# BAB III

# OBJEK DAN METODOLOGI PENELITIAN

## 3.1 Objek Penelitian

Menurut Umi Narimawati (2010) objek penelitian merupakan hal awal (suatu permasalahan) yang harus ditentukan dalam kegiatan penelitian sehingga penelitian dapat dilakukan secara efektif dan efisien sesuai dengan tujuan penelitian. Dalam artian objek penelitian merupakan variabel yang akan dikaji, yang menjadi fokus penelitian disertai keberadaan/tempat dilakukannya objek penelitian tersebut".

Tempat yang menjadi objek penelitian dalam penyusunan skripsi ini adalah sebuah toko pakaian, yakni Underground Workshop yang berlokasi di Ambon Plaza, Jl. Sultan Hairun No.126 Lantai 1, Blok B, No.4, Ambon, Maluku.

### 3.1.1 Sejarah Singkat Perusahaan

Awalnya ketika Erwin Busmar yang masih kuliah semester 3. Kemudian terjadi konflik SARA di Ambon pada tahun 1999. Rumah, toko milik orangtua habis terbakar. Tidak ada barang yang tersisa, orangtua tidak memiliki uang, uang untuk membeli makan saja susah karena terjadinya konflik tersebut. Akhirnya Erwin, disapa panggilan akrabnya mencoba berjualan asongan dengan uang tabungan harian Rp.200.000. Dengan modal Rp.200.000, Erwin berjualan rokok dan air mineral di pelabuhan kapal demi meringankan beban orangtua.

Pada tahun 2002 dengan modal Rp.2.000.000, Erwin mulai berangkat ke Jogja dengan menggunakan kapal untuk membeli barang disana. Pada waktu itu tariff tiket kapal gratis dikarena banyaknya pengungsi. Waktu di Jogja,

Erwin mengaambil produk DAGADU yang pada tahun itu barangnya masih trend.

Setahun kemudian Erwin beralih dagangan  dengan berjualan kaset VCD sebagai pedagang kaki lima dgn modal Rp.5.000.000. Pada tahun 2004, Erwin telah mendapatkan keuntungan sebesar Rp.15.000.000. Dengan uang tersebut Erwin mulai mengontrak sebuah toko di Ambon Plaza selama 2 tahun dengan harga Rp.6.000.000 per tahunnya. Uang yang dimiliki oleh Erwin pun menjadi Rp.3.000.000. Dengan sisa uang Rp.3.000.000, Erwin menjajaki kota Bandung, di ajak sama saudara dengan niat sekedar jalan-jalan ke Bandung. Ternyata ketika sampai di Bandung, Erwin memiliki ide, yaitu membeli aksesoris dan baju kaos music kemudian menjualnya di Ambon. Ketika sampai di Ambon, barangnya pun habis terjual, walaupun membutuhkan waktu yang agak lama.

Pada tahun 2005 Erwin kembali pergi ke bandung, pada waktu itu sedang ngetrendnya Distro, Dengan modal awal Rp.30.000.000, Erwin mencoba untuk berdagang Distro, walaupun harganya masih mahal. Alhamdulillah berkat doa orangtua dan kemauan yang dimiliki oleh Erwin, Barang Distro yang dijual oleh Erwin di Ambon pun ramai peminatnya, (waktu itu belum ada saingan). Dalam jangka 2 tahun, Erwin sudah bias membeli 1 rumah dan 1 Toko.

Pada tahun 2009, Erwin mulai menambah dagangannya dengan menjual sepatu, tas, sandal dan topi, hingga sekarang.

### 3.1.2 Visi dan Misi Perusahaan

**Visi**

Ingin memperluas usaha menjadi lebih berkembang.

**Misi**

1. Memberikan pelayanan yang terbaik untuk para konsumen.
2. Mampu menyediakan variasi pilihan pakaian-pakaian yang selalu mengikuti trend masa kini.
3. Mensejahterakan karyawan.
4. Membuka lapangan pekerjaan untuk mengurangi angka pengangguran.

### 3.1.3 Struktur Organisasi Perusahaan

Setiap perusahaan dalam menjalankan tugasnya selalu berusaha menciptakan suatu tata kerja yang baik, teratur dan rapi sebagai alat untuk mencapai tujuan yang telah ditetapkan dan digariskan oleh perusahaan itu sendiri. Tata kerja yang baik, teratur dan rapi diharapkan dapat terwujud dan terlaksana apabila ada struktur organisasi yang baik pula, yaitu struktur organisasi yang sederhana dapat bekerja secara efisien serta memungkinkan adanya pemisahan tugas, tanggung jawab dan wewenang yang jelas pada setiap bagian yang ada dalam perusahaan itu sendiri.

Dengan demikian kesinambungan antar pimpinan dan *staff* dibawahnya dapat tercapai dengan baik. Untuk penggambaran lebih jelasnya dapat dilihat secara detail pada struktur organisasi sebagai berikut:

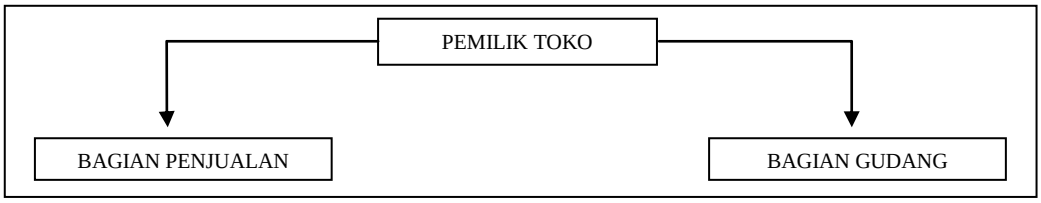
Struktur organisasi pada toko Underground Workshop adalah sebagai berikut:

1) Pemilik Toko

2) Bagian Gudang

3) Bagian Penjualan

Untuk lebih jelasnya mengenai Struktur Organisasi pada Underground Workshop dapat dilihat pada gambar di bawah ini :




**Gambar 3.1 Struktur Organisasi Toko Underground Workshop**

(Sumber : Erwin Busmar, Pemilik Toko Underground Workshop)

### 3.1.4 Deskripsi Tugas

Setiap perusahaan dalam menjalankan tugas dan kegiatannya selalu berusaha mempunyai deskripsi tugas (*job description*) yang baik dan dibuat sendiri oleh perusahaan tersebut. Adapun uraian tugas-tugas, tanggung jawab dan wewenang pokok organisasi toko Underground Workshop adalah sebagai berikut :

1. Pemilik

   Pemilik adalah pimpinan dari toko yang juga sebagai penyandang dana atau modal. Tugas pokok dari seorang pemilik toko Underground Workshop ini diantaranya:

a. Memeriksa setiap transaksi yang berlangsung pada setiap bulannya.
b. Mengatur keuangan toko.
c. Memesan barang ke s*upplier*.
d. Memutuskan menerima atau tidak barang yang ditawarkan oleh *supplier*.

2. Bagian Penjualan

Bagian Penjualan adalah seorang yang dipilih oleh pemilik untuk mengurusi mengenai penjualan barang. Tugas dari bagian penjualan barang di toko Underground Workshop ini diantaranya:

a. Melayani konsumen
b. Melayani transaksi penjualan barang kepada konsumen.
c. Membuat laporan pertanggung jawaban penjualan barang per harinya secara rinci.

3. Bagian Gudang.

Bagian Gudang adalah bagian yang dipilih oleh pemilik untuk mengurusi mengenai pembelian barang. Tugas dari Bagian Gudang di Toko Underground Workshop ini diantaranya:

a. Menerima barang dari *supplier*.
b. Mengecek kondisi barang yang diterim dari *supplier*.
c. Mengembalikan barang ke *supplier* jika ada barang yang rusak.
d. Mengisi barang yang kosong di toko.
e. Membuat laporan pembelian dan retur barang.

## 3.2 Metode Penelitian

Metode penelitian merupakan suatu mekanisme, teknik atau cara untuk mencari, memperoleh, mengumpulkan, atau mencatat data yang dapat digunakan untuk keperluan menyusun karya ilmiah atau penelitian dengan prosedur yang didasarkan pada suatu struktur logis yang terdiri dari beberapa tahapan kerja dan kemudian menganalisa faktor-faktor yang berhubungan dengan pokok-pokok permasalahan sehingga akan didapat suatu kebenaran atas data yang diperoleh.

Suatu penelitian tidak berjalan dengan baik apabila tidak dilakukan dalam suatu proses yang teratur dan terarah. Oleh karena itu diperlukan suatu metodologi untuk melaksanakan suatu penelitian, metodologi yang digunakan pada perancangan sistem informasi penjualan dan pembelian barang pada toko Underground Workshop berdasarkan pada:

### 3.2.1 Desain Penelitian

Desain penelitian adalah keseluruhan proses yang diperlukan dalam proses perencanaan dan pelaksanaan penelitian, sehingga persoalan-persoalan yang ada dapat memperoleh jalan keluarnya. Dalam membangun sebuah sistem informasi yang kompleks membutuhkan metode-metode atau paradigma pengembangan yang mampu membantu menganalisis dan mendesain secara lebih detail sehingga informasi yang dihasilkan lebih akurat.

Penyusunan usulan penelitian ini menggunakan metode pendekatan kualitatif deskriptif, yaitu mengumpulkan data kemudian menganalisanya serta memaparkan hasil pengamatan di lapangan. Dalam penelitian ini, penulis

menetapkan desain penelitian yang lebih luas, yang mencangkup proses-proses berikut ini :

1. Identifikasi masalah
2. Merumuskan masalah
3. Memilih metode pengumpulan data
4. Pelaporan hasil penelitian termasuk proses penelitian dan interpretasikan data.

### 3.2.2 Jenis dan Metode Pengumpulan Data

Jenis data adalah bentuk data penelitian berdasarkan cara mendapatkannya. Sedangkan metode pengumpulan data adalah cara atau jalan yang digunakan oleh peneliti untuk mengumpulkan data dalam penelitian. Adapun cara atau jalan yang digunakan peneliti untuk mengumpulkan data adalah sebagai berikut:

#### 3.2.2.1 Sumber Data Primer

Data primer adalah data yang diperoleh dengan cara melakukan pengamatan langsung ke lapangan (observasi) dan wancara (*interview*) kepada pemilik toko Underground Workshop. Adapun metode pengumpulan data yang dilakukan penulis adalah sebagai berikut :

1. Observasi (*Observation*).

   Observasi adalah pengamatan langsung kepada suatu objek yang akan diteliti, sedangkan penelitian lapangan adalah usaha

pengumpulan data dan informasi secara intensif disertai dengan analisa dan pengujian kembali atas semua data yang telah dikumpulkan.

2. Wawancara *(Interview).*

   Wawancara adalah suatu cara untuk mengumpulkan data dengan mengajukan pertanyaan langsung kepada seorang informan. Disini penulis mewawancarai langsung pemilik dan karyawan dari toko Underground Workshop. Pertanyaan yang penulis ajukan adalah mengenai bagaimana proses bisnis yang berjalan pada toko Underground Workshop.

#### 3.2.2.2 Sumber Data Sekunder

Data sekunder disajikan dalam bentuk dokumentasi. Dokumentasi yang digunakan dalam penelitian ini yaitu, daftar data barang, data penjualan barang atau data pemasukan dari pembelian dan penjualan barang.

### 3.2.3 Metode Pendekatan dan Pengembangan Sistem

Metode pendekatan sistem merupakan salah satu cara penyelesaian persoalan yang dimulai dengan dilakukannya identifikasi terhadap sejumlah kebutuhan, sehingga dapat menghasilkan suatu operasi dari sistem yang dianggap efektif. Sedangkan metode pengembangan sistem terdiri dari kegiatan

yang dapat dikelompokkan menjadi beberapa tahapan, yang membantu kita dalam pengembangan sistem.

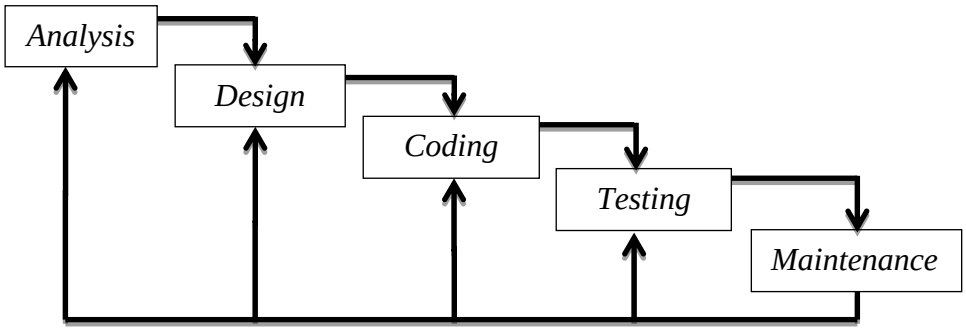
#### 3.2.3.1 Metode Pendekatan Sistem

Metode pendekatan sistem yang penulis gunakan dalam perancangan sistem adalah metode pendekatan sistem yang berorientasi objek (*Object-Oriented*), yang mencakup analisis dan desain yang disebut *OOAD* (*Object Oriented Analysis and Design*). *Object-oriented analysis* adalah metode analisis yang memeriksa *requirements* (syarat/keperluan yang harus dipenuhi suatu sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup permasalahan. Sedangkan *Object-oriented analysis* adalah metode untuk mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek sistem atau subsistem. Alat bantu analisis dan perancangan berorientasi objek adalah *Unified Modelling Language* (*UML*). *UML* adalah sebuah bahasa yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. *UML* menawarkan sebuah standar untuk merancang model sebuah sistem dengan berfokus pada objek.

### 3.2.3.2 Metode Pengembangan Sistem

Metode pengembangan sistem sangat di butuhkan dalam perancangan sebuah sistem, karena sebelum memulai dalam pembuatan koding – koding (*code*) hendaknya merancang terlebih dahulu metode pemodelan seperti apa yang harus di gunakan dengan memprioritaskan ketepaan waktu selesai dan efektifitas dalam perancangan sebuah sistem.

Untuk metode pengembangan sistem, penulis menggunakan model *waterfall*. Model klasik ini bersifat sistematis, berurutan dalam membangun berurutan dalam membangun *software*. Proses dalam model *waterfall* akan berlanjut jika proses sebelumnya telah selesai dilakukan, jika tidak maka proses selanjutnya juga tidak dapat berjalan (prosesnya berjalan secara berurutan). Berikut ini adalah fase-fase dalam model *waterfall* :




**Gambar 3.2 Model *Waterfall Sommerville***

(Sumber :Andy Down Town, McGraw Hill, *Enginneering Computer Interface*)

Berdasarkan gambar 3.2, metode pengembangan menggunakan model *waterfall sommerville* memiliki 5 tahapan, yaitu :

a. *Analysis*

Langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa malakukan sebuah penelitian, wawancara atau *study literatur*. Seorang sistem analis akan menggali informasi sebanyak-banyaknya dari *user* sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh *user* tersebut. Tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan sistem. Dokumen inilah yang akan menjadi acuan sistem *analis* untuk menterjemahkan kedalam bahasa pemograman.

b. *Design*

Proses desain akan menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat pengkodean (*Coding*). Proses ini berfokus pada struktur data, arsitektur perangkat lunak, representasi *interface,* dan *detail* (*algoritma*) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirment*. Dokumen inilah yang akan digunakan *programmer* untuk melakukan aktivitas pembuatan sistemnya.

c. *Coding*

*Coding* merupakan penerjemahan *design* dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh *programmer* yang akan

meterjemahkan transaksi yang diminta oleh *user*. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem.

Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat tadi. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki.

d. *Testing*

Tahapan ini bisa dikatakan *final* dalam pembuatan sebuah sistem. Setelah melakukan *analysis*, *design* dan pengkodean maka sistem yang sudah jadi akan digunakan oleh *user*.

e. *Maintenance*

Perangkat lunak yang disampaikan kepada pelanggan pasti akan mengalami beberapa perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena program harus menyesuaikan dengan lingkungan (sistem operasi) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

Oleh karena itu, perlu adanya pemeliharaan berupa perbaikan jika ada *error* pada program dan juga beberapa perbaikan jika ada beberapa kekurangan baru yang ditemukan pada program.

Tetapi ada kelebihan dan kelemahan dari penggunaan model ini diantaranya :

**Kelebihan dari *Waterfall* yaitu :**

Keunggulan model pendekatan pengembangan *software* dengan metode *waterfall* adalah pencerminan kepraktisan rekayasa, yang membuat kualitas *software* tetap terjaga karena pengembangannya yang terstruktur dan terawasi. Disisi lain model ini merupakan jenis model yang bersifat dokumen lengkap, sehingga proses pemeliharaan dapat dilakukan dengan mudah.

Dokumentasi kode program yang lengkap juga secara tak langsung menghapus ketergantungan pengembang terhadap pemrogram yang keluar dari tim pengembang. Hal ini sangat menguntungkan bagi pihak pengembang dikarenakan proses pengembangan perangkat lunak tetap dapat dilanjutkan tanpa bergantung pada pemogram tertentu.

**Kelemahan dari *Waterfall* yaitu :**

Kelemahan pengembangan *software* dengan metode *waterfall* yang utama adalah lambatnya proses pengembangan perangkat lunak. Dikarenakan prosesnya yang satu persatu dan tidak bisa diloncat-loncat menjadikan model klasik ini sangat memakan waktu dalam pengembangannya.

Disisi lain, pihak *clien* tidak dapat mencoba sistem sebelum sistem benar-benar selesai pembuatannya. Kelemahan yang lain adalah kinerja personil yang tidak optimal dan efisien karena terdapat proses menunggu suatu tahapan selesai terlebih dahulu.

### 3.2.3.3 Alat bantu Analisis dan Perancangan

Mengembangkan sebuah sistem informasi ini, penulis memakai beberapa alat bantu pembuatan model analisis dan perancangan sistem informasi yaitu :

a. *Use Case Diagram*

*Use case* adalah rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. *Use case* digunakan untuk membentuk tingkah laku benda dalam sebuah mode serta direalisasikan oleh sebuah *collaborator*, umumnya *use case* digambarkan dengan sebuah *elips* dengan garis yang solid, biasanya mengandung nama. *Use case* menggambarkan proses sistem dan juga kebutuhan sistem dari sudut pandang *user*.

b. *Activity Diagram*

*Activity Diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity Diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity Diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*).

Oleh karena itu *Activity Diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dipakai pada *business modeling* untuk memperlihatkan urutan aktifitas proses bisnis.

Struktur diagram ini mirip *Flowchart* atau *Data Flow Diagram* pada perancangan terstruktur. Sangat bermanfaat apabila kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan. *Activity Diagram* dibuat berdasarkan sebuah atau beberapa *use case* pada *use case* diagram.

c. *Collaboration Diagram*

*Collaboration Diagram* menggambarkan kolaborasi (kerja sama) yang dilakukan oleh kelas-kelas pada sistem.

d. *Sequence Diagram*

*Sequence diagram* adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, tampilam, dan sebagainya berupa pesan.

*Sequence diagram* digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian (*event)* untuk menghasilkan *output* tertentu.

*Sequence diagram* diawali dari apa yang me-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara *internal* dan *output* apa yang dihasilkan.

Diagram ini secara khusus berasosiasi dengan *use case diagram*. *Sequence diagram* juga memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam *use case*. *Sequence diagram* juga dapat merubah *atribut* atau *method* pada *class* yang telah dibentuk oleh *class diagram*, bahkan menciptakan sebuah *class* baru. *Sequence diagram* memodelkan aliran logika dalam sebuah sistem dalam cara yang visual.

e. *Class Diagram*

*Class diagram* digunakan untuk menampilkan kelas-kelas dan paket-paket di dalam sistem. *Class diagram* memberikan gambaran sistem secara statis dan relasi antar mereka. Biasanya, dibuat beberapa *class diagram* untuk sistem tunggal. Beberapa diagram akan menampilkan subset dari kelas-kelas dan relasinya. Dapat dibuat beberapa diagram sesuai dengan yang diinginkan untuk mendapatkan gambaran lengkap terhadap sistem yang dibangun.

*Class diagram* adalah alat perancangan terbaik untuk tim pengembang. Diagram tersebut membantu pengembang mendapatkan struktur sistem sebelum kode ditulis, dan membantu untuk memastikan bahwa sistem adalah desain terbaik.

f. *Package Diagram*

*Package Diagram* utamanya digunakan untuk mengelompokkan elemen diagram *UML* yang berlainan secara bersama-sama ke dalam tingkat pembangunan yang lebih tinggi yaitu berupa sebuah paket. *Package diagram* pada dasarnya adalah *class diagram* yang hanya menampilkan paket, disamping kelas, dan hubungan ketergantungan, disamping hubungan khas yang ditampilkan pada *class diagram*.

g. *Component Diagram*

*Component diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*).

h. *Deployment Diagram*

*Deployment diagram* menggambarkan *detail* bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisikal.

### 3.2.4 Pengujian Perangkat Lunak (*Software*)

Pengujian sistem merupakan bagian yang sangat penting dalam siklus pembangunan perangkat lunak. Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kelemahan dari perangkat lunak. Tujuan dari pengujian ini adalah untuk menjamin bahwa perangkat lunak yang dibangun memiliki kualitas yang handal.

Dalam tahap pengujian ini penulis menggunakan pengujian *black box*. Alasannya karena pengujian *black box* dapat mengetahui apakah perangkat lunak yang dibuat dapat berfungsi dengan benar dan telah sesuai dengan yang diharapkan. Pengujian *Black Box* digunakan untuk menguji fungsi-fungsi khusus perangkat lunak yang dirancang. Kebenaran perangkat lunak yang diuji hanya dilihat berdasarkan keluaran yang dihasilkan dari data atau kondisi masukan yang diberikan untuk fungsi yang ada tanpa melihat bagaimana proses untuk mendapatkan keluaran tersebut.

Dengan demikian, pengujian *Black Box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi Input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *Black Box* berusaha menemukan kesalahan dalam kategori sebagai berikut:

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses *database eksternal*.
4. Kesalahan kinerja.

## 3.3 Analisis Sistem Yang Berjalan

Analisis sistem yang berjalan bertujuan untuk mengetahui lebih jelas bagaimana cara kerja sistem tersebut dan masalah yang dihadapi sistem untuk dijadikan landasan usulan perancangan analisis sistem yang sedang berjalan yang dilakukan berdasarkan urutan kejadian yang ada agar sistem yang dibuat menghasilkan *output* yang diinginkan dan mencapai tujuan yang direncanakan.

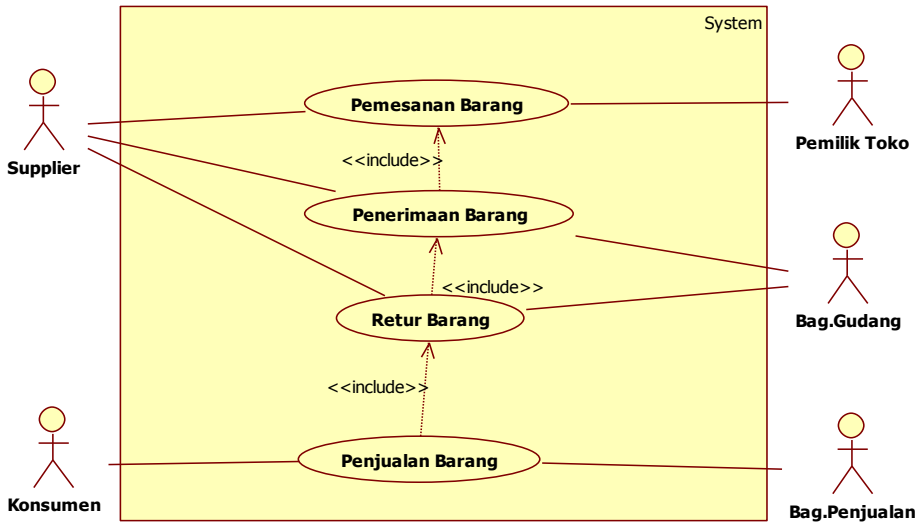
Analisis yang berjalan di toko Underground Workshop ini terdiri dari *Use Case Diagram*, *Skenario Diagram* dan *Activity Diagram*.

### 3.3.1 Analisa Prosedur Yang Berjalan

Sebelum membuat suatu sistem, ada baiknya untuk terlebih dahulu melakukan analisis prosedur serta mengetahui kebutuhan-kebutuhan apa saja yang diperlukan. Analisis ini menggambarkan secara sistematis kegiatan atau aktivitas-aktivitas yang terjadi dalam proses penjualan dan pembelian barang.

#### 3.3.1.1 *Use Case Diagram* Yang Sedang Berjalan

*Use case diagram* menggambarkan siapa saja aktor yang melakukan prosedur dalam sistem serta fungsi-fungsi (proses) yang terlibat dalam sistem tersebut. Berikut *use case diagram* yang berjalan saat ini pada proses Penjualan dan Pembelian Barang yang terdapat pada Toko Underground Workshop:




**Gambar 3.3 *Use Case Diagram* Yang Sedang Berjalan**

#### 3.3.1.1.1 Defenisi Aktor dan Deskripsi

Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat maupun diluar sistem informasi yang akan dibuat itu sendiri, tapi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda diawal *frase* nama aktor.

**Tabel 3.1** Definisi Aktor dan Deskripsi

| No. | Nama | Deskripsi |
|---|---|---|
| 1 | Pemilik Toko | Pihak yang dapat melakukan proses pembelian barang ke *supplier*. |
| 2 | Suplier | Pihak atau tempat yang dimana Pemilik dapat membeli barang. |
| 3 | Bagian Gudang | Pihak yang menerima barang dari *supplier* dan mengecek kondisi barang. |
| 4 | Bagian Penjualan | Pihak yang melayani kebutuhan Konsumen. |
| 5 | Konsumen | Pihak yang membeli barang di toko. |

#### 3.3.1.2 *Skenario Diagram*

*Skenario diagram* adalah suatu fungsi untuk menjelaskan lebih detail mengenai elemen-elemen yang ada di dalam *Use Case Diagram.*

##### 3.3.1.2.1 *Skenario Diagram* Pemesanan Barang.

*Skenario Diagram* pemesanan barang dalam sistem yang berjalan adalah sebagai berikut:

**Tabel 3.2** *Skenario Diagram* pemesanan barang yang berjalan

| | | |
|---|---|---|
| No | 1 | |
| Nama | Pemesanan Barang. | |
| Tujuan | Pemilik toko memesan barang yang dibutuhkan. | |
| Aktor | Pemilik toko dan *Supplier*. | |
| Deskripsi | Melakukan proses pemesanan barang. | |
| Skenario | | |
| Kondisi Awal | Pemilik Toko menghubungi *supplier* untuk memesan barang. | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Pemilik toko memesan barang ke *Supplier*. | | |
| | | 2. *Supplier* mencatat barang yang dipesan dan menghitung total harga. |
| | | 3. *Supplier* memberikan nomor rekening untuk melakukan pembayaran. |
| Kondisi Akhir | Pemilik toko membayar total harga barang yang dibeli melalui transfer ke rekening *Supplier*. | |

#### 3.3.1.2.2 *Skenario Diagram* Penerimaan Barang

*Skenario Diagram* penerimaan barang dalam sistem yang berjalan adalah sebagai berikut:

**Tabel 3.3** *Skenario Diagram* penerimaan barang yang berjalan

| | | |
|---|---|---|
| No | 2 | |
| Nama | Menerima Barang. | |
| Tujuan | Bagian Gudang menerima barang. | |
| Aktor | Supplier dan Bagian Gudang. | |
| Deskripsi | Melakukan proses penerimaan barang yang sedang berjalan. | |
| Skenario | | |
| Kondisi Awal | Supplier mengirimkan barang beserta nota. | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Bagian gudang menerima barang beserta nota dari supplier. | | |
| 2. Bagian gudang mengecek kondisi barang. | | |
| | | 3. Jika kondisi barang rusak, maka akan masukkan ke daftar Retur Barang. |
| | | 4. Jika kondisi barang bagus maka akan didisplay di toko |
| Kondisi Akhir | Barang dijual di toko | |

#### 3.3.1.2.3 *Skenario Diagram* Retur Barang.

*Skenario Diagram* retur barang dalam sistem yang berjalan adalah sebagai berikut:

**Tabel 3.4** *Skenario Diagram* retur barang yang berjalan

| | | |
|---|---|---|
| No | 3 | |
| Nama | Retur Barang. | |
| Tujuan | Bag.Gudang melakukan retur barang yang rusak ke *supplier*. | |
| Aktor | Bagian Gudang dan *Supplier*. | |
| Deskripsi | Melakukan proses retur barang rusak yang sedang berjalan. | |
| Skenario | | |
| Kondisi Awal | Bagian gudang mendapati barang yang rusak. | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Bagian gudang mencatat data barang yang rusak. | | |
| 2. Bagian gudang mengembalikan barang yang rusak ke *supplier*. | | |
| 3. *Supplier* menerima barang yang rusak. | | |
| 4. *Supplier* mengganti barang yang rusak dengan yang baru. | | |
| Kondisi Akhir | *Supplier* mengirimkan barang. | |

#### 3.3.1.2.4 *Skenario Diagram* Penjualan Barang.

*Skenario Diagram* penjualan barang dalam sistem yang berjalan adalah sebagai berikut:

**Tabel 3.5** *Skenario Diagram* penjualan barang yang berjalan

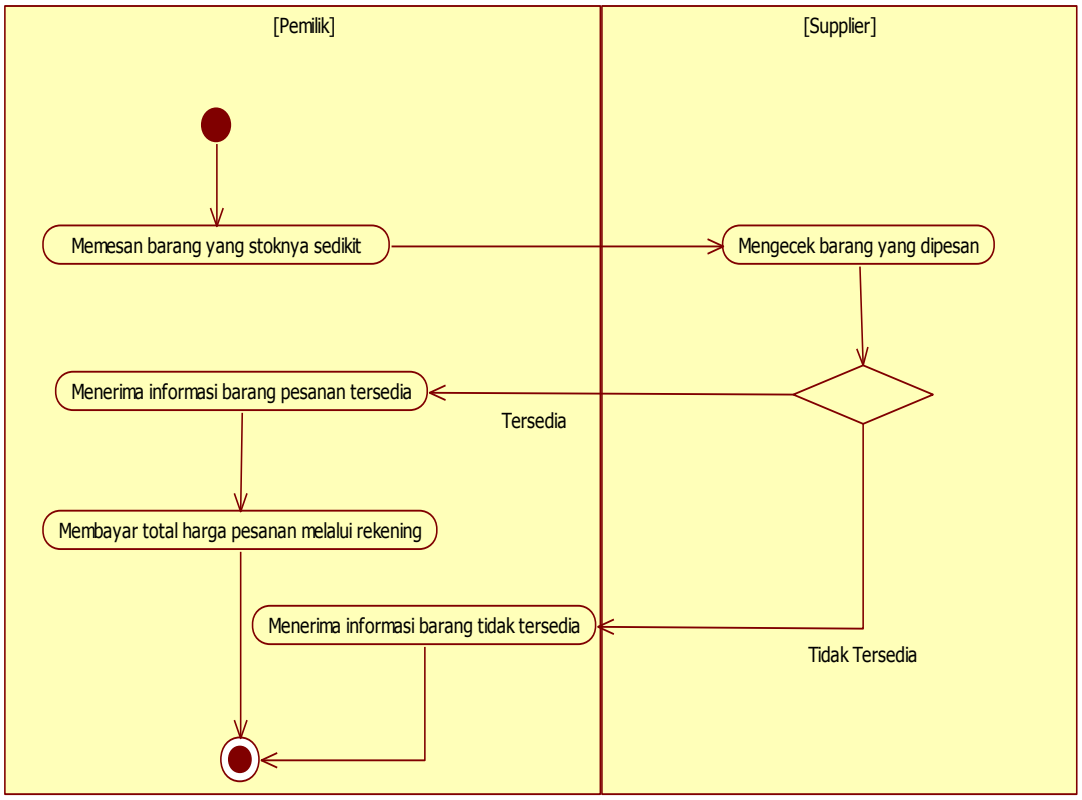
| | | |
|---|---|---|
| No | 4 | |
| Nama | Penjualan Barang. | |
| Tujuan | Bag.Penjualan melakukan transaksi penjualan. | |
| Aktor | Bagian Penjualan dan Konsumen. | |
| Deskripsi | Melakukan proses penjualan barang yang sedang berjalan. | |
| Skenario | | |
| Kondisi Awal | Konsumen mengunjungi toko. | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Bagian Penjualan menawarkan barang yang dijual. | | |
| | | 2. Konsumen mencari memilih barang yang diinginkan. |
| 3. Bagian Penjualan menghitung total harga barang yang dibeli oleh konsumen. | | |
| | | 4. Konsumen membayar total harga dari barang yang dibeli. |
| 5. Menerima uang pembayaran. | | |
| 6. Bagian penjualan memberikan barang ke konsumen. | | |
| | | 7. Konsumen menerima barang. |
| Kondisi Akhir | Bagian penjualan memasukkan data penjualan pada buku laporan penjualan barang. | |

### 3.3.1.3 *Activity Diagram*

*Activity Diagram* digunakan untuk menggambarkan kegiatan-kegiatan yang ada di dalam suatu sistem, dimana merupakan penggambaran aktivitas dari *case* yang ada pada *Use Case diagram*. Agar dapat lebih memahami tentang sistem yang akan dibuat, maka perlu dibuatkan *Activity Diagram* seperti berikut:

#### 3.3.1.3.1 *Activity Diagram* Pemesanan Barang.

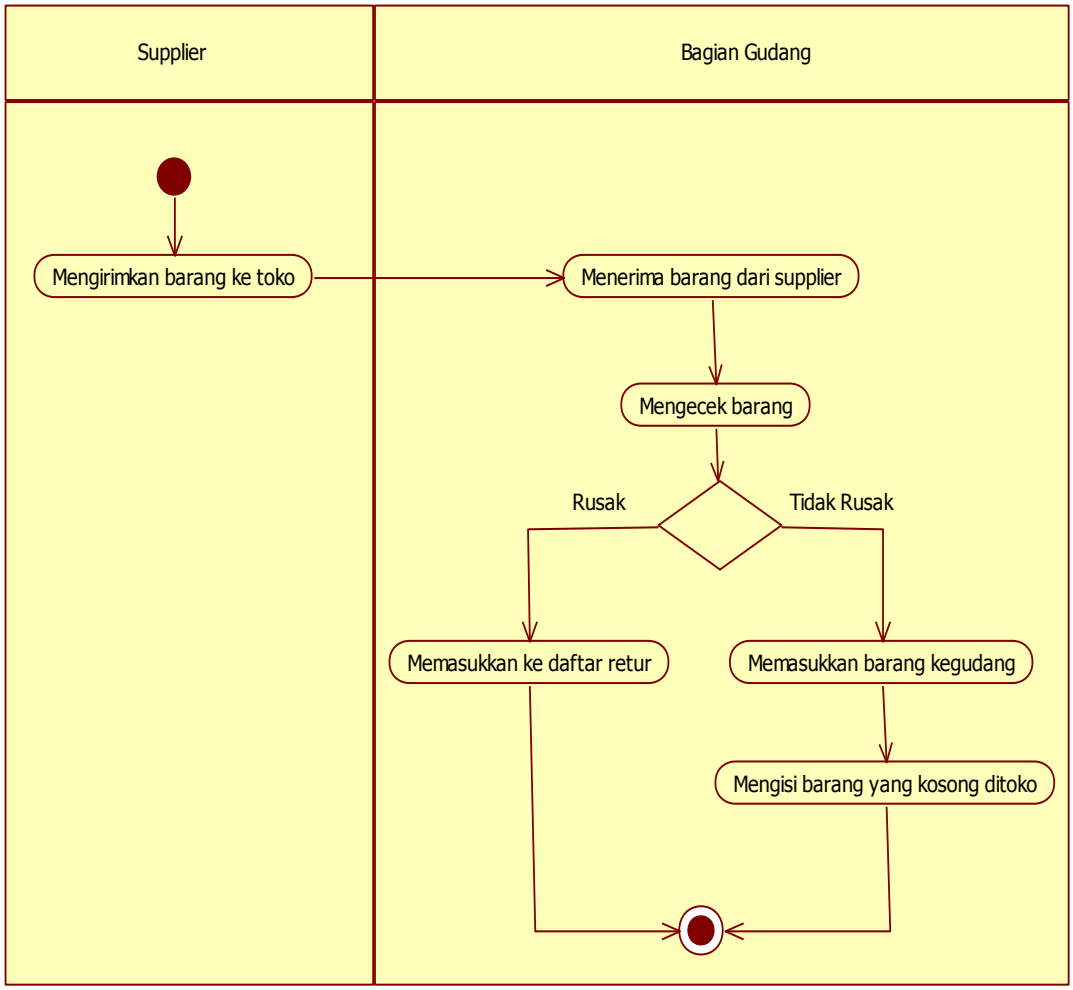
*Activity Diagram* pemesanan barang dalam sistem yang berjalan adalah sebagai berikut:




**Gambar 3.4 *Activity Diagram* Pemesanan Barang**

#### 3.3.1.3.2 *Activity Diagram* Penerimaan Barang

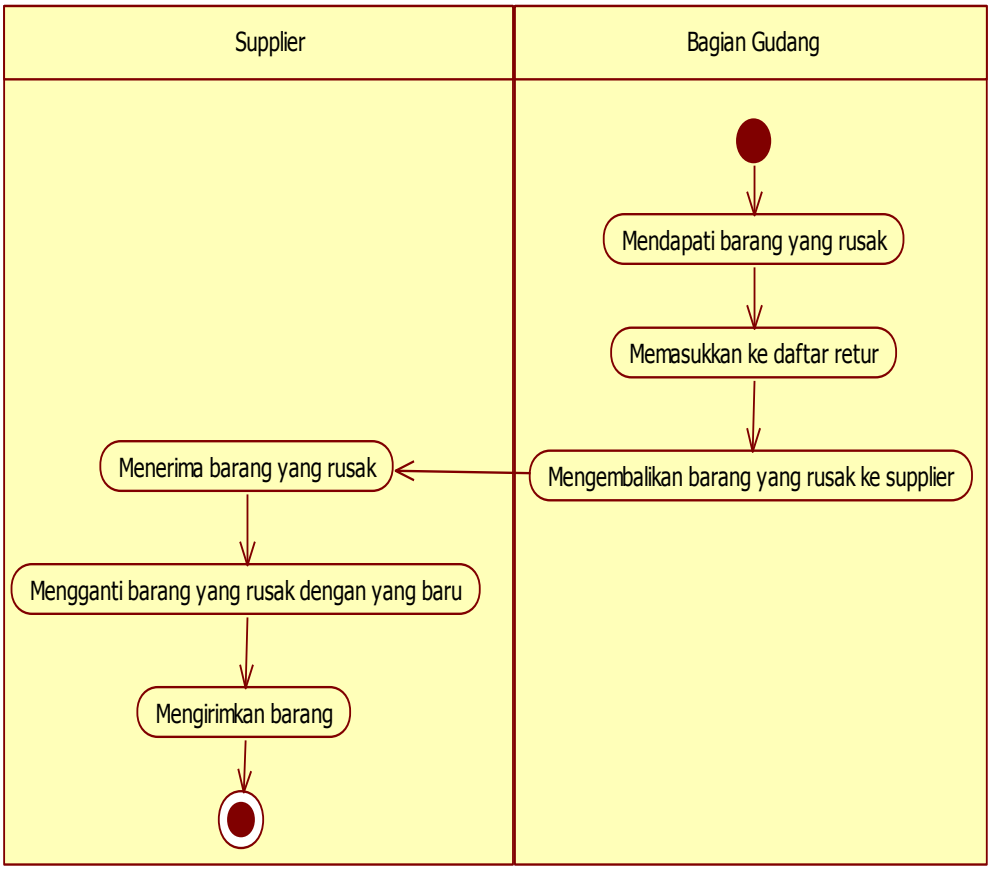
*Activity Diagram* penerimaan barang dalam sistem yang berjalan adalah sebagai berikut:




**Gambar 3.5 *Activity Diagram* Penerimaan Barang**

### 3.3.1.3.3 *Activity Diagram* Retur Barang.

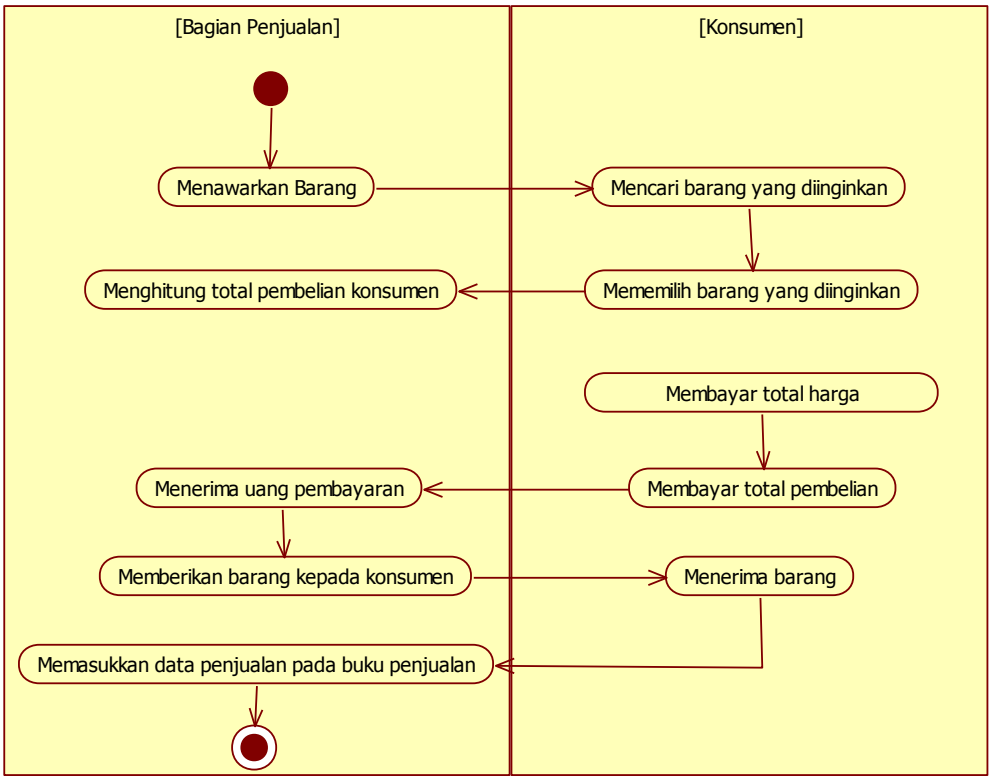
*Activity Diagram* retur barang dalam sistem yang berjalan adalah sebagai berikut:




**Gambar 3.6 *Activity Diagram* Retur Barang**

### 3.3.1.3.4 *Activity Diagram* Penjualan Barang.

*Activity Diagram* penjualan barang dalam sistem yang berjalan adalah sebagai berikut:




**Gambar 3.7 *Activity Diagram* Penjualan Barang**

### 3.1.2 Evaluasi Sistem Yang Berjalan

Setelah dilakukan analisis terhadap sistem yang sedang berjalan, maka selanjutnya penulis mengevaluasi sistem yang berjalan, evaluasi tersebut penulis sajikan melalui tabel berikut:

**Tabel 3.5 Evaluasi Sistem Yang Berjalan.**

| Masalah | Solusi |
|---|---|
| Belum tersedianya media informasi untuk data *supply* stok barang dan data retur pembelian, sehingga bagian gudang kesulitan dalam mengolah data *supply* stok barang dan data retur barang ke *supplier*. | Menyediakan layanan untuk pengolahan data stok dan retur barang pada sistem yang diusulkan, agar bagian gudang mudah mengolah data stok dan retur barang. |
| Proses transaksi pembelian dan penjualan barang masih menggunakan media tertulis, yaitu dengan cara mencatat kedalam buku kas, sehingga belum terintegrasi. | Menyediakan layanan untuk transaksi pembelian dan penjualan, agar proses transaksi pembelian dan penjualan dapat terintegrasi. |
| Kurang efektifnya dalam pembuatan laporan transaksi pembelian dan penjualan barang. | Menyediakan layanan cetak laporan pada sistem yang baru, agar memudahkan dalam menghasilkan laporan dari transaksi pembelian dan penjualan. |