQ} = \sqrt{\frac{2.Co.D}{h}} \qquad (2.1)$$

Keterangan:

EOQ : Jumlah pemesanan ekonomis

Co : Biaya Sekali Pesan

D : Kebutuhan Tahunan

h : Biaya Penyimpanan

Untuk perhitungan persediaan pengaman dapat dilakukan dengan rumus

ROP = ( Permintaan per hari) x (lead time) = d x l

$$D = \frac{D}{Jumlah\ periode\ waktu\ pertahun} \quad (2.2)$$

Perusahaan- perusahaan menetapkan kebijakan dalam menentukan titik pemesanan ulang, sebagai berikut

1. Menentukan jumlah penggunaan selama lead time, yaitu waktu mulai barang dipesan sampai barang datang.
2. Menetapkan jumlah penggunaan selama lead time ditambah penggunaan selama periode tertentu sebagai *safety stok.*
3. Penetapan lead time dengan biaya yang ekonomis atau minimum

## 2.6 PHP

Adalah sebuah Bahasa progam dalam web yang mempunyai sifat serverside, artinya semua Bahasa pemrogaman yang berjenis script yang di jalankan dan disimpan di server computer akan menjadi sebuah script HTML di computer client.

Berikut ini adalah karakter script PHP [3]

1. Script PHP biasanya diawali dengan tag '<?' atau '<?php' dan ditutup dengan tag '?>'
2. File PHP dapat menginduk atau disisipkan pada bahasa script lainnya atau dapat berdiri sendiri
3. file PHP disimpan dengan extensi filenya yaitu : *.php3, *.php4, *.php.

### 2.6.1 Variabel PHP

Di dalam Bahasa pemrogaman PHP semua variable dimulai dengan tanda $. Sebagai contoh misal ingin memberi variable D maka di tulis $D. pada saat progam dijalankan otomatis akan menentukan jenis variable dan mengikuti konteks apa yang akan di pakai.

Pembuatan aplikasi web database merupakan hal penting utama pada saat PHP ingin mengambil atau melakukan apa yang telah pemrogaman berbasis CGI yang lain. Produk perangkat lunak DBMS bisa di dukung dengan PHP yang berjalan pada sistem operasi Linux ataupun Windows [3]. Ada beberapa DBMS yang bisa dijalankan dengan PHP berikut adalah jenis-jenisnya

1. Adabas
2. D Ingres
3. Oracle (OCI7 and OCI8)
4. Dbase
5. InterBase
6. Ovrimos
7. Empress
8. FrontBase
9. PostgreSQL FilePro (read-only)
10. mSQL
11. Solid
12. Hyperwave
13. Direct MS-SQL
14. Sybase
15. IBM DB2
16. MySQL
17. Velocis
18. Informix
19. ODBC
20. Unix dbm

### 2.6.2 Struktur Kontrol

Berikut adalah Struktur Kontrol dalam PHP:

1. Continue
2. Break
3. Foreach
4. For
5. do..while
6. if
7. else
8. elseif

### 2.6.3 Tipe Dasar PHP

Berikut ini adalah tipe-tipe dasar php [3]:

1. Boolean dalam mendefinikan suatu kebenaran Boolean adalah tipe yang tepat. Ada dua jenis nilai dalam Boolean yaitu FALSE dan TRUE

```
<?php
        $test = true;
        If ($test == true) {
                Echo "Nilai Tipe Data Boolean TRUE";
        } else {
                Echo "Nilai Tipe Data Boolean FALSE";
        }
?>
```

**Gambar 2.1** Contoh Boolean

2. Integer berfungsi dalam mendefinisikan data yang berbentuk bilangan bulat, tipe Integer mempunyai ukuran batas maksimum dari 32 bit sampai dapat mendifinisan nilai hingga 2 milyar

```
$test = 123;        // desimal
$test = -123;       // negatif desimal
$test = 0123;       // octal
$test = 01A;        // hexadecimal
```

**Gambar 2.2** Contoh Integer

3. Float digunakan dalam mengartikan jenis data ke bentuk bilangan decimal atau pecahan

```
$test = 1.234;
$test = 1.2e3;
```

**Gambar 2.3** Contoh Float

4. Double digunakan dalam mengartikan jenis data ke bentuk bilangan decimal atapun pecahan
5. String digunakan dalam mengartikan jenis data ke dalam bentuk karakter dan teks

```
<?php
        $test = "Pemrograman Web dengan PHP";
        Echo " variable test = $test";
?>
```

**Gambar 2.4** Contoh String

### 2.6.4 Operator Dalam PHP

Operator digunakan dalam mengolah nilai.beberapa kategori operator PHP adalah seperti di bawah ini [3]:

1. Operator Hitung

| Operator | Penjelasan | Contoh | Hasil |
|---|---|---|---|
| + | Pertambahan | x=2<br>y=x+2 | y=4 |
| - | Pengurangan | x=2<br>y=5-x | y=3 |
| * | Perkalian | x=4<br>y=x*5 | y=20 |
| / | Pembagian | y=15/5 | y=3 |
| % | Sisa hasil bagi | x=10%5<br>y=10%8<br>z=5%2 | x=0<br>y=2<br>z=1 |
| ++ | Inkremen | x=5<br>x++ | x=6 |
| -- | dekremen | x=5<br>x-- | x=4 |

**Gambar 2.5** Operator Hitung

2. Operator Perbandingan

| Operator | Penjelasan | Contoh |
|---|---|---|
| == | sama dengan | 5==8 mengembalikan nilai *false* |
| != | tidak sama dengan | 5!=8 mengembalikan nilai *true* |
| <> | tidak sama dengan | 5<>8 mengembalikan nilai *true* |
| > | lebih besar dari | 5>8 mengembalikan nilai *false* |
| < | lebih kecil dari | 5<8 mengembalikan nilai *true* |
| >= | lebih besar dari atau sama dengan | 5>=8 mengembalikan nilai *false* |
| <= | lebih kecil dari atau sama dengan | 5<=8 mengembalikan nilai *true* |

**Gambar 2.6** Operator Perbandingan

3. Operator Logika

| Operator | Nama | Contoh | Penjelasan |
|---|---|---|---|
| ! | Not | !$a | true jika $a false |
| && | And | $a && $b | true jika $a dan $b true |
| \|\| | Or | $a \|\| $b | true jika $a atau $b true |

**Gambar 2.7** Operator Logika

## 2.7 Database

Pengertian mengenai basis data adalah data yang diorganisasi dan di integerasi dan bertujuan dalam memenuhi kebutuhan user di suatu organisasi.basis data mempunyai konsep yaitu sebagai sebuah sistem penyimpanan berkas data yang saling terpadu dan sebagai kumpulan data yang saling terpadu. Adapun bebrapa keuntungan menggunakan basis data yaitu data kerangkapan lebih terkontrol, data lebih terorientasi dan lebih luwes. Namun juga ada beberapa kelemahan nya diantaranya adalah kerusakan dalam sistem basis data dapat mempengaruhi departemen lain yang terkait, harga software yang cukup mahal, dibutuhkan tenaga spesialis, dan juga butuh penyimpanan yang besar. [4]

Dibawah ini adalah paparan istilah dalam basis data yang sering di gunakan

1. Enterprise merupakan arti dari bentuk badan usaha atau objek. Misalnya seperti Perusahaan, rumah sakit, universitas, atau sekolah dll. Data oprasional dari sebuah interprise adalah data yang nantinya akan disimpan di dalam database. Dibawah ini merupakan contoh data operasional
   a. data siswa
   b. data mahasiswa
   c. data orang tua
   d. data keuangan
   e. data karyawan
2. Entitas merupakan kata yang berarti obyek yang berbeda-beda serta obyek yang unik dan dapat di terapkan didalam sebuah *database*.Berikut adalah :
   a. Dibawah ini adalah contoh entitas di lingkungan universitas :
      1) Mahasiswa

2) Dosen
3) Fakultas
4) Jurusan

b. Berikut adalah contoh entitas dalam lingkup perusahaan :
1) Karyawan
2) *department*
3) keuangan

himpunan entitas merupakan entitas-entitas yang terdapat di dalam sebuah basis data. Misal, Seluruh murid, semua mahasiswa.

3. Sebuah entitas memiliki beberapa karakteristik yaitu atribut atau bisa disebut sebagai elemen data. Sebagai contoh entitas nama murid yang mempunyai nilai data: nomor absen, nama murid, nilai ujian dan alamat murid
4. Data Value atau nilai data adalah nilai atau isi yang terdapat dari satu data yang terdapat didalam setiap elemen data: contoh Rifqi, Acun, Affandy
5. Key Data Elemen atau Kunci Elemen Data merupakan tanda pengenal yang mempunyai karkter sendiri atau bersifat unik berfungsi untuk mengenali atau mengidentifikasi dari sebuah entitas: Misalnya entitas Pegawai Negeri Sipil yang memilikiiatribut NIP, Namaipegawai, serta bidang pekerjaan, bisa memakai NIP untuk kata kunci elemen data[5].

Kumpulan isi elemen data atau atribut yang saling berhubungan disebut Record Data. Contohnya kumpulan Atribut IPK, Alamat Mahasiswa, Nama Mahasiswa, dan NIM berisikan"A12.2013.04997", "Rifqi", ". Jepara"

## 2.8 UML

Didalam dunia industry Unified Modelling Language (UML) sudah menjadi standar untuk dokumentasi, merancang dan mevisualisasikan sistem perangkat lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

UML merupakan standart Bahasa yang biasa di gunakan di dunia perindustrian untuk menggambar arsitektur dalam pemrogaman berorientasi objek, desain,

membuat analisis serta mendefinisikan *requitmen*. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram[6].

Berikut ini adalah penjelasan mengenai berbagai diagram dasar dalam UML, yaitu:

1. *Use Case Diagram*

   Dalam membuat sistem informasi yang akan dibuat *usecase Diagram* akan digunakan duntuk kelakuan *(Behavior)*. Dengan sistem informasi yang akan dibuat *usecase* akan mendeskripsikan sebuah interaksi satu atau lebih. Secara kasar *usecase* berfungsi dalam mengetahui siapa saja yang yang berhak menggunakan setiap fungsi dan fungsi apa saja yang ada di dalam sebuah sistem informasi.

   Kriteria dalam penamaan *usecase* adalah nama dapat mudah dipahami dan di definisikan sesimpel mungkin. Berikut adalah dua hal utama pada *usecase*:

   a. Aktor

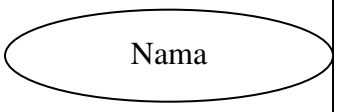
      Aktor bisa berarti manusia, sistem atau proses lain yang berhubungan dengan sistem informasi yang dibuat di luar sistem informasi itu sendiri

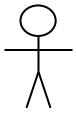
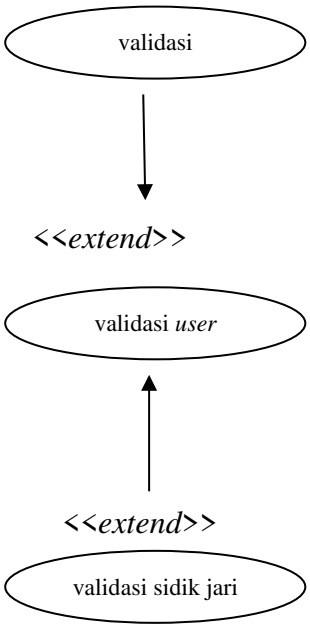
   b. *Usecase*

      Adalah fungsi yang sudah ada karena sudah disediakan sistem sebagai unit-unit yang saling bertukar pesan antara aktor atau unit

   Dibawah akan dijelaskan simbol-simbol diagram usecase:

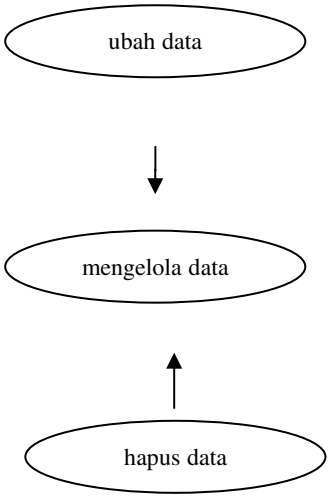
**Table 2.2** Simbol Use Case

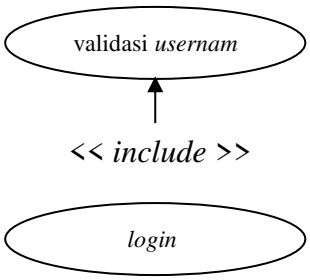
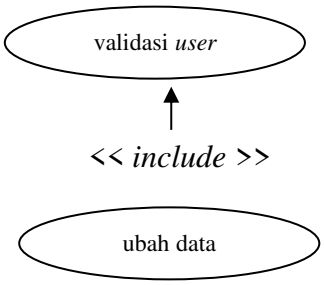
| Nama dan Simbol | Keterangan |
|---|---|
| *Usecase*<br>  <br> | fungsionalitas yang disediakan sistem sebagai *unit-unit* yang saling bertukar pesan antar aktor atau unit.<br>*Usecase* biasanya dinyatakan dengan menggunakan kata kerja diawal fase |

| | |
|---|---|
| | *usecase*. |
| Aktor/*actor*  | Sistem atau Proses lain yang berhubungan dengan sistem informasi yang akan dibuat diluar sistem informasi itu sendiri, meskipun symbol aktor berbentuk orang bukan berarti itu orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor. |
| Assosiasi/*Association* _____ | Komunikasi antara *usecase* dengan aktor yang berpartisipasi pada *usecase* atau *usecase* memiliki interaksi dengan aktor |
| Ekstensi/*Extend* ----------> << *extend* >> | Relasi *usecase* tambahan ke sebuah *usecase* dimana *usecase* yang ditambahkan dapat berdiri sendiri walau tanpa *usecase* tambahan itu mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek<br>Biasanya *usecase* memiliki nama depan yang sama dengan *usecase* yang ditambahkan, contoh<br>  |

| | |
|---|---|
| | Arah panah pada *usecase* yang ditambahkan; biasanya *usecase* yang menjadi *extend*-nya merupakan jenis yang sama dengan *usecase*yang menjadi induknya. |

**Table 2.3** Lanjutan Simbol UseCase

| Nama dan Simbol | Deskripsi |
|---|---|
| Generalisasi/ *Generalization* | Hubungan sepesialisasi serta generalisasi (umum-khusus) antara dua buah *usecase* dimana fungsi yang satu merupakan fungsi yang lebih umum dari yang lain, Contoh:   Arah panah mengarah pada usecase yang menjadi generalisasinya |
| Menggunakan/ *Include/Uses* ----------> << *include* >> << *uses* >> | Relasi *usecase* tambahan ke sebuah *usecase* dimana *usecase* yang ditambahkan memerlukan *usecase* ini untuk menjalankan fungsinya atau sebagai syarat dijalankannya *usecase* ini<br>Include yang ada pada usecase mempunyai dua sudut pandang yang cukup besar[7] :<br>• *include* berarti *usecase* yang ditambahkan akan selalu dipanggil saat *usecase* tambahan |

| | |
|---|---|
| | dijalankan misal pada kasus berikut :<br>  <br><br>*include* adalah *usecase* yang tambahan akan selalu melakukan pengecekan apakah *usecase* yang ditambahkan telah dijalankan sebelum usecase tambahan dijalankan, contoh pada kasus berikut :<br>  <br><br>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan. |

### 2.8.1 Activity Diagram

*Activity* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah Proses ataupun sistem bisnis atau menu yang ada pada software. Hal yang perlu diperhatikan bahwa *activity* diagram menggambarkan aktivitas sistem bukan apa yang aktor kerjakan, jadi aktivitas yang dapat dilakukan oleh sistem.

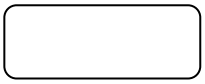
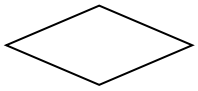
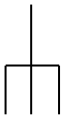
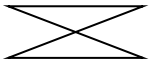
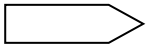
*Activity* diagram biasanya dipakai untuk mendefinisikan hal-hal berikut[8]:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.

2. Urutan atau pengelompokan tampil dari sistem atau *user interface* dimana setiap, aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan *menu* yang ditampilkan pada perangkat lunak

Berikut adalah simbol-simbol yang ada pada *activity* diagram

**Table 2.4** Activity Diagram

| Simbol | Nama Notasi | Keterangan |
|---|---|---|
| | Titik Awal | Titik dimulainya suatu aktivitas. |
|  | *Activity* | Memuat bagaimana masing – masing kelas antarmuka berinteraksi |
|  | *If* | Pilihan untuk mengambil keputusan |
| | *Fork* | Menggabungkan dua kegiatan menjadi satu |
|  | *Rage* | Menunjukkan adanya dekomposisi |
|  | Tanda waktu | Mengindikasikan waktu |
|  | Tanda Pengiriman | Tanda Pengiriman |
| | Tanda Penerimaan | Tanda Penerimaan |

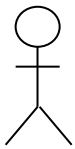
| | | |
|---|---|---|
| | Aliran Akhir | Aliran Akhir |
| | Titik Akhir | Titik berhentinya suatu aktivitas |

### *2.8.2 Squence Diagram*

Diagram *sequence* menggambarkan kelakuan objek pada *usecase* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirim dan diterima antar objek. Menggambar diagram *sequence* harus diketahui objek yang terlibat dalam sebuah *usecase* serta metode yang dimiliki kelas yang diinstansiasi menjadi objek itu, dalam membuat diagram *sequence* juga dibutuhkan untuk melihat skenario yang ada pada *usecase*[9]. Banyaknya diagram *sequence* yang harus digambar adalah minimal sebanyak pendefinisian *usecase* yang memiliki proses sendiri atau yang penting semua *usecase* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram *sequence* sehingga semakin banyak *usecase* yang didefinisikan maka diagram *sequence* yang dibuat juga semakin banyak.

Berikut adalah simbol-simbol yang ada pada diagram *sequence:*

**Table 2.5** *Diagram Squence*

| Simbol | Deskripsi |
|---|---|
| Aktor<br>  <br>Nama aktor<br>Atau | Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan |

| | |
|---|---|
| Nama<br><br>Tanpa waktu aktif | menggunakan kata benda di awal frase nama actor. |
| Garis hidup atau *lifeline*<br><br>…… | Menyatakan kehidupan suatu objek. |
| Objek<br>Nama objek:nama kelas | Menyatakan objek yang berinteraksi pesan |
| Waktu | Menyatakan objek dalam posisi berinteraksi dan aktif, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, contoh<br><br>2: cek status *login*()<br>1: *login*()<br>3: *open*() |

| | |
|---|---|
| | Maka cek status *login* dan *open* dilakukan dalam metode *login*<br><br>Aktor tidak memiliki waktu aktif . |
| Pesan tipe *create*<br><br><< *create* >> | Arah panah mengarah pada objek yang dibuat dan menyatakan suatu objek yang lain. |

**Table 2.6** Lanjutan Diagram *Sequence*

| Simbol | Deskripsi |
|---|---|
| Pesan tipe *call* | Menyatakan apabila objek melakukan panggilan operasi atau dirinya sendiri atau metode yang ada pada objek,<br><br>1: nama_metode ()<br><br>Arah panah menunjuk ke objek yang mempunyai operasi (metode), ini karena memanggil operasi (metode) maka operasi (metode) yang diundang harus sudah ada di dalam kelas diagram mengikuti kelas objek yang terhubung. |

| | |
|---|---|
| Pesan tipe *send*<br><br>1: masukan<br>⟶ | Menjelaskan , arah panah mengarah pada objek yang dimiliki |
| Pesan tipe *return*<br><br>1: keluaran<br>----------> | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah menunjuk kepada objek yang nantinya akan menerima kembalian. |
| Pesan tipe *destroy* | Untuk memutuskan atau mengakhiri objek lain yang pernah ada, sebaiknya jika terdapat *create* berarti ada *destroy* |

### *2.8.3 Class Diagram*

*Class diagram* atau biasa disebut diagram kelas, digunakan untuk menggambarkan struktur sistem dari segi pendefinisian kelas - kelas yang akan dibuat untuk membangun sistem. Didalam diagram kelas terdapat operasi atau metode serta atribut[10].

**Table 2.7** Simbol Class Diagram

| Simbol | Deskripsi |
|---|---|
| Kelas<br>Nama kelas<br>+atribut<br>+operasi () | Kelas pada struktur sistem |

| | |
|---|---|
| Antarmuka ( *interface* )<br>○<br>Nama_*interface* | Sama dengan konsep *interface* dalam pemrograman berorientasi objek. |
| Asosiasi ( *association* )<br>———— | Hubungan ke setiap kelas , dan biasanya *multiplicy* selalu menyertai asosiasi |
| Asosiasi berarah ( *directed association* )<br>——→ | Hubungan setiap kelas dengan makna kelas lain telah dipakai kelas lain, *multiplicity* mengikuti asosiasi yang sudah ada. |
| Generalisasi<br>——▷ | Hubungan antar kelas yang berarti spesialisasi-generalisasi (khusus-umum) |
| Kebergantungan (*dependency*)<br>——→ | Ini berarti kelas yang ada saling terhubung atau bergantung |
| Agregasi (*aggregation*)<br>——◇ | Relasi antar kelas denngan makna semua bagian (*whole part*) |