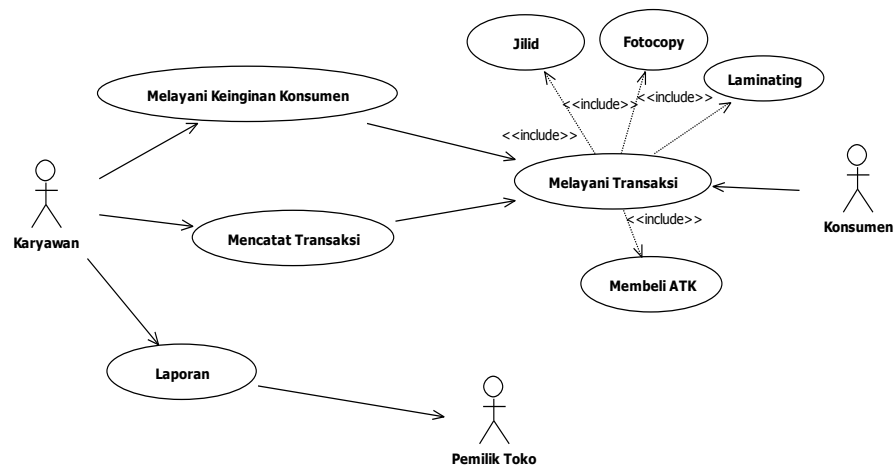


## BAB III PERMASALAHAN PERUSAHAAN

### 3.1 Analisa Permasalahan yang Dihadapi Oleh Perusahaan

Pengolahan data di perusahaan masih memakai metode yang klasik dan konvensional dengan cara mencatat di buku. Metode ini yang masih menyulitkan pemilik perusahaan dalam memonitoring data transaksi penjualan alat tulis kantor serta jasa fotocopy.

#### 3.1.1 Usecase sistem yang sedang berjalan :



**Gambar 3.1** Usecase sistem yang sedang berjalan

#### Analisis sistem penjualan yang sedang berjalan

Untuk merancang suatu sistem baru maka seorang perancang sistem harus mengetahui terlebih dahulu alur sistem yang sedang berjalan saat ini dan kekurangan yang ada didalam sistem yang sedang berjalan saat ini,. Adapun alur sistem dan kekurangan sistem yang sedang berjalan di perusahaan Tata Fotocopy Bandar Lampung adalah sebagai berikut :

1. Konsumen datang untuk membeli alat tulis kantor atau menggunakan jasa seperti fotocopy, jilid dan, laminating.

2. Karyawan toko akan melayani apa yang diinginkan oleh pelanggan seperti membeli alat tulis kantor atau menggunakan jasa seperti fotocopy, jilid dan, laminating.
3. Setelah toko tutup karyawan akan mencatat transaksi yang ada dihari ini dan pendapatan atau pengeluaran toko yang ada dihari ini.
4. Pemilik toko dapat melihat laporan ketika toko tutup, jika ingin melihatnya sebelum toko tutup pemilik toko harus datang ke toko atau ke lokasi.

**a. Kekurangan sistem yang sedang berjalan**

Sistem yang sedang berjalan saat ini memiliki banyak kekurangan, antara lain :

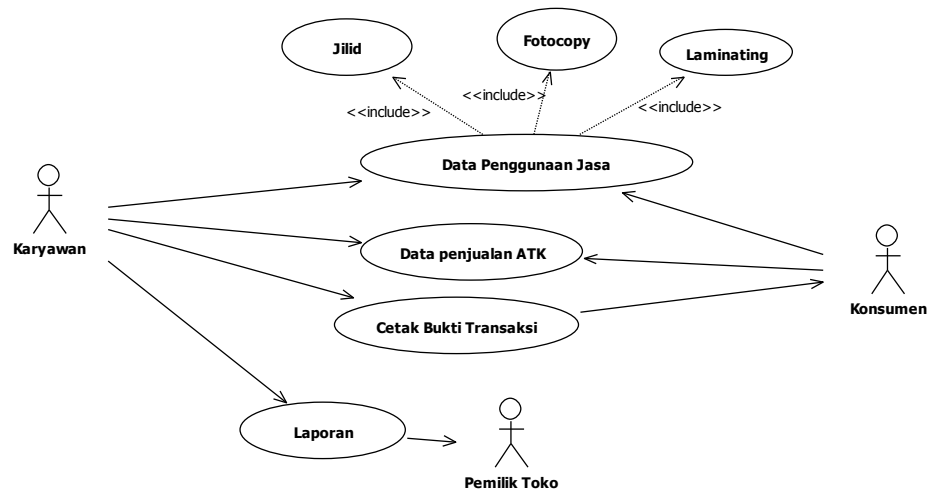
1. Sulit mengolah data transaksi yang banyak untuk dijadikan informasi.
2. Mencari data menjadi sulit karna masih di catat didalam buku.
3. Kontrol/ Monitoring kerap terjadi selisih ketika mencatat.

**3.1.2 Perumusan Masalah**

Berdasarkan temuan masalah yang telah dijelaskan diatas, maka yang menjadi permasalahan ialah sebagai berikut :

- a. Bagaimana Merancang Bangun Monitoring Penjualan UMKM pada Tata Fotocopy Bandar Lampung agar transaksi penjualan alat tulis kantor serta jasa fotocopy dapat menjadi efektif dan efisien.

### 3.1.3 Kerangka Pemecahan Masalah



Gambar 3.2 Kerangka Pemecahan Masalah

## 3.2 Landasan Teori

### 3.2.1 Konsep Dasar Sistem

Sistem dapat didefinisikan sebagai suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berinteraksi untuk mencapai suatu tujuan, sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*systema*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi.

#### A. Karakteristik Sistem

Suatu sistem mempunyai beberapa karakteristik, yaitu:

##### 1. Komponen Sistem (*Components*)

Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian sistem, yang mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem keseluruhan.

##### 2. Batas Sistem (*Boundary*)

Batas sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environments*)

Lingkungan luar (*environments*) dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan merugikan sistem.

4. Penghubung Sistem (*Interface*)

Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem lainnya sehingga memungkinkan sumber-sumber daya mengalir antara subsistem yang satu dengan yang lain.

5. Masukan Sistem (*Input*)

Masukan (*input*) adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

6. Keluaran Sistem (*Output*)

Keluaran (*output*) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem.

7. Pengolah Sistem (*Process*)

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan jadi keluaran.

8. Sasaran Sistem (*Objectives*)

Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

## **B. Klasifikasi Sistem**

Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut :

Sistem diklasifikasikan sebagai sistem abstrak (*abstract system*) dan sistem fisik (*physical system*). Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem teologia dan sistem fisik merupakan sistem yang ada secara fisik, misalnya sistem komputer.

Sistem diklasifikasikan sebagai sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*). Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia, misalnya sistem perputaran bumi dan sistem buatan manusia adalah sistem yang dirancang oleh manusia.

Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*). Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi dan sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

Sistem diklasifikasikan sebagai sistem tertutup (*closed system*) dan sistem terbuka (*open system*). Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya dan sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya.

### 3.2.2 Informasi

Informasi adalah data yang telah di olah dan di organisasi melalui suatu proses dan dengan maksud tertentu. Data yang sudah ada dikemas dan diolah sedemikian rupa sehingga menjadi sebuah informasi yang berguna. Sebagai contoh bila kita memasukan nama-nama mahasiswa dengan nilai rata-rata, nama-nama konsumen dengan saldo bank, jumlah gaji dengan jumlah jam kerja, kita akan mendapatkan informasi yang berguna. Berikut adalah pembahasan definisi informasi berdasarkan berbagai sumber.

### 3.2.3 Diagram UML

Untuk mendapatkan banyak pandangan terhadap sistem informasi yang akan dibangun, UML menyediakan beberapa diagram visual yang menunjukkan berbagai aspek dalam sistem. Ada beberapa diagram yang disediakan dalam UML antara lain :


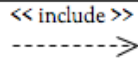
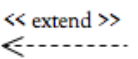



#### A. Use Case Diagram

*Use case* atau diagram *Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendefinisikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu, (Rosa A.S, M.Shalahuddin, 2013).

syarat penamaan pada use care adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut aktor dan *use case*.

- Aktor merupakan orang, proses, atau sistem laen yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar kesan antar unit atau aktor.

Berikut adalah simbol-simbol yang ada diagram *use case*:

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
3		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
4		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
5		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
6		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

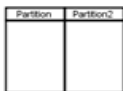





## B. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem, (Rosa A.S, M.Shalahuddin, 2013).

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

- Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antar muka tampilan.
- Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
- Rancangan menu yang ditampilkan pada perangkat lunak.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Swimlane</i>	Menunjukkan siapa yang bertanggung jawab dalam melakukan aktivitas dalam suatu diagram.
2		<i>Action</i>	Langkah-langkah dalam sebuah activity. Action bisa terjadi saat memasuki activity, meninggalkan activity, atau pada event yang spesifik.
3		<i>Initial State</i>	Menunjukkan dimana aliran kerja dimulai.
4		<i>Activity Final Node</i>	Menunjukkan dimana aliran kerja diakhiri.
5		<i>Decision Node</i>	Menunjukkan suatu keputusan yang mempunyai satu atau lebih transisi dan dua atau lebih transisi sesuai dengan suatu kondisi.
6		<i>Control Flow</i>	Menunjukkan bagaimana kendali suatu aktivitas terjadi pada aliran kerja dalam tindakan tertentu.

### C. Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendefinisikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use*



case beserta metode-metode yang dimiliki kelas yang diinstalasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak, (Rosa A.S, M.Shalahuddin, 2013).

Berikutnya adalah simbol-simbol yang ada pada Sequence Diagram:

<p><b>aktor</b></p>  <p>atau <span style="border: 1px solid black; padding: 2px;">nama_aktor</span></p>	<ul style="list-style-type: none"> <li>• orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi dan mendapat manfaat dari sistem.</li> <li>• Berpartisipasi secara berurutan dengan mengirimkan dan / atau menerima pesan.</li> <li>• Ditempatkan di bagian atas diagram.</li> </ul>
<p><b>objek</b></p> <span style="border: 1px solid black; padding: 2px;">objek:kelas</span>	<p><b>Sebuah objek:</b></p> <ul style="list-style-type: none"> <li>• Berpartisipasi secara berurutan dengan mengirimkan dan / atau menerima pesan.</li> <li>• Ditempatkan di bagian atas diagram.</li> </ul>
<p><b>Garis hidup objek</b></p> 	<ul style="list-style-type: none"> <li>• Menandakan kehidupan obyek selama urutan.</li> <li>• diakhiri tanda X pada titik di mana kelas tidak lagi berinteraksi.</li> </ul>
<p><b>Objek sedang aktif berinteraksi</b></p> 	<p><b>Fokus kontrol:</b></p> <ul style="list-style-type: none"> <li>• Adalah persegi panjang yang sempit panjang ditempatkan di atas sebuah garis hidup.</li> <li>• Menandakan ketika suatu objek mengirim atau menerima pesan.</li> </ul>
<p><b>pesan</b></p> 	<p>objek mengirim satu pesan ke objek lainnya</p>
<p><b>&lt;&lt;create&gt;&gt;</b></p> 	<p>menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
<p><b>1:masukan</b></p> 	<p>menyatakan bahwa suatu objek mengirimkan masukan ke objek lainnya arah panah mengarah pada objek yang dikirim</p>
<p><b>- 1:keluaran - -</b></p> 	<p>objek/metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian</p>
<p><b>destroy()</b></p> 	<p>menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy</p>

## D.Class Diagram

Diagram kelas atau *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, (Rosa A.S, M.Shalahuddin, 2013).

- Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
- Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

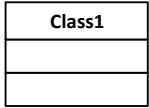
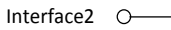


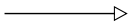
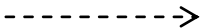

Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan didalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Banyak berbagai kasus, perancangan kelas yang dibuat tidak sesuai dengan kelas-kelas yang dibuat pada perangkat lunak, sehingga tidaklah ada gunanya lagi sebuah perancangan karena apa yang dirancangan dan hasil jadinya tidak sesuai.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak atau *programmer* dapat membuat kelas-kelas didalam program perangkat lunak sesuai dengan perancangan diagram kelas. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

- Kelas main  
Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
- Kelas yang menangani tampilan sistem (*view*)  
Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.
- Kelas yang diambil dari pendefinisian *use case* (*controller*)  
Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use care*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

- Kelas yang diambil dari pendefinisian data (*model*)

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data. Semua tabel yang dibuat dibasis data dapat dijadikan kelas, namun untuk tabel dari hasil relasi atau atribut multivalued pada ERD dapat dijadikan kelas tersendiri dapat juga tidak asalkan pengaksesannya dapat dipertanggung jawabkan atau tetap ada didalam perancangan kelas.

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem.
Natarmuka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek.
Asosiasi 	Relasi antar kelas dalam makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
Kebergantungan 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi 	Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> ).

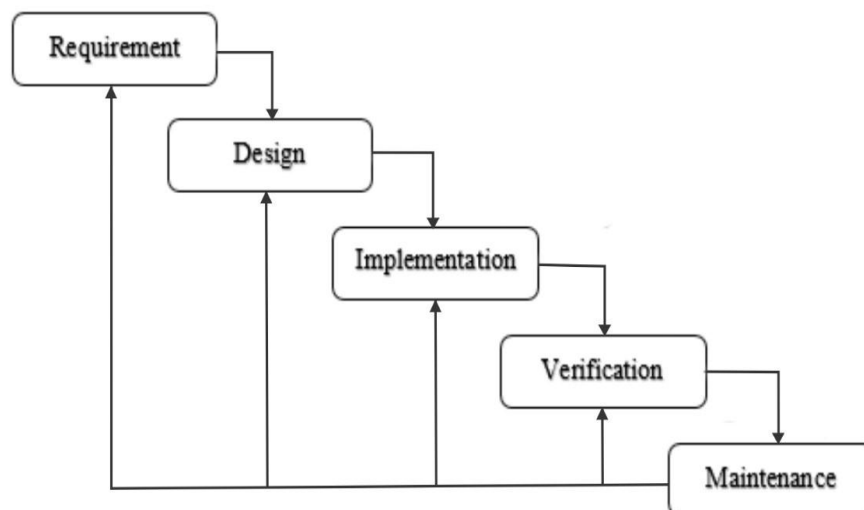
### 3.3 Metode Pengembangan Sistem

#### Model Waterfall (Air Terjun)

Metode pengembangan sistem sangat dibutuhkan dalam perancangan sebuah sistem karena sebelum memulai pembuatan koding – koding hendaknya merancang terlebih dahulu metode pemodelan seperti apa yang harus digunakan dengan memprioritaskan ketepatan waktu selesai dan efektifitas dalam perancangan sebuah sistem.(Rosa A.S, M.Shalahuddin, 2016).

Model air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model *waterfall* menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).

Berikut adalah gambar model *waterfall*:



**Gambar 3.3** model *waterfall*

### 3.3.1 Tahap-tahap metode WATERFALL

1. *Requirement system*

Proses pengumpulan kebutuhan dilakukan secara intensif untuk memesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. *System design*

Desain perangkat lunak adalah proses multi langkah yang focus pada desain pembuatan program perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yagn dihasilkan pada tahap ini juga perlu didokumentasikan.

3. *Implementation*

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. *Integration & testing*

Pengujian focus pada perangkat lunak secara dari logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yagn dihasilkan sesuai dengan yang diinginkan.

5. *Operation and Maintenance*

Dalam tahapan ini, sistem diinstal dan mulai digunakan. Selain itu juga memperbaiki *error* yang tidak ditemukan pada tahap pembuatan. Dalam tahap ini juga dilakukan pengembangan sistem seperti penambahan fitur dan fungsi baru.

### **3.3.2 Keunggulan dan Kelemahan Metode Waterfall**

Metode pengembangan waterfall mempunyai keunggulan dalam membangun dan mengembangkan suatu sistem, antara lain:

1. Kualitas dari sistem yang dihasilkan akan baik. Ini dikarenakan oleh pelaksanaannya secara bertahap. Sehingga tidak terfokus pada tahapan tertentu.
2. Dokumen pengembangan sistem sangat terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya. Jadi setiap fase atau tahapan akan mempunyai dokumen tertentu.

Dalam proses membangun dan mengembangkan suatu sistem, metode waterfall mempunyai beberapa kelemahan, antara lain:

1. Diperlukan manajemen yang baik, karena proses pengembangan tidak dapat dilakukan secara berulang sebelum terjadinya suatu produk..
2. Kesalahan kecil akan menjadi masalah besar jika tidak diketahui sejak awal pengembangan.
3. Pelanggan sulit menyatakan kebutuhan secara eksplisit sehingga tidak dapat mengakomodasi ketidakpastian pada saat awal pengembangan

### **3.4. Metode Pengumpulan Data**

Dalam melaksanakan kerja praktek ini, penulis metode studi pustaka dan perancangan sistem yang bersangkutan yaitu :

#### **A. Studi Pustaka**

Studi pustaka diterapkan dengan melakukan membaca laporan kerja praktek mahasiswa atau jurnal yang ada di perpustakaan dan didalam internet.

#### **B. Observasi**

Observasi dilakukan selama 30 hari dari tanggal 21 September 2019 sampai dengan 21 Oktober 2019. Proses penelitian dilakukan dengan

melakukan pengamatan langsung terhadap objek dan aktivitas yang ada didalam Tata Fotocopy. Observasi dilakukan untuk memperoleh data dan informasi mengenai sistem yang akan dikembangkan secara detail dan akurat. Selain itu, melalui observasi juga dapat memperoleh gambaran langsung terhadap alur kerja sistem atau aktivitas sistem yang sedang berjalan secara jelas.

